

Desenvolvimento de uma plataforma de gestão da construção

Luiz Henrique de Barros de Oliveira

Dissertação apresentada à Escola Superior de Tecnologia e Gestão para obtenção do grau de mestre em Informática no âmbito da dupla diplomação com a Universidade Tecnológica Federal do Paraná

Orientadores:

Prof. Dr. Paulo Alexandre Vara Alves

Prof. Dr. Rafael Liberato Roberto

Bragança

Junho 2024

Desenvolvimento de uma plataforma de gestão da construção

Luiz Henrique de Barros de Oliveira

Dissertação apresentada à Escola Superior de Tecnologia e Gestão para obtenção do
grau de mestre em Informática no âmbito da dupla diplomação com a Universidade
Tecnológica Federal do Paraná

Orientadores:

Prof. Dr. Paulo Alexandre Vara Alves

Prof. Dr. Rafael Liberato Roberto

Bragança

Junho 2024

Dedicatória

Dedico este trabalho a Deus, cuja infinita sabedoria tem iluminado meu caminho e guiado cada passo da minha jornada. À minha mãe, Juraci Fatima de Barros, cujo apoio incondicional, inspiração constante e exemplo inabalável tornou tudo possível. Também dedico esta conquista ao meu irmão gêmeo, Luiz Paulo de Barros de Oliveira, e ao meu tio, Milton Lopes de Barros, que sempre foram fontes de inspiração e força inestimáveis para mim.

Estendo esta dedicatória a todos que, assim como eu, acreditam em seus sonhos e se empenham com determinação para torná-los realidade. Que este trabalho sirva como um testemunho de que, com fé, resiliência e dedicação, qualquer objetivo pode ser alcançado.

Agradecimentos

Inicialmente, agradeço a Deus por me conceder força e sabedoria para enfrentar desafios e alcançar meus objetivos.

Gostaria de expressar minha mais profunda gratidão a todos que contribuíram para a conclusão deste trabalho e para o meu percurso acadêmico. Um agradecimento especial ao valioso apoio financeiro da ATTRACT, que proporcionou melhores condições durante minha jornada.

Sou imensamente grato às instituições de ensino IPB (Instituto Politécnico de Bragança) e UTFPR (Universidade Tecnológica Federal do Paraná) pela oportunidade de participar do programa de dupla diplomação. Esta experiência foi enriquecedora e abriu novos horizontes na minha formação acadêmica e profissional. Ao meu orientador, Prof. Dr. Paulo Alexandre Vara Alves, e coorientador, Prof. Dr. Rafael Liberato Roberto, meus sinceros agradecimentos. Sua orientação, paciência e conhecimento foram essenciais para a conclusão deste trabalho.

Estendo meus agradecimentos aos professores da UTFPR - Campo Mourão e do IPB pela qualidade do ensino, pelo valioso apoio e pela transmissão de conhecimento.

Minha gratidão à minha família é imensurável; seu apoio inabalável durante toda a minha jornada acadêmica, tanto no Brasil quanto em Portugal, foi fundamental.

Não poderia deixar de agradecer aos meus colegas de turma, em especial Daniel Augusto Rodrigues Farina, que compartilhou comigo esta jornada, e aos meus amigos, pelo apoio incondicional e incentivo constante.

Por fim, agradeço a todos que, de alguma forma, contribuíram para a conclusão deste

trabalho. Cada palavra de apoio, cada conselho e cada gesto de incentivo foram fundamentais para esta conquista. Muito obrigado a todos.

Abstract

The work presents the development of a construction management platform for the company Abel Luís Nogueiro e Irmãos Lda (ALN), an SME based in Bragança, Portugal, specialized in civil construction. The motivation for this project arose from the obsolescence of the current web platform, created in collaboration with Instituto Politécnico de Bragança (IPB) in 2013. This platform, built with technologies that are now outdated due to lack of maintenance, resulted in the loss of functionality and reduced efficiency operational.

With the aim of modernizing the existing system and introducing new functionalities, development was conducted using agile methodologies, such as Scrum and Kanban. The process involved a detailed analysis of requirements, creation of mockups and implementation using Docker and Docker Compose. We opted for modern technologies, such as .Net Core (version 7) for building the API and ReactJS (version 18) for developing the platform interface. In addition to having good performance, it guarantees compatibility with current browsers, improves the user experience and contributes to the scalability necessary to meet the company's future demands. Comparison with existing solutions on the market was crucial to ensure the resulting platform was relevant and effective for the construction industry.

Development was guided by user stories and regular meetings to ensure alignment with ALN needs and to present results as they progressed. This significantly contributed to the successful development and deployment of the platform within the company. Currently, the platform offers features such as recording the presence of employees at construction sites, issuing invoices, launching consumables and services, as well as cost control and

issuing budget execution for each project, allowing detailed monitoring of progress.

The new platform not only met the initial requirements, but also provided a solid foundation for future expansions and improvements. Furthermore, this project serves as a model for other companies in the sector seeking to modernize their management systems. The development of this platform represents a significant advance for the company ALN, enabling more efficient and integrated management of its construction projects and establishing a clear path for continuous improvements and adaptation to market demands. Currently, the construction management platform is in the testing phase at ALN and is open to receiving new feedback. In conclusion, this work highlights the benefits of technological innovation, adaptation to market needs and the use of agile approaches to drive company growth and competitiveness in the construction sector.

Keywords: ERP System, Management System, Construction Management, .NET, ReactJs, Agile Methodologies, Budget Execution, Invoice Control, Financial Control, Civil Construction.

Resumo

O trabalho apresenta o desenvolvimento de uma plataforma de gestão da construção para a empresa Abel Luís Nogueiro e Irmãos Lda (ALN), uma PME sediada em Bragança, Portugal, especializada em construção civil. A motivação para este projeto surgiu da obsolescência da plataforma web atual, criada em colaboração com o IPB em 2013. Esta plataforma, construída com tecnologias agora desatualizadas devido à falta de manutenção, resultou na perda de funcionalidades e na redução da eficiência operacional.

Com o objetivo de modernizar o sistema existente e introduzir novas funcionalidades, o desenvolvimento foi conduzido utilizando metodologias ágeis, como Scrum e Kanban. O processo envolveu uma análise detalhada dos requisitos, a criação de mockups e a implementação utilizando Docker e Docker Compose. Optou-se por tecnologias modernas, como .Net Core (versão 7) para a construção da API e ReactJS (versão 18) para o desenvolvimento da interface da plataforma. Além de possuir uma boa performance, garante a compatibilidade com os navegadores atuais, melhorar a experiência do usuário e contribui para a escalabilidade necessária para atender às futuras demandas da empresa. A comparação com soluções existentes no mercado foi crucial para garantir que a plataforma resultante fosse relevante e eficaz para a indústria da construção.

O desenvolvimento foi guiado pelas histórias de usuários e reuniões regulares para garantir o alinhamento com as necessidades da ALN e para apresentar os resultados à medida que avançavam. Isso contribuiu significativamente para o desenvolvimento e implantação bem-sucedidos da plataforma na empresa. Atualmente, a plataforma oferece funcionalidades como registro de presença dos colaboradores nas obras, emissão de faturas, lançamento de consumíveis e serviços, além de controle de custos e emissão de execução

orçamental para cada obra, permitindo o acompanhamento detalhado do progresso.

A nova plataforma não apenas atendeu aos requisitos iniciais, mas também forneceu uma base sólida para futuras expansões e melhorias. Além disso, este projeto serve como um modelo para outras empresas do setor que buscam modernizar seus sistemas de gestão. O desenvolvimento desta plataforma representa um avanço significativo para a empresa ALN, possibilitando uma gestão mais eficiente e integrada de seus projetos de construção e estabelecendo um caminho claro para melhorias contínuas e adaptação às demandas do mercado. Atualmente, a plataforma de gestão da construção está em fase de teste na empresa ALN e está aberta para receber novos feedbacks. Em conclusão, este trabalho destaca os benefícios da inovação tecnológica, da adaptação às necessidades do mercado e do uso de abordagens ágeis para impulsionar o crescimento e a competitividade da empresa no setor da construção.

Palavras-chave: Sistema ERP, Sistema de Gestão, Gestão de Obras, .NET, ReactJs, Metodologias Ágeis, Execução Orçamental, Controle de Faturas, Controle Financeiro, Construção Civil.

Conteúdo

1	Introdução	1
2	Estado da arte	5
2.1	Sistemas relacionados	5
2.1.1	Software Orçamentação	6
2.1.2	Cegid Primavera	7
2.1.3	CentralGest	7
2.1.4	PHC	8
2.1.5	Comparação entre os sistemas	9
3	Análise de Requisitos e Interface	13
3.1	Análise de projeto	14
3.1.1	História de usuário (<i>User Story</i>)	15
3.2	Interface e Experiência do Usuário (UI/UX)	16
3.2.1	Definição de cores	17
3.2.2	Componentes	18
3.2.3	Páginas	21
4	Desenvolvimento	27
4.1	Arquitetura do sistema	29
4.2	Banco de dados	31
4.2.1	Utilizador, Log, Parâmetros e Feriado	31

4.2.2	Colaborador	34
4.2.3	Fornecedor, Encomenda, Consumível e Pedido de Cotação	36
4.2.4	Faturas, Material e Unidade	40
4.2.5	Cliente, Obra e Nota de Crédito	42
4.2.6	Equipamento e Viatura	45
4.2.7	Seguro, Valores Agregados e Padronização	48
4.3	Desenvolvimento do Servidor Backend (API)	50
4.3.1	Estrutura de diretório	51
4.3.2	Controladores	54
4.3.3	Serviços	58
4.3.4	Execução Orçamental	60
4.4	Desenvolvimento do Frontend	60
4.4.1	Funcionalidades	65
4.4.2	Adicionar e Atualizar por Etapas	67
5	Resultados	71
6	Conclusão	73
A	Histórias de Usuários	A1
B	Modelagem da Interface	B1
C	Documentação - Swagger	C1
D	Paginas da aplicação web	D1

Lista de Tabelas

2.1	Tabela de comparação das ferramentas.	9
3.1	Exemplo de histórias de usuários	16
4.1	Principais tecnologias do frontend e do backend.	28
4.2	Principais ferramentas utilizadas no projeto.	29
4.3	Métodos http da biblioteca MVC do ASP.NET Core	57

Lista de Figuras

3.1	Paleta de cores da interface	18
3.2	<i>Mockup</i> dos componentes de gráficos e listagem e dados.	19
3.3	Componentes de conjunto de gráficos e listagem de presenças	20
3.4	Componentes de <i>input</i> de dados, alerta e listagem.	20
3.5	Componentes de listagem e formulário de cadastro e edição.	21
3.6	Página de listagem de presenças na versão <i>dark</i>	22
3.7	Página de listagem de presenças na versão <i>light</i>	22
3.8	<i>Mockup</i> da página de listagem de viaturas.	23
3.9	<i>Mockup</i> da página de adicionar uma viatura.	24
3.10	<i>Mockup</i> da página de editar uma viatura.	24
3.11	<i>Mockup</i> da página de deletar uma viatura.	25
4.1	Arquitetura da aplicação	30
4.2	Tabelas do banco de dados referente aos recursos de utilizador, log, parâmetros e feriados.	32
4.3	Tabelas do banco de dados referente aos recursos do colaborador.	35
4.4	Tabelas do banco de dados referente aos recursos da tabela "fornecedor", como: encomenda, consumível e pedido de cotação.	38
4.5	Tabelas do banco de dados referente aos recursos sobre as faturas e materiais.	41
4.6	Tabelas do banco de dados referente aos recursos de cliente, nota de crédito e obras.	43
4.7	Tabelas do banco de dados referente aos recursos de equipamento e viatura.	46

4.8	Tabelas do banco de dados referente aos recursos de seguro, valores agregados (juros, iva, subsidio de alimentação) e customizações dos dados da empresa.	49
4.9	Estrutura de diretório do projeto do backend.	52
4.10	Controlador do pedido de cotação.	56
4.11	Serviço do módulo de pedido de cotação.	59
4.12	Estrutura de Diretório do Frontend	61
4.13	Menus das funcionalidades do frontend	66
4.14	Formulário para realizar um pedido de cotação	69
D.1	Página - Equipamentos e Serviços / Materiais	D1
D.2	Página - Equipamentos e Serviços / Consumíveis e Serviços	D2
D.3	Página - Obra / Obra	D2
D.4	Página - Obra / Execução Orçamental	D3
D.5	Página - Obra / Presenças	D3
D.6	Página - Obra / Preço Material	D4
D.7	Página - Obra / Material por Obra	D4
D.8	Página - Obra / Fatura Material	D5
D.9	Página - Obra / Pedido Cotação	D5
D.10	Página - Obra / Fatura Cliente	D6
D.11	Página - Outros custos / IUC	D6
D.12	Página - Outros custos / Inspeção Periódica	D7
D.13	Página - Outros custos / Leasing Viaturas	D7
D.14	Página - Outros custos / Leasing Equipamento	D8
D.15	Página - Outros custos / Seguro Automóvel	D8
D.16	Página - Outros custos / Seguro Trabalho	D9
D.17	Página - Outros custos / Consumíveis e serviços	D9

Siglas

AAA Authentication, Authorization and Accounting. xix

ALN Abel Luís Nogueiro & Irmãos Lda. xix, 14, 23, 31

API Application Programming Interface. xix

CMS Content Management System. xix

DAO Data Access Object. xix

DB Data Base. xix

DOM Document Object Model. xix

ERP Enterprise Resource Planning. xix, 5, 73

ESTiG Escola Superior de Tecnologia e Gestão. xix

HTTP HyperText Transfer Protocol. xix, 29

ID Identificador. xix

IP Protocolo de rede. xix

IPB Instituto Politécnico de Bragança. ix, xi, xix, 1, 14

IUC Imposto Único de Circulação. xix

IVA Imposto sobre Valor Acrescentado. xix, 37

NIF número de identificação fiscal. xix, 37

PME pequena e média empresa. xix

PME's pequenas e médias empresas. xix, 6

SGDB Sistema de Gerenciamento de Banco de Dados. xix, 31

Capítulo 1

Introdução

No cenário atual das empresas de construção civil, a gestão eficiente de obras e processos administrativos é crucial para assegurar a competitividade e sustentabilidade do negócio. A ALN - Abel Luís Nogueiro e Irmãos Lda, uma PME situada em Bragança, Portugal, é especializada na execução, reabilitação e remodelação de obras públicas e construções particulares.

Em 2013, uma parceria entre a ALN e o IPB resultou no desenvolvimento de uma plataforma web de gestão de processos empresariais. Esta ferramenta, essencial para a eficiência e organização dos processos da empresa, substituiu o uso de planilhas eletrônicas. No entanto, o avanço tecnológico dos navegadores e a obsolescência das tecnologias utilizadas na plataforma têm gerado desafios significativos, incluindo a perda de funcionalidades e impacto negativo na eficiência das operações. Estes problemas destacam a necessidade de desenvolver uma nova plataforma com tecnologias modernas para resolver incompatibilidades, incorporar novas funcionalidades e melhorar a experiência do usuário.

O objetivo geral deste projeto é reconstruir o sistema de gestão de processos da ALN, utilizando tecnologias modernas para garantir compatibilidade com navegadores atuais, melhorar a experiência do usuário e adicionar funcionalidades que otimizem a gestão de obras e processos administrativos.

Os objetivos específicos incluem:

1. Atualizar a tecnologia do sistema para garantir compatibilidade com navegadores modernos.
2. Adicionar funcionalidades como envio de e-mails e utilização de APIs para validação de dados.
3. Melhorar a experiência do usuário através de uma interface intuitiva.
4. Utilizar metodologias ágeis no desenvolvimento, como Scrum e Kanban.
5. Implementar a integração e implantação do sistema utilizando Docker e Docker Compose.

Existem diversas soluções no mercado que abordam a gestão de processos administrativos e de obras, como Orçamentação, Cegid Primavera, CentralGest e PHC. Essas soluções oferecem funcionalidades variadas que podem incluir gestão financeira, controle de estoque, planejamento de projetos e relatórios detalhados. A análise dessas soluções fornece uma base comparativa importante para o desenvolvimento do novo sistema da ALN.

As principais contribuições deste trabalho incluem a atualização tecnológica do sistema de gestão da ALN, a adição de novas funcionalidades e a melhoria da experiência do usuário. Além disso, a utilização de metodologias ágeis e a implementação com Docker e Docker Compose representam avanços significativos na forma como a empresa gerencia e implanta seu software.

Os demais capítulos desta dissertação seguem uma organização estruturada. No Capítulo 2, é apresentado o Estado da Arte, comparando soluções semelhantes, tais como Orçamentação, Cegid Primavera, CentralGest e PHC. Em seguida, no Capítulo 3, são apresentados detalhes sobre a Análise de Requisitos, abrangendo as histórias de usuário e o processo de elaboração dos mockups, os quais foram desenvolvidos utilizando a ferramenta Figma. No Capítulo 4, são descritos o Desenvolvimento do projeto, abordando a estrutura do banco de dados, o desenvolvimento do frontend e o desenvolvimento do

backend. Os resultados obtidos neste projeto são apresentados no Capítulo 5. Por fim, no Capítulo 6, são apresentadas as Conclusões do trabalho.

Capítulo 2

Estado da arte

Este capítulo apresenta o estado da arte, onde foi construído um estudo base para entender as necessidades e os desafios encontrado por uma empresa no processo de aquisição de um sistema e também identificar algumas opções de sistemas disponíveis no mercado, assim como suas funcionalidades.

2.1 Sistemas relacionados

A busca pela automatização surge através da necessidade em tornar processos mais eficientes, reduzir erros e liberar recursos para atividades mais estratégicas, transformando tarefas manuais, repetitivas e demoradas em processos automáticos e programáveis, melhorando a produtividade, a qualidade e a competitividade da organização [1]–[3].

Software de gestão empresarial ou Enterprise Resource Planning (ERP) auxiliam as empresas em diversas áreas, melhorando o compartilhamento de informações entre os departamentos e o fluxo de trabalho, integrando dados, processos e tecnologia. Além disso, reduz os custos operacionais e podem proporcionar reduções no *lead time* em 60% e 99% no prazo de entrega, obtendo maior confiabilidade nas informações, possibilitando tomadas de decisões mais assertivas [1], [2]. Soluções ERP fazem grande diferença nas rotinas das organizações, oferecendo controle e agilidade em seus processos, tornando uma ferramenta indispensável para seu crescimento e sucesso [1].

A aquisição de um sistema ERP pode ser feita de quatro formas: compra, locação, terceirização ou através do próprio desenvolvimento [4]. Cada modelo possui suas vantagens e desvantagens. A compra de um ERP acarreta um investimento inicial muito elevado e um valor de manutenção inacessível para a maioria das pequenas e médias empresas (PME's)[5].

Na locação, o custo de aquisição é relativamente baixo, porém as customizações são limitadas ou inexistentes. Caso a opção seja terceirização, informações como regras de negócio e/ou informações mais sensíveis estarão disponível fora da empresa. Porém os custo de desenvolvimento são reduzidos, o que não acontece no desenvolvimento interno, que acarretará em alto custo com time de TI qualificado e um tempo elevado no desenvolvimento, por outro lado, o sistema tornará específico para os recursos e processos da empresa [4], [5].

Com intuito de apresentar algumas soluções disponíveis no mercado com foco nas empresas de construção civil, foram selecionadas quatro ferramentas populares para a gestão de obras. Na subseção 2.1.1 é apresentada uma breve introdução sobre o software Orçamentação desenvolvido pela empresa Anturio [6]. Em seguida, a subseção 2.1.2 apresenta a ferramenta Cegid Primavera que pertence ao grupo Cegid [7]. Logo após, na subseção 2.1.3 é abordada a ferramenta CentralGest ERP da CentralGest [8]. Para compor o conjunto de ferramentas, na subseção 2.1.4 trata das funcionalidades do ERP PHC que pertence a PHC [9]. Toda informação sobre as ferramentas estão de acordo com as descrições obtidas no site de cada fornecedor. Nenhum teste empírico foi realizado nessas ferramentas.

2.1.1 Software Orçamentação

A empresa de tecnologia da informação Anturio [6], lançou seu novo software para o setor de construção civil chamado Orçamentação [10]. De acordo com o site da empresa [10], o software apresenta algumas funcionalidades que auxiliam nas tomadas de decisões com base no controle financeiro. Nele é possível executar as tarefas como: Gestão de

orçamento, consultas de encargos, planeamento de obras, cálculo proporcional de outros custos, importar orçamentos de arquivos csv e mapeamento simplificado de mapa de encargos.

No entanto a ferramenta não apresenta funcionalidades para recursos humanos, como por exemplo o lançamento de presenças dos colaboradores na obra, que contribui para aferir os custos de mão de obra por cada obra. A Anturio fornece soluções para outros setores [11] incluindo distribuição, indústria e tecnologias de informação, no entanto essa pesquisa se limita apenas em ferramentas com foco na construção civil.

2.1.2 Cegid Primavera

O grupo Cegid [12] atua nas áreas de finanças, recursos humanos, contabilidade e varejo. Além disso, é detentora do ERP Cegid Primavera, um sistema de gestão dos processos de construção [7]. De acordo com as informações coletadas em seu site [7], o sistema Cegid Primavera compõe um conjunto amplo de recursos, como: gestão administrativa e financeira, gestão de recursos humanos, gestão de materiais, gestão de subempreitadas, planeamento de obras, orçamentação, disponibilização de informação nos estaleiros, relatório legal e fiscal e informação analítica de apoio à decisão.

Com um conjunto de recursos de gestão, o sistema Cegid Primavera demonstra ser uma ferramenta mais completa que a Orçamentação da Anturio. No entanto, não foi realizado nenhum teste na ferramenta comprovar essa afirmação, somente as informações coletadas nos sites das empresas.

2.1.3 CentralGest

O software de construção do CentralGest ERP foi projetado para operações de construção de grandes obras públicas e civis. Com a integração com outros setores da empresa, a ferramenta compõe uma gama de funcionalidades disponíveis para o controle das obras. Além disso, para as organizações que necessitam de orçamentar um projecto e controlar

sua execução, em particular para o mercado da construção civil e obras públicas, a CentralGest desenvolveu o software de orçamentação como uma solução vertical integrada no seu ERP. Além das ferramentas padrão para a gestão corrente da empresa, disponibiliza um conjunto de ferramentas específicas para o setor de construção civil.

A solução de gestão de obras da CentralGest [13] oferece as seguintes funcionalidades: Criação automática da obra a partir do orçamento, reorçamentação da obra, gestão de subempreitada, geração de documentos, planeamento de actividades, planeamento de recursos, autos de medição, gestão de desvios do orçamento, gestão de custos por linha e mapas de desvios de custos.

Já as funcionalidades de orçamentação [14], são: Estrutura de artigos (artigos compostos), tarefas, classificadores, tabela de preços do orçamento, cotações, multivistas da orçamentação, obras, criação de orçamentos através de uma estrutura definida, importação de orçamentos a partir de arquivo csv, gestão de versões de um orçamento, criação de mapas temporais, plano de pagamentos e cronograma financeiro, obtidos automaticamente a partir do mapa temporal, gestão de custos, gestão da margem de lucro, gestão do dossier, definição de propriedades do orçamento, gestão de alvarás e consórcios.

O CentralGest possui uma das ferramentas mais completas para o controle de obras. Além das funcionalidades citadas, possibilita também a gestão de obras remotas e revisão de preços.

2.1.4 PHC

A empresa PHC é uma empresa multinacional no setor de tecnologia da informação. Desde 1993 vem oferecendo várias soluções em diversos setores. Dentre essas soluções, a PHC disponibiliza uma ferramenta para o setor da construção [9]. De acordo com o site da empresa, os recursos encontrados nessa ferramenta são: Gestão administrativa, gestão de obras, gestão financeira e contabilidade, gestão de orçamentos e controle de gastos, documentos digitais.

2.1.5 Comparação entre os sistemas

Cada ferramenta possui um conjunto de funcionalidades como é possível observar na tabela 2.1. Em destaque, nota-se funcionalidades semelhantes entre elas, como recursos para obras, outros custos e orçamentos, podendo indicar os principais requisitos de um software para construção civil. Também é possível observar a ampla diferença na quantidade de funcionalidades entre o Orçamentação e os demais, funcionalidades que podem fazer total diferença nos processos da empresa. Entretanto, um teste empírico poderia identificar funcionalidades não encontrada nas descrições dos sites dos fornecedores.

Funcionalidades	Ferramentas			
	Orçamentação	Cegid Primavera	CentralGest	PHC
Concursos públicos		●		
Controle de Materiais		●	●	●
Controle de encargos	●			
Cotações			●	
Fatura de cliente		●	●	●
Fatura de fornecedor		●	●	●
Gestão administrativa		●	●	●
Gestão financeira		●	●	●
Gestão de obras	●	●	●	●
Gestão de recursos humanos		●	●	●
Gestão de subempreitadas		●	●	
Gestão de outros custos	●	●	●	●
Gestão da margem de lucro		●	●	●
Gestão do dossier			●	
Gestão de alvarás			●	
Importar arquivos CSV	●		●	
Informação dos estaleiros		●	●	
Informações analítica		●	●	●
Orçamentação	●	●	●	●
Relatórios		●	●	●

Tabela 2.1: Tabela de comparação das ferramentas.

Em seguida, foi adicionada uma lista que descreve cada funcionalidade apresentada na tabela 2.1.

- **Concursos Públicos:** Esta funcionalidade abrange desde a identificação de oportunidades de licitação até o acompanhamento do processo completo de participação em concursos públicos. Isso inclui a gestão de documentos necessários, prazos de entrega, requisitos de qualificação e elaboração de propostas competitivas.
- **Controle de Materiais:** Permite o registro detalhado de todos os materiais utilizados em projetos ou operações. Isso envolve o acompanhamento do estoque disponível, a solicitação de compras quando necessário, o recebimento de materiais, o armazenamento adequado e o controle de desperdícios.
- **Controle de Encargos:** Esta funcionalidade ajuda a monitorar e controlar todos os encargos relacionados a projetos ou operações. Isso inclui custos trabalhistas, como salários, benefícios e impostos, bem como outras despesas indiretas, como taxas administrativas, seguros e manutenção.
- **Cotações:** Facilita o processo de solicitação, recebimento e comparação de cotações de fornecedores para produtos ou serviços específicos. Isso pode envolver a criação de solicitações de cotação, o acompanhamento de respostas de fornecedores e a análise de propostas para selecionar a melhor opção em termos de preço, qualidade e prazo de entrega.
- **Fatura de Cliente:** Gerencia todo o ciclo de faturamento para clientes, desde a emissão de faturas até o acompanhamento de pagamentos. Isso inclui a geração de faturas precisas e detalhadas, o envio para os clientes, o registro de pagamentos recebidos e o monitoramento de saldos em aberto.
- **Fatura de Fornecedor:** Similar à funcionalidade de fatura de cliente, mas voltada para o registro e controle das faturas recebidas dos fornecedores. Isso envolve a verificação da precisão das faturas, o registro das despesas incorridas, a programação de pagamentos e o acompanhamento de saldos a pagar.
- **Gestão Administrativa:** Engloba uma variedade de tarefas administrativas essenciais para o funcionamento eficiente da empresa. Isso pode incluir a gestão de

documentos, a organização de agendas, o agendamento de reuniões, a comunicação interna e a manutenção de registros organizacionais.

- **Gestão Financeira:** Esta funcionalidade abrange todas as atividades relacionadas à gestão financeira da empresa. Isso inclui contabilidade, planejamento financeiro, controle de custos, gestão de caixa, análise de viabilidade de projetos e elaboração de relatórios financeiros.
- **Gestão de Obras:** Foca na gestão eficaz de projetos de construção ou obras. Isso envolve o planejamento detalhado de cada fase do projeto, a alocação eficiente de recursos, o acompanhamento do progresso, o controle de qualidade e a gestão de riscos e custos.
- **Gestão de Recursos Humanos:** Envolve todas as atividades relacionadas à gestão de colaboradores da empresa. Isso inclui recrutamento, seleção, treinamento, avaliação de desempenho, administração de folha de pagamento, gestão de benefícios e desenvolvimento de políticas de RH.
- **Gestão de Subempreitadas:** Refere-se à gestão de contratos e relacionamentos com subempreiteiros ou fornecedores terceirizados. Isso inclui a definição de requisitos, a negociação de contratos, o monitoramento do desempenho, o controle de qualidade e o processamento de pagamentos.
- **Gestão de Outros Custos:** Além dos custos diretos relacionados a materiais e mão de obra, essa funcionalidade abrange outros tipos de despesas associadas a projetos ou operações. Isso pode incluir despesas administrativas, logísticas, legais, de marketing ou quaisquer outros custos operacionais.
- **Gestão da Margem de Lucro:** Permite o cálculo, monitoramento e otimização da margem de lucro da empresa. Isso envolve a análise detalhada dos custos e receitas, a identificação de oportunidades de redução de custos ou aumento de receitas e o estabelecimento de metas e estratégias para melhorar a lucratividade.

- **Gestão do Dossier:** Refere-se à gestão de toda a documentação e informações relacionadas a projetos, clientes, fornecedores ou colaboradores. Isso inclui a organização, armazenamento seguro, acesso controlado e recuperação eficiente de documentos e dados relevantes.
- **Gestão de Alvarás:** Envolve o acompanhamento e a renovação de todas as licenças e autorizações necessárias para operar legalmente. Isso pode incluir licenças de construção, licenças ambientais, licenças de operação e quaisquer outras autorizações regulatórias exigidas para as atividades da empresa.
- **Importar Arquivos CSV:** Permite a importação de dados de arquivos no formato CSV (Comma-Separated Values) para o sistema. Isso facilita a integração de informações de outras fontes, como planilhas eletrônicas, bancos de dados ou sistemas de terceiros, para uso dentro do sistema de gestão.
- **Informação dos Estaleiros:** Fornece informações detalhadas sobre os estaleiros ou locais de operação da empresa. Isso pode incluir dados sobre infraestrutura, equipamentos disponíveis, capacidade de produção, requisitos de segurança e quaisquer outras informações relevantes para a gestão eficaz das operações.
- **Informações Analíticas:** Oferece ferramentas e relatórios para análise detalhada de dados e desempenho. Isso inclui análises financeiras, análises de custos, análises de vendas, análises de produtividade e quaisquer outras análises relevantes para apoiar a tomada de decisões estratégicas.
- **Orçamentação:** Permite a elaboração, gestão e controle de orçamentos para projetos ou operações. Isso envolve a estimativa de custos, a alocação de recursos, o acompanhamento de despesas reais em comparação com o orçamento planejado e a identificação e gestão de variações orçamentárias.
- **Relatórios:** Gera relatórios detalhados sobre várias áreas e aspectos do negócio. Isso inclui relatórios financeiros, relatórios de desempenho operacional e quaisquer outros relatórios necessários para monitorar e avaliar o desempenho da empresa.

Capítulo 3

Análise de Requisitos e Interface

A análise de requisitos e a criação de *mockups* são etapas importantes no desenvolvimento de sistemas, especialmente no contexto de engenharia de software e design de sistemas de informação. Ambas as práticas trabalham em conjunto para garantir que o produto final atenda às necessidades dos usuários e funcione de maneira eficiente.

A análise de requisitos é o processo de estudar e entender as necessidades dos *stakeholders* para projetar um sistema que atenda a essas demandas. Essa etapa proporciona uma compreensão profunda dos requisitos e estabelece uma base estruturada para o desenvolvimento do sistema.

A utilização dos *mockups* é cada vez mais frequente no desenvolvimento de software, pois é uma maneira clara de visualizar como o sistema ficará após sua conclusão. No contexto da engenharia de software e do design de sistemas de informação, os *mockups* desempenha o papel de demonstrar aos *stakeholders* como o sistema será. Isso contribui para uma melhor compreensão dos requisitos e facilita a comunicação entre equipe e clientes durante todo o processo de desenvolvimento.

A utilização de *mockups* combinado com a análise de requisitos torna o desenvolvimento de projetos mais assertivo. A análise de requisitos fornece uma compreensão dos requisitos do projeto e o *mockup* permite visualizar e refinar esses requisitos em um ambiente flexível e rápido. Juntas, essas práticas melhoram a qualidade do desenvolvimento

do sistema final, reduzindo riscos, aprimorando a comunicação e garantindo que o produto atenda às expectativas dos *stakeholders*. A Seção 3.1 descreve a análise do projeto, incluindo as histórias de usuários desenvolvidas durante a coleta de requisitos. Na Seção 3.2, é apresentado os *mockups* do sistema, abordando os recursos de cores, componentes e páginas, utilizados para obter *feedback* antecipado e auxiliar no desenvolvimento.

3.1 Análise de projeto

No cenário atual das empresas de construção civil, a gestão eficiente de obras e processos administrativos é importante para garantir a competitividade e a sustentabilidade do negócio. A empresa ALN - Abel Luís Nogueiro e Irmãos Lda é uma PME situada em Bragança, na região de Trás-os-Montes, Portugal. Fundada em 1974 e detentora do alvará n.º 569, a ALN se especializou na execução, reabilitação e remodelação de obras públicas, além de construções particulares.

Em 2013, uma parceria entre a empresa ALN e o IPB levou ao desenvolvimento de uma plataforma web para melhorar a eficiência da gestão e possibilitar a desmaterialização dos processos. Esta ferramenta foi projetada para atender às necessidades específicas da empresa, substituindo o uso de planilhas eletrônicas e tornando-se essencial para a eficiência e organização dos seus processos.

Com o tempo, essa plataforma tornou-se obsoleta devido à falta de compatibilidade com as versões mais recentes dos navegadores web. Como resultado, a manutenção da ferramenta tornou-se ineficaz devido à deterioração da tecnologia utilizada no projeto. Surgiu, portanto, a necessidade de implementar uma nova plataforma, com recursos atualizados e uma tecnologia moderna que proporcionasse uma experiência de usuário aprimorada, incluindo UX/UI para melhor usabilidade e experiência para os usuários.

Durante o processo de análise, foi disponibilizado um relatório técnico detalhado do projeto da plataforma de gestão da empresa Abel Luís Nogueiro & Irmãos Lda (ALN). Esse documento foi fundamental para identificar e compreender a maioria dos requisitos necessários para o desenvolvimento de uma nova ferramenta.

Ao examinar cuidadosamente o relatório, observou-se uma série de requisitos relacionados à gestão. Uma pesquisa detalhada realizada no capítulo de Estado da Arte 2 revelou que os sistemas ERP tinham um custo significativo tanto em sua aquisição quanto em sua manutenção. Na locação, era comum que esses sistemas não possuísem todos os recursos necessários e as opções de manutenção fossem limitadas. Embora a terceirização do desenvolvimento pudesse parecer uma alternativa viável, os custos associados à manutenção do sistema podiam se tornar elevados. Diante desses desafios, concluiu-se que desenvolver uma nova aplicação era a melhor opção para a empresa. Com base nessa premissa, deu-se início ao processo de elaboração das histórias de usuário.

3.1.1 História de usuário (*User Story*)

O levantamento de requisitos é uma etapa fundamental no planejamento para compreender as necessidades e desejos dos *stakeholders*. Um dos recursos que pode ser utilizado para coleta de requisitos é a história de usuário, que por meio de uma breve história são relatados os anseios dos usuários.

Nessa etapa foram elaboradas histórias de usuário, permitindo uma análise mais aprofundada dos requisitos funcionais e não funcionais do sistema. No total, foram escritas 135 *user stories* abrangendo uma variedade de recursos, tais como: autenticação de usuário, gestão de materiais, controle de consumíveis e serviços, gestão de obras, registro de presença de colaboradores, controle de faturas, emissão de pedidos de cotação, execução orçamentária, registro de *leasing* de equipamentos e veículos, seguros de trabalho e automóvel, e outros recursos detalhados no Apêndice A.

As histórias de usuário foram priorizadas de acordo com as necessidades da empresa. A tabela 3.1 apresenta algumas das principais histórias. Cada história contém o peso e a descrição da história de usuário, sendo o peso um valor numérico inteiro que varia de um a cinco, sendo cinco o valor de maior prioridade para a empresa.

A plataforma deve permitir o acesso a dois perfis: Administrador e Operacional. O perfil Administrador possui acesso irrestrito a todas as funcionalidades da plataforma. Já

o perfil Operacional tem acesso limitado, incluindo gestão de equipamentos e serviços, clientes, fornecedores, gestão de materiais, entidades, faturas, pedidos de cotação, notas de encomenda, gestão de outros custos e gerenciamento das amortizações fiscais. No entanto, o perfil Operacional não tem permissão para acessar informações relacionadas a salários, presenças e execução orçamental.

Tabela 3.1: Exemplo de histórias de usuários.

Peso	História de Usuário
5	Como Administrador, eu quero efetuar o login.
5	Como Operacional, eu quero efetuar o login.
4	Como Operacional, eu quero inserir novos materiais.
5	Como Operacional, eu quero listar os clientes registrados.
5	Como Operacional, eu quero pesquisar/filtrar os fornecedores registrados.
5	Como Operacional, eu quero enviar <i>email</i> com os pedidos de cotação para os fornecedores.
5	Como Administrador, eu quero emitir a execução orçamental de cada obra.
5	Como Administrador, eu quero inserir um nova obra.
5	Como Administrador, eu quero listar as presenças registrados.
5	Como Operacional, eu quero inserir um nova fatura de materiais.
3	Como Operacional, eu quero inserir um novo seguro de automóvel.
3	Como Operacional, eu quero inserir um novo imposto único de circulação.
3	Como Operacional, eu quero inserir um <i>leasing</i> de viatura.
3	Como Operacional, eu quero inserir um novo seguro de trabalho.
1	Como Administrador, eu quero listar os salários dos colaboradores.
1	Como Administrador, eu quero listar os parâmetros.

3.2 Interface e Experiência do Usuário (UI/UX)

No mundo digital contemporâneo, dois conceitos fundamentais moldam a forma como os usuários interagem com dispositivos e software: UI e UX. UI, ou interface do usuário, é o ponto de interação entre o usuário e um dispositivo ou software, abrangendo elementos visuais como telas, teclados, mouses e ícones, bem como o layout e design geral da interface. O objetivo do UI é criar uma aparência atraente e intuitiva, facilitando a navegação e a interação do usuário com o sistema [15]–[18].

Por outro lado, UX, ou experiência do usuário, refere-se à maneira como uma pessoa se sente ao interagir com um produto ou sistema, como um site ou aplicativo. O UX envolve aspectos como usabilidade, acessibilidade e eficiência da interação, além da satisfação geral do usuário. O foco do UX é assegurar que o produto atenda às necessidades dos usuários de maneira eficaz e agradável, proporcionando uma experiência positiva e envolvente. Juntos, UI e UX desempenham papéis importantes na criação de produtos digitais que não apenas funcionam bem, mas também encantam e satisfazem os usuários [15]–[18].

Com o objetivo de promover uma experiência agradável para o usuário, os *mockups* foram desenvolvidos como representações visuais estáticas da interface da plataforma. Amplamente utilizados no desenvolvimento de software e design de sistemas de informação, os *mockups* oferecem uma visualização detalhada do layout, incluindo elementos como botões, menus e formulários, permitindo que os *stakeholders* avaliem a interface antes da implementação. Desta forma, os *mockups* ajudam na obtenção de *feedback* antecipado e na validação dos requisitos, reduzindo riscos e economizando tempo e recursos. Além de contribuir para a identificação de possíveis melhorias e garantindo uma experiência de usuário mais eficaz [18], [19].

Durante a construção dos *mockups*, foi utilizada a ferramenta Figma [20]. Com ela, foi possível desenvolver os componentes seguindo um padrão consistente, criando uma identidade visual coesa para o sistema. Além disso, o Figma permite adicionar ações entre os objetos projetados, possibilitando interações através de cliques em regiões específicas do *mockup*. Isso simula funcionalidades como a de um botão, aprimorando a experiência do usuário e contribuindo para resultados mais realistas e eficazes. A modelagem foi organizada em três etapas: definição das cores utilizadas, desenvolvimento dos componentes e por fim, as construções das páginas do *mockup*.

3.2.1 Definição de cores

A escolha das cores são importante para manter a consistência da identidade visual, como cores dos botões, bordas, fontes e entre outros. Trabalhar com um segmento limitado de

cores é importante para manter a interface limpa e organizada. Já o uso adequado de contrastes garante que o texto e outros elementos sejam facilmente legíveis [21]. Na Figura 3.1 apresenta a paleta de cores escolhidas para o projeto.

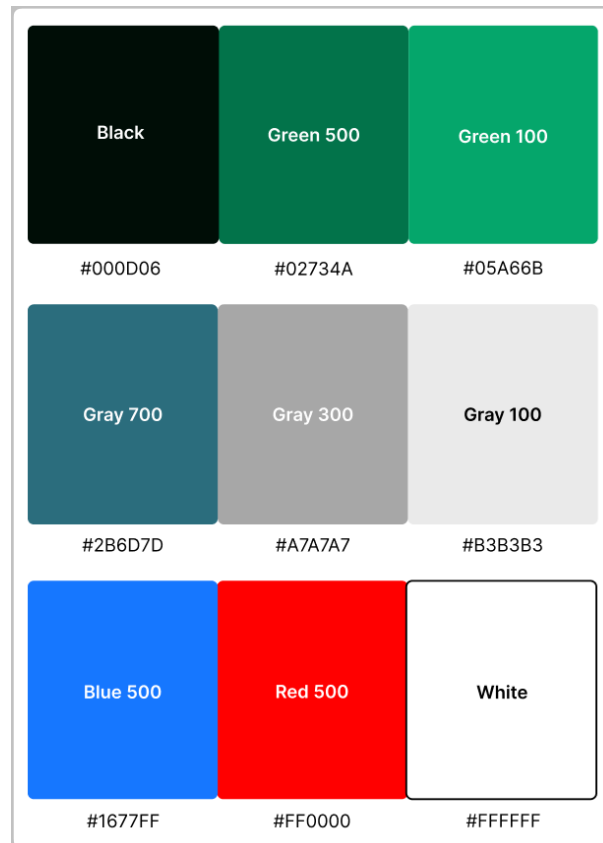


Figura 3.1: Paleta de cores da interface

3.2.2 Componentes

Componentes são elementos reutilizáveis no Figma. Eles podem ser botões, ícones, ou qualquer outro elemento que deseja usar repetidamente no design, podendo ser até um conjunto de outros componentes. Uma vez criado, qualquer alteração no componente principal (Master) é refletida em todas as suas instâncias.

Nesta etapa foram construídos os componentes da interface. A Figura 3.2 apresenta os componentes de gráfico e de listagem, já na Figura 3.3 mostra a listagem de presença e componentes composto por outros componentes, como no caso dos gráficos. O mesmo

acontece nas figuras 3.4 e 3.5, onde na Figura 3.4 apresenta os formulários com vários componentes de *inputs* e na Figura 3.5 apresenta o conjunto de componentes de *inputs* e também componentes de listagem.

O uso de componentes no Figma é uma prática poderosa que ajuda a manter a consistência, eficiência e qualidade no design. Ao criar e usar componentes de maneira eficaz, é possível criar interfaces de usuário bem organizadas e facilmente mantidas.

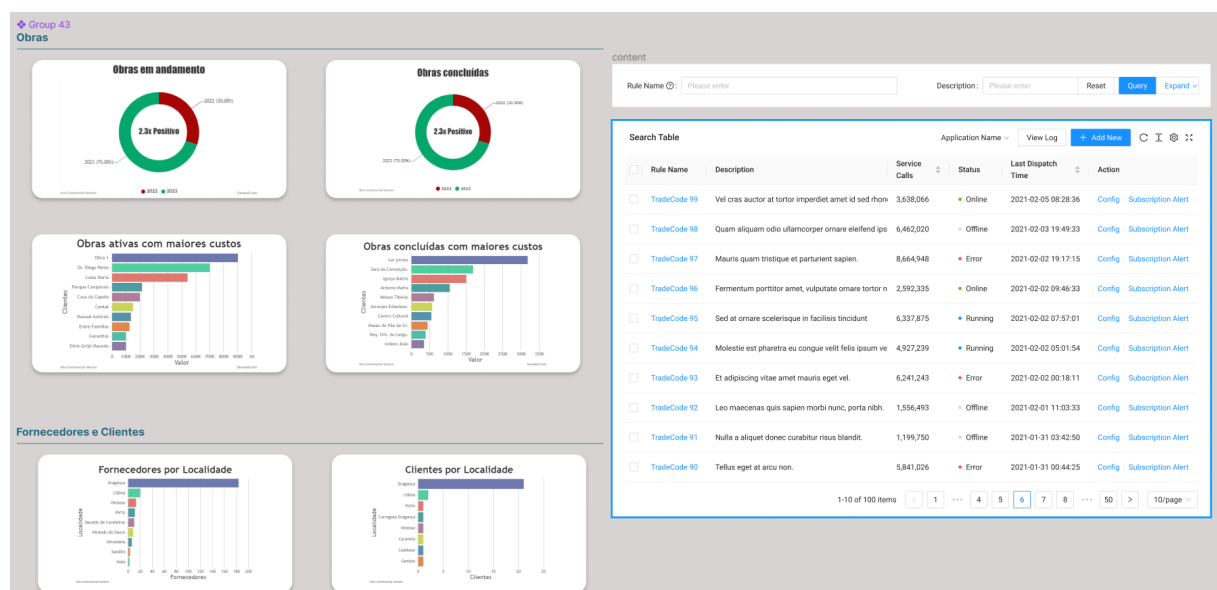


Figura 3.2: Mockup dos componentes de gráficos e listagem e dados.

A Figura 3.2 ilustra os modelos de componentes gráficos que serão úteis para representar os custos e o tempo de execução de cada obra. Além dos gráficos, também é apresentado o modelo de listagem, que será utilizado em quase todas as páginas da plataforma. Esta listagem inclui filtro de busca, paginação e ações como adicionar, editar, remover e *download* de faturas, quando aplicável.

Na Figura 3.3 apresenta os modelos de componentes relacionados a estatísticas e relatórios, utilizando os gráficos mostrados na Figura 3.2. Além disso, é possível observar os modelos de registro de presença nas obras, organizados conforme o mês selecionado, permitindo o levantamento dos custos com mão-de-obra para cada projeto de construção.

A Figura 3.4 apresenta os modelos de componentes essenciais, como campos de texto,

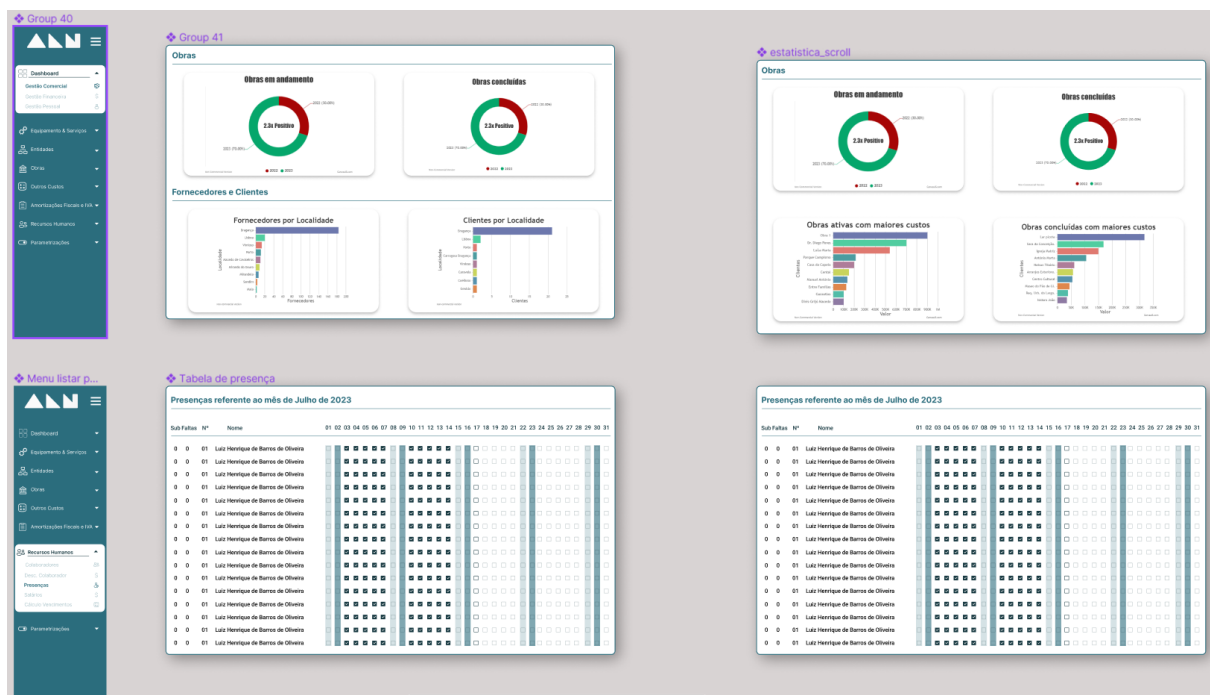


Figura 3.3: Componentes de conjunto de gráficos e listagem de presenças

Figure 3.4 displays a form for data input, an alert, and a listing. The form is divided into several sections:

- Ordenar:** A dropdown menu for sorting options.
- Atualização de Material:** A section for updating material information, including fields for 'Material', 'Referencia', 'Unidade', 'Modelo', 'Marca', 'Matricula', 'Tipo', 'Fornecedor', 'Data Registro', 'Valor Entrada', 'Valor Residual', and 'Nº de Rendas'.
- Adicionar Viatura:** A section for adding a vehicle, including fields for 'Material', 'Referencia', 'Unidade', 'Modelo', 'Marca', 'Matricula', 'Tipo', 'Fornecedor', 'Data Registro', 'Valor Entrada', 'Valor Residual', and 'Nº de Rendas'.
- Alerta:** A modal alert box titled 'Aviso!' with the message 'Tem certeza que deseja remover este item? Esta ação é irreversível!' and buttons for 'Cancelar' and 'Confirmar'.
- Material:** A table listing materials with columns for 'Name', 'Referencia', 'Unidade', and 'Actions'. The table contains 6 rows of data.

Figura 3.4: Componentes de *input* de dados, alerta e listagem.

botões, textos, fontes, cores, ícones e componentes de seleção. Esse conjunto de componentes forma a base para a construção dos demais modelos de interface e outros elementos da plataforma.

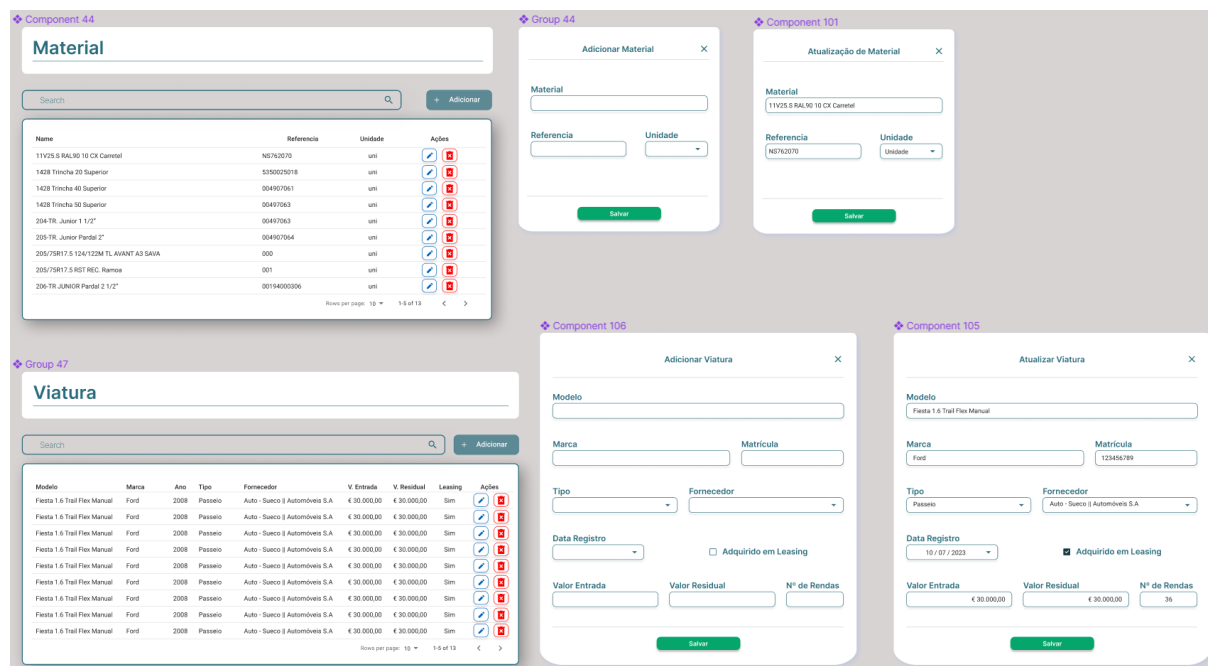


Figura 3.5: Componentes de listagem e formulário de cadastro e edição.

A Figura 3.5 apresenta modelos mais completos, incluindo um componente de listagem mais elaborado, além dos componentes para adição e edição de informações de materiais e viaturas. Nesta etapa, é possível observar que os componentes estão mais consolidados e prontos para serem utilizados na projeção das páginas.

3.2.3 Páginas

O *mockup* das páginas de um sistema é uma etapa essencial no processo de *design* e desenvolvimento de produtos digitais. Essa prática não só melhora a visualização do produto final, como também facilita a comunicação entre as equipes envolvidas e contribui para um planejamento mais eficaz.

O uso desses *mockups* permitem que todos os *stakeholders*, incluindo clientes, desenvolvedores e designers, visualizem o *design* final do sistema antes de sua implementação.

Isso ajuda a planejar e ajustar o *layout*, a funcionalidade e a aparência geral do sistema, garantindo que todos os aspectos visuais estejam em conformidade com os objetivos do projeto.

Ao apresentar os *mockups* aos *stakeholders*, é possível coletar *feedback* valioso antes de iniciar a fase de desenvolvimento. Esse *feedback* antecipado possibilita a identificação e correção de problemas potenciais desde o início, evitando retrabalho caro e demorado e assegurando que o produto final atenda às expectativas dos usuários.

Nessa etapa foram desenvolvidos dois modelos de *mockups*, a primeira versão com menu lateral no estilo *dark* e o segundo modelo no estilo *light*. A intenção foi oferecer dois modelos que seguisse o mesmo padrão das páginas, e que oferecesse um estilo diferente ao menu da aplicação. Nas figuras 3.6 (estilo *dark*) e 3.7 (estilo *light*) é possível observar essa diferença. Os dois modelos foram apresentados ao gerente da ALN que deixou claro sua preferência pela versão *light*.

Modelo	Marca	Ano	Tipo	Fornecedor	V. Entrada	V. Residual	Leasing	Ações
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 

Figura 3.8: Mockup da página de listagem de viaturas.

Ao observar os requisitos do sistema, notou-se uma sequência de recursos semelhantes como: listar, adicionar, editar e deletar itens. Seguindo a lógica desses recursos foram

Adicionar Viatura

Modelo

Marca Matrícula

Tipo Fornecedor

Data Registro ☐ Adquirido em Leasing

Valor Entrada Valor Residual Nº de Rendas

Salvar

Figura 3.9: Mockup da página de adicionar uma viatura.

Atualizar Viatura

Modelo

Marca Matrícula

Tipo Fornecedor

Data Registro ☒ Adquirido em Leasing

Valor Entrada Valor Residual Nº de Rendas

Salvar

Figura 3.10: Mockup da página de editar uma viatura.

desenvolvidos os *mockups* para servir como um padrão. Nas figuras 3.8, 3.9, 3.10 e 3.11 apresentam os padrões de componentes a serem seguidos nos demais *mockups*. Todos os *mockups* das páginas foram adicionados no Apêndice B.

A Figura 3.8 mostra o mockup da página destinada à viatura. Esta página adota um formato padronizado semelhante ao das páginas de materiais, consumíveis/serviços e pedidos de cotação. O layout inclui uma tabela com informações sobre o recurso, uma barra de pesquisa, um link para o formulário de registro de novos recursos e opções para editar e remover cada item listado na tabela.

A Figura 3.9 apresenta o formulário utilizado para adicionar uma viatura. Ele contém campos para inserir o modelo, a marca, a matrícula, o tipo, o fornecedor, a data de registro e os valores relacionados à aquisição da viatura.

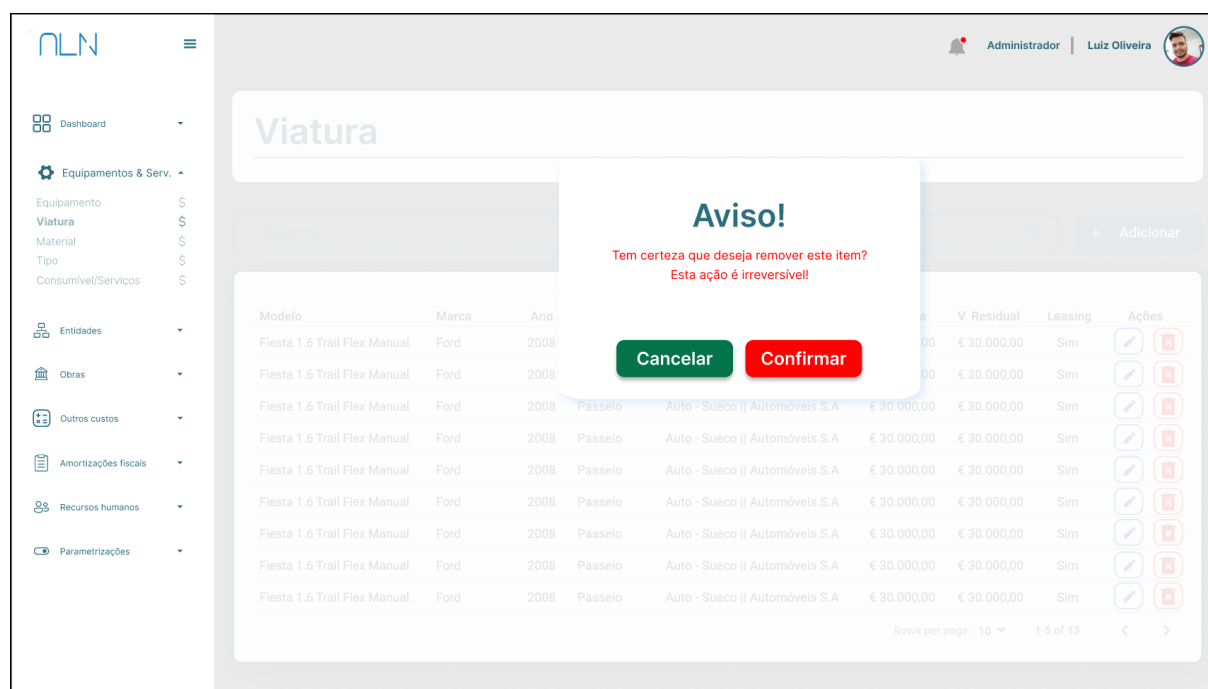


Figura 3.11: Mockup da página de deletar uma viatura.

Assim como o formulário para registrar novas viaturas, é possível atualizar as informações de uma viatura, conforme ilustrado na Figura 3.10. Esse recurso é necessário para corrigir erros feitos durante o cadastro. Também é possível remover uma viatura que foi registrada na plataforma. O *mockup* dessa funcionalidade é apresentado na Figura 3.11.

Capítulo 4

Desenvolvimento

O presente capítulo descreve a reconstrução do sistema de gestão de obras da empresa ALN, focando na modernização da plataforma e na implementação de novas funcionalidades. O sistema original, responsável por gerenciar tanto as obras em andamento quanto as atividades internas da empresa, apresentava problemas de compatibilidade com os navegadores atuais devido à depreciação das tecnologias utilizadas, comprometendo a usabilidade e a eficiência do sistema.

A nova versão do sistema, estruturado em back-end e front-end, foi desenvolvida com metodologias ágeis e o framework Scrum, visando solucionar os problemas de compatibilidade e aprimorar a experiência do usuário. Para tal, foram implementadas novas funcionalidades, como o envio automático de emails, a utilização de APIs para validação de dados e a criação de interfaces mais intuitivas. Essas funcionalidades visavam otimizar o fluxo de trabalho, aumentar a confiabilidade dos dados e facilitar a visualização de informações relevantes.

O desenvolvimento do sistema seguiu metodologias ágeis, utilizando o Scrum como framework. As histórias de usuários foram transformadas em tarefas no Kanban do GitHub. Essa abordagem permitiu o acompanhamento do progresso do desenvolvimento do projeto em tempo real e a realização de ajustes conforme necessário.

O backend da aplicação foi desenvolvido para controlar o acesso dos usuários, gerenciar recursos de fatura, usuário e material, gerar relatórios e enviar emails. Desta forma, o

backend é responsável por garantir o acesso às informações e funcionalidades relevantes aos usuários autorizados. Já o frontend da aplicação apresenta os recursos de fatura, amortização fiscal, presenças nas obras e visualização de relatórios com gráficos.

A integração e implantação do sistema no servidor da empresa foram realizadas utilizando Docker e Docker Compose, otimizando o processo e garantindo a portabilidade da aplicação. Durante a integração, os esquemas de dados de algumas tabelas do banco de dados foram atualizados para garantir a compatibilidade com o novo sistema.

A Tabela 4.1 apresenta as principais tecnologias utilizadas no desenvolvimento do frontend e do backend da aplicação. No frontend, foram empregadas as linguagens HTML, CSS e JavaScript, utilizando o TypeScript como extensão e os frameworks ViteJS e ReactJS. Para a estilização dos componentes, a biblioteca do Ant Design foi utilizada. Já no backend foi feito o uso da linguagem C# e do framework .NET. Para o mapeamento objeto-relacional, optou-se pelo Entity Framework. Além disso, foi feita a integração com a API externa CTT.

Tabela 4.1: Principais tecnologias do frontend e do backend.

Frontend	Backend
HTML e CSS	C#
TypeScript	.NET
ViteJS	Entity Framework
React JS	API CTT
Ant Design	-

O desenvolvimento do projeto se baseou em um conjunto de ferramentas destacadas na Tabela 4.2, categorizadas em três grupos principais: bancos de dados, arquitetura e ferramentas diversas.

No âmbito dos bancos de dados, o MySQL foi o sistema de gerenciamento de dados escolhido, enquanto o DBeaver foi utilizado como ferramenta de interface gráfica para interagir com o banco de dados. A linguagem SQL foi fundamental para realizar consultas e manipulações nos dados.

Em relação à arquitetura do projeto, a arquitetura REST foi adotada para definir

a estrutura da API e a comunicação entre os componentes da aplicação. O Docker foi utilizado para containerizar a aplicação, facilitando a implantação e o gerenciamento em diferentes ambientes.

Por fim, o grupo de ferramentas diversas engloba o GitHub, utilizado para controle de versão do código, o Swagger para documentação da API e o Visual Studio Code como editor de código-fonte.

Tabela 4.2: Principais ferramentas utilizadas no projeto.

Bancos de Dados	Arquitetura	Outras
MySQL	REST	Docker
DBever	MVC	GitHub
SQL	-	Swagger
-	-	Visual Studio Code

4.1 Arquitetura do sistema

A arquitetura de sistema refere-se à estrutura organizacional fundamental de um sistema, definida por seus componentes, suas relações e como interagem entre si para atingir objetivos específicos. Em termos simples, é o design e a estrutura de um sistema, que pode ser software, hardware ou uma combinação de ambos [22].

Este projeto segue a arquitetura cliente-servidor monolítica orientada a serviços, separando o frontend e o backend. O cliente que solicita serviços ou recursos ao servidor e exibe os dados ou resultados recebidos. Já o servidor é o componente que fornece serviços ou recursos solicitados pelos clientes. Ele processa as requisições, executa as regras de negócios e acessa os dados necessários no banco de dados. Toda comunicação é feita pelo protocolo HyperText Transfer Protocol (HTTP). A Figura 4.1 apresenta o esquema da arquitetura do sistema.

Frontend (Camada de Apresentação): Desenvolvido com tecnologias como HTML, CSS, JavaScript, frameworks como React ou Angular. Backend (Camada de Aplicação): Servidores de aplicação usando Node.js, Django, Spring Boot, etc. Banco de Dados

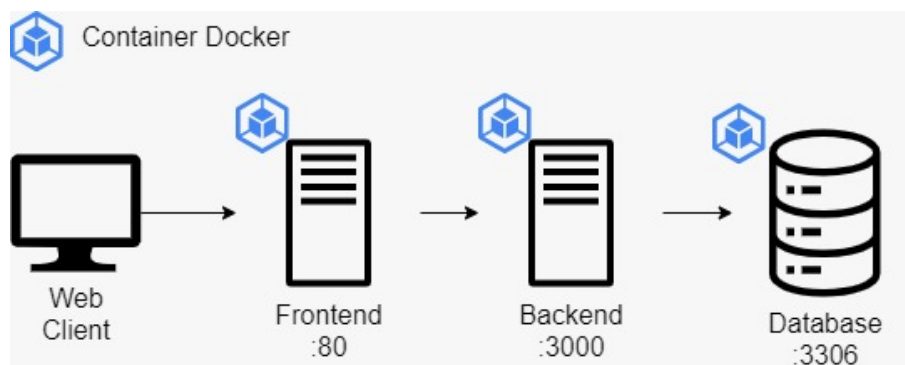


Figura 4.1: Arquitetura da aplicação

(Camada de Dados): Sistemas de gerenciamento de banco de dados como MySQL, PostgreSQL, MongoDB

Todos os serviços (frontend, backend e banco de dados) devem executar em um servidor local utilizando Microsoft Windows 11. Para ajudar na performance, foram realizadas algumas pesquisas para entender quais linguagens de programação utilizar no backend. Com as pesquisas realizadas [23]–[26], foi possível encontrar testes de benchmark para diferentes linguagens de programação em ambientes Windows. Com base nas pesquisas foi possível escolher a linguagem com melhor performance para o desenvolvimento do backend, que foi a linguagem de programação C# e também o uso do Entity Framework Core para desenvolvimento da API.

No frontend, foi definido o uso do React, por ser uma tecnologia recente e que está sendo muito utilizada em projetos web. Para acompanhar esse framework, foi utilizado a linguagem de programação Typescript, uma linguagem de fácil aprendizado e tipada, que ajuda na organização do código.

Além dessas tecnologias, a primeira versão do frontend foi criada usando a nova arquitetura *server components* do React, usando o framework NextJs, para reduzir a quantidade de JavaScript que será enviado para o browser (navegador). A estilização do frontend gerou alguns gargalos no decorrer do projeto, na primeira versão, foi utilizado o Material UI, portando é necessário um bom conhecimento de CSS para criar novos componentes estilizados. Outras frameworks foram testadas para comparar o ganho de performance e

também no ganho de tempo de desenvolvimento. A versão atual possibilitou a remoção do NextJs do projeto devido a sua complexidade no desenvolvimento e também a substituição do framework de css Material UI pelo Ant Design, que possibilitou uma gama maior de componentes pré definidos que acelera o desenvolvimento. Também foi adicionado o ViteJs que acelerou o tempo de build da aplicação.

4.2 Banco de dados

A importância dos bancos de dados reside na sua capacidade de organizar, proteger e gerenciar grandes volumes de dados de maneira eficiente e segura. A empresa ALN já utilizava o banco de dados MySQL para suas operações diárias. Dessa forma, foi decidido manter o mesmo Sistema de Gerenciamento de Banco de Dados (SGDB) devido a disponibilidade de uma cópia com dados mascarados como versão de desenvolvimento e facilitar o processo de migração de dados. Esta seção apresenta todas as 58 tabelas do banco de dados e seus relacionamentos.

4.2.1 Utilizador, Log, Parâmetros e Feriado

A demanda pelo controle de acesso ao sistema gerou a necessidade do armazenamento dos dados do utilizador. Além disso, o registro de parâmetros e feriados foram úteis para a configuração do sistema. A Figura 4.2 apresenta as tabelas referente ao controle de acesso, parâmetros de configuração e os feriados fixos e feriados móveis, que tem alteração de data em relação ao ano.

A tabela "utilizador" armazena os dados necessários para o controle de acesso dos utilizadores na plataforma, incluindo identificador, nome, credenciais de login, email, associações a colaboradores e tipos de utilizadores. As chaves estrangeiras garantem que as associações de tipos de utilizadores sejam válidas.

A tabela "tipo_utilizador" registra os diferentes tipos de utilizadores na plataforma. Esta tabela é essencial para definir e gerenciar os níveis de acesso dos utilizadores, assegurando que as permissões e os privilégios sejam adequados ao papel de cada utilizador.

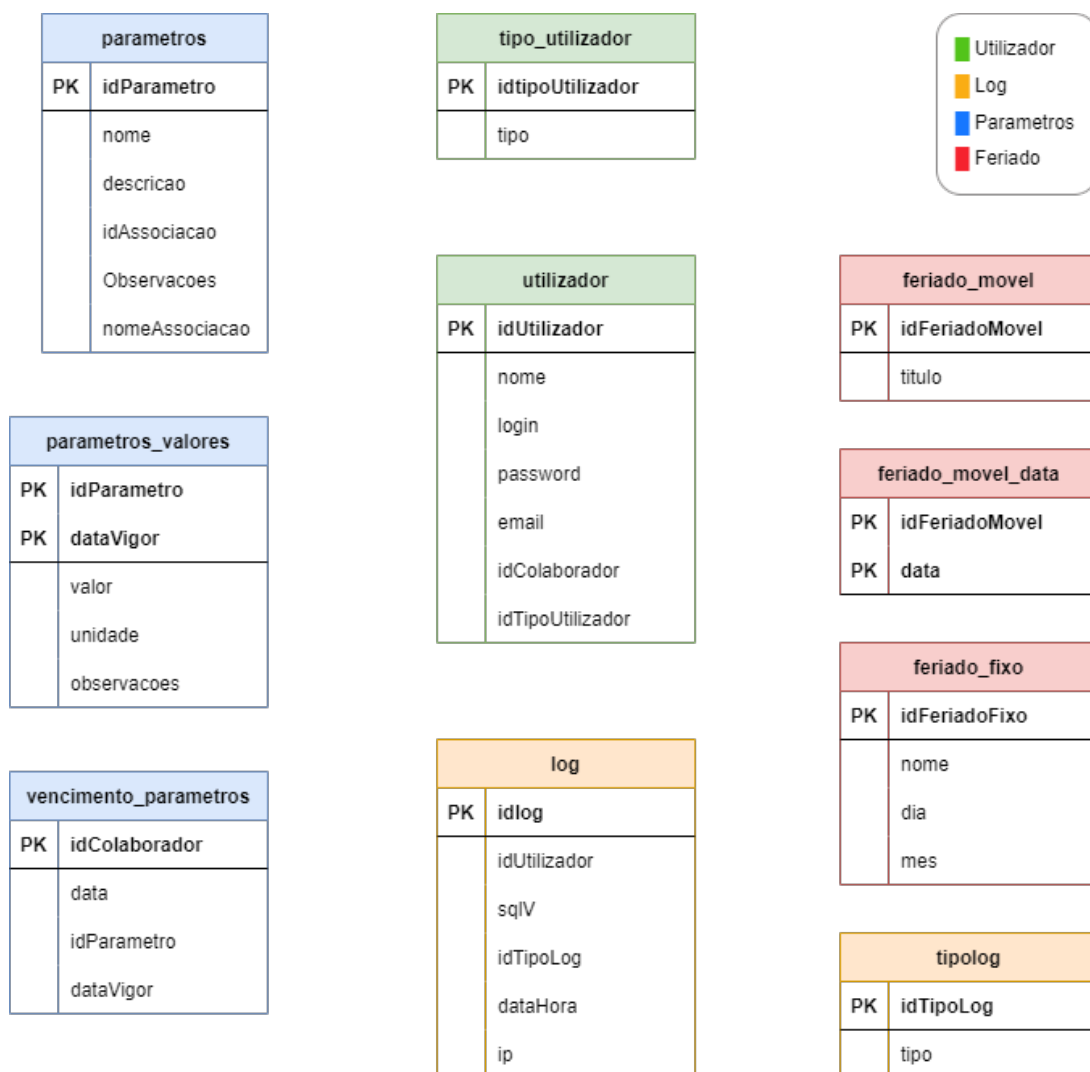


Figura 4.2: Tabelas do banco de dados referente aos recursos de utilizador, log, parâmetros e feriados.

A tabela "log" armazena registros detalhados de todas as operações realizadas na plataforma. Esta tabela é importante para auditoria e monitoramento de atividades dos utilizadores, permitindo rastrear ações específicas e garantir a segurança e integridade do sistema.

A tabela "parametros" armazena informações sobre parâmetros utilizados no sistema, proporcionando flexibilidade na configuração e personalização de diversos aspectos da aplicação. Esta tabela inclui campos para descrever o nome, a descrição, e possíveis associações a outros elementos do sistema, juntamente com observações adicionais.

Os parâmetros possuem outros valores que são armazenados na tabela "parametros_valores" que inclui identificador, data de vigor, valor, unidade e observações. Cada parâmetro possui uma validade, essas informações são adicionadas na tabela "vencimento_parametros", que compõe os seguintes atributos: identificador do colaborador, data, identificador do parâmetro e data de vigor.

Ainda na Figura 4.2 é possível observar as tabelas referente aos feriados móveis e feriados fixos. Os feriados móveis são os feriados que não possuem uma a mesma data todos os anos ou não acontece todos os anos. Já os feriados fixos são feriados que acontecem todos os anos na mesma data. Nas tabelas "feriado_movel", "feriado_movel_data" e "feriado_fixo" são registrados todos os feriados.

A tabela "feriado_movel" registra informações sobre feriados que não ocorrem em datas fixas, mas variam a cada ano. Com esta tabela, feriados móveis podem ser gerenciados de maneira eficiente, garantindo que todas as partes interessadas estejam cientes das datas importantes ao longo do ano, mesmo quando essas datas variam.

Na tabela "feriado_movel_data" são registradas as datas específicas em que os feriados móveis ocorrem, ligando cada feriado móvel às suas respectivas datas anuais. Isso permite a rastreabilidade e controle dos feriados móveis.

Por fim, a tabela "feriado_fixo" armazena informações sobre feriados que ocorrem em datas fixas todos os anos. Esta estrutura é útil para a gestão de calendário, planejamento de atividades, e cálculos de dias úteis. Incluindo identificador, nome, dia e mês.

4.2.2 Colaborador

As informações dos colaboradores permitem desde a identificação dos profissionais até uma gestão mais eficiente dos custos operacionais da empresa. Dados como informações pessoais, contratos, férias, profissão, habilidades acadêmicas, salário e dependentes ajudam a compreender as particularidades de cada colaborador, possibilitando a criação de políticas internas para melhorar a qualidade do trabalho. A Figura 4.3 apresenta as tabelas que armazenam as informações relacionadas aos colaboradores.

A tabela "colaborador" armazena informações detalhadas sobre os colaboradores da empresa, abrangendo dados pessoais, detalhes de contato e informações empregatícias. Este modelo tem como objetivo proporcionar uma visão abrangente e clara dos funcionários da organização. Foram incorporados campos como identificador do colaborador, nome, número de identificação pessoal (BI), número de identificação fiscal (NIF), número de segurança social, data de início de atividade, status de ativação, endereço, identificador de cargo, identificador de profissão, código postal, cidade, data de nascimento, número de funcionário, tamanho de calçado, subsídio de alimentação, identificador de habilidades acadêmicas, estado civil, nome do cônjuge, NIF do cônjuge, status de emprego do cônjuge, profissão do cônjuge, status de seguro, status de segurança social, status de folha de pagamento, identificador de contrato, status de folha de pagamento exclusiva e subsídio duo décimos.

Alguns dados dos descendentes dos colaboradores são armazenados na tabela "descendentedecolaborador" que inclui o identificador do descendente, identificador do colaborador, nome e data de nascimento.

O estado civil do colaborador é registrado na tabela "estado_civil", que contém informações como identificador do estado civil, e a descrição do estado civil. As habilidades acadêmicas do colaborador são registradas na tabela "habilidades_academicas", com os atributos identificador da habilitação e a descrição da habilitação.

A tabela "ferias" registra informações sobre os períodos de férias dos colaboradores, incluindo o identificador de férias, identificador do colaborador, data de inicio e fim das

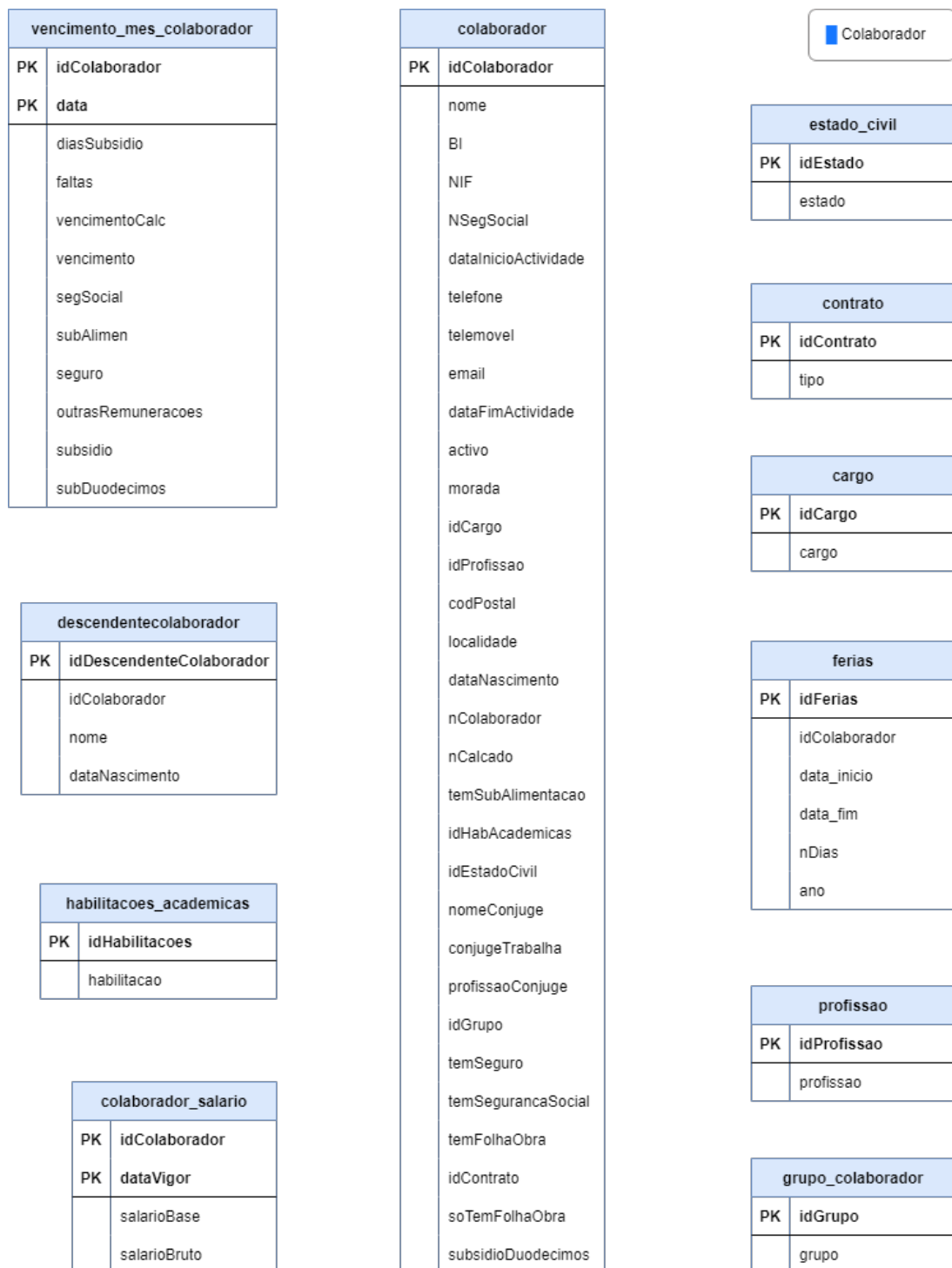


Figura 4.3: Tabelas do banco de dados referente aos recursos do colaborador.

férias, número de dias e o ano que as férias foram agendadas.

Na tabela "cargo" armazena os nomes das posições dos colaboradores na empresa, incluindo seu identificador e o nome do cargo.

A tabela "profissao" registra os nomes das atividades exercidas pelos colaboradores na empresa, incluindo seu identificador e o nome da profissão.

O tipo de contrato dos colaboradores são registrados na tabela "contrato", incluindo o identificador do contrato e o tipo do contrato oferecido pela empresa aos seus colaboradores.

O salário de cada colaborador é armazenado na tabela "colaborador_salario", incluindo o identificador do colaborador, data de vigor, salário base e salário bruto.

Para o cálculo dos vencimentos, são necessários outros dados que são adicionados na tabela "vencimento_mes_colaborador" como identificador do colaborador, data, dias de subsídio, faltas, vencimentos, seguro social, subsídio para alimentação, seguro, outras remunerações, valor do subsídio, subsídio duo décimos.

Cada colaborador pertence a um grupo de colaboradores. A tabela "grupo_colaborador" inclui o identificador do grupo e a descrição.

Por fim, na tabela "habilitacoes_academicas" armazena o identificador e o nomes das habilidades acadêmias.

4.2.3 Fornecedor, Encomenda, Consumível e Pedido de Cotação

O registro das informações dos fornecedores contribuem para a emissão dos pedidos de cotação, solicitação de encomendas, consumíveis e serviços. A tabela "fornecedor" armazena informações sobre os fornecedores com os quais a empresa mantém relações comerciais. A tabela de "nota_encomenda" registra os pedidos de produtos feitos pela empresa aos fornecedores. A tabela "consumivel_servicos" cataloga os materiais de uso regular pela empresa, facilitando o controle de estoque e reposição. Por fim, a tabela "pedido_cotação" registra as solicitações de preços enviadas aos fornecedores para avaliação e comparação antes da realização de uma encomenda formal. Essas tabelas formam a base estrutural

que permite a gestão e organização dos processos de compra e abastecimento da empresa. Na Figura 4.4 apresenta as tabelas referente a esse conjunto de informações.

A tabela "fornecedor" armazena todas as informações necessárias sobre os fornecedores, incluindo detalhes de contato e endereço. O identificador único idFornecedor garante que cada fornecedor possa ser diferenciado dos outros. Os campos obrigatórios, como nome, morada, número de identificação fiscal (NIF), e emailContacto, asseguram que informações essenciais sejam sempre fornecidas, enquanto os campos opcionais, como telefone e fax, permitem armazenar dados adicionais de contato quando disponíveis.

A tabela "pedido_cotacao" registra os detalhes dos pedidos de cotação, incluindo informações como a data do pedido, descrições, número de itens, descontos aplicados, e informações sobre arquivos associados ao pedido (como PDFs). A relação com a tabela "obra" permite associar cada pedido de cotação a uma obra específica, assegurando que qualquer modificação ou remoção de obras seja propagada para os pedidos de cotação relacionados.

A tabela "pedido_cotacao_fornecedor" registra as cotações feitas por fornecedores em resposta a pedidos de cotação. Cada registro na tabela liga um fornecedor específico (idFornecedor) a um pedido de cotação específico (idPedidoCotacao). As chaves estrangeiras garantem que as associações sejam válidas, ou seja, que o pedido de cotação e o fornecedor existam nas suas respectivas tabelas. Os índices criados nas colunas idPedidoCotacao e idFornecedor ajudam a otimizar as consultas que filtram ou ordenam por esses campos.

A tabela "pedido_cotacao_material" detalha os materiais incluídos em um pedido de cotação. Cada registro especifica um material, a quantidade solicitada, o valor, descontos aplicados, e referências adicionais. As relações com as tabelas "pedido_cotacao", material e unidade garantem que todos os itens estejam corretamente associados aos seus pedidos de cotação, materiais e unidades de medida. As chaves estrangeiras asseguram a integridade referencial e permitem operações em cascata para manter a consistência dos dados.

A tabela "nota_encomenda_itens" armazena os materiais incluídos em uma nota de encomenda. Cada registro especifica um material, a quantidade solicitada, o valor, o Imposto sobre Valor Acrescentado (IVA) aplicado e a data do preço. As relações com as

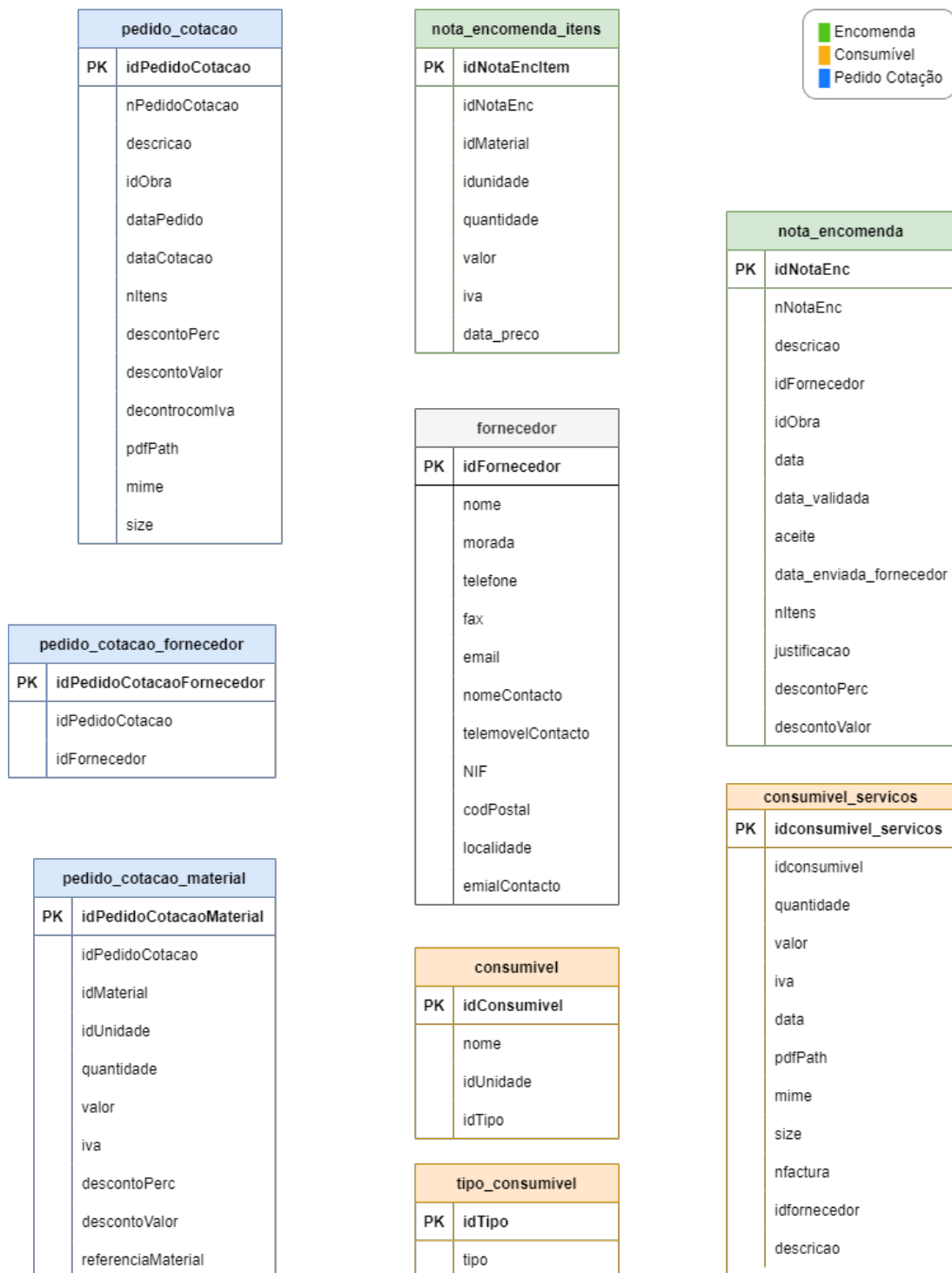


Figura 4.4: Tabelas do banco de dados referente aos recursos da tabela "fornecedor", como: encomenda, consumível e pedido de cotação.

tabelas "nota_encomenda", "material" e "unidade" garantem que todos os itens estejam corretamente associados às suas notas de encomenda, materiais e unidades de medida.

Na tabela "consumivel" são registrados os materiais que são consumidos. Cada registro contém informações sobre o nome do consumível, sua unidade de medida e seu tipo. As chaves estrangeiras idUnidade e idTipo garantem que as unidades de medida e tipos referenciados sejam válidos, enquanto os índices criados nessas colunas ajudam a otimizar as consultas relacionadas a esses dados.

A tabela "tipo_consumivel" armazena os diferentes tipos de consumíveis disponíveis. Cada registro contém o nome de um tipo de consumível, como por exemplo "Papel A4" e "Tinta para Impressora". Esses tipos podem ser utilizados para categorizar e organizar os consumíveis. O idTipo serve como um identificador único para cada tipo de consumível na tabela.

Na tabela "nota_encomenda" são registradas informações detalhadas sobre notas de encomenda, como número da nota, descrição, fornecedor, obra, datas importantes e informações de desconto. Os campos opcionais permitem registrar dados adicionais, como datas de validação e envio, número de itens, justificativas e descontos. As chaves estrangeiras garantem a integridade referencial, garantindo que os fornecedores e obras associados sejam válidos, enquanto os índices ajudam a otimizar as consultas relacionadas.

A tabela "consumivel_servicos" armazena informações sobre consumíveis e serviços. Cada entrada na tabela é identificada por um número único e automático. Os campos incluem detalhes como a quantidade, valor, data de aquisição, documentos relacionados (como arquivos PDF), número da fatura e uma breve descrição.

Chaves estrangeiras conectam os consumíveis e fornecedores a outras tabelas do sistema, garantindo a integridade dos dados. Índices são criados para otimizar a recuperação de informações relacionadas. Este design permite o rastreamento eficiente de consumíveis e serviços ao longo do tempo, facilitando a gestão e análise desses dados.

4.2.4 Faturas, Material e Unidade

O armazenamento das faturas são essenciais para o controle financeiro da empresa. As tabelas referente as faturas armazenam as informações relacionadas às transações financeiras da empresa, incluindo detalhes sobre valores, datas, quantidades e descontos. A tabela "material" registra os diferentes tipos de materiais e produtos que a empresa adquiriu, contendo informações como descrição e referências. Já a tabela "unidade" define as unidades de medida utilizadas para quantificar os materiais, como quilogramas, litros, metros, entre outros, garantindo a padronização e precisão no controle de estoque e nas operações de compra e consumo. A Figura 4.5 ilustra as tabelas referente a fatura, material e unidade de medida.

A tabela "factura_obra" registra informações sobre faturas de obras, como número da fatura, descrição, fornecedor, obra, data e informações de desconto. Os campos opcionais permitem registrar dados como o caminho do arquivo PDF, tipo de mídia, tamanho do arquivo, número de itens, valores e percentuais de desconto, acertos e se o desconto inclui IVA.

As chaves estrangeiras garantem a integridade referencial, assegurando que os fornecedores e obras associados sejam válidos, enquanto os índices ajudam a otimizar as consultas relacionadas.

A tabela "factura_cliente_item" registra informações sobre os itens incluídos nas faturas dos clientes, como quantidade, valor, IVA e descrição de cada item. Os campos obrigatórios garantem que os dados essenciais estejam sempre presentes, enquanto a chave primária assegura a identificação única de cada item na tabela.

Na tabela "factura_obra_material" são armazenados os dados de materiais incluídos nas faturas de obras, como quantidade, valor, IVA, descontos e referências aos materiais, unidades e faturas associados. As chaves estrangeiras idMaterial, idUnidade e idFactura garantem a integridade referencial, assegurando que os materiais, unidades e faturas associados sejam válidos e consistentes.

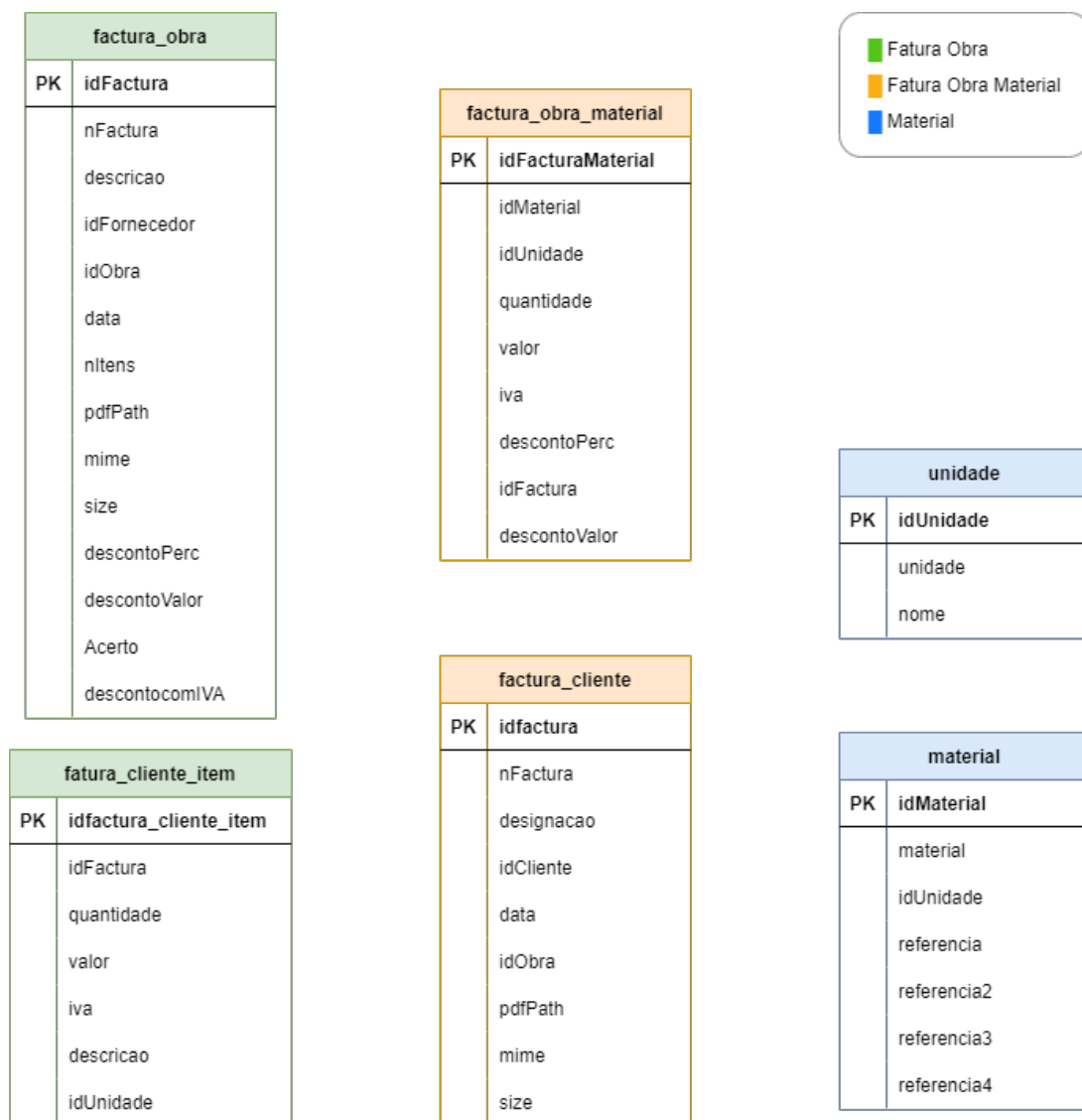


Figura 4.5: Tabelas do banco de dados referente aos recursos sobre as faturas e materiais.

A tabela "factura_cliente" registra as informações sobre as faturas emitidas para clientes, incluindo número da fatura, designação, cliente, data e obra associada. Os campos opcionais permitem registrar dados como o caminho do arquivo PDF, tipo de mídia, tamanho do arquivo. Cada fatura é identificada de maneira única, e as informações essenciais como cliente, data e obra associada são sempre registradas, facilitando a gestão e a análise das faturas emitidas.

As informações sobre unidades de medida, que são usadas para quantificar materiais e outros itens em diferentes tabelas do banco de dados, são registrada na tabela "unidade". Cada unidade é identificada de forma única e possui um nome e sua respectiva sigla.

A tabela "material" armazena informações sobre os materiais utilizados em diversos processos, como obra, pedido de cotação, encomenda, devolução ou uso interno da empresa. Cada material é identificado de forma única, contém a descrição, possibilita até quatro referências e está associado a uma unidade de medida específica. A chave estrangeira idUnidade assegura que a unidade de medida associada a cada material seja válida, referenciando a tabela unidade.

4.2.5 Cliente, Obra e Nota de Crédito

Armazenar informações dos clientes são essenciais para a localização e emissão de faturas para os donos das obras. Já o registro das obras é um recurso chave para a gestão de alocações de recursos, mão de obra e o acompanhamento do progresso e dos custos de cada empreendimento. A tabela de "ncredito_obra" permite o registro das notas de crédito que representam créditos concedidos aos clientes, detalhando valores, descontos, datas, entre outras informações essenciais para o controle financeiro e a correção de discrepâncias em transações comerciais. A Figura 4.6 apresenta as tabelas que armazenam as informações referente aos clientes, obras e notas de crédito.

A tabela "cliente" armazena informações sobre os clientes, incluindo nome, endereço, informações de contato e identificadores fiscais. A presença de campos opcionais permite a flexibilidade necessária para capturar informações adicionais conforme disponível,

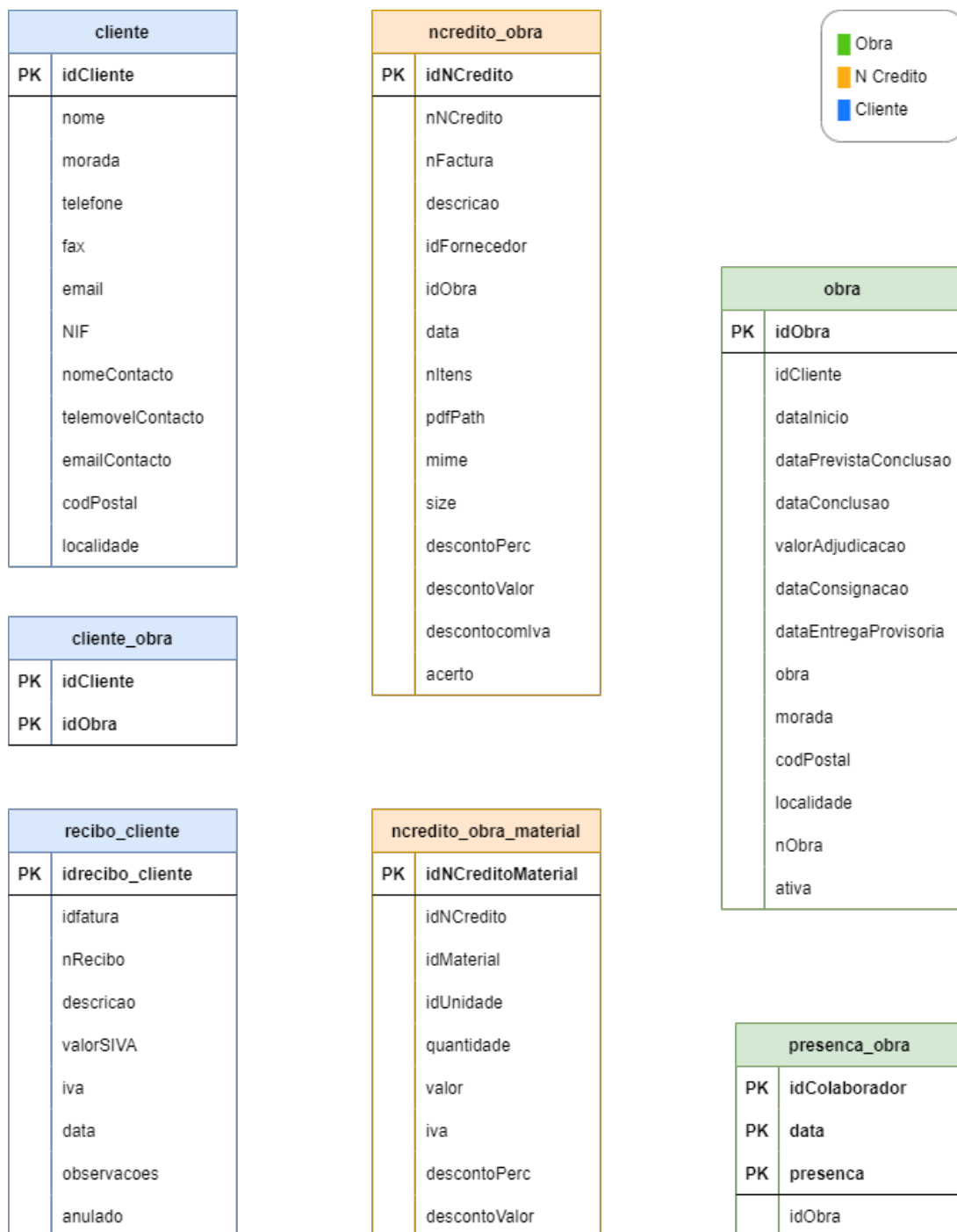


Figura 4.6: Tabelas do banco de dados referente aos recursos de cliente, nota de crédito e obras.

enquanto os campos obrigatórios, como nome, morada e NIF, asseguram que os dados cruciais estejam sempre presentes. Isso facilita a gestão de clientes e o suporte a operações de negócios eficientes.

A tabela "cliente_obra" é uma tabela de associação (tabela de junção) que estabelece uma relação muitos-para-muitos entre clientes e obras. Essa tabela permite que um cliente esteja associado a várias obras e que uma obra tenha vários clientes associados.

A tabela "recibo_cliente" armazena informações dos recibos emitidos para clientes, incluindo referências à fatura associada, número do recibo, descrição, valores, IVA, data, se foi anulado e observações.

Na tabela "ncredito_obra" são armazenadas as informações sobre notas de crédito emitidas para obras, incluindo número da nota de crédito, número da fatura, descrição, fornecedor, obra, data, número de itens, documentos relacionados (como arquivos PDF), informações de desconto e acerto. Esta estrutura permite o acompanhamento das transações financeiras relacionadas às obras.

A tabela "ncredito_obra_material" registra informações sobre os materiais incluídos em notas de crédito emitidas para obras. Isso inclui dados como quantidade, valor, IVA, descontos e referências aos materiais, unidades e a nota de crédito associada.

Os índices são criados para otimizar as consultas relacionadas aos campos idNCredito, idMaterial e idUnidade. As chaves estrangeiras garantem a integridade referencial, assegurando que os materiais, unidades e notas de crédito associadas sejam válidos, e estão configuradas para excluir em cascata os registros de materiais associados caso a nota de crédito correspondente seja excluída da tabela de notas de crédito. Isso ajuda a manter a consistência dos dados no banco de dados.

A tabela "obra" armazena os dados das obras, incluindo dados como cliente associado, datas importantes, valores, endereço, descrição, número e status da obra. A chave estrangeira idCliente garante a integridade referencial, assegurando que o cliente associado à obra seja válido, referenciando a tabela cliente.

Essa estrutura permite um gerenciamento das informações relacionadas às obras, facilitando a análise, planejamento e execução das atividades relacionadas à construção ou

realização de projetos específicos.

A tabela "presenca_obra" registra a presença de colaboradores em obras de acordo com a data. Isso permite o acompanhamento da presença dos colaboradores ao longo do tempo em diferentes obras. A chave primária composta (idColaborador, data, presenca) garante que cada registro seja único para um colaborador em uma data específica, garantindo a integridade dos dados. Essa estrutura permite o registro da presença dos colaboradores nas obras ao longo do tempo, facilitando o controle e a gestão da força de trabalho em diferentes projetos.

4.2.6 Equipamento e Viatura

Os equipamentos e viaturas fazem parte do patrimônio da empresa, portanto o registro desses recursos são essenciais para a contabilidade financeira dos custos operacionais da empresa. Esses recursos ajudam a empresa a controlar seus ativos e garantir a continuidade das operações. A Figura 4.7 apresenta as tabelas que gerenciam os equipamentos e as viaturas da empresa.

A tabela "equipamento" registra informações sobre os equipamentos, incluindo dados como nome, descrição, data de aquisição, estado, se tem leasing, IVA, valores financeiros, número de rendas, fornecedor associado e observações. Essa estrutura permite um gerenciamento dos equipamentos utilizados na organização. A chave estrangeira idFornecedor garante a integridade referencial, assegurando que o fornecedor associado seja válido, referenciando a tabela fornecedor.

A tabela "leasing_equipamento" é responsável por armazenar as informações sobre as operações de leasing relacionadas aos equipamentos. Isso inclui dados financeiros, datas, identificadores de equipamentos e observações adicionais. Essa estrutura permite o acompanhamento financeiro e a gestão dos contratos de leasing associados aos ativos da organização.

A tabela "amortizacao_equipamento" registra informações sobre a amortização de

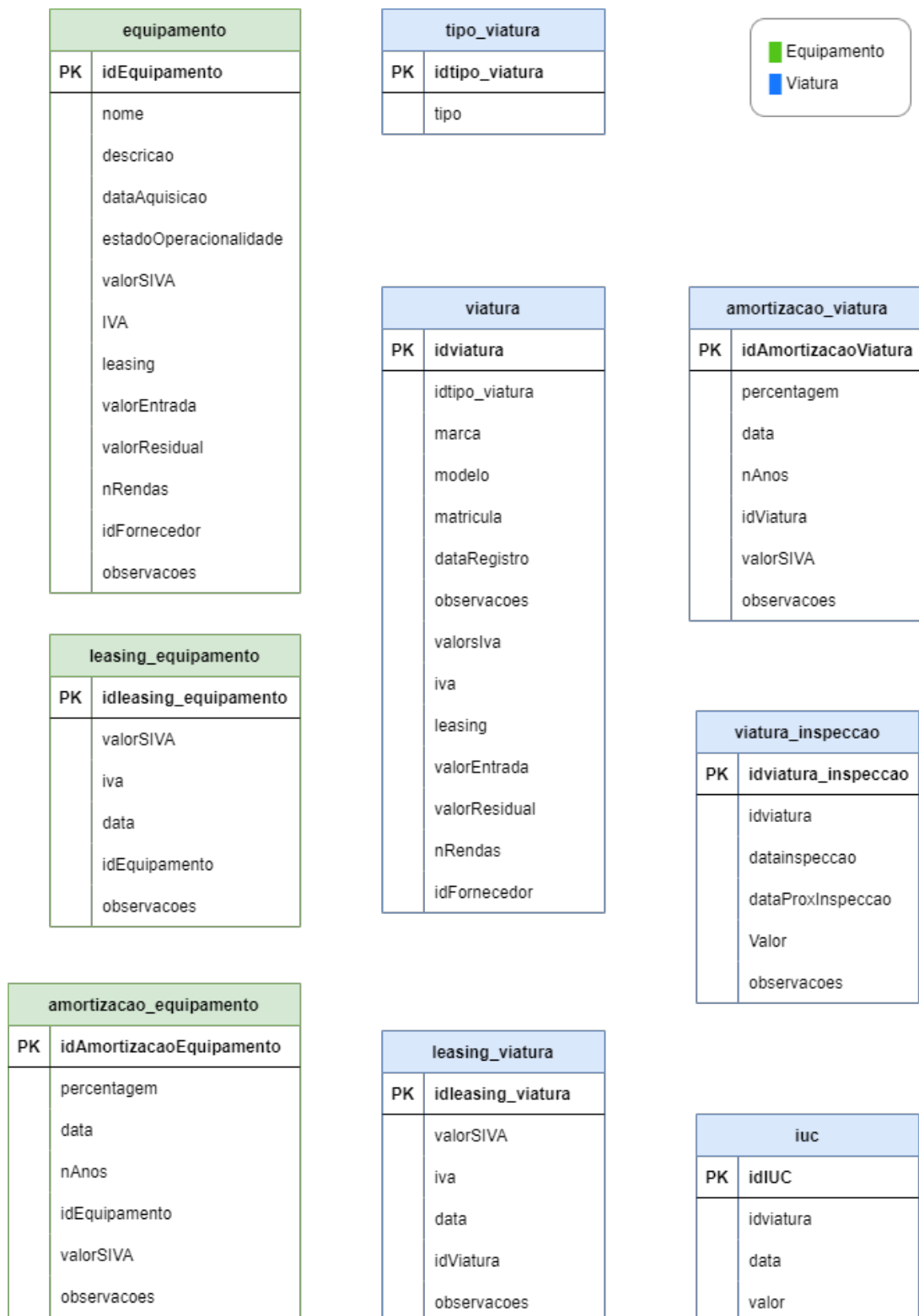


Figura 4.7: Tabelas do banco de dados referente aos recursos de equipamento e viatura.

equipamentos, incluindo a porcentagem da amortização, datas, duração em anos, identificadores de equipamentos e observações adicionais. Essa estrutura permite um controle da amortização dos equipamentos, facilitando o acompanhamento financeiro e a gestão dos ativos da organização ao longo do tempo.

A tabela "tipo_viatura" registra os diferentes tipos de viaturas, permitindo a categorização e organização das viaturas no sistema.

Na tabela "viatura" são armazenada as informações sobre as viaturas, incluindo dados sobre tipo, marca, modelo, matrícula, data de aquisição, IVA, valores financeiros, leasing, número de rendas, fornecedor e observações adicionais. As chaves estrangeiras garantem a integridade referencial, assegurando que o tipo de viatura e o fornecedor associados sejam válidos, referenciando as tabelas tipo_viatura e fornecedor. Essa estrutura auxilia na gestão de ativos, acompanhamento de manutenção e análises financeiras relacionadas às viaturas utilizadas pela organização.

A tabela "leasing_viatura" registra informações sobre os contratos de leasing de viaturas, incluindo identificadores de leasing, valores, impostos, datas, viaturas associadas e observações. Os campos opcionais permitem registrar dados adicionais, como observações sobre o contrato. A chave primária garante a integridade dos registros, enquanto as chaves estrangeiras e os índices ajudam a otimizar as consultas relacionadas e manter a integridade referencial.

A tabela "amortizacao_viatura" registra informações detalhadas sobre a amortização das viaturas, incluindo identificadores de amortização, percentagens, datas, durações, viaturas associadas e observações. Os campos opcionais permitem registrar dados adicionais, como observações sobre a amortização. A chave primária garante a integridade dos registros.

A tabela "viatura_inspecao" registra informações sobre as inspeções das viaturas, como identificadores de inspeção, viaturas associadas, datas de inspeção, próximas inspeções, valores e observações. Os campos opcionais permitem registrar dados adicionais, como a data da próxima inspeção e observações sobre a inspeção. A chave primária garante a integridade dos registros, enquanto os índices ajudam a otimizar as consultas

relacionadas.

A tabela "iuc" registra informações sobre o Imposto Único de Circulação (IUC) das viaturas, como identificadores de IUC, viaturas associadas, datas e valores do imposto. Os campos opcionais permitem registrar dados adicionais, como a data de pagamento e o valor do imposto. A chave primária garante a integridade dos registros, enquanto os índices ajudam a otimizar as consultas relacionadas.

4.2.7 Seguro, Valores Agregados e Padronização

Os seguros de automóveis e de trabalho são atribuídos aos custos operacionais das obras em vigor. O armazenamento desses recursos são essenciais para consultas das despesas da empresa. Já os valores agregados são informações sobre os benefícios financeiros adicionais ou custos extras associados a produtos ou serviços, como impostos, descontos ou encargos. A tabela 4.8 apresenta os registros dos seguro de automóvel, seguro trabalho, IVA, juros, subsídio de alimentação e dados de padronização da empresa.

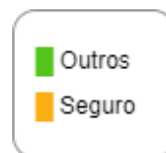
A tabela "subsídio_alimentacao" registra informações detalhadas sobre os subsídios de alimentação, incluindo identificadores de subsídio, valores, e datas de entrada em vigor. Os campos opcionais permitem registrar dados adicionais, como a data de entrada em vigor do subsídio. A chave primária garante a integridade dos registros.

A tabela "iva" registra informações detalhadas sobre os valores de IVA (Imposto sobre o Valor Acrescentado), incluindo identificadores de registro, datas, valores de IVA calculado e recebido, valores de IVA calculado e pago, e valores de IVA devolvido e pago. Os campos opcionais permitem registrar dados adicionais, como datas e valores específicos de IVA. A chave primária garante a integridade dos registros.

A tabela "juros" registra informações detalhadas sobre juros, incluindo identificadores de registro, títulos, valores, datas de início e fim. Os campos opcionais permitem registrar dados adicionais, como as datas de vigência dos juros. A chave primária garante a integridade dos registros.

A tabela "dados_aln" registra informações genéricas relacionadas aos dados de ALN

subsídio_alimentacao	
PK	idSubsidioAlimentacao
	valor
	data_entrada_vigor



iva	
PK	idiva
	data
	ivaCalculadoRecebido
	ivaCalculadoPago
	ivaDevolvidoPago

seguro_automovel	
PK	id SeguroAutomovel
	idviatura
	premio
	idFornecedor
	dataApolice
	observacoes

juros	
PK	idJuros
	titulo
	valor
	dataInicio
	dataFim

dados_aln	
PK	idDadosALN
	item
	valor

seguro_trabalho	
PK	id SeguroTrabalho
	idFornecedor
	dataVencimento
	observacoes
	taxa
	RespCivil
	massaSalarial
	dataApolice

Figura 4.8: Tabelas do banco de dados referente aos recursos de seguro, valores agregados (juros, iva, subsídio de alimentação) e customizações dos dados da empresa.

(Alfanuméricos), contendo identificadores de registro, itens e seus valores associados. Os campos opcionais permitem a inclusão de valores específicos para cada item registrado. A chave primária garante a integridade dos registros.

A tabela "seguro_automovel" registra informações sobre os seguros de automóveis, incluindo identificadores de seguro, de viatura e de fornecedor, prêmio do seguro, data da apólice e observações adicionais. Os campos opcionais permitem registrar observações sobre o seguro. As chaves estrangeiras garantem a integridade referencial, associando corretamente o seguro à viatura e ao fornecedor correspondentes.

A tabela "seguro_trabalho" registra informações sobre os seguros de trabalho, incluindo identificadores de seguro, de fornecedor, datas de vencimento da apólice, data da apólice, observações, taxa, responsabilidade civil, e massa salarial. Os campos opcionais permitem registrar observações adicionais sobre o seguro de trabalho e a data da apólice. As chaves estrangeiras garantem a integridade referencial, associando corretamente o seguro ao fornecedor correspondente.

4.3 Desenvolvimento do Servidor Backend (API)

O backend ou API, é a camada de uma arquitetura cliente-servidor, encarregada de assegurar o processamento preciso dos dados, a aplicação das regras de negócio e a entrega de respostas adequadas às solicitações dos usuários. Quando uma requisição válida é recebida, o backend interage com o banco de dados para processar os dados e satisfazer a solicitação. Essa centralização dos processamentos em um serviço separado da aplicação do usuário resulta em melhorias significativas de desempenho, especialmente em momentos de alta carga de processamento.

Além disso, o backend se comunica com o frontend, a interface do usuário, para garantir a precisão e a eficiência das respostas fornecidas. Muitas vezes, essa interação é facilitada por meio de REST APIs (Representational State Transfer), um conjunto de regras e convenções para a construção e consumo de serviços web. As REST APIs permitem que o backend expanda suas funcionalidades e dados por meio de endpoints acessíveis via

HTTP, simplificando a comunicação entre o frontend e o backend.

Ao utilizar REST APIs, o backend consegue gerenciar operações como criação, leitura, atualização e exclusão de dados (CRUD) de maneira estruturada e escalável. Isso não apenas aprimora a modularidade e a manutenção do sistema, mas também facilita a comunicação consistente e previsível entre os diferentes componentes do sistema.

Neste projeto, foi utilizado o .NET Core para o desenvolvimento da API e seguiu as convenções REST API. Além disso, o acesso e a interação com o banco de dados foram feitos usando a tecnologia Entity Framework Core (EF Core), que é uma estrutura de mapeamento objeto-relacional (ORM) fornecida pela Microsoft. Com o EF Core é possível mapear objetos de domínio para tabelas de banco de dados, realizar consultas, inserções, atualizações e exclusões usando código .NET padrão, sem a necessidade de escrever consultas SQL manualmente.

4.3.1 Estrutura de diretório

A estrutura de diretório é a organização hierárquica de pastas e arquivos dentro de um sistema de arquivos. Ela é fundamental para manter a organização e a facilidade de navegação em um projeto de software. Embora não haja uma estrutura de diretório "certa" ou "errada", existem convenções amplamente adotadas em diferentes tipos de projetos.

A escolha de uma estrutura de diretório pode influenciar a produtividade e a manutenibilidade do código ao longo do tempo. Uma boa prática é seguir convenções amplamente aceitas na comunidade de desenvolvimento relacionada ao tipo de projeto que está sendo desenvolvido.

Este projeto adere às convenções estruturais do .NET Core, que são semelhantes às do framework .NET tradicional, mas com algumas adaptações específicas para atender às necessidades do projeto e aproveitar as características únicas do .NET Core. A Figura 4.9 apresenta a estrutura de diretórios deste projeto. A seguir, são descritos os principais arquivos e diretórios do backend do sistema.

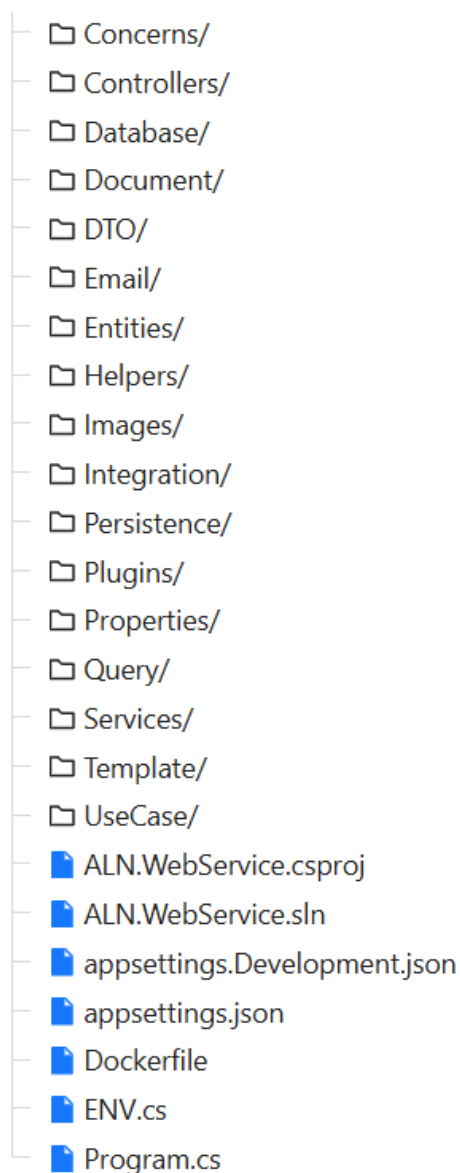


Figura 4.9: Estrutura de diretório do projeto do backend.

O arquivo *Program.cs* é um dos principais arquivos da estrutura do backend, onde são configurados todos os recursos do software, incluindo CORS, permissões de acesso por IP ou URL, execução dos controladores, Swagger, autenticação e funcionamento da API. Além disso, o diretório *Plugins/* armazena os arquivos para a configuração dos serviços da aplicação Swagger e configuração das exceções da aplicação.

Já o diretório *Controllers/* estão os arquivos de classe responsáveis por receber e validar solicitações HTTP, configurar métodos, definir endpoints, indicar rotas com autenticação e chamar o serviço apropriado para resolver a solicitação e retornar uma resposta adequada.

O diretório *Services/* contém os arquivos de classe responsáveis por processar e validar as informações do banco de dados de acordo com as regras de negócio, retornando respostas válidas para as chamadas dos controladores. Operações mais complexas, como a emissão da execução orçamental, são reservadas ao diretório *UseCase/*, que envolve vários cálculos e consultas ao banco de dados.

Para a manipulação do banco de dados, o diretório *Database/* contém os arquivos SQL necessários para criar e atualizar as estruturas de dados, enquanto o diretório *Entities/* possui os arquivos de classe que mapeiam essas entidades no banco. O diretório *Query/* é reservado para requisições mais complexas que ainda não foram desenvolvidas com o EF Core, e o diretório *Persistence* contém o arquivo de configuração de acesso ao banco de dados.

A geração de relatórios do sistema é feita a partir dos arquivos armazenados no diretório *Document/*, e a padronização dos dados transferidos nas respostas das solicitações é garantida pelos arquivos de classe presentes no diretório *DTO/* (Data Transfer Object).

O diretório *Concerns/* é um padrão de organização de código que separa e encapsula funcionalidades compartilhadas entre diferentes partes da aplicação, sendo utilizado para extrair lógica comum de modelos, controladores ou outros componentes, agrupando-a em módulos. Nele, estão presentes os arquivos de classe dos erros da aplicação e do modelo genérico de resultados lidos por páginas.

No que se refere ao envio de emails, o diretório *Email/* inclui tanto o arquivo de configuração do serviço SMTP quanto os arquivos responsáveis por encaminhar os emails de acordo com os padrões de cada recurso. Os layouts dos emails enviados pela plataforma estão definidos em formato HTML dentro do diretório *Template/*.

O diretório *Helpers/* armazena funções auxiliares no sistema, como verificação de senha, cálculo de impostos e criação de diretórios para armazenamento de documentos.

As imagens utilizadas no sistema e no envio de emails estão localizadas no diretório *Images/*. Para a integração com serviços terceirizados, o diretório *Integration/* possui os arquivos de configuração, como a API de endereços da CTT - Correios de Portugal, S.A. Configurações adicionais de serviços HTTP, HTTPS e Swagger são definidas nos arquivos JSON presentes no diretório *Properties/*.

Outros arquivos incluem *ALN.WebService.csproj* e *ALN.WebService.sln*, que armazenam informações sobre o projeto, como a versão do .NET utilizada, as bibliotecas instaladas e outras informações relevantes para a compilação. As configurações das variáveis de ambiente de produção e desenvolvimento são armazenadas, respectivamente, nos arquivos *appsettings.json* e *appsettings.Development.json*. E a definição das etapas de build da aplicação via Docker é descrita no arquivo *Dockerfile*, enquanto o arquivo de classe *ENV.cs* define outras variáveis de ambiente de serviços externos, como o serviço de busca de endereço da CTT e o serviço de email.

4.3.2 Controladores

Os controladores (controllers) são responsáveis por gerenciar as requisições HTTP recebidas pelo servidor e determinar como essas requisições devem ser manipuladas. Os controladores agem como intermediários entre o frontend (ou cliente) e a lógica de negócios implementada nos serviços. Cada controlador é mapeado para um conjunto específico de endpoints (URLs) e métodos HTTP (GET, POST, PUT, DELETE, etc.). Quando o servidor recebe uma requisição, ele encaminha essa requisição ao método apropriado do controlador.

Dentro do método do controlador, os parâmetros da requisição são recebidos e validados. Esses parâmetros podem incluir dados da URL, cabeçalhos HTTP, corpo da requisição e outros.

Após validar os parâmetros da requisição, o controlador chama os serviços apropriados que contêm a lógica de negócios necessária para processar a requisição. Isso mantém a separação de responsabilidades, permitindo que os controladores permaneçam leves e

focados em gerenciar o fluxo de dados.

Após o serviço processar a requisição, o controlador formata a resposta apropriada e a envia de volta ao cliente. Neste projeto, as respostas podem ser dados JSON, arquivos ou códigos de status HTTP, dependendo do tipo de requisição.

Um controlador em um projeto .NET Core é geralmente uma classe que herda de *Controller* ou *ControllerBase* como foi feito neste projeto, e utiliza atributos para mapear métodos a endpoints específicos. A Figura 4.10 apresenta o controlador *PedidoCotacaoController* com a definição do endpoint *ListarPedidoCotacao* para o acesso a listagem dos pedidos de cotação.

A Figura 4.10 apresenta o código de um controlador da API no ambiente .Net. Cada controlador possui um ou mais pontos de acesso (endpoints) para comunicação com ferramentas externas, podendo definir métodos que respondem a diferentes métodos HTTP. Os controladores e seus métodos podem ser anotados para permitir o acesso de usuários autenticados, utilizando a anotação "[*Authorize*]" da biblioteca *Authorization*. Além disso, os controladores e suas rotas são definidos automaticamente a partir das anotações "[*ApiController*]" e "[*Route*]" fornecidas pela biblioteca *Mvc* do .Net.

O serviço de autenticação utiliza JSON Web Token (JWT) para manter os usuários autenticados na plataforma. O JWT é uma forma segura de transmitir informações entre diferentes partes pela internet, sendo compacto, fácil de usar e baseado em JSON. Composto por três partes: cabeçalho, carga útil e assinatura, o JWT contém todas as informações necessárias para verificar sua autenticidade. O JWT é amplamente utilizado em sistemas web para autenticação e troca segura de dados entre diferentes serviços.

Os métodos do HyperText Transfer Protocol (HTTP) são ações que podem ser realizadas em serviços web. Esses métodos são padrões de comunicação que permitem enviar, buscar, remover e atualizar informações. No .Net, esses métodos podem ser atribuídos a cada *endpoint* por meio das anotações *HttpGet*, *HttpPost*, *HttpPut*, *HttpPatch*, *HttpDelete*, *HttpHead* e *HttpOptions*. A tabela 4.3 apresenta as utilidades de cada método.

Um *endpoint* é um ponto de acesso de comunicação em um serviço web que serve como interface para interação entre clientes e servidores. Em outras palavras, um *endpoint* é

```

1  using ALN.WebService.DTO;
2  using ALN.WebService.Entities;
3  using ALN.WebService.Services;
4  using Microsoft.AspNetCore.Authorization;
5  using Microsoft.AspNetCore.Mvc;
6
7  namespace ALN.WebService.Controllers;
8
9  [Authorize]
10 [ApiController]
11 [Route("[controller]")]
12 public class PedidoCotacaoController : ControllerBase
13 {
14     private readonly PedidoCotacaoService _service;
15
16     public PedidoCotacaoController(PedidoCotacaoService service)
17     {
18         _service = service;
19     }
20
21     [HttpGet("[action]")]
22     public DTOListPedidoCotacao ListarPedidoCotacao(
23         [FromQuery(Name = "page")] int page = 1,
24         [FromQuery(Name = "pageSize")] int pageSize = 100,
25         [FromQuery(Name = "search")] string? search = null,
26         [FromQuery(Name = "orderBy")] string orderBy = "obra",
27         [FromQuery(Name = "startTime")] DateOnly? startTime = null,
28         [FromQuery(Name = "endTime")] DateOnly? endTime = null
29     ) => _service.ListarPedidoCotacao(page, pageSize, search, startTime, endTime, orderBy);
30
31     [HttpGet("[action]/{idPedido}")]
32     public DTOPedidoCotacaoToUpdate GetByIdToUpdate(int idPedido) {...}
33
34     [HttpGet("[action]")]
35     public DTOPedidoCotacaoNumero GetNewCodePedidoCotacao() {...}
36
37     [HttpPost("[action]")]
38     public IActionResult PostPedidoCotacao(...) {...}
39
40     [HttpPut("[action]/{idPedido}")]
41     public IActionResult PutPedidoCotacao(...) {...}
42
43     [HttpPost("[action]")]
44     public DTOFileResponse UploadFilePedidoCotacao(...) {...}
45 }

```

Figura 4.10: Controlador do pedido de cotação.

Tabela 4.3: Métodos http da biblioteca MVC do ASP.NET Core.

Método	Utilidade
HttpGet	Solicita dados de um recurso específico no servidor.
HttpPost	Envia dados ao servidor para criar um novo recurso ou realizar processamento.
HttpPut	Substitui um recurso existente ou criar um novo recurso se ele não existir
HttpPatch	Aplica modificações parciais a um recurso existente.
HttpDelete	Remove um recurso específico no servidor.
HttpHead	Solicita os cabeçalhos de resposta de um recurso, sem o corpo da resposta.
HttpOptions	Descreve as opções de comunicação para o recurso de destino, como os métodos HTTP suportados.

uma URL específica onde um recurso ou serviço pode ser acessado por meio de métodos HTTP.

Cada *endpoint* é projetado para realizar uma função específica, como recuperar dados de um banco de dados, enviar informações para processamento ou modificar recursos existentes. Na lista abaixo estão os *endpoints* do pedido de cotação, obtidos pela documentação do Swagger. O acesso aos endpoints contém o prefixo "localhost:3000/". Todos os endpoints desenvolvidos neste projeto foram anexados no Apêndice C.

PedidoCotacao:

- GET /PedidoCotacao/ListarPedidoCotacao
- GET /PedidoCotacao/GetByIdToUpdate/{idPedido}
- GET /PedidoCotacao/GetNewCodePedidoCotacao
- POST /PedidoCotacao/PostPedidoCotacao
- PUT /PedidoCotacao/PutPedidoCotacao/{idPedido}
- POST /PedidoCotacao/UploadFilePedidoCotacao

4.3.3 Serviços

Serviços são componentes ou módulos de software que fornecem funcionalidades específicas para uma aplicação. Além de serem responsáveis por lidar com tarefas e processos complexos, como manipulação de dados, autenticação de usuários, integração com sistemas externos, entre outros. São projetados para serem reutilizáveis, modulares e independentes, facilitando a manutenção, o desenvolvimento e a escalabilidade do sistema. A Figura 4.11 apresenta os serviços do módulo de pedido de cotação.

O método construtor `PedidoCotacaoService` define as dependências de log, banco de dados e o diretório que armazenará os arquivos PDF dos pedidos de cotação. Os demais métodos oferecem serviços que processam as demandas dos controladores.

No método `ListarPedidoCotacao`, é realizada uma consulta na base de dados para encontrar os fornecedores, materiais e obras relacionadas a cada pedido, considerando os parâmetros recebidos da requisição (conforme apresentado na Figura 4.10). Após coletar as informações, os dados são preparados de acordo com as especificações do DTO correspondente, neste caso, o `DTOListPedidoCotacao`, e as informações são retornadas ao controlador que solicitou o serviço.

O método `UploadFilePedidoCotacao` é responsável por controlar o armazenamento de arquivos PDF. Ao receber um arquivo, ele o envia para o método `SaveFile`, que o armazena no diretório especificado pela variável `__PDF_Directory` (apresentada na Figura 4.10).

```

1  namespace ALN.WebService.Services;
2
3  public class PedidoCotacaoService
4  {
5      private readonly ILogger<PedidoCotacaoService> _logger;
6      private readonly AlnDatabaseContext _db;
7      private string _PDF_Directory;
8
9      public PedidoCotacaoService(...) {...}
10
11     public DTOListPedidoCotacao ListarPedidoCotacao(...) {...}
12
13     public DTOFileResponse UploadFilePedidoCotacao(...) {...}
14
15     public DTOPedidoCotacaoNumero GetNewCodePedidoCotacao(...) {...}
16
17     public void InserirPedidoCotacao(...) {...}
18
19     public DTOPedidoCotacaoToUpdate GetByIdToUpdate(...) {...}
20
21     public void UpdatePedidoCotacao(...) {...}
22 }

```

Figura 4.11: Serviço do módulo de pedido de cotação.

Após o armazenamento do arquivo, as informações são retornadas ao controlador por meio do `DTOFileResponse`.

A inserção de um novo pedido de cotação exige um código exclusivo para identificá-lo. Para isso, foi desenvolvido o método `GetNewCodePedidoCotacao`, que identifica o último código emitido e gera um novo código para o próximo pedido. O método `InserirPedidoCotacao` é responsável por inserir um novo pedido. Ao receber as informações necessárias, ele armazena os dados no banco de dados. Além do registro, um e-mail com o pedido de cotação é enviado automaticamente para cada fornecedor associado ao pedido.

A atualização de um pedido de cotação requer informações detalhadas. Para isso, foi desenvolvido o método `GetByIdToUpdate`, que fornece dados adicionais sobre o pedido, como informações dos fornecedores e detalhes dos materiais. Para realizar a atualização, foi criado o método `UpdatePedidoCotacao`. Ao atualizar as informações no banco de dados, um e-mail é enviado aos fornecedores, seguindo as mesmas abordagens utilizadas ao inserir um novo pedido.

4.3.4 Execução Orçamental

O levantamento dos custos e do período de execução de cada obra é realizado por meio da execução orçamental. Para atingir esses resultados, é necessário calcular todas as despesas incorridas pela empresa durante a obra, como custos com transporte de materiais, mão-de-obra, equipamentos, entre outros custos operacionais.

O cálculo da execução orçamentária é feito com base no período estabelecido em meses. Para cada mês, são considerados os custos com materiais, mão-de-obra direta e indireta, consumíveis e serviços, além da proporção do seguro de automóvel, Imposto Único de Circulação (IUC), inspeções periódicas, *leasing* de veículos, *leasing* de equipamentos e outros custos, se houver.

No caso de materiais devolvidos aos fornecedores por diversos motivos, as notas de crédito referentes às devoluções são lançadas no sistema. Esse valor é descontado do custo total com materiais no cálculo final da execução orçamentária.

4.4 Desenvolvimento do Frontend

A Figura 4.12 apresenta a organização dos diretórios do frontend da aplicação web. A Figura 4.12a exibe a estrutura de diretórios e arquivos na raiz do projeto, enquanto a Figura 4.12b detalha a organização das pastas para cada página da aplicação. O padrão adotado para os diretórios das páginas segue o endereço de acesso correspondente a cada recurso. A seguir, são descritos os principais arquivos e diretórios do frontend da aplicação web.

Os arquivos *App.tsx* e *App.css* definem o componente principal da aplicação, *App*. Eles contêm regras de CSS que se aplicam a vários componentes, garantindo uma aparência consistente, e configuram a estrutura inicial da aplicação, incluindo o roteamento e outros componentes globais. Já os arquivos *index.css* e *index.tsx* são responsáveis por importar estilos globais e renderizar a aplicação React no DOM, sendo *index.tsx* o ponto de entrada principal onde a aplicação é montada no *index.html*. O arquivo *main.tsx* é geralmente utilizado para inicializações adicionais, como configurações de provedores de contexto ou

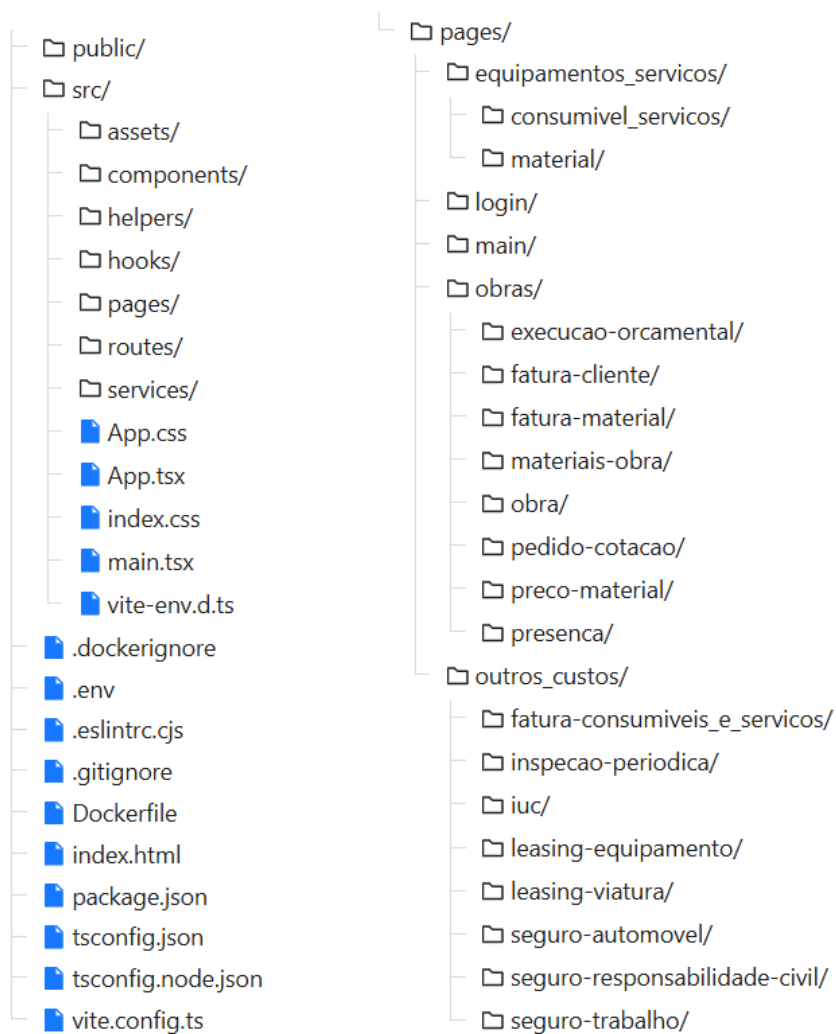


Figura 4.12: Estrutura de Diretório do Frontend

outras inicializações globais.

O diretório *pages/* organiza as páginas da aplicação, com cada arquivo representando uma página específica, facilitando a navegação e a estruturação da aplicação. O diretório *components/* contém componentes React reutilizáveis, que podem variar de elementos simples como botões a componentes complexos como formulários completos. Manter os componentes em um diretório separado facilita a reutilização e a manutenção. O diretório *routes/* define as rotas da aplicação, utilizando bibliotecas como *react-router-dom*, permitindo que a aplicação suporte navegação entre diferentes páginas de maneira eficiente.

O diretório *public/* contém arquivos estáticos servidos diretamente pelo servidor, incluindo o arquivo *index.html*, que é o ponto de entrada da aplicação React e define a estrutura básica do HTML onde a aplicação será montada. O diretório *assets/* armazena recursos estáticos como imagens, fontes e ícones utilizados na aplicação, ajudando a organizar os ativos necessários.

O diretório *hooks/* armazena hooks personalizados, que são funções que permitem usar o estado e outros recursos do React sem escrever classes, facilitando a lógica compartilhada entre componentes. O diretório *helpers/* contém funções auxiliares utilizadas em diferentes partes da aplicação, como utilitários para formatação de datas, manipulação de strings e cálculos específicos. O diretório *services/* gerencia a comunicação com APIs externas ou outras fontes de dados, incluindo funções para fazer requisições HTTP, manipular respostas e gerenciar erros.

Para a implantação da aplicação, o arquivo *Dockerfile* define as instruções para criar uma imagem Docker da aplicação, especificando como ela deve ser construída e executada dentro de um contêiner, enquanto o arquivo *.dockerignore* especifica quais arquivos e diretórios devem ser ignorados pelo Docker ao construir a imagem.

Falando sobre os arquivos base e de configuração, o arquivo *.env* contém variáveis de ambiente utilizadas na aplicação, como URLs de APIs e chaves de configuração. O arquivo *.eslintrc.cjs* configura regras e plugins para o ESLint, uma ferramenta de análise estática de código para identificar problemas em código JavaScript/TypeScript. O arquivo

package.json armazena informações sobre o projeto, incluindo dependências, scripts de build e outras configurações.

Outros arquivos de configuração incluem o arquivo *vite.config.ts*, que configura o Vite, incluindo plugins, aliases e outras opções de build, e o arquivo *vite-env.d.ts*, que define tipos específicos para a configuração do Vite. Os arquivos *tsconfig.json* e *tsconfig-node.json* definem a configuração do TypeScript para o projeto, especificando opções de compilação e paths de módulos.

O desenvolvimento frontend e o design de interface de usuário (UI) são componentes essenciais na criação de experiências digitais modernas e envolventes.

O frontend abrange todas as partes visíveis e interativas de um site ou aplicação web, enquanto a UI se foca nos elementos visuais e na interação do usuário com o sistema.

No frontend, utilizamos tecnologias como HTML, CSS e JavaScript para construir interfaces que não só sejam visualmente atraentes, mas também funcionais e responsivas. Ferramentas e frameworks como React, Angular e Vue.js permitem a criação de aplicativos dinâmicos e eficientes, oferecendo uma base sólida para desenvolvedores implementarem soluções complexas de maneira eficaz.

O design de interface de usuário se preocupa em criar uma experiência intuitiva e agradável para o usuário. Isso inclui a disposição lógica dos elementos, a escolha adequada de cores, tipografia, ícones, e a garantia de que a interface seja acessível a todos os usuários. Uma boa UI não apenas melhora a estética de uma aplicação, mas também sua usabilidade, tornando mais fácil e agradável a interação do usuário com o produto.

Combinando o desenvolvimento frontend com um design de UI bem pensado, é possível criar produtos digitais que não só atendam às necessidades dos usuários, mas também superem suas expectativas, oferecendo uma experiência fluida e envolvente.

A Interface de Usuário (UI) é a parte de uma aplicação ou sistema que permite a interação entre o usuário e a máquina. Ela inclui todos os elementos visuais e interativos que facilitam a comunicação e a execução de tarefas pelo usuário. A UI é crucial para a experiência do usuário (UX) e pode ser o fator decisivo entre uma aplicação ser intuitiva e agradável ou frustrante e confusa.

A interface do usuário – user interface (UI) – é o espaço onde os usuários interagem com um site, programa, aplicativo, etc. Isso pode incluir telas de exibição, um mouse, teclados e a aparência de uma área de trabalho.

O design da interface do usuário deve ser o mais amigável possível para pessoas com deficiência. O objetivo é criar uma interface suave, intuitiva e visualmente agradável. As pessoas podem nem mesmo perceber ativamente como uma boa UI é fácil de usar, mas sentirão isso, o que as fará voltar.

ReactJS é uma biblioteca JavaScript desenvolvida pelo Facebook para construir interfaces de usuário, especialmente para aplicações de página única (SPAs). Lançada em 2013, ela rapidamente se tornou uma das ferramentas mais populares no desenvolvimento frontend devido à sua eficiência, flexibilidade e robustez.

O front-end é uma abstração, simplificando o componente subjacente pelo fornecimento de uma interface amigável, como por um exemplo um navegador de Internet, ou um formulário para um determinado usuário.

Neste exemplo, a UI do formulário de login é projetada para ser simples e intuitiva, com campos de entrada claramente rotulados e um botão de submissão destacado. O design responsivo e o feedback visual são implementados através de estilos CSS.

Uma boa UI é fundamental para garantir que os usuários possam interagir eficientemente com a aplicação, proporcionando uma experiência agradável e produtiva.

O frontend é a parte de uma aplicação web que interage diretamente com os usuários. Ele é responsável por tudo o que o usuário vê e interage, incluindo layouts, botões, imagens, textos e outros elementos visuais. O desenvolvimento frontend envolve a criação de interfaces de usuário (UIs) e a experiência do usuário (UX).

No desenvolvimento do frontend foi utilizado

Principais Tecnologias e Ferramentas HTML (HyperText Markup Language):

A linguagem de marcação padrão usada para criar páginas web. Define a estrutura do conteúdo na web. CSS (Cascading Style Sheets):

Usado para descrever a apresentação de um documento escrito em HTML. Controla o layout, cores, fontes e outros aspectos visuais. JavaScript:

Uma linguagem de programação que permite a implementação de elementos interativos em sites. Essencial para adicionar dinamismo e funcionalidades avançadas. Frameworks e Bibliotecas:

React: Biblioteca JavaScript mantida pelo Facebook, usada para construir interfaces de usuário. Utiliza um modelo baseado em componentes. Angular: Framework JavaScript desenvolvido pelo Google, ideal para construir aplicações web de grande escala. Vue.js: Um framework JavaScript progressivo para construir interfaces de usuário. Conhecido por sua simplicidade e flexibilidade. Pré-processadores CSS:

Sass e LESS: Ferramentas que estendem o CSS com recursos como variáveis, aninhamento e mixins. Ferramentas de Build:

Webpack: Um empacotador de módulos JavaScript que permite a combinação de vários arquivos em um único arquivo para otimizar a performance. Gulp e Grunt: Ferramentas de automação de tarefas que ajudam a automatizar tarefas repetitivas como minificação, compilação e linting. Gerenciamento de Estado:

Redux: Biblioteca para gerenciamento de estado, frequentemente usada com React.

4.4.1 Funcionalidades

O sistema oferece diversas funcionalidades organizadas em menus específicos, como demonstrado na Figura 4.13. As páginas de cada menu estão detalhadas no Apêndice D. A seguir, é apresentada uma descrição detalhada de cada item do menu.

Começando pelo menu *Equipamento e Serviços*, mostrado na Figura 4.13a, funcionalidades relacionadas a materiais, consumíveis e serviços podem ser acessadas.

A página de *Material* (Figura D.1) gerencia os materiais adquiridos para obras e uso interno da empresa. Já a página *Consumível e Serviços* (Figura D.2) controla os produtos e serviços utilizados na empresa.

Em seguida, o menu *Obras*, apresentado na Figura 4.13b, contém todos os recursos relacionados às obras, incluindo execução orçamental, presenças dos colaboradores, preços pagos pelos materiais, materiais utilizados nas obras, faturas dos materiais e dos clientes,

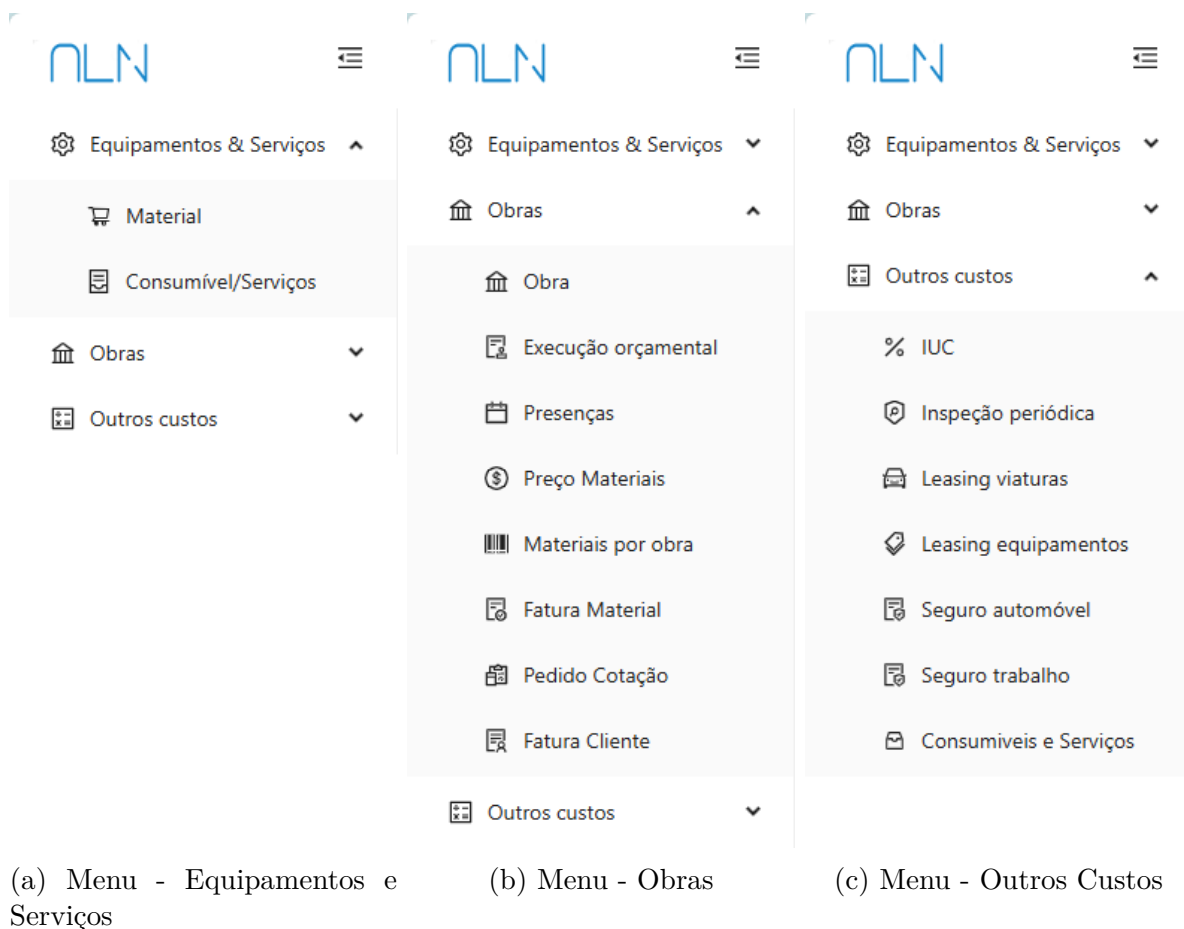


Figura 4.13: Menus das funcionalidades do frontend

e também a emissão de pedidos de cotações. A página *Obra* (Figura D.3) controla as obras ativas e inativas no sistema. A página *Materiais por obra* (Figura D.7) apresenta um relatório em formato de tabela com informações sobre os materiais utilizados em cada obra, como quantidade, preço, fornecedor e fatura. A página *Presenças* (Figura D.5) fornece uma folha de presenças onde é feito o controle dos colaboradores nas obras, permitindo a emissão de relatórios de presença.

A página *Fatura Material* (Figura D.8) gerencia as faturas de compras de materiais, possibilitando o lançamento das faturas e o anexo de documentos. A página de *Fatura Cliente* (Figura D.10) controla as faturas emitidas aos clientes, permitindo o lançamento de faturas e seus anexos. A página *Preço Materiais* (Figura D.6) emite um relatório em formato de tabela dos preços dos materiais ao longo do tempo, permitindo a visualização

da variação de preços. A página *Pedido Cotação* (Figura D.9) controla as cotações emitidas aos fornecedores, permitindo o anexo de arquivos e o envio automático de e-mails para cada fornecedor.

A página *Execução Orçamental* (Figura D.4) é especial, pois apresenta relatórios dos custos para todas as obras, obras ativas ou uma obra específica. O relatório inclui um resumo dos valores investidos em materiais, mão de obra, custos adicionais e impostos, além de gráficos de custos por mês, período de execução e execução financeira.

Finalmente, o menu *Outros Custos*, mostrado na Figura 4.13c, oferece funcionalidades relacionadas ao controle de custos internos da empresa, incluindo leasings, seguros, inspeções e impostos.

Falando sobre leasing, a página *Leasing viatura* (Figura D.13) gerencia todas as rendas pagas no financiamento das viaturas da empresa, enquanto a página *Leasing equipamento* (Figura D.14) gerencia todas as rendas pagas no financiamento de equipamentos da empresa.

No que diz respeito a seguros, a página *Seguro automóvel* (Figura D.15) gerencia os custos anuais com os seguros de automóveis, e a página *Seguro trabalho* (Figura D.16) gerencia os custos com os seguros de trabalho para os colaboradores da empresa.

Para inspeções e impostos, a página *Inspeção periódica* (Figura D.12) controla as inspeções realizadas nas viaturas e equipamentos da empresa. A página *Consumíveis e Serviços* (Figura D.17) gerencia as faturas dos consumíveis e serviços utilizados pela empresa, possibilitando o lançamento das faturas e o anexo de documentos. A página *IUC* (Figura D.11) gerencia os Impostos Único de Circulação (IUC), permitindo o controle dos impostos pagos por cada viatura da empresa.

4.4.2 Adicionar e Atualizar por Etapas

Durante o desenvolvimento das interfaces da plataforma, identificou-se uma dificuldade em manter os formulários de adição e atualização em uma única tela devido à grande quantidade de informações. Para solucionar esse problema, foram criados formulários em

etapas, separando as responsabilidades de cada recurso a ser registrado.

A Figura 4.14 mostra o formulário de registro de pedido de cotação. Cada etapa deste formulário solicita informações específicas. Na primeira etapa (Figura 4.14a), são registradas todas as informações gerais relacionadas à identificação do pedido, obra vinculada, fornecedores que receberão o pedido e, se necessário, a descrição e anexo de um arquivo PDF.

Na segunda etapa (Figura 4.14b), são registrados os materiais a serem cotados, com suas quantidades e unidades de medida. A terceira e última etapa (Figura 4.14a) foi reservada para um recurso futuro, que incluirá um relatório com todas as informações inseridas nas etapas anteriores, permitindo a validação dos dados antes de confirmar o registro.

Esse modelo de formulário em etapas também foi implementado na funcionalidade de atualização de recursos. Em alguns casos, os formulários podem ter duas etapas para registros de informações mais simples ou mais de três para casos mais complexos. No entanto, independentemente da quantidade de etapas, a última etapa deve sempre ser reservada para um relatório geral das informações registradas.

Manter esse modelo de formulário ajuda a organizar melhor as informações e evita sobrecarregar a interface do usuário, tornando-a mais objetiva.

ADICIONAR PEDIDO

1

Informe os dados do pedido

2

Informe os produtos

3

Concluir

Nº de Pedido

PC-0005-2024

Data Pedido

31/05/2024

Obra

Igreja Matriz de Algosó

Nº itens

3

+ Fornecedor

Reconco

exemplo@email.com

Melo-Material de Construção, Lda

exemplo@email.com

Descrição

Pedido de cotação para obra da igreja do próximo mês

PDF Pedido (Máximo: 1)

Adicionar arquivo

Avançar

(a) Etapa 1 - Dados do pedido

ADICIONAR PEDIDO

1

Informe os dados do pedido

2

Informe os produtos

3

Concluir

Material

Cimento Normal 35kg

Quantidade

50

Unidade de Medida

SCO

Material

Cal Hidratada 20 Kg

Quantidade

35

Unidade de Medida

UNI

Material

Areia

Quantidade

20

Unidade de Medida

M3

Voltar

Avançar

(b) Menu - Obras

ADICIONAR PEDIDO

1

Informe os dados do pedido

2

Informe os produtos

3

Concluir

Clique em Enviar para gravar as informações e encaminhar o pedido aos fornecedores.

Voltar

Enviar

(c) Menu - Outros Custos

Figura 4.14: Formulário para realizar um pedido de cotação

Capítulo 5

Resultados

O desenvolvimento da plataforma de gestão da construção foi um projeto desafiador que se manteve focado no objetivo de entregar um sistema robusto e fácil de operar. Para alcançar essa meta, foram empregados recursos de UI/UX e *mockups*, com o intuito de alinhar as perspectivas visuais e a experiência do usuário de acordo com os requisitos coletados. Além disso, o uso de linguagens e ferramentas modernas contribuiu para o ciclo de vida do projeto, garantindo sua eficiência e relevância na substituição da plataforma anterior.

Durante o desenvolvimento do projeto, foram realizadas reuniões importantes e enriquecedoras para alinhar as necessidades, coletar requisitos e *feedback*, implantar a plataforma e realizar testes no ambiente de produção pelos colaboradores da empresa.

As funcionalidades de registro de presença, lançamento de faturas de materiais, consumíveis e serviços são essenciais para controlar os principais custos associados às obras. Além dessas, os registros de imposto (IUC), inspeções periódicas, *leasing* e seguros de veículos e equipamentos contribuem significativamente para o aprimoramento do controle financeiro da empresa. Juntas, essas funcionalidades proporcionam recursos importantes para uma execução orçamental mais precisa, permitindo a determinação exata dos custos e do período de execução de cada obra, favorecendo tomadas de decisões mais assertivas.

Os resultados funcionais obtidos até o momento, considerando que o sistema ainda está em fase de teste na empresa ALN, indicam um progresso significativo no desenvolvimento

da plataforma de gestão da construção. Durante a fase de testes, foram identificados pontos fortes e áreas de melhoria no sistema, permitindo ajustes e refinamentos para garantir sua eficácia e usabilidade.

Os testes realizados até o momento demonstraram que a plataforma é capaz de atender às necessidades da empresa ALN, proporcionando uma gestão mais eficiente de obras e processos administrativos. A compatibilidade com navegadores atuais e a melhoria da experiência do usuário foram aspectos positivos destacados durante os testes.

No entanto, foram identificados alguns desafios e áreas de aprimoramento, como a necessidade de ajustes na interface do usuário, a otimização de determinadas funcionalidades e a garantia da integridade e segurança dos dados. Essas questões estão sendo abordadas durante a fase de testes para garantir que o sistema atenda plenamente às expectativas da empresa.

Em resumo, os resultados preliminares dos testes na empresa ALN indicam um avanço promissor no desenvolvimento da plataforma de gestão da construção. Com ajustes e refinamentos contínuos, espera-se que o sistema atenda de forma eficaz às necessidades da empresa e contribua para a melhoria de seus processos operacionais.

Capítulo 6

Conclusão

Ao longo deste projeto, foi possível desenvolver e implantar uma plataforma de gestão da construção personalizada para atender às necessidades específicas da empresa ALN. A implementação de um sistema ERP é fundamental para integrar os diversos processos da empresa, promovendo eficiência, melhor controle financeiro e tomadas de decisão mais assertivas.

O desenvolvimento do sistema da ALN utilizando tecnologias modernas foi um passo importante para resolver os problemas de incompatibilidade e degradação enfrentados anteriormente, além de adicionar novas funcionalidades e melhorar a experiência do usuário. A comparação com soluções existentes, como Orçamentação, Cegid Primavera, Central-Gest e PHC, forneceu insights relevantes para o desenvolvimento da plataforma.

A utilização de metodologias ágeis, como Scrum e Kanban, juntamente com a implementação de Docker e Docker Compose, demonstrou ser uma abordagem eficaz para o desenvolvimento e implantação do sistema. As 135 histórias de usuário elaboradas durante o processo permitiram uma análise aprofundada dos requisitos funcionais e não funcionais, permitindo a continuidade dos processos da empresa.

As contribuições deste trabalho até o momento incluem a modernização tecnológica do sistema de gestão da ALN, a adição de novas funcionalidades e a melhoria da experiência do usuário. Os testes iniciais apontaram que a plataforma é capaz de atender às necessidades da empresa. No entanto, novas funcionalidades e áreas de aprimoramento foram

identificados durante uma visita à empresa. Atualmente, a ferramenta encontra-se em fase de teste na empresa ALN e ainda não possui um resultado definitivo. Novos *feedback* da empresa estão sendo aguardados para confirmar se o sistema atende às expectativas e necessidades previamente identificadas. A validação por parte da ALN é necessária para determinar se o objetivo proposto foi alcançado ou se ajustes adicionais são necessários.

Bibliografia

- [1] R. Hall, «Enterprise resource planning systems and organizational change: transforming work organization?» *Strategic Change*, vol. 11, n.º 5, pp. 263–270, 2002. DOI: <https://doi.org/10.1002/jsc.602>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/jsc.602>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jsc.602>.
- [2] S. Rao Siriginidi, «Enterprise resource planning in reengineering business,» en, *Bus. Proc. management J.*, vol. 6, n.º 5, pp. 376–391, dez. de 2000. DOI: <https://doi.org/10.1108/14637150010352390>.
- [3] M. Gervalla e K. Ternai, «The Impact of Industry 4.0 to the ERP Approach,» vol. 13, pp. 56–62, out. de 2019.
- [4] H. W. Harrell, L. Higgins e S. E. Ludwig, «Expanding ERP Application Software: Buy, Lease, Outsource, or Write Your Own?» *Journal of Corporate Accounting & Finance*, vol. 12, n.º 5, pp. 37–43, 2001. DOI: <https://doi.org/10.1002/jcaf.1006>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcaf.1006>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jcaf.1006>.
- [5] O. Gómez, P. Wriedt e F. Zhao, «Build or Buy: A Case Study for ERP System Selection in SMEs,» em *Interacción*, 2016. URL: <https://api.semanticscholar.org/CorpusID:33573410>.
- [6] Anturio, *Empresa especializada em tecnologias de informação*, <https://www.anturio.com/>, Accessed: 2024-05-02.

- [7] C. Primavera, *ERP para gestão dos processos de construção*, <https://pt.primaverabss.com/pt/pagina/construcao/>, Accessed: 2024-05-02.
- [8] C. ERP, *Software de Construção do CentralGest ERP*, <https://www.centralgest.com/software/construcao/>, Accessed: 2024-05-02.
- [9] PHC, *Software PHC para Construção civil*, <https://phcsoftware.com/pt/sectores/construcao-civil/>, Accessed: 2024-05-02.
- [10] N. A. Orçamentação, *Software para área da construção civil*, <https://www.anturio.com/setores/construcao/>, Accessed: 2024-05-02.
- [11] A. -. Setores, *Setores de atuação*, <https://www.anturio.com/solucoes/gestao-e-bpm-erp>, Accessed: 2024-05-02.
- [12] Cegid, *Informações sobre a empresa Cegid*. <https://www.cegid.com/ib/es/>, Accessed: 2024-05-02.
- [13] C. recursos de gestão, *Informações sobre as funcionalidades de gestão*. <https://www.centralgest.com/software/construcao/gestao-de-obras-funcionalidades>, Accessed: 2024-05-02.
- [14] C. recursos de orçamentação, *Informações sobre as funcionalidades de orçamentação*, <https://www.centralgest.com/software/construcao/orcamentacao-funcionalidades>, Accessed: 2024-05-02.
- [15] P. A. Akiki, A. K. Bandara e Y. Yu, «Adaptive Model-Driven User Interface Development Systems,» *ACM Comput. Surv.*, vol. 47, n.º 1, mai. de 2014, ISSN: 0360-0300. DOI: 10.1145/2597999. URL: <https://doi.org/10.1145/2597999>.
- [16] A. O. Penha-Junior, C. E. De Souza, G. M. Souza et al., «Challenges in the Development of a Global Software User Interface by Multicultural Teams: an Industrial Experience,» em *2021 2nd Asia Service Sciences and Software Engineering Conference*, sér. ASSE '21, Macau, Macao: Association for Computing Machinery, 2021, pp. 41–47, ISBN: 9781450389082. DOI: 10.1145/3456126.3456144. URL: <https://doi.org/10.1145/3456126.3456144>.

- [17] T. G. Gonçalves e A. R. C. da Rocha, «Development process for intelligent user interfaces: an initial approach,» em *Proceedings of the XVIII Brazilian Symposium on Software Quality*, sér. SBQS '19, Fortaleza, Brazil: Association for Computing Machinery, 2019, pp. 210–215, ISBN: 9781450372824. DOI: 10.1145/3364641.3364665. URL: <https://doi.org/10.1145/3364641.3364665>.
- [18] I. Bouchrika, L. Ait-Oubelli, A. Rabir e N. Harrathi, «Mockup-based navigational diagram for the development of interactive web applications,» em *Proceedings of the 2013 International Conference on Information Systems and Design of Communication*, sér. ISDOC '13, Lisboa, Portugal: Association for Computing Machinery, 2013, pp. 27–32, ISBN: 9781450322997. DOI: 10.1145/2503859.2503864. URL: <https://doi.org/10.1145/2503859.2503864>.
- [19] I. Bouchrika, L. Ait-Oubelli, A. Rabir e N. Harrathi, «Mockup-based navigational diagram for the development of interactive web applications,» em *Proceedings of the 2013 International Conference on Information Systems and Design of Communication*, sér. ISDOC '13, Lisboa, Portugal: Association for Computing Machinery, 2013, pp. 27–32, ISBN: 9781450322997. DOI: 10.1145/2503859.2503864. URL: <https://doi.org/10.1145/2503859.2503864>.
- [20] Figma, *Figma*, <https://www.figma.com/>, Accessed: 2024-05-17.
- [21] F. G. e Equipe Aela, *Cores Em UI: Um Guia Rápido Para Usar Em Seus Projetos*, <https://aelaschool.com/pt/designvisual/cores-em-ui-um-guia-rapido-para-usar-em-seus-projetos/>, Accessed: 2024-05-18.
- [22] M. M. Gonçalves, *Arquitetura de Software*, <https://medium.com/sicreditech/arquitetura-de-software-estilos-e-padrões-de-design-50d62d684ef2>, Accessed: 2024-05-23.
- [23] J.-J. Merelo-Guervós, I. Blancas-Álvarez, P. A. Castillo et al., «A comparison of implementations of basic evolutionary algorithm operations in different languages,» em *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 1602–1609. DOI: 10.1109/CEC.2016.7743980.

- [24] L. Prechelt, «An empirical comparison of seven programming languages,» *Computer*, vol. 33, n.º 10, pp. 23–29, 2000. DOI: 10.1109/2.876288.
- [25] D. Jackson e G. Clynch, «An Investigation of the Impact of Language Runtime on the Performance and Cost of Serverless Functions,» em *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, 2018, pp. 154–160. DOI: 10.1109/UCC-Companion.2018.00050.
- [26] A. Huang, «Comparison of programming performance: Promoting STEM and computer science education,» em *2015 IEEE Integrated STEM Education Conference*, 2015, pp. 123–125. DOI: 10.1109/ISECon.2015.7119903.

Apêndice A

Histórias de Usuários

Peso	História de Usuário
5	Como Administrador, eu quero efetuar o login.
5	Como Operacional, eu quero efetuar o login.
5	Como Operacional, eu quero enviar email com os pedidos de cotação para os fornecedores.
5	Como Administrador, eu quero emitir a execução orçamental de cada obra.
4	Como Operacional, eu quero listar os equipamentos da empresa.
4	Como Operacional, eu quero pesquisar/filtrar os equipamentos da empresa.
4	Como Operacional, eu quero inserir novos equipamentos.
4	Como Operacional, eu quero alterar um equipamento registrado.
4	Como Operacional, eu quero remover um equipamento cadastrado.
4	Como Operacional, eu quero listar os materiais da empresa.
4	Como Operacional, eu quero pesquisar/filtrar os materiais da empresa.
4	Como Operacional, eu quero inserir novos materiais.
4	Como Operacional, eu quero alterar um material registrado.
4	Como Operacional, eu quero remover um material cadastrado.
3	Como Operacional, eu quero listar os combustíveis lançados.
3	Como Operacional, eu quero pesquisar/filtrar os combustíveis lançados.
3	Como Operacional, eu quero inserir uma nova fatura de combustível.
3	Como Operacional, eu quero alterar uma fatura de combustível lançada.
3	Como Operacional, eu quero remover uma fatura de combustível lançada.
5	Como Operacional, eu quero listar os clientes registrados.
5	Como Operacional, eu quero pesquisar/filtrar os clientes registrados.
5	Como Operacional, eu quero inserir um novo cliente.

Peso	História de Usuário
5	Como Operacional, eu quero alterar dados de um cliente.
5	Como Operacional, eu quero remover um cliente.
5	Como Operacional, eu quero listar os fornecedores registrados.
5	Como Operacional, eu quero pesquisar/filtrar os fornecedores registrados.
5	Como Operacional, eu quero inserir um novo fornecedor.
5	Como Operacional, eu quero alterar dados de um fornecedor.
5	Como Operacional, eu quero remover um fornecedor.
5	Como Administrador, eu quero listar as obras registrados.
5	Como Administrador, eu quero pesquisar/filtrar as obras registrados.
5	Como Administrador, eu quero inserir um nova obra.
5	Como Administrador, eu quero alterar dados de uma obra.
5	Como Administrador, eu quero remover uma obra.
3	Como Administrador, eu quero listar os orçamentos registrados.
3	Como Administrador, eu quero pesquisar orçamentos registrados.
3	Como Administrador, eu quero importar um orçamento para plataforma.
5	Como Administrador, eu quero listar as presenças registrados.
5	Como Administrador, eu quero inserir uma nova presença.
5	Como Administrador, eu quero alterar dados de uma presença.
2	Como Operacional, eu quero listar as notas de encomendas registrados.
2	Como Operacional, eu quero pesquisar/filtrar as nota de encomenda registrados.
2	Como Operacional, eu quero inserir uma nota de encomenda.
2	Como Operacional, eu quero alterar dados de uma nota de encomenda.
2	Como Operacional, eu quero remover uma nota de encomenda.
5	Como Operacional, eu quero listar as faturas de materiais registradas.
5	Como Operacional, eu quero pesquisar/filtrar as faturas de materiais registradas.
5	Como Operacional, eu quero inserir um nova fatura de materiais.
5	Como Operacional, eu quero alterar dados de uma fatura de materiais.
5	Como Operacional, eu quero remover uma fatura de materiais.
4	Como Operacional, eu quero listar os pedidos de cotação registradas.
4	Como Operacional, eu quero pesquisar/filtrar os pedidos de cotação registrados.
4	Como Operacional, eu quero inserir um nova pedido de cotação.
4	Como Operacional, eu quero alterar dados de um pedido de cotação.
4	Como Operacional, eu quero remover um pedido de cotação.
4	Como Operacional, eu quero listar as notas de crédito de materiais registradas.
4	Como Operacional, eu quero pesquisar/filtrar as notas de crédito de materiais registradas.

Peso	História de Usuário
4	Como Operacional, eu quero inserir uma nova nota de crédito de materiais.
4	Como Operacional, eu quero alterar dados de uma nota de crédito de materiais.
4	Como Operacional, eu quero remover uma nota de crédito de materiais.
5	Como Administrador, eu quero listar as faturas registradas.
5	Como Administrador, eu quero importar SAFT para a plataforma.
3	Como Operacional, eu quero listar os seguros dos automóveis registrados.
3	Como Operacional, eu quero pesquisar/filtrar os seguros dos automóveis registrados.
3	Como Operacional, eu quero inserir um novo seguro de automóvel.
3	Como Operacional, eu quero alterar dados de um seguro de automóvel.
3	Como Operacional, eu quero remover um seguro de automóvel.
3	Como Operacional, eu quero listar os impostos únicos de circulação registrados.
3	Como Operacional, eu quero pesquisar/filtrar os impostos únicos de circulação registrados.
3	Como Operacional, eu quero inserir um novo imposto único de circulação.
3	Como Operacional, eu quero alterar dados de um imposto único de circulação.
3	Como Operacional, eu quero remover um imposto único de circulação.
3	Como Operacional, eu quero listar as inspeções periódicas registradas.
3	Como Operacional, eu quero pesquisar/filtrar as inspeções periódicas registrados.
3	Como Operacional, eu quero inserir uma nova inspeção periódica.
3	Como Operacional, eu quero alterar dados de uma inspeção periódica.
3	Como Operacional, eu quero remover uma inspeção periódica.
3	Como Operacional, eu quero listar os leasing de viaturas registrados.
3	Como Operacional, eu quero pesquisar/filtrar os leasing de viaturas registrados.
3	Como Operacional, eu quero inserir um leasing de viatura.
3	Como Operacional, eu quero alterar dados de um leasing de viatura.
3	Como Operacional, eu quero remover um leasing de viatura.
3	Como Operacional, eu quero listar os leasing de equipamentos registrados.
3	Como Operacional, eu quero pesquisar/filtrar os leasing de equipamentos registrados.
3	Como Operacional, eu quero inserir um leasing de equipamento.
3	Como Operacional, eu quero alterar dados de um leasing de equipamento.
3	Como Operacional, eu quero remover um leasing de equipamento.
3	Como Operacional, eu quero listar os seguros de trabalho registrados.
3	Como Operacional, eu quero pesquisar/filtrar os seguros de trabalho registrados.
3	Como Operacional, eu quero inserir um novo seguro de trabalho.

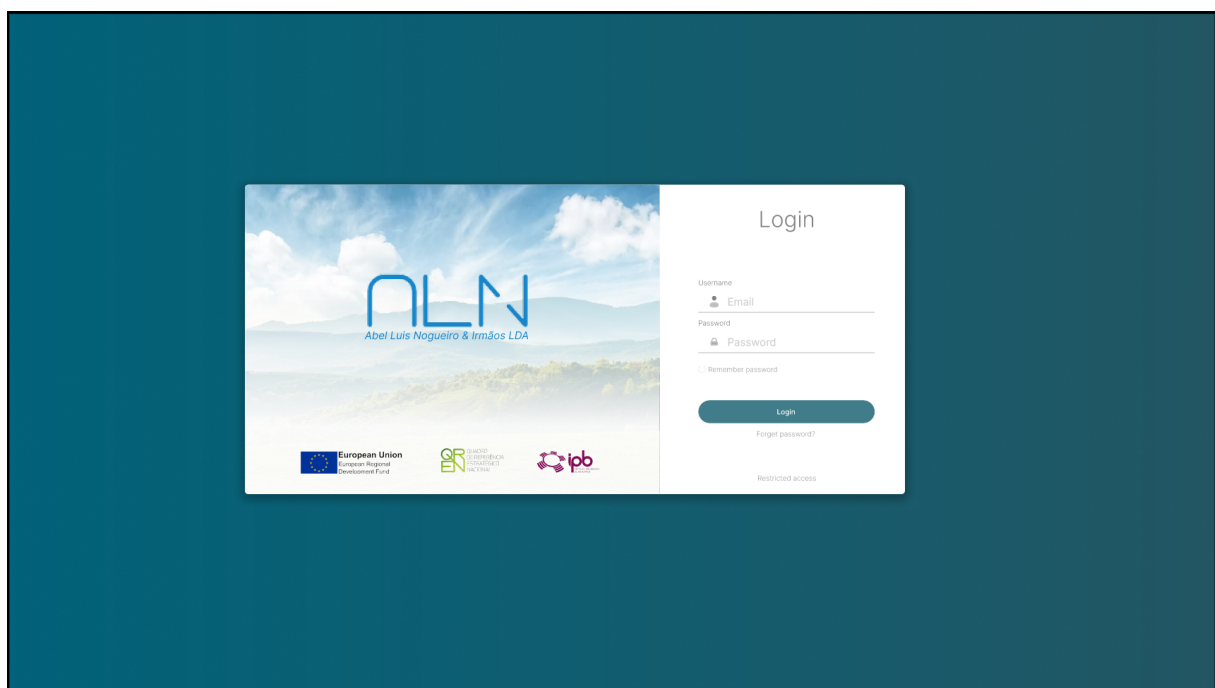
Peso	História de Usuário
3	Como Operacional, eu quero alterar dados de um seguro de trabalho.
3	Como Operacional, eu quero remover um seguro de trabalho.
4	Como Operacional, eu quero listar os consumíveis e serviços registrados.
4	Como Operacional, eu quero pesquisar/filtrar os consumíveis e serviços registrados.
4	Como Operacional, eu quero inserir um novo consumível e serviço.
4	Como Operacional, eu quero alterar dados de um consumível e serviço.
4	Como Operacional, eu quero remover um consumível e serviço.
2	Como Operacional, eu quero listar as amortização de viaturas registradas.
2	Como Operacional, eu quero pesquisar/filtrar as amortização de viaturas registrados.
2	Como Operacional, eu quero inserir uma nova amortização de viatura.
2	Como Operacional, eu quero alterar dados de uma amortização de viatura.
2	Como Operacional, eu quero remover uma amortização de viatura.
2	Como Operacional, eu quero listar os cálculos do IVA registrados.
2	Como Operacional, eu quero pesquisar/filtrar os cálculos do IVA registrados.
2	Como Operacional, eu quero inserir um novo cálculo do IVA.
2	Como Operacional, eu quero alterar dados de um cálculos do IVA.
2	Como Operacional, eu quero remover um cálculo do IVA.
4	Como Operacional, eu quero listar os colaboradores registrados.
4	Como Operacional, eu quero pesquisar/filtrar os colaboradores registrados.
4	Como Operacional, eu quero inserir um novo colaborador.
4	Como Operacional, eu quero alterar dados de um colaborador.
4	Como Operacional, eu quero remover um colaborador.
1	Como Operacional, eu quero listar os descendentes dos colaboradores registrados.
1	Como Operacional, eu quero pesquisar/filtrar os descendentes dos colaboradores registrados.
1	Como Operacional, eu quero inserir um novo descendente de colaborador.
1	Como Operacional, eu quero alterar dados de um descendente de colaborador.
1	Como Operacional, eu quero remover um descendente de colaborador.
5	Como Administrador, eu quero imprimir folha de obra.
5	Como Administrador, eu quero imprimir folha de presença.
5	Como Administrador, eu quero remover a presença de um colaborador.
5	Como Administrador, eu quero emitir a execução orçamental de todas as obras.
5	Como Administrador, eu quero emitir a execução orçamental das obras ativas.
1	Como Administrador, eu quero listar os salários dos colaboradores.
1	Como Administrador, eu quero inserir salário a um colaborador.
1	Como Administrador, eu quero alterar o salário de um colaborador.

Peso História de Usuário

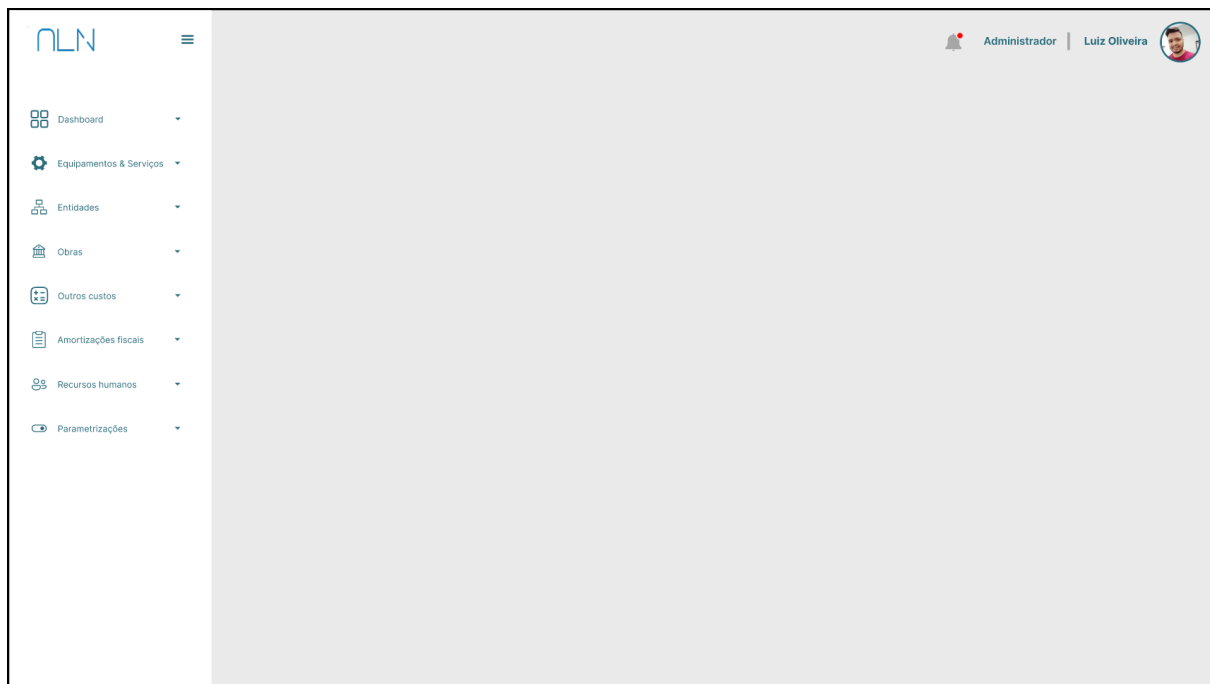
- | | |
|---|---|
| 1 | Como Administrador, eu quero efetuar o cálculo dos vencimentos dos colaboradores. |
| 1 | Como Administrador, eu quero inserir os feriados móveis. |
| 1 | Como Administrador, eu quero alterar os feriados móveis. |
| 1 | Como Administrador, eu quero remover os feriados móveis. |
| 1 | Como Administrador, eu quero listar os feriados móveis e fixos. |
| 1 | Como Administrador, eu quero listar os parâmetros. |
| 1 | Como Administrador, eu quero alterar o valor do iva. |
| 2 | Como Operacional, eu quero listar as amortização de equipamentos registradas. |
| 2 | Como Operacional, eu quero inserir uma amortização de equipamento. |
| 2 | Como Operacional, eu quero alterar uma amortização de equipamento. |
-

Apêndice B

Modelagem da Interface







NLN

Dashboard

Equipamentos & Serv.

Equipamento

Viatura

Material

Tipo

Consumível/Serviços

Entidades

Obras

Outros custos

Amortizações fiscais

Recursos humanos

Parametrizações

Administrador

Luiz Oliveira

Viatura

Search

+ Adicionar

Modelo	Marca	Ano	Tipo	Fornecedor	V. Entrada	V. Residual	Leasing	Ações
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 
Fiesta 1.6 Trail Flex Manual	Ford	2008	Passeio	Auto - Sueco Automóveis S.A	€ 30.000,00	€ 30.000,00	Sim	 

Rows per page: 101-5 of 13

NLN

Dashboard

Equipamentos & Serv.

Equipamento

Viatura

Material

Tipo

Consumível/Serviços

Entidades

Obras

Outros custos

Amortizações fiscais

Recursos humanos

Parametrizações

Administrador

Luiz Oliveira

Viatura

Modelo

Marca

Matrícula

Tipo

Fornecedor

Data Registro

☐ Adquirido em Leasing

Valor Entrada

Valor Residual

Nº de Rendas

Salvar

NLN

Dashboard

Equipamentos & Serv.

Equipamento

Viatura

Material

Tipo

Consumível/Serviços

Entidades

Obras

Outros custos

Amortizações fiscais

Recursos humanos

Parametrizações

Administrador

Luiz Oliveira

Material

Search

+ Adicionar

Name	Referencia	Unidade	Ações
11V25.S RAL90 10 CX Carretel	NS762070	uni	
1428 Trincha 20 Superior	5350025018	uni	
1428 Trincha 40 Superior	004907061	uni	
1428 Trincha 50 Superior	004907063	uni	
204-TR. Junior 1 1/2"	004907063	uni	
205-TR. Junior Pardal 2"	004907064	uni	
205/75R17.5 124/122M TL AVANT A3 SAVA	000	uni	
205/75R17.5 RST REC. Ramoa	001	uni	
206-TR JUNIOR Pardal 2 1/2"	00194000306	uni	

Rows per page: 101-5 of 13

NLN

Dashboard

Equipamentos & Serv.

Equipamento

Viatura

Material

Tipo

Consumível/Serviços

Entidades

Obras

Outros custos

Amortizações fiscais

Recursos humanos

Parametrizações

Administrador

Luiz Oliveira

Material

Search

+ Adicionar

Adicionar Material

Material

Referencia

Unidade

Salvar

Name	Referencia	Unidade	Ações
11V25.S RAL90 10 CX Carretel		uni	
1428 Trincha 20 Superior		uni	
1428 Trincha 40 Superior		uni	
1428 Trincha 50 Superior		uni	
204-TR. Junior 1 1/2"		uni	
205-TR. Junior Pardal 2"		uni	
205/75R17.5 124/122M TL AVANT A3 SAVA		uni	
205/75R17.5 RST REC. Ramoa		uni	
206-TR JUNIOR Pardal 2 1/2"	00194000306	uni	

Rows per page: 101-5 of 13

NLN

Dashboard

Equipamentos & Serv.

Equipamento

Viatura

Material

Tipo

Consumível/Serviços

Entidades

Obras

Outros custos

Amortizações fiscais

Recursos humanos

Parametrizações

Material

Salvar

Adicionar

Atualização de Material

Material

11V25.S RAL90 10 CX Carretel

Referencia

NS762070

Unidade

Unidade

Salvar

Unidade	Ações
uni	
uni	
uni	
uni	
uni	
uni	
uni	
uni	
uni	
uni	

Rows per page: 10 1-5 of 13

NLN

Dashboard

Equipamentos & Serv.

Equipamento

Viatura

Material

Tipo

Consumível/Serviços

Entidades

Obras

Outros custos

Amortizações fiscais

Recursos humanos

Parametrizações

Material

Salvar

Adicionar

Aviso!

Tem certeza que deseja remover este item?

Esta ação é irreversível!

Cancelar

Confirmar

Unidade	Ações
uni	
uni	
uni	
uni	
uni	
uni	
uni	
uni	
uni	
uni	

Rows per page: 10 1-5 of 13

NLN

Dashboard

Equipamentos & Serviços

Entidades

ALN

Clientes

Fornecedores

Obras

Outros custos

Amortizações fiscais

Recursos humanos

Parametrizações

Administrador

Luiz Oliveira

Fornecedores

Search

+ Adicionar

Nome	Telefone	Email	NIF	Localidade	Actions
Fornecedor de materiais S.A	+351 123 456 789	example@example.com	123456789	Bragança	
Fornecedor de materiais S.A	+351 123 456 789	example@example.com	123456789	Bragança	
Fornecedor de materiais S.A	+351 123 456 789	example@example.com	123456789	Bragança	
Fornecedor de materiais S.A	+351 123 456 789	example@example.com	123456789	Bragança	
Fornecedor de materiais S.A	+351 123 456 789	example@example.com	123456789	Bragança	
Fornecedor de materiais S.A	+351 123 456 789	example@example.com	123456789	Bragança	
Fornecedor de materiais S.A	+351 123 456 789	example@example.com	123456789	Bragança	
Fornecedor de materiais S.A	+351 123 456 789	example@example.com	123456789	Bragança	
Fornecedor de materiais S.A	+351 123 456 789	example@example.com	123456789	Bragança	

Rows per page: 101-5 of 13

NLN

Dashboard

Equipamentos & Serviços

Entidades

ALN

Clientes

Fornecedores

Obras

Outros custos

Amortizações fiscais

Recursos humanos

Parametrizações

Administrador

Luiz Oliveira

Fornecedores

Search

+ Adicionar

Adicionar Fornecedor

Nome

Telefone

NIF

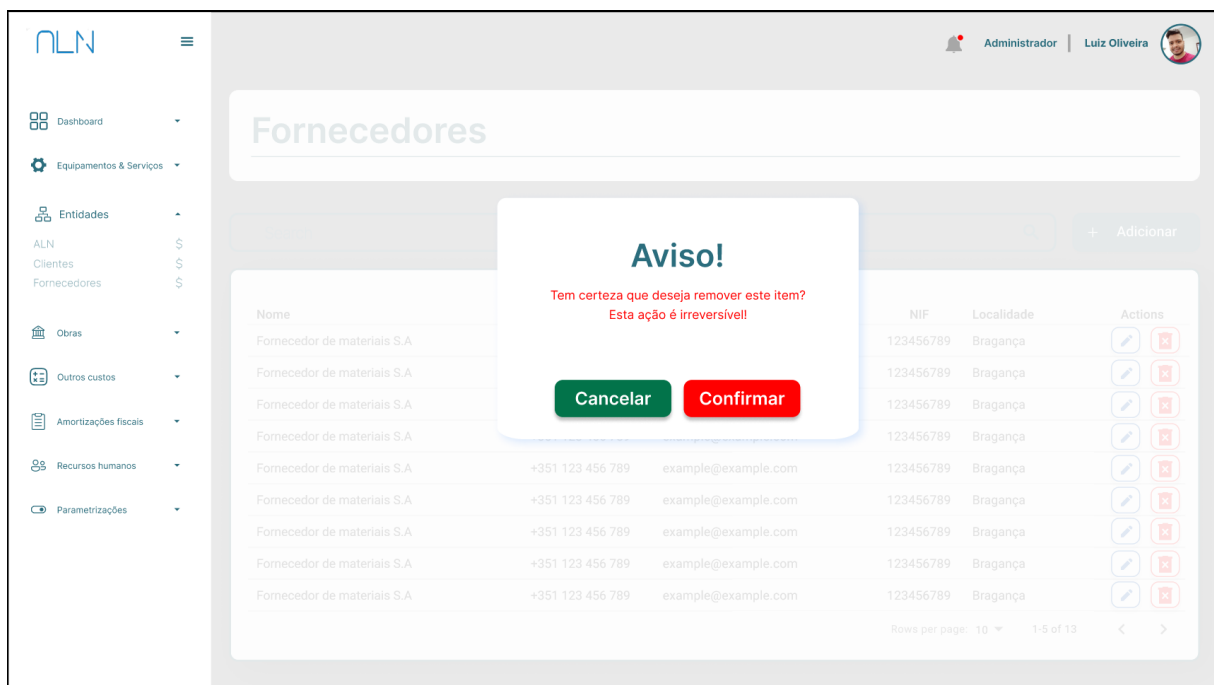
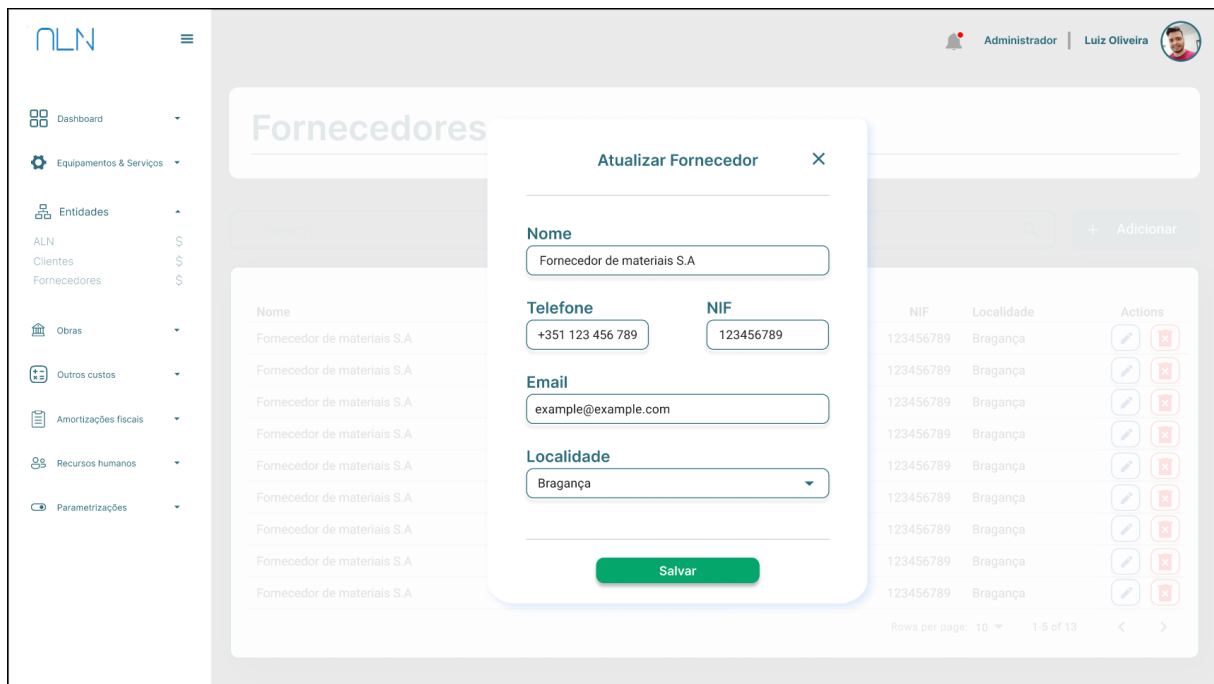
Email

Localidade

Salvar

Nome	Telefone	NIF	Localidade	Actions
Fornecedor de materiais S.A				
Fornecedor de materiais S.A				
Fornecedor de materiais S.A				
Fornecedor de materiais S.A				
Fornecedor de materiais S.A				
Fornecedor de materiais S.A				
Fornecedor de materiais S.A				
Fornecedor de materiais S.A				
Fornecedor de materiais S.A				

Rows per page: 101-5 of 13



Apêndice C

Documentação - Swagger

Documentação da API REST extraída do Swagger.

Auth:

- POST /Auth/Login

Cliente:

- GET /Cliente/GetListName
- GET /Cliente/GetByName
- GET /Cliente/GetByTaxNumber
- GET /Cliente/GetById/{idCliente}

Colaborador:

- GET /Colaborador/GetListName
- GET /Colaborador/GetByName

Consumivel:

- GET /Consumivel/GetByName
- GET /Consumivel/ListarConsumiveis
- POST /Consumivel/PostConsumivel
- GET /Consumivel/GetByIdToUpdate/{idConsumivel}

- PUT /Consumivel/PutConsumivel/{idConsumivel}
- DELETE /Consumivel/DeleteConsumivel/{idConsumivel}

ConsumivelServicos:

- GET /ConsumivelServicos/ListarConsumivelServicos
- GET /ConsumivelServicos/GetByIdToUpdate/{idconsumivelServicos}
- POST /ConsumivelServicos/PostConsumivelServicos
- PUT /ConsumivelServicos/PutConsumivelServicos
/{idconsumivelServicos}
- DELETE /ConsumivelServicos/DeleteConsumivelServicos
/{idconsumivelServicos}

Ctt:

- GET /Ctt/GetAddresses/{postalCode}

Equipamento:

- GET /Equipamento/GetByName

FacturaCliente:

- GET /FacturaCliente/ListarFaturaCliente
- GET /FacturaCliente/GetByIdToUpdate/{idFatura}
- POST /FacturaCliente/PostFaturaCliente
- PUT /FacturaCliente/PutFaturaCliente/{idFatura}
- POST /FacturaCliente/UploadFileFaturaCliente
- DELETE /FacturaCliente/DeleteFaturaCliente/{idFatura}

FacturaObra:

- GET /FacturaObra/GetByIdToUpdate/{idFatura}
- GET /FacturaObra/ListarFaturaObra

- POST /FacturaObra/PostFaturaObra
- PUT /FacturaObra/PutFaturaObra/{idFatura}
- POST /FacturaObra/UploadFileFaturaObra
- DELETE /FacturaObra/DeleteFaturaObra/{idFatura}

Fornecedor:

- GET /Fornecedor/GetListName
- GET /Fornecedor/GetByName

IUC:

- GET /IUC/ListarIUC
- GET /IUC/GetByIdToUpdate/{idIuc}
- POST /IUC/PostIUC
- PUT /IUC/PutIUC/{idIuc}
- DELETE /IUC/DeletarIUC/{idIuc}

LeasingEquipamento:

- GET /LeasingEquipamento/ListarLeasingEquipamento
- GET /LeasingEquipamento/GetByIdToUpdate/{IdLeasingEquipamento}
- POST /LeasingEquipamento/PostLeasingEquipamento
- PUT /LeasingEquipamento/PutLeasingEquipamento
/{IdLeasingEquipamento}
- DELETE /LeasingEquipamento/DeletarLeasingEquipamento
/{IdLeasingEquipamento}

LeasingViatura:

- GET /LeasingViatura/ListarLeasingViatura
- GET /LeasingViatura/GetByIdToUpdate/{IdLeasingViatura}

- POST /LeasingViatura/PostLeasingViatura
- PUT /LeasingViatura/PutLeasingViatura/{IdLeasingViatura}
- DELETE /LeasingViatura/DeletarLeasingViatura/{IdLeasingViatura}

Material:

- GET /Material/GetListName
- GET /Material/GetByName
- GET /Material/ListarPrecoMateriais
- GET /Material/ListarMateriaisPorObra
- GET /Material/ListarMateriais
- POST /Material/PostMaterial
- GET /Material/GetByIdToUpdate/{idMaterial}
- PUT /Material/PutMaterial/{idMaterial}
- DELETE /Material/DeleteMaterial/{idMaterial}

Obra:

- GET /Obra/GetByName
- GET /Obra/GetListName
- GET /Obra/GetListOptionName
- GET /Obra/GetExecucaoOrcamental
- GET /Obra/ListarObra
- POST /Obra/PostObra
- GET /Obra/GetByIdToUpdate/{IdObra}
- GET /Obra/GetByTaxNumber
- GET /Obra/GetByIdCliente
- GET /Obra/GetById

- PUT /Obra/PutObra/{IdObra}
- GET /Obra/ProximoNumeroObra

Parametros:

- GET /Parametros/GetListName

PedidoCotacao:

- GET /PedidoCotacao/ListarPedidoCotacao
- GET /PedidoCotacao/GetByIdToUpdate/{idPedido}
- GET /PedidoCotacao/GetNewCodePedidoCotacao
- POST /PedidoCotacao/PostPedidoCotacao
- PUT /PedidoCotacao/PutPedidoCotacao/{idPedido}
- POST /PedidoCotacao/UploadFilePedidoCotacao

PresencaObra:

- GET /PresencaObra/GetAllPresence
- POST /PresencaObra/PostPresence

SeguroAutomovel:

- GET /SeguroAutomovel/ListarSeguroAutomovel
- GET /SeguroAutomovel/GetByIdToUpdate/{idSeguroAutomovel}
- POST /SeguroAutomovel/PostSeguroAutomovel
- PUT /SeguroAutomovel/PutSeguroAutomovel/{idSeguroAutomovel}
- DELETE /SeguroAutomovel/DeletarSeguroAutomovel/{idSeguroAutomovel}

SeguroTrabalho:

- GET /SeguroTrabalho/ListarSeguroTrabalho
- GET /SeguroTrabalho/GetByIdToUpdate/{idSeguroTrabalho}

- POST /SeguroTrabalho/PostSeguroTrabalho
- PUT /SeguroTrabalho/PutSeguroTrabalho/{idSeguroTrabalho}
- DELETE /SeguroTrabalho/DeletarSeguroTrabalho/{idSeguroTrabalho}

TipoConsumivel:

- GET /TipoConsumivel/GetListName

Unidade:

- GET /Unidade/GetListName
- GET /Unidade/GetByName

Utilizador:

- GET /Utilizador/GetUtilizador

Viatura:

- GET /Viatura/GetByName

ViaturaInspecao:

- GET /ViaturaInspecao/ListarViaturaInspecao
- GET /ViaturaInspecao/GetByIdToUpdate/{IdViaturaInspecao}
- POST /ViaturaInspecao/PostViaturaInspecao
- PUT /ViaturaInspecao/PutViaturaInspecao/{IdViaturaInspecao}
- DELETE /ViaturaInspecao/DeletarViaturaInspecao
/{IdViaturaInspecao}

Apêndice D

Paginas da aplicação web

Figura D.1: Página - Equipamentos e Serviços / Materiais

OLN

Administrador

Luiz Henrique

Equipamentos & Serviços

Material

Consumíveis/Serviços

Obras

Outros custos

Pesquisar...		+ Adicionar	Editar	Remover
☐	Consumível/Serviço	Unidade	Tipo	
☐		EUR	Empresa	
☐		UNI	Empresa	
☐		UNI	Empresa	
☐		EUR	Empresa	
☐		UNI	Empresa	
☐		UNI	Empresa	
☐		UNI	Empresa	
☐		UNI	Empresa	
☐		UNI	Empresa	
☐		UNI	Empresa	
☐		UNI	Empresa	
☐		CENT.	Empresa	
☐		EUR	Obra	
☐		EUR	Obra	
☐		EUR	Obra	
☐		EUR	Obra	

<

1

2

>

50 / página

[illegible]

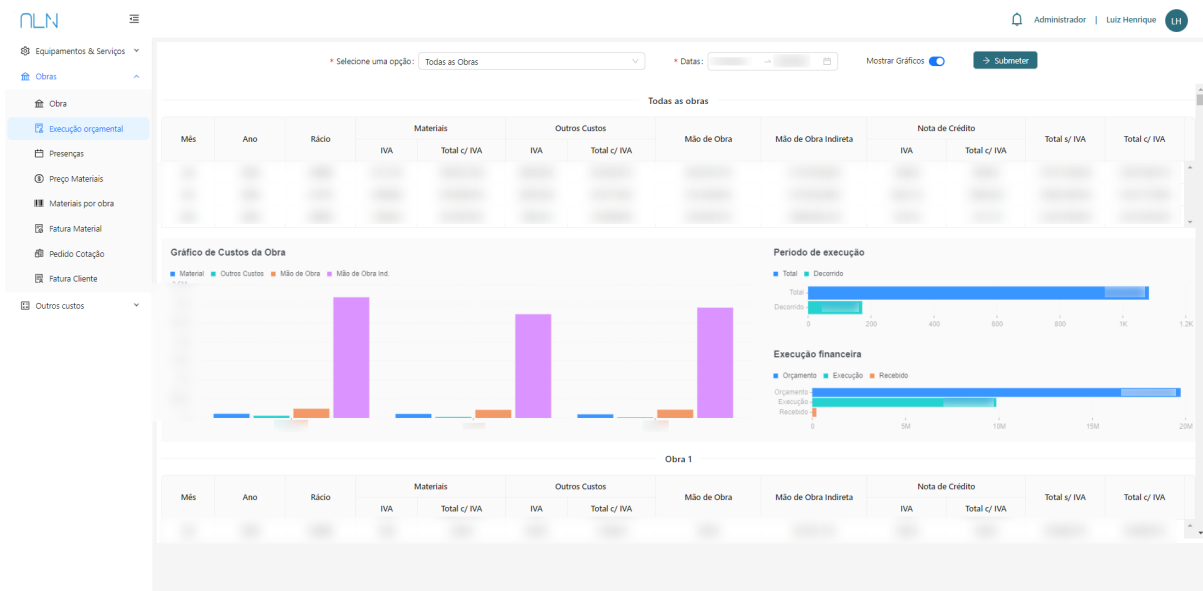


Figura D.4: Página - Obra / Execução Orçamental

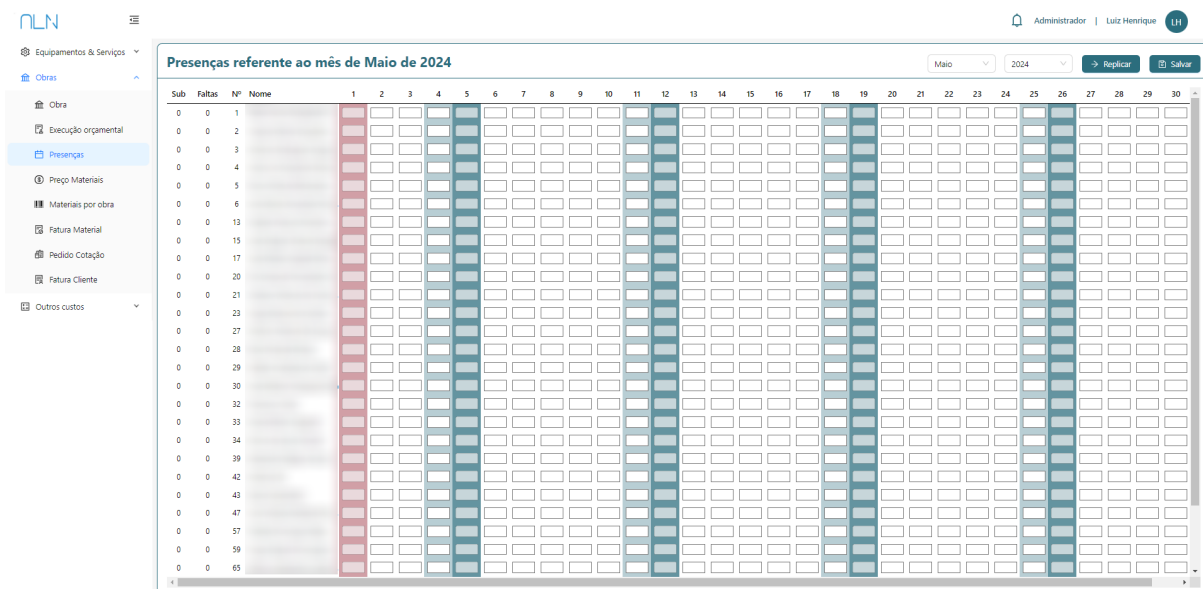


Figura D.5: Página - Obra / Presenças

[illegible][illegible]

[illegible]

Figura D.8: Página - Obra / Fatura Material

OLN
Administrador | Luiz Henrique

Equipamentos & Serviços

- Obras
- Obra
- Eexecução orçamental
- Presenças
- Preço Materiais
- Materiais por obra
- Fatura Material
- Pedido Cotação
- Fatura Cliente
- Outros custos

Nº Pedido	Data Pedido	Data Cotação	Obra	Fornecedores	Itens	Total S/IVA	IVA	Total C/IVA	Desc. \$	Valor Pagar

< 1 >

Figura D.9: Página - Obra / Pedido Cotação

OLN

Administrador | Luiz Henrique

Equipamentos & Serviços

Obras

	Nº Fatura	Data Fatura	Obra	Cliente	Total S/IVA	IVA	Total C/IVA
☐	FAC 1/17	30/06/2014					
☐	NC 1/2	23/06/2014					
☐	FAC 1/16	20/06/2014					
☐	FAC 1/15	17/06/2014					
☐	FAC 1/10	08/04/2014					
☐	FAC 1/11	08/04/2014					
☐	FAC 1/8	31/03/2014					
☐	FAC 1/9	31/03/2014					
☐	FAC 1/7	24/03/2014					
☐	FAC 1/6	13/03/2014					
☐	FAC 1/5	12/03/2014					
☐	NC 1/1	12/03/2014					
☐	FAC 1/4	06/03/2014					
☐	FAC 1/2	24/02/2014					
☐	FAC 1/3	24/02/2014					

1

[illegible][illegible]

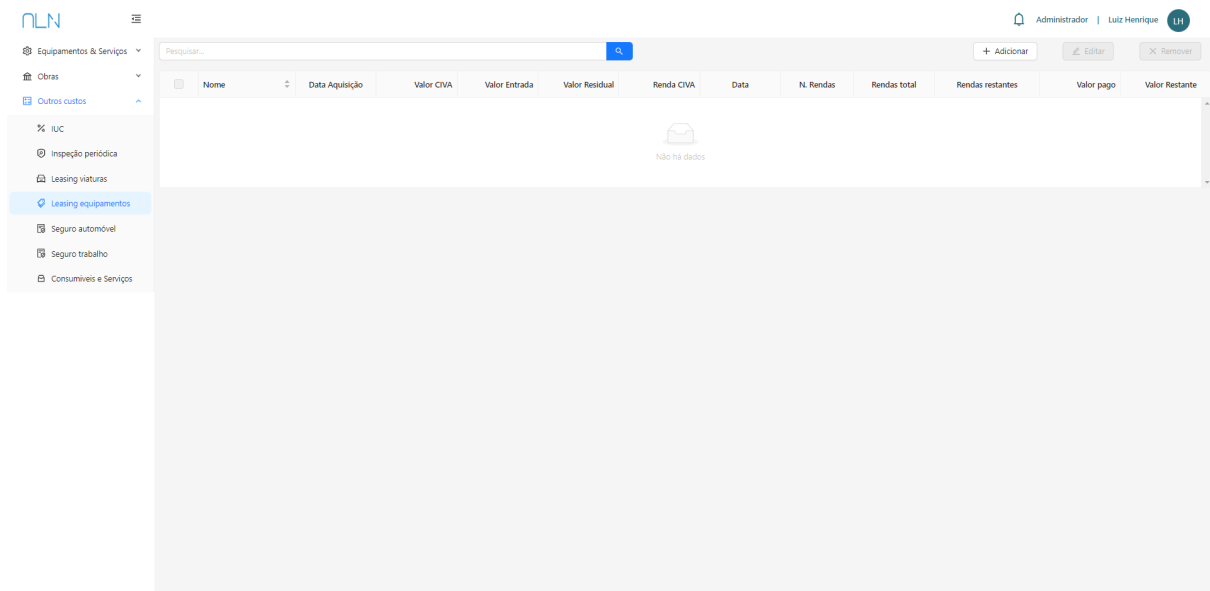


Figura D.14: Página - Outros custos / Leasing Equipamento

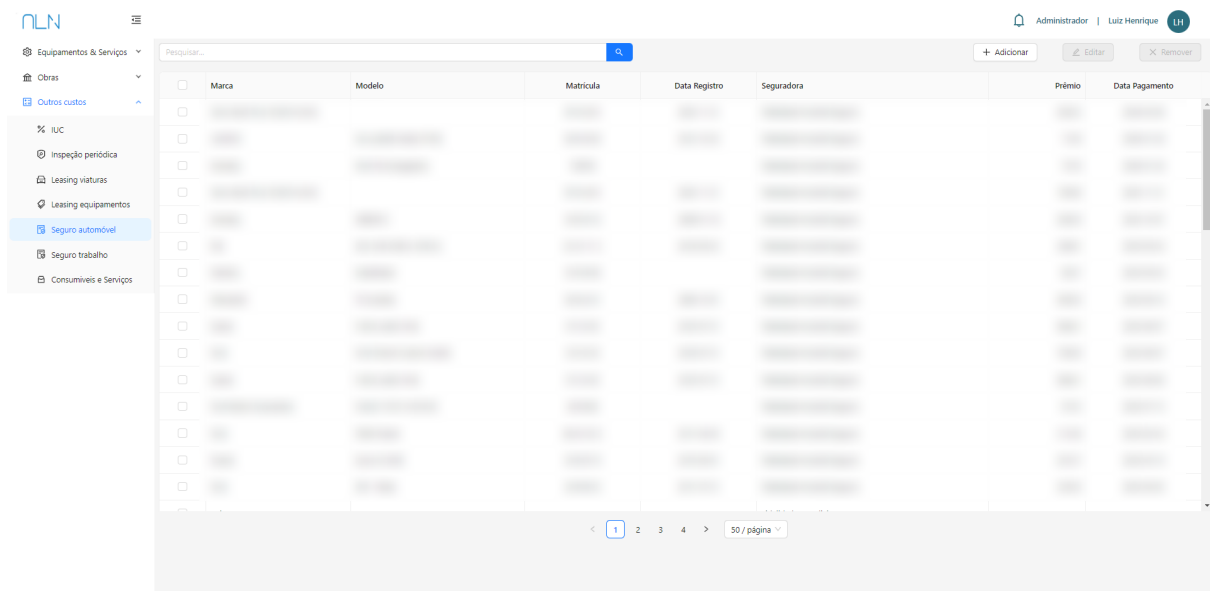


Figura D.15: Página - Outros custos / Seguro Automóvel

[illegible][illegible]