

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

PUC Minas Virtual

Pós-graduação *Lato Sensu* em Engenharia de *Software*

Projeto Integrado

Relatório Técnico

EntregaSegura

Danilo de Carvalho Silva
Jamil Tadeu Zazu

Belo Horizonte
Outubro, 2023.

SUMÁRIO

1.	Cronograma de Trabalho	4
2.	Introdução	5
3.	Definição Conceitual da Solução	7
3.1.	Diagrama de Casos de Uso	7
3.2.	Requisitos Funcionais	8
3.3.	Requisitos Não-Funcionais	10
4.	Protótipo Navegável do Sistema	11
5.	Diagrama de Classes de Domínio	12
6.	Arquitetura da Solução	13
6.1.	Padrão Arquitetural	13
6.2.	C4 model - Diagrama de Contexto	14
7.	Frameworks de Trabalho	15
8.	Estrutura Base do Front End	16
9.	Modelo Relacional ou Projeto de Banco de Dados NoSQL	21
10.	Plano de Testes	22
11.	Apropriação de Horas no Projeto	25
12.	Código da Aplicação	26
13.	Avaliação Retrospectiva	27
13.1	Objetivos Estimados	27
13.2	Objetivos Alcançados	27
13.3	Lições aprendidas	28
14.	Referências	29

1. Cronograma de Trabalho

Datas		Atividade / Tarefa	Produto / Resultado
De	Até		
01/03/2023	04/03/2023	1. Contextualização e objetivos do trabalho	Escopo do projeto definido
07/03/2023	13/03/2023	2. Elaboração da proposta de projeto	Proposta do projeto
16/03/2023	23/03/2023	3. Levantamento de requisitos funcionais	Lista de requisitos funcionais
27/03/2023	02/04/2023	4. Levantamento de requisitos não-funcionais	Lista de requisitos não-funcionais
04/04/2023	10/04/2023	5. Documentação dos Requisitos Levantados	Documentação dos requisitos
12/04/2023	18/04/2023	6. Elaboração do Diagrama de Casos de Uso	Diagrama de casos de uso
20/04/2023	26/04/2023	7. Desenvolvimento do Diagrama de Classes de Domínio	Diagrama de classes de domínio
28/04/2023	04/05/2023	8. Definição do Padrão Arquitetural	Padrão arquitetural definido
07/05/2023	13/05/2023	9. Estrutura de Banco de Dados	Estrutura de banco de dados definida
15/05/2023	15/07/2023	10. Desenvolvimento do sistema	Sistema desenvolvido
18/07/2023	24/07/2023	11. Elaboração do Plano de Testes	Plano de testes
27/07/2023	02/08/2023	12. Revisão da documentação do Projeto	Documentação revisada
05/08/2023	10/08/2023	13. Criação do repositório no GitHub e deploy do protótipo	Repositório no GitHub e protótipo publicado
20/08/2023	26/08/2023	14. Revisão da Apropriação de Horas	Registro de horas apropriadas revisto
01/09/2023	17/09/2023	15. Revisão de Formatação do Relatório Técnico	Relatório Técnico de acordo com as normas
20/09/2023	30/09/2023	16. Avaliação retrospectiva	Retrospectiva do projeto
03/10/2023	03/10/2023	17. Entrega final	Projeto entregue

2. Introdução

A era digital tem sido marcada por um notável crescimento no comércio eletrônico no Brasil, impulsionando as vendas online a níveis impressionantes. Em 2022, segundo a Abranet, as vendas online alcançaram a marca de R\$ 169,6 bilhões, registrando um crescimento de 5% em relação ao ano anterior. Esse panorama é reflexo da crescente demanda por compras realizadas via dispositivos móveis, que já representam 55% de todas as vendas online. Nesse contexto, a gestão de entregas residenciais em condomínios emerge como um desafio específico que precisa ser enfrentado prontamente.

O cerne deste projeto consiste na resolução de uma dificuldade latente: o gerenciamento seguro e eficaz das entregas em condomínios. Esse desafio envolve assegurar que as entregas alcancem os destinatários corretos, garantir o controle preciso da entrada e saída de pacotes e promover uma comunicação eficiente entre os funcionários do condomínio e os moradores. A "dor do cliente" reside na necessidade de aprimorar esse processo, tornando-o mais eficiente, confiável e seguro, considerando o aumento exponencial da demanda por compras online e o contínuo crescimento do comércio eletrônico.

A motivação deste projeto é oferecer uma solução para um problema que cresce de forma incessante no mercado. Com a implementação do sistema *EntregaSegura*, almeja-se obter uma economia significativa de tempo tanto para os funcionários quanto para os moradores, além de aprimorar a segurança no processo de entrega de encomendas. Essa solução trará benefícios relevantes como: a redução de atrasos nas entregas, a otimização de recursos e a maior satisfação dos residentes, tornando-se uma inovação relevante no contexto atual de crescimento do comércio eletrônico e da crescente demanda por entregas em condomínios.

O objetivo geral deste trabalho é apresentar a descrição detalhada do projeto do sistema *EntregaSegura*, desenvolvido com o propósito de otimizar o gerenciamento de entregas em condomínios.

Objetivos específicos são:

- Descrever minuciosamente os requisitos da aplicação *EntregaSegura*;
- Elucidar como o sistema atua na solução do problema identificado, apresentando suas funcionalidades e vantagens;

- Demonstrar a relevância e aplicabilidade do *EntregaSegura* no atual contexto de crescimento do comércio eletrônico e da demanda por entregas em condomínios.

Ao longo deste trabalho, cada um desses objetivos será explorado em detalhes, evidenciando o valor que o sistema *EntregaSegura* pode proporcionar aos condomínios residenciais no atual cenário da era digital. Com esse projeto, busca-se estabelecer uma base sólida para aprimorar a eficiência e a segurança do processo de entregas em condomínios, beneficiando tanto os prestadores de serviço quanto os moradores.

3. Definição Conceitual da Solução

3.1. Diagrama de Casos de Uso

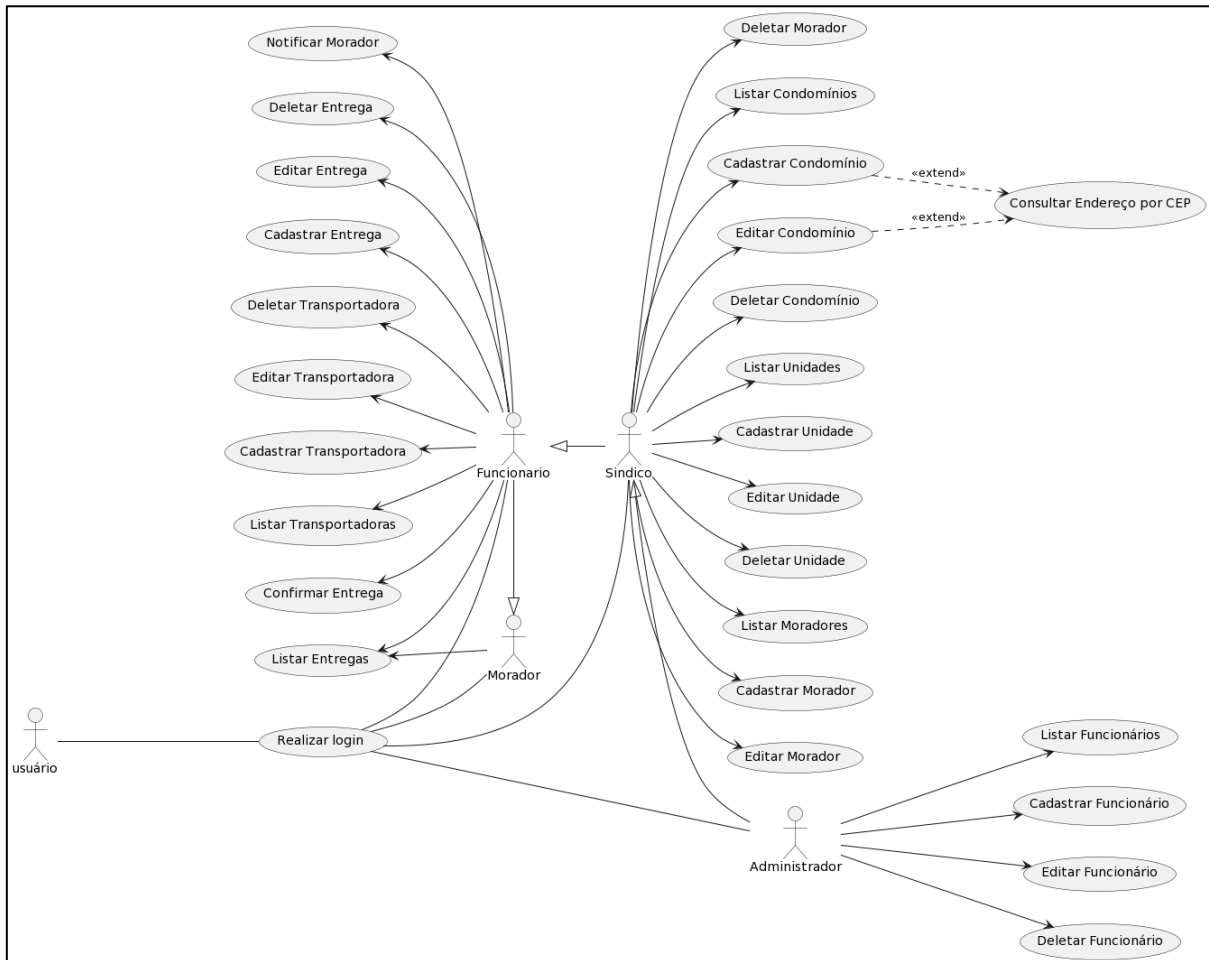


Figura 1: Casos de Uso

Fonte: Os Autores

O arquivo original pode ser encontrado em: [Casos de Uso](#)

3.2. Requisitos Funcionais

ID	Descrição Resumida	Dificuldade (B/M/A)*	Prioridade (B/M/A)*
RF01	O administrador deve poder listar os funcionários cadastrados no sistema.	B	A
RF02	O administrador deve poder cadastrar novos funcionários no sistema.	B	A
RF03	O administrador deve poder editar as informações dos funcionários cadastrados.	M	A
RF04	O administrador deve poder excluir funcionários do sistema.	M	A
RF05	O administrador deve poder listar os condomínios registrados.	B	B
RF06	O administrador deve poder cadastrar novos condomínios.	M	A
RF07	O administrador deve poder editar informações dos condomínios.	M	B
RF08	O administrador deve poder excluir condomínios do sistema.	M	B
RF09	O administrador deve poder listar as unidades registradas no sistema.	B	B
RF10	O administrador deve poder cadastrar novas unidades.	M	B
RF11	O administrador deve poder editar informações das unidades cadastradas.	M	B
RF12	O administrador deve poder excluir unidades do sistema.	M	B
RF13	O administrador deve poder consultar o endereço através do CEP ao cadastrar ou editar condomínios.	A	A
RF14	O administrador deve poder listar os moradores cadastrados.	B	B
RF15	O administrador deve poder cadastrar novos moradores.	M	B
RF16	O administrador deve poder editar informações dos moradores cadastrados.	M	B
RF17	O administrador deve poder excluir moradores do sistema.	M	B
RF18	O administrador deve poder carregar o dashboard do sistema.	B	A
RF19	O síndico deve poder listar os condomínios registrados.	B	B
RF20	O síndico deve poder cadastrar novos condomínios.	M	B

RF21	O síndico deve poder editar informações dos condomínios.	M	B
RF22	O síndico deve poder excluir condomínios do sistema.	M	B
RF23	O síndico deve poder listar as unidades registradas no sistema.	B	B
RF24	O síndico deve poder cadastrar novas unidades.	M	B
RF25	O síndico deve poder editar informações das unidades cadastradas.	M	B

* B = Baixa, M = Média, A = Alta.

3.3. Requisitos Não-Funcionais

ID	Descrição	Prioridade B/M/A
RNF01	O sistema deve apresentar um tempo de resposta abaixo de 200 ms para 95% das consultas.	A
RNF02	A interface do sistema deve ser responsiva e adaptar-se a diferentes tamanhos de tela.	M
RNF03	O sistema deve ser compatível com os navegadores mais populares, como Chrome, Firefox e Safari.	M
RNF04	A segurança do sistema deve ser robusta, protegendo os dados dos usuários contra acesso não autorizado.	A
RNF05	O sistema deve ser capaz de lidar com um grande volume de usuários simultâneos sem degradação de desempenho.	A
RNF06	A disponibilidade do sistema deve ser de, no mínimo, 99,9%, permitindo um tempo de inatividade programado mínimo.	A
RNF07	O sistema deve ser facilmente escalável, permitindo a adição de recursos conforme necessário para suportar o crescimento do número de usuários.	M
RNF08	O sistema deve ser intuitivo e fácil de usar, exigindo um treinamento mínimo para os usuários aprenderem a utilizá-lo.	M
RNF09	O sistema deve fornecer mensagens de erro claras e úteis para ajudar os usuários a corrigirem problemas que possam surgir durante a utilização.	M
RNF10	O sistema deve ser atualizável, permitindo a adição de novos recursos ou a correção de bugs de forma fácil e rápida.	M
RNF11	O sistema deve ter uma aparência visual atrativa e profissional, promovendo uma experiência de usuário positiva.	B

* B = Baixa, M = Média, A = Alta.

4. Protótipo Navegável do Sistema

- Link do Protótipo Navegável e interativo: [Protótipo navegável e interativo](#)
- Link do Vídeo de apresentação: [Vídeo de apresentação](#)
- Link do Design System no Figma: [Design System](#)

5. Diagrama de Classes de Domínio

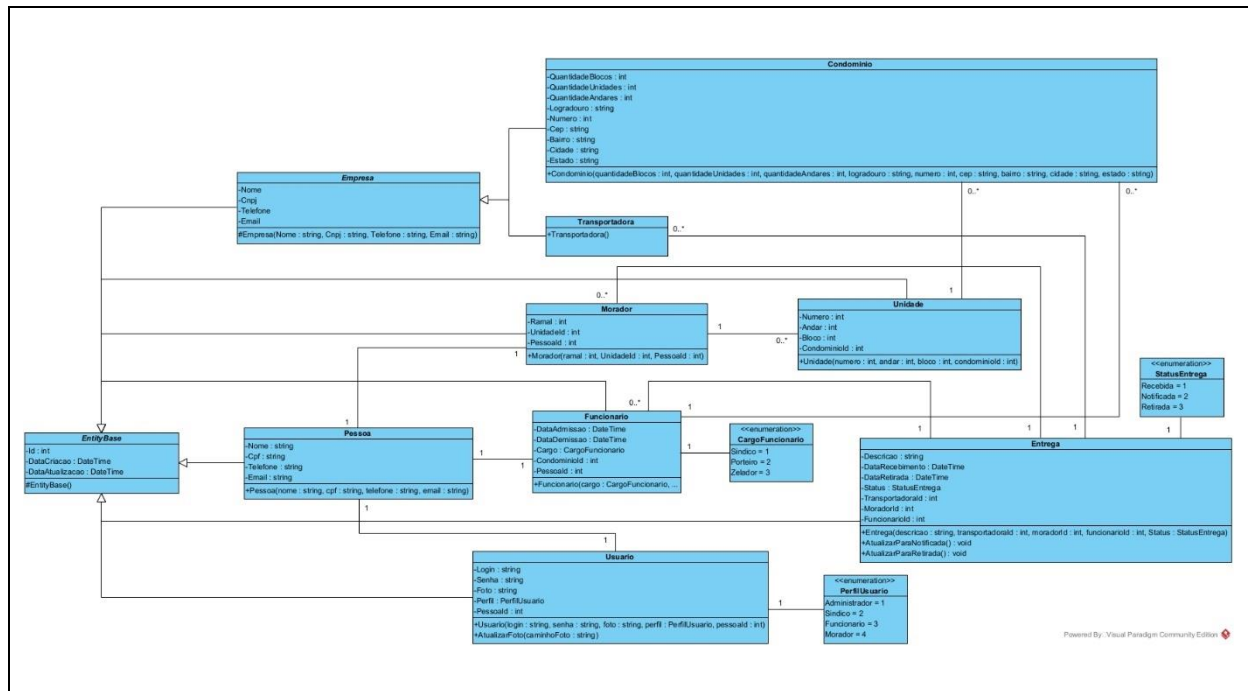


Figura 2: Diagrama de Classes de Domínio

Fonte: Os Autores

O arquivo original pode ser encontrado em: [Diagrama de Classes de Domínio](#)

6. Arquitetura da Solução

6.1. Padrão Arquitetural

Ao analisar os requisitos funcionais e não funcionais do sistema, optamos pelo padrão arquitetural **Arquitetura em Camadas**, com ênfase na Clean Architecture. Essa combinação promove uma separação clara e intencional das responsabilidades, tornando o código mais limpo, modular e testável.

O frontend foi desenvolvido em Angular 14, seguindo as melhores práticas de Single Page Application (SPA). Nessa camada, utilizamos o padrão MVVM (Model-View-ViewModel) para separar de forma efetiva a lógica da interface dos dados. A integração de bibliotecas como Ngx-mask, Ngx-spinner, Ngx-toastr, Bootstrap 5 e Angular Material proporciona uma interface de usuário rica e responsiva.

No que diz respeito ao backend, temos o projeto **EntregaSegura.Application** que atua como intermediário entre a camada de apresentação e a camada de domínio. Aqui, são gerenciados os DTOs, notificações, interfaces e lógica de negócio de alto nível. A integração com o AutoMapper permite um mapeamento eficiente entre entidades e DTOs.

A essência do sistema está no **EntregaSegura.Domain**. Nesse núcleo, encontramos as entidades, os modelos de domínio, as validações e as interfaces de repositório, garantindo autonomia sem depender de camadas externas. Já o **EntregaSegura.Infra.Data** cuida da persistência e do acesso aos dados, implementando as interfaces do domínio e utilizando o padrão Repository para abstrair o acesso ao banco de dados. A tecnologia subjacente é o Entity Framework Core, que facilita o mapeamento objeto-relacional e a gestão da conexão com o SQL Server.

EntregaSegura.API desempenha um papel vital como interface no padrão MVC (Model-View-Controller) para o backend. Ele recebe requisições do frontend e as direciona para as camadas apropriadas. Além disso, a gestão de autenticação e usuários é efetivamente tratada pelo ASP.NET Identity. O Fluent Validation é utilizado para garantir a aplicação rigorosa das regras de negócio, enquanto o **EntregaSegura.Infra.IoC** foi implementado para gerenciar a inversão de controle e a injeção de dependência, aproveitando a Injeção de Dependência Nativa do .NET.

Para armazenar os dados, a escolha do SQL Server como solução de banco de dados foi motivada por sua robustez, segurança e desempenho avançado. Em conclusão, a arquitetura multicamadas adotada, juntamente com os princípios da Clean Architecture, garante a escalabilidade, modularidade e facilidade de manutenção do sistema, preparando-o para futuras extensões e melhorias com um impacto mínimo nos componentes existentes.

6.2. C4 model - Diagrama de Contexto

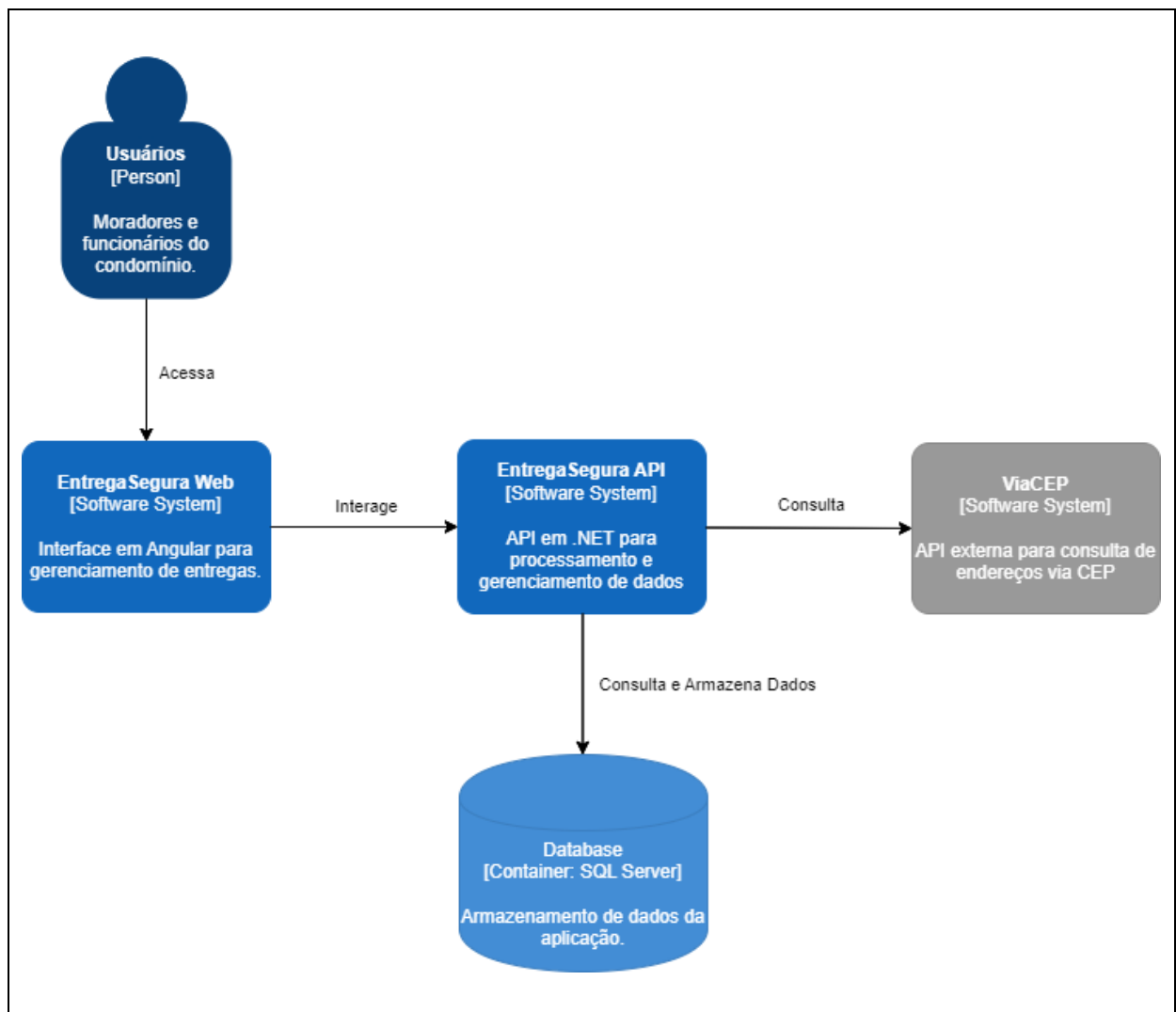


Figura 3: Arquitetura da solução (C4 model)

Fonte: Os Autores

O arquivo original pode ser encontrado em: [Arquitetura da solução \(C4 model\)](#)

7. Frameworks de Trabalho

O projeto "EntregaSegura" adotou uma variedade de frameworks e tecnologias, garantindo um desenvolvimento robusto e eficiente.

Front-End (EntregaSegura Web):

- Desenvolvido em **Angular 14**.
- Estilização e interface com **Angular Material**.
- Melhoria na experiência do usuário e interações aprimoradas com **Ngx-mask**, **Ngx-spinner** e **Ngx-toastr**.
- Autenticação gerenciada com **@auth0/angular-jwt**.
- Otimização de recursos e reatividade com **RxJS**.

Back-End (EntregaSegura API):

- Desenvolvido em **.NET 6**.
- Autenticação utilizando **JwtBearer**.
- Gestão de identidades com **Identity**.
- Manuseio de dados com **Entity Framework Core**.
- Validadores e regras de negócio com **Fluent Validation**.
- Testes unitários com **FluentAssertions** e **XUnit**.
- Mapeamento com **AutoMapper**.
- Criptografia baseado no algoritmo **Rfc2898DeriveBytes** para garantir a segurança das senhas dos usuários.

Persistência de Dados:

- **SQL Server Lite** é utilizado como a base de dados principal, por ser uma solução leve e eficiente para aplicações de menor escala, garantindo a integridade e confiabilidade dos dados sem a sobrecarga de sistemas de gerenciamento de banco de dados mais complexos.
- As consultas de endereço são gerenciadas através da integração com uma API externa chamada ViaCEP.

8. Estrutura Base do Front End

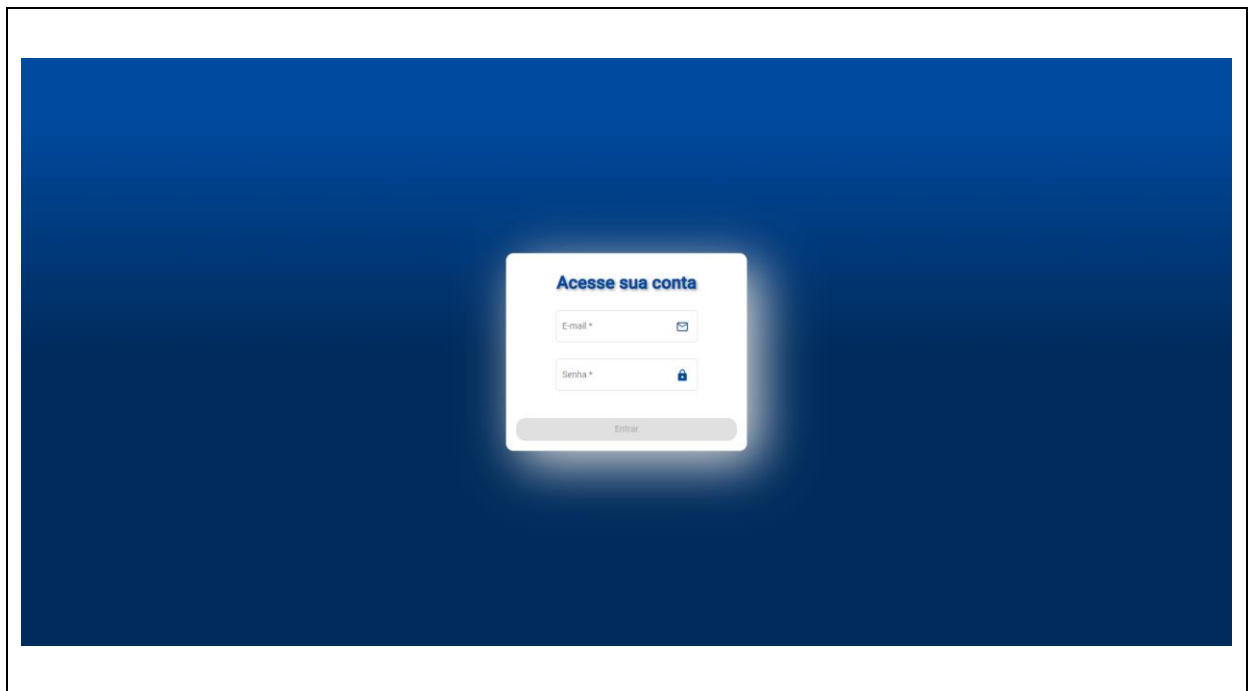


Figura 4: Página de Login

Fonte: Os Autores

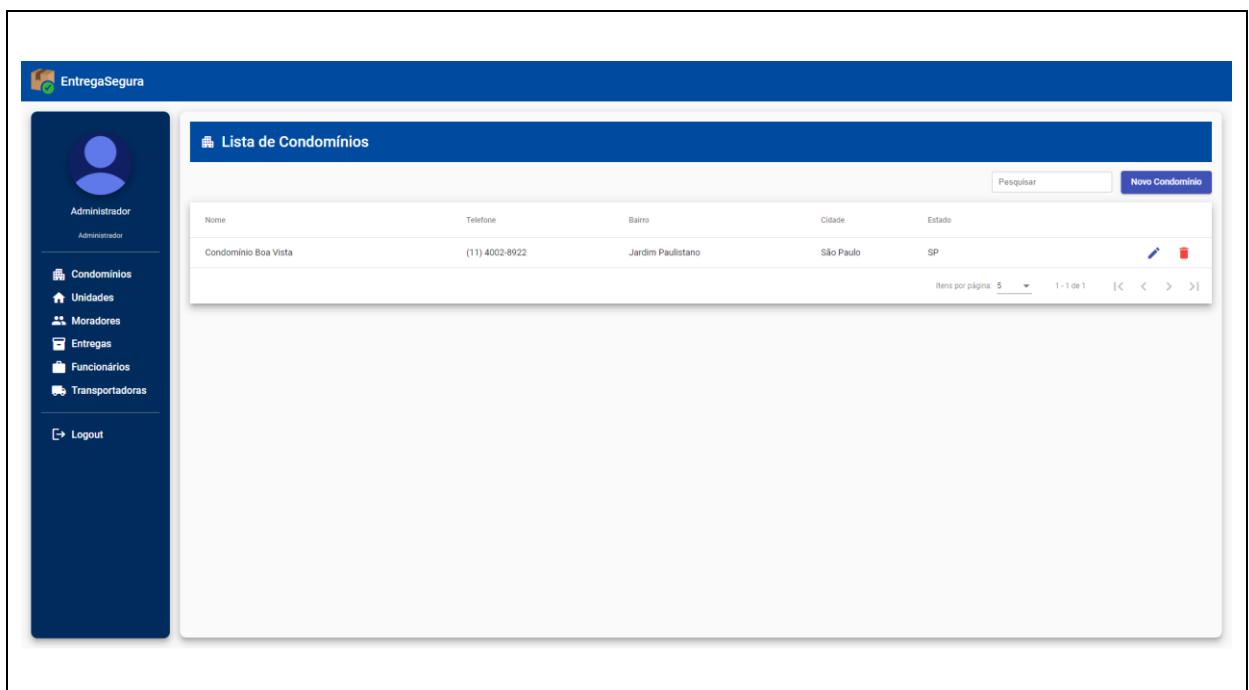


Figura 5: Lista de Condomínios

Fonte: Os Autores

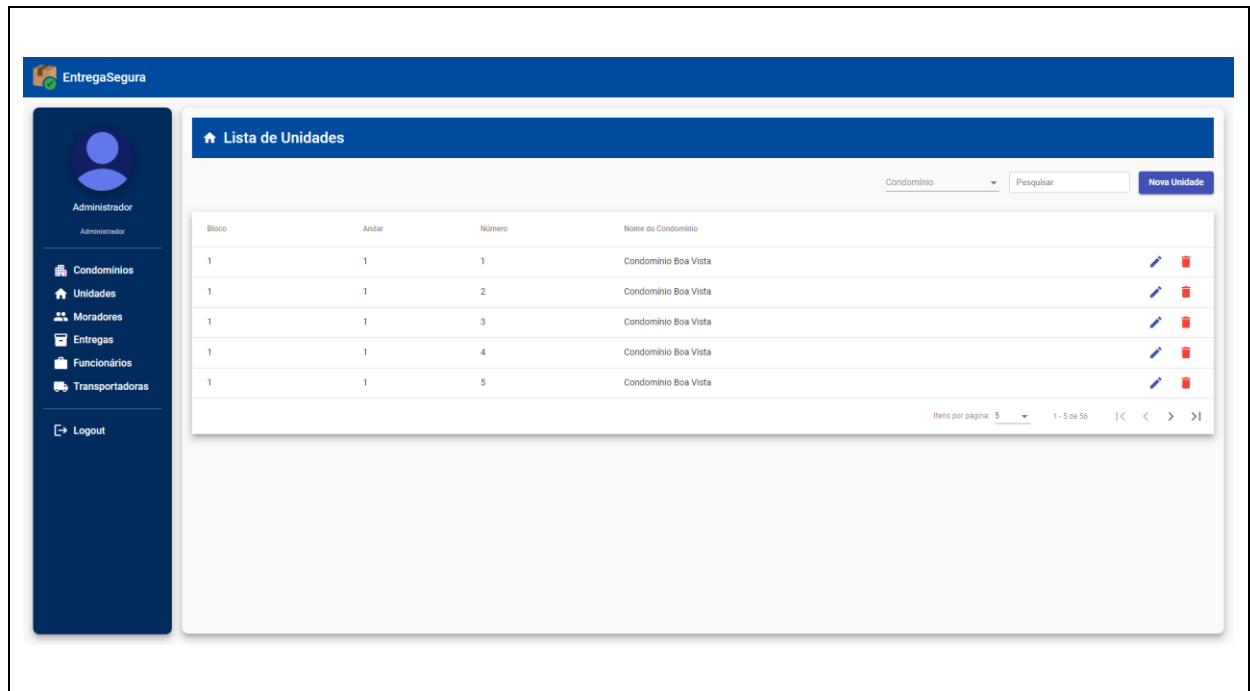


Figura 6: Lista de Unidades

Fonte: Os Autores

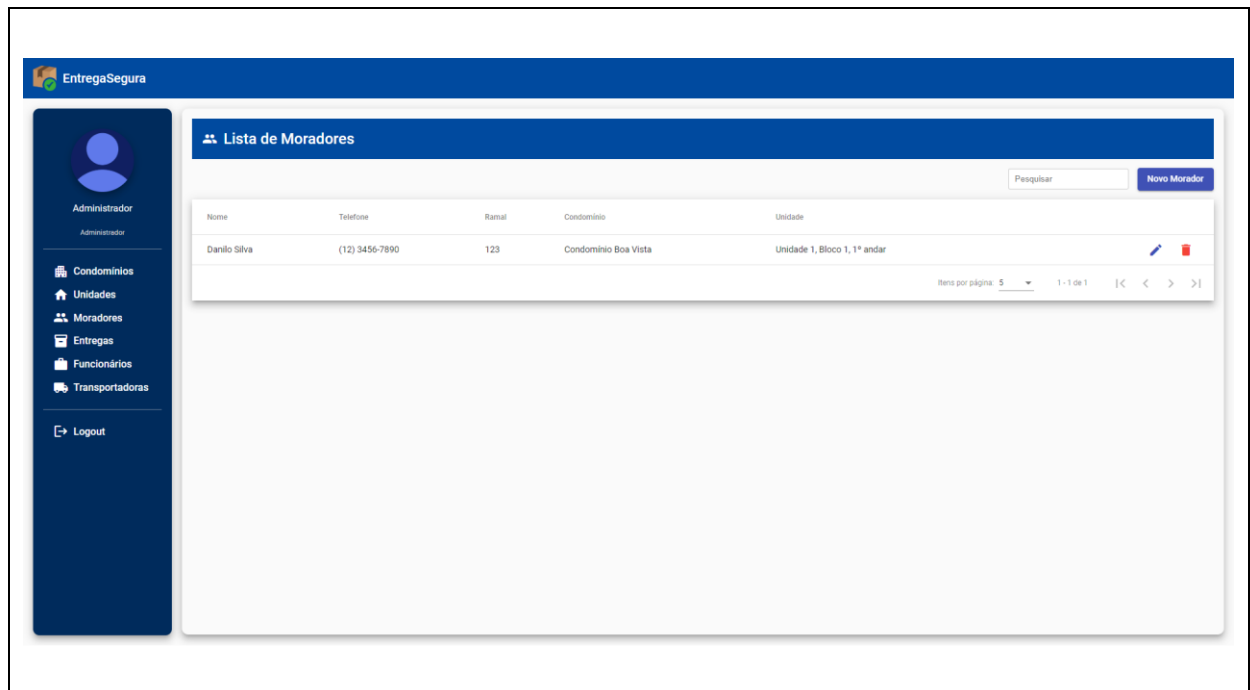


Figura 7: Lista de Moradores

Fonte: Os Autores

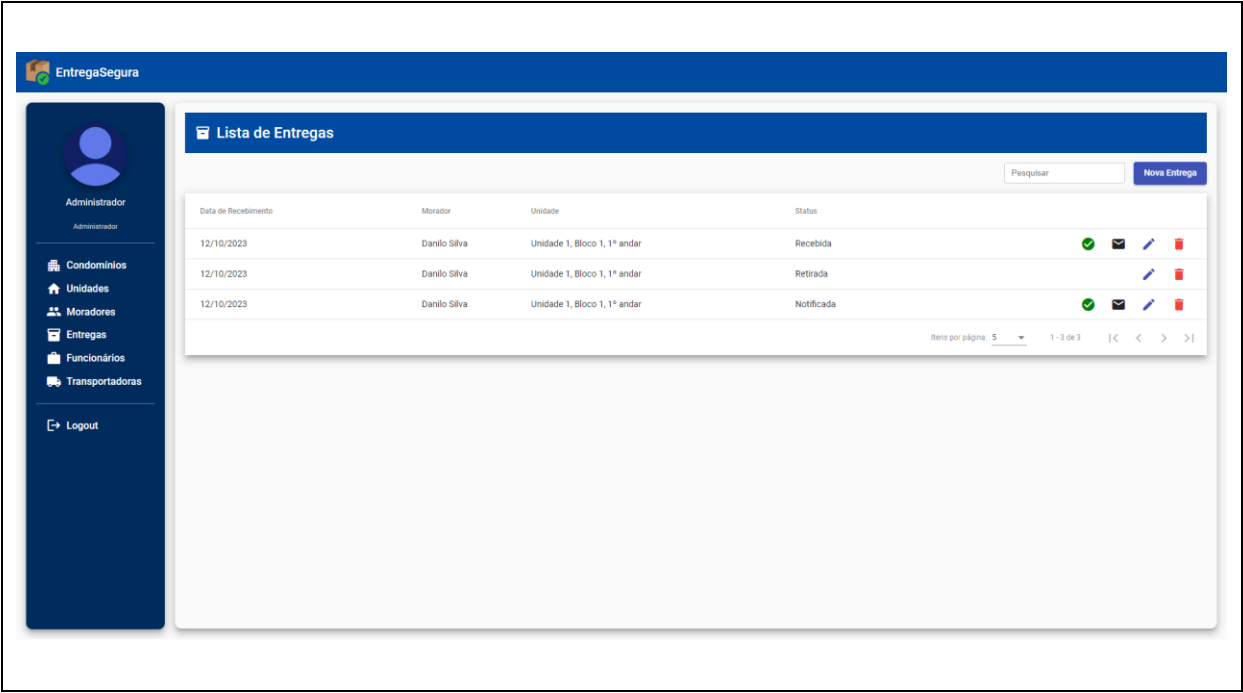


Figura 8: Lista de Entregas

Fonte: Os Autores

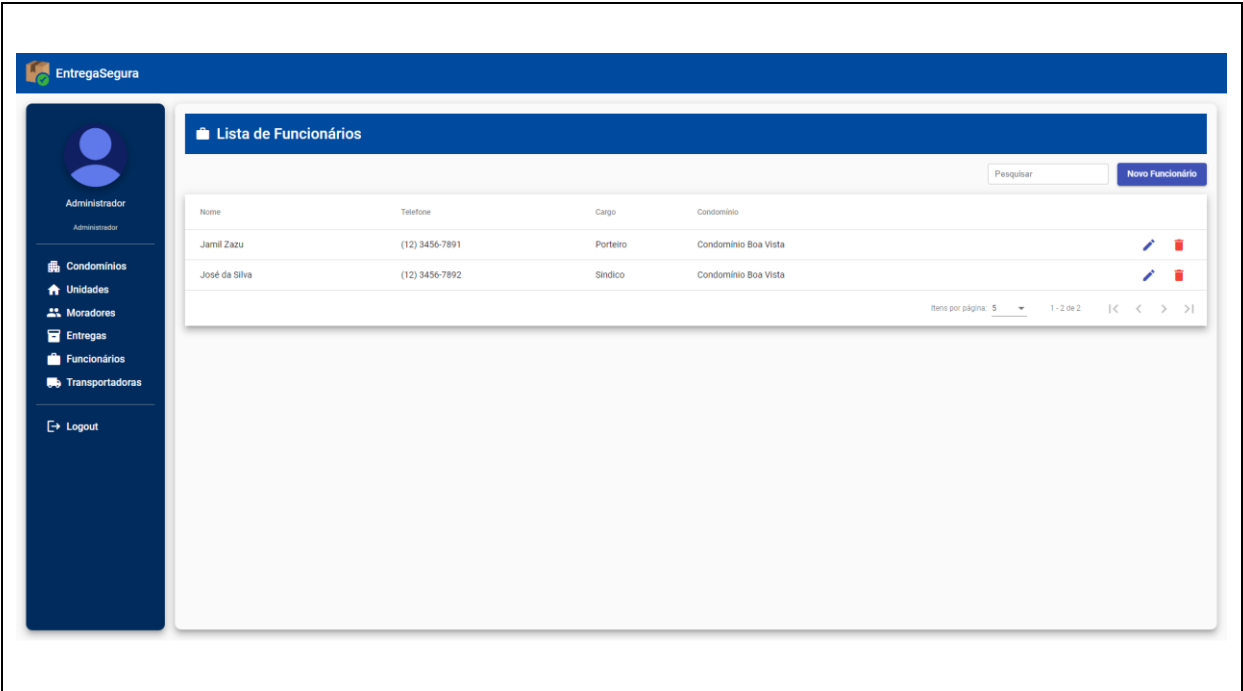


Figura 9: Lista de Funcionários

Fonte: Os Autores

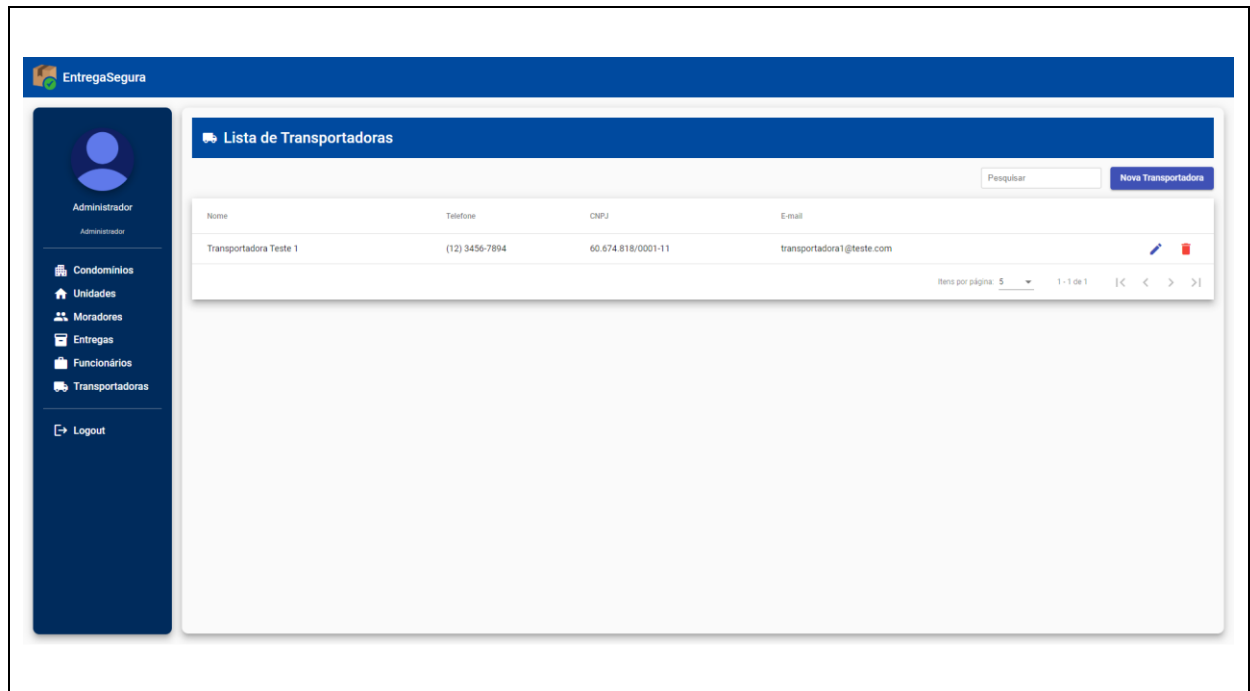


Figura 10: Lista de Transportadoras

Fonte: Os Autores

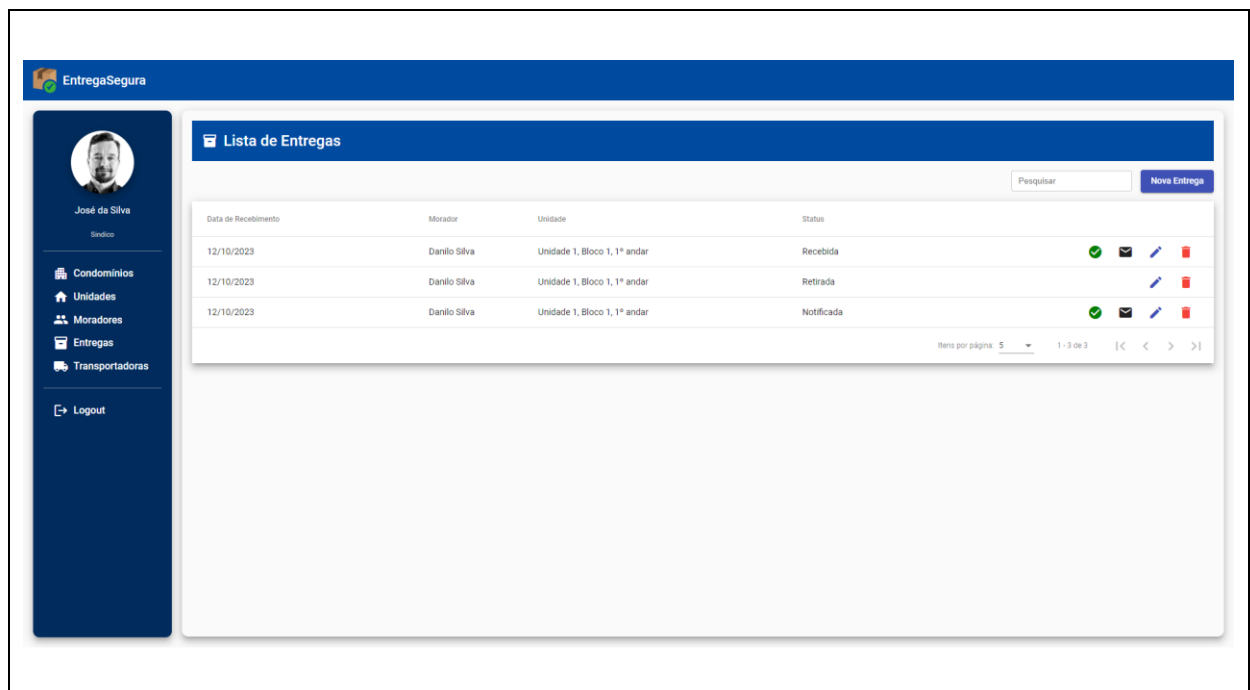


Figura 11: Visão Geral do Perfil de Síndico

Fonte: Os Autores

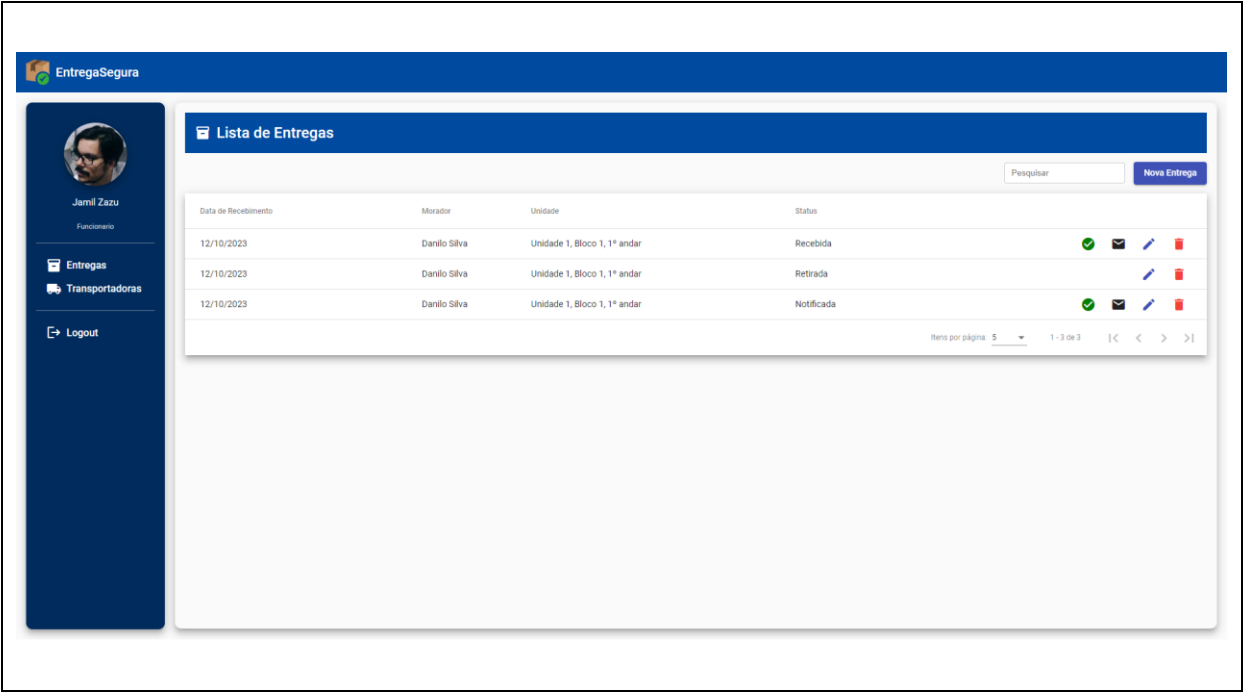


Figura 12: Visão Geral do Perfil de Funcionário

Fonte: Os Autores

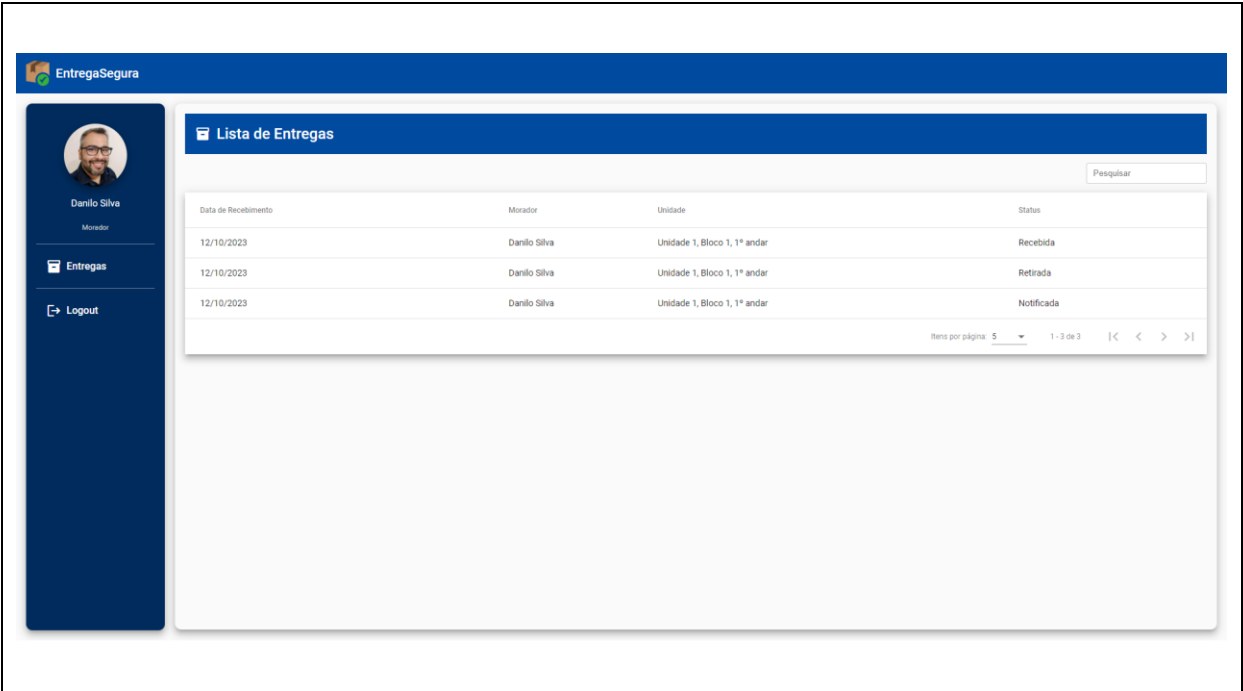


Figura 13: Visão Geral do Perfil de Morador

Fonte: Os Autores

9. Modelo Relacional ou Projeto de Banco de Dados NoSQL

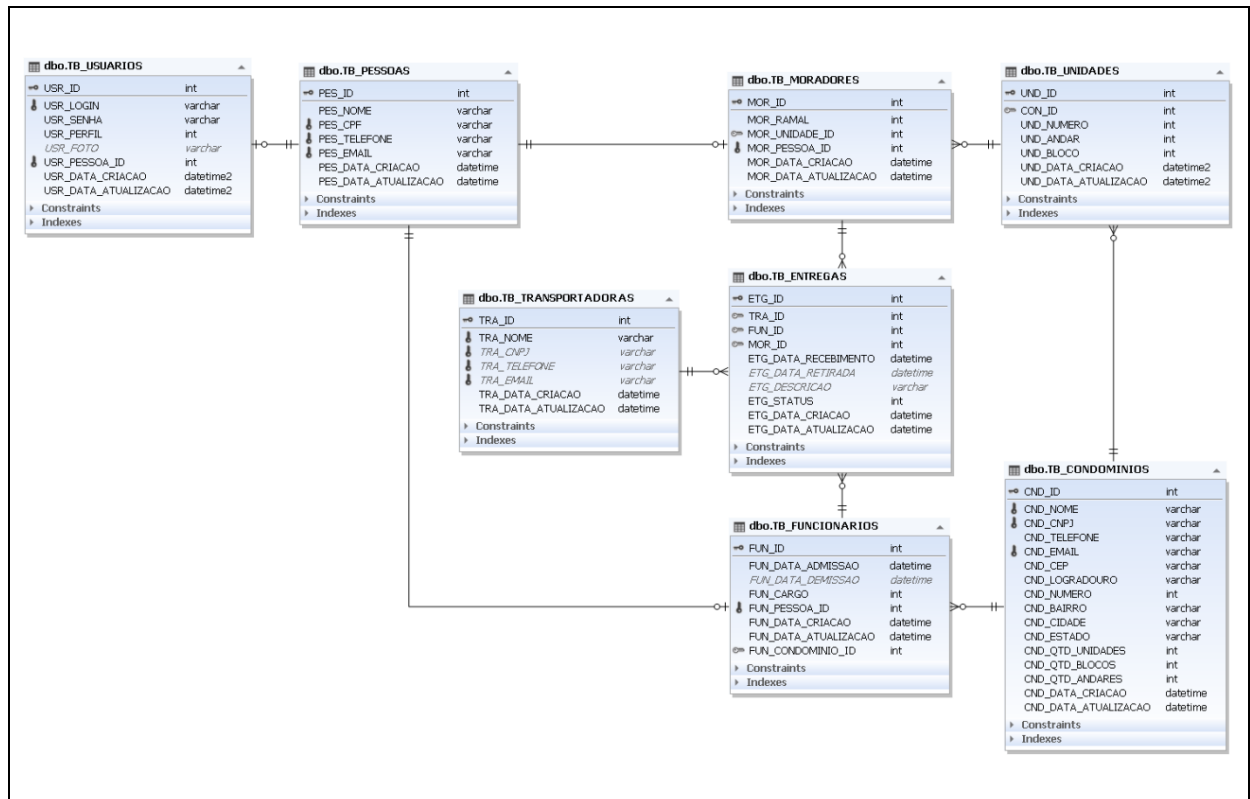


Figura 14: Modelo Entidade Relacionamento (MER)

Fonte: Os Autores

O arquivo original pode ser encontrado em: [Modelo Entidade Relacionamento \(MER\)](#)

10. Plano de Testes

ID	Caso de Uso	Objetivo do Caso de Teste	Entradas	Resultados Esperados
1	Cadastro de Condomínio	Cadastro com dados válidos	Dados válidos de condomínio	Condomínio cadastrado com sucesso.
2	Cadastro de Condomínio	Verificação de e-mail duplicado	E-mail já cadastrado	Mensagem: "Já existe um condomínio com este E-mail."
3	Cadastro de Condomínio	Verificação de nome duplicado	Nome já cadastrado	Mensagem: "Já existe um condomínio com este Nome."
4	Cadastro de Condomínio	Verificação de CNPJ duplicado	CNPJ já cadastrado	Mensagem: "Já existe um condomínio com este CNPJ."
5	Atualização de Condomínio	Atualização com dados válidos	Dados válidos e Condomínio existente	Condomínio é atualizado com sucesso.
6	Exclusão de Condomínio	Verificar exclusão correta	Condomínio existente	Condomínio é excluído com sucesso.
7	Cadastro de Unidade	Cadastro com dados válidos	Dados válidos e Condomínio existente	Unidade é cadastrada com sucesso.
8	Cadastro de Unidade	Verificação de bloco inexistente	Bloco inexistente	Mensagem: "O bloco especificado não existe neste condomínio."
9	Cadastro de Unidade	Verificação de informações duplicadas	Mesmo número, andar e bloco	Mensagem: "Já existe uma unidade com este número no mesmo andar neste bloco e condomínio."
10	Cadastro de Unidade	Verificação de condomínio inexistente	Condomínio inexistente	Mensagem: "O condomínio especificado não existe."
11	Atualização de Unidade	Atualização com dados válidos	Dados válidos e Unidade existente	Unidade é atualizada com sucesso.
12	Exclusão de Unidade	Verificar exclusão correta	Unidade existente	Unidade é excluída com sucesso.
13	Cadastro de Morador	Cadastro com dados válidos	Dados válidos e Unidade existente	Morador é cadastrado com sucesso.

14	Cadastro de Morador	Verificação de e-mail duplicado	E-mail já cadastrado	Mensagem: "Já existe um morador com este E-mail informado."
15	Cadastro de Morador	Verificação de nome duplicado	Nome já cadastrado	Mensagem: "Já existe um morador com este Nome informado."
16	Cadastro de Morador	Verificação de CPF duplicado	CPF já cadastrado	Mensagem: "Já existe um morador com este CPF informado."
17	Atualização de Morador	Atualização com dados válidos	Dados válidos e Morador existente	Morador é atualizado com sucesso.
18	Atualização de Morador	Verificar atualização de morador inexistente	Morador inexistente	Mensagem: "Morador não encontrado."
19	Exclusão de Morador	Verificar exclusão correta	Morador existente	Morador é excluído com sucesso.
20	Cadastro de Funcionário	Cadastro com dados válidos	Dados válidos e Condomínio existente	Funcionário é cadastrado com sucesso.
21	Cadastro de Funcionário	Verificação de telefone de morador	Telefone de um morador existente	Mensagem: "Não é permitido cadastrar um funcionário com o telefone de um morador."
22	Cadastro de Funcionário	Verificação de e-mail duplicado	E-mail já cadastrado	Mensagem: "Já existe um funcionário com este E-mail informado."
23	Cadastro de Funcionário	Verificação de nome duplicado	Nome já cadastrado	Mensagem: "Já existe um funcionário com este Nome informado."
24	Cadastro de Funcionário	Verificação de CPF duplicado	CPF já cadastrado	Mensagem: "Já existe um funcionário com este CPF informado."
25	Atualização de Funcionário	Atualização com dados válidos	Dados válidos e Funcionário existente	Funcionário é atualizado com sucesso.
26	Exclusão de Funcionário	Verificar exclusão correta	Funcionário existente	Funcionário é excluído com sucesso.
27	Cadastro de Transportadora	Cadastro com dados válidos	Dados válidos	Transportadora é cadastrada com sucesso.
28	Cadastro de Transportadora	Verificação de e-mail duplicado	E-mail já cadastrado	Mensagem: "Já existe uma transportadora com este E-mail."

29	Cadastro de Transportadora	Verificação de nome duplicado	Nome já cadastrado	Mensagem: "Já existe uma transportadora com este Nome."
30	Cadastro de Transportadora	Verificação de CNPJ duplicado	CNPJ já cadastrado	Mensagem: "Já existe uma transportadora com este CNPJ."
31	Atualização de Transportadora	Atualização com dados válidos	Dados válidos e transportadora existente	Transportadora é atualizada com sucesso.
32	Exclusão de Transportadora	Verificar exclusão correta	Transportadora existente	Transportadora é excluída com sucesso.
33	Cadastro de Entrega	Cadastro com dados válidos	Dados válidos	Entrega é cadastrada com sucesso.
34	Atualização de Entrega	Atualização com dados válidos	Dados válidos e entrega existente	Entrega é atualizada com sucesso.
35	Exclusão de Entrega	Verificar exclusão correta	Entrega existente	Entrega é excluída com sucesso.
36	Notificação da Entrega	Notificação da Entrega desejada	Notificação enviada	E-mail enviado para o Morador.
37	Retirada da Entrega	Retirada da Entrega desejada	Alteração de status da Entrega	Status da Entrega alterado "Retirada"

11. Apropriação de Horas no Projeto

Histórico de apropriação de horas		
Data do registro	Atividade	Quantidade de horas
04/03/2023	Contextualização e objetivos do trabalho	12:00
13/03/2023	Elaboração da proposta de projeto	10:00
23/03/2023	Levantamento de requisitos funcionais	18:00
02/04/2023	Levantamento de requisitos não-funcionais	8:00
10/04/2023	Documentação dos Requisitos Levantados	10:00
18/04/2023	Elaboração do Diagrama de Casos de Uso	16:00
26/04/2023	Desenvolvimento do Diagrama de Classes de Domínio	16:00
04/05/2023	Definição do Padrão Arquitetural	24:00
13/05/2023	Estrutura de Banco de Dados	18:00
15/07/2023	Desenvolvimento do sistema	200:00
24/07/2023	Elaboração do Plano de Testes	21:00
02/08/2023	Revisão da documentação do Projeto	18:00
10/08/2023	Criação do repositório no GitHub e deploy do protótipo	6:00
26/08/2023	Revisão da Apropriação de Horas	16:00
17/09/2023	Revisão de Formatação do Relatório Técnico	18:00
30/09/2023	Avaliação retrospectiva	06:00
03/10/2023	Entrega final	03:00

12. Código da Aplicação

- Link para o repositório com o código fonte do projeto: [Código Fonte](#)
- Link para a aplicação hospedada no Azure: [EntregaSegura](#)
- Link para o vídeo de apresentação da aplicação: Video de Apresentação

13. Avaliação Retrospectiva

O desenvolvimento deste projeto envolveu várias fases, desde a concepção inicial até a entrega final. Durante esse período, foram realizadas várias atividades que, em conjunto, contribuíram para a concretização da solução proposta.

13.1 Objetivos Estimados

Os objetivos estimados para este projeto foram:

- Compreender o contexto e os objetivos do trabalho;
- Propor um projeto eficaz com base nas necessidades identificadas;
- Realizar levantamentos detalhados de requisitos funcionais e não funcionais;
- Documentar todos os requisitos identificados;
- Desenvolver diagramas relevantes, como Casos de Uso e Classes de Domínio;
- Definir uma arquitetura robusta;
- Estruturar o banco de dados;
- Desenvolver o sistema conforme as especificações;
- Elaborar um plano de testes eficaz;
- Revisar a documentação do projeto, garantindo sua completude e precisão;
- Realizar a implementação no ambiente final e testar o protótipo;
- Assegurar o correto registro das horas e a formatação do relatório técnico.

13.2 Objetivos Alcançados

Os objetivos alcançados para este projeto foram:

- A contextualização e definição clara dos objetivos do trabalho foram realizadas com sucesso.
- A proposta de projeto foi elaborada e aprovada.
- Os requisitos, tanto funcionais quanto não-funcionais, foram levantados e documentados de forma detalhada.
- Foram desenvolvidos diagramas, incluindo Casos de Uso e Classes de Domínio.

- Um padrão arquitetural sólido foi definido.
- O banco de dados foi estruturado de acordo com as necessidades do projeto.
- O sistema foi desenvolvido, testado e implementado.
- A documentação do projeto foi revisada e atualizada conforme necessário.
- A entrega final foi realizada dentro do prazo estabelecido.

13.3 Lições aprendidas

Retrospectiva (Lições Aprendidas)		
	Descrição da Lição	Classificação
1	Compreender a importância de contextualização e definição clara dos objetivos	Positiva
2	A necessidade de revisões frequentes na documentação para garantir precisão	Positiva
3	O desenvolvimento do sistema consumiu a maior parte do tempo	Negativa
4	Planejamento eficaz das horas é essencial para evitar sobrecargas	Positiva
5	A importância de um plano de testes bem estruturado	Positiva
6	Garantir backups regulares, especialmente antes de grandes implementações	Positiva
7	O uso de ferramentas adequadas para visualização e documentação é crucial	Positiva
8	O compromisso contínuo é necessário para alcançar os objetivos dentro do prazo	Positiva
9	Desafios na estruturação do banco de dados reforçam a necessidade de um planejamento inicial robusto	Negativa
10	A colaboração e feedback contínuo são essenciais para o desenvolvimento de projetos	Positiva
11	O deploy e teste do protótipo são etapas críticas que requerem atenção especial	Negativa
12	Revisões periódicas são essenciais para garantir qualidade e aderência aos padrões	Positiva
13	Ferramentas de modelagem de banco de dados, como dbForge, podem ser mais eficientes do que abordagens manuais	Positiva
14	A importância de conhecer e explorar diferentes ferramentas de diagramação, como draw.io e visual paradigm	Positiva
15	Estar aberto a mudanças e adaptações ao longo do projeto ajuda a superar desafios	Positiva
16	Reservar um tempo específico para avaliações retrospectivas fornece insights valiosos para projetos futuros	Positiva

14. Referências

Comércio eletrônico gerou R\$ 169,6 bilhões em vendas em 2022. **Abranet (Associação Brasileira de Internet)**, 25 de janeiro de 2023. Disponível em: <[https://www.abranet.org.br/Noticias/Comercio-eletronico-gerou-R\\$-169,6-bilhoes-em-vendas-em-2022-4181.html](https://www.abranet.org.br/Noticias/Comercio-eletronico-gerou-R$-169,6-bilhoes-em-vendas-em-2022-4181.html)> Acesso em: 10 de maio de 2023.