



UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA DE CIÊNCIAS EXATAS E TECNOLÓGICAS
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

**HARRY CICILIANI SCHLORKE
ANTONIO JUNIOR DOS SANTOS
GUILHERME AMARAL CARDOSO
JULIANO BOAVENTURA FIALHO
CESAR AUGUSTO PEREIRA**

**Hands On Work VI
Projeto e Desenvolvimento de Sistema Computacional
com Persistência de Dados**

RELATÓRIO TÉCNICO DA ATIVIDADE HANDS ON WORK VI

Etapa 1

ITAJAÍ – SANTA CATARINA – BRASIL

2025

**HARRY CICILIANI SCHLORKE
ANTONIO JUNIOR DOS SANTOS
GUILHERME AMARAL CARDOSO
JULIANO BOAVENTURA FIALHO
CESAR AUGUSTO PEREIRA**

Hands On Work VI
**Projeto e Desenvolvimento de Sistema Computacional
com Persistência de Dados**

Relatório técnico apresentado ao curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade do Vale do Itajaí – UNIVALI, como parte das atividades práticas da disciplina de Projeto e Desenvolvimento de Sistema Computacional com Persistência de Dados.

Orientador: Prof. Andre Luis Alice Raabe

Etapa 1

ITAJAÍ – SANTA CATARINA – BRASIL

2025

EasyRent : Sistema Web para Gestão de Locação de Veículos com Modelagem Orientada a Objetos e Persistência de Dados / Harry Ciciliani Schlorke ... [et al.]. - Porto Alegre, 2025.

30 f. : il., color.

Projeto de Iniciação Científica Universidade do Vale do Itajaí – UNIVALI, Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, 2025.

Orientador: André Luis Alice Raabe

1. Sistema Web. 2. Locação de Veículos. 3. Programação Orientada a Objetos. 4. Persistência de Dados. 5. HTML, CSS e JavaScript. 6. Diagrama de Classes. I. Schlorke, Harry Ciciliani. II. Dos Santos, Antonio Junior. III. Cardoso, Guilherme Amaral. IV. Fialho, Juliano Boaventura. V. Raabe, André Luis Alice. VI. Universidade do Vale do Itajaí. VII. Título.

E-mail: harry.8314900@edu.univali.br

Dedico este trabalho ao meu filho, Pedro, cuja existência ilumina todos os meus dias e fortalece minha jornada acadêmica.

À minha família, pelo apoio constante, amor incondicional e paciência durante os momentos de maior desafio.

Aos professores e colegas do curso de Tecnologia em Análise e Desenvolvimento de Sistemas da UNIVALI, por contribuírem significativamente para minha formação profissional e pessoal.

Agradecimentos

Agradeço primeiramente à minha família, por todo o apoio emocional, incentivo constante e compreensão durante os desafios enfrentados ao longo desta jornada acadêmica.

Ao meu filho, Pedro, que mesmo tão pequeno, representa a maior motivação para seguir e buscar um futuro melhor. Sua presença me fortalece e inspira diariamente.

Aos colegas de curso, em especial Antonio, pela parceria nos projetos e apoio mútuo em diversas etapas.

Aos professores da UNIVALI, especialmente ao professor Andre Luis Alice Raabe, pelo comprometimento com o ensino, pelos ensinamentos transmitidos com dedicação e pela orientação técnica ao longo da atividade.

A todos que, de forma direta ou indireta, contribuíram para a realização deste trabalho, meus sinceros agradecimentos.

“O único lugar onde o sucesso vem antes do trabalho é no dicionário.”

— *Albert Einstein*

Resumo

Este trabalho apresenta a Etapa 1 do projeto Hands On Work VI, desenvolvido no curso de Tecnologia em Análise e Desenvolvimento de Sistemas da UNIVALI. O objetivo desta etapa foi modelar, estruturar e prototipar duas entidades inter-relacionadas de um sistema de locação de veículos. Essa modelagem baseou-se nos princípios da Programação Orientada a Objetos (POO) e do banco de dados relacional. Foram produzidos um diagrama de classes e um diagrama entidade-relacionamento (DER), além de protótipos de interface para as funcionalidades de cadastro e listagem. A atividade contribuiu para o desenvolvimento da fluência tecnológica dos alunos. Isso foi alcançado por meio da aplicação prática de técnicas de persistência de dados e modelagem de sistemas computacionais, em alinhamento com os princípios do projeto de extensão “Lite is Cool”.

Palavras-chave: POO. DER. Locação de Veículos. Protótipos. Persistência de Dados

Abstract

This paper presents Stage 1 of the Hands On Work VI project, developed in the undergraduate program of Technology in Systems Analysis and Development at UNIVALI. The objective of this stage was to model, structure, and prototype two interrelated entities of a vehicle rental system. This modeling was based on the principles of Object-Oriented Programming (OOP) and relational database design. A class diagram and an entity-relationship diagram (ERD) were created, in addition to interface prototypes for registration and listing functionalities. The activity contributed to the development of students' technological fluency through the practical application of data persistence techniques and systems modeling, aligned with the principles of the "Lite is Cool" extension project.

Keywords: OOP. ERD. Vehicle rental. Prototype. Data persistence

Lista de ilustrações

Figura 1 – Protótipo da Tela de Login	17
Figura 2 – Protótipo da Tela de Cadastro de Usuário	18
Figura 3 – Protótipo da Página de Locação	19
Figura 4 – Protótipo do Painel Administrativo	20
Figura 5 – Diagrama de Classes UML do Sistema EasyRent	26
Figura 6 – Diagrama de Sequência UML do Login de Usuário no Sistema EasyRent	27
Figura 7 – Diagrama de Sequência UML do Fluxo de Locação no Sistema EasyRent	28
Figura 8 – Diagrama Entidade-Relacionamento (DER) do Sistema EasyRent	29

Lista de abreviaturas e siglas

ABNT	ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DER	Diagrama Entidade-Relacionamento
DOM	Document Object Model
HTML	HyperText Markup Language
ID	Identificador
JS	JavaScript
MER	Modelo Entidade-Relacionamento
PDF	Portable Document Format
POO	Programação Orientada a Objetos
SGBD	Sistema Gerenciador de Banco de Dados
SQL	Structured Query Language
TI	Tecnologia da Informação
UML	Unified Modeling Language
UNIVALI	Universidade do Vale do Itajaí

Sumário

1	INTRODUÇÃO	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	Programação Orientada a Objetos (POO)	13
2.2	Diagrama Entidade-Relacionamento (DER)	13
2.3	Persistência de Dados	13
2.4	Prototipação de Interfaces Web	14
3	MATERIAIS E MÉTODOS	15
3.1	Materiais	15
3.2	Métodos	15
3.2.1	Levantamento de Requisitos	15
3.2.2	Prototipação de Telas	16
3.2.3	Implementação das Interfaces	16
3.2.4	Aplicação de Responsividade	16
3.2.5	Modelagem de Dados	16
3.2.6	Protótipos de Interface – Figma	16
3.2.7	Modelagem Orientada a Objetos	20
4	RESULTADOS E DISCUSSÃO	21
4.1	Interfaces Desenvolvidas	21
4.1.1	Página de Login	21
4.1.2	Página de Cadastro	21
4.1.3	Página de Locação (Home)	21
4.1.4	Painel Administrativo	21
4.2	Modelagem e Estrutura	22
4.2.1	Diagrama de Classes	22
4.2.2	Diagrama Entidade-Relacionamento	22
4.3	Discussão	22
5	CONSIDERAÇÕES FINAIS	23
	Referências	24
	APÊNDICES	25
	APÊNDICE A – Diagrama de Classes UML do Sistema EasyRent	26

APÊNDICE B – Diagrama de Sequência: Login de Usuário	27
APÊNDICE C – Diagrama de Sequência: Fluxo de Locação	28
APÊNDICE D – Modelo e Diagrama Entidade-Relacionamento (MER/DER) do Sistema EasyRent	29

1 INTRODUÇÃO

A transformação digital impulsiona o desenvolvimento de soluções tecnológicas cada vez mais eficazes para os setores de mobilidade e serviços. O mercado de locação de veículos, em especial, exige sistemas que integrem usabilidade, persistência de dados e segurança nas operações. Essas demandas se alinham à necessidade de formar profissionais capacitados para projetar, modelar e implementar soluções computacionais robustas (Sommerville, 2019), especialmente no contexto do curso de Tecnologia em Análise e Desenvolvimento de Sistemas da UNIVALI.

Diante desse cenário, o presente trabalho foi desenvolvido no âmbito da disciplina “Projeto e Desenvolvimento de Sistema Computacional com Persistência de Dados”. O objetivo foi aplicar na prática os conhecimentos de modelagem orientada a objetos e estruturação de banco de dados relacional.

A proposta consiste na criação do sistema EasyRent, voltado à gestão de locação de veículos, com foco na modelagem de duas entidades principais: Marca e Carro.

Essas entidades foram representadas por meio de dois modelos distintos (Larman; Craig, 2007). O primeiro é um diagrama de classes UML, elaborado com base nos princípios da Programação Orientada a Objetos (POO). O segundo é um diagrama entidade-relacionamento (DER), voltado à estruturação do banco de dados relacional.

Além disso, foram desenvolvidos protótipos de interface para as funcionalidades de cadastro e listagem, utilizando tecnologias web como HTML, CSS e JavaScript (MOZILLA FOUNDATION, 2025).

Este trabalho também se alinha ao projeto de extensão da UNIVALI “Lite is Cool” (UNIVERSIDADE DO VALE DO ITAJAÍ, 2025), que promove a fluência tecnológica por meio da aprendizagem prática baseada em projetos. O sistema EasyRent visa não apenas consolidar os conhecimentos adquiridos na disciplina, mas também apresentar uma solução funcional e estruturada com base em boas práticas de desenvolvimento.

Este relatório está estruturado da seguinte forma: na Seção 2, é apresentada a fundamentação teórica, abordando os principais conceitos que sustentam o projeto. A Seção 3 detalha os materiais, métodos e tecnologias utilizados. A Seção 4 apresenta os resultados obtidos com a modelagem e prototipação. Por fim, a Seção 5 traz as conclusões e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta os fundamentos teóricos que embasam o desenvolvimento do sistema EasyRent. Os principais conceitos envolvem a Programação Orientada a Objetos (POO), a modelagem de dados com Diagramas Entidade-Relacionamento (DER), técnicas de persistência de dados e a prototipação de interfaces web. A seguir, cada um desses tópicos será abordado de forma objetiva, considerando sua aplicação no contexto da disciplina e do projeto.

2.1 Programação Orientada a Objetos (POO)

A Programação Orientada a Objetos (POO) é um paradigma de desenvolvimento que organiza o software com base em objetos — elementos que encapsulam dados (atributos) e comportamentos (métodos). Esse paradigma proporciona maior modularidade, reutilização e manutenção do código (Larman; Craig, 2007), sendo amplamente utilizado em sistemas modernos.

A POO é sustentada por quatro pilares fundamentais. O primeiro é a *abstração*, que permite representar entidades reais com simplicidade. O segundo é o *encapsulamento*, responsável por proteger o estado interno dos objetos. O terceiro pilar é a *herança*, que favorece a reutilização de código. Por fim, o *polimorfismo* possibilita a redefinição de comportamentos em classes derivadas.

No sistema EasyRent, a modelagem de objetos foi aplicada na construção das entidades Marca e Carro, possibilitando uma representação lógica eficiente do domínio do sistema.

2.2 Diagrama Entidade-Relacionamento (DER)

O Diagrama Entidade-Relacionamento (DER) é uma ferramenta da engenharia de software utilizada para modelar graficamente os dados de um sistema. Ele permite representar entidades, atributos e os relacionamentos entre elas, sendo a base para o projeto de banco de dados relacionais (Pressman; Maxim, 2016).

No projeto EasyRent, o DER representa duas entidades principais. A entidade Marca contém atributos como ID e nome. Já a entidade Carro possui atributos como ID, modelo, cor, ano e uma chave estrangeira que referencia a entidade Marca. Essa modelagem garante integridade referencial e facilita a implementação futura do banco de dados.

2.3 Persistência de Dados

A persistência de dados refere-se à capacidade de um sistema de manter os dados armazenados (Date, 2004) de forma permanente, mesmo após o encerramento da apli-

cação. Em sistemas relacionais, isso é realizado por meio de tabelas e bancos de dados estruturados.

Embora o sistema EasyRent ainda não esteja conectado a um SGBD, a estrutura lógica definida no DER é suficiente para representar a persistência dos dados no nível conceitual. Isso viabiliza uma futura implementação com bancos como MySQL ou PostgreSQL (MYSQL, 2025; POSTGRESQL, 2025).

2.4 Prototipação de Interfaces Web

A prototipação é uma etapa fundamental no desenvolvimento de interfaces (Garrett; James, 2011), pois permite validar, testar e aprimorar a experiência do usuário antes da implementação final. Prototipar possibilita detectar falhas de usabilidade, além de facilitar a comunicação entre desenvolvedores e usuários.

No projeto EasyRent, foram desenvolvidos protótipos utilizando HTML, CSS e JavaScript. As interfaces contemplam funcionalidades como login, cadastro, listagem e painel administrativo, todas organizadas de maneira responsiva e funcional.

3 MATERIAIS E MÉTODOS

3.1 Materiais

O desenvolvimento do sistema EasyRent utilizou ferramentas e linguagens voltadas à criação de interfaces web modernas e responsivas. Os principais materiais empregados foram:

- **HTML5:** Para a estruturação semântica das páginas do sistema.
- **CSS3:** Para a estilização visual das interfaces, com uso de flexbox, grid e media queries para responsividade.
- **JavaScript (ES6+):** Para a lógica de interação do menu, manipulação de elementos DOM e comportamentos dinâmicos.
- **Visual Studio Code:** Ambiente de desenvolvimento utilizado ao longo de toda a codificação.
- **Figma:** Ferramenta gráfica utilizada para criação dos protótipos de interface.
- **Paradigm Community Edition:** Utilizada na modelagem dos diagramas UML (VISUAL PARADIGM, 2025), incluindo diagrama de classes e diagramas de sequência.
- **Google Fonts / Material Icons:** Recursos utilizados para a personalização da tipografia e aplicação de ícones na interface.

3.2 Métodos

A metodologia utilizada no projeto seguiu uma abordagem prática e incremental, baseada na construção de protótipos funcionais. As etapas metodológicas foram:

3.2.1 Levantamento de Requisitos

Inicialmente, foram definidos os requisitos essenciais para o funcionamento da aplicação, com base em um cenário fictício de locadora de veículos. Os principais requisitos definidos foram:

- Cadastro de marcas, carros e usuários;
- Visualização e listagem de veículos;
- Painel administrativo com ações de edição e exclusão;
- Tela de login e cadastro com formulário validado.

3.2.2 Prototipação de Telas

A prototipação das telas foi realizada no Figma, permitindo validar a disposição dos elementos antes da implementação. Foram prototipadas as páginas: **Login**, **Cadastro**, **Locação (home)** e **Painel Administrativo**.

3.2.3 Implementação das Interfaces

Com os protótipos validados, iniciou-se o desenvolvimento das interfaces utilizando HTML, CSS e JavaScript. O projeto seguiu uma estrutura modular, com a separação de responsabilidades por arquivos (style.css, script.js, e arquivos HTML individuais).

3.2.4 Aplicação de Responsividade

O sistema foi projetado para se adaptar a diferentes tamanhos de tela, utilizando media queries e unidades relativas no CSS. O menu principal também foi adaptado com comportamento de **menu hambúrguer** para dispositivos móveis.

3.2.5 Modelagem de Dados

A modelagem de dados foi realizada por meio de um **Diagrama Entidade-Relacionamento (DER)** e um **Modelo Entidade-Relacionamento (MER)**, contemplando as entidades Marca e Carro. O relacionamento foi representado com integridade referencial, indicando que cada carro pertence a uma marca.

3.2.6 Protótipos de Interface – Figma

Com o objetivo de validar visualmente as funcionalidades propostas para o sistema EasyRent antes de sua implementação, foram desenvolvidos protótipos utilizando a ferramenta **Figma**. Esses protótipos possibilitaram simular a navegação, avaliar a usabilidade e alinhar o design das interfaces com os requisitos levantados na etapa de levantamento e análise.

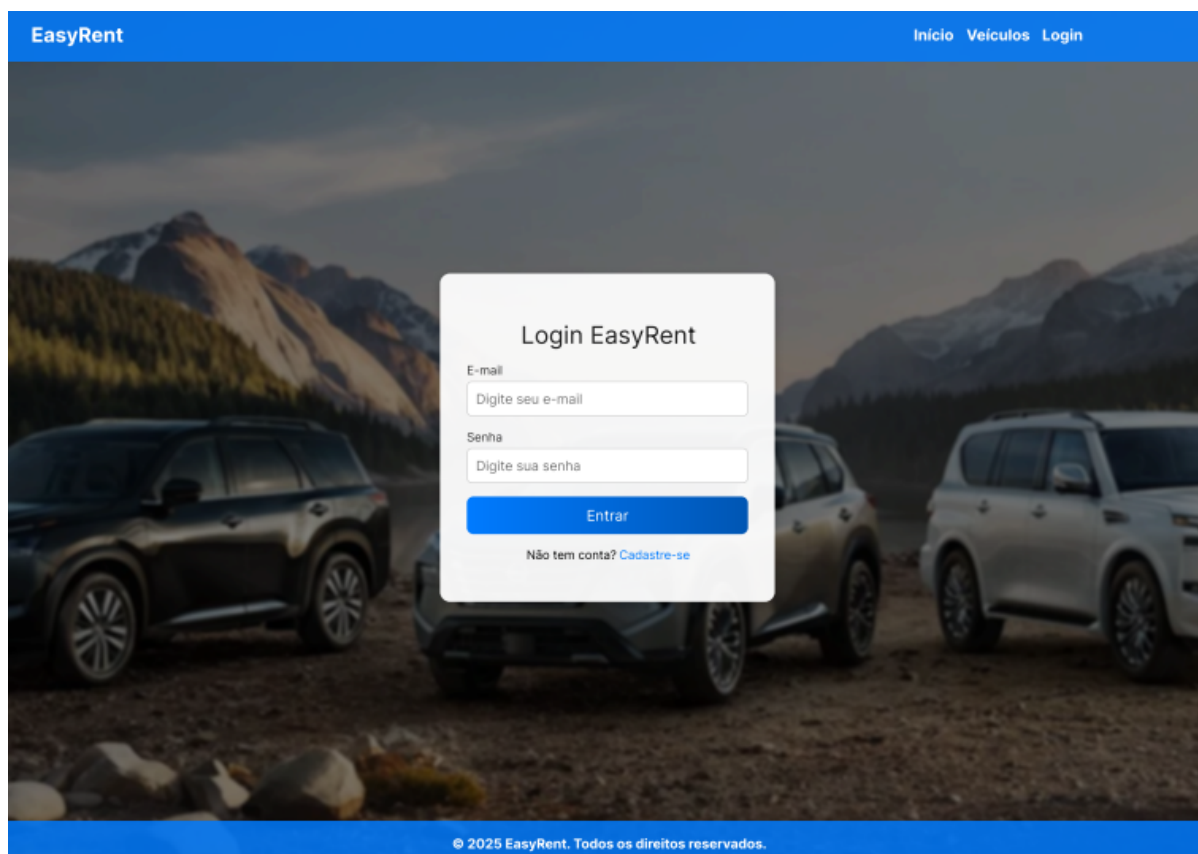
A seguir, estão os protótipos desenvolvidos para as principais telas do sistema:

- **Tela de Login:** permite ao usuário inserir e-mail e senha para acessar o sistema.
- **Tela de Cadastro:** possibilita o registro de novos usuários com campos para nome, e-mail, senha e confirmação.
- **Página de Locação (Home):** exibe uma galeria de veículos disponíveis com cards contendo imagem, nome, descrição e preço por dia, além de botão “Alugar”.

- **Painel Administrativo:** reúne os formulários para cadastro de marcas, carros e usuários, além da listagem de veículos já registrados com botões de edição e exclusão.

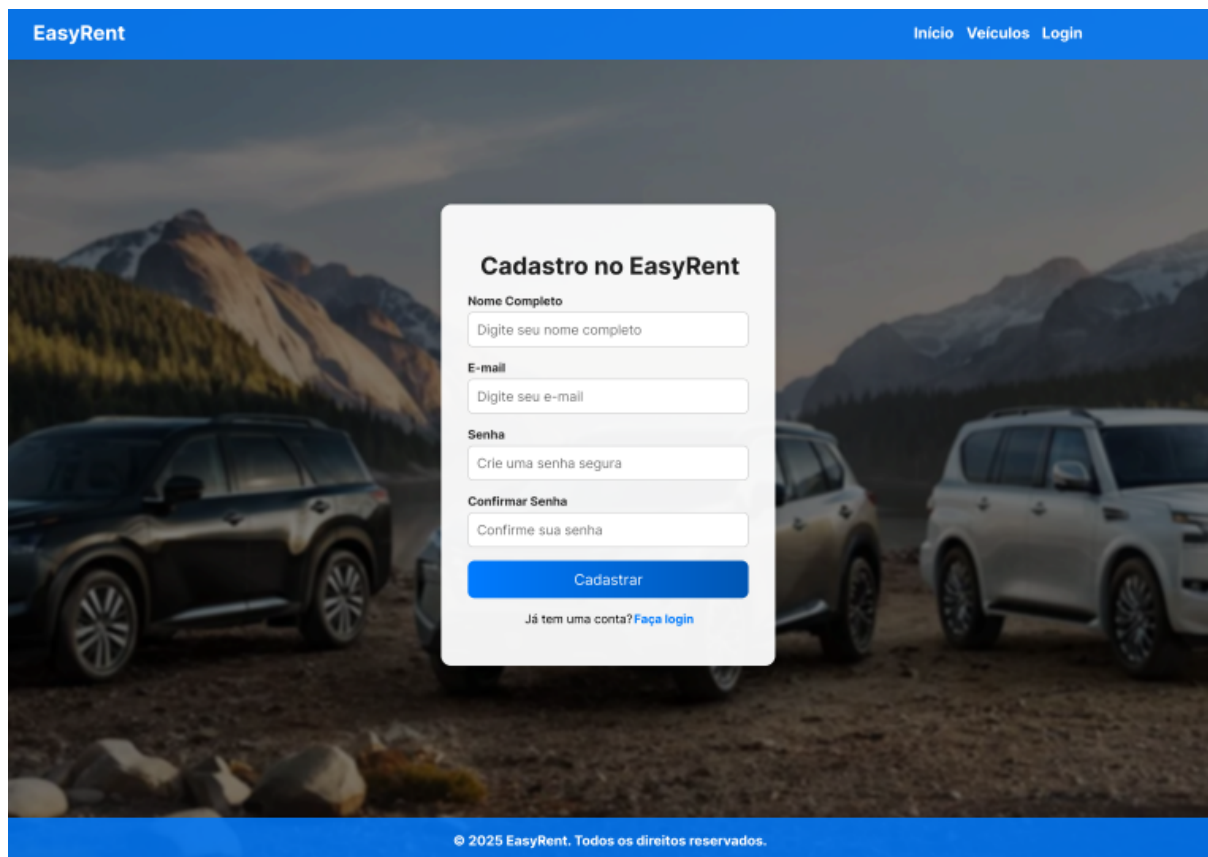
Os protótipos foram desenvolvidos com atenção à responsividade e acessibilidade, respeitando boas práticas de design de interface e usabilidade.

Figura 1 – Protótipo da Tela de Login



Fonte: Elaborado pelo Harry Schlorke (2025)

Figura 2 – Protótipo da Tela de Cadastro de Usuário



O protótipo da tela de cadastro do EasyRent apresenta uma interface limpa e moderna. No topo, uma barra azul contém o logo "EasyRent" à esquerda e os links "Início", "Veículos" e "Login" à direita. O fundo da página é uma imagem de paisagem com montanhas e carros estacionados. Centralizado na tela, há um formulário branco com o título "Cadastro no EasyRent". O formulário contém campos para "Nome Completo", "E-mail", "Senha" e "Confirmar Senha", cada um com uma etiqueta e um campo de entrada. Abaixo dos campos, há um botão azul "Cadastrar" e um link "Já tem uma conta? Faça login". Na base da página, uma barra azul contém o texto "© 2025 EasyRent. Todos os direitos reservados."

EasyRent

Início Veículos Login

Cadastro no EasyRent

Nome Completo
Digite seu nome completo

E-mail
Digite seu e-mail

Senha
Crie uma senha segura

Confirmar Senha
Confirme sua senha

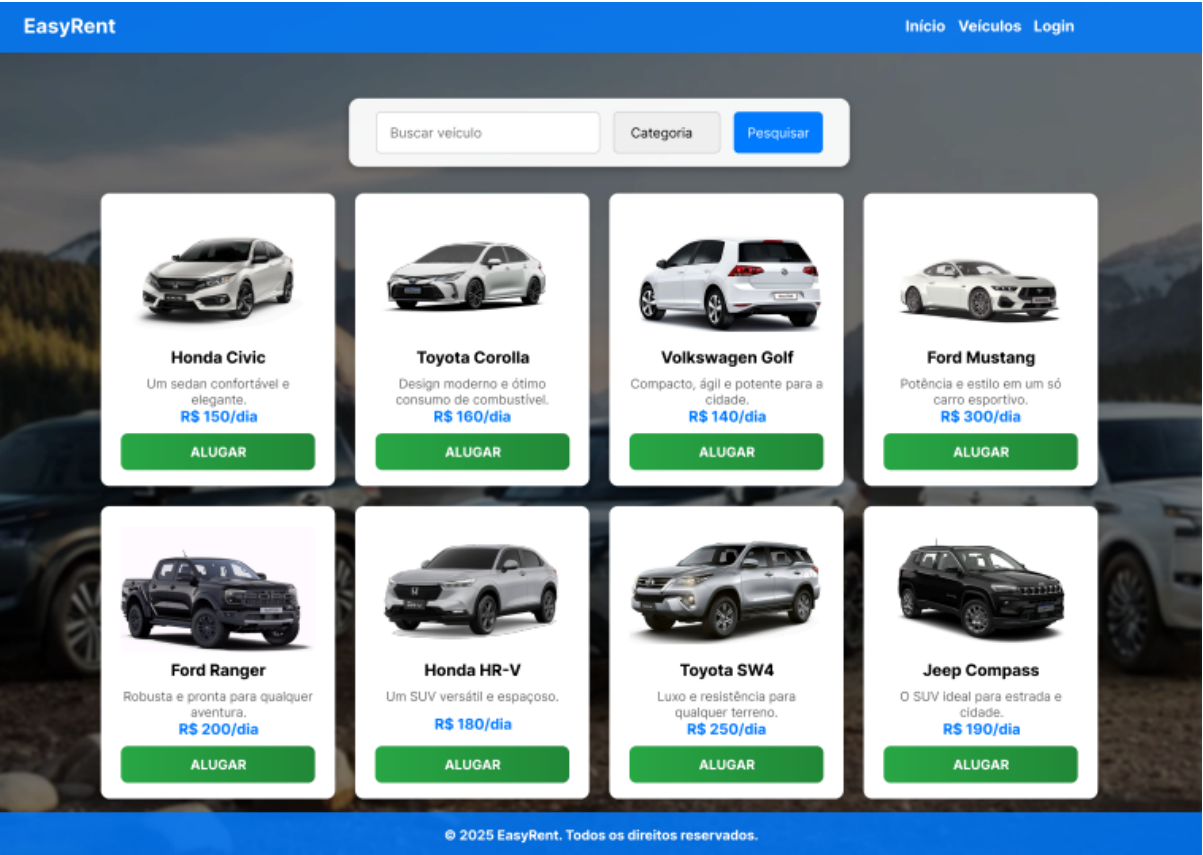
Cadastrar

Já tem uma conta? [Faça login](#)

© 2025 EasyRent. Todos os direitos reservados.

Fonte: Elaborado pelo Harry Schlorke (2025)

Figura 3 – Protótipo da Página de Locação



Fonte: Elaborado pelo Harry Schlorke (2025)

Figura 4 – Protótipo do Painel Administrativo

Fonte: Elaborado pelo Harry Schlorke (2025)

3.2.7 Modelagem Orientada a Objetos

Também foi elaborado um **Diagrama de Classes UML**, representando as classes **Marca** e **Carro**, com seus respectivos atributos e métodos, conforme os princípios da Programação Orientada a Objetos.

4 RESULTADOS E DISCUSSÃO

4.1 Interfaces Desenvolvidas

Com base nas etapas metodológicas, foram construídas quatro páginas principais do sistema EasyRent:

4.1.1 Página de Login

A página de login conta com campos de entrada para e-mail e senha, além de redirecionamento para a página de cadastro. A interface é responsiva e centralizada, com fundo escurecido e formulário em destaque.

4.1.2 Página de Cadastro

A página de cadastro permite que novos usuários preencham nome, e-mail e senha com confirmação. O layout mantém a consistência visual com a página de login, sendo adaptável para diferentes tamanhos de tela.

4.1.3 Página de Locação (Home)

A página inicial apresenta uma **galeria de veículos disponíveis para locação**, com cards contendo imagem, nome, descrição e preço por dia. Os cards são exibidos em grid responsivo e contam com botão de ação “Alugar”.

4.1.4 Painel Administrativo

O painel administrativo contém três seções:

- **Cadastro de Marcas:** Com campos para código e nome da marca;
- **Cadastro de Carros:** Com campos para código, marca, modelo, ano, cor, descrição e observações;
- **Cadastro de Usuários:** Com campos para código, nome, e-mail e senha.

Abaixo do painel, são listados os veículos já cadastrados, com opções de **editar** e **excluir**, organizados em cards semelhantes aos da página inicial.

4.2 Modelagem e Estrutura

4.2.1 Diagrama de Classes

O diagrama de classes desenvolvido representa as entidades Marca e Carro, com atributos e métodos representando o encapsulamento das informações. Essa modelagem reforça a aplicação dos conceitos de POO aprendidos em aula (Larman; Craig, 2007).

4.2.2 Diagrama Entidade-Relacionamento

O DER estrutura a base de dados com relações entre as entidades Marca e Carro. Cada carro está vinculado a uma marca, garantindo a integridade dos dados. O modelo permite futura expansão para incluir locações, clientes e administradores.

4.3 Discussão

O desenvolvimento do EasyRent demonstrou a eficácia da prototipação na construção de sistemas funcionais, mesmo sem integração com banco de dados real. As boas práticas de design e desenvolvimento aplicadas ao projeto resultaram em uma interface clara, acessível e adaptável, permitindo a visualização de um sistema coeso.

A implementação dos formulários com validações básicas e a organização das telas confirmaram a viabilidade de evoluir o projeto para uma aplicação dinâmica com backend e banco de dados real em versões futuras.

5 CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo aplicar, de forma prática, os conhecimentos adquiridos na disciplina de Projeto e Desenvolvimento de Sistema Computacional com Persistência de Dados. Para isso, foi desenvolvido um sistema web denominado EasyRent, voltado à gestão de locação de veículos.

Durante o desenvolvimento do projeto, foram abordadas técnicas de modelagem orientada a objetos, estruturação de banco de dados relacional e prototipação de interfaces web responsivas. A elaboração de diagramas como o Diagrama de Classes UML e o Diagrama Entidade-Relacionamento (DER) possibilitou uma visão clara da estrutura lógica do sistema, favorecendo a compreensão das relações entre os dados.

A implementação das interfaces com HTML, CSS e JavaScript resultou em um protótipo funcional. Esse protótipo contempla as principais funcionalidades esperadas para um sistema de locadora de veículos, como o cadastro e listagem de marcas e carros. Além disso, foram implementadas as funcionalidades de login, cadastro de usuários e painel administrativo.

Embora o projeto ainda não esteja integrado a um sistema de gerenciamento de banco de dados (SGBD), a estrutura lógica desenvolvida já permite uma fácil adaptação para persistência real dos dados. Tecnologias como MySQL ou PostgreSQL podem ser utilizadas em futuras versões do sistema (MYSQL, 2025; POSTGRESQL, 2025).

Como sugestão para trabalhos futuros, a implementação da camada de persistência utilizando um backend em Node.js ou PHP, com conexão a um banco de dados relacional. Também seria possível expandir o sistema para contemplar funcionalidades como histórico de locações, relatórios administrativos e controle de disponibilidade dos veículos.

Em síntese, o desenvolvimento do EasyRent proporcionou uma experiência significativa de aplicação dos conceitos estudados em aula, reforçando a importância da modelagem e prototipação no ciclo de vida de sistemas computacionais. Confira o sistema completo em:

Schlorke (2025) – *EasyRent – Sistema Web para Locação de Veículos*.

Disponível em: <<https://github.com/schlorke/EasyRent>>. Acesso em: 22 maio 2025.

Referências

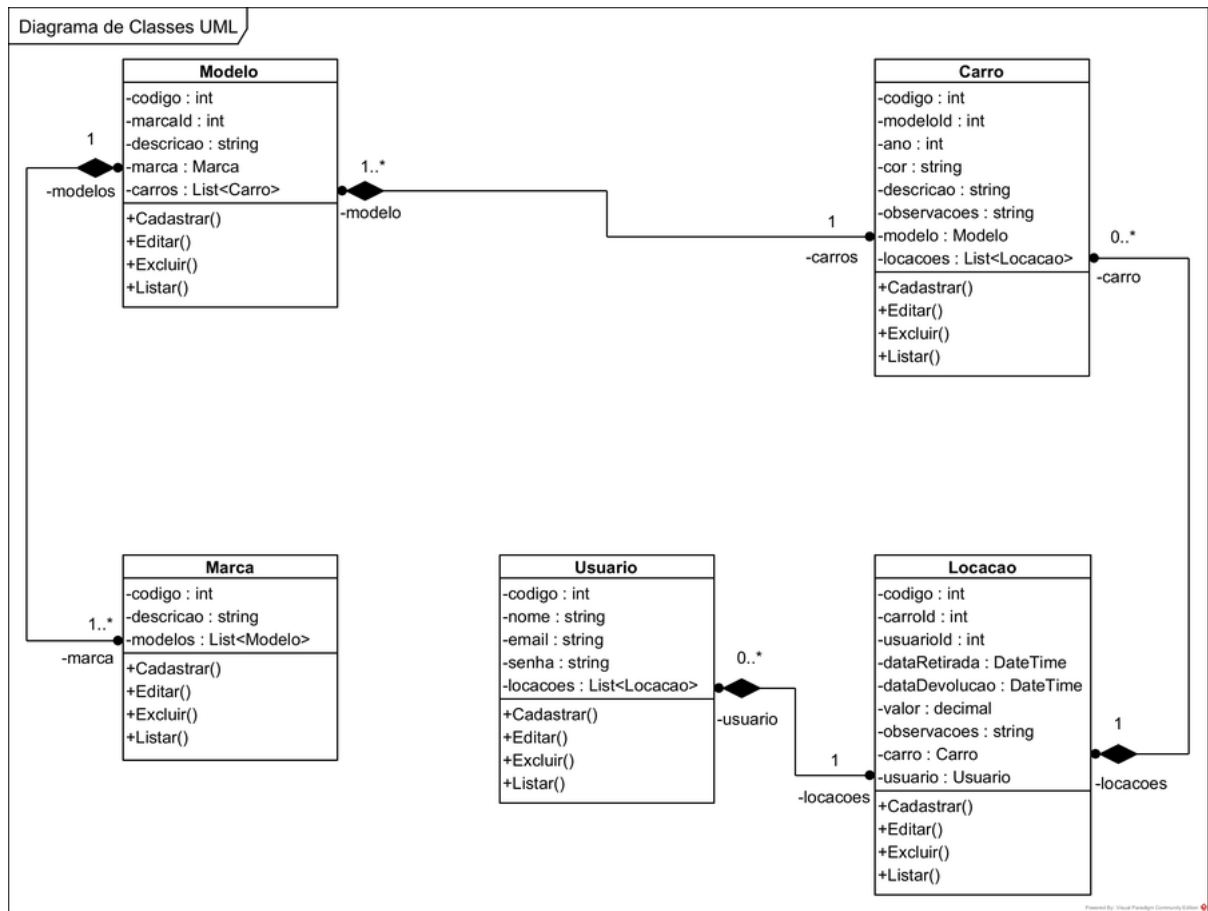
- DATE, C. J. **Introdução a Sistemas de Bancos de Dados**. 8. ed. Rio de Janeiro: Campus, 2004. Citado na página 13.
- GARRETT; JAMES, Jesse. **A experiência do usuário**: design centrado no usuário para a web. 2. ed. Rio de Janeiro: Alta Books, 2011. Citado na página 14.
- LARMAN; CRAIG. **Utilizando UML e padrões**. 3. ed. Porto Alegre: Bookman, 2007. Citado 3 vezes nas páginas 12, 13 e 22.
- MOZILLA FOUNDATION. **MDN Web Docs – HTML, CSS, JS**. 2025. Disponível em: <https://developer.mozilla.org>. Acesso em: 22/05/2025. Citado na página 12.
- MYSQL. **MySQL 8.3 Reference Manual**. 2025. Disponível em: <https://dev.mysql.com/doc>. Acesso em: 22/05/2025. Citado 2 vezes nas páginas 14 e 23.
- POSTGRESQL. **The PostgreSQL Global Development Group**. 2025. Disponível em: <https://www.postgresql.org>. Acesso em: 22/05/2025. Citado 2 vezes nas páginas 14 e 23.
- PRESSMAN, Roger S; MAXIM, Bruce R. **Engenharia de Software**. 8. ed. Porto Alegre: AMGH, 2016. Citado na página 13.
- SCHLORKE, Harry Ciciliani. **EasyRent – Código-fonte no GitHub**. 2025. Repositório. Disponível em: <https://github.com/Schlorke/EasyRent>. Acesso em: 22/05/2025. Citado na página 23.
- SOMMERVILLE, Ian. **Engenharia de Software**. 10. ed. São Paulo: Pearson Education, 2019. Citado na página 12.
- UNIVERSIDADE DO VALE DO ITAJAÍ. **Projeto Lite is Cool – UNIVALI**. 2025. Projeto institucional disponível online. Disponível em: <https://www.univali.br>. Acesso em: 22/05/2025. Citado na página 12.
- VISUAL PARADIGM. **UML Diagrams**. 2025. Documentação de ferramenta online. Disponível em: <https://www.visual-paradigm.com>. Acesso em: 22/05/2025. Citado na página 15.

Apêndices

APÊNDICE A – Diagrama de Classes UML do Sistema EasyRent

Este diagrama apresenta a modelagem orientada a objetos do sistema EasyRent, representando as classes Marca e Carro, seus respectivos atributos e métodos. Foi elaborado com base nos princípios da Programação Orientada a Objetos (POO), por meio da ferramenta Visual Paradigm.

Figura 5 – Diagrama de Classes UML do Sistema EasyRent

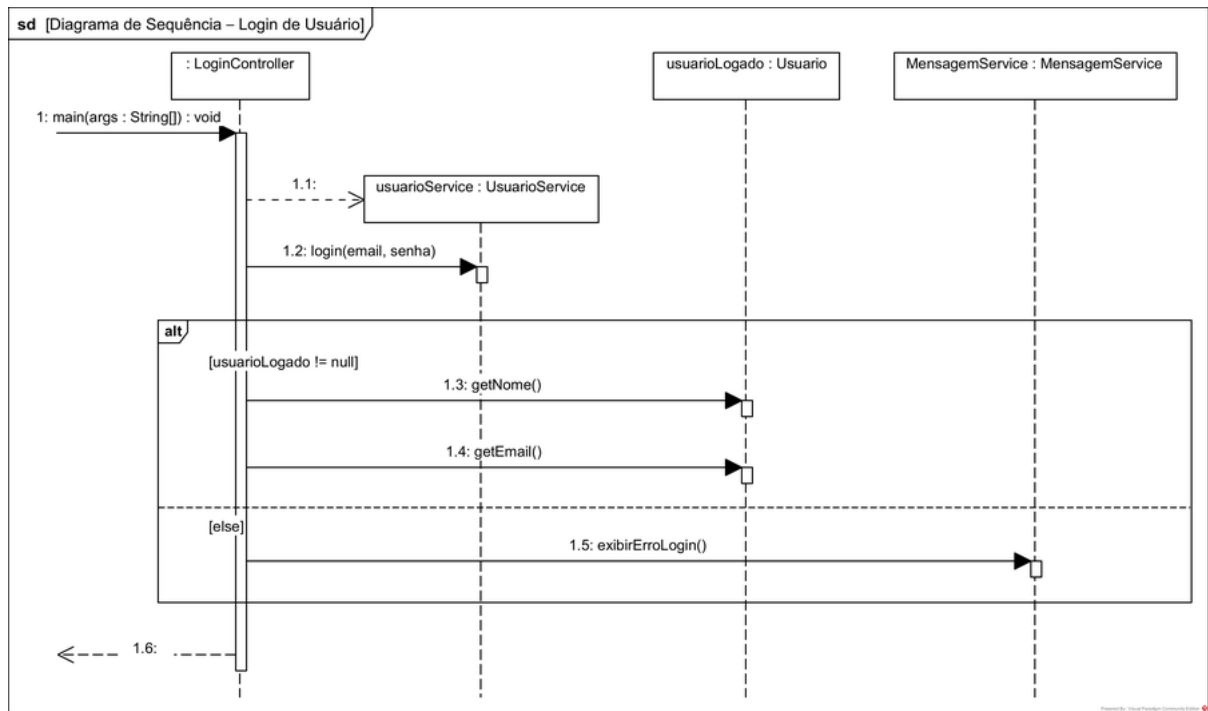


Fonte: Elaborado pelo Harry Schlorke (2025)

APÊNDICE B – Diagrama de Sequência: Login de Usuário

Este diagrama representa o processo de autenticação de usuários no sistema EasyRent. A modelagem mostra a sequência de interações entre o usuário, a interface do sistema e o componente responsável pela verificação de credenciais. O objetivo é garantir acesso seguro à plataforma e permitir a navegação apenas por usuários autorizados.

Figura 6 – Diagrama de Sequência UML do Login de Usuário no Sistema EasyRent

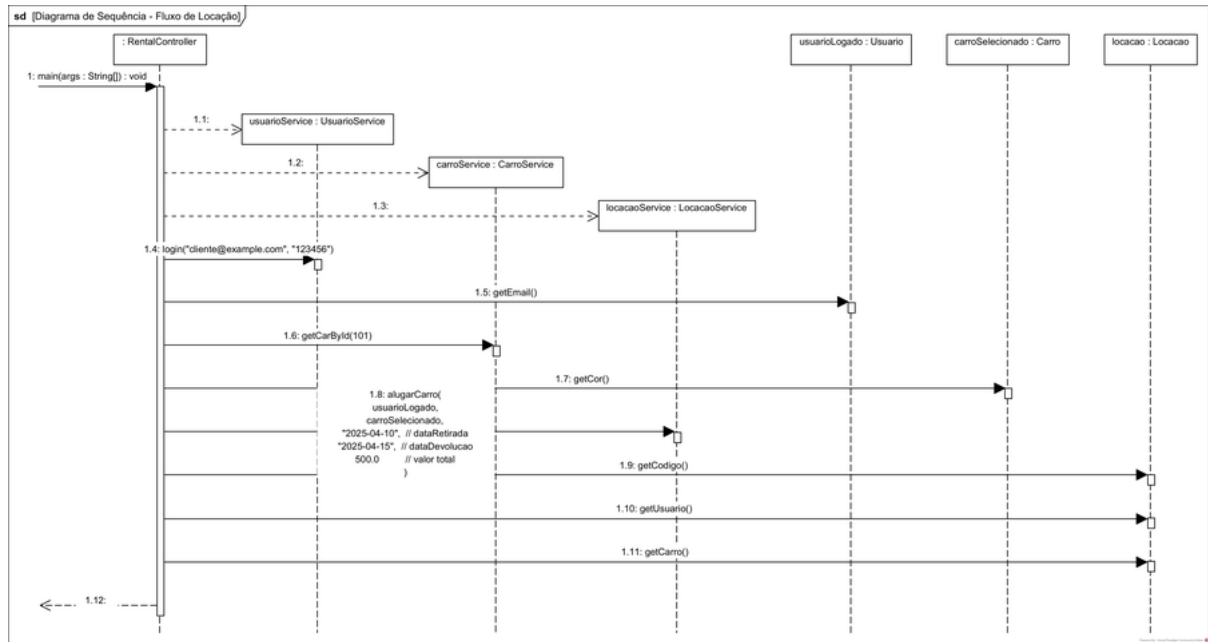


Fonte: Elaborado pelo Harry Schlorke (2025)

APÊNDICE C – Diagrama de Sequência: Fluxo de Locação

O diagrama a seguir descreve a sequência de interações para o processo de locação de veículos no sistema EasyRent. Ele demonstra como o usuário realiza a busca, seleciona um veículo e finaliza a locação. Esse fluxo assegura que as informações sejam tratadas corretamente e que o sistema responda com os dados esperados em cada etapa da locação.

Figura 7 – Diagrama de Sequência UML do Fluxo de Locação no Sistema EasyRent



Fonte: Elaborado pelo Harry Schlorke (2025)

APÊNDICE D – Modelo e Diagrama Entidade-Relacionamento (MER/DER) do Sistema EasyRent

O modelo de dados do sistema EasyRent foi construído com base no Modelo Entidade-Relacionamento (MER), que representa a estrutura conceitual do banco de dados. Este modelo descreve logicamente as entidades (como Usuário, Marca, Modelo, Carro e Locação), seus atributos e os relacionamentos entre elas, servindo como base para a modelagem relacional.

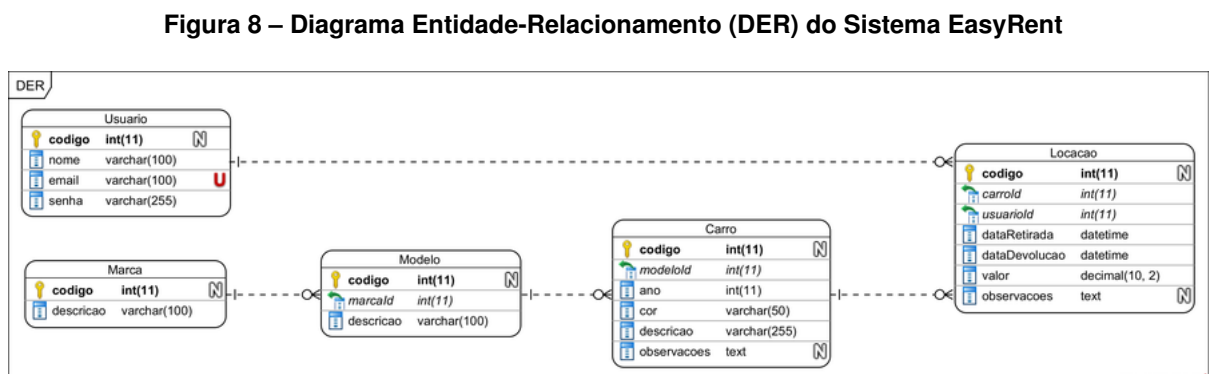
A partir desse modelo conceitual, foi elaborado o Diagrama Entidade-Relacionamento (DER), que representa de forma lógica e gráfica a estrutura do banco de dados. No DER, as entidades são representadas como tabelas com seus respectivos atributos, incluindo tipos de dados, chaves primárias e estrangeiras, e os relacionamentos são indicados por linhas que conectam essas tabelas. Esse formato facilita a análise e o entendimento do modelo lógico do sistema.

A estrutura representada no DER define:

- As chaves primárias e chaves estrangeiras.
- As cardinalidades dos relacionamentos.
- Os tipos de dados e vínculos entre tabelas.

Essa modelagem garante a integridade referencial e possibilita a futura implementação do sistema em um SGBD relacional, como MySQL ou PostgreSQL, com base sólida em boas práticas de modelagem de dados.

Figura 8 – Diagrama Entidade-Relacionamento (DER) do Sistema EasyRent



Fonte: Elaborado pelo Harry Schlorke (2025)