

Laboratório de Sistema de Software


Andrei Pochmann Koenich

Rafael Nascimento Leite



BikeMatch

- **Ideia principal: viabilizar, por meio de uma aplicação web, que determinados usuários (locadores) possam disponibilizar suas bicicletas para aluguel por outros usuários (locatários).**



Objetivos da Aplicação Desenvolvida



Objetivos da Aplicação Desenvolvida

- **Possibilitar a obtenção de renda extra por parte dos locadores;**
- **Ofertar uma nova opção de deslocamento aos necessitados;**



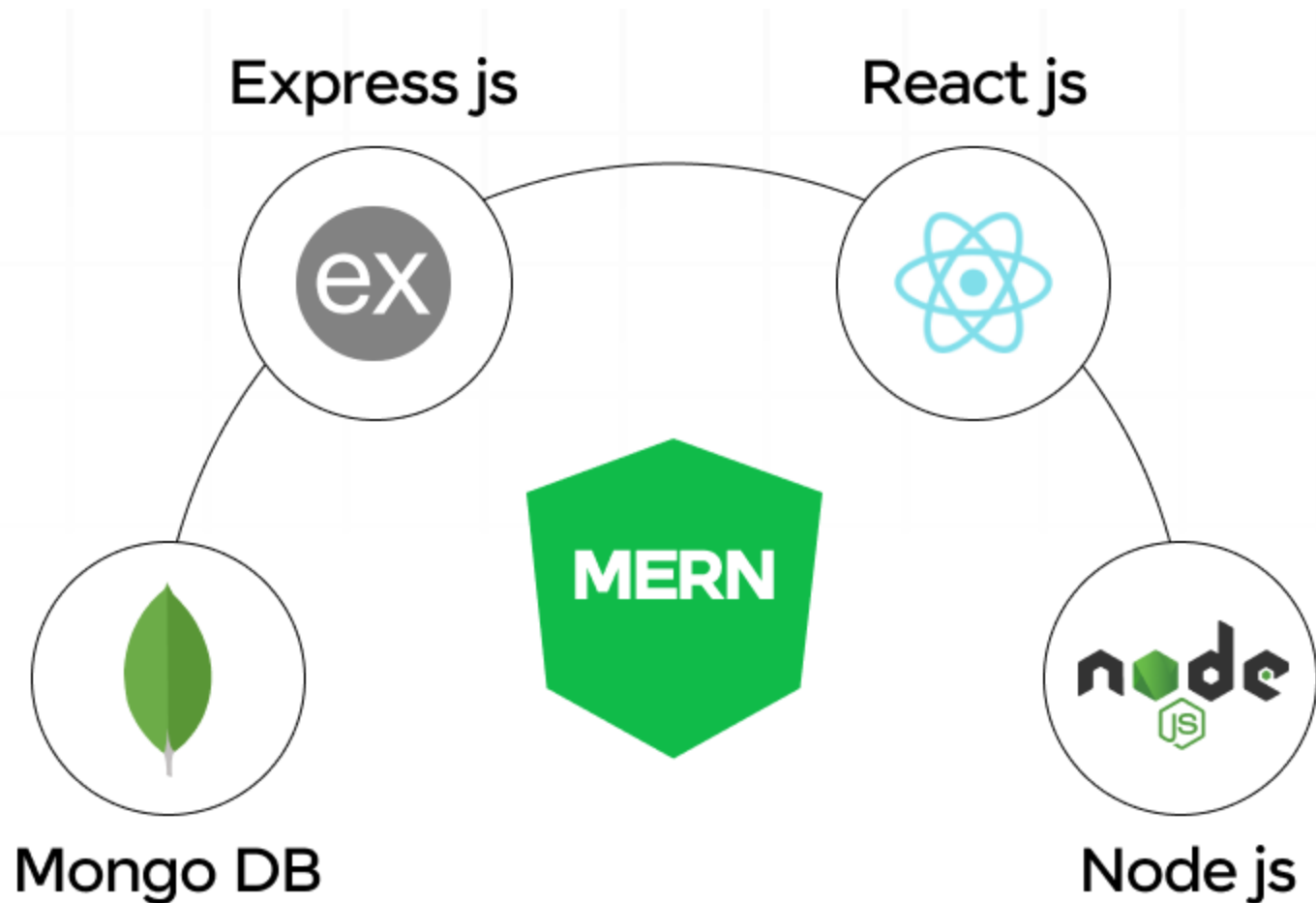
Objetivos da Aplicação Desenvolvida

- **Contribuir para redução no tráfego nos grandes centros urbanos;**
- **Incentivar a prática dos exercícios físicos relacionados ao ciclismo.**



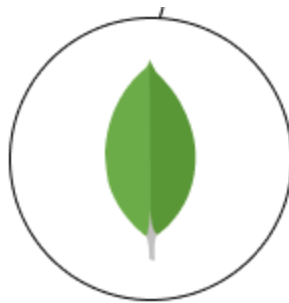
Tecnologias Utilizadas

Tecnologias utilizadas



Tecnologias utilizadas - MongoDB

- Banco de dados não-relacional, utilizado para armazenar dados de forma flexível (*JSON-like*);
- Suporta eventuais grandes volumes de dados no sistema, com alta escalabilidade horizontal;
- Modelagem de dados simples (*user, bike e rent*).



Mongo DB



Tecnologias utilizadas - Node.js

- Ambiente de execução do *Javascript*;
- Oferece desempenho e escalabilidade (conexões simultâneas, operações assíncronas);
- Ecossistema rico, com vasta coleção de pacotes *node package manager*.



Node js

Tecnologias utilizadas - Express.js

- **Framework de back-end:** simplifica a criação de servidores e *APIs* com *Node.js*;
- Definição de rotas, processamento de requisições e respostas HTTP;
- Suporte robusto para utilização de *middlewares*.

Express js

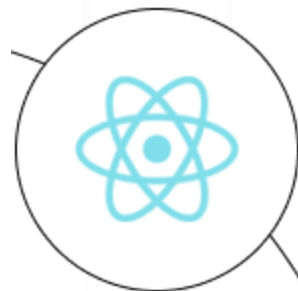




Tecnologias utilizadas - React.js

- Desenvolvimento de interfaces de usuário interativas e dinâmicas para o *front-end*;
- Componentização e reutilização: facilidade para manutenção e expansão;
- Atualização eficiente via *DOM*.


React js





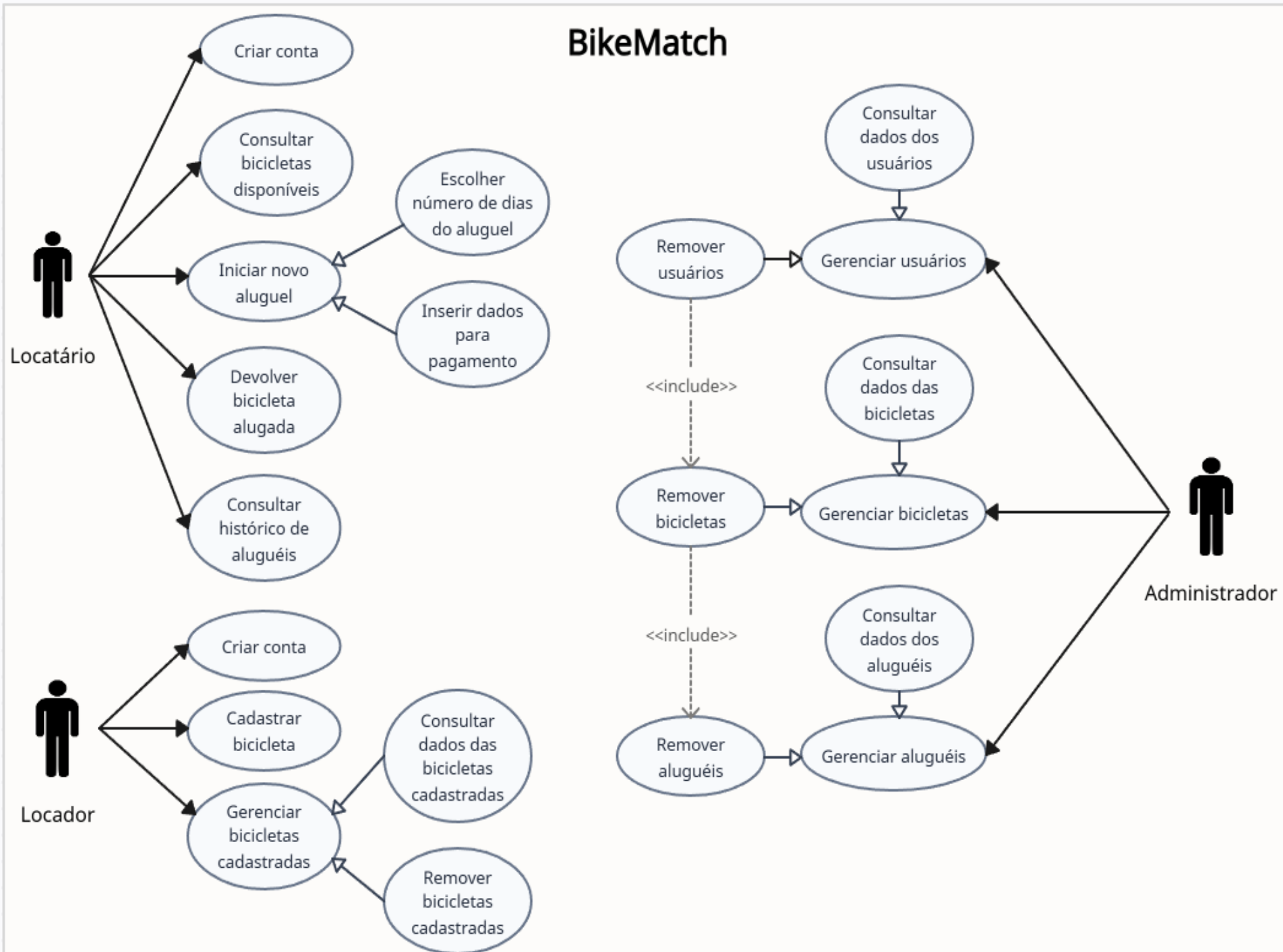
Outras tecnologias utilizadas

- ***Axios***: comunicação entre *front-end* e *back-end*;
- ***Node packet manager (npm)***: instalação, desinstalação e atualização de dependências do projeto;
- ***Joi***: biblioteca *Javascript* para validação de dados (cadastro dos usuários);
- ***Docker***: garantia de portabilidade;
- ***JSON Web tokens***: padrão para autenticação de dados de forma segura (*login* dos usuários).



Funcionamento do Sistema: Casos de Uso

BikeMatch





Arquitetura MSC

(Model-Service-Controller)



Arquitetura *Model-Service-Controller*

- ***Model***: representa os objetos e entidades do domínio, com interação direta com o banco de dados.
- ***Controller***: recebe requisições do cliente e retorna as respostas. Chama as operações de serviço;
- ***Service***: encapsula a lógica de aplicação e interage com o Modelo para acessar os dados desejados.



Processo utilizado: *Scrumban*



Processo utilizado: *Scrumban*

- Benefícios estruturados do Scrum junto à flexibilidade do Kanban:
- *Foram realizadas iterações (ciclos de trabalho fixos e repetitivos), com flexibilidade no planejamento;*
- *Priorização dinâmica: priorização e reorganização das tarefas de forma contínua, adaptando-se aos novos requisitos percebidos ao longo do desenvolvimento.*



Conclusões

- **Quantidade de novas funcionalidades possíveis de serem implementadas é "infinita". Entretanto, as principais funcionalidades do sistema foram corretamente implementadas, garantindo que as transações do sistema possam ocorrer;**



Conclusões

- O sistema desenvolvido possui modularidade suficiente para ser colocado em prática no mercado real, mediante a implementação de novas funcionalidades, tais como:
- *mecanismos robustos, realistas e seguros relacionados com transações financeiras;*
- *possibilidade de contato entre os usuários e os administradores do sistema;*
- *suporte técnico aos usuários, tornando a aplicação segura do ponto de vista legal.*

Laboratório de Sistema de Software

Andrei Pochmann Koenich

Rafael Nascimento Leite



BikeMatch