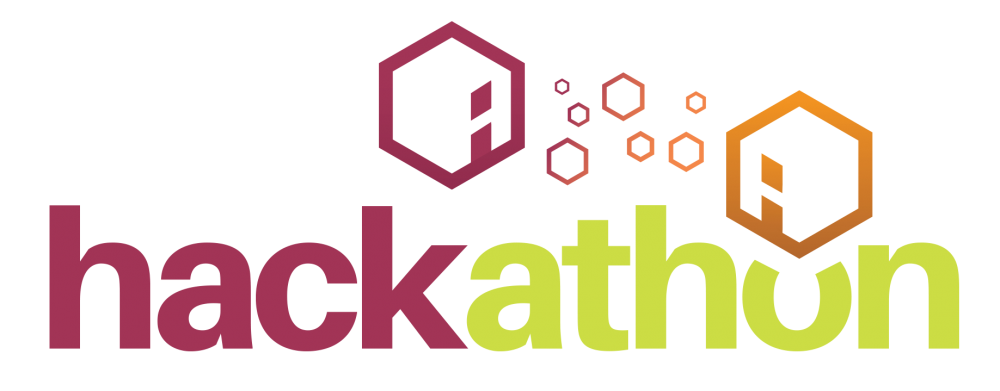


**Hackthon**

*Documentação* *Hackthon*



Alunos:

Breno Labs So ares da Cruz *(RA: 14000)*

João Felipe Bacarin da Silva *(RA: 13979)*

Guilherme Fernandes de Souza *(RA: 14539)*

Gabriel Pereira Henrique Gomes *(RA: 13606)*

Maria Eduarda Baia Guirelli *(RA: 14524)*

# Index

Visão Geral do Sistema de Saúde para Idosos

Nosso sistema de saúde para idosos foi projetado para enfrentar os desafios únicos que essa população enfrenta, especialmente em relação à vacinação e cuidados preventivos. Com uma abordagem centrada na acessibilidade e praticidade, o software oferece um registro completo de saúde, incluindo histórico médico detalhado e informações sobre vacinas recomendadas. Facilita o agendamento de visitas de profissionais de saúde através de uma aplicação, permitindo que familiares e cuidadores programem as vacinações de acordo com a conveniência do idoso. Além disso, o sistema envia alertas automáticos sobre datas de vacinação e fornece informações educativas regulares sobre imunização, abordando mitos comuns e promovendo a importância da vacinação para a saúde dos idosos. Com isso, nosso objetivo é melhorar significativamente a adesão às vacinas essenciais, proporcionando um cuidado de saúde preventivo e personalizado, adaptado às necessidades específicas dos idosos, independentemente de sua localização ou mobilidade.

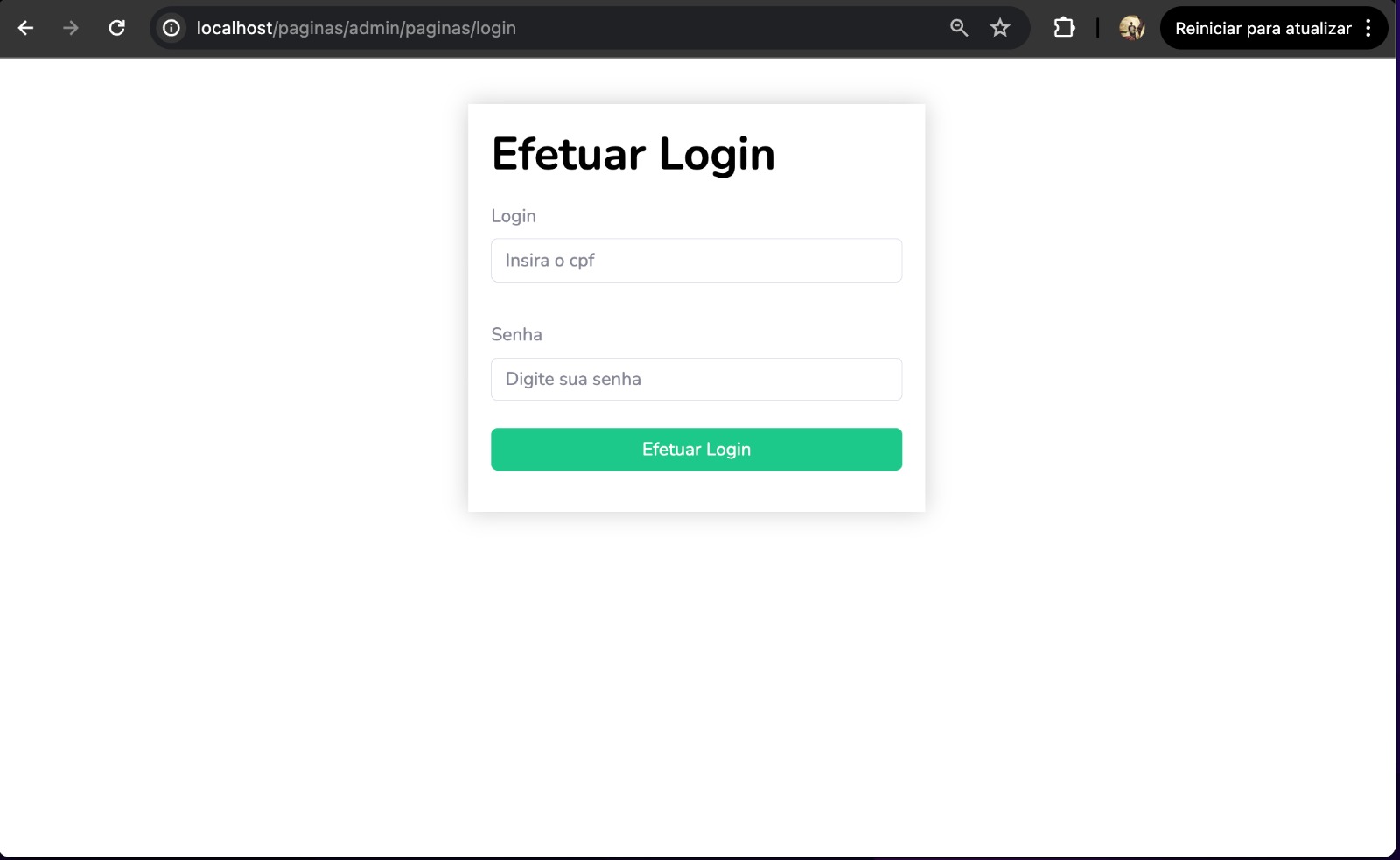
# Sequência Lógica do Software

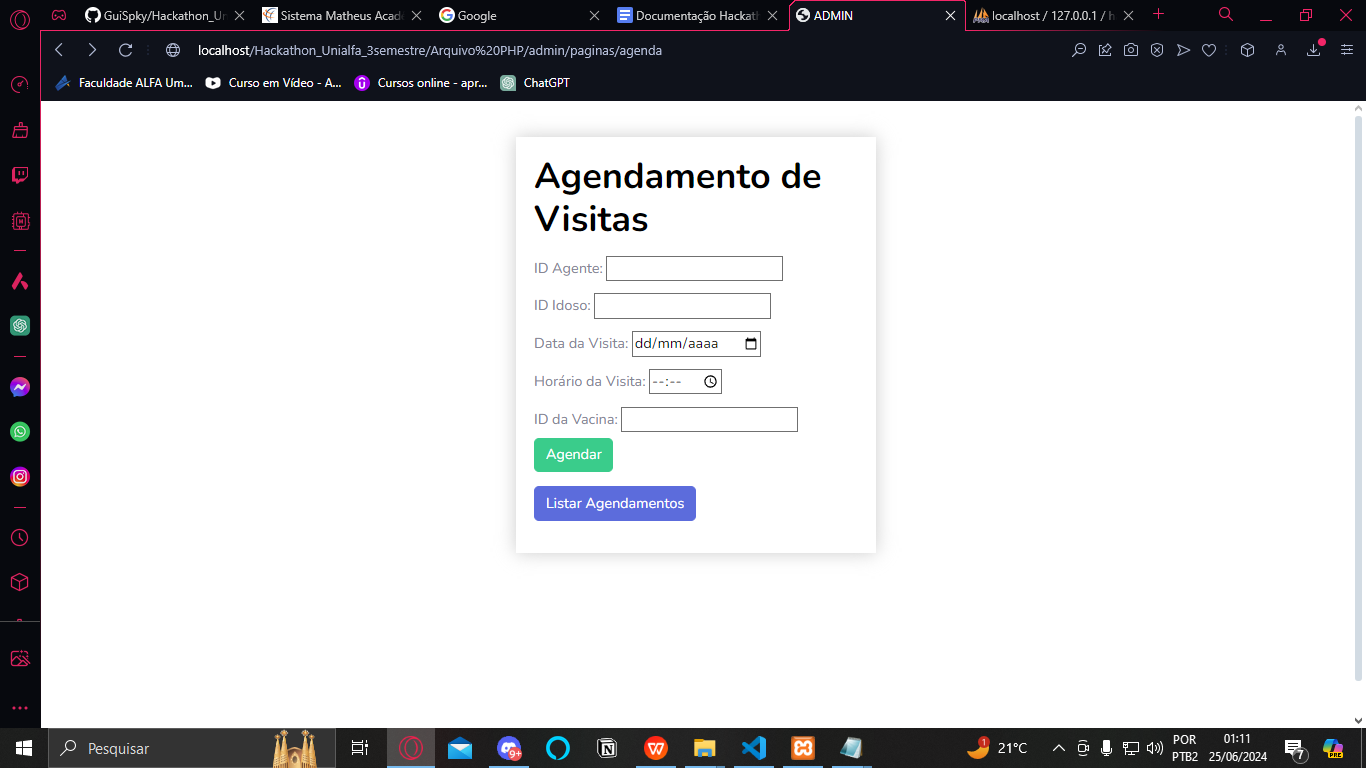
Este software contará com tecnologias da linguagem de: PHP; Node.js; Banco de dados relacional SQL e Java.

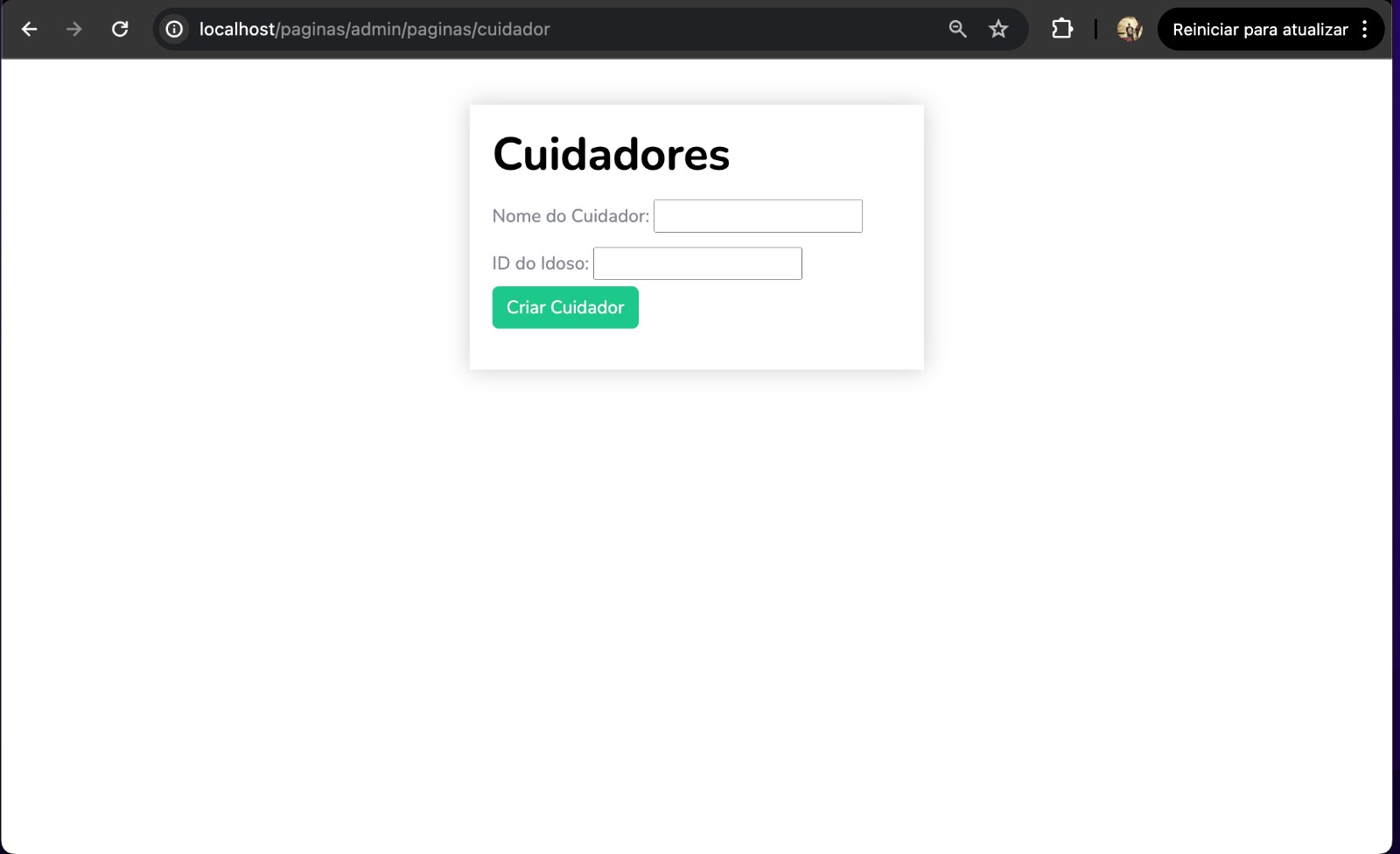
*PHP*

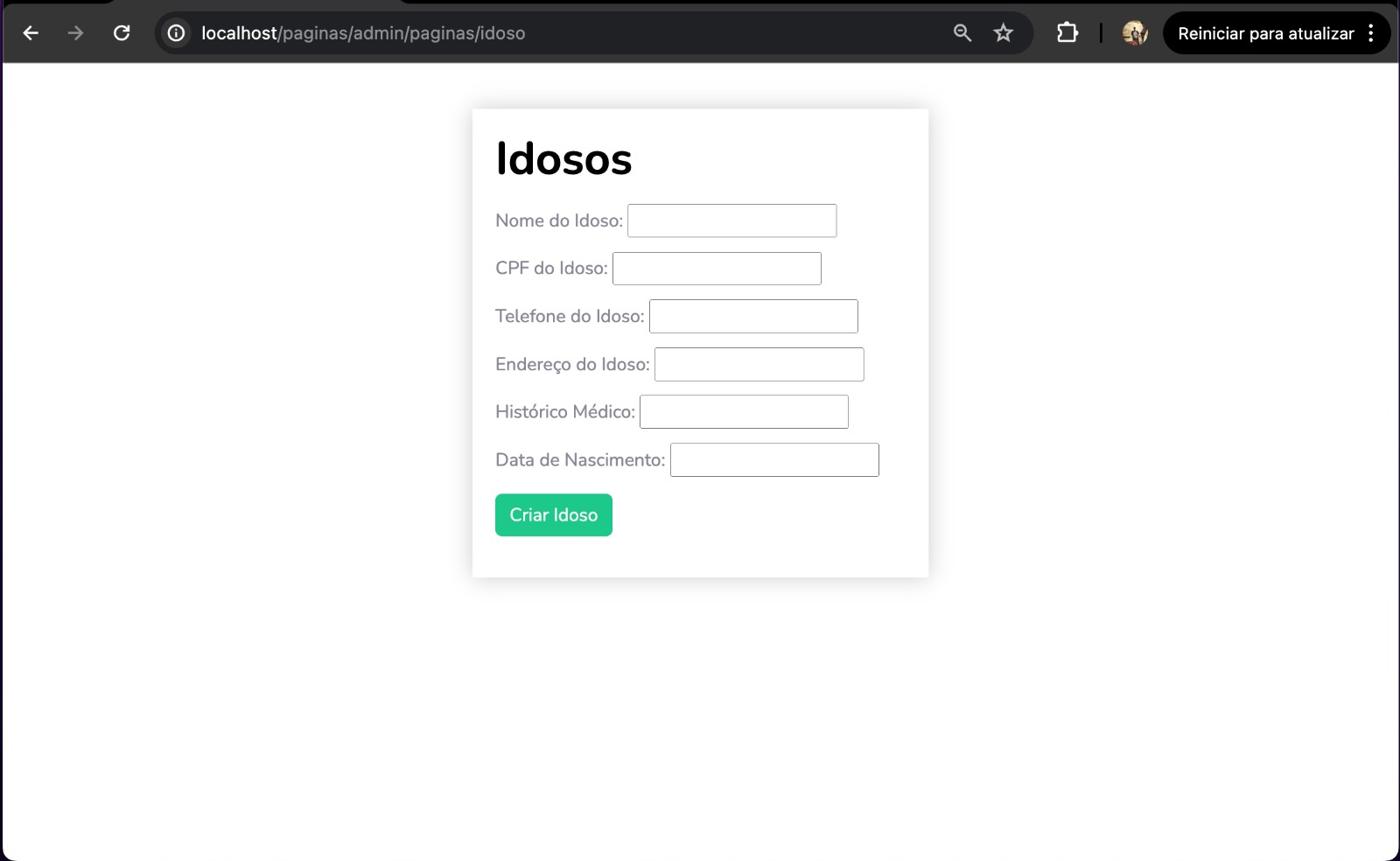
Na parte do desenvolvimento em PHP, foi feito a criação das páginas com estilização básica, elaborando os arquivos correspondentes as tabelas onde o foco foi de conexão com a API, foram feitas execuções do node e o funcionamento da porta, a mesma usada na URL dos arquivos PHP.

Abaixo, as telas correspondentes:









*API Node.js*

Src/Server.ts: Faz a criação do server localhost, definido na porta 3003, ficando com o endereço: “http://localhost:3003/” para o acesso ao server.

Knexfile.js: executa a conexão da API com o banco de dados MySql hackathon. src/database/knex/index.ts. Cria a variável conexão onde estão as configurações do knex e do banco de dados.

Src/routers/index: Arquivo responsável por gerenciar e criar as rotas e as URL´s para todas as rotas (idoso, cuidador e agenda). Foi criado as rotas de tipo “get”, “post”, “put” e “delete”. Todas as rotas utilizam o knex para fazer a a conexão ao banco e passar ou obter os valores.

Get, busca os dados e gera um arquivo .json com os dados.

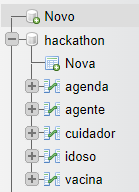
Post, utiliza um arquivo .json para criar novos objetos nas tabelas.

Put, utiliza um arquivo .json para atualizar dados que ja estão cadastrados.

Delete, recebe dados pelo “param” da url e deleta dados indicados do banco.

*Banco relacional MySQL*

Na etapa da produção do banco de dados relacional precisávamos receber via api das demais linguagens as informações dos agentes de saúde, os registros de saúde os idosos e o agendamento de visitas para os mesmos. Segue abaixo as tabelas e seus demais atributos:

Tabelas:

Atributos das tabelas, respectivamente:

Tabela agenda:



Tabela agente:

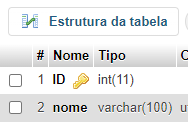


Tabela cuidador:

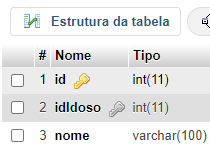


Tabela idoso:

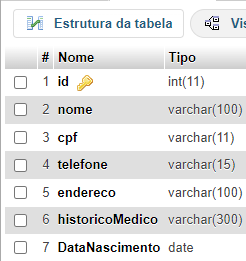
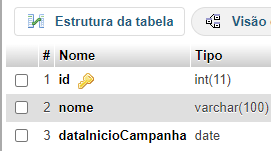


Tabela vacina:



*JAVA*

A função do Java no sistema é encabeçar o gerenciamento dos cadastros e a manipulação dos mesmos por meio de listagens. Abaixo, segue as telas correspondentes:

