APRESENTAÇÃO
SISTEMAS DISTRIBUIDOS

SISTEMA DE GESTÃO DE EVENTOS

Professor: João Ferreira
Instituição: Uninassau-Graças

Equipe

Diego Lima - 01401412 - Documentador

Gilvanelson Nascimento - 01395387 - Scrum Master

Alesson Calaça - 01378540 - Desenvolvedor Front-End

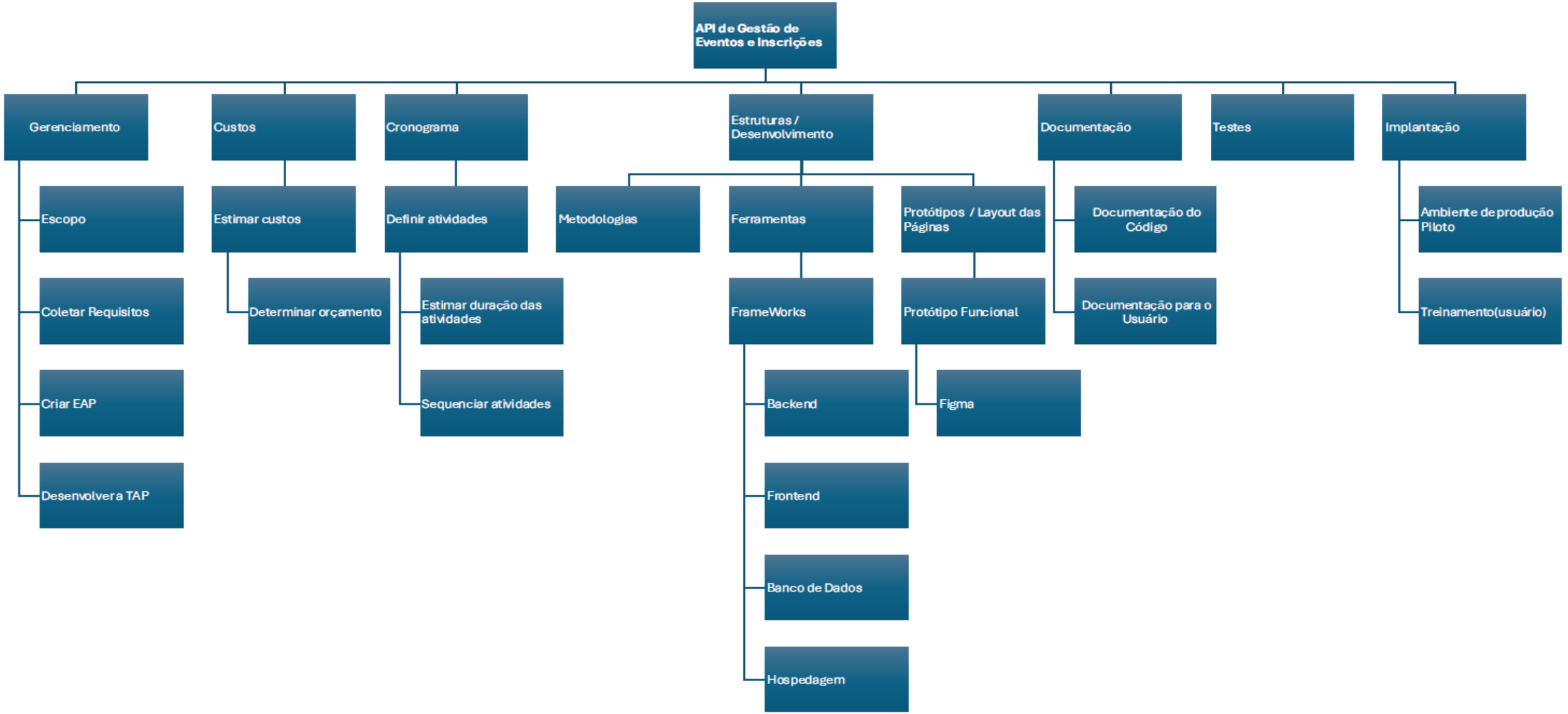
Luiz Pereira 01170935 - Desenvolvedor Back-End

Indice

Descrição do Sistema

A API Mega Eventos é uma solução eficiente e flexível, permitindo o gerenciamento completo de atividades e interações relacionadas a eventos. Os usuários podem explorar a lista de eventos disponíveis, visualizar informações detalhadas, realizar inscrições de forma prática e receber notificações personalizadas sobre atualizações ou mudanças nas atividades. Projetada para atender a diversas organizações, como instituições religiosas, ONGs e empresas, a API é altamente escalável e focada na experiência do usuário, integrando comunicação em tempo real e relatórios para facilitar a administração de eventos.

Estrutura Analítica do Projeto



Análise de Custos

ID:	Descrição da tarefa:	Fase:	Recurso:
1	Elaboração do Diagrama de Casos de Usos	Planejamento	Analise de Sistemas
2	Design da Interface do Usuário	Design	Designer UX/UI
3	Desenvolvimento do Back-And com C# e.Net	Desenvolvimento	Desenvolvedor Back-And
4	Desenvolvimento do Front-And com React	Desenvolvimento	Desenvolvedor Front-And
5	Configuração do Banco de dados MYSQL	Implementacão	DBA
6	Hospedagem na AWS / Cloud Flare	Implementacão	Cloud Enginier

Tabela de Custos			
ID:	**Descrição da tarefa:**	**Fase:**	**Recurso:**
1	*Elaboração do Diagrama de Casos de Usos*	*Planejamento*	*Analise de Sistemas*
2	Design da Interface do Usuário	Design	Designer UX/UI
3	Desenvolvimento do Back-And com C# e.Net	Desenvolvimento	Desenvolvedor Back-And
4	Desenvolvimento do Front-And com React	Desenvolvimento	Desenvolvedor Front-And
5	Configuração do |Banco de dados MYSQL	Implementação	DBA
6	Hospedagem na AWS / Cloud Flare	Implementação	Cloud Enginier

ID:	**Quantidade de Horas:**	**Valor Unitário**	**Valor Total:**
1	40	*R$ 100.00*	*R$ 4,000.00*
2	80	R$ 80.00	*R$ 6,400.00*
3	320	R$ 120.00	*R$ 38,400.00*
4	240	R$ 100.00	*R$ 24,000.00*
5	40	R$ 100.00	*R$ 4,000.00*
6	8	R$ 150.00	*R$ 1,200.00*

ID:	**Porcentagem do Imposo**	**Valor do Imposto**	**Valor Total com Imposto**
1	20.00%	R$ 800.00	R$ 4,800.00
2	20.00%	R$ 1,280.00	R$ 7,680.00
3	20.00%	R$ 7,680.00	R$ 46,080.00
4	20.00%	R$ 4,800.00	R$ 28,800.00
5	20.00%	R$ 800.00	R$ 4,800.00
6	20.00%	R$ 240.00	R$ 1,440.00

ID:	Data Prevista:	Data Realizada	Status:
1	06/09/24	15/09/24	Concluída
2	16/09/24	30/09/24	Em Andamento
3	01/11/24		Em Andamento
4	01/11/24		Em Andamento
5	15/11/24		Aguardando Serviço
6	20/11/24		Instanciando Servidores EC2

ID:	Observações:
1	Detalhando as Funcionalidades do Sistema
2	Utilizando Figma
3	Ultilizando Visual Studio e .Net Core
4	Ultilizando Creat React App
5	
6	

Cronograma do Projeto

1. Elaboração do Diagrama de Casos de Usos

   Concluída - Data Realizada: 15/09/24

2. Design da Interface do Usuário

   Em Andamento - Data Prevista: 30/09/24

3. Desenvolvimento do Back-End

   Em Andamento - Data Prevista: 01/11/24

4. Desenvolvimento do Front-End

   Em Andamento - Data Prevista: 01/11/24

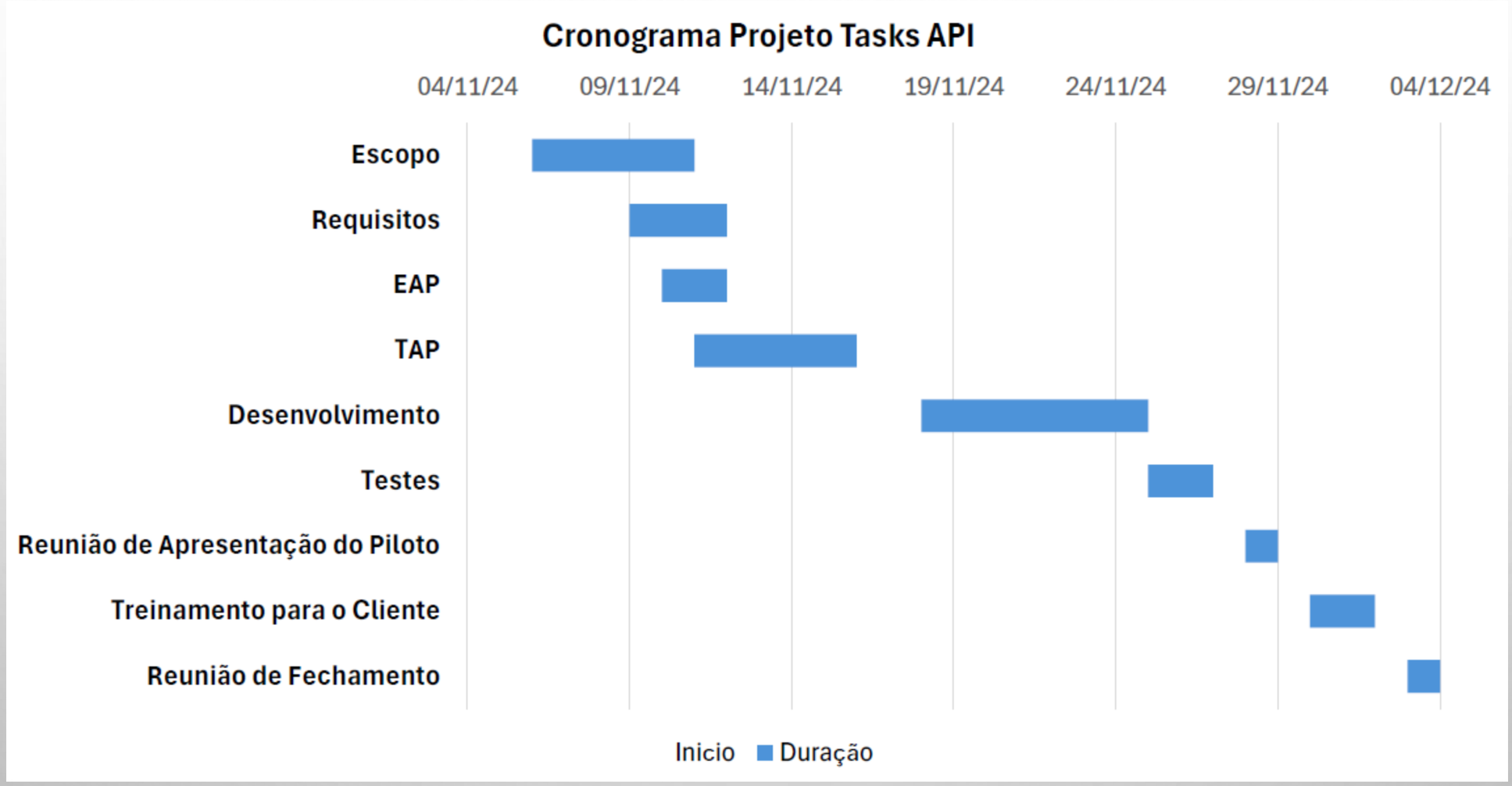
5. Configuração do Banco de Dados

   Aguardando Serviço - Data Prevista: 15/11/24

6. Hospedagem na AWS / Cloud Flare

   Instanciando Servidores EC2 - Data Prevista: 20/11/24

Cronograma do Projeto



Ferramentas e Tecnologias Utilizadas

Figma

Utilizado para o design da interface do usuário.

Visual Studio e .Net Core

Utilizados para o desenvolvimento do back-end.

Create React App

Utilizado para o desenvolvimento do front-end.

MySQL

Utilizado como banco de dados para armazenar informações do sistema.

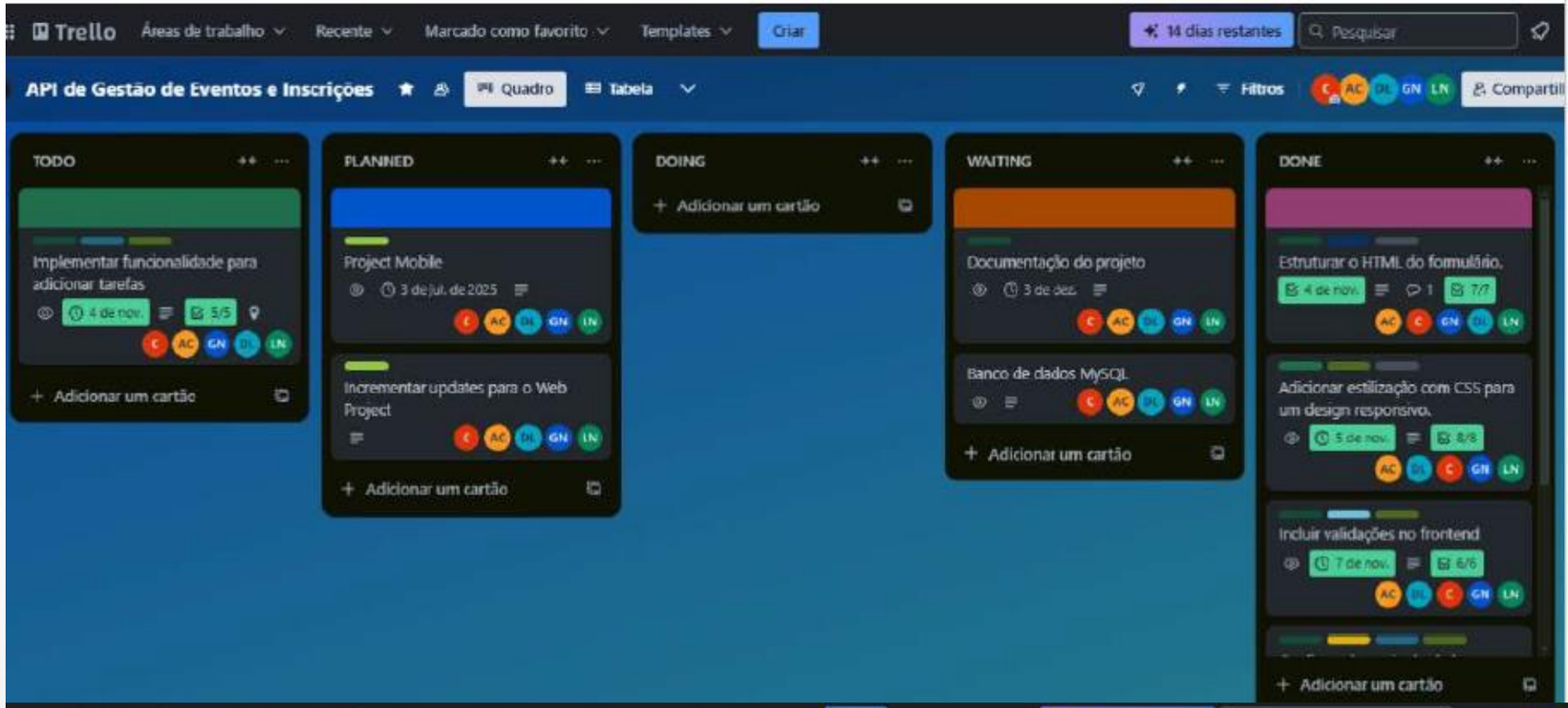
AWS / Cloud Flare

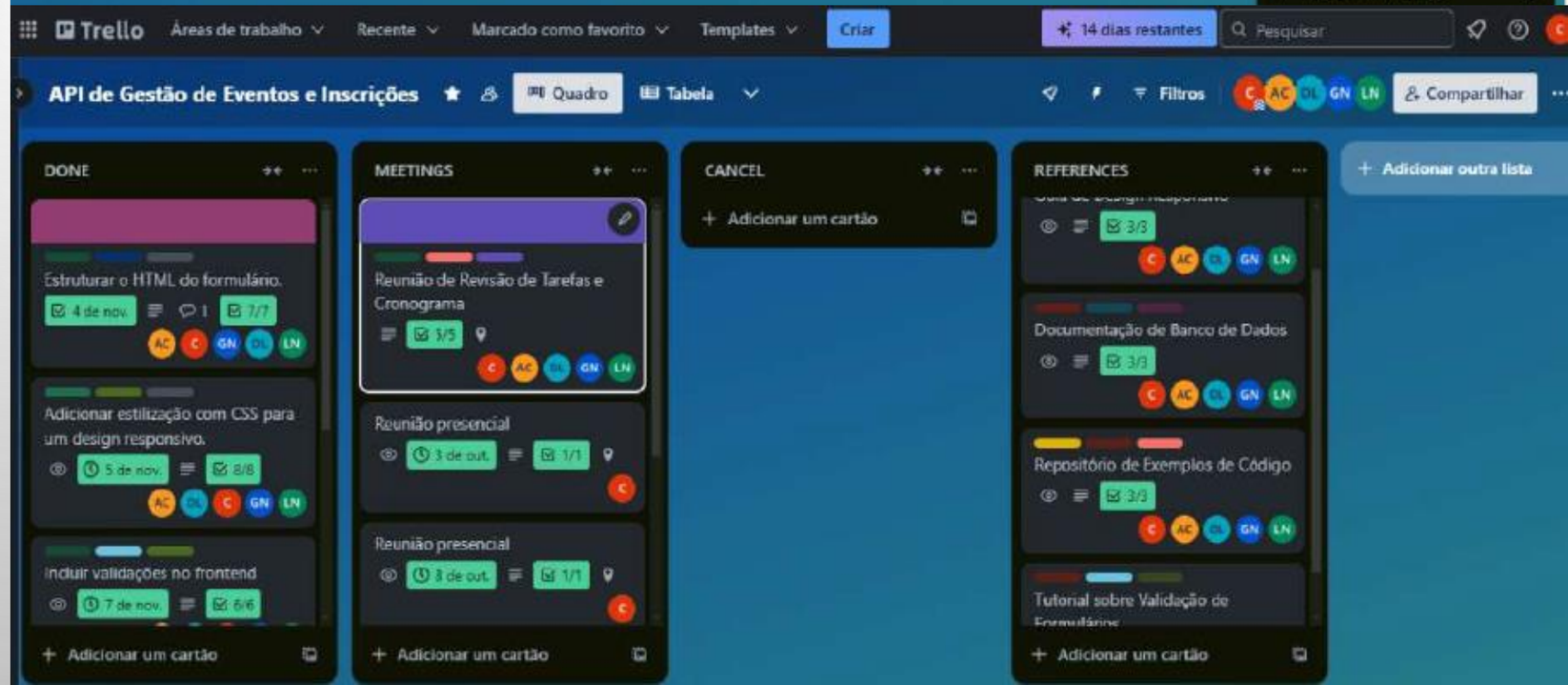
Utilizados para hospedagem e infraestrutura do sistema.

Resumo do Trello

Trello

Utilizamos para gerenciar as tarefas do projeto, acompanhar o progresso e facilitar a comunicação entre os membros da equipe.



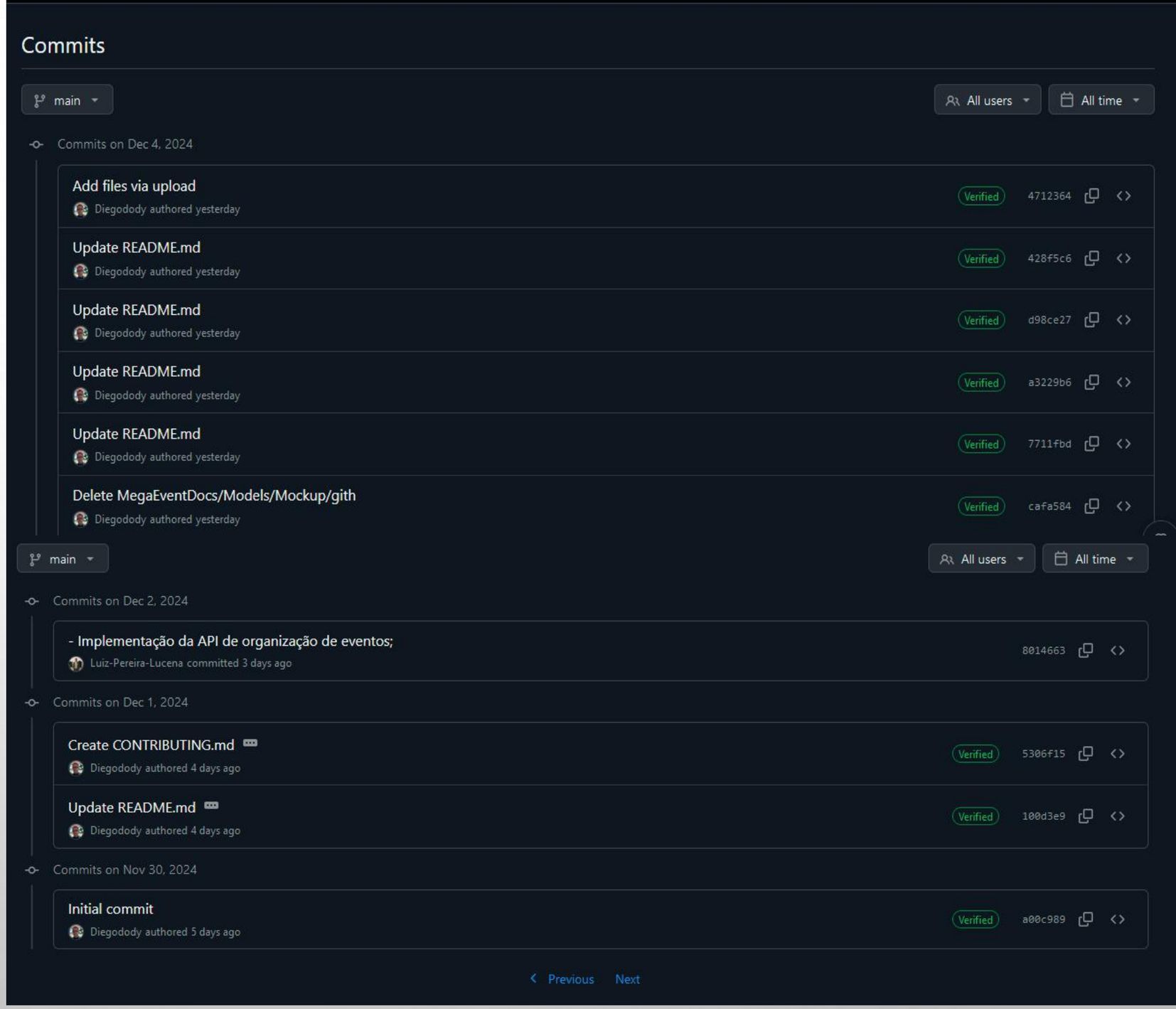


Github Commits

Github

Utilizamos o github para desenvolver toda produção do projeto e atualizar sempre as mudanças, e torna algo acessível a todos da equipe.

Link: https://github.com/Diegodody/MegaEventApi



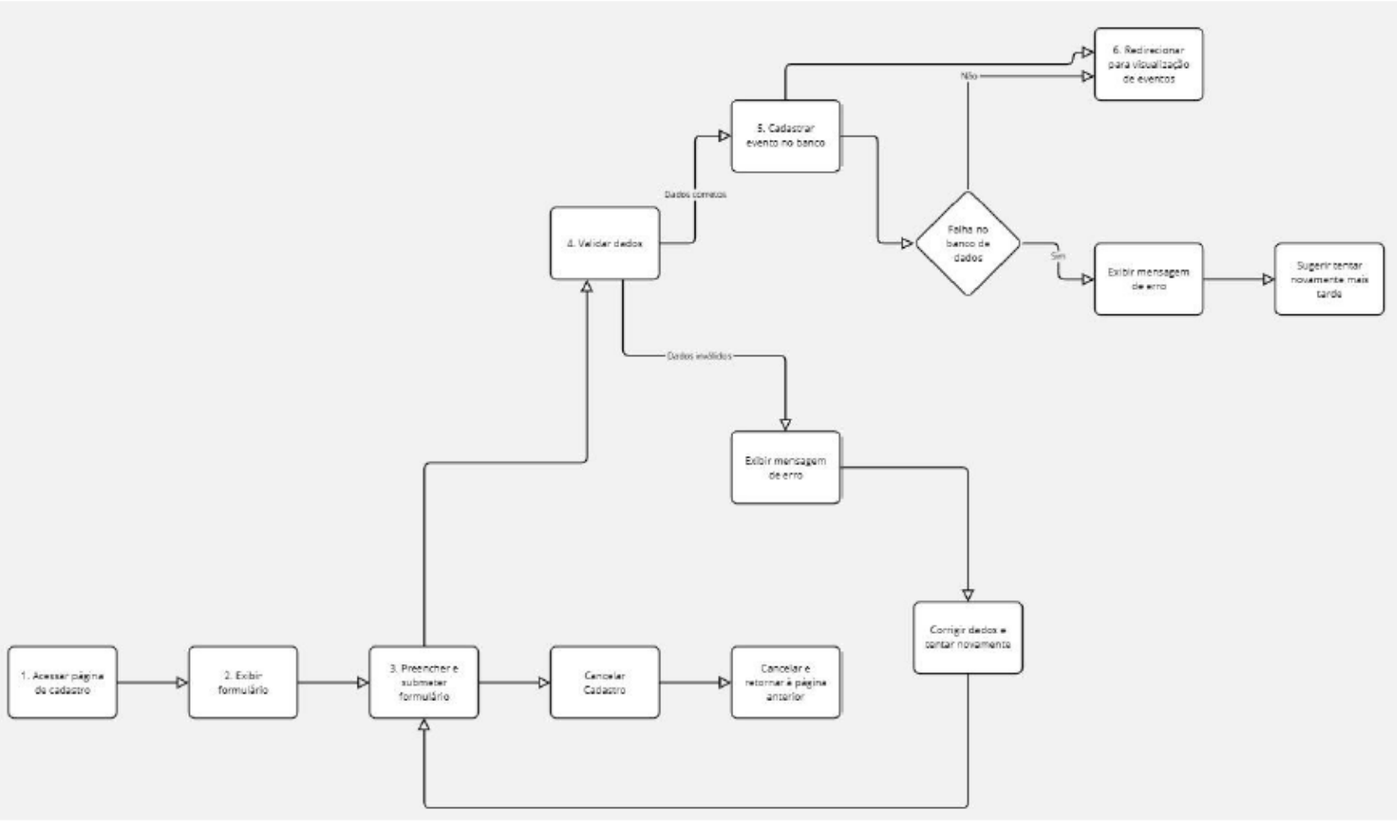
Caso de Uso: Cadastrar Evento

O Caso de Uso UC721 “Cadastrar Evento” permite que o administrador cadastre um novo evento no sistema, com fluxos para cadastro bem-sucedido, cancelamento e tratamento de erros.

Projeto 3: API de Gestão de Eventos e Inscrições

UC721 - Cadastrar Evento

ID	UC721
Título	Cadastrar Evento
Ator principal	Administrador
Objetivo	Permitir que o administrador cadastre um novo evento no sistema.
Pré-condição	O administrador deve estar autenticado no sistema.
Pós-condição	O evento será salvo no banco de dados e poderá ser visualizado na lista de eventos.
Diagrama	**[Fluxograma]**



Fluxos do Caso de Uso: Visualizador de Evento

1 Fluxo Principal

O administrador acessa a página de cadastro, preenche o formulário e salva o evento no sistema.

2 Fluxo Opcional

O administrador pode cancelar o processo de cadastro e retornar à p anterior.

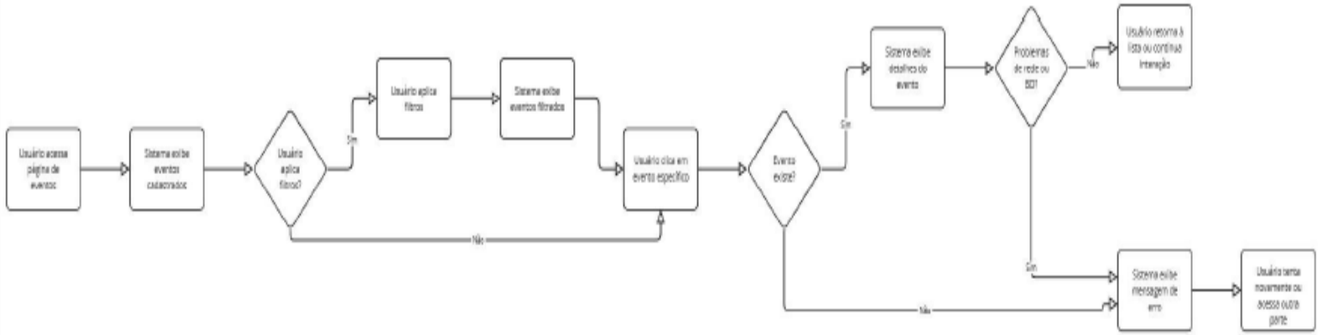
3 Fluxo Alternativo

O sistema exibe mensagem de erro caso os dados informados sejam inválidos.

Projeto 3: API de Gestão de Eventos e Inscrições

UC722 - Visualizar Evento

ID	UC722
Título	Visualizar Evento
Ator principal	Usuário, Administrador
Objetivo	Permitir que o usuário ou administrador visualize os detalhes de um evento específico.
Pré-condição	O evento já deve estar cadastrado no sistema. O usuário ou administrador deve estar autenticado (caso necessário).
Pós-condição	O usuário ou administrador verá as informações detalhadas sobre o evento, como nome, data, descrição e local.
Diagrama	[Fluxograma]



PROTÓTIPO

Participe de nossos Eventos ♡

Adelphos Solutions † Eventos:
Aqui você encontrará um ou mais eventos para apoiar.
Sua participação é de grande ajuda.
Não fique de fora!!! 🤩

Formulário de Cadastro de Evento

Nome do Evento

Event

Local do Evento

text

Data Final

dd / mm / aaaa

Vagas Disponíveis

0

Cadastrar

Formulário de Cadastro de Usuário

Nome Completo

Full name

E-mail

E-mail

CPF

CPF

Endereço

Adress

Cidade

City

Estado

Neighborhood

Cadastrar

Cadastrar usuário em nossos Eventos

Usuário

Selecione um usuário

Evento

Selecione um evento

Cadastrar

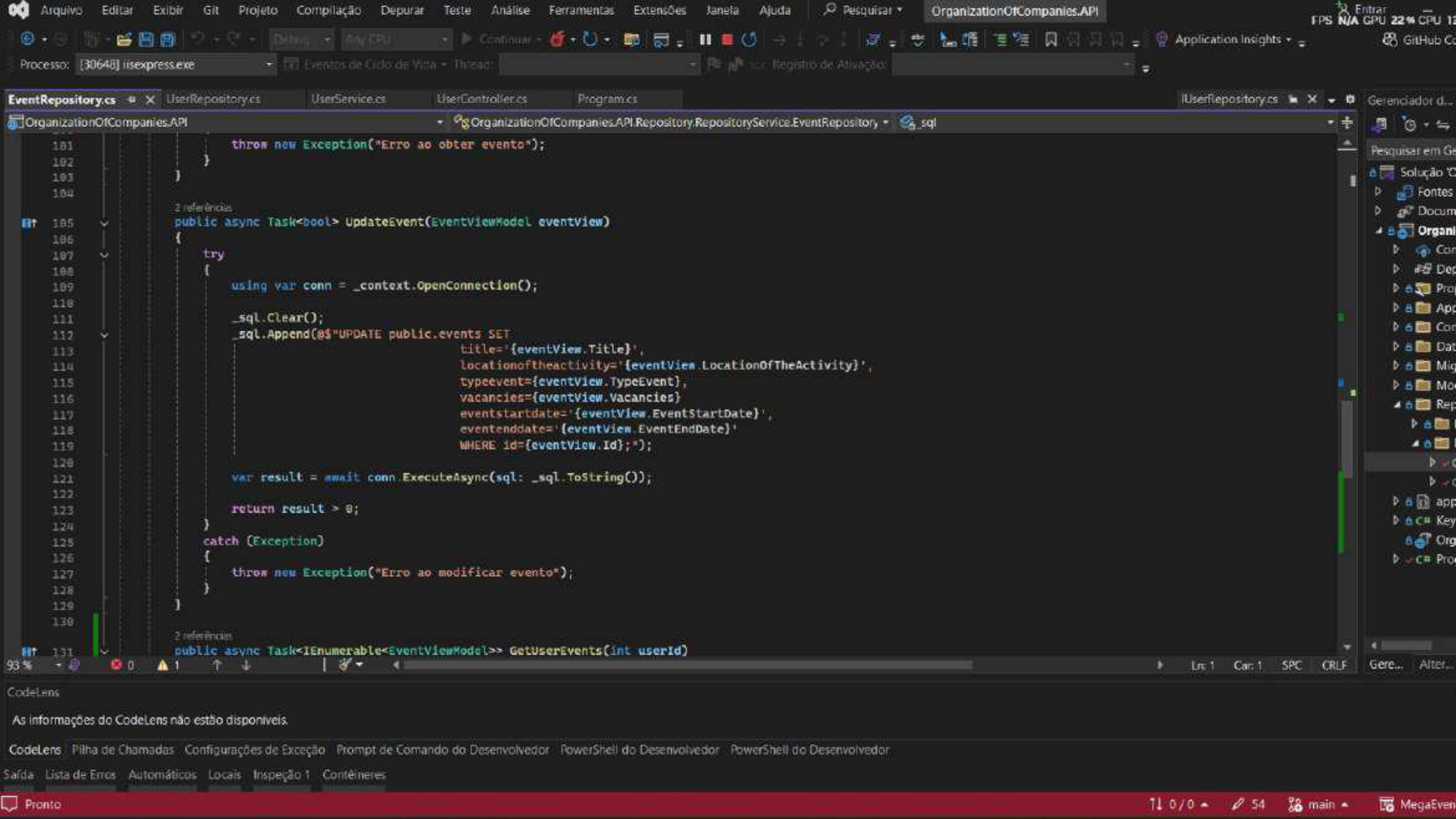
Listar Eventos do Usuário

Selecione o Usuário

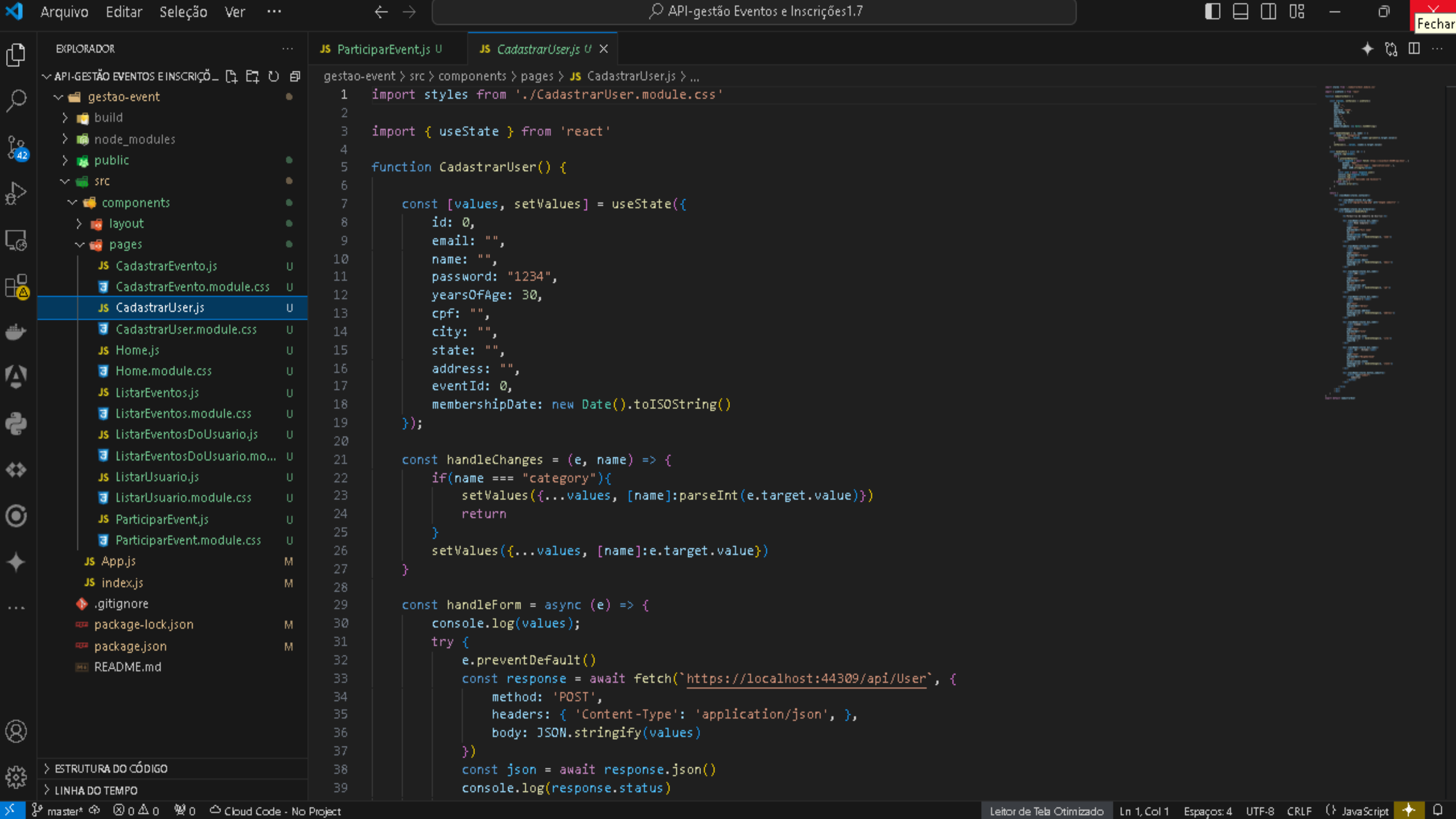
Selecione um usuário

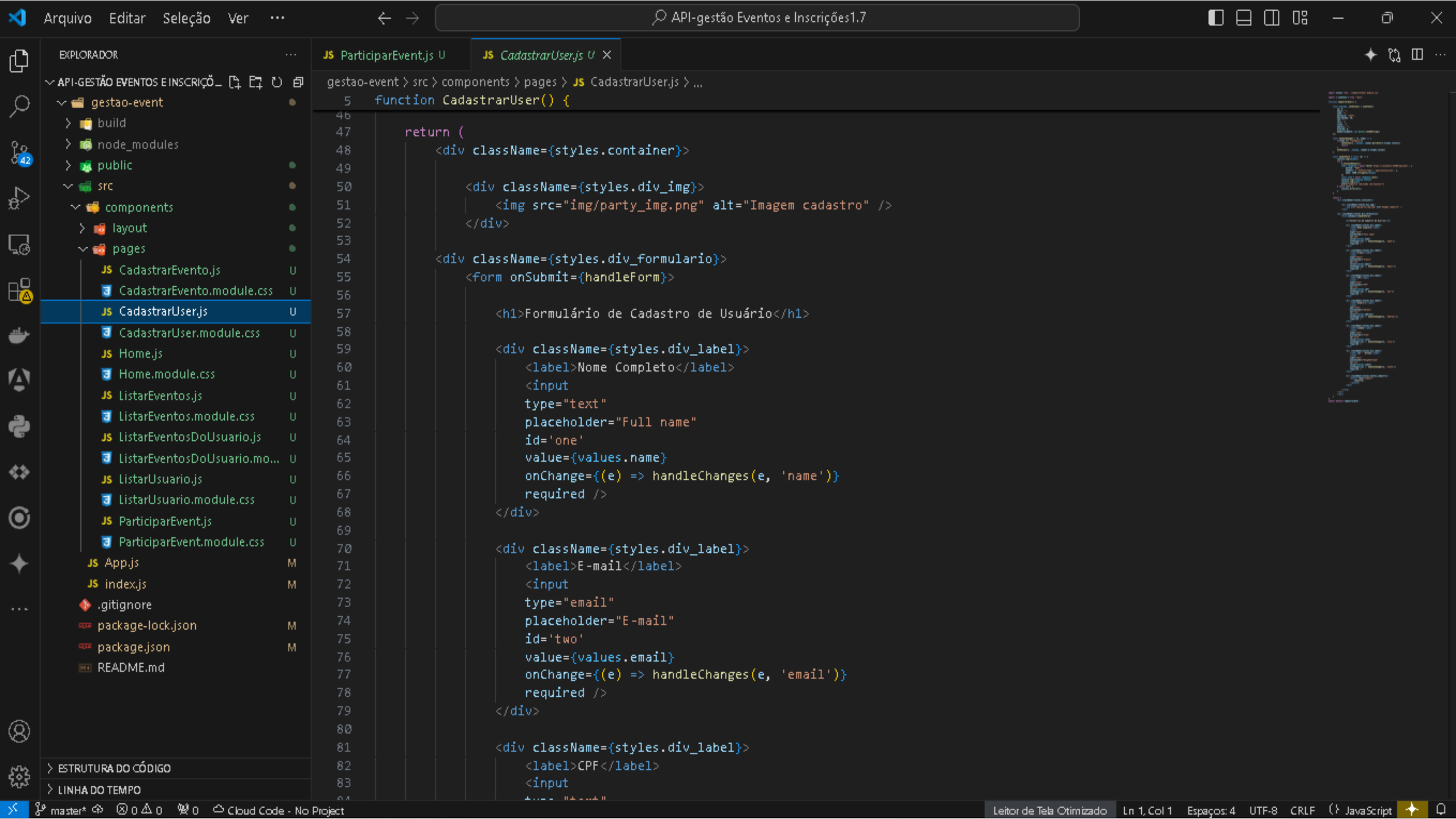
Eventos Participados:

CÓDIGO BACK-END


Arquivo Editar Exibir Git Projeto Compilação Depurar Teste Análise Ferramentas Extensões Janela Ajuda
Pesquisar
OrganizationOfCompanies.API
Entrar
FPS N/A GPU 22% CPU
Application Insights
GitHub Co
Processo: [30648] iisexpress.exe
EventRepository.cs
UserRepository.cs
UserService.cs
UserController.cs
Program.cs
IUserRepository.cs
OrganizationOfCompanies.API
OrganizationOfCompanies.API.Repository.RepositoryService.EventRepository
_sql
throw new Exception("Erro ao obter evento");
}
}
2 referências
public async Task<bool> UpdateEvent(EventViewModel eventView)
{
try
{
using var conn = _context.OpenConnection();
_sql.Clear();
_sql.Append(@$"UPDATE public.events SET
title='{eventView.Title}',
locationoftheactivity='{eventView.LocationOfTheActivity}',
typeevent={eventView.TypeEvent},
vacancies={eventView.Vacancies}
eventstartdate='{eventView.EventStartDate}',
eventenddate='{eventView.EventEndDate}'
WHERE id={eventView.Id};");
var result = await conn.ExecuteAsync(sql: _sql.ToString());
return result > 0;
}
catch (Exception)
{
throw new Exception("Erro ao modificar evento");
}
}
2 referências
public async Task<IEnumerable<EventViewModel>> GetUserEvents(int userId)
93 %
0
1
Ln: 1
Car: 1
SPC
CRLF
Gerenciador d...
Pesquisar em Ge
Solução 'O
Fontes
Docum
Organi
Con
Dep
Pro
App
Con
Dat
Mig
Mo
Rep
app
Key
Org
Pro
Gere...
Alter...
CodeLens
As informações do CodeLens não estão disponíveis.
CodeLens
Pilha de Chamadas
Configurações de Exceção
Prompt de Comando do Desenvolvedor
PowerShell do Desenvolvedor
PowerShell do Desenvolvedor
Saída
Lista de Erros
Automáticos
Locais
Inspeção 1
Contêineres
Pronto
0 / 0
54
main
MegaEven

CÓDIGO FRONT-END


Arquivo Editar Seleção Ver ···
API-gestão Eventos e Inscrições1.7
Fechar
EXPLORADOR
API-GESTÃO EVENTOS E INSCRIÇÕ...
gestao-event
build
node_modules
public
src
components
layout
pages
CadastrarEvento.js U
CadastrarEvento.module.css U
CadastrarUser.js U
CadastrarUser.module.css U
Home.js U
Home.module.css U
ListarEventos.js U
ListarEventos.module.css U
ListarEventosDoUsuario.js U
ListarEventosDoUsuario.mo... U
ListarUsuario.js U
ListarUsuario.module.css U
ParticiparEvent.js U
ParticiparEvent.module.css U
App.js M
index.js M
.gitignore
package-lock.json M
package.json M
README.md
ESTRUTURA DO CÓDIGO
LINHA DO TEMPO
ParticiparEvent.js U
CadastrarUser.js U
gestao-event > src > components > pages > CadastrarUser.js > ...
1 import styles from './CadastrarUser.module.css'
2
3 import { useState } from 'react'
4
5 function CadastrarUser() {
6
7 const [values, setValues] = useState({
8 id: 0,
9 email: "",
10 name: "",
11 password: "1234",
12 yearsOfAge: 30,
13 cpf: "",
14 city: "",
15 state: "",
16 address: "",
17 eventId: 0,
18 membershipDate: new Date().toISOString()
19 });
20
21 const handleChanges = (e, name) => {
22 if(name === "category"){
23 setValues({...values, [name]:parseInt(e.target.value)})
24 return
25 }
26 setValues({...values, [name]:e.target.value})
27 }
28
29 const handleForm = async (e) => {
30 console.log(values);
31 try {
32 e.preventDefault()
33 const response = await fetch(`https://localhost:44309/api/User`, {
34 method: 'POST',
35 headers: { 'Content-Type': 'application/json', },
36 body: JSON.stringify(values)
37 })
38 const json = await response.json()
39 console.log(response.status)
master* 0 0 0 Cloud Code - No Project
Leitor de Tela Otimizado Ln 1, Col 1 Espaços: 4 UTF-8 CRLF JavaScript


Arquivo Editar Seleção Ver ···
API-gestão Eventos e Inscrições1.7
EXPLORADOR
API-GESTÃO EVENTOS E INSCRIÇÕ_
gestao-event
build
node_modules
public
src
components
layout
pages
CadastrarEvento.js U
CadastrarEvento.module.css U
CadastrarUser.js U
CadastrarUser.module.css U
Home.js U
Home.module.css U
ListarEventos.js U
ListarEventos.module.css U
ListarEventosDoUsuario.js U
ListarEventosDoUsuario.mo... U
ListarUsuario.js U
ListarUsuario.module.css U
ParticiparEvent.js U
ParticiparEvent.module.css U
App.js M
index.js M
.gitignore
package-lock.json M
package.json M
README.md
ESTRUTURA DO CÓDIGO
LINHA DO TEMPO
ParticiparEvent.js U
CadastrarUser.js U
gestao-event > src > components > pages > CadastrarUser.js > ...
5 function CadastrarUser() {
47 return (
48
50
51
52
54
55 <form onSubmit={handleForm}>
57 <h1>Formulário de Cadastro de Usuário</h1>
59
60 <label>Nome Completo</label>
61 <input
62 type="text"
63 placeholder="Full name"
64 id='one'
65 value={values.name}
66 onChange={(e) => handleChanges(e, 'name')}
67 required />
68
70
71 <label>E-mail</label>
72 <input
73 type="email"
74 placeholder="E-mail"
75 id='two'
76 value={values.email}
77 onChange={(e) => handleChanges(e, 'email')}
78 required />
79
81
82 <label>CPF</label>
83 <input
master*
0 0 0
Cloud Code - No Project
Leitor de Tela Otimizado
Ln 1, Col 1
Espaços: 4
UTF-8
CRLF
JavaScript

Lições Aprendidas no Projeto

•**Planejamento como Pilar**: Investir tempo no planejamento inicial reduz retrabalho e problemas durante a execução.

•**Escolhas Tecnológicas**: React no frontend trouxe agilidade, e C# no backend garantiu robustez e escalabilidade.

•**Estrutura e Organização**: Repositórios bem estruturados e práticas de versionamento evitaram conflitos e facilitaram colaborações.

•**Integração Contínua e Testes**: CI e testes automatizados preveniram bugs e melhoraram a qualidade do sistema.

•**Documentação e Feedback**: Documentação clara e feedback contínuo alinharam o projeto às expectativas.

•**Conteinerização**: Docker simplificou ambientes de desenvolvimento e produção.

Conclusão: A experiência acumulada reforça a importância de planejamento, colaboração e práticas técnicas sólidas para o sucesso de projetos futuros.