



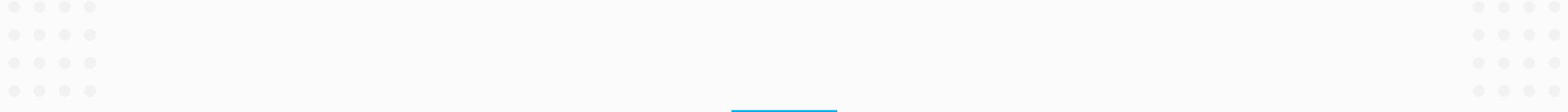
• • •

Node.js

CORS
Cadastrar produto



Você lembra disso?



```
res.setHeader('Access-Control-Allow-Origin', '*')
```

O que é?

É utilizado em servidores web para permitir que qualquer domínio (origem) acesse os recursos (como dados de uma API) fornecidos por esse servidor, superando a política de segurança padrão dos navegadores conhecida como **CORS (Cross-Origin Resource Sharing)**

Aonde estamos usando?

Porque estamos usando?

Vamos mudar

Apague toda a linha com esse código:

```
res.setHeader('Access-Control-Allow-Origin', '*')
```

No terminal:

```
# npm install cors
```

No começo do nosso código, abaixo da linha `const app = express()` coloque:

```
// necessário para permitir requisições de diferentes origens(domínios/servidores)
const cors = require('cors')
app.use(cors())
```

CORS - Cross-Origin Resource Sharing

É um mecanismo de segurança utilizado pelos navegadores para permitir que recursos (APIs, fontes, imagens) de um domínio (origem) sejam acessados por scripts rodando em outro domínio. Ele contorna a "Política de Mesma Origem" (Same-Origin Policy), autorizando requisições legítimas entre origens diferentes.

Erro Comum: Ocorre quando um front-end (localhost:5500) tenta acessar uma API (localhost:3000) sem que o servidor desta API tenha configurado os cabeçalhos CORS adequados.

Sem o CORS, navegadores bloqueiam solicitações fetch para domínios diferentes, restringindo a integração de aplicações modernas.



Teste o seu site para
ver nada quebrou



Cadastro de produtos

FrontEnd

- Criar o formulário
- Enviar os dados do formulário para a API

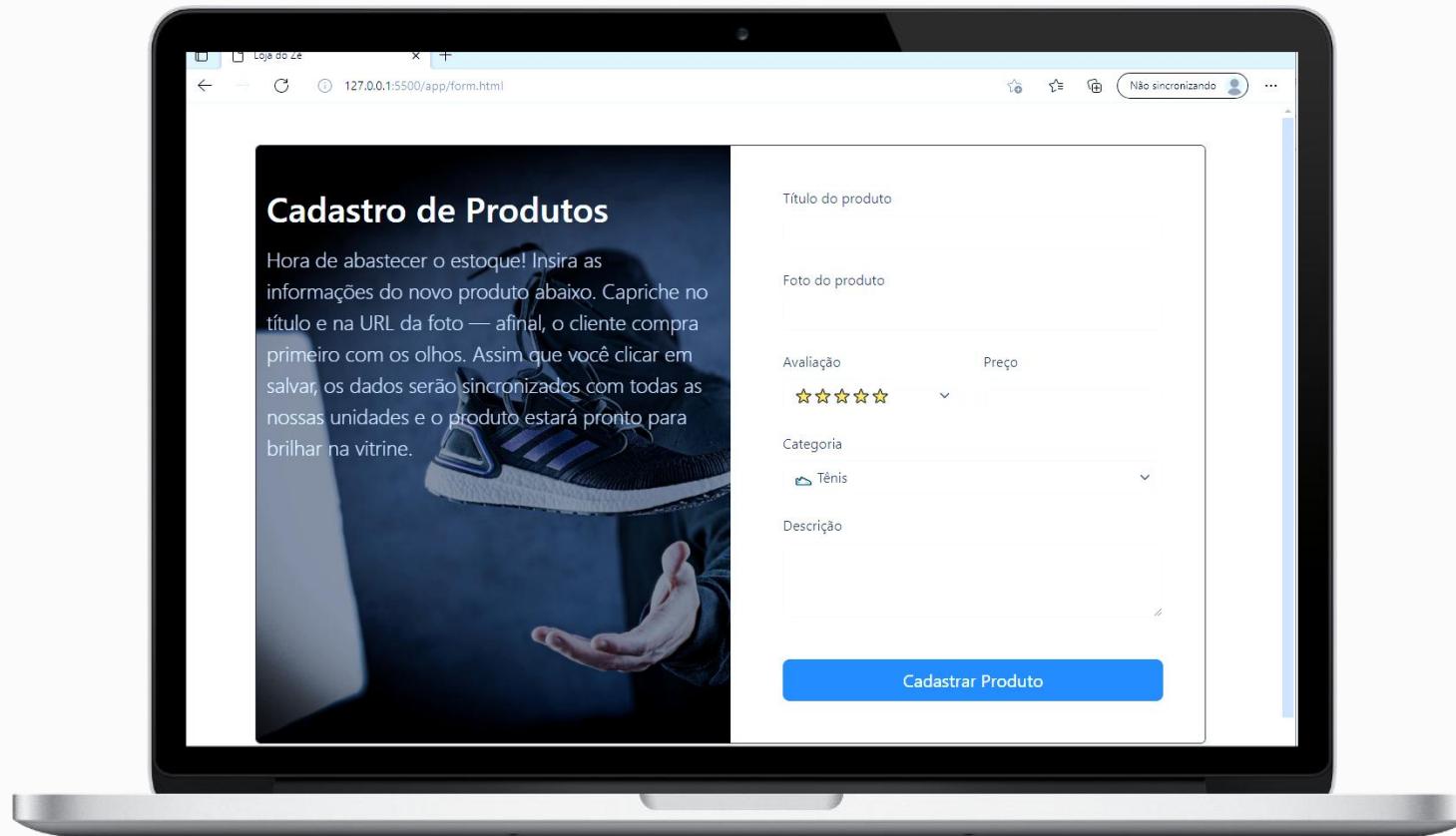
BackEnd

- rota /produto[POST]
 - criar a rota
 - receber os dados do frontend
 - montar o sql de insert
 - executar o insert e retornar resposta para o frontend

FRONTEND



Formulário de cadastro



FrontEnd

Dentro da pasta **site-loja** crie os arquivos:

- form.html
- app-form.js

Cole o código fornecido em cada um dos arquivos:

app-form.js - fnCadastrarProdutos()

```
let formDados = {  
    titulo: document.getElementById("_____").value,  
    preco: document.getElementById("_____").value,  
    descricao: document.getElementById("_____").value,  
    avaliacao: document.getElementById("_____").value,  
    foto: document.getElementById("_____").value,  
    categoria: document.getElementById("_____").value  
}
```

app-form.js - fnCadastrarProdutos()

```
fetch('http://localhost:3000/produto/', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify(formDados)
})

.then(reposta => reposta.json())
.then((dados) => {
    fnLimparCampos()
    console.log(dados)
})
.catch(erro => console.log(erro.message))
}
```

funções

```
function mensagem(){ alert('oi') } // função
```

```
function(){alert('oi')} // função anônima
```

```
() => {alert('oi')} // arrow function
```

funções



```
btn_salvar.addEventListener("click", function salvar() {
```

```
    fnCadastrarProdutos()
```

```
})
```

```
btn_salvar.addEventListener("click", function() {
```

```
    fnCadastrarProdutos()
```

```
})
```

```
btn_salvar.addEventListener("click", () => {
```

```
    fnCadastrarProdutos()
```

```
})
```

BACKEND



index.js

Vamos criar o verbo app.post("/produto/"

index.js

```
app.post("/produto/", function (req, res) {
  const { titulo, preco, descricao, avaliacao, foto, categoria } = req.body;
  conexao.query(`  

    INSERT INTO produtos(titulo, foto, descricao, preco, avaliacao, categoria)  

    values('${titulo}', '${foto}', '${descricao}', ${preco}, ${avaliacao}, '${categoria}')`,  

    function (erro, resultado) {
      if (erro) {
        res.json(erro);
      }
      res.send(resultado.insertId);
    });
})
```

atividades

Crie a tela de cadastro de unidades

  gutoxa27_bd_loja unidades
⌚ id : int(11)
⌚ nome_da_loja : varchar(250)
⌚ telefone : varchar(20)
⌚ email : varchar(250)
⌚ endereco : varchar(1000)
⌚ latitude : varchar(100)
⌚ longitude : varchar(100)
⌚ foto : varchar(1000)

  gutoxa27_bd_loja produtos
⌚ id : int(11)
⌚ titulo : varchar(100)
⌚ foto : varchar(1000)
⌚ descricao : text
preco : float
avaliacao : int(11)
⌚ categoria : varchar(250)