

ARQUITETURA DE SOFTWARE

João Choma Neto

joao.choma@unicesumar.edu.br

<https://github.com/JoaoChoma/arquitetura-software>

Unicesumar – Maringá



PORTAS DE SERVIÇO

- Se um serviço que estava rodando na porta 3000 caiu e você precisa subir um novo serviço em uma porta diferente
- Vamos evitar a necessidade de o cliente saber da mudança de porta.

Balanceamento de Carga com Redundância

- Com um balanceador de carga, os clientes não precisam saber em qual porta ou instância específica o serviço está rodando.
- O balanceador de carga gerencia as instâncias de servidor e redireciona as requisições conforme necessário.

```
http {  
    upstream myapp {  
        server 127.0.0.1:3000;  
        server 127.0.0.1:4000; # Outra instância ou porta  
    }  
  
    server {  
        listen 80;  
  
        location / {  
            proxy_pass http://myapp;  
        }  
    }  
}
```

```
const serverUrl = 'http://localhost'; // NGINX escuta na porta 80
```

```
axios.get(`${serverUrl}/api/data`)  
  .then(response => {  
    console.log(response.data);  
  })  
  .catch(error => {  
    console.error('There was an error!', error);  
  });
```

ARQUITETURA DE SOFTWARE

João Choma Neto

joao.choma@unicesumar.edu.br

<https://github.com/JoaoChoma/arquitetura-software>

Unicesumar – Maringá

