

2. Variáveis de Ambiente — Mudando Config sem alterar código

Por que isso existe?

No mundo real você roda o **mesmo teste** em vários ambientes:

Ambiente	URL
DEV	http://localhost:3000
PROD	https://api.meusistema.com

Se o QA precisa ficar editando o arquivo `.js` sempre que muda de ambiente → **vira bagunça**.

Em CI/CD (Jenkins, GitHub Actions) isso é **impossível** de manter.

Por isso o K6 permite pegar valores de **variáveis de ambiente**, sem mudar o código.

Como passar variáveis de ambiente

```
k6 run -e BASE_URL=http://localhost:3000 teste.js
```

No Jenkins:

- Ambiente DEV → `BASE_URL=http://dev.api`
- Ambiente PROD → `BASE_URL=https://api.prod`

O script **nunca muda**.

Quem muda é **o ambiente**.

Como usar no código

```
const BASE_URL = __ENV.BASE_URL || 'http://localhost:3000';
```

`__ENV.VARIAVEL` é como o K6 lê variáveis de ambiente.

Se nada for enviado, usa o valor padrão (`localhost`).

Controlando VUs e Iterações

Definir usuários virtuais

```
k6 run --vus 10 teste.js
```

Definir número total de iterações

```
k6 run --iterations 20 teste.js
```

Rodar 5 usuários executando 50 iterações cada

```
k6 run --vus 5 --iterations 50 teste.js
```

Controlando a duração do teste

```
k6 run --vus 10 --duration 30s teste.js
```

CLI sempre vence o arquivo

Se no script você tem:

```
export const options = {  
  vus: 10,  
  duration: '30s'  
};
```

Mas você roda assim:

```
k6 run --vus 1 --duration 5s teste.js
```

Quem vale é **o CLI**, não o arquivo.
