Para saber mais: custo do bcrypt

Nessa aula, usamos o `bcrypt.hash()` para proteger as senhas do usuários. Ela recebe um fator de custo e, para nossa aplicação, usamos um custo de `12`. Entretanto, isso pode não ser a melhor opção para o **seu caso específico**.

Como a própria OWASP (https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html#work-factors) comenta, não existe uma resposta definitiva para o *custo ideal*. Chegar nesse número depende de fatores como

- capacidade de processamento da sua máquina;
- razão performance/segurança da sua aplicação;
- tráfego do seu site;
- capacidade computacional da época.

E, mesmo que você precise de segurança, é importante tomar cuidado ao colocar um fator muito maior do que sua máquina suporta pois pode levar a ataques DoS (https://pt.wikipedia.org/wiki/Ataque_de_nega%C3%A7%C3%A3o_de_servi%C3%A7o).

Uma regra geral que a OWASP comenta é que o cálculo de um hash **não** deve levar **mais de um segundo** (e bem menos que isso para servidores com alto tráfego).

Você pode testar vários custos na sua máquina executando

```
const bcrypt = require('bcrypt');
for (let custo = 6; custo < 18; custo++) {
  const tempoInicial  = Date.now();
  bcrypt.hash('A', custo).then(
    () => console.log(`custo: ${custo}; tempo: ${
Date.now() - tempoInicial} ms`)
  );
}
```

COPIAR CÓDIGO

Com *benchmarks* de servidores relativamente atuais (https://github.com/cptwin/Password-Hashing-Algorithm-Benchmark-Tool-PHABT-/blob/master/Results/Xeon%20E3-1275-v3.csv) a gente consegue chegar que um fator entre 12 e 14 é o ideal (de 289ms a 1155ms).

Mesmo assim, como a Auth0 (https://auth0.com/blog/hashing-in-action-understanding-bcrypt/#L-code-bcrypt--code--Best-Practices) já

disse, o processo ideal é realizar uma pesquisa de UX para descobrir o tempo aceitável de espera em registro e login para seus usuários (100ms a 1000ms (https://web.dev/rail/?hl%3Den%23response_respond_in_under_100ms=), num caso geral). Com esse tempo, pegue o **maior custo** que se aproxime dele na sua máquina.

Se o custo for grande demais, existem alguns mecanismos de defesa de DoS como um sistema *proof-of-work* (https://pt.wikipedia.org/wiki/Prova_de_trabalho).