

THE TWELVE-FACTOR APP

INTRODUÇÃO

Na era moderna, software é comumente entregue como um serviço: denominados *web apps*, ou *software-como-serviço*. A aplicação doze-fatores é uma metodologia para construir softwares-como-serviço que:

- Usam formatos **declarativos** para automatizar a configuração inicial, minimizar tempo e custo para novos desenvolvedores participarem do projeto;
- Tem um **contrato claro** com o sistema operacional que o suporta, oferecendo **portabilidade máxima** entre ambientes que o executem;
- São adequados para **implantação** em modernas **plataformas em nuvem**, evitando a necessidade por servidores e administração do sistema;
- **Minimizam a divergência** entre desenvolvimento e produção, permitindo a **implantação contínua** para máxima agilidade;
- E podem **escalar** sem significativas mudanças em ferramentas, arquiteturas, ou práticas de desenvolvimento.

A metodologia doze-fatores pode ser aplicada a aplicações escritas em qualquer linguagem de programação, e que utilizem qualquer combinação de serviços de suportes (banco de dados, filas, cache de memória, etc).

EXPERIÊNCIA

Os contribuidores deste documento estão diretamente envolvidos no desenvolvimento e implantação de centenas de aplicações, e indiretamente testemunhando o desenvolvimento, operação e escalada de centenas de milhares de aplicações através de seu trabalho na plataforma [Heroku](#).

Este documento sintetiza toda nossa experiência e observação em uma variedade de aplicações que operam como software-como-serviço. Isto é a triangulação de práticas ideais ao desenvolvimento de software, com uma atenção particular a respeito das dinâmicas de crescimento orgânico de uma aplicação ao longo do tempo, a dinâmica de colaboração entre desenvolvedores trabalhando em uma base de código, e evitando os [custos de erosão de software](#)

Nossa motivação é aumentar a consciência de alguns problemas sistêmicos que temos visto no desenvolvimento de aplicações modernas, prover um vocabulário comum para discussão destes, e oferecer um amplo conjunto de soluções conceituais para esses problemas com a terminologia que os acompanha. O formato é inspirado nos livros de Martin Fowler [Padrões de Arquitetura de Aplicações Enterprise](#) e [Refatorando](#).

QUEM DEVE LER ESTE DOCUMENTO?

Qualquer desenvolvedor que esta construindo aplicações que rodam como serviço. Engenheiros de Operações que implantam ou administram tais aplicações.

OS DOZE FATORES

I. Base de Código

Uma base de código com rastreamento utilizando controle de revisão, muitos deploys

II. Dependências

Declare e isole as dependências

III. Configurações

Armazene as configurações no ambiente

IV. Serviços de Apoio

Trate os serviços de apoio, como recursos ligados

V. Construa, lance, execute

Separe estritamente os builds e execute em estágios

VI. Processos

Execute a aplicação como um ou mais processos que não armazenam estado

VII. Vínculo de porta

Exporte serviços por ligação de porta

VIII. Concorrência

Dimensione por um modelo de processo

IX. Descartabilidade

Maximizar a robustez com inicialização e desligamento rápido

X. Dev/prod semelhantes

Mantenha o desenvolvimento, teste, produção o mais semelhante possível

XI. Logs

Trate logs como fluxo de eventos

XII. Processos de Admin

Executar tarefas de administração/gerenciamento como processos pontuais

[Italiano \(it\)](#) | [简体中文 \(zh_cn\)](#) | [Русский \(ru\)](#) | [Slovensky \(sk\)](#) | [ภาษาไทย \(th\)](#) | [Česky \(cs\)](#) | [한국어 \(ko\)](#)
| [Brazilian Portuguese \(pt_br\)](#) | [Deutsch \(de\)](#) | [Turkish \(tr\)](#) | [Español \(es\)](#) | [Ελληνικά \(el\)](#) | [Français \(fr\)](#) |
[日本語 \(ja\)](#) | [Українська \(uk\)](#) | [English \(en\)](#) | [Tiếng Việt \(vi\)](#) | [Polski \(pl\)](#)

Written by Adam Wiggins

Last updated 2017

[Sourcecode](#)

[Download ePub Book](#)

[Privacy Policy](#)