



## Automação de tarefas 'osso' na operação

Bruno Ramos e Silva

Analista de Redes – PoP-BA/RNP



- 'Osso' é a tarefa que anda de mãos dadas com um serviço em produção e tende a ser:
  - Manual
  - Repetitiva
  - Automatizável
  - Baseada em interrupção
  - Sem valor na atividade, e
  - Que escala linearmente a medida que o serviço cresce
- Nem todas as tarefas chamadas de 'osso' tem todos estes atributos
  - Mas, quanto mais características destas casarem, maiores as chances de ser 'osso'

- Quanto menos 'osso' melhor
  - Se menosprezadas, acabam tomando 100% do tempo de toda a equipe
  - Estudos <sup>1</sup> estimam que 40 a 90% do custo total do sistema se dá após sua implantação
  - Tarefas 'Osso' nem sempre são uma 'tristeza', principalmente se forem em baixas doses, mas em altas doses:
    - Pouco ânimo, tédio, descontentamento
    - Menor progresso, 'apagar incêndio'
    - Deixa precedente para mais tarefas 'osso'
    - Estagnação da carreira, sentimento de 'fui enganado'

<sup>1</sup> R. Glass, Facts and Fallacies of Software Engineering, Addison-Wesley Professional, 2002.

- Eficiência na automação
  - Começar automatizando o que mais toma tempo da equipe na semana
    - i. Medir o que mais toma tempo
    - ii. Fazer projeto para automação
    - iii. Reservar tempo da equipe para projeto
      - Treinamento se necessário
  
- 'Mas nunca automatizei, não sei como iniciar o projeto.'  
Como começo? O que aprender?

- Orquestração de infraestrutura de servidores:
  - Puppet, Ansible, Salt
- Automação de infraestrutura de redes:
  - Salt + Napalm, Ansible
- Orquestração de containers
  - Docker, Kubernetes
- Integração contínua
  - Jenkins, Travis CI
- Programação
  - Shell scripting (Bash, Power Shell), Python, Perl, etc
    - i. Podem fazer a automação de atividades diárias 'sozinhos' ou em conjunto com as ferramentas acima

- Exemplos do PoP-BA:
  - Gestão de configuração dos servidores do PoP com SALT
    - i. Aplicar mudanças em todo o parque de uma só vez
    - ii. Backup automatico de configuração de roteadores
    - iii. Manter configurações de roteadores e servidores (evitar mudanças)
  - Automação de relatórios de disponibilidades mensais (insuportáveis)
    - i. Python para verificar padrões no preenchimento de colunas de arquivo CSV
    - ii. Geração automática de PDF no formato LATEX via Perl e Bash
  - Automação de operação do IX.br
    - i. Ativação/remoção de configuração (bilaterais/quarentena/produção)
      - Em desenvolvimento

- Automação da operação do IX.br
  - Um dos passos do projeto é configurar automaticamente VLANs de Switch A para Switch B
    - Script/Programa deve:
      - Gerar grafo da rede
        - Gerar caminho de A para B por algoritmo de menor caminho
          - Configurar VLAN em todas as portas dos switches no caminho

- Automação da operação do IX.br
  - Configurar automaticamente VLANs de Switch A para Switch B
    - Como o script/programa pode acessar e passar comandos para os switches?
      - [Python + biblioteca NetworkX] + Ansible
      - [Python + biblioteca NetworkX] + biblioteca Netmiko
      - [Python + biblioteca NetworkX] + [Salt + Napalm]
    - Testes da solução envolve debug de rede e de programação!
      - 'Falha da solução foi na programação ou na configuração de rede?'
    - Pensar em excluir NOC de toda a operação futura
      - Serviço Web self-service para participante do PTT?

- Conclusão
  - Atividades 'osso' são chatas e atrasam melhorias/ inovação a longo prazo
  - Automatizar ambiente para ganhar tempo livre para focar em projetos que agregam valor
    - Via programação e ferramentas de automação
      - Busquem conhecimento sobre elas :)
    - De preferência, iniciar pelo problema/atividade que toma mais tempo da equipe semanalmente
      - Reservar pessoas ou tempo semanal da equipe para isto!



# WTR

WORKSHOP  
DE TECNOLOGIAS DE REDES DO POP-BA

**Obrigado!**

Bruno Ramos e Silva  
Analista de Redes - PoP-BA/RNP

Contato: [brunoramos@pop-ba.rnp.br](mailto:brunoramos@pop-ba.rnp.br) / [brunorscc@gmail.com](mailto:brunorscc@gmail.com)

- Referências:
  - Murphy et. al, O'Reilly, Site Reliability Engineering: How Google Runs Production Systems, 2016
  - R. Glass, Facts and Fallacies of Software Engineering, Addison-Wesley Professional, 2002.