

**Endereço:**  
Jundiaí / SP

**Contato:**  
**E-mail:** carneiro.leandro@outlook.com  
**Celular:** (11) 97244-5778

---

## LEANDRO SOUZA CARNEIRO

---

Senior SysAdmin / SRE em iFood. Conhecimentos em Cloud Computing, API Gateway, cultura DevOps e sistemas GNU/Linux.

**Objetivo:** Atuar na área de Infraestrutura / SRE / DevOps

---

### Formação Acadêmica:

**Curso:** Engenharia de Computação  
**Grau:** Bacharel  
**Conclusão:** Dez/2012  
**Instituição:** Universidade São Francisco

**Curso:** Técnico em Computação  
**Grau:** Técnico  
**Conclusão:** Dez/2007  
**Instituição:** Colégio Divino Salvador

---

### Idiomas:

**Inglês**  
**Nível:** Avançado / Fluente

**Intercâmbio:**  
**Curso:** English for non-Americans  
**Duração:** 1 mês  
**Conclusão:** Mar/2014  
**Instituição:** ELS Language School - Silicon Valley

---

### Experiência Profissional:

**iFood**  
out/2018 até o presente momento  
SRE - Sr SysAdmin

**ADTsys Software**  
mar/2018 até set/18  
Analista de Infraestrutura em TI Sr. II

**Programarte Consultoria e Desenvolvimento de Softwares**  
jul/2010 até mar/2018  
Analista de Infraestrutura em TI

---

### Projetos no iFood:

- Gerenciamento e evolução da solução de API Gateway do iFood, baseada em Kong.
  - Implementação de monitoramento (Prometheus) com métricas por serviços, rotas e HTTP Codes de todos os API Gateways.
  - Desenvolvimento de plugins para o API Gateway a fim de lidar com autenticação, autorização e roteamentos.
  - Tratativa de logs de ELBs e API Gateway com solução baseada em ELK.
  - Desenvolvimento de cookbook em Chef para melhorar a automação da configuração do serviço de API Gateway.
  - Implementação e sustentação da infraestrutura do iFood.
-

## Projetos na ADTsys:

- Desenvolvimento de métodos em Groovy para servir em pipelines do Jenkins a fim de automatizar processos de um projeto que não tem APIs.
- Desenvolvimento de ferramenta Python, para ofertar instancias Spot na AWS e registrá-las nas *Stacks/Layers* do OpsWorks. Quando o CloudWatch detecta uma sobrecarga na instância, com base no limite personalizado, é acionado um webhook exposto pelo Jenkins, que pode receber, além do valor da oferta para o lance, parâmetros como "layerId", "stackID", tamanho da instância e muito mais. No final do ciclo de vida da instância, uma função Lambda é acionada para remover a instância do OpsWorks. Ferramenta usada como parte da integração para ambiente de infraestrutura efêmera.
- Criação e gerenciamento de receitas 'Chef' para configuração de máquinas via OpsWorks.
- Desenvolvimento de 'manifestos' no Puppet para codificar os ambientes dos nossos clientes, legados e novos.
- Tradução de Ansible 'playbooks' para Puppet 'manifests' para equalizar o ambiente.
- Desenvolvimento de 'pipelines' no Jenkins para automação de *deploys*. Ambientes CI / CD. Quando um cliente solicita um novo pipeline de CI / CD, cria-se um novo repositório no GitLab com duas *branches*, pipelines no Jenkins e configuração do webhook para aplicar as alterações com um evento push a uma *branch* específica. Como a ADTsys vende esses ambientes para muitos clientes, o trabalho se tornou repetitivo, então um pipeline foi desenvolvido para automatizar o processo de criação desses ambientes.  
Para os clientes que já possuem seu próprio repositório, cria-se uma automação customizada para construir e "*deployar*" o código do cliente conforme suas necessidades.
- Automação no Jenkins para expor um webhook para enviar notificações por SMS para trabalhar com o UpTime Robot. O script python recebe os parâmetros (number, monitoring name, url) e usa a API directcall para enviar as notificações.

---

## Projetos na Programarte:

- Pesquisa, implementação e documentação de cluster de WildFly, containerizado com Docker, geo-redundante e escalável. Projeto desenvolvido do zero para ambiente de alta disponibilidade do sistema "Neo" da "simplesmenteUse" em infraestrutura AWS ou on-premises. Este projeto foi desenvolvido com cluster de Docker Swarm para o ambiente SaaS, com Weave plugin, também foi utilizado PeerVPN para criar uma rede mesh, com multicast habilitado inclusive nas redes AWS. Todos os containers tem health-checks, auto-healing and habilidade para escalar horizontalmente. A arquitetura contém DNS via R53, Apache Containers com Mod\_Cluster via multicast to auto-discovery dos Containers WildFly, MongoDB clusterizado (replicaSet) e RDS server, ou MySQL com Percona.
- Pesquisa e documentação de cluster de balanceamento de carga com mod\_jk, dois JBoss Application Servers e Apache com SSL, utilizando algoritmo 'Round Robin', conectados ao SQL Server mirroring a fim de configurar ambiente de alta disponibilidade para a versão SaaS do "simplesmenteUse" em infraestrutura AWS.
- Migração de servidores de e-mail da "simplesmenteUse" para Zimbra. Configuração do próprio Zimbra, além dos DNS e firewall. Maior estabilidade adquirida e disponibilidade de análise dos logs de envio / recebimento.
- Automatização de processos em shell script de diversas rotinas como instalação e deploy, checagem de servidores, backup / restore de arquivos do FileServer e também de bancos de dados Oracle, SQL Server e MySQL, minimizando possíveis falhas humanas durante estes processos.

---

## Conhecimentos:

**Sistemas Operacionais:** Linux (preferencialmente Red Hat-ish), Windows

**Cloud:** AWS, GCP, Azure

**Containerização:** Kubernetes, Docker e Swarm

**Virtualização:** KVM, VirtualBox

**Linguagens:** Lua, Ruby, Python, Groovy, Shell Script

**Automação:** Jenkins, Vagrant, Terraform

**Gerenciadores de Configuração:** Puppet, Chef, OpsWorks

**Web Servers / Load-Balancer / Proxies:** Kong API Gateway, OpenResty, Nginx, HA Proxy, Apache

---