

Especificações Técnicas da Infraestrutura do LIT para desenvolvimento

Versão 1.2

Última revisão 27/04/2016

Editor Matheus Silva Santos

Resumo

Após consideração das opções disponíveis no mercado, estáveis e com bom suporte, ficaram definidas as ferramentas listadas abaixo como partes da infraestrutura padrão disponível para desenvolvimento no LIT.

- **IDE:** Editor de texto de preferência do desenvolvedor
- **Interpretador:** CPython 3.4
- **Framework web:** Django 1.9 e Flask 0.10
- **SGBD:** PostgreSQL 9.4
- **Versionamento:** Git, hospedado no Bitbucket
- **Servidor HTTP:** “Apache” 2.4 + Gunicorn 19.4
- **SO:** CentOS 7
- **Integração contínua:** Jenkins 1.651, se for necessário

Justificativas

A metodologia utilizada para a escolha das ferramentas leva em conta (na ordem): **relevância, licença, performance, preferência do desenvolvedor** (se aplicável). A versão de cada uma leva em conta a **estabilidade e suporte**.

IDE

Não há muitas opções viáveis de IDE para a linguagem Python, em comparação com Java, por exemplo. E as que existem, ou são ferramentas pagas: PyCharm, Komodo IDE, Wing IDE; ou são instáveis e experimentais: Netbeans com plugin nbPython, Eclipse com plugin PyDev. Algumas das ferramentas pagas (notavelmente o PyCharm) são gratuitas caso o projeto desenvolvido seja de [código aberto](#) ou [puramente educacional](#); isto é, nenhum projeto desenvolvido nela poderia ter fins lucrativos.

A maioria dos desenvolvedores Python prefere utilizar um simples **editor de texto**, mas poderoso o suficiente para algumas tarefas automatizadas. **Emacs**, vi, nano, gedit, **Atom**, Notepad++, são gratuitos para qualquer fim e suficientes para desenvolvimento.

Interpretador

Python é uma linguagem originalmente interpretada. Com o aumento de sua popularidade, foram criados interpretadores e compiladores alternativos para a linguagem, focando em uma ou outra característica a ser melhorada. Como o interpretador oficial (**CPython**) é muito estável, presente na maioria das plataformas e suficientemente otimizado, ele será utilizado. A versão 2 [será descontinuada em breve](#), logo a adoção do Python 3 é obrigatória. Fora escolhida a *minor version* 3.4 por ser a mais popular entre as distribuições.

Framework web

Curiosamente se fez necessário a opção por duas ferramentas invés de uma. Dentre tantos e excelentes frameworks web para Python, o Django e o Flask se destacam por suas vantagens. O Django possui tudo que é necessário para construir uma aplicação web robusta, com uma API fácil de persistência de dados. Por esses motivos também, ele pode ser um exagero ao se considerar tarefas simples, onde pode ser substituído pelo Flask.

O Django está em versão estável 1.9 e o Flask na 0.10 com boa maturidade.

SGBD

A escolha do Postgres foi simples. A maioria da comunidade Python o prefere (da mesma forma que a comunidade PHP prefere o MySQL), logo, é fácil encontrar suporte para a utilização de frameworks Python com o Postgres. Pesa também na decisão o desempenho do [Postgres vs. MySQL](#).

A versão utilizada será a 9.4, que trouxe melhorias significativas de desempenho e é a mais popular entre as distribuições Linux.

Versionamento

Tendo duas opções claras para ferramentas de versionamento, Git e Subversion, a decisão de utilizar o Git foi instantânea. O Git está em ascensão e é, não só mais popular que o SVN, como o de melhor benefício atualmente.

A questão real foi decidir a hospedagem do serviço do Git, entre Bitbucket, GitHub e hospedar nos próprios servidores do LIT. A apesar de ser viável e indiscutivelmente a mais barata das soluções, hospedar no LIT exigiria uma estrutura de backup e segurança que está fora de cogitação. O GitHub oferece gratuitamente apenas repositórios públicos, ou seja, apenas para projetos de código aberto. Por sua vez, o Bitbucket [oferece ilimitados repositórios privados gratuitamente](#), contanto que sua equipe não passe de 5 pessoas.

Servidor HTTP

Para prover o serviço é necessário obviamente de um servidor web. Com as opções de uso do Apache HTTP e do Nginx, a escolha do Apache se deu pela popularidade e termos de licença, que são totalmente livres. A versão escolhida é a 2.4, pela sua maturidade.

Contudo, um servidor web como o Apache não consegue se comunicar com as aplicações Python se estas estiverem em outros servidores ou dentro de máquinas virtuais diferentes. É preciso de um servidor HTTP intermediário interligado às aplicações, como o Gunicorn. Assim o Apache pode atuar como proxy reverso, redirecionando o tráfego para as aplicações. A última versão do Gunicorn até o momento é a 19.4.

Sistema Operacional

Tendo como opções cogitadas o Debian, Ubuntu Server, Fedora Server e CentOS, o CentOS foi escolhido por nenhuma razão especial que não seja a compatibilidade com o que já é adotado pela Algar Telecom. Todas as opções são baseadas no kernel Linux, têm as ferramentas necessárias, performam bem, tem versões muito estáveis e pelo histórico, as atualizações de segurança são rápidas. Obviamente foi escolhida a versão CentOS 7, a última estável com previsão de suporte longo.

Integração contínua

Não há outra opção com o mesmo nível de independência, estabilidade e suporte para CI com Python, se não o Jenkins. Há o Travis CI que funciona como SaaS, mas os planos são caros e impossibilitam seu uso no LIT.

A versão do Jenkins a ser utilizada, se realmente for necessário, será a 1.651.