

APOSTILA “PYTHON: CONCEITOS E APLICAÇÕES”

Bruno Fernandes Duarte de Melo¹, Emmanuel Guerra de Lima¹, Ronivy de Melo Soares¹, Maurício Neves Asenjo², Katya Lais Ferreira Patella Couto².

¹Alunos do 4º. ano do Curso Técnico em Informática Integrado ao Ensino Médio do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – campus Cubatão (IFSP-Cubatão).

²Docentes do Curso Técnico em Informática Integrado ao Ensino Médio do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – campus Cubatão (IFSP-Cubatão).

RESUMO

O Curso Técnico em Informática Integrado ao Ensino Médio do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – *campus* Cubatão não oferece, em sua grade curricular, a linguagem de programação Python (uma linguagem atual, intuitiva e bastante exigida no mercado). Este trabalho objetiva elaborar uma apostila educacional acerca desse assunto, pois, entre diversas linguagens de programação existentes, Python foi a que mais cresceu nos últimos anos. A apostila almeja guiar os alunos nos passos iniciais de seus projetos em Python por meio de conceitos básicos, exemplos e exercícios de fixação.

PALAVRAS-CHAVE: Python; apostila; projeto.

1 INTRODUÇÃO

De acordo com o *site* “Becode”, a linguagem de programação Python tem ganhado um enorme destaque nos últimos cinco anos, devido a sua simplicidade. Criada para diminuir e simplificar o código fonte de um *software* de computador, essa linguagem facilita a declaração de variáveis ou a execução de tarefas. Além disso, possui código aberto, uma comunidade imensa e um grande mercado.

Essa linguagem é tão versátil que, diferentemente das outras linguagens, começou “de cima para baixo”, ou seja, começou sendo utilizada pelos programadores das mais renomadas empresas ligadas à Tecnologia da Informação, tais como: Google, Youtube e Yahoo. E, por ser concisa e muito produtiva, foi adotada por diversas companhias pelo mundo.

O presente projeto, denominado *Apostila “python: conceitos e aplicações”*, o qual é tema deste artigo, almeja estudar uma linguagem de programação que não é oferecida pelo Curso Técnico em Informática Integrado ao Ensino Médio do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – *campus* Cubatão, em sua grade curricular, a linguagem Python. Este trabalho objetiva elaborar uma apostila educacional acerca desse assunto. A

apostila pretende guiar alunos nos passos iniciais de projetos em Python, por meio de conceitos básicos, exemplos e exercícios de fixação.

Este artigo é composto por seis seções, sendo está a primeira, na qual é oferecida uma visão geral da linguagem que será apresentada. A segunda seção oferece textos que contextualizam o desenvolvimento da apostila e justificam a sua criação. A terceira seção é referente aos matérias e métodos, em que a metodologia utilizada no desenvolvimento e os recursos aplicados são expostos. A quarta seção contém resultados e discussões. A quinta seção fornece as considerações finais dos desenvolvedores da apostila sobre todas as etapas de criação. A sexta e última seção apresenta as referências.

2 DESENVOLVIMENTO DA APOSTILA

2.1 Idealização

A ideia inaugural do projeto era que se criasse uma apostila referente a uma linguagem de programação, de forma que os profissionais da área de tecnologia da informação iniciassem e aprofundassem seus estudos. No entanto, questões como qual linguagem abordar e quais *frameworks* apresentar surgiram.

Como a própria pesquisa mostrada na apostila sugere, Python é uma linguagem muito produtiva, de fácil aprendizado, concisa e que vem crescendo cada vez mais no ramo de criação de *softwares*.

Após discussões feitas em grupo, a proposta final surgiu: uma apostila com abrangência de conteúdo, que não só apresentasse uma introdução à linguagem, mas também mostrasse muito do que o Python tem a oferecer, como algumas bibliotecas e *frameworks*, que possibilitam ao desenvolvedor trabalhar com as mais diversas cargas de trabalho.

2.2 Construção da apostila

Inicialmente, buscou-se definir qual seria a IDE (Ambiente Integral de Desenvolvimento) a ser utilizada. Após algumas pesquisas, decidiu-se utilizar o Visual Studio Community 2017 versão 15.9.12, visto a simplicidade e a amplitude de recursos oferecidos por tal ferramenta de desenvolvimento.

Destarte, a construção da apostila deu-se: da pesquisa em grupo sobre tal linguagem de programação; da parte prática, na qual se desenvolveram algumas aplicações que posteriormente foram utilizadas como exemplo na apostila; e da parte escrita, que diz respeito ao processo de escrita da apostila.

3 MATERIAIS E MÉTODOS

Para a realização deste artigo, foi necessário executar uma ampla pesquisa na *internet* sobre o assunto específico, a fim de encontrar suporte à implementação da proposta inicial. A metodologia adotada para realizar a apostila de Python possui uma abordagem explicativa que foi dividida em quatro fases. A ideia surgiu a partir do desejo de se aprender uma nova linguagem que não foi ensinada na grade do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – *campus* Cubatão, simultaneamente à busca de um projeto de aprendizado eficaz para o cotidiano.

A princípio, foi delimitada a ferramenta por meio da qual seriam desenvolvidas as aplicações e, conseqüentemente, o curso. Foi escolhido o Visual Studio Community 2017 como IDE, o qual suporta o Python 3.6.6, que é uma linguagem responsável por toda a criação do ambiente no qual se deseja trabalhar, seja ele WEB, seja Inteligência Artificial ou Database.

Houve, ainda, um teste inicial para saber qual seria a melhor IDE a ser utilizada bem como a demonstração de um passo a passo para instalação da mesma. Chegou-se à conclusão de que seria a IDE exposta anteriormente (Visual Studio Community 2017). Em seguida, foi iniciado o desenvolvimento de projetos básicos com funções simples diretamente com o Python, baseando-se nos aprendizados de linguagens passadas (if/else, laços de repetição, etc.) para se fazerem testes lógicos e reconhecimento da sintaxe, tendo como objetivo inicial conhecer as ferramentas de que ela dispõe.

Na segunda etapa, denominada “Estruturação lógica”, começaram a surgir projetos de nível intermediário, como definição de uma classe, instanciação de objetos, elaboração de método construtor e destrutor e utilização de propriedades e métodos estáticos. Nessa etapa, apareceram alguns desafios simples, solucionados com pesquisas relacionadas ao tema. Foi então que se deu início à construção inicial da apostila. Erguem-se aí os primeiros capítulos.

A terceira etapa, denominada “Construção visual”, exigiu muito mais pesquisa e dedicação. Foi uma etapa fundamental para conhecer melhor o Python. Nela houve a

elaboração de programas com partes visuais e percebeu-se que, diferentemente das linguagens com que o grupo lidou anteriormente, onde era possível, por meio de uma ferramenta chamada Toolbox, clicar e arrastar, inserindo, assim, botões, *label's* e caixas de texto, a linguagem Python não dispunha de uma interface visual simultânea ao desenvolvimento do código. Mesmo com os desafios, foi possível desenvolver uma calculadora simples com as particularidades do Python, e o resultado foi um sucesso.

Após tudo isso, foi necessário avançar no conhecimento da linguagem e, para tanto, houve a busca por um *framework* que possibilitasse a capacidade de “clicar e arrastar”, recurso comum em diversas outras linguagens. A partir dessa necessidade, surgiu o Ironpython, o qual supriu as necessidades situacionais do grupo.

Para complementar ainda mais a apostila, chegou-se à quarta fase, denominada “Finalização da apostila”. Foi organizado e compilado todo o material produzido, bem como o que ainda está em desenvolvimento, como uma Inteligência Artificial e um projeto WEB.

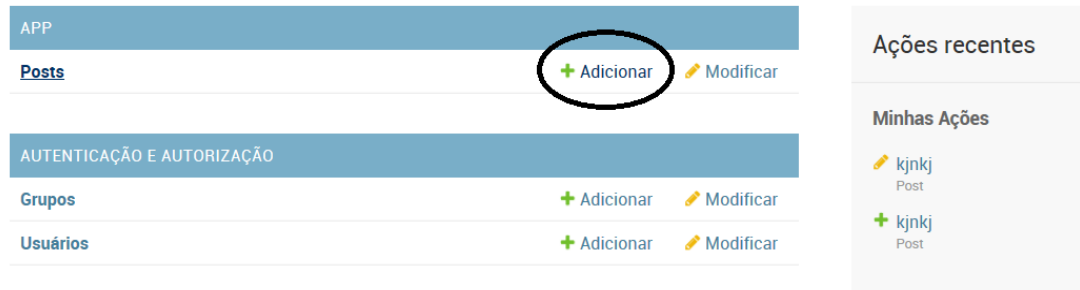
Por fim, organizaram-se os capítulos finais da apostila e estabeleceram-se as referências de pesquisas e organização das fontes, de forma que se cumprisse a proposta inicial oferecida pela apostila, que é guiar os alunos nos passos iniciais de seus projetos em Python por meio de conceitos básicos, exemplos e exercícios de fixação. Atualmente, diante do mercado tão crescente e promissor, aprender Python se torna cada vez mais uma tarefa necessária dentre os programadores.

4 RESULTADOS E DISCUSSÃO

A apostila fornece uma breve apresentação introdutória sobre a linguagem de programação Python, debatendo algumas de suas características gerais, e apresenta diversas explicações e conceitos de um modo didático, desde implementações básicas da linguagem até mesmo uso de alguns *frameworks* e inteligência artificial (IA).

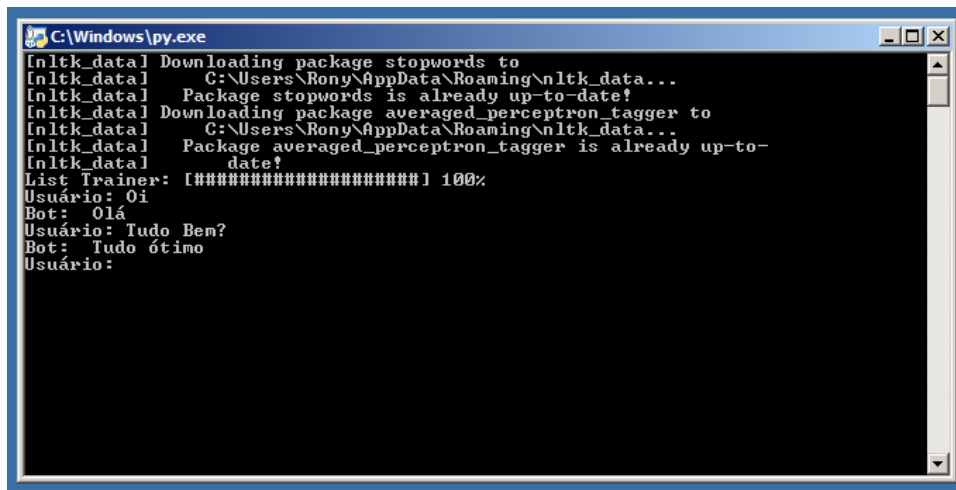
Dentre os diversos projetos realizados na elaboração do material, são destaques: uma aplicação *web*, ilustrada na Figura 1, que realiza cadastros no banco de dados MySQL por meio da *framework* Django, e um Chatbot, ou seja, um robô de conversação que simula diálogos com pessoas via *internet*, por meio da inteligência artificial, ilustrado na Figura 2.

Administração do Site



The screenshot displays the Django Admin interface. The main content area is divided into sections: 'APP' and 'AUTENTICAÇÃO E AUTORIZAÇÃO'. Under 'APP', there is a 'Posts' section with two buttons: '+ Adicionar' (circled in red) and 'Modificar'. Under 'AUTENTICAÇÃO E AUTORIZAÇÃO', there are sections for 'Grupos' and 'Usuários', each with '+ Adicionar' and 'Modificar' buttons. On the right side, there are two panels: 'Ações recentes' and 'Minhas Ações', both showing a single action: 'kjkj Post'.

Figura 1 - Django



```
C:\Windows\py.exe
Inltk_data] Downloading package stopwords to
Inltk_data] C:\Users\Rony\AppData\Roaming\nltk_data...
Inltk_data] Package stopwords is already up-to-date!
Inltk_data] Downloading package averaged_perceptron_tagger to
Inltk_data] C:\Users\Rony\AppData\Roaming\nltk_data...
Inltk_data] Package averaged_perceptron_tagger is already up-to-
Inltk_data] date!
List Trainer: [#####] 100%
Usuário: Oi
Bot: Olá
Usuário: Tudo Bem?
Bot: Tudo ótimo
Usuário:
```

Figura 2 - Chatbot

Em relação aos resultados obtidos, foram observados quatro aspectos considerados importantes.

Em primeiro lugar, programar em Python requer menos código para realizar tarefas básicas quando comparado a outras linguagens, como Java e C#.

Em segundo lugar, ao se iniciar o aprendizado de uma nova linguagem, geralmente surgem obstáculos, como: erro de sintaxe, mensagem de erro ou mesmo identificação de um erro qualquer. O Python, por apresentar uma comunidade grande e ativa, soluciona tais problemas com mais facilidade. Como pode ser percebido no Gráfico 1, houve um crescimento de visualizações em questões relacionadas a Python (cerca de 10.5% das visualizações mensais do StackOverflow).

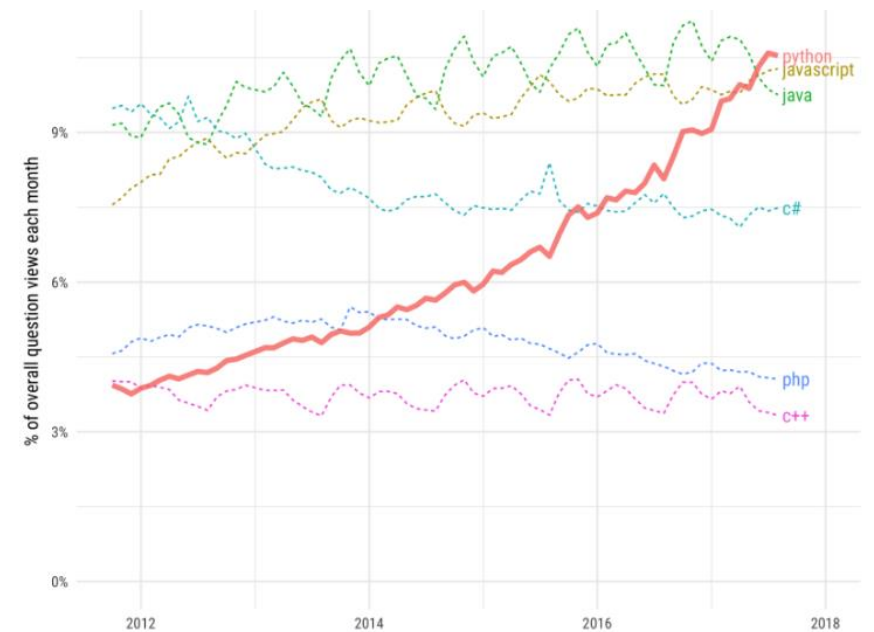


Gráfico 1 - Visualizações mensais do StackOverflow

Em terceiro lugar, a linguagem em estudo apresenta diversificação de conteúdo, desde a parte mais básica até alguns conceitos mais avançados, como Django e IA.

Por fim, o estudante programador poderá iniciar e/ou aprofundar seus conhecimentos a respeito da linguagem Python, tecendo habilidades muito valorizadas no mercado atual, como desenvolver aplicações em uma grande diversidade de plataformas, e possuir familiaridade com as bibliotecas e estruturas para a análise de dados.

5 CONSIDERAÇÕES FINAIS

Neste trabalho, foi desenvolvida uma apostila para a introdução dos estudos em Python, mostrando que a linguagem é fundamental no mercado atual.

O material produzido cumpre os propósitos de maneira simples e objetiva, podendo ser acessado por qualquer pessoa interessada em linguagens de programação, seja para o desenvolvimento de aplicações em Django, seja para o desenvolvimento de Inteligência Artificial. Sendo a informática uma área em constante atualização, é preciso sempre dar continuidade ao estudo da linguagem de programação Python, para que mais pessoas possam se beneficiar do aprendizado.

Este trabalho foi de suma importância, pois proporcionou a conscientização da constante necessidade de atualização dos conhecimentos. Além de ter agregado novos

saberes, permitiu pôr em prática teorias e estudos feitos durante todos os anos. Pôde, ainda, abrir caminhos para pesquisas futuras, uma vez que algumas ideias podem ser implementadas e melhoradas para agregar ainda mais acessibilidade ao conhecimento comum, como, por exemplo, usar um IDE mais leve ou até mesmo fazer o uso de bancos de dados para os códigos desenvolvidos.

6 REFERÊNCIAS

AMARAL, Rodrigo. **Qual a melhor IDE para Python.** nov. 2012. Disponível em: <<http://rodrigoamaral.net/2012/11/11/qual-a-melhor-ide-para-python/>>. Acesso em: 07 abr. 2019.

COMO criar minha primeira classe em Python. **DEVMEDIA.** dez. 2017. Disponível em: <<https://www.devmedia.com.br/como-criar-minha-primeira-classe-em-python/38912>>. Acesso em: 01 maio 2019.

COMPARE NetBeans vs Visual Studio IDE. **FinancesOnline.** mar. 2019. Disponível em: <<https://comparisons.financesonline.com/netbeans-vs-visual-studio-ide>>. Acesso em: 04 abr. 2019.

CRIANDO um Chatbot com Python. **Treinaweb.** jun. 2016. Disponível em: <<https://www.treinaweb.com.br/blog/criando-um-chatbot-com-python/>>. Acesso em: 25 ago. 2019.

CRIAR um Chatbot com o Chatterbot no Python. **Blog de Ruan Bekker: De uma mente curiosa a posts no Github.** jun. 2017. Disponível em: <<https://www.treinaweb.com.br/blog/criando-um-chatbot-com-python/https://blog.ruanbekker.com/blog/2017/12/13/create-a-chatbot-with-chatterbot-on-python/>>. Acesso em: 22 ago. 2019.

EXISTEM métodos estáticos em Python. **Stackoverflow.** out. 2017. Disponível em: <<https://pt.stackoverflow.com/questions/248035/existem-m%C3%A9todos-est%C3%A1ticos-em-python/248041>>. Acesso em: 03 maio 2019.

MÉTODOS estáticos e de classe. **Python.** jun. 2010. Disponível em: <<http://python.6.x6.nabble.com/Metodos-estaticos-e-de-classe-td2092625.html>>. Acesso em: 10 jun. 2019.

O que é Django. **DjangoGirls**. jun. 2016. Disponível em: <<https://tutorial.djangogirls.org/pt/django/>>. Acesso em: 14 ago. 2019.

PROPRIEDADES em python. **PYTHONHELP**. jun. 2012. Disponível em: <<https://pythonhelp.wordpress.com/2012/06/22/propriedades-em-python/>>. Acesso em: 06 maio 2019.

PYTHON/Conceitos básicos/Classes. **WikiLivros**. dez. 2013. Disponível em: <https://pt.wikibooks.org/wiki/Python/Conceitos_b%C3%AAsicos/Classes>. Acesso em: 04 abr. 2019.

PYTHON e Programação Orientada a Objeto. **Python_Brasil**. set. 2008. Disponível em: <https://wiki.python.org.br/ProgramacaoOrientadaObjetoPython#A5.6_M.2BAOk-todos_Est.2BAOE-ticos>. Acesso em: 03 jun. 2019.

PYTHON Orientado a Objetos com o Framework CherryPy. **DEVMEDIA**. nov. 2017. Disponível em: <<https://www.devmedia.com.br/python-orientado-a-objetos-com-o-framework-cherrypy/33489>>. Acesso em: 05 maio 2019.

QUAIS as vantagens de usar o Visual Studio para desenvolver aplicativos. **Impacta**. fev. 2018. Disponível em: <<https://www.impacta.com.br/blog/2018/02/16/quais-vantagens-usar-visual-studio-desenvolver-aplicativos/>>. Acesso em: 20 maio 2019.

QUAL é o sentido das propriedades em Python. **Imasters**. out. 2012. Disponível em: <<https://imasters.com.br/back-end/qual-e-o-sentido-das-propriedades-em-python>>. Acesso em: 03 maio 2019.

TUTORIAL: Instalar o Visual Studio no Windows. **Platzi**. out. 2017. Disponível em: <<https://platzi.com/blog/tutorial-instalar-visual-studio-en-windows/>>. Acesso em: 02 abr. 2019.