

## PROPOSTA PARA ELABORAÇÃO DE CURSOS DE EXTENSÃO

1. AUTOR: Prof Me. Carlos Eduardo de França Roland

2. COLABORADOR (ES): Prof Dr Daniel Facciolo Pires

### 3. INTRODUÇÃO

Python ocupa a quarta posição no ranking de popularidade de linguagens de programação pelo índice TIOBE de Junho/2018, atrás de Java, C, e C++; é a terceira linguagem que melhor remunera profissionais pelo ranking da ChallengeRocket.com; é a terceira linguagem no ranking do GITHUB em número de *pull requests*; é a primeira das 5 mais bem pagas e com maior demanda por profissionais do ranking de 2018 da TechRepublic no Reino Unido; e é mais fácil de aprender que outras linguagens de programação, além de ser a ferramenta mais adequada para o aprendizado e prática de Data Science, cuja demanda por perfís técnicos está revolucionando o mercado de trabalho. Aprender a programar computadores usando Python permite que se foque no problema, sem se perder tempo com a sintaxe da linguagem; resulta em alta produtividade; é simples de iniciar, sem esgotar as possibilidades de uso.

### 4. PÚBLICO ALVO

Alunos que queiram conhecer os fundamentos de programação de computadores com a linguagem Python.

### 5. PLANO DE ENSINO DO CURSO

#### 5.1. OBJETIVO DO CURSO

O estudante deverá ser capaz de resolver problemas de programação de computadores usando linguagem Python, a fim de apresentar habilidades e competências no competitivo mercado de desenvolvimento de sistemas.

#### 5.2. CONTEÚDO PROGRAMÁTICO

##### 5.2.1. **Porque aprender programação de computadores**

- 5.2.1.1. Criatividade e motivação
- 5.2.1.2. Arquitetura de hardware de computadores
- 5.2.1.3. Entendendo programação
- 5.2.1.4. Palavras e sentenças
- 5.2.1.5. Conversando com Python
- 5.2.1.6. Terminologia: interpretador e compilador
- 5.2.1.7. Escrevendo um programa
- 5.2.1.8. O que é um programa?
- 5.2.1.9. Os blocos construtivos de programas
- 5.2.1.10. O que possivelmente pode dar errado?
- 5.2.1.11. Depurando
- 5.2.1.12. A jornada do aprendizado

##### 5.2.2. **Variáveis, expressões e comandos**

- 5.2.2.1. Valores e tipos
- 5.2.2.2. Variáveis
- 5.2.2.3. Nomes de variáveis e palavras-chave
- 5.2.2.4. Comandos
- 5.2.2.5. Operadores e operandos

- 5.2.2.6. Expressões
- 5.2.2.7. Ordem de operações
- 5.2.2.8. Operador módulo
- 5.2.2.9. Operações com cadeias de caracteres
- 5.2.2.10. Pedindo dados para o usuário
- 5.2.2.11. Comentários
- 5.2.2.12. Escolhendo nomes mnemônicos para variáveis
- 5.2.2.13. Depuração
- 5.2.3. **Execução condicional**
- 5.2.3.1. Expressões booleanas
- 5.2.3.2. Operadores locais
- 5.2.3.3. Execução condicional
- 5.2.3.4. Execução alternativa
- 5.2.3.5. Condicionais encadeadas
- 5.2.3.6. Condicionais aninhadas
- 5.2.3.7. Capturando exceções usando *try* e *except*
- 5.2.3.8. Avaliação curto-circuito de expressões lógicas
- 5.2.3.9. Depurando
- 5.2.4. **Funções**
- 5.2.4.1. Chamadas de funções
- 5.2.4.2. Funções internas
- 5.2.4.3. Funções de conversão de tipo
- 5.2.4.4. Funções matemáticas
- 5.2.4.5. Números aleatórios
- 5.2.4.6. Adicionando novas funções
- 5.2.4.7. Definições e usos
- 5.2.4.8. Fluxo de execução
- 5.2.4.9. Parâmetros e argumentos
- 5.2.4.10. Funções frutíferas e funções vazias
- 5.2.4.11. Porque funções?
- 5.2.4.12. Depurando
- 5.2.5. **Iterações**
- 5.2.5.1. Atualizando variáveis
- 5.2.5.2. O comando *while*
- 5.2.5.3. Laços infinitos
- 5.2.5.4. “Laços infinitos” e *break*
- 5.2.5.5. Encerrando iterações com *continue*
- 5.2.5.6. Laços definidos usando *for*
- 5.2.5.7. Padrões de laços
- 5.2.5.7.1. Laços de contagem e somas
- 5.2.5.7.2. Laços de máximos e mínimos
- 5.2.5.8. Depurando
- 5.2.6. **Cadeias de caracteres**
- 5.2.6.1. Uma cadeia é uma sequência
- 5.2.6.2. Obtendo o comprimento de uma cadeia com *len*
- 5.2.6.3. Percorrendo uma string com um laço
- 5.2.6.4. Fatiamento de cadeias
- 5.2.6.5. Cadeias são imutáveis
- 5.2.6.6. Executando laços e contagens
- 5.2.6.7. O operador *in*
- 5.2.6.8. Comparações de cadeias

- 5.2.6.9. Métodos *string*
- 5.2.6.10. Analisando cadeias
- 5.2.6.11. Operador de formatação
- 5.2.6.12. Depurando
- 5.2.7. **Arquivos**
- 5.2.7.1. Persistência
- 5.2.7.2. Abrindo arquivos
- 5.2.7.3. Arquivos de textos e linhas
- 5.2.7.4. Lendo arquivos
- 5.2.7.5. Procurando através de um arquivos
- 5.2.7.6. Permitindo que o usuário escolha o nome do arquivo
- 5.2.7.7. Usando *try*, *except*, e *open*
- 5.2.7.8. Escrevendo arquivos
- 5.2.7.9. Depurando
- 5.2.8. **Listas**
- 5.2.8.1. Uma lista é uma sequência
- 5.2.8.2. Listas são mutáveis
- 5.2.8.3. Percorrendo listas
- 5.2.8.4. Operações com listas
- 5.2.8.5. Fatiamento de listas
- 5.2.8.6. Métodos de listas
- 5.2.8.7. Deletando elementos
- 5.2.8.8. Listas e funções
- 5.2.8.9. Listas e cadeias
- 5.2.8.10. Analisando listas
- 5.2.8.11. Objetos e valores
- 5.2.8.12. Apelidos
- 5.2.8.13. Argumentos de listas
- 5.2.8.14. Depurando
- 5.2.9. **Dicionários**
- 5.2.9.1. Dicionário como um conjunto de contadores
- 5.2.9.2. Dicionários e arquivos
- 5.2.9.3. Executando laços e dicionários
- 5.2.9.4. Análise avançada de textos
- 5.2.9.5. Depurando
- 5.2.10. **Tuplas**
- 5.2.10.1. Tuplas são imutáveis
- 5.2.10.2. Comparando tuplas
- 5.2.10.3. Atribuição a tupla
- 5.2.10.4. Dicionários e tuplas
- 5.2.10.5. Atribuições múltiplas com dicionários
- 5.2.10.6. As palavras mais comuns
- 5.2.10.7. Usando tuplas como chaves em dicionários
- 5.2.10.8. Sequências: cadeias, listas, e tuplas – Oh My!
- 5.2.10.9. Depurando

Aula	Conteúdo	Atividade
1	<b>Porque aprender programação</b> <b>Variáveis, expressões e comandos</b>	

Aula	Conteúdo	Atividade
2	<b>Execução condicional</b> <b>Funções</b>	
3	<b>Iteração</b> <b>Cadeias</b>	
4	<b>Arquivos</b> <b>Listas</b>	
5	<b>Dicionários</b> <b>Tuplas</b>	

### 5.3. EMENTA

A programação de computadores. A comunicação entre o programador e computador: linguagens de programação. Linguagens compiladas e interpretadas. O interpretador Python. Conceitos e componentes fundamentais da linguagem de programação Python. Controle de fluxo de processamento e funções. Iterações e operações com cadeias de caracteres. Manipulação de arquivos de dados. Estruturas de dados nativas em Python.

### 5.4. SISTEMA DE AVALIAÇÃO DA DISCIPLINA

Presença em 75% das aulas e entrega das soluções das atividades propostas

### 5.5. METODOLOGIA DE ENSINO

Aulas expositivas sobre os conceitos dos tópicos tratados, estudo dirigido, e atividades práticas em microcomputadores para fixação de conteúdos

### 5.6. BIBLIOGRAFIA BÁSICA E COMPLEMENTAR

DOWNEY, Allen; ELKNER, Jeffrey; MEYERS, Chris. HOW TO THINK LIKE A COMPUTER SCIENTIST: Learning with Python. Massachusetts: Green Tea Press. 2002.

MENEZES, Nilo N. C. INTRODUÇÃO À PROGRAMAÇÃO COM PYTHON: algoritmos e lógica de programação para iniciantes. São Paulo: NOVATEC, 2010.

SANTANA, Osvaldo; GALESI, Thiago. PYTHON E DJANGO: desenvolvimrnto ágil de aplicações web. São Paulo: NOVATEC. 2010.

SEVERANCE, Charles R. PYTHON FOR EVERYBODY: exploring data using Python 3. Licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. Disponível em: [http://do1.dr-chuck.com/pythonlearn/EN\\_us/pythonlearn.pdf](http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf). Acesso em: 15.jun.2018.

### 5.7. NECESSIDADE DE RECURSOS AUDIO VISUAIS E/OU SOFTWARES ESPECÍFICOS

Laboratório de informática que acomode todos os alunos interessados, e projetor multimídia.

### 6. CARGA HORÁRIA E HORÁRIOS

20 horas, segunda a sexta das 19:15 às 22:30

16 a 20 de julho de 2018

Franca, 15 de junho de 2018.