

## DCO2004 – Laboratório de Prototipagem de Sistemas de Comunicações

Prof. Dr. Vicente Angelo de Sousa Junior

Universidade Federal do Rio Grande do Norte (UFRN) – 2019.2



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

1

## Conteúdo de Hoje

- Apresentação e Objetivos da Disciplina
- Conteúdo Programático e Cronograma
- Metodologia de Ensino, Avaliação e estatísticas de notas
- Sobre o Professor
- Conhecendo a Turma e Seus Objetivos



Departamento de Engenharia de Comunicações

**DCO**

Curso de Engenharia de Telecomunicações

**CETEL**

Curso de Engenharia de Elétrica

**Engenharia Elétrica UFRN**

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

2

## Apresentação da Disciplina

- Laboratório de Prototipagem de Sistemas de Comunicações
  - **Laboratório:** práticas (**hands-ons**) voltados a **experimentação prática de conceitos teóricos** relacionados com sistemas de comunicações;
  - **Prototipagem:** prototipagem em **software** de aspectos técnicos;
  - **Sistemas de comunicações:** explorar conceitos desde processamento de sinais, modulação analógica e digital, até a prototipagem de sistemas de comunicação comerciais.
- **Expectativa de aprendizado**
  - Relembrar conceitos vistos em várias disciplinas do curso: princípios de telecomunicações, sistemas de telecomunicações, comunicações digitais, comunicações sem fio\* e comunicações móveis\*;
  - Aprendizado e prática de linguagens de programação não compiladas ou interpretadas (**scripting** ou **linguagem de script**);
  - Aprendizado/prática de tecnologias de **sistemas de versionamento (gitHub)**.
  - Prática com **Linux**.

3

## Objetivos da Disciplina

- **Reapresentar e discutir os conceitos fundamentais de comunicação analógica e digital**
  - Modelos matemáticos e problemas
  - Aplicações práticas
- **Discutir sobre alguns aspectos de implementação baseada em software (prototipagem)**
- **Praticar linguagens de programação Matlab e Python**

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
 vicente.sousa@ufrn.edu.br

4

# Linguagens de programação

- **Python:**

- Criação/distribuidor: anos 90 (Python Software Foundation).

- **Características:**

- Linguagem de altíssimo nível;
- Suporta diferentes paradigmas de programação (e.g. OOP);
- Legibilidade do código (sintaxe moderna e clara), favorecendo códigos organizados, fáceis de compreender e manter, sem perda em produtividade;
- Desenvolvimento comunitário, o que facilita a constante evolução da linguagem;
- Open source;
- Uso difundido: aplicações web, soluções de telecom, jogos, Inteligência Artificial e Aprendizado de Máquina (*Machine Learning – ML*) e entre outros (Google, Yahoo! e NASA usam Python).



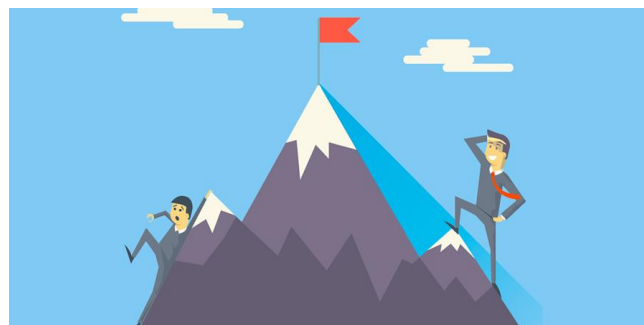
5

# Linguagens de programação

- **Mercado (09/02/2017)**

- “Python continuou a crescer em popularidade em 2016 e subiu dois lugares em nosso ranking para ser a terceira língua mais comum por **postagem de emprego**”

**E-COGNI TREINAMENTOS** (<https://www.e-cogni.com.br/programacao/as-9-linguagens-de-programacao-com-maior-demanda-para-2017/>).



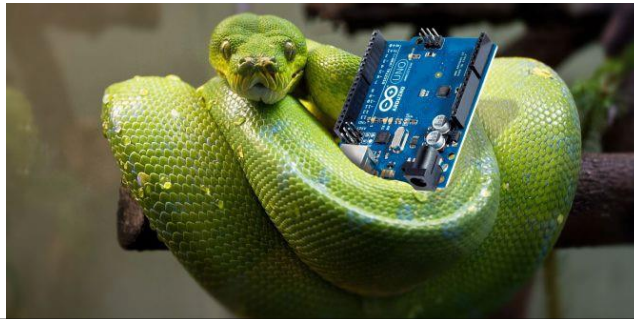
6

## Linguagens de programação

- **Mercado (21/04/2018)**

- “É possível ver que Java ainda é a linguagem mais utilizada no mundo, mas Python é uma das linguagem que mais vem crescendo ao longo dos anos.”

**OBSERVATÓRIO DE TENDÊNCIAS DIGITAIS DA PUCPR,**  
<https://medium.com/tend%C3%Aancias-digitais/python-engolindo-o-mercado-6872769800b2>).



7

## Linguagens de programação (Julho 2017)

- **TIOBE** (<https://www.tiobe.com/tiobe-index/>): search query + "<language> programming"

Jul 2017	Jul 2016	Change	Programming Language
1	1		Java
2	2		C
3	3		C++
4	4		Python
5	5		C#

8

## Linguagens de programação (Julho 2019)

- TIOBE (<https://www.tiobe.com/tiobe-index/>): search query + "<language> programming"

Jul 2019	Jul 2018	Change	Programming Language	Ratings	Change
1	1		Java	15.058%	-1.08%
2	2		C	14.211%	-0.45%
3	4	▲	Python	9.260%	+2.90%
4	3	▼	C++	6.705%	-0.91%
5	6	▲	C#	4.365%	+0.57%
6	5	▼	Visual Basic .NET	4.208%	-0.04%
7	8	▲	JavaScript	2.304%	-0.53%
8	7	▼	PHP	2.167%	-0.67%
9	9		SQL	1.977%	-0.36%
10	10		Objective-C	1.686%	+0.23%
11	12	▲	Ruby	1.636%	+0.43%

“Python, sem dúvidas, é uma linguagem bem divertida e interessante para se começar a estudar.” (<http://blog.geekhunter.com.br/5-linguagens-de-programacao-para-aprender-em-2017/>)

9

## Linguagens de programação (Julho 2019)

- TIOBE (<https://www.tiobe.com/tiobe-index/>): search query + "<language> programming"

### Programming Language Hall of Fame

The hall of fame listing all "Programming Language of the Year" award winners is shown below. The award is given to the programming language that has the highest rise in ratings in a year.

Year	Winner
2018	🏆 Python
2017	🏆 C
2016	🏆 Go
2015	🏆 Java
2014	🏆 JavaScript
2013	🏆 Transact-SQL
2012	🏆 Objective-C
2011	🏆 Objective-C
2010	🏆 Python

10

## Linguagens de programação (Julho 2017 e Julho 2018)

- Top Programming Languages 2017: **IEEE Spectrum**  
(<http://spectrum.ieee.org/computing/software/top-programming-languages-2017-focus-on-jobs>)
  - **Método de medida:** [http://spectrum.ieee.org/ns/IEEE\\_TPL\\_2017/methods.html](http://spectrum.ieee.org/ns/IEEE_TPL_2017/methods.html)
- Top Programming Languages 2019: **IEEE Spectrum**  
(<https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>)
  - **Método de medida:** <https://spectrum.ieee.org/static/ieee-top-programming-languages-2018-methods>
  - **App:** [https://spectrum.ieee.org/ns/IEEE\\_TPL\\_2018/index/2018/1/1/1/1/1/50/1/50/1/50/1/30/1/30/1/30/1/20/1/20/1/5/1/5/1/20/1/100/](https://spectrum.ieee.org/ns/IEEE_TPL_2018/index/2018/1/1/1/1/1/50/1/50/1/50/1/30/1/30/1/30/1/20/1/20/1/5/1/5/1/20/1/100/)



11

## Linguagens de programação (Julho 2019)

2017

Choose a Ranking (choose a weighting or make your own)

IEEE Spectrum Trending Jobs Open Custom

Edit Ranking | Add a Comparison | [Twitter](#) | [Facebook](#)

Language Types (click to hide)

Web Mobile Enterprise Embedded

Language Rank Types Spectrum Ranking

1. Python	Web	100.0
2. C	Mobile	99.7
3. Java	Web	99.4
4. C++	Mobile	97.2
5. C#	Web	88.6
6. R	Mobile	88.1
7. JavaScript	Web	85.5
8. PHP	Web	81.4
9. Go	Web	76.1
10. Swift	Mobile	75.3
11. Arduino	Mobile	73.0
12. Ruby	Web	72.4
13. Assembly	Mobile	70.1

2018

Choose a Ranking (choose a weighting or make your own)

IEEE Spectrum Trending Jobs Open Custom

Edit Ranking | Add a Comparison | [Twitter](#) | [Facebook](#)

Language Types (click to hide)

Web Mobile Enterprise Embedded

Language Rank Types Spectrum Ranking

1. Python	Web	100.0
2. C++	Mobile	99.7
3. Java	Web	97.5
4. C	Mobile	96.7
5. C#	Web	89.4
6. PHP	Web	84.9
7. R	Mobile	82.9
8. JavaScript	Web	82.6
9. Go	Web	76.4
10. Assembly	Mobile	74.1
11. Matlab	Web	72.8
12. Scala	Web	72.1
13. Ruby	Web	71.4
14. HTML	Web	71.2

12

# Linguagens de programação

- IEEE Spectrum: Focus on Jobs

2017

















































Choose a Ranking (choose a weighting or make your own)

IEEE Spectrum Trending **Jobs** Open Custom

Edit Ranking | Add a Comparison | [Twitter](#) [Facebook](#)

Language Types (click to hide)

☒ Web ☒ Mobile ☒ Enterprise ☒ Embedded

Language Rank	Types	Jobs Ranking
1. Java	   	100.0
2. C	   	99.3
3. Python	   	99.3
4. C++	   	92.6
5. JavaScript	   	90.2
6. C#	   	86.6
7. PHP	   	81.0
8. HTML	   	79.6
9. Ruby	   	77.2
10. Swift	   	77.2
11. Assembly	   	76.2
12. Shell	   	74.7

2018

























































Choose a Ranking (choose a weighting or make your own)

IEEE Spectrum Trending **Jobs** Open Custom

Edit Ranking | Add a Comparison | [Twitter](#) [Facebook](#)

Language Types (click to hide)

☒ Web ☒ Mobile ☒ Enterprise ☒ Embedded

Language Rank	Types	Jobs Ranking
1. Python	   	100.0
2. Java	   	99.2
3. C	   	98.8
4. C++	   	94.6
5. C#	   	86.2
6. JavaScript	   	85.7
7. Assembly	   	83.4
8. PHP	   	83.1
9. HTML	   	81.3
10. Scala	   	76.5
11. Shell	   	76.3
12. Ruby	   	75.5
13. Matlab	   	68.2
14. R	   	67.5

13

# Linguagens de programação

- IEEE Spectrum: Focus on Trending

2017

















































Choose a Ranking (choose a weighting or make your own)

IEEE Spectrum **Trending** Jobs Open Custom

Edit Ranking | Add a Comparison | [Twitter](#) [Facebook](#)

Language Types (click to hide)

☒ Web ☒ Mobile ☒ Enterprise ☒ Embedded

Language Rank	Types	Trending Ranking
1. Python	   	100.0
2. C	   	98.4
3. C++	   	97.7
4. Java	   	97.0
5. Swift	   	88.6
6. JavaScript	   	87.2
7. Go	   	86.5
8. R	   	84.4
9. C#	   	80.2
10. Ruby	   	79.1
11. PHP	   	78.8
12. Assembly	   	75.0

2018

























































Choose a Ranking (choose a weighting or make your own)

IEEE Spectrum Trending **Jobs** Open Custom

Edit Ranking | Add a Comparison | [Twitter](#) [Facebook](#)

Language Types (click to hide)

☒ Web ☒ Mobile ☒ Enterprise ☒ Embedded

Language Rank	Types	Trending Ranking
1. Python	   	100.0
2. C++	   	96.7
3. Java	   	94.6
4. C	   	93.7
5. Go	   	85.5
6. JavaScript	   	80.8
7. PHP	   	79.9
8. Scala	   	78.6
9. Ruby	   	77.2
10. HTML	   	75.5
11. Assembly	   	75.0
12. C#	   	74.0
13. Shell	   	72.6
14. R	   	72.0

14

# Linguagens de programação

- IEEE Spectrum: Focus on Trending

2017



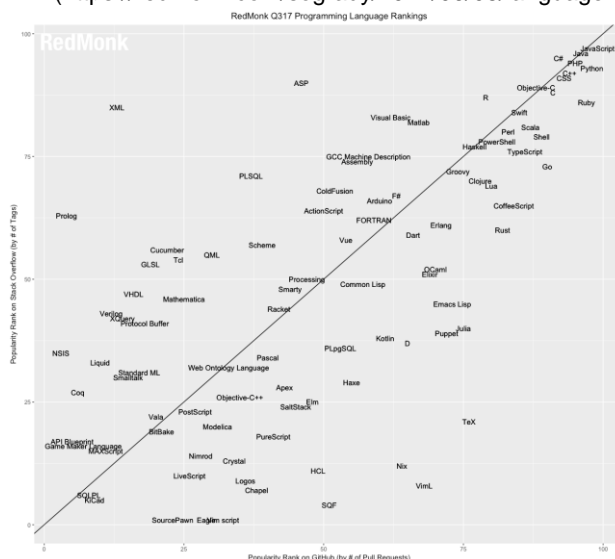
2018



15

# Linguagens de programação

- Top Programming Languages 2017: Redmonk  
(<https://redmonk.com/sogrady/2017/06/08/language-rankings-6-17/>)



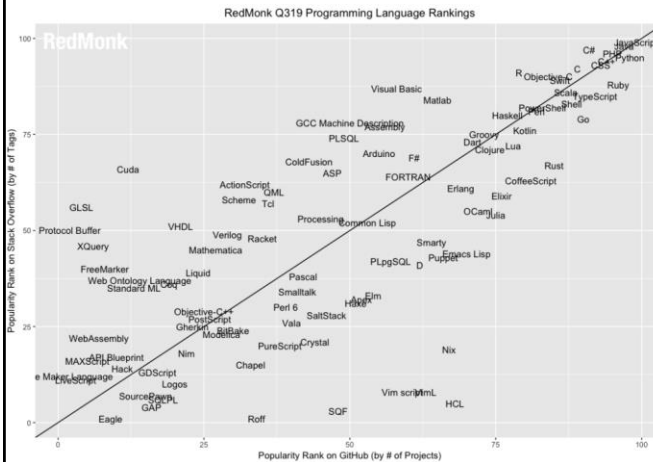
- 1 JavaScript
- 2 Java
- 3 Python**
- 4 PHP
- 5 C#
- 6 C++
- ...
- 14 R
- 15 Go
- 15 Perl
- 17 TypeScript
- 18 PowerShell
- 19 Haskell
- 20
- CoffeeScript
- 20 Lua
- 20 Matlab**

16



# Linguagens de programação

- **Top Programming Languages 2019 (July 18, 2019): Redmonk**  
([https://redmonk.com/sograpy/2019/07/18/language-rankings-6-19/?utm\\_source=rss&utm\\_medium=rss&utm\\_campaign=language-rankings-6-19](https://redmonk.com/sograpy/2019/07/18/language-rankings-6-19/?utm_source=rss&utm_medium=rss&utm_campaign=language-rankings-6-19))



- 1 JavaScript
- 2 Java
- 3 Python
- 4 PHP
- 5 C++
- 5 C#
- 7 CSS
- 8 Ruby
- 9 C
- 10 TypeScript
- 11 Swift
- 12 Objective-C
- 13 Scala
- 14 Shell
- 15 R
- 16 Go
- 17 PowerShell

17

# Linguagens de programação

- **Matlab (*Matrix Laboratory*)**
  - Criação: anos 1970
    - 1970: Cleve Moler, chefe do departamento de ciência da computação da Universidade do Novo México (bibliotecas baseadas em Fortran);
    - 1984: Cleve Moler, Jack Little (Universidade de Stanford) e Steve Bangert reescreveram o Matlab em C e criaram a Mathworks;
    - As bibliotecas reescritas ficaram conhecidas como LAPACK e EISPACK.
- **Vídeo interessante: Origins of MATLAB, Cleve Moler**  
(<https://www.mathworks.com/videos/origins-of-matlab-70332.html>)

**Cleve Moler, Chief Mathematician**



Recognition



2017 Computer History Museum Fellow Award



2014 IEEE John von Neumann Medal



2012 IEEE Computer Society Pioneer Award

**Jack Little, President**



**Fonte:** The Origins of MATLAB,  
[https://www.mathworks.com/company/newsletters/articles/the-origins-of-matlab.html?s\\_tid=srchtitle](https://www.mathworks.com/company/newsletters/articles/the-origins-of-matlab.html?s_tid=srchtitle)

18

# Linguagens de programação

- **Matlab (*Matrix Laboratory*)**

- Distribuidor:

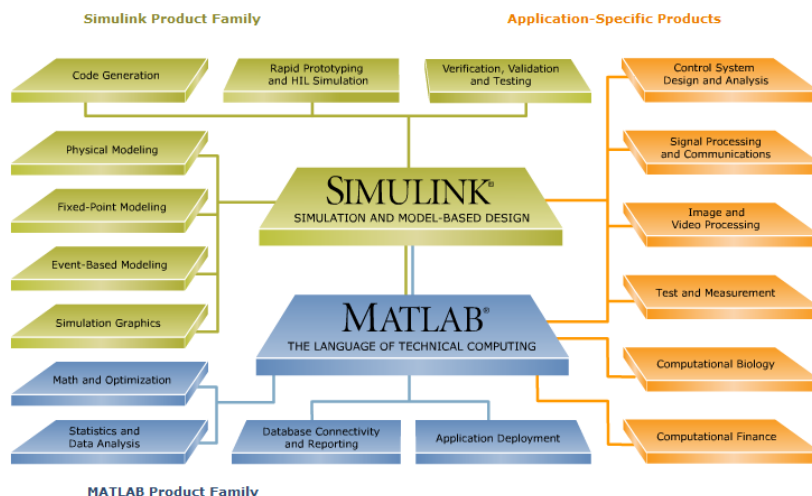
- Desenvolvedor: Mathworks  
(<http://www.mathworks.com>)
    - Distribuidor no Brasil: (<http://www.opencadd.com.br>)
    - Seminários gratuitos na internet :  
<http://www.opencadd.com.br/site/content/news/webinars/>
    - Infinitude de material gratuito na **Matlab Central**:  
<http://www.mathworks.com/matlabcentral/>



19

# Linguagens de programação

- Mais de 100 produtos (<https://www.mathworks.com/products/pfo.html>)



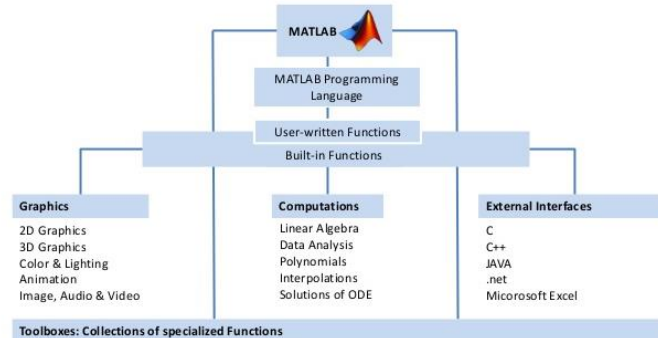
Exemplos diversos de funcionalidades do Matlab: <https://www.mathworks.com/examples/product-family/matlab>

20

## Famílias de produtos do Matlab

- **Programar em Matlab é:**

- Usar (**modificar**) inúmeras funções escritas pelos usuários, por você mesmo e pela Mathworks;
- Usar **interfaces** com outras linguagens, softwares e hardwares específicos (se necessário);
- Usar **Built-in Functions**: funções com código escondido (geralmente escritas em C++) que são parte do código base do Matlab ou de toolboxes.



Fonte: <http://www.slideshare.net/ashishmeet/introduction-to-matlab-18425069>

21

## Vantagens e desvantagens

- **Principais vantagens**

- Fácil de aprender a programar (linguagem de alto nível);
- Facilidade de manipulação gráfica (2D, 3D, imagem, vídeo);
- Código otimizado para realizar operações matriciais;
- Fácil identificação de erros (bugs);
- Vasta biblioteca de comandos (**toolboxes**);
- Implementação mais **fácil** e **rápida**;

- **Principais desvantagens**

- Desempenho em termos de tempo de processamento é inferior quando comparado a aplicações em C++ (**linguagem interpretada**);
- Má programação deixa o código muito lento (tempo de execução): o desempenho do código depende muito da programação ser baseada em operações matriciais (não se deve programar **like-C**);
- **Laços** de concatenados geralmente são extremamente lentos.

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

22

## Sistema de versionamento

- Sistema de controle de versões (ou versionamento):
  - **Outros nomes:** VCS (Version Control System) ou SCM (Source Code Management)
  - **Porque é importante?**
  - **Funcionalidade:** gerenciar diferentes versões (histórico de mudanças) no desenvolvimento de um documento (código) qualquer, seja por uma ou várias pessoas.
  - **Sistemas:**
    - Open source: CVS, Mercurial, Git e SVN;
    - Comerciais: SourceSafe, TFS (Microsoft), PVCS (Serena) e ClearCase (IBM).
  - **Tendência:** o **Git** (com repositórios no GitHub) vem substituindo o **SVN**, que por sua vez é um sucessor do **CVS**.

23

## Sistema de versionamento

- **Vantagens**
  - **Trabalho em equipe:** diversas pessoas trabalhar sobre o mesmo conjunto de documentos ao mesmo tempo, minimizando o desgaste provocado por problemas com conflitos de edições (**merge**);
  - **Controle do histórico e marcação e resgate de versões estáveis:** a maioria dos sistemas permite marcar (**tag**) onde é que o documento estava com uma versão estável, podendo ser facilmente resgatado no futuro;
  - **Ramificação de projeto (branch):** a maioria das implementações possibilita a divisão do projeto em várias linhas de desenvolvimento, que podem ser trabalhadas paralelamente, sem que uma interfira na outra. As mudanças podem ser agrupadas em um mesmo código (**merge**) de maneira automática.

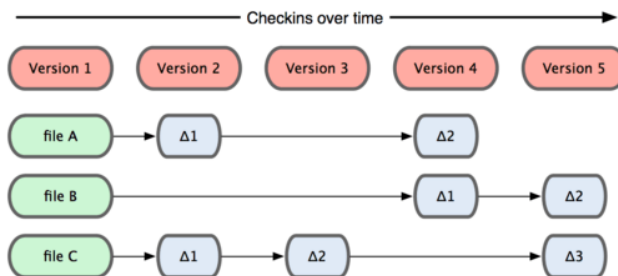
24

## Git e Github

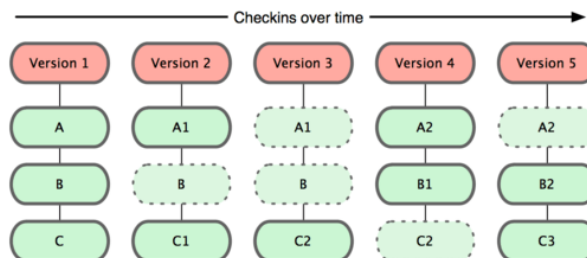
- **Git** é um sistema de controle de versão distribuído de código fonte (documentos em geral), open source com ênfase em velocidade.
  - **Alvo inicial:** desenvolvimento do kernel Linux, mas está sendo adotado por muitos outros projetos.
  - Cada diretório de trabalho do Git é um **repositório com um histórico completo** e habilidade total de acompanhamento das revisões, **não dependente de acesso a uma rede ou a um servidor central**.
- **GitHub** é uma plataforma de hospedagem para controle de versão e colaboração.

25

## Git vs outros sistemas



**Tradicional:**  
Armazena dados como mudanças em uma versão inicial de cada arquivo.



**GIT:** uma foto de todos os seus arquivos naquele momento é armazenada. Para arquivos sem mudanças, apenas um link para o arquivo idêntico anterior é armazenado.

26

## Sistema de versionamento

- **Principais operações**

- **Clone**, cópia local do repositório (com o git, todo o repositório é baixado para a máquina local);
- **Commit**, fase em que as alterações são armazenadas apenas localmente;
- **Push**, fase em que as alterações são enviadas para um servidor que concentra todos os conjuntos de alterações (chamados de *changesets*) que podem ser recombinaados entre si livremente.

- **Um bom guia sobre o git:**

[http://rogerdudler.github.io/git-guide/index.pt\\_BR.html](http://rogerdudler.github.io/git-guide/index.pt_BR.html)

(na primeira aula prática do curso será feito um treinamento sobre o git!!!)

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

27

## Sistema de versionamento

- **Curiosidade:** Quando perguntado sobre o porquê do nome, **Linus Torvalds** satirizou sobre o termo "Git" (uma gíria em inglês britânico para *cabeça dura*):

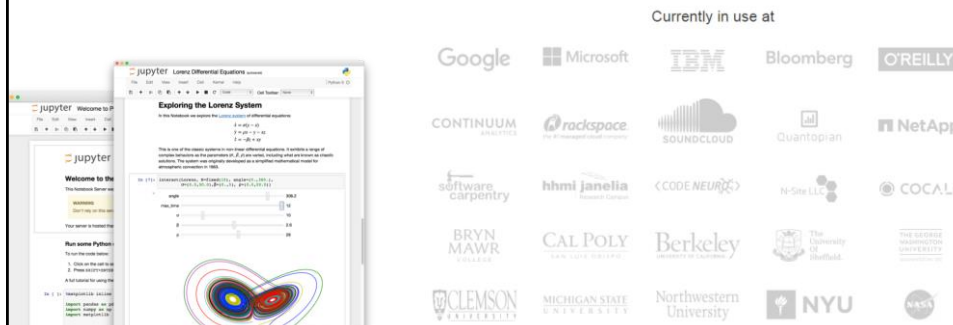
*“Eu sou um desgraçado egocêntrico, então batizo todos os meus projetos com meu nome. Primeiro Linux, agora Git.”*

**Linus Torvalds**

28

## Compartilhamento de material

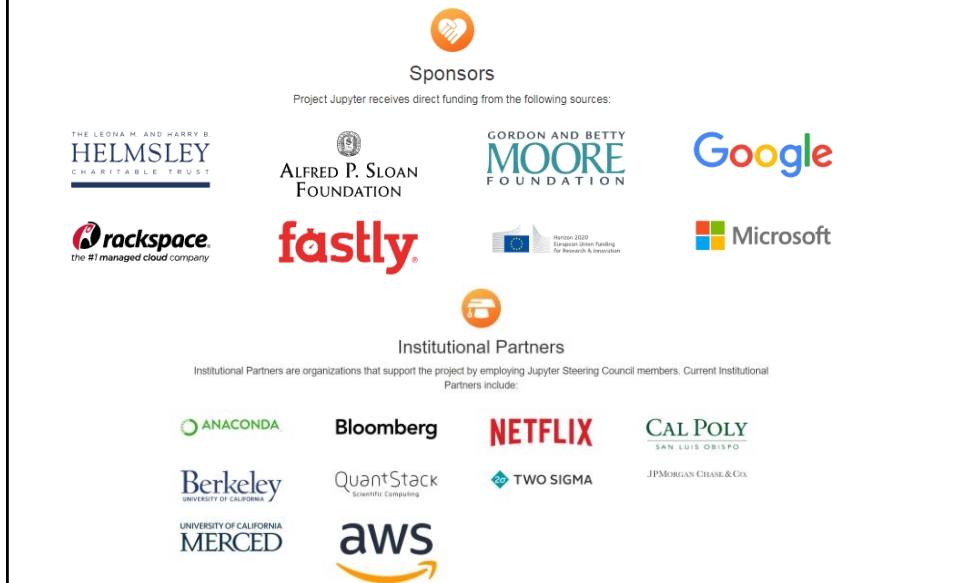
- O material da disciplina foi feito usando **Jupyter Notebook**;
  - Aplicação web de código aberto
  - Permite a criação e compartilhamento de documentos que contém código vivo, equações, gráficos e **(inclusive) texto explicativo**.



29

## Compartilhamento de material

- O projeto Jupyter começou em 2014



30

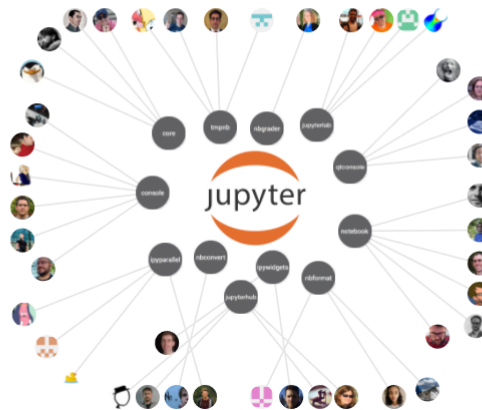
# Compartilhamento de material

- O projeto Jupyter começou em 2014



## Community

Our vibrant community empowers us to create an amazing platform. Dynamic developers, cutting edge scientists as well as everyday users work together to further Jupyter's best-in-class tools. Here is a dynamic visualization of our most recent contributors:



31

# Compartilhamento de material

- Novidade: JupyterLab 1.0: Jupyter's Next-Generation Notebook Interface

32



## Organização da Disciplina

- **UNIDADE I: Ramp-up e manipulação de sinais**
  - **Hands-on 00:** Preparação do ambiente de prática e desenvolvimento
  - **Hands-on 01:** Ferramentas de prototipagem (Matlab and Python Basics)
  - **Hands-on 02:** Visualização de sinais no tempo (vendo e ouvindo um sinal)
  - **Hands-on 03:** Visualização de sinais na frequência (gerando e vendo o espectro de um sinal)
  - **Hands-on 04:** Operação básicas com sinais (autocorrelação, cálculo de potência, energia e aplicações - remoção de eco com a autocorrelação)

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
 vicente.sousa@ufrn.edu.br

33

## Organização da Disciplina

- **UNIDADE II: Modulação e ruído**
  - **Hands-on 05:** Modulação Analógica (modulando e demodulando AM-DSB, AM-DSB-SC, AM-SSB, QAM, FM)
  - **Hands-on 06:** Transição analógico/digital (amostragem, reconstrução ideal, quantização, codificação PCM e multiplexação TDM)
  - **Hands-on 07:** Caracterização de variáveis aleatórias (PDF e CDF), Metodologia de Monte Carlo e Teorema Central do Limite
  - **Hands-on 08:** Ruído AWGN (caracterização e geração de amostras)
  - **Hands-on 09:** Modulação Digital (modulação, demodulação e desempenho em ruído AWGN)

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
 vicente.sousa@ufrn.edu.br

34

## Organização da Disciplina

- **UNIDADE III: Desvanecimento, OFDM e análise sistêmica**
  - **Hands-on 10:** Desvanecimento plano (caracterização e modelagem)
  - **Hands-on 11:** Modulação Digital em Canal com Desvanecimento
  - **Hands-on 12:** Modulação OFDM (ortogonalidade, transmissão e recepção, desempenho em canal AWGN)
  - **Hands-on 13:** Prototipagem sistêmica (modelagem de um sistema multiusuário multicelular, cálculo de interferência e SINR)

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

35

## Recursos

- **Utilizar a mesma máquina durante todo o curso**
- **Credenciais da máquina**
  - Login: labsim
  - Senha: dco2004
- **Bolsistas de apoio técnico (também alunos da disciplina)**
  - Lucas Ismael (lucasismael.gppcom@ufrn.edu.br)
  - Ravan Oliveira (ravan@ufrn.edu.br)
- **Horários de funcionamento do LabSim**
  - 8:00 as 22:15 hrs
- **Agendamento**  
eletrônico: <https://www.jotform.com/labsim/reserva>

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

36

## Metodologia de Ensino



- **Ensino**
  - Apresentação de conteúdo técnico de cada prática (15 a 30 minutos do início das aulas)
  - Aulas de laboratórios via hands-ons autocontidos
  - Experimentos em Matlab e Python
- **Canais de comunicação**
  - **Sala:** CTEC, sala 105
  - **Facebook:** <https://www.facebook.com/groups/246764072033399/>
  - **A qualquer momento:** turma virtual do SIGAA (notícias, enquetes, compartilhamento de material)
  - **Email:** [vicente.sousa@ufrn.edu.br](mailto:vicente.sousa@ufrn.edu.br)

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

37

## Metodologia de Acompanhamento da disciplina

- **Canal 01 (não o único): SIGAA** (notas, comunicados, frequência e alguns materiais didáticos);
- **Canal 02:** Outro canal (essencialmente importante como o SIGAA) serão repositórios no **gitHub**;
- O sistema operacional de trabalho será o **Linux (UBUNTU 14.04.05 LTS ou UBUNTU 18.04.02 LTS )**;
- Uma **máquina virtual** pode (deve) ser criada para o uso na disciplina!!! Sugiro o uso do VMWARE!!!



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

38

## Metodologia de Acompanhamento da disciplina

- O endereço (repositório Github) do material da disciplina é: <https://github.com/vicentesousa/DCO2004>
- Ele pode ser visualizada diretamente do seu Browser!!
- Para execução do código inline, você precisa **clonar** o repositório na sua máquina Linux.



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

39

## Metodologia de Avaliação

- **Avaliações**
  - Prova sem consulta (prova tradicional)
  - Relatórios das práticas
  - Projetos
- **Quarta prova**
  - Prova ou Projeto
- **Entrega dos projetos**
  - Via github



**As presenças serão contabilizadas com rigor e o critério de reprovação por falta será adotado como manda o regulamento de graduação.**

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

40

## Bibliografia (revisão teórica)



### Sistemas de Comunicações Analógicas e Digitais Modernos - 4ª Ed.

Autor: Lathi, B. P.  
Editora: Oxford



### Sistemas de Comunicação

Autor: Haykin, Simon  
Editora: Bookman

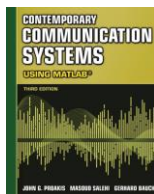


### Digital Communications: Fundamentals and Applications

Autor: Bernard Sklar  
Editora: Prentice Hall

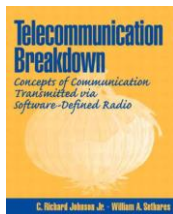
41

## Bibliografia (Experimentos)



### Contemporary Communication Systems Using MATLAB®

Autor: John G. Proakis, Masoud Salehi e Gerhard Bauch  
Edição: Third Edition  
Editora: CENAGE



### Telecommunication Breakdown Concepts of Communication Transmitted via Software-Defined Radio (How I Learned to Stop Worrying and Love the Digital Radio)

Autor: C. Richard Johnson Jr. e William A. Sethares  
Editora: Prentice Hall

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

42

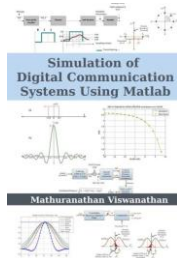
## Bibliografia (Experimentos)



### Software Receiver Design: Build your Own Digital Communication System in Five Easy Steps

**Autor:** C. Richard Johnson Jr, William A. Sethares e Andrew G. Klein

**Editora:** Cambridge University Press



### Simulation of Digital Communication systems using Matlab

**Autor:** Mathuranathan Viswanathan

**Edição:** Second edition

Autor do blog <http://www.gaussianwaves.com/>

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

43

## Sobre o Professor

### Vicente Angelo de Sousa Junior

- Nascimento: Crato, Ceará em 1977
- Formação em telecomunicações
  - Com fases na academia e na indústria



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

44

## Sobre o Professor

### Graduação em Engenharia Elétrica (ênfase: telecomunicações)

- **Instituição:** Departamento de Engenharia Elétrica da Universidade Federal do Ceará (UFC)
- **Data da colação:** 18 de Dezembro de 2000
- **Principais Realizações:**
  - **Iniciação científica:**
    - **Bolsa do CNPq:** Projeto WAVES (Brasil/Alemanha), Análise do Problema de Migração Populacional Usando Computação Simbólica, Laboratório de Computação Híbrida (OMNI), UFC.
    - **Bolsa do CNPq:** Desinfecção de Água Potável para Regiões Isoladas por Meio de Eletrólise de Cloro com Suprimento de Energia solar Fotovoltaica, Grupo de Processamento de Energia e Controle (GPEC), UFC.
  - **Bolsa PCT (Motorola/Instituto Eldorado):** Certificados em Técnicas de Comunicação Digital, Processamento Digital de Sinais, Telefonia Celular e Telefonia Digital.
- **Trabalhos apresentados:**
  - 2 artigos em encontro universitário de iniciação científica (UFC).

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

45

## Sobre o Professor

### Mestrado em Engenharia Elétrica (ênfase: telecomunicações)

- **Título:** Avaliação de Desempenho de Sistemas Multi-Serviço WCDMA com Antenas Inteligentes
- **Instituição:** Programa de pós-graduação em Engenharia Elétrica da Universidade Federal do Ceará (DEE-UFC)
- **Financiamento:** Conselho de Desenvolvimento Científico e Tecnológico (CNPq)
- **Data da Defesa:** 11 de Novembro de 2002
- **Principais Realizações:**
  - Artigos publicados:
    - 9 artigos em conferências internacionais (5 VTC, 1 ICT, 1 WNCG, 1 WWC, 1 ITS);
    - 4 artigos em conferência nacional (SBrT);
    - 1 artigo em periódico.
  - Criação do Grupo de Pesquisa em Telecomunicação Sem Fio (GTFL), sob coordenação do Prof. Dr. Francisco Rodrigo Porto Cavalcanti.
  - Convite do Prof. Doutor Elvino S. Sousa para ingressar como aluno de doutorado no *Wireless Communications Lab*, Departamento de Engenharia Elétrica e Computação da Universidade de Toronto.

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

46

## Sobre o Professor

### Doutorado em Engenharia de Telecomunicações

- **Título:** Seleção de Acesso e Gerência de Recursos de Rádio em Redes Multi-Acesso
- **Orientador:** Prof. Dr. Francisco Rodrigo Porto Cavalcanti
- **Instituição:** Programa de Pós-graduação em Engenharia de Teleinformática da Universidade Federal do Ceará (DETI-UFC)
- **Financiamento:** Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP) e Ericsson do Brasil
- **Data da Defesa:** 05 de Novembro de 2009
- **Principais Realizações:**
  - Artigos publicados:
    - 8 artigos em conferências internacionais (2 PIMRC, 5 ITS, 1 WWRF);
    - 1 capítulo de livro (Editora Springer);
    - 1 artigo de jornal (JCIS).
  - Citações:
    - 7 citações (ISWPC, ICTTA, ACM, Elsevier, CCECE)
  - Premiação:
    - PhD *Student Grant*: financiamento para a participação no 15th WWRF meeting, Paris, França (1000 Euros)

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

47

## Sobre o Professor



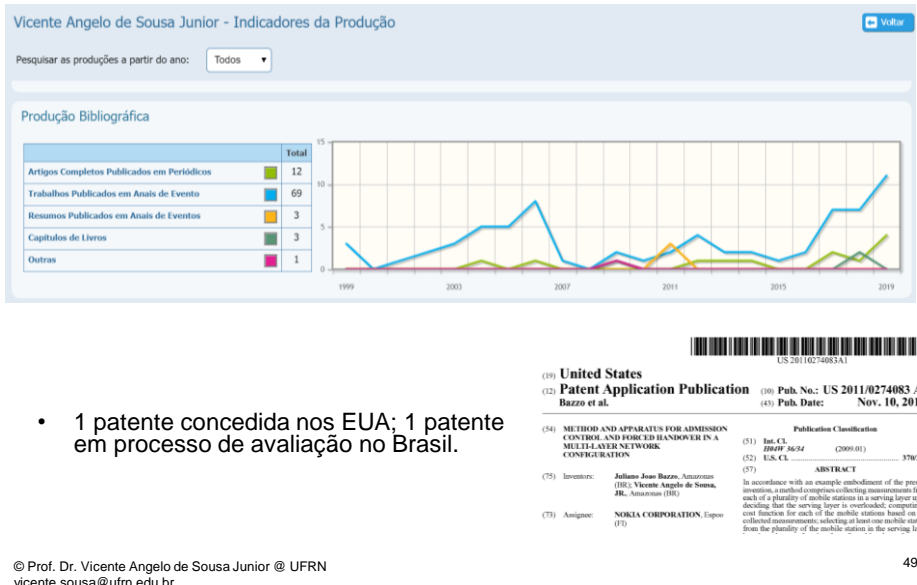
- 2000: Pesquisador Junior:
  - PCT-Motorola (Universidade Federal do Ceará - UFC, 2000).
- 2001-2006: Pesquisador sênior:
  - Cooperação Técnico-Científica entre a Ericsson do Brasil e a Universidade Federal do Ceará (UFC, GTEL, Ceará, 2001-2006).
- 2006-2010: pesquisador, Lider de P&D e gerente de projetos
  - Instituto Nokia de Tecnologia - INdT, Manaus, 2006-2010.
  - Simuladores em Matlab para: (i) desenvolvimento de novos produtos; (ii) padronização; (iii) pré-venda e auxílio ao núcleo de negócios.
- 2010-futuro: Professor
  - Universidade Federal do Ceará, 2003-2004.
  - Universidade Federal do Rio Grande do Norte, 2010-futuro
  - **Coordenador do Grupo de pesquisa: Grupo de Pesquisa em Prototipagem Rápida de Soluções para Comunicação Sem Fio (GppCom)**

© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

48



# Sobre o Professor



- 1 patente concedida nos EUA; 1 patente em processo de avaliação no Brasil.

(19) **United States**  
(12) **Patent Application Publication** (10) Pub. No.: **US 2011/0274083 A1**  
Bazzo et al. (43) Pub. Date: **Nov. 10, 2011**

(54) **METHOD AND APPARATUS FOR ADMISSION CONTROL AND FORCED HANDOVER IN A MULTI-LAYER NETWORK CONFIGURATION**

(51) Int. Cl. **H04W 36/52** (2009.01) 370/331  
(52) U.S. Cl. **370/331**

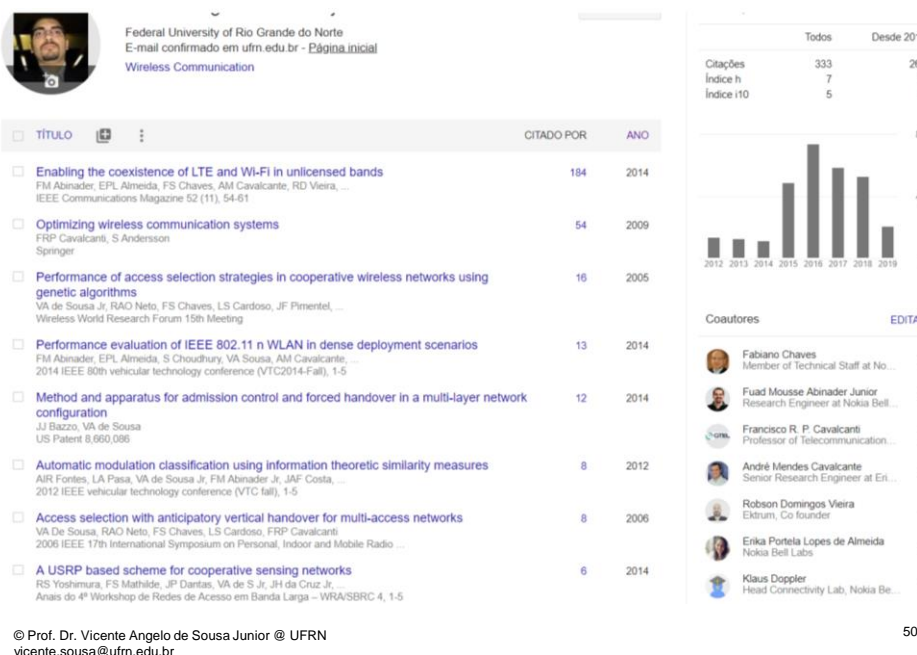
(75) Inventors: **Juliano Jean Bazzo, Accorsons (101); Vicente Angelo de Sousa, JR., Accorsons (101)**

(73) Assignee: **NOKIA CORPORATION, Espoo (FI)**

**ABSTRACT**  
In accordance with an example embodiment of the present invention, a method comprises collecting measurements from each of a plurality of mobile stations in a serving layer upon deciding that the serving layer is overloaded; computing a cost function for each of the mobile stations based on the collected measurements; selecting at least one mobile station from the plurality of the mobile stations in the serving layer

49

# Sobre o Professor (2019.2)



50

## Quais seus objetivos...

- ??????



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

51

## Para refletirmos...

- Feedback dos estagiários da UFRN (Telecom) – Ericsson, CPqD, COSERN, GVT...
  - LINUX
  - Programação (no mínimo para automatização de tarefas)
  - Gestão
    - Materiais
    - Pessoas
    - Existem disciplinas na UFRN com esse fim
  - Banco de dados
  - Comunicação e trabalho em equipe para execução de projetos
  - Certificações diversas (TI, programação e gestão)
- Grande lição dos alunos do final do curso
  - Aprender realmente e aplicar os conhecimentos aprendidos
  - Isso serve para vida corporativa e para as disciplinas do final do curso



© Prof. Dr. Vicente Angelo de Sousa Junior @ UFRN  
vicente.sousa@ufrn.edu.br

52