

IC_01 - Desenvolvimento de Software

Fase 1: Fundamentação Teórica + Prática Básica (Out-Nov/2025)

Dedicação: 12-15h/semana | **Abordagem:** 70% prática, 30% teoria

Semana 1-2: Introdução aos Biopotenciais com Python

O que fazer:

- Instalar Anaconda e configurar ambiente Python
- Primeiro contato com sinais sEMG usando dados simulados
- Plotar e visualizar sinais básicos

Como fazer:

python

```
# Tutorial prático - Primeiro contato
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as signal

# Gerar sinal sEMG simulado
def gerar_semg_simulado():
    t = np.linspace(0, 1, 1000) # 1 segundo, 1kHz
    # Sinal base + ruído
    semg = np.random.normal(0, 0.1, 1000) * np.sin(2*np.pi*50*t)
    return t, semg

t, sinal = gerar_semg_simulado()
plt.plot(t, sinal)
plt.title('Meu Primeiro Sinal sEMG')
plt.show()
```

Ferramentas:

- **Google Collab** (ambiente Python on-line da google)
- **Jupyter Notebook** (desenvolvimento interativo)
- **Spyder** (IDE científica)

Recursos de Estudo Simplificados:

• Monografia:

- ⇒ SILVA, João Vitor. **Análise comparativa de modelos de aprendizado de máquina para a classificação de movimentos finos de mão com sinais de eletromiografia de superfície**. 2025. f. Trabalho de Conclusão de Curso (Graduação em Engenharia Biomédica) – Universidade Federal de Uberlândia, Uberlândia, 2025.
 - ★ URL.: <https://repositorio.ufu.br/bitstream/123456789/46845/1/AnaliseTecnicasAPsEMG.pdf>
 - ★ URI: <https://repositorio.ufu.br/handle/123456789/46845>
- ⇒ VIANA, Paulo Augusto Alves Luz. **Identificação de movimentos da mão por machine learning utilizando eletromiografia de superfície**. 2019. Trabalho de Conclusão de Curso (Graduação) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2019.
 - ★ URL.: https://bdta.abcd.usp.br/directbitstream/4c87b904-528b-4050-b01d-7617eb527e45/Viana_PauloAugusto_tcc.pdf

- **Artigos:**

- ⇒ Makowski, D., Pham, T., Lau, Z.J. et al. **NeuroKit2: A Python toolbox for neurophysiological signal processing**. Behav Res 53, 1689–1696 (2021). <https://doi.org/10.3758/s13428-020-01516-y>.
 - ★ URL: <https://link.springer.com/content/pdf/10.3758/s13428-020-01516-y.pdf>
- ⇒ Patrícia Bota, Rafael Silva, Carlos Carreiras, Ana Fred, Hugo Plácido da Silva. **BioSPPy: A Python toolbox for physiological signal processing**, SoftwareX, Volume 26, 2024, ISSN 2352-7110, <https://doi.org/10.1016/j.softx.2024.101712>.
 - ★ URL: <https://www.sciencedirect.com/science/article/pii/S2352711024000839>
- ⇒ Appriou, A.; Pillette, L.; Trocellier, D.; Dutartre, D.; Cichocki, A.; Lotte, F. **BioPyC, an Open-Source Python Toolbox for Offline Electroencephalographic and Physiological Signals Classification**. Sensors 2021, 21, 5740. <https://doi.org/10.3390/s21175740>
 - ★ URL: <https://www.mdpi.com/1424-8220/21/17/5740>

- **Outros:**

- ⇒ DE LIMA, Gabriel Molina; CAMPOS, Daniel Prado; MANTOVANI, Rafael Gomes. **A Review on the Recent use of Machine Learning for Gesture Recognition using Myoelectric Signals**. In: ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL E COMPUTACIONAL (ENIAC), 21. , 2024, Belém/PA. Anais (...). Porto Alegre: Sociedade Brasileira de Computação, 2024 . p. 180-191. ISSN 2763-9061. DOI: <https://doi.org/10.5753/eniac.2024.245071>.
 - ★ URL: <https://sol.sbc.org.br/index.php/eniac/article/view/33792>
- ⇒ Kok, C.L.; Ho, C.K.; Tan, F.K.; Koh, Y.Y. **Machine Learning-Based Feature Extraction and Classification of EMG Signals for Intuitive Prosthetic Control**. Appl. Sci. 2024, 14, 5784. <https://doi.org/10.3390/app14135784>.
 - ★ URL: <https://www.mdpi.com/2076-3417/14/13/5784>

Entregável Semanal:

- Notebook Jupyter/**Collab** com 5 exemplos de sinais simulados
- Gráficos básicos comentados

Semana 3-4: Filtros Digitais na Prática

O que fazer:

- Implementar filtros passa-baixa, passa-alta e passa-banda
- Testar com sinais reais de bases públicas
- Comparar antes e depois da filtragem

Como fazer:

python

```
# Tutorial prático - Filtros básicos
from scipy.signal import butter, filtfilt

def filtro_passa_banda(sinal, fs=1000, low=20, high=500):
    # Projeto do filtro Butterworth
    nyquist = fs / 2
    low_norm = low / nyquist
    high_norm = high / nyquist

    b, a = butter(4, [low_norm, high_norm], btype='band')
    sinal_filtrado = filtfilt(b, a, sinal)
    return sinal_filtrado

# Exemplo de uso
sinal_limpo = filtro_passa_banda(sinal_ruidoso)
```

Ferramentas:

- **SciPy** (scipy.signal para filtros)
- **Base de dados:** EMG Database (Kaggle)
- **Simulador online:** DSP Filter Design Tool

Recursos de Estudo Simplificados:

- **Tutorial Video:**

⇒ **Filtros digitais em Python**

★ URL: <https://www.youtube.com/watch?v=lzPWF0N3zhM>

- **Livro:**

⇒ MACIEL, Carlos Dias. **Processamento de sinais**. Universidade de São Paulo. Escola de Engenharia de São Carlos, 2022. DOI: <https://doi.org/10.11606/9786586954180>

★ URL: www.livrosabertos.abcd.usp.br/portaldelivrosUSP/catalog/book/928

- **Dissertação:**

⇒ CAMPOS, Daniel Prado de. **Avaliação do comportamento ingestivo utilizando eletromiografia de superfície do músculo masseter**. 2016. 161 f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Tecnológica Federal do Paraná, Pato Branco, 2016.

★ URL: <http://repositorio.utfpr.edu.br/jspui/handle/1/1756>

- **Tutorial:** PySDR: A Guide to SDR and DSP using Python - <https://pysdr.org/content/intro.html>

Entregável Semanal:

- 3 tipos de filtros implementados e testados
- Comparativo visual antes/depois
- Análise espectral básica