

# CEIA-POSIA04 - Meta-heurísticas e Otimização Combinatória

Danilo Sipoli Sanches

Departamento Acadêmico de Computação  
Universidade Tecnológica Federal do Paraná  
Cornélio Procópio



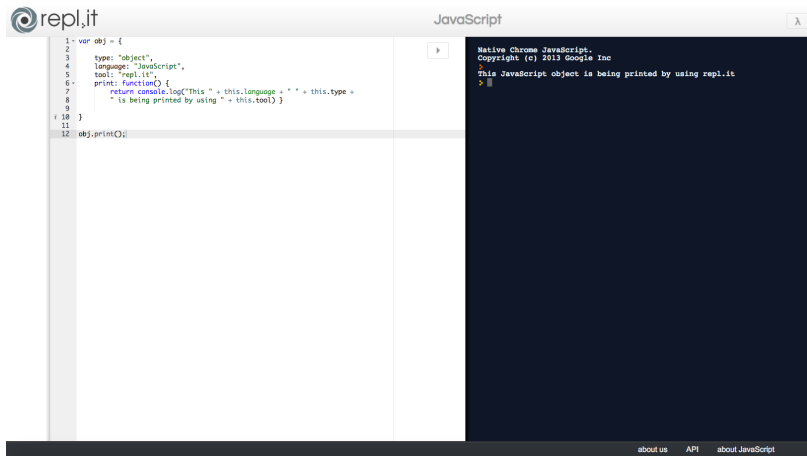
- Fornecer aos alunos uma forte base na área de Computação Evolutiva, abordando principalmente métodos de projeto e otimização de desempenho de Algoritmos Evolutivos;
- Pretende-se ainda, apresentar uma visão geral que reflita a situação atual da área, abordando os conceitos fundamentais; aplicações em diferentes problemas de otimização e também do mundo real;
- Serão abordados bibliotecas e frameworks em Python.

# O que vamos aprender?

- ① Resolução de problemas por busca
- ② Algoritmos de busca local:
  - Subida de Encosta e Recozimento Simulado
- ③ Algoritmos Genéticos:
  - representação;
  - operadores de reprodução e seleção;
  - controle de parâmetros
- ④ Variações de Algoritmos Evolutivos;
- ⑤ Frameworks Python;
- ⑥ Aplicações,

# Interpretador Online Python

<https://repl.it/>



The screenshot shows the repl.it online JavaScript interpreter interface. The top bar features the repl.it logo on the left, the text "JavaScript" in the center, and a small icon on the right. The main area is split into two panels. The left panel is a code editor with a light gray background, showing a JavaScript code snippet with line numbers 1 through 12. The code defines an object and a function to log information about the object. The right panel is a dark-themed console area showing the output of the code execution. The output text reads: "Native Chrome JavaScript. Copyright (c) 2013 Google Inc. This JavaScript object is being printed by using repl.it". At the bottom of the interface, there is a dark gray bar with links for "about us", "API", and "about JavaScript".

```
1 var obj = {  
2   type: "object",  
3   language: "JavaScript",  
4   tool: "repl.it",  
5   print: function() {  
6     return console.log("This " + this.language + " " + this.type +  
7       " is being printed by using " + this.tool) }  
8 }  
9  
10  
11  
12 obj.print();
```

Native Chrome JavaScript.  
Copyright (c) 2013 Google Inc.  
This JavaScript object is being printed by using repl.it  
>

[about us](#) [API](#) [about JavaScript](#)

[https://https://cocalc.com/](https://cocalc.com/)



Collaborative Calculation in the Cloud

<https://colab.research.google.com/>

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

Share Sign in

Table of contents

- Getting started
- Data science
- Machine learning
- More Resources
- Machine Learning Examples
- Section

## What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

### Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ] seconds_in_a_day = 24 * 60 * 60
    seconds_in_a_day
```

86400

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

Variables that you define in one cell can later be used in other cells: