



# Desenvolvimento de Software para Nuvem

Prof. Dr. Fernando Antonio Mota Trinta  
Prof. Dr. Paulo Antonio Leal Rego



**Uma plataforma de desenvolvimento e marketplace para  
algoritmos**

## História

- ❑ A startup Algorithmia surgiu em 2014 e levantou cerca de \$2.4M para criar uma Marketplace para algoritmos.
- ❑ O objetivo era conectar desenvolvedores do mundo inteiro e pesquisadores/academia.
- ❑ Em 2017, conseguiram \$10.5M para focar no desenvolvimento de serviços para machine learning.
- ❑ Hoje, a plataforma Algorithmia é executada sobre a AWS, Azure e Google Cloud.

## Algorithms as a Service

- ❑ Começou como uma Marketplace para algoritmos.
- ❑ Depois evoluiu para uma plataforma de treinamento de modelos e provisão de serviços de aprendizado de máquina.
- ❑ Características:
  - ❑ Auto scaling, rápido deploy usando microserviços, uso otimizado de GPUs, colaborativo, pagamento baseado no uso
- ❑ Hoje: mais de 6700 algoritmos e 70000 desenvolvedores

## Qual a ideia?

- ❑ Desenvolvedores disponibilizam seus algoritmos para terceiros e ganham \$ a cada requisição recebida.
  1. Prepara o código fonte
  2. Envia o código fonte para o Algorithmia
  3. Algorithmia encapsula o algoritmo em um microserviço
  4. Algorithmia disponibiliza o acesso ao algoritmo através de um Webservice
  5. Algorithmia cuida da elasticidade automática do algoritmo

# Tipos de Algoritmos



## Text Analysis

Make sense of unstructured text



## Machine Learning

Teach your app to teach itself



## Computer Vision

Identify objects in images



## Deep Learning

Learn from your data

<https://algorithmia.com/algorithms>

# Alguns dos frameworks de machine learning suportados



Caffe



Chainer



Dlib



Gensim



OpenCV



PMML



PyTorch



Rusty Machine



H2O.ai



Keras



MXNet



NLTK



Scikit-Learn



TensorFlow



Theano



Weka

## Algumas das linguagens de programação suportadas



Java



JavaScript



Python



R



Ruby



Rust



Scala



# Clientes suportados



.NET

.NET / C#



AWS Lambda



Android



CLI



Node



PHP



Perl



Python



Swift



CURL



Go



Java



Javascript



R



Ruby



Rust



Scala

## Preço

- ❑ O custo de cada chamada depende do tempo de computação:
  - ❑ 1 crédito por segundo + royalties (a critério do autor)
  - ❑ \$1 = 10000 créditos
- ❑ Ao criar a conta, recebe 50000 créditos grátis.
- ❑ Todo mês, recebe mais 5000 créditos (não acumula).
- ❑ O autor pode transformar o dinheiro ganho em crédito ou sacar ao acumular \$100.

<https://algorithmia.com/pricing>

## Existem dois tipos de planos

- ❑ Serverless AI Layer
  - ❑ Hospedado na Algorithmia
- ❑ Enterprise
  - ❑ Hospedado na nuvem do cliente
    - ❑ Amazon AWS
    - ❑ Google
    - ❑ Azure
    - ❑ Nuvem privada

<https://algorithmia.com/pricing>



## Hands-on

# Criar conta

## Start on our free plan

- ✓ Serverless microservices utilize GPUs or CPUs
- ✓ Pay only for compute time on your models
- ✓ Simple and cost effective for scaling to any size
- ✓ Add AI to any application with ease

## Here's 50,000 credits to get you started

Plus, you will receive 5,000 complimentary credits every month.

**CREATE YOUR FREE ACCOUNT**

## ALGORITHMIA

Create Your Account

USERNAME

aluno

EMAIL

aluno@ufc.br

PASSWORD

.....

PROMO CODE (OPTIONAL)

Create Your Account

## Conta

- ❑ É possível acessar a conta via CLI ou Console Web
- ❑ É possível criar uma Organização e adicionar usuários
- ❑ Toda conta tem uma ou mais API Keys - necessárias para executar os algoritmos

# Console

☰

ALGORITHMIA

AI LAYER


🏠 Home

🔍 Algorithms

📊 Data Sources

📄 Docs →

[Download](#) Whitepaper: The Roadmap to ML Maturity—[Download!](#) ×

 **pauloalr** 64k credits

pauloalr ★ 10 ↑ 0 ↓ 62

[Create New](#) ▾

Overview

My Algorithms

Organizations

API Keys

Account

Settings

My algorithms

[View All](#)

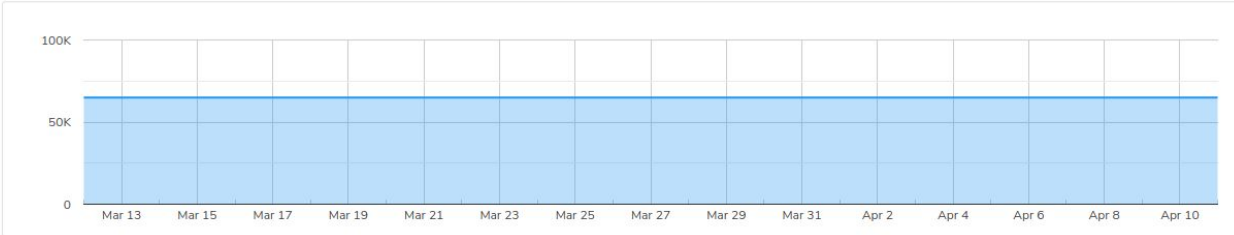
[Hello World](#)

A simple "Hello World" algorithm, which returns "Hello " + the algorithm input.

pauloalr ↓ <1k

Account balance

[Go To Account](#)



Date	Balance (Credits)
Mar 13	64,000
Mar 15	64,000
Mar 17	64,000
Mar 19	64,000
Mar 21	64,000
Mar 23	64,000
Mar 25	64,000
Mar 27	64,000
Mar 29	64,000
Mar 31	64,000
Apr 2	64,000
Apr 4	64,000
Apr 6	64,000
Apr 8	64,000
Apr 10	64,000

# Gerenciamento de Keys

- ❑ É preciso definir:
  - ❑ Label
  - ❑ Algoritmos que podem ser acessados:
    - ❑ algo:/// \* libera tudo
  - ❑ Tipo de acesso
  - ❑ ...

## Create a new simple key



API keys are how you authenticate with the platform, allowing you to call algorithms and interact with the data API.

KEY LABEL

ALGORITHM ACCESS (KEY CAN ONLY CALL)

 Add

algo:/// \*



ALLOW CALLING ALGORITHMS FROM

- ☒ Native clients (curl, java, etc..)
- ☐ Web browser (Javascript CORS)

DATA ACCESS

- ☒ No Access
- ☐ Read Only
- ☐ Read & Write

MANAGEMENT APIS Danger Zone

- ☐ Allow this key to manage my algorithms

This option controls whether this key can be used to call algorithm management endpoints on your behalf, thereby allowing the creation, querying, updating, and publishing of your algorithms.

Create Simple Key



# Algoritmos

- ❑ Cada algoritmo tem:
  - ❑ dono, nome, versão, custo, ...

 [sfw](#) / NudityDetection / 1.0.5 

Run Docs **Cost** Discussion

## README.MD

This algorithm detects nudity in pictures. For a demo of this algorithm, check out: <https://isitnude.com/>

### Idea:

The idea behind the algorithm is based primarily on observations that in general, nude images contain large amounts of skin, people have different skin tones, and skin regions in nude images are relatively close to each other. In order to make the algorithm more robust, we have incorporated face detection for skin ratio tweaking and skin color detection for limiting the generic skin color value interval.

 [sfw](#) / NudityDetection / 1.0.5 

Run Docs **Cost** Discussion

All API calls charge based on **compute time** of the algorithm (1 credit per usage second) plus a **royalty per call** (if the author charges a royalty).

For more details please visit the [pricing docs](#) or the [pricing FAQs](#).

This algorithm has permission to call other algorithms which may incur separate royalty and usage costs.

## Estimate Pricing

ROYALTY PER CALL	COST PER SECOND
No royalty	1 credit

API CALL DURATION (SEC)

1

X

API CALLS

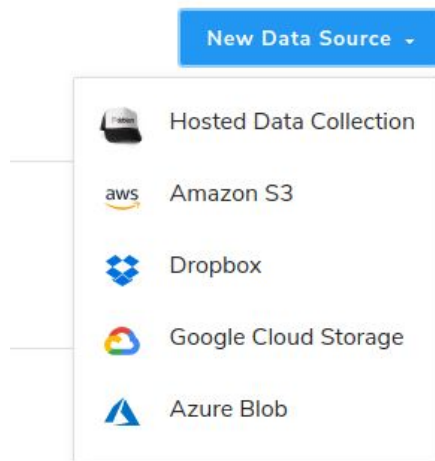
10000

ESTIMATED COST PER 10,000 CALLS

**10k credits (\$1.00 USD)**

## Data Sources

- ❑ Os algoritmos podem acessar dados de Data Sources
  - ❑ Conjunto de Coleções, que armazenam arquivos
- ❑ Pode conectar outras fontes de dados:
  - ❑ Dropbox, Amazon S3, outros



## Executando nosso primeiro algoritmo

### ❏ Hello World

#### ❏ CURL

- ❏ `curl -X POST -d '"pauloalr"' -H 'Content-Type: application/json' -H 'Authorization: Simple sim+MINHAKEY' https://api.algorithmia.com/v1/algo/demo/Hello/`
- ❏ Saída:

```
{  
  "result": "Hello pauloalr",  
  "metadata": { "content_type": "text", "duration": 0.000458119}  
}
```

# Executando nosso primeiro algoritmo

## ❑ Hello World

### ❑ Python

❑ pip install algorithmia

❑ Código:

```
import Algorithmia
input = "pauloalr"
client = Algorithmia.client('sim+MINHAKEY')
algo = client.algo('demo/Hello/')
print(algo.pipe(input).result)
print(algo.pipe(input).metadata)
```

Saída:

```
Hello pauloalr
Metadata(content_type='text',
duration=0.000490849,stdout=None)
```

# CLI

- ❑ Instalação/Configuração:
  - ❑ `curl -sSLf https://algorithmia.com/install.sh | sh`
  - ❑ Se for usar Python, criar a variável de ambiente:
    - ❑ `export LANGUAGE_VERSION=python2` ou
    - ❑ `export LANGUAGE_VERSION=python3`
  - ❑ Configurar autenticação:
    - ❑ `algo auth`
    - ❑ Entrar com API Endpoint e API Key.
  - ❑ Testar:
    - ❑ `algo ls`

<https://algorithmia.com/developers/clients/cli>

# CLI

## ❑ Executar algoritmos

### ❑ SentimentAnalysis

```
algo run nlp/SentimentAnalysis/1.0.5 -d '{  
  "document": "I really like Algorithmia!"  
}' --timeout 300
```

### ❑ Prime factors

```
algo run kenny/factor -d 1531530
```

# CLI

## ❑ Data API

- ❑ algo <comando>

- ❑ Comandos:

- ❑ ls: lista o conteúdo do diretório List contents of a data directory
- ❑ mkdir: criar um diretório
- ❑ rmdir: remover um diretório
- ❑ rm: remover arquivos de um diretório
- ❑ cp: copiar arquivos para/de um diretório
- ❑ cat: concatenar e imprimir o conteúdo de arquivos

<https://algorithmia.com/developers/clients/cli>

# CLI

## ❏ Data API - Exemplos

- ❏ algo <comando>

- ❏ Comandos:

- ❏ \$ algo mkdir .my/fotos

- ❏ \$ algo cp brasaoufc.png data://.my/fotos

- ❏ \$ algo cat data://pauloalr/txt/livro.txt

<https://algorithmia.com/developers/clients/cli>



# Criar Algoritmo

## Create a new algorithm



Put your work on the AI Layer and get all the benefits of scalable microservices available through a simple API.

OWNER

pauloalr

ALGORITHM NAME

algo://pauloalr/AlgorithmURL

DESCRIPTION

Brief description of what the algorithm does. Optional but recommended.

Select a language ⓘ



Java



Javascript



Python 2.x  
- Beta



Python 2.x



Python 3.x  
- Beta



Python 3.x



R



Ruby



Rust



Scala

# Criar Algoritmo

## Select Properties

Properties can always be changed later in the algorithm's settings tab.

### SOURCE CODE ⓘ

- ☒ Open source
- ☐ Closed source

### LICENSE

Algorithmia Platform License ▼

## Permissions

Permissions can always be changed later in the algorithm's settings tab.

### INTERNET ⓘ

- ☐ Internet access
- ☒ No internet access

### PIPELINING ⓘ

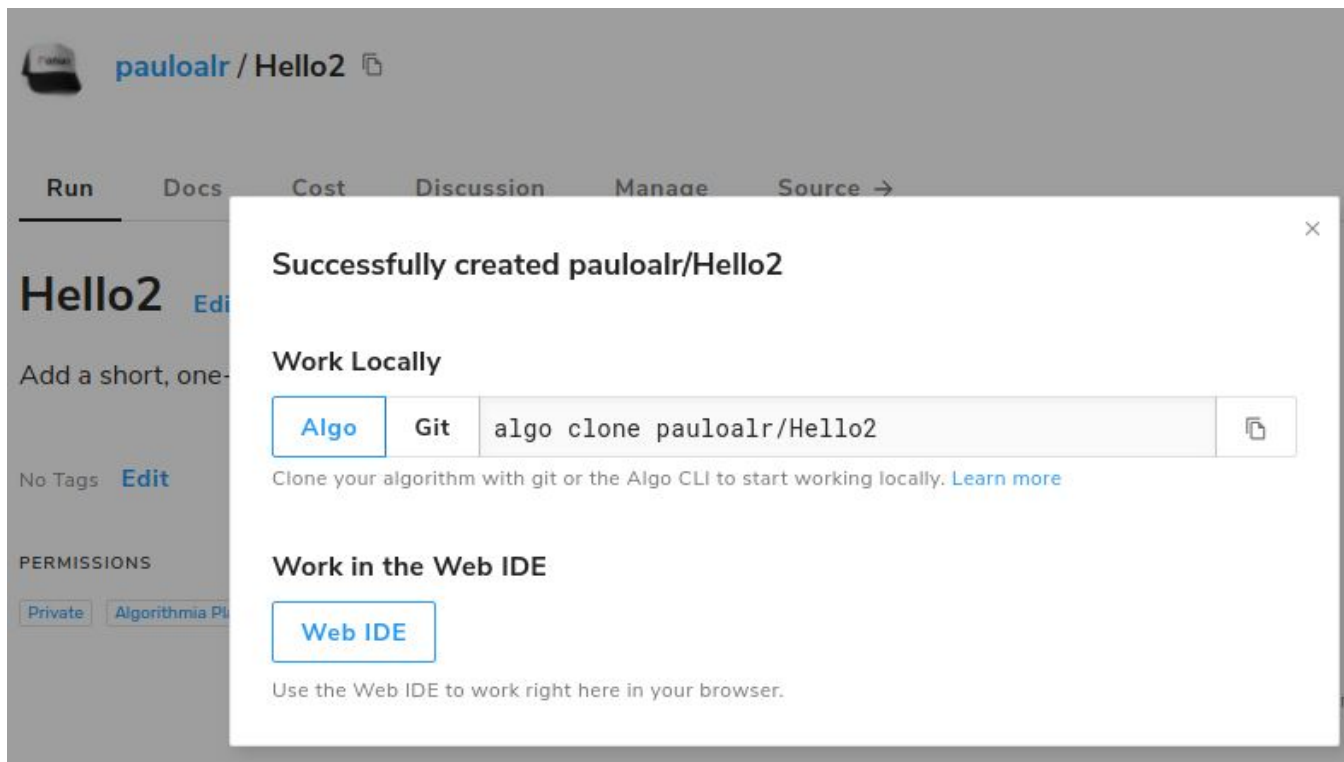
- ☐ Can call other algorithms
- ☒ Not allowed to call other algorithms

### HARDWARE ⓘ


- ☐ This algorithm requires CUDA/GPU

Create New Algorithm

# Criar Algoritmo



# Criar Algoritmo



```
1  import Algorithmia
2
3  # API calls will begin at the apply() method, with the request body passed as 'input'
4  # For more details, see algorithmia.com/developers/algorithm-development/languages
5  def apply(input):
6      return ("hello {}".format(input)).upper()
7
```

Save --> Build --> Publish

# Criar Algoritmo

## Publishing pauloalr/Hello2

Changes **Sample I/O** Versioning

### Sample Input

```
Paulo
```

▶ RUN

### Sample Output

```
"HELLO PAULO"
```

**Sample input** is often a user's first experience running your algorithm. Ensuring you have clear, simple, working sample input is important to the first experience for users who find your algorithm.

◀ PREVIOUS

CANCEL

NEXT

# Criar Algoritmo

## Publishing pauloalr/Hello2

[Changes](#)[Sample I/O](#)[Versioning](#)

### Callability

- ☐ Public (Anyone can call this version)
- ☒ Private (Only you can call this version)

### Pricing [Help me understand pricing](#)

 cr/call

You will receive 70% of the royalty cost: **7.00cr** per call.

### Semantic Versioning

#### What version should this be published as?

- ☐ 1.0.0 (Major: for API breaking changes)
- ☒ 0.1.0 (Minor: for backward-compatible features)
- ☐ 0.0.1 (Revision: n/a when changing price or permissions)

#### About Callability

**Callability** affects who can view and call your algorithm. Algorithms with at least one public version are listed publicly on the platform, and cannot be deleted.

#### About Versioning

A **major version** represents breaking changes to your algorithm. A major version may break API compatibility.

A **minor version** represents a backwards-compatible feature change. Minor versions may include pricing changes.

A **revision** represents a backwards-compatible bug fix to your algorithm. Revisions may not incorporate a pricing change.

[◀ PREVIOUS](#)

Publishing pauloalr/Hello2

**As Version: 0.1.0**[CANCEL](#)[PUBLISH](#)