

# Tutorial do framework Flask

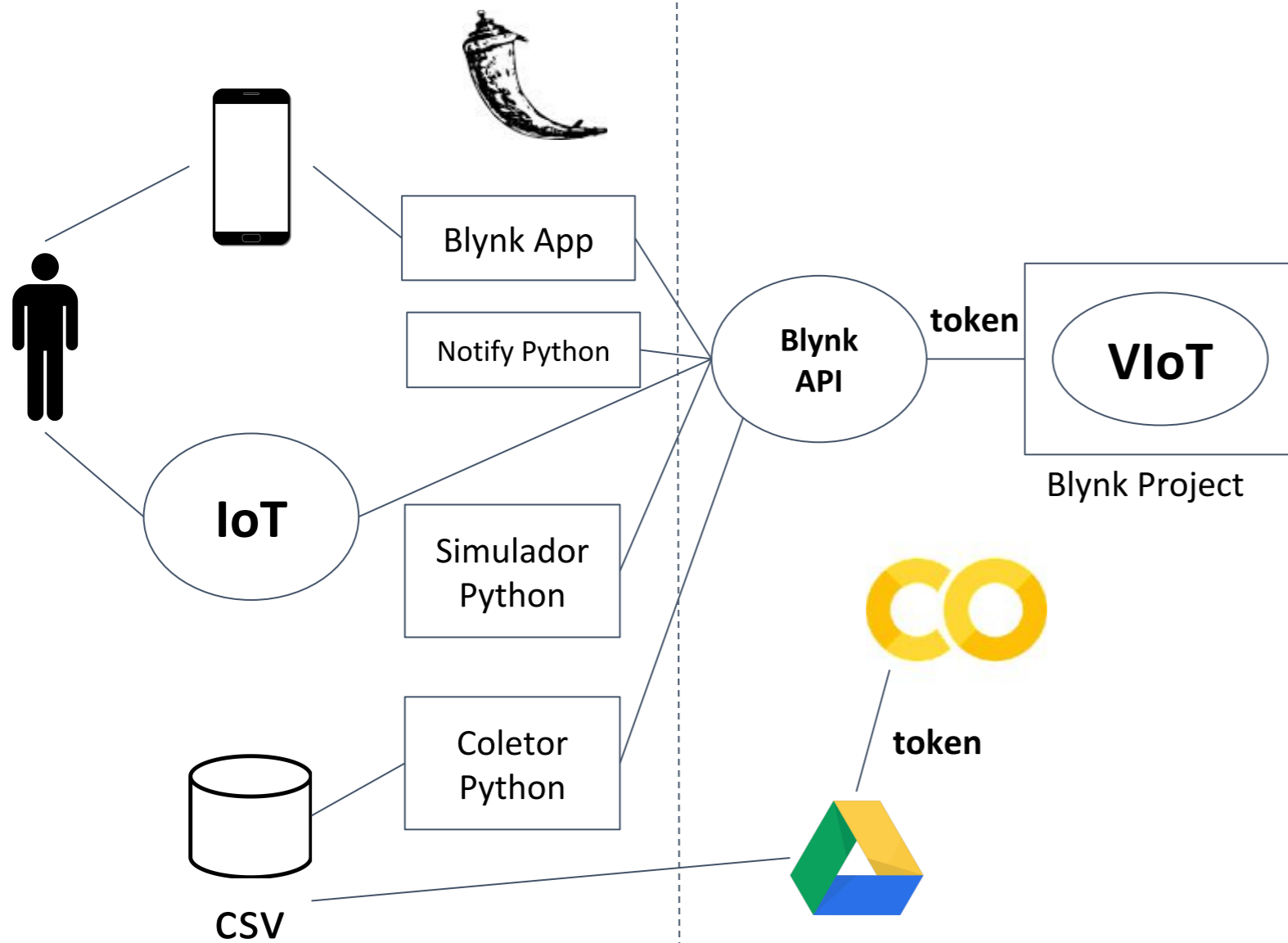
PCS3100

Prof. Reginaldo Arakaki  
Victor Hayashi  
Tiago Fujii

# Arquitetura

LOCAL

CLOUD



# Tutoriais

1. Blynk: Cloud IoT com interface Mobile
  - a. Mobile App Android/iOS + HTTP request
  - b. Simular comportamento IoT com Python
  - c. Usar API para notificação no celular
2. Coleta de Dados
  - a. Coletor Python
  - b. Ambiente Google Colab e Google Drive
3. Servidor Local e Tunelamento
  - a. Servidor local no Flask
  - b. Tunelamento no ngrok

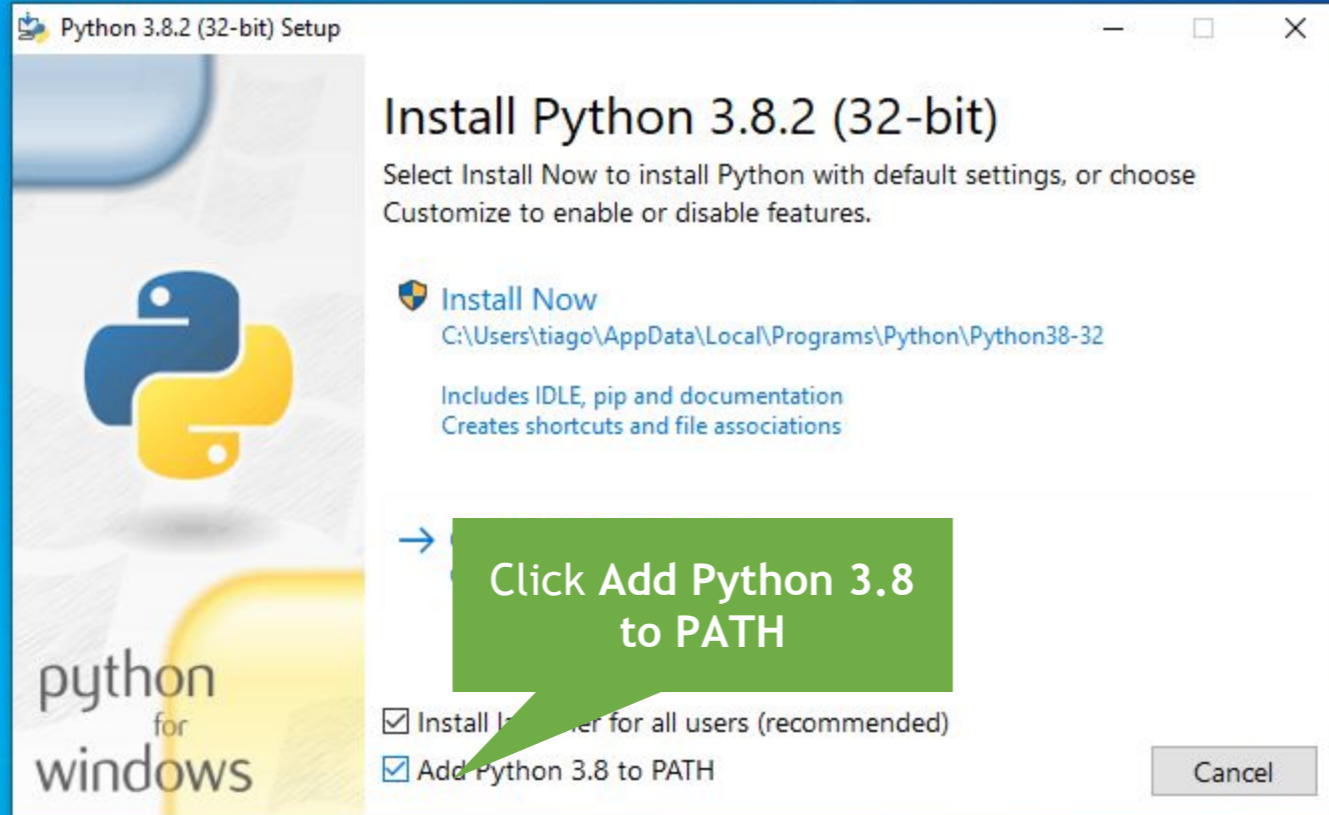
# Características/vantagens do Flask

- Microframework web para Python
- Permite criar páginas e aplicações web
- Caixa branca – maior liberdade de desenvolvimento
- Menos monolítico que Django
- Microserviços

# Pré-requisito: Instalação do Python

Instalar Python3: <https://www.python.org/downloads/>

(Obs: Para seguir este tutorial, adicionar o Python ao PATH do Windows para executar os comandos do Python no terminal)



# Instalar Flask

Abrir um prompt de comando e digitar o comando:

```
pip install Flask
```

C:\Users\tiago>p\_

pip install Flask

```
C:\Users\tiago>pip install Flask
```

```
Collecting Flask
```

```
Using cached https://files.pythonhosted.org/packages/9b/93/628509b8d5dc749656a9641f4caf13540e2cdec85276964ff8f43bbb1d3b/Flask-1.1.1-py2.py3-none-any.whl
```

```
Requirement already satisfied: Werkzeug>=0.15 in c:\users\tiago\appdata\local\programs\python\python38-32\lib\site-packages (from Flask) (1.0.1)
```

```
Requirement already satisfied: click>=5.1 in c:\users\tiago\appdata\local\programs\python\python38-32\lib\site-packages (from Flask) (7.1.1)
```

```
Requirement already satisfied: itsdangerous>=0.24 in c:\users\tiago\appdata\local\programs\python\python38-32\lib\site-packages (from Flask) (1.1.0)
```

```
Requirement already satisfied: Jinja2>=2.10.1 in c:\users\tiago\appdata\local\programs\python\python38-32\lib\site-packages (from Flask) (2.11.1)
```

```
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\tiago\appdata\local\programs\python\python38-32\lib\site-packages (from Jinja2>=2.10.1->Flask) (1.1.1)
```

```
Installing collected packages: Flask
```

```
Successfully installed Flask-1.1.1
```

```
WARNING: You are using pip version 19.2.3, however version 20.0.2 is available.
```

```
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

```
C:\Users\tiago>_
```



# Hello World

**Criar arquivo index.py**  
**Copiar o seguinte conteúdo no arquivo:**

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"

if __name__ == "__main__":
    app.run()
```

→ Cria um callback para a função imediatamente a seguir (hello) quando o sub-domínio URL "/" for requisitado

**Salvar arquivo index.py na pasta Documentos**

Microsoft Windows [versão 10.0.18363.720]  
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\tiago>cd Documents

C:\Users\tiago\Documents>\_



cd Documents

```
C:\Users\tiago>cd Documents
```

```
C:\Users\tiago\Documents>set FLASK_APP=index.py
```

```
C:\Users\tiago\Documents>
```

set FLASK\_APP=index.py

Obs: no Linux, utilizar: export FLASK\_APP=index.py



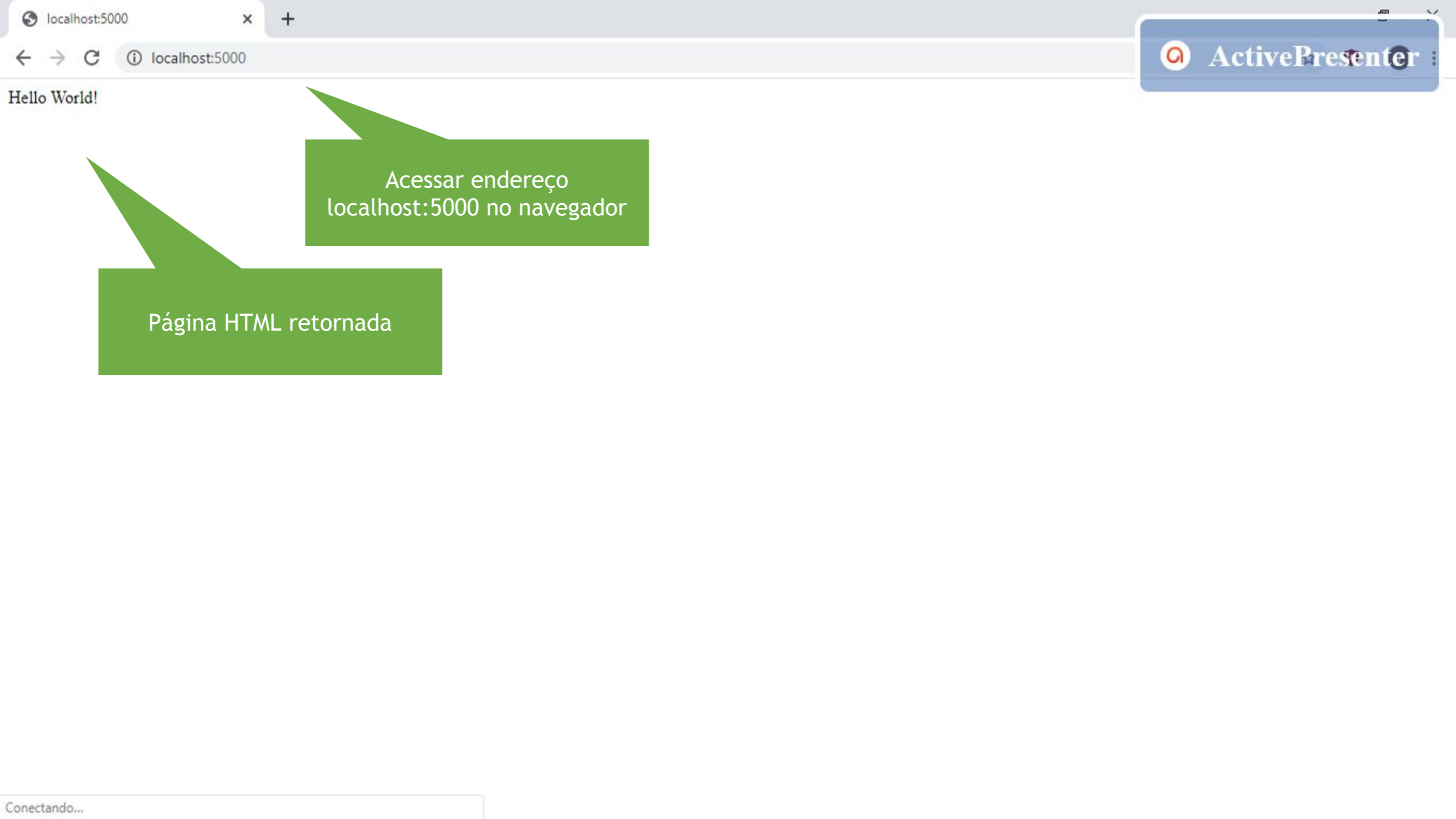
```
C:\Users\tiago>cd Documents
```

```
C:\Users\tiago\Documents>set FLASK_APP=index.py
```

```
C:\Users\tiago\Documents>flask run  
* Serving Flask app "index.py"  
* Environment: production  
WARNING: This is a development server. Do not use it in a production deployment.  
Use a production WSGI server instead.  
* Debug mode: off  
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

flask run

Servidor local executando na porta 5000



Hello World!

Acessar endereço localhost:5000 no navegador

Página HTML retornada

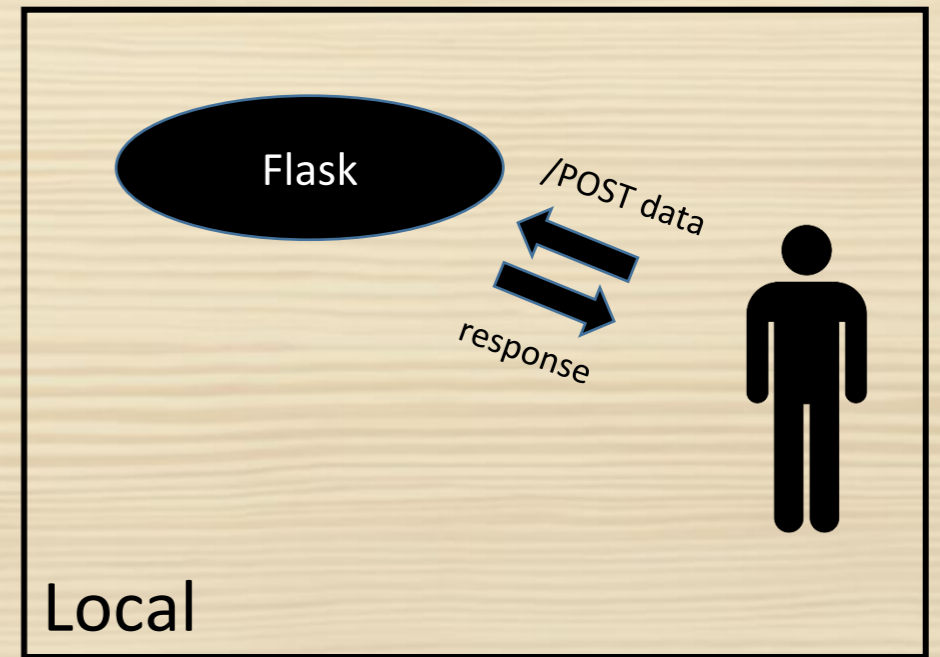
# Criar uma API em Flask

## Motivação:

- Desenvolvimento independente
- Permite integração entre serviços

## Webhook API:

1. Usuário faz requisição por meio de HTTP Post, enviando ao servidor data.json
2. Flask faz o tratamento de data.json, e retorna response.json



# Criar uma API em Flask

## Modificar o arquivo index.py

```
from flask import Flask, request

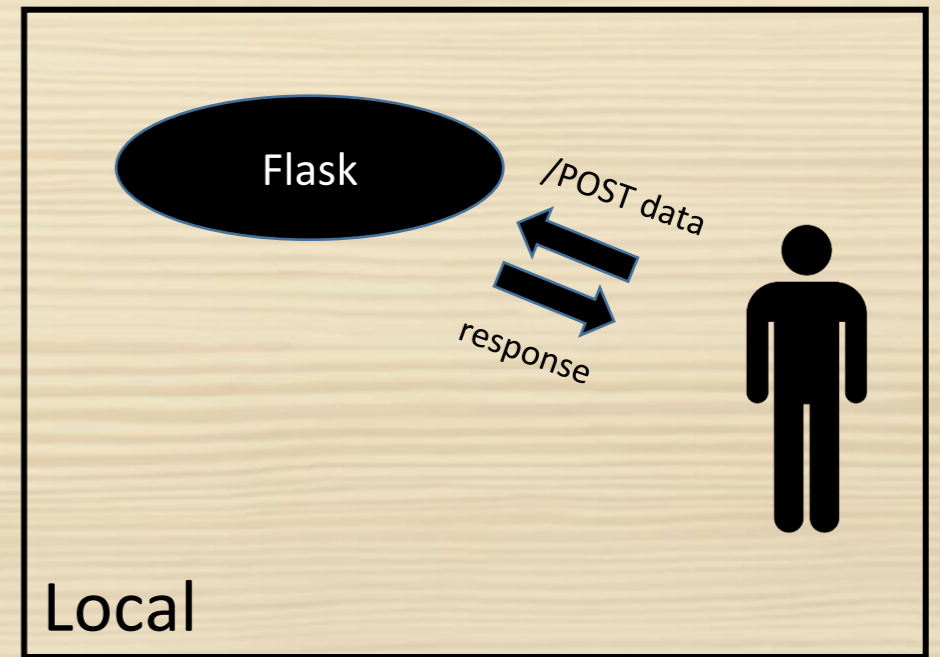
app = Flask(__name__)

@app.route('/', methods=['POST'])
def webhook():
    data = request.get_json(force=True)
    return 'Recebido: {}\n'.format(data['data'])

if __name__ == "__main__":
    app.run()
```

Métodos HTTP permitidos

Transforma dados recebidos pelo POST em dicionário



```
C:\Users\tiago>cd Documents
```

```
C:\Users\tiago\Documents>set FLASK_APP=index.py
```

```
C:\Users\tiago\Documents>flask run
```

```
* Serving Flask app "index.py"
```

```
* Environment: production
```

```
WARNING: This is a development server. Do not use it in a production deployment.
```

```
Use a production WSGI server instead.
```

```
* Debug mode: off
```

```
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

```
127.0.0.1 - - [01/Apr/2020 17:13:32] "[37mGET / HTTP/1.1[0m" 200 -
```

Pressione Ctrl+C para  
abortar o servidor Flask



C:\Users\tiago&gt;cd Documents

C:\Users\tiago\Documents&gt;set FLASK\_APP=index.py

C:\Users\tiago\Documents&gt;flask run

\* Serving Flask app "index.py"

\* Environment: production

WARNING: This is a development server. Do not use it in a production deployment.  
Use a production WSGI server instead.

\* Debug mode: off

\* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

127.0.0.1 - - [01/Apr/2020 17:13:32] "[37mGET / HTTP/1.1[0m" 200 -

C:\Users\tiago\Documents&gt;flask run

\* Serving Flask app "index.py"

\* Environment: production

WARNING: This is a development server. Do not use it in a production deployment.  
Use a production WSGI server instead.

\* Debug mode: off

\* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)



flask run

# Enviar requisição à API

Abrir um segundo terminal, e digitar o comando:

```
curl -H \Content-Type:application/json\ -X POST --data "{\"data\":\"123\"}" http://localhost:5000/
```

Obs.: No Linux, utilizar comando:

```
curl -H \Content-Type: application/json\ -X POST --d "{\"data\":\"123\"}" http://localhost:5000/
```

```
Prompt de Comando - flask run
Microsoft Windows [versão 10.0.18363.720]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\tiago>cd Documents
C:\Users\tiago\Documents>set FLASK_APP=index.py
C:\Users\tiago\Documents>flask run
* Serving Flask app "index.py"
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
123
127.0.0.1 - - [01/Apr/2020 19:59:55] "[37mPOST / HTTP/1.1[0m" 200 -
```

```
Selecionar Prompt de Comando
Microsoft Windows [versão 10.0.18363.720]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\tiago>curl -H \Content-Type:application/json\ -X POST --data '{"data": "123\"}" http://localhost:5000/
Recebido: 123

C:\Users\tiago>
```

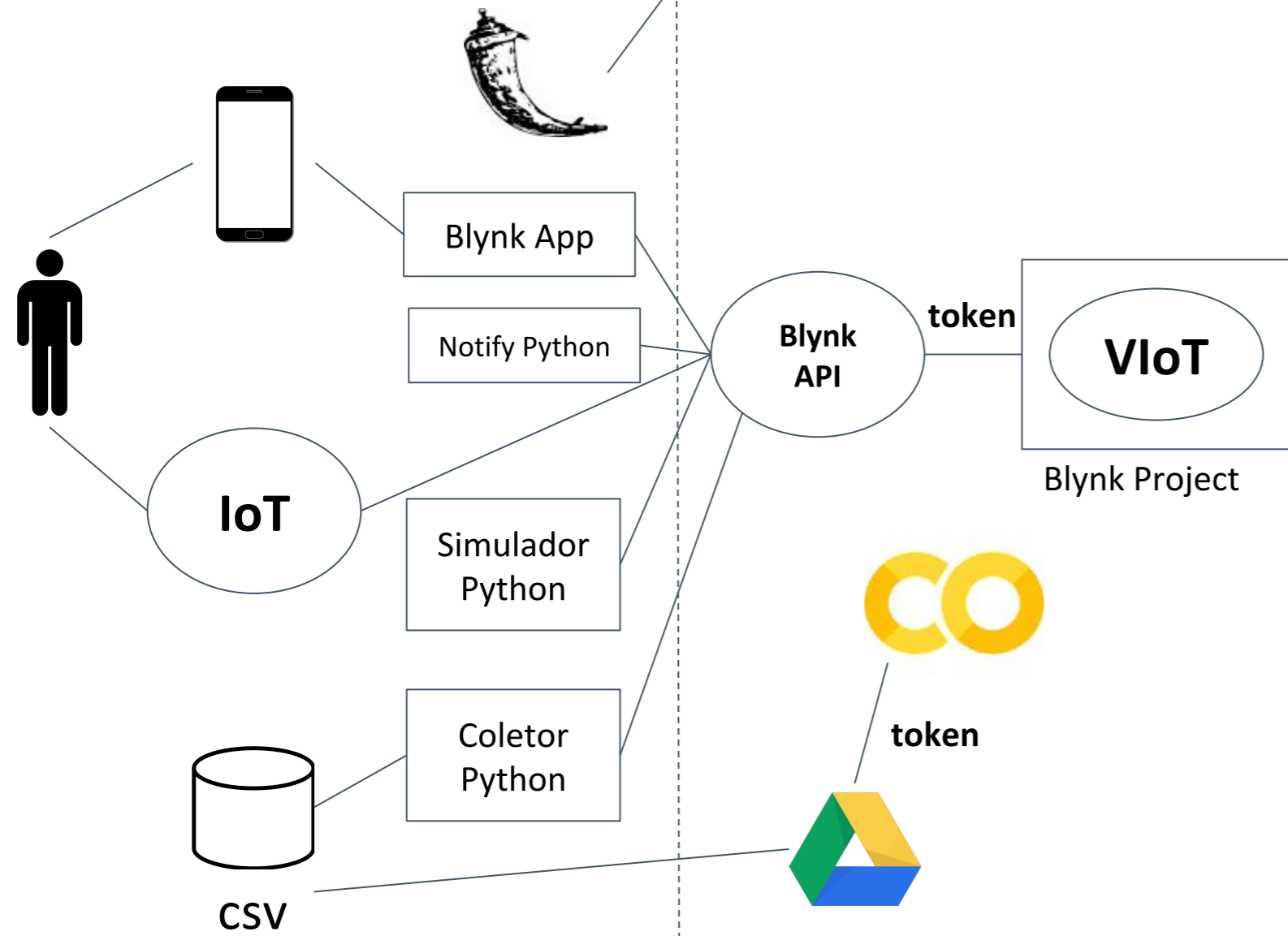


Enviar comando curl

Resposta do Flask: "Recebido: 123"

# Arquitetura

LOCAL CLOUD  
**ngrok**



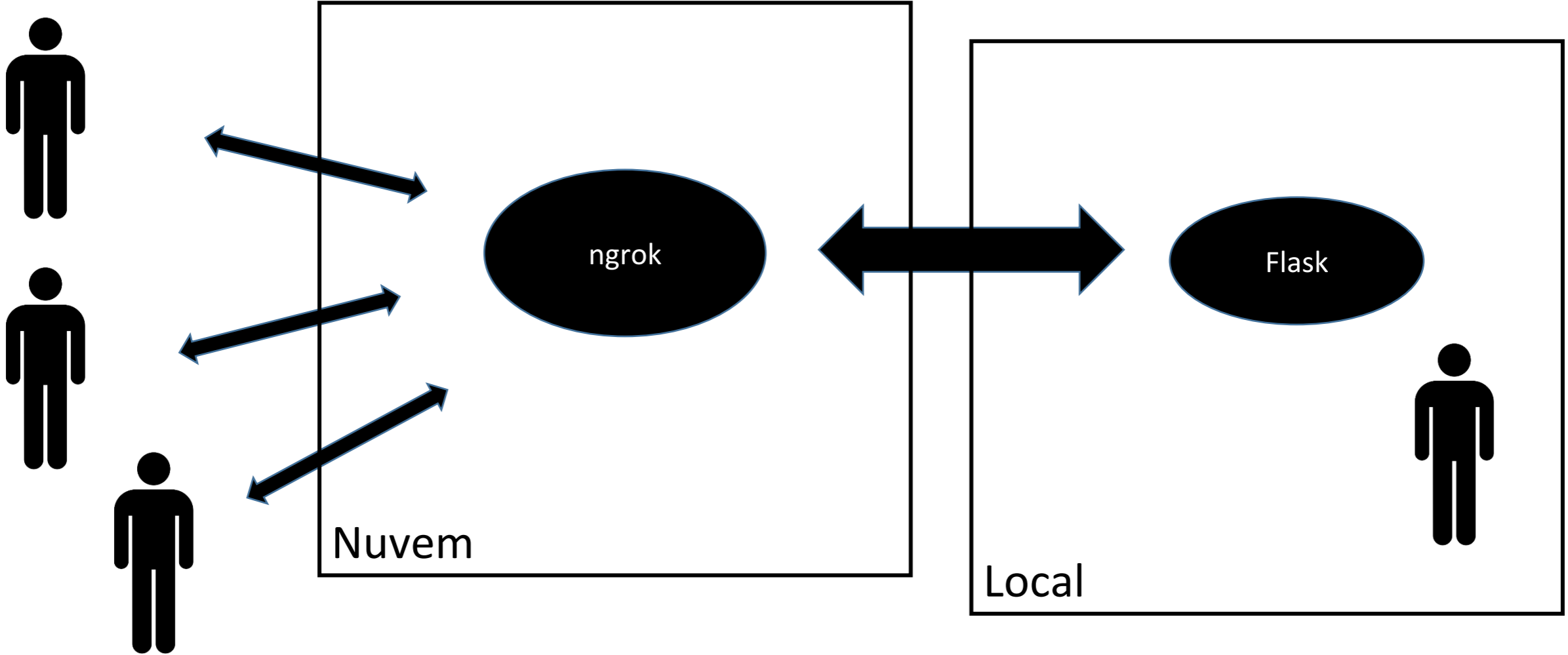
# Tutoriais

1. Blynk: Cloud IoT com interface Mobile
  - a. Mobile App Android/iOS + HTTP request
  - b. Simular comportamento IoT com Python
  - c. Usar API para notificação no celular
2. Coleta de Dados
  - a. Coletor Python
  - b. Ambiente Google Colab e Google Drive
3. Servidor Local e Tunelamento
  - a. Servidor local no Flask
  - b. Tunelamento no ngrok

# Fazer tunelamento na nuvem

O servidor local criado no Flask permite comunicação apenas entre computadores na mesma rede local

Para fazer comunicação entre os membros do grupo ou outros serviços web, pode-se utilizar uma ferramenta de tunelamento como o ngrok: <https://ngrok.com/>



Crie uma conta no ngrok



# Public URLs for testing your chatbot.

Spend more time programming. One command for an instant, secure URL to your localhost server through any NAT or firewall.

Get started for free →

```
$ ./ngrok http 3000
ngrok by @inconshreveable

Session Status  online
Account         Kate Libby (Plan: Pro)
Web Interface    http://127.0.0.1:4040
Forwarding      http://katesapp.ngrok.io -> localhost:3000
Forwarding      https://katesapp.ngrok.io -> localhost:3000
```

- slack GitHub SendGrid twilio ATlassian

As well as Amazon Web Services and many more.

Ask a question

Siga as instruções para baixar e autenticar o ngrok

No Windows, substituir ./ngrok pelo comando ngrok.exe

Explore ngrok

- Status
- Reserved
- Auth
- Team
- Admin
- Billing

Want more from ngrok?  
[Upgrade now](#)

## Setup & Installation

### 1 Download ngrok

ngrok is easy to install. Download a single binary with zero run-time dependencies.

[Mac OS X](#)

[Download for Windows](#)

[Linux](#) [Mac \(32-Bit\)](#) [Windows \(32-Bit\)](#) [Linux \(ARM\)](#)  
[Linux \(32-Bit\)](#) [FreeBSD \(64-Bit\)](#) [FreeBSD \(32-Bit\)](#)

### 2 Unzip to install

On Linux or OSX you can unzip ngrok from a terminal with the following command. On Windows, just double click ngrok.zip.

```
$ unzip /path/to/ngrok.zip
```

Most people keep ngrok in their user folder or set an alias for easy access.

### 3 Connect your account

Running this command will add your account's authtoken to your ngrok.yml file. This will give you more features and all open tunnels will be listed here in the dashboard.

```
$ ./ngrok authtoken [redacted]
```

### 4 Fire it up

Read [the documentation](#) on how to use ngrok. Try it out by running it from the command line:

```
$ ./ngrok help
```

To start a HTTP tunnel on port 80, run this next:

```
$ ./ngrok http 80
```



```
C:\Users\tiago>cd Downloads
```

```
C:\Users\tiago\Downloads>
```

Navegar para a pasta em que o ngrok.exe foi extraído

```
C:\Users\tiago>cd Downloads
```

```
C:\Users\tiago\Downloads>ngrok.exe authtoken [REDACTED]
```



ngrok.exe authtoken [código de autenticação]



```
C:\Users\tiago>cd Downloads
```

```
C:\Users\tiago\Downloads>ngrok.exe authtoken [REDACTED]  
Authtoken saved to configuration file: C:\Users\tiago/.ngrok2/ngrok.yml
```

```
C:\Users\tiago\Downloads>ngrok.exe http 5000_
```



ngrok.exe http 5000



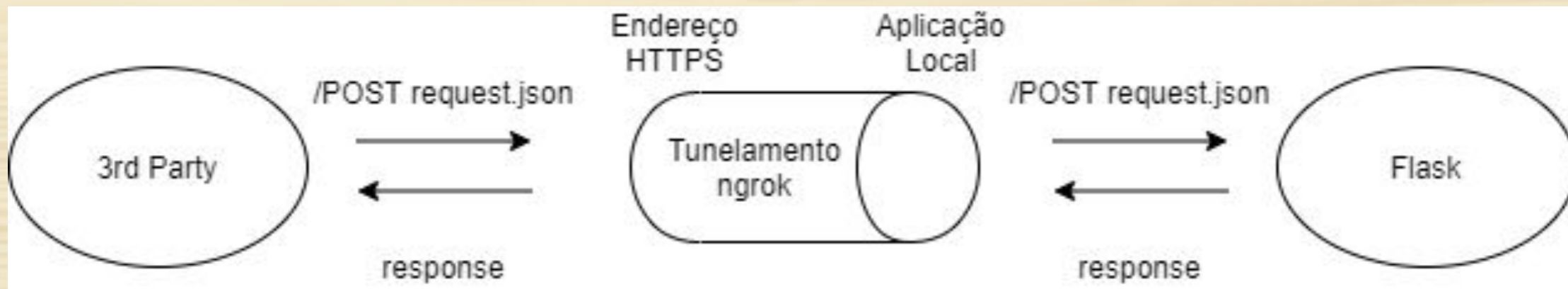
```
Session Status      online
Account             (Plan: Free)
Version             2.3.35
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://8018c13b.ngrok.io -> http://localhost:5000
                    https://8018c13b.ngrok.io -> http://localhost:5000
Connections
  ttl   opn   rt1   rt5   p50   p90
   0    0     0    0.00 0.00  0.00
```

ngrok inicia tunelamento da porta local 5000 para um endereço web temporário

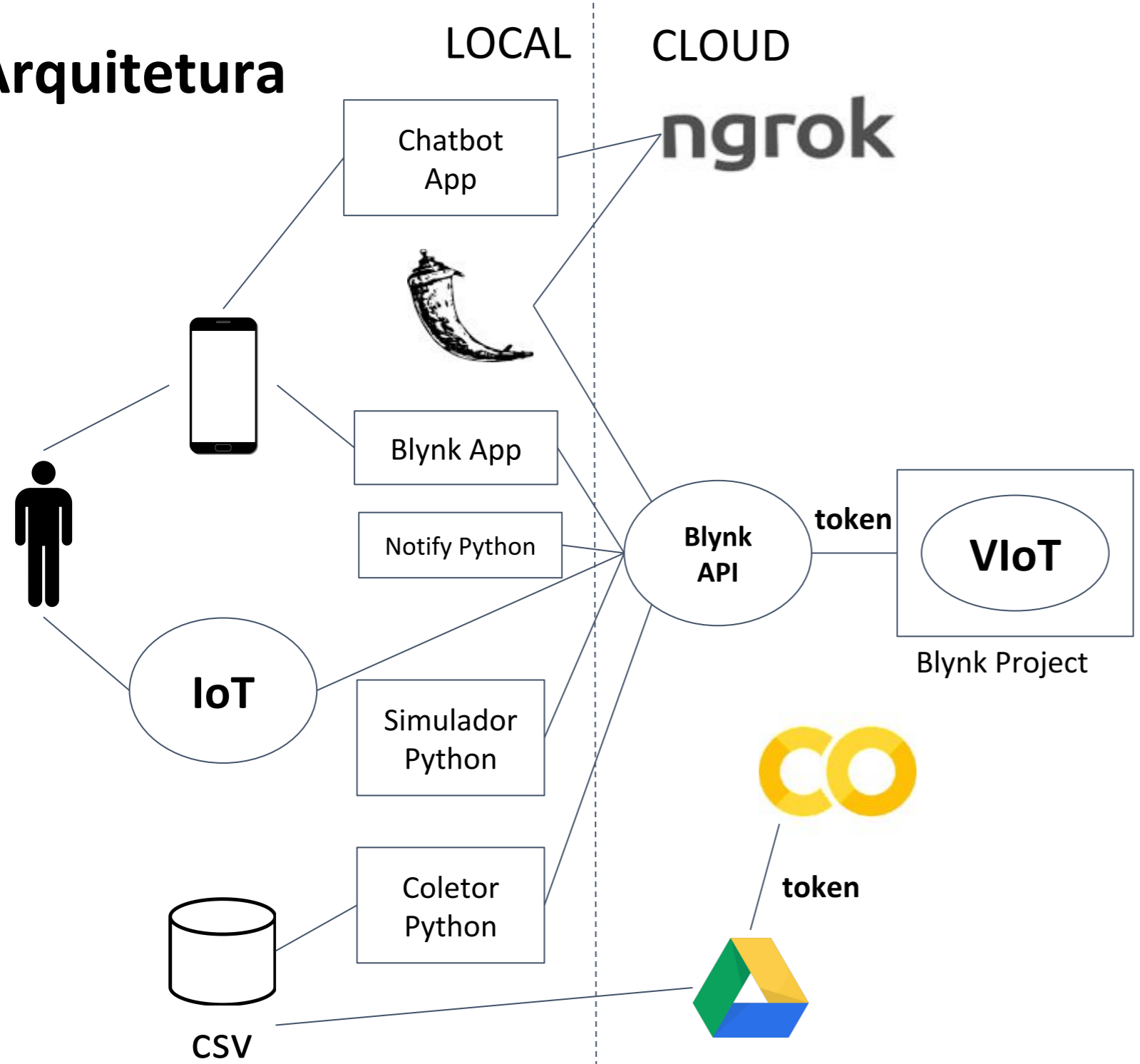
# Comunicação com o servidor Flask

Outros membros do grupo agora podem enviar POST requests para o endereço https:

```
curl -H \Content-Type: application/json\ -X POST -d "{\"data\":\"123\"}" https://8018c13b.ngrok.io
```



# Arquitetura



# Tutoriais

1. Blynk: Cloud IoT com interface Mobile
  - a. Mobile App Android/iOS + HTTP request
  - b. Simular comportamento IoT com Python
  - c. Usar API para notificação no celular
2. Coleta de Dados
  - a. Coletor Python
  - b. Ambiente Google Colab e Google Drive
3. Servidor Local e Tunelamento
  - a. Servidor local no Flask
  - b. Tunelamento no ngrok

# DialogFlow

- Plataforma de processamento de linguagem natural do Google
- Permite criação de chatbots
- Integração com Facebook Messenger, Telegram, LINE, Google Assistant



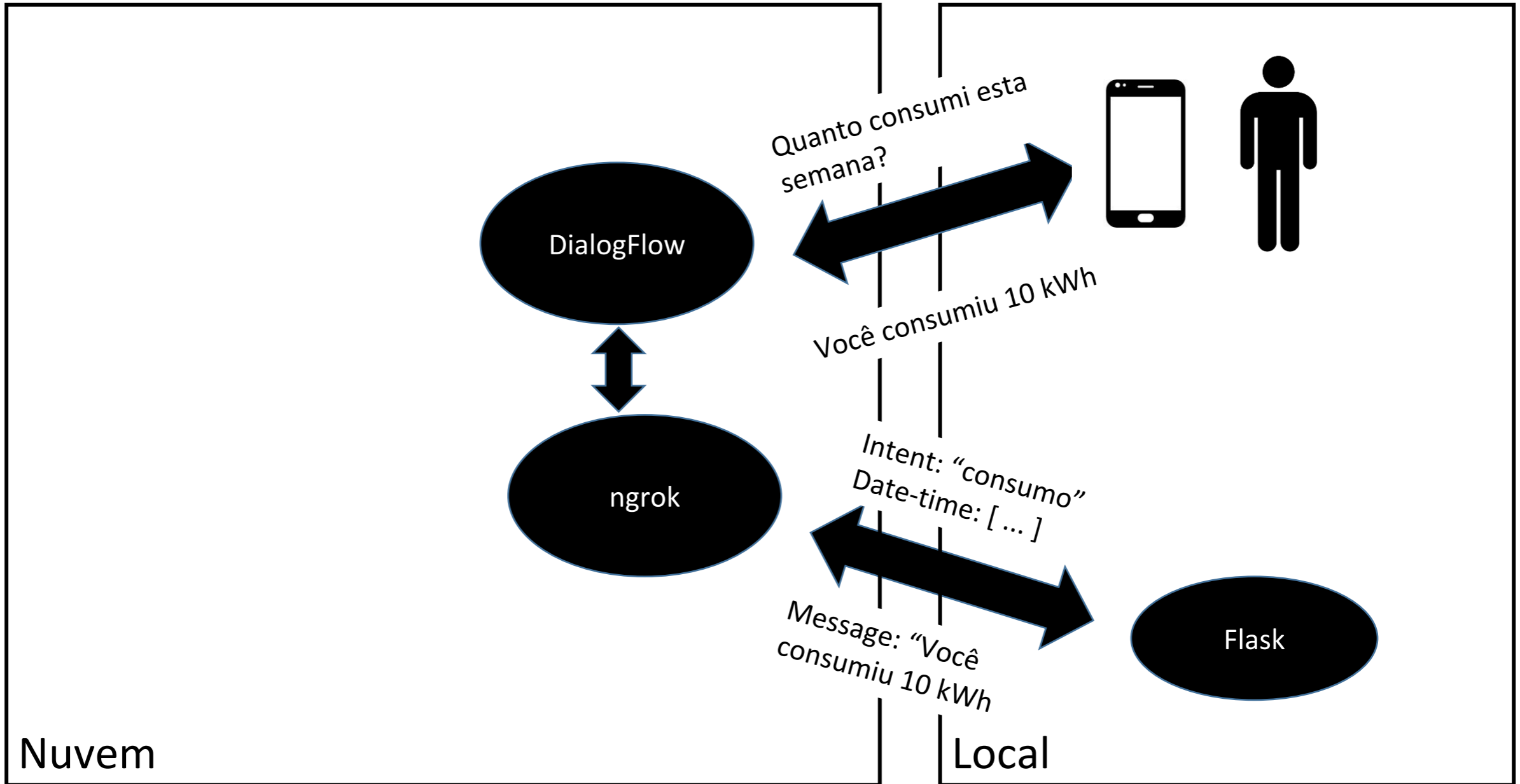
# Integração do DialogFlow com Flask+ngrok

**Vantagem:** gerar respostas customizadas de acordo com a requisição do usuário (busca em banco de dados, ativação de controlador)

**Exemplo:** Chatbot para monitoramento do consumo energético residencial

- 1.Usuário pergunta para chatbot (DialogFlow) "Quanto consumi esta semana?"
- 2.DialogFlow processa intenção do usuário e faz requisição POST para endereço do ngrok
- 3.ngrok faz tunelamento para o servidor Flask
- 4.Flask consulta base de dados pelo consumo desta semana
- 5.Flask retorna para o ngrok a mensagem: "Você consumiu 10 kWh esta semana"
- 6.ngrok retorna para o DialogFlow: "Você consumiu 10 kWh esta semana"





# Criando um Chatbot do Google

1. Baixar os arquivos index.py e response.json disponibilizados e colocar pasta Documents
  - a) Link para os arquivos:  
[https://drive.google.com/drive/folders/1nR0NSiuWN8BSCGnwKjetf\\_bluunfXqfa?usp=sharing](https://drive.google.com/drive/folders/1nR0NSiuWN8BSCGnwKjetf_bluunfXqfa?usp=sharing)
2. Reiniciar o flask run
3. Ir no site do DialogFlow: <https://dialogflow.com/>
4. Criar uma conta e ir no console: <https://dialogflow.cloud.google.com/>



Agent name

CREATE

Try it now



BoaEnergia



pt-BR

DEFAULT LANGUAGE ?

DEFAULT TIME ZONE

Clique com o botão esquerdo do mouse aqui

pt-BR (Default) - Portuguese (Brazil) is the default language for this agent. Other languages can be

(GMT-3:00) America/Buenos\_Aires  
Date and time requests are resolved using this timezone.

Please use test console above to try a sentence.

See how it works in [Google Assistant](#).

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

GOOGLE PROJECT

Create a new Google project

Enables Cloud functions, Actions on Google and permissions management.

AGENT TYPE

Set as Mega Agent

Combine multiple Dialogflow agents (i.e. sub agents) into a single agent (i.e. mega agent).



BoaEnergia

Create new agent

View all agents

Knowledge [beta]

Fulfillment

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

Agent name

CREATE

Try it now

DEFAULT LANGUAGE

English - en  
Primary language for your agent. Other languages can be

DEFAULT TIME ZONE

(GMT-3:00) America/Buenos\_Aires  
Date and time requests are resolved using this timezone.

Create a new Google project  
Enables Cloud functions, Actions on Google and permissions management.

AGENT TYPE

Set as Mega Agent  
Combine multiple Dialogflow agents (i.e. sub agents) into a single agent (i.e. mega agent).

Clique com o botão esquerdo do mouse aqui

Please use test console above to try a sentence.

See how it works in Google Assistant.



# TesteBot

CREATE

Clique com o botão esquerdo do mouse aqui

- BoaEnergia
- pt-BR
- Intents
- Entities
- Knowledge <sup>[beta]</sup>
- Fulfillment
- Integrations
- Training
- Validation
- History
- Analytics
- Prebuilt Agents

## DEFAULT LANGUAGE ?

Portuguese (Brazilian) — pt-br  
Primary language for your agent. Other languages can be added later.

(GMT-3:00) America/Buenos\_Aires  
Date and time requests are resolved using this timezone.

## GOOGLE PROJECT

Create a new Google project  
Enables Cloud functions, Actions on Google and permissions management.

## AGENT TYPE

Set as Mega Agent  
Combine multiple Dialogflow agents (i.e. sub agents) into a single agent (i.e. mega agent).

Try it now

Please use test console above to try a sentence.

See how it works in [Google Assistant](#).



TesteBot

pt-BR

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

## Intents

CREATE INTENT

Search intents

Clique com o botão esquerdo do mouse aqui

- Default Fallback Intent
- Default Welcome Intent



No regular intents yet. [Create the first one.](#)

Intents are mappings between a user's queries and actions fulfilled by your software. [Read more here.](#)

Before you start, check out [Prebuilt Agents](#), a collection of agents developed by the Dialogflow team.

Try it now

Please use test console above to try a sentence.

See how it works in [Google Assistant](#).

Agent training completed OK



TesteBot pt-BR

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

Intent name

SAVE

Contexts

Events

Training phrases



Train the intent with what your users will say

Provide examples of the intent in natural language with different variations and synonyms to improve the accuracy of intent matching.

Clique com o botão esquerdo do mouse aqui

ADD TRAINING PHRASES

Action and parameters

Extract the action and parameters

Parameters are specific values extracted from a user's request when entities are matched. The values captured by parameters can be used in fulfillment, or in building a

Try it now

Please use test console above to try a sentence.

See how it works in Google Assistant.



Intent name

SAVE

Try it now

TesteBot

pt-BR

Intents

Adicionar frases quanto consumi este mes? e quanto consumi hoje?

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

Training phrases

Search training phrase

Add user expression

Action and parameters

### Extract the action and parameters



Parameters are specific values extracted from a user's request when entities are matched. The values captured by parameters can be used in fulfillment, or in building a response. If you mark parameters as required, Dialogflow will prompt the user if their values were not extracted from their initial request. [Learn more](#)

ADD PARAMETERS AND ACTION

Please use test console above to try a sentence.

See how it works in Google Assistant.





Intent name

SAVE

Try it now

TesteBot

pt-BR

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

Validation

History

Analytics

Prebuilt Agents



### Extract the action and parameters

Parameters are specific values extracted from a user's request when entities are matched. The values captured by parameters can be used in fulfillment, or in building a response. If you mark parameters as required, Dialogflow will prompt the user if their values were not extracted from their initial request. [Learn more](#)

[MANAGE PARAMETERS AND ACTION](#)

Responses

Clique com o botão esquerdo do mouse aqui



### Execute and respond to the user

Respond to your users with a simple message, or build custom rich messages for the integrations you support. [Learn more](#)

[ADD RESPONSE](#)

Fulfillment

Please use test console above to try a sentence.

See how it works in [Google Assistant](#).



### Intent name

SAVE

TesteBot  
pt-BR

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

” quanto consumi hoje?  
” quanto consumi este mes?

### Action and parameters

Enter action name

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	date-time	@sys.date-time	Sdate-time	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

### Responses

Execute and respond to the user

Try it now

Please use test console above to try a sentence.

See how it works in Google Assistant.



TesteBot

pt-BR

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

consumo

SAVE

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	date-time	@sys.date-time	Sdate-time	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

Responses



### Execute and respond to the user

Respond to your users with a simple message, or build custom rich messages for the integrations you support.

[Learn more](#)

**ADD RESPONSE**

Clique com o botão esquerdo do mouse aqui

Fulfillment



Try it now

Agent

USER SAYS  
quanto consumi esta semana?

DEFAULT RESPONSE  
Not available

INTENT  
consumo

ACTION  
Not available

PARAMETER	VALUE
date-time	{ "startDate": "2020-03-29T00:00:00-03:00", "endDate": "2020-04-04T23:59:59-03:00" }

DIAGNOSTIC INFO



consumo

SAVE

Try it now

- TesteBot
- pt-BR
- Intents**
- Entities
- Knowledge [beta]
- Fulfillment
- Integrations
- Training
- Validation
- History
- Analytics
- Prebuilt Agents

Enter name     Enter entity     Enter value   

+ New parameter

Responses



Execute and respond to the user

Respond to your users with a simple message, or build custom rich messages for the integrations you support.

[Learn more](#)

**ADD RESPONSE**

Fulfillment

- Enable webhook call for this intent
- Enable webhook call for slot filling

Agent

USER SAYS  
quanto consumi esta semana?

DEFAULT RESPONSE  
Not available

INTENT  
consumo

ACTION  
Not available

PARAMETER	VALUE
date-time	{ "startDate": "2020-03-29T00:00:00-03:00", "endDate": "2020-04-04T23:59:59-03:00" }

DIAGNOSTIC INFO



TesteBot  
pt-BR

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

### consumo

SAVE

<input type="checkbox"/>	date-time	@sys.date-time	Sdate-time	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

### Responses



### Execute and respond to the user

Respond to your users with a simple message, or build custom rich messages for the integrations you support.

[Learn more](#)

ADD RESPONSE

### Fulfillment

Enable webhook call for this intent

Enable webhook call for slot filling

Clique com o botão esquerdo do mouse aqui

Try it now

Agent

USER SAYS  
quanto consumi esta semana?

DEFAULT RESPONSE

Not available

INTENT

consumo

ACTION

Not available

PARAMETER

VALUE

date-time

```
{ "startDate": "2020-03-29T00:00:00-03:00", "endDate": "2020-04-04T23:59:59-03:00" }
```

DIAGNOSTIC INFO



TesteBot  
pt-BR

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

consumo

SAVE

<input type="checkbox"/>	date-time	@sys.date-time	Sdate-time
<input type="checkbox"/>	Enter name	Enter entity	

Clique com o botão esquerdo do mouse aqui

+ New parameter

Responses



### Execute and respond to the user

Respond to your users with a simple message, or build custom rich messages for the integrations you support. [Learn more](#)

ADD RESPONSE

Fulfillment

Enable webhook call for this intent

Enable webhook call for slot filling

Try it now

Agent

USER SAYS  
quanto consumi esta semana?

DEFAULT RESPONSE  
Not available

INTENT  
consumo

ACTION  
Not available

PARAMETER	VALUE
date-time	{ "startDate": "2020-03-29T00:00:00-03:00", "endDate": "2020-04-04T23:59:59-03:00" }

DIAGNOSTIC INFO



# consumo

WORKING...

Try it now

- TesteBot
- pt-BR
- Intents**
- Entities
- Knowledge [beta]
- Fulfillment
- Integrations
- Training
- Validation
- History
- Analytics
- Prebuilt Agents

REQUIRED	PARAMETER NAME	ENTITY	VALUE	
<input type="checkbox"/>	date-time	@sys.date-time	\$date-time	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

## Responses



### Execute and respond to the user

Respond to your users with a simple message, or build custom rich messages for the integrations you support.

[Learn more](#)

**ADD RESPONSE**

## Fulfillment

- Enable webhook call for this intent
- Enable webhook call for slot filling

## Agent

USER SAYS  
quanto consumi esta semana?

DEFAULT RESPONSE  
Not available

INTENT  
consumo

ACTION  
Not available

PARAMETER	VALUE
date-time	{ "startDate": "2020-03-29T00:00:00-03:00", "endDate": "2020-04-04T23:59:59-03:00" }

DIAGNOSTIC INFO



# Intents

CREATE INTENT

- TesteBot
- pt-BR
- Intents**
- Entities
- Knowledge [beta]
- Fulfillment
- Integrations
- Training
- Validation
- History
- Analytics
- Prebuilt Agents

Clique com o botão esquerdo do mouse aqui

Search intents
consumo
back Intent
ome Intent

Try it now

Please use test console above to try a sentence.

See how it works in [Google Assistant](#).





# Fulfillment

- TesteBot
- pt-BR
- Intents
- Entities
- Knowledge <sup>[beta]</sup>
- Fulfillment**
- Integrations
- Training
- Validation
- History
- Analytics
- Prebuilt Agents

## Webhook

DISABLED

Your web service will receive a POST request from Dialogflow in the form of the response to a query matched by intents with webhook enabled. Be sure that your web service meets the [requirements](#) specific to the API version enabled in this agent.

Clique com o botão esquerdo do mouse aqui

## Inline Editor

(Powered by Cloud Functions for Firebase)

DISABLED

Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. [Docs](#)

```

index.js package.json
1 // See https://github.com/dialogflow/dialogflow-fulfillment-nodejs
2 // for Dialogflow fulfillment library docs, samples, and to report issues
3 'use strict';
4
5 const functions = require('firebase-functions');
6 const {WebhookClient} = require('dialogflow-fulfillment');
7 const {Card, Suggestion} = require('dialogflow-fulfillment');
8
9 process.env.DEBUG = 'dialogflow:debug'; // enables lib debugging statements
10
11 exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request, re
12   const agent = new WebhookClient({ request, response });
13   console.log('Dialogflow Request headers: ' + JSON.stringify(request.headers)
14   console.log('Dialogflow Request body: ' + JSON.stringify(request.body));

```

Try it now



Please use test console above to try a sentence.

See how it works in [Google Assistant](#).



## Fulfillment

Try it now

- TesteBot
- pt-BR
- Intents
- Entities
- Knowledge [beta]
- Fulfillment**
- Integrations
- Training
- Validation
- History
- Analytics
- Prebuilt Agents

### Webhook

ENABLED

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

URL\*

Usar endereço do ngrok

Basic Auth

Header

Header

[+ Add header](#)

SMALL TALK  Disable webhook for Smalltalk

Inline Editor (Powered by Cloud Functions for Firebase)  DISABLED

Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. [Docs](#)

**i** Please use test console above to try a sentence.

See how it works in [Google Assistant](#).



# Fulfillment

Try it now



TesteBot

pt-BR

Intents

Entities

Knowledge <sup>[beta]</sup>

**Fulfillment**

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

## Inline Editor (Powered by Cloud Functions for Firebase)

DISABLED

Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. [Docs](#)

```

index.js package.json
25 // // Uncomment and edit to make your own intent handler
26 // // uncomment `intentMap.set('your intent name here', yourFunctionHandler)
27 // // below to get this function to be run when a Dialogflow intent is match
28 // function yourFunctionHandler(agent) {
29 //   agent.add(`This message is from Dialogflow's Cloud Functions for Fireba
30 //   agent.add(new Card({
31 //     title: `Title: this is a card title`,
32 //     imageUrl: `https://developers.google.com/assistant/sdk/develop-for-a
33 //     text: `This is the body text of a card`,
34 //     buttonText: `This is a button`,
35 //     buttonUrl: `https://assistant.google.com/terms
36 //   })
37 // });
38 // agent.add(new Suggestion(`Quick Reply`));
39

```



DONE

Please use test console above to try a sentence.





# Fulfillment

TesteBot

pt-BR

Intents

Entities

Knowledge <sup>[beta]</sup>

**Fulfillment**

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

## Webhook

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

URL\*

BASIC AUTH

HEADERS

[+ Add header](#)

SMALL TALK

**Inline Editor** (Powered by Cloud Functions for Firebase) DISABLED

Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. [Docs](#)

Perguntar "Quanto consumi esta semana?"

quanto

quanto consumi esta semana?

quanto consumi hoje?

try a

---

See how it works in [Google Assistant](#).



# Fulfillment

- TesteBot
- pt-BR
- Intents
- Entities
- Knowledge [beta]
- Fulfillment**
- Integrations
- Training
- Validation
- History
- Analytics
- Prebuilt Agents

## Webhook

ENABLED

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

URL\*

BASIC AUTH

HEADERS

[+ Add header](#)

SMALL TALK  Disable webhook for Smalltalk

Inline Editor (Powered by Cloud Functions for Firebase)  DISABLED

Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. [Docs](#)

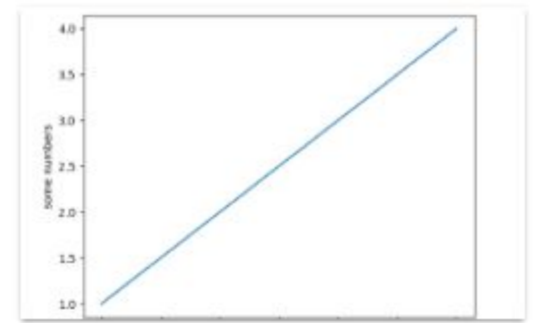
Try it now

See how it works in [Google Assistant](#).

### Agent

USER SAYS  [COPY CURL](#)

DEFAULT RESPONSE



Resposta recebida do Flask

# Próximos Passos

- Respostas customizadas, consultando dados de sensores (Pandas) ou ativando sensores (Blynk)
- Integração com outras plataformas de mensageria (Google Assistant, Telegram, etc)

# Referências

[1] Tutorial de instalação do Flask: <https://flask.palletsprojects.com/en/1.1.x/quickstart/>

[2] Webhook flask: <https://ogma-dev.github.io/posts/simple-flask-webhook/>

[3] Site do ngrok (tunelamento): <https://ngrok.com/>

[4] Guia para webhooks do DialogFlow: <https://cloud.google.com/dialogflow/docs/fulfillment-webhook>

[5] Código-fonte do chatbot para monitoramento do consumo residencial:  
<https://github.com/tiagoyukio12/webhook>

# Dúvidas?

[tiagoyukio12@gmail.com](mailto:tiagoyukio12@gmail.com)