

wise**ful**

Victor Hayashi

+55 11 94870-6144
victortakashih@gmail.com



SCAN ME

amazon.com.br Assine Prime

Enviar para Victor Santo André 09110260

Ofertas do Dia Mais Vendidos Victor, sua Amazon.com.br

Skills da Alexa Casa inteligente Jogos e curiosidades Estilo de vida Suas skills Primeiros

Alexa Skills › Educação e Referência

Boa Energia

por Victor Hayashi

★★★★★ 0

Grátis para ativar

"Alexa falar com boa energia"

Descrição

Aprenda dicas para o uso consciente de energia: dicas para geladeira e freezer, ...

A skill Boa Energia quer promover o uso consciente de energia elétrica, evitando ...

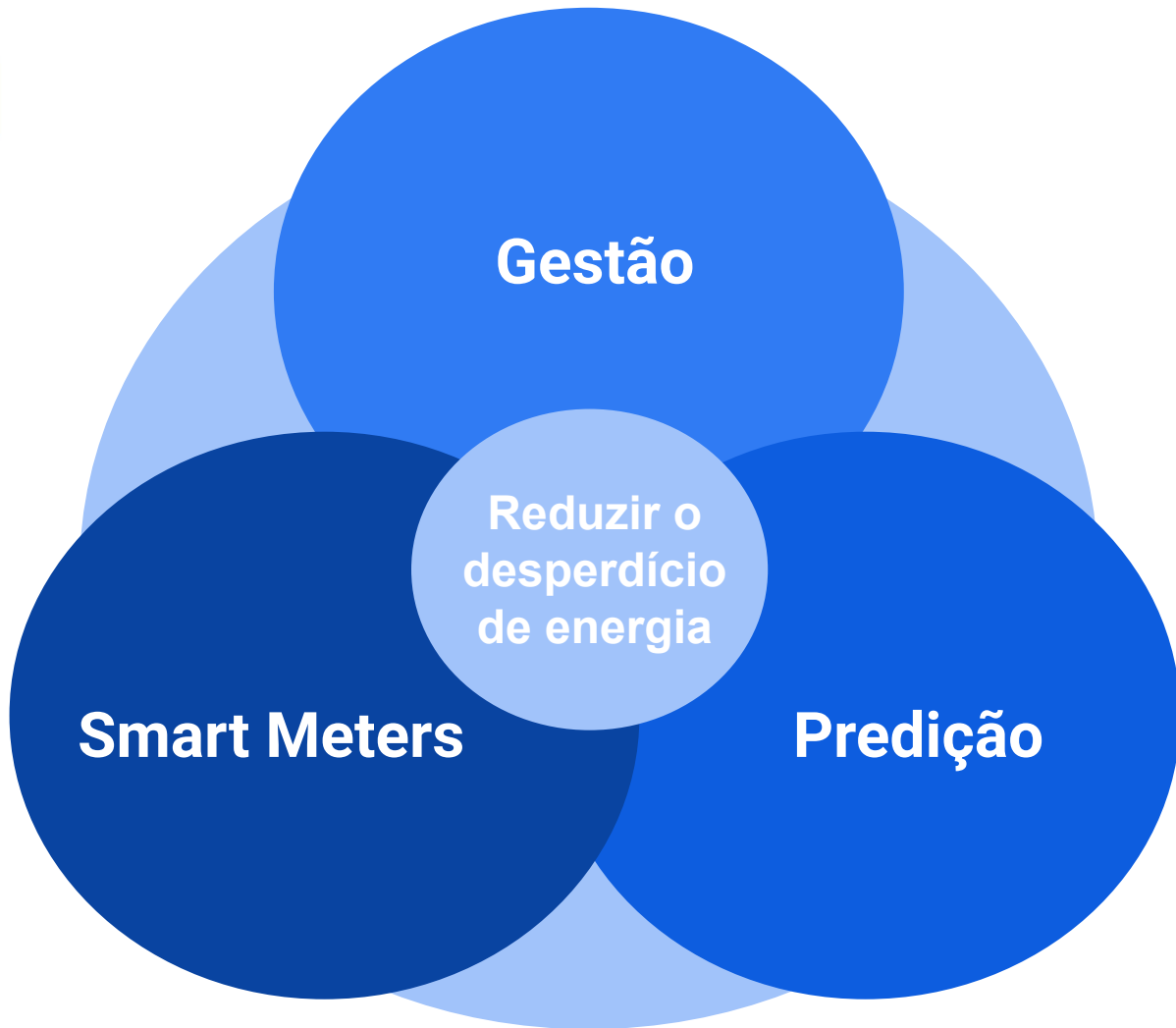
Para iniciar, apenas fale "Alexa, falar com boa energia" ou "Alexa, pergunte ao B ...

A qualquer momento, você pode dizer "Alexa, pare"

wise**ful**

IoT

IA



Domínio do Problema

No Estado de São Paulo, temos mais de 1 milhão de Micro e Pequenas Empresas no setor de comércio e mais de 1,1 milhão no setor industrial, segundo o Sebrae ([Sebrae, 2014](#)). Pequenos negócios energointensivos ([Sebrae, 2017](#)), como panificação, hotelaria, bares e restaurantes podem ser nossos clientes, e considerando um modelo de subscrição com R\$40,00 por mês (de forma a obter payback no módulo em 3 meses), e justificado pela economia gerada ser de pelo menos R\$100,00 por mês ([estudo da EU de 2015](#) mostra potencial de redução de consumo de até 25%, e estamos supondo uma conta mensal de R\$400,00 por negócio) e que tomemos 1% do mercado, temos um potencial anual de receita de ~10 milhões. No Brasil, temos 17,5 milhões * 16.6% = 2 milhões de médios negócios ([Sebrae, 2017](#)).



USP gasta cerca de
35M BRL
 em energia por ano

Seções **CORREIO BRAZILIENSE** Economia
 GoodStorage

Governo federal teve gasto de R\$ 2,4 bilhões com energia em 2017

A despesa da administração federal é vista como um valor excessivo na avaliação de especialistas

Hamilton Ferrari
 postado em 07/05/2018 06:00 / atualizado em 07/05/2018 10:08

Luz no desperdício*
 Despesa da administração pública federal com energia elétrica cai em relação a 2016, mas continua alta

Seguindo em frente
 Mesmo com queda de 7% em 2017 ante o ano anterior, o gasto foi de R\$ 2,4 bilhões. Segundo analistas, ainda é uma quantia considerável e precisa ser reduzida. Eles atribuem a despesa elevada à falta de consciência no serviço público.

Valor pago por órgão

Órgão	Valor (em milhões)
Camara dos Deputados	14,062
Senado Federal	14,081
TCU	6,939
STF	5,391
STJ	6,803
Justiça Federal	77,291

Schedule A Free Consultation
 Learn the Benefits of Upgrading Access Control Systems

wise**ful**

Oportunidade

15%

25%

savings through Action

Villain Detection
Shift Suggestion
Tariff Plan

~10%

savings through Information

Real Time Data
Information

Total savings

wise**ful** Solução

Hoje:

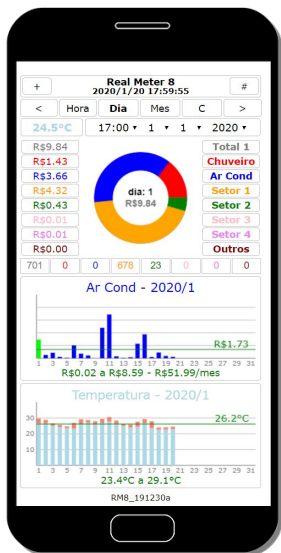
- Sem informação;
- Sem gestão;
- Sem melhoria.













wise**ful**

Smart Meter + App

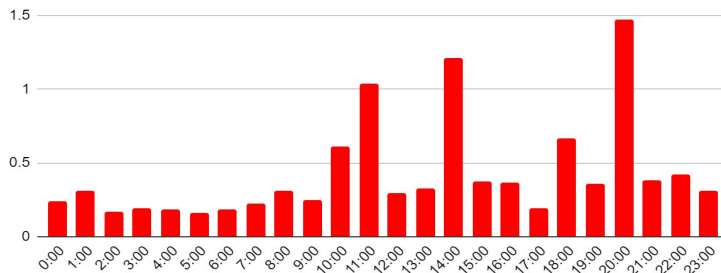
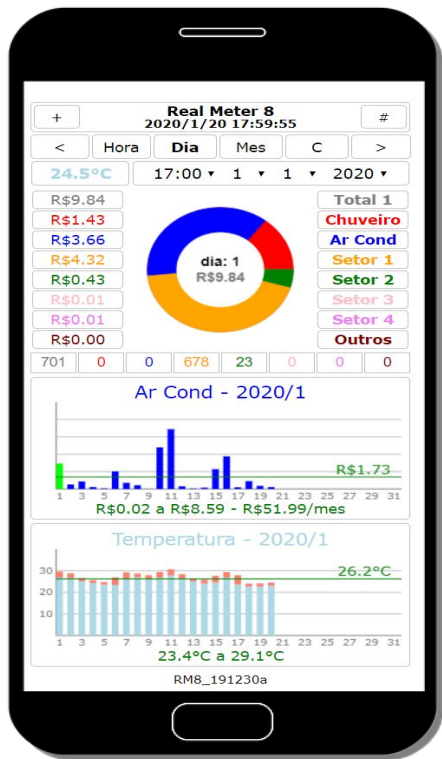


		TARIFF PLAN	SHIFT SUGGESTION	SMART METER DATA	PROPERTY EXTERIOR DATA	DOMESTIC & BUSINESS AVAILABILITY	CONSUMPTION VILLAIN DETECTION	SOCIAL NETWORK
EDP re:dy		○	○	○	○	○	○	○
Tibber		●	●	●	○	○	○	○
Sense Monitor		○	○	●	○	○	●	○
Eyedro		○	○	●	○	○	○	○
WEG		○	○	○	○	○	●	○
CUBi		○	○	●	○	○	○	○
Energy Portfolio Manager		○	○	○	○	●	○	○
Wiseful Platform		●	●	●	●	●	●	●

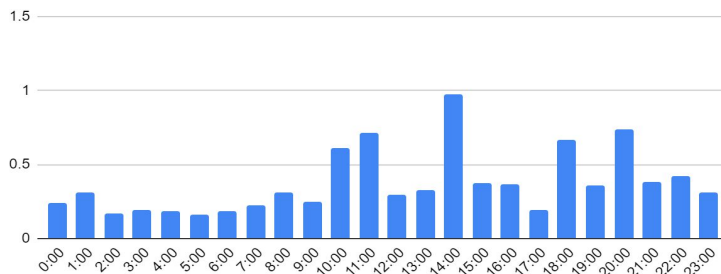
wise**ful** Vantagem Competitiva

Plataforma de dados de consumo atualizada em tempo real que permite:

- Economia de energia dos clientes através da gestão de energia;
- Oferecimento de soluções personalizadas por parceiros, como plano de tarifação, troca e manutenção de equipamentos, instalação de painéis solares (pré e pós venda)
- Redução do pico de demanda através de sugestões personalizadas em aplicação móvel e relatórios de previsão de demanda para as distribuidoras, visando reduzir o risco de operação.

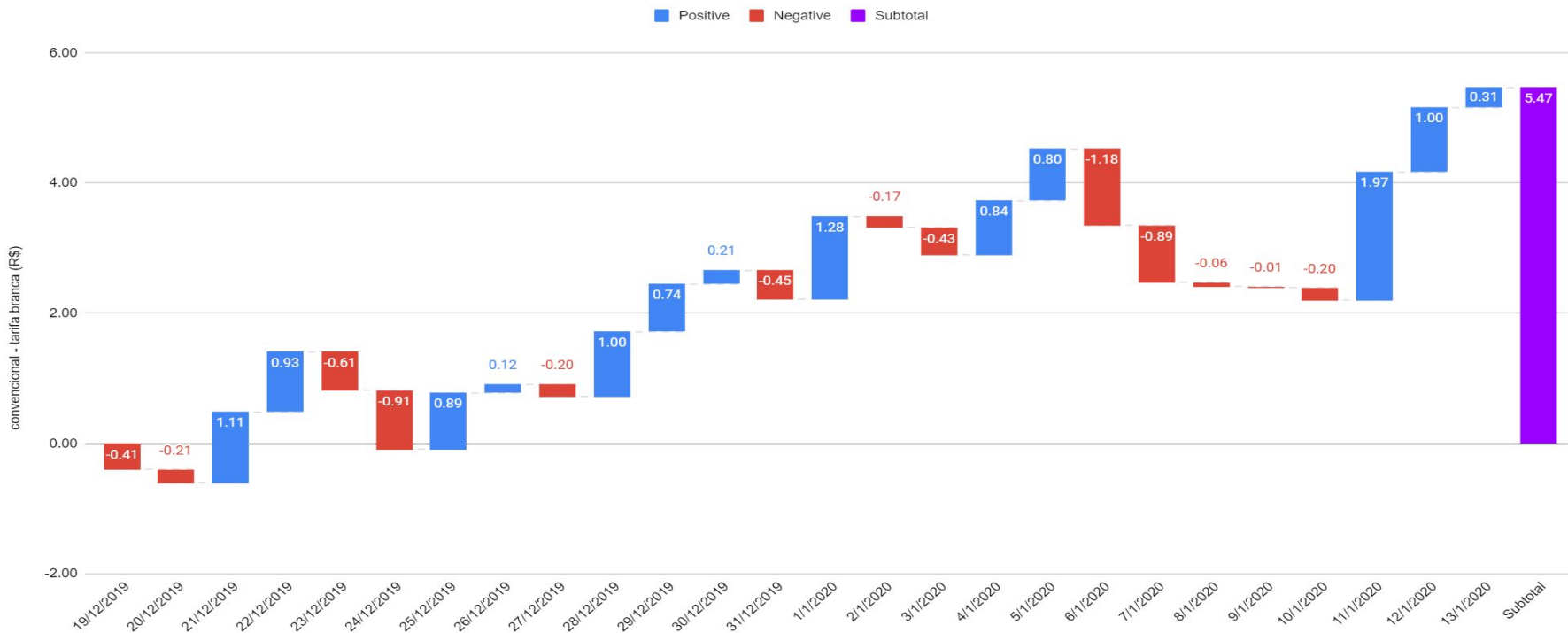


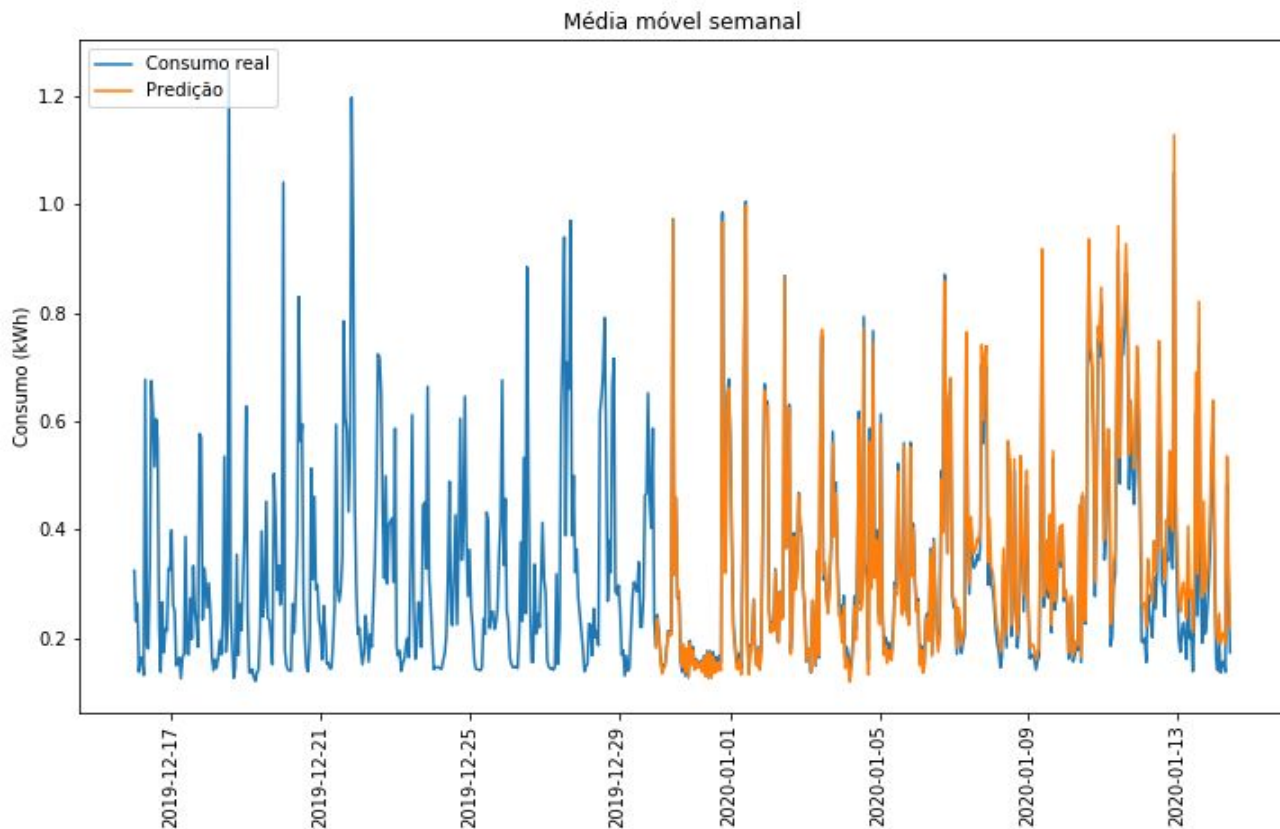
Default
Brazil



Reduction
20% energy bill reduction

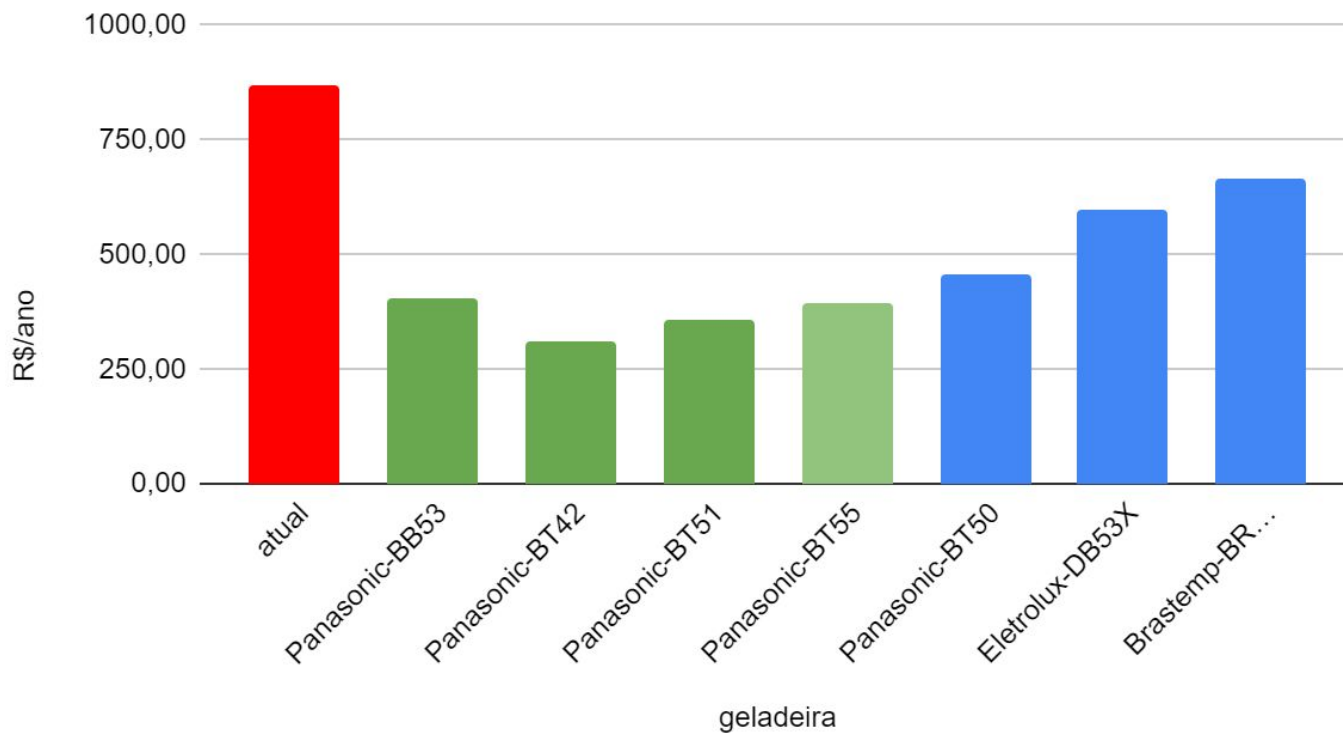
Simulação Tarifa Branca: o quanto a conta ficou mais barata?





wise**ful** Parceiros: Análise para Sugestão de Substituição

R\$/ano gastos com geladeira (dados de novembro de 2019)



wise**ful**

Desenvolvimento de padrão aberto Open Source

<https://github.com/vthayashi/OKIoT/wiki>



vthayashi / OKIoT

<> Code

! Issues 0

🔗 Pull requests 0

▶ Actions

📁 Projects 0

📖 Wiki

🛡 S

Home

Victor Hayashi edited this page 5 hours ago · 5 revisions

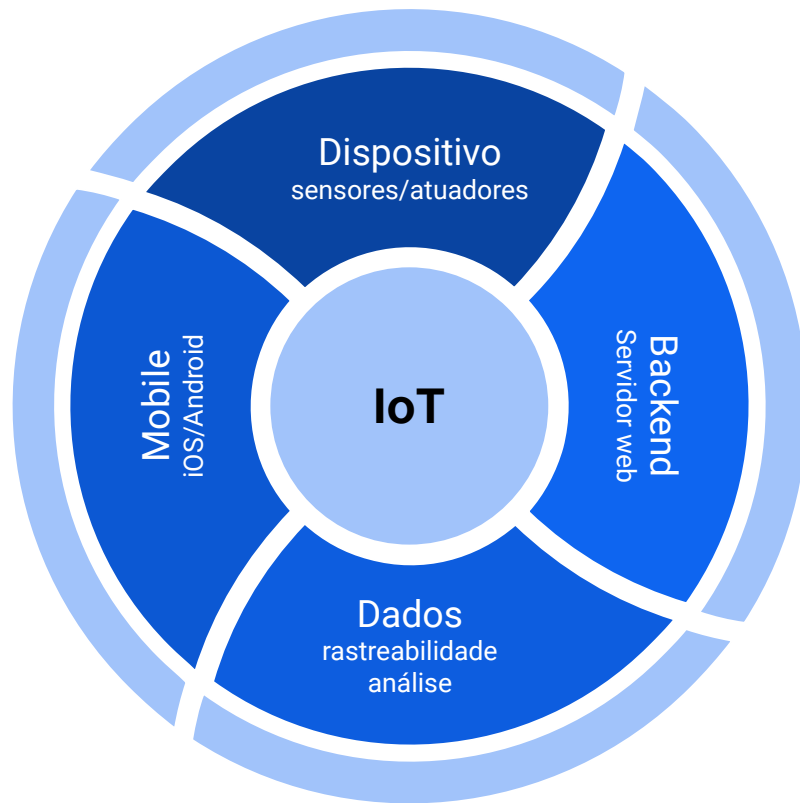
Welcome to the OKIoT wiki!

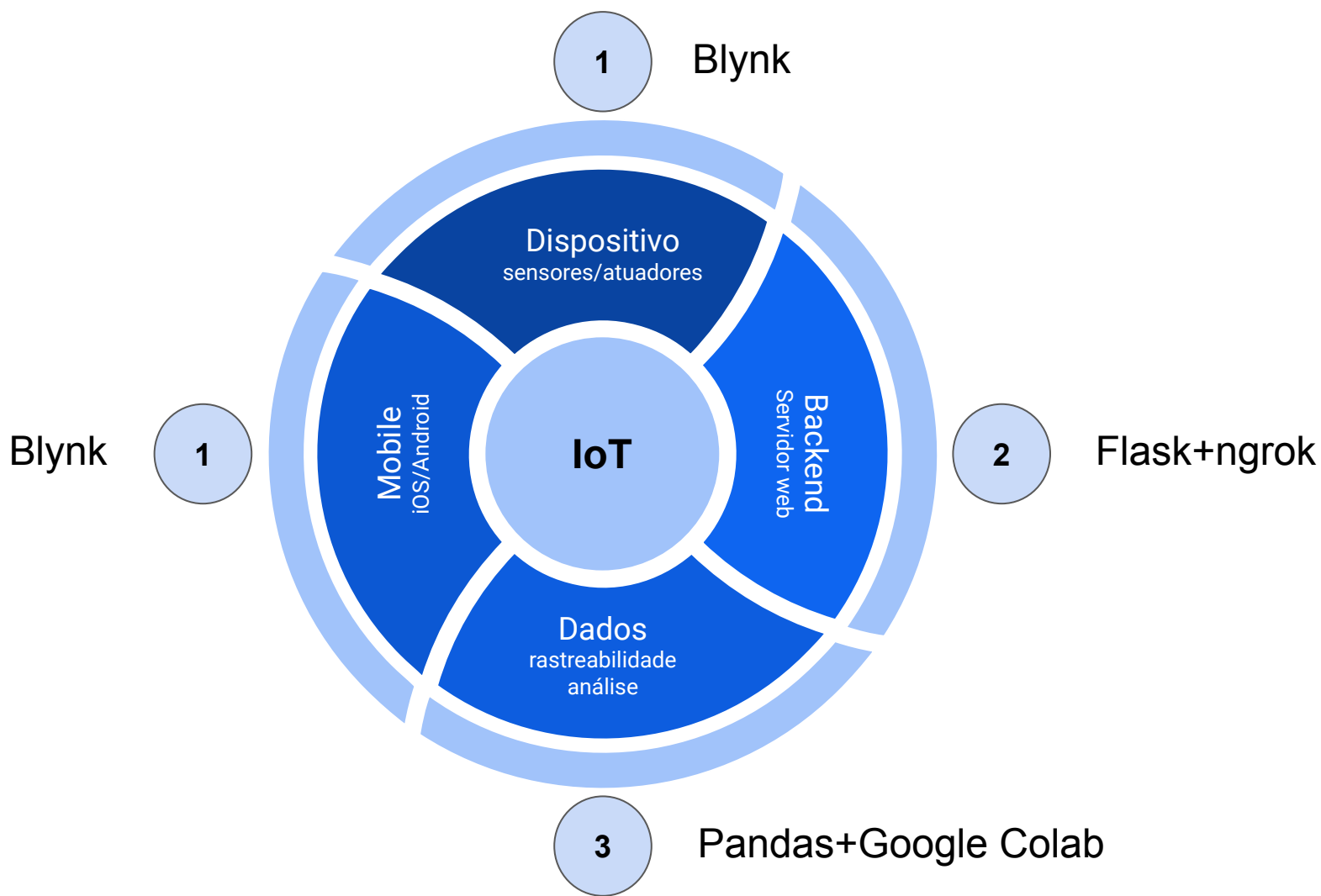
In the projects described below Victor Hayashi, Reginaldo Arakaki and Fabio Hayashi were responsible for the specification of business drivers, functional and non-functional require restrictions, architecture, implementation and integration / coordination between the vari modules (for example, Internet modules of Things-IoT, Artificial Intelligence-AI, as well as conversational interfaces). Software Engineering methods, like non-functional requiremer tolerance, availability and response time combined with agile methods of innovation and creation (e.g. Osterwalder and Design Thinking) were used in the various projects, and us

Fim da parte 1

+55 11 94870-6144

victortakashih@gmail.com

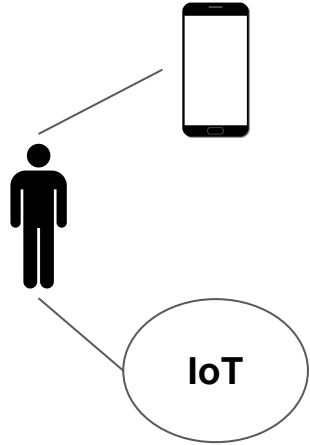




Arquitetura

LOCAL

CLOUD

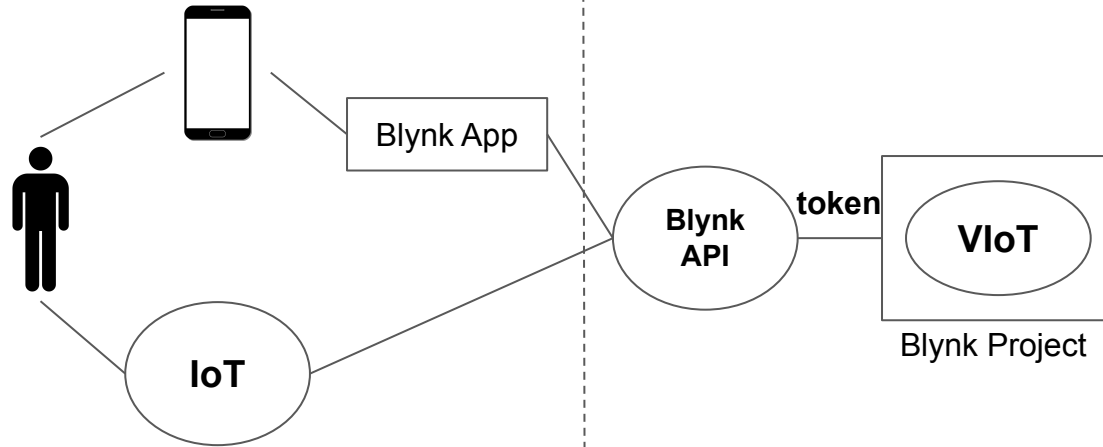


Tutoriais

Arquitetura

LOCAL

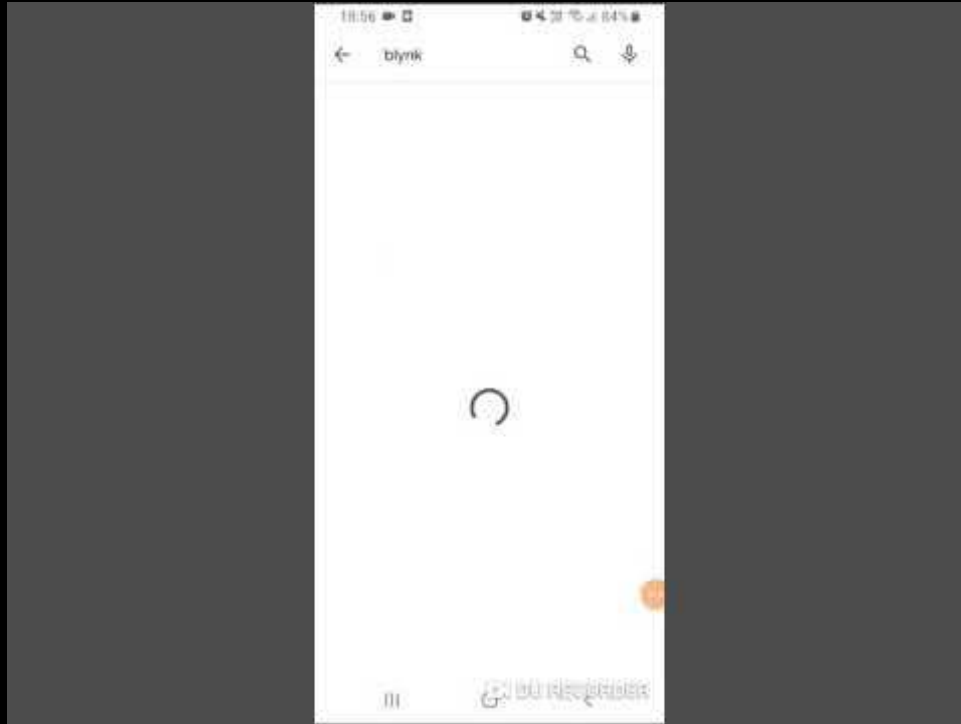
CLOUD



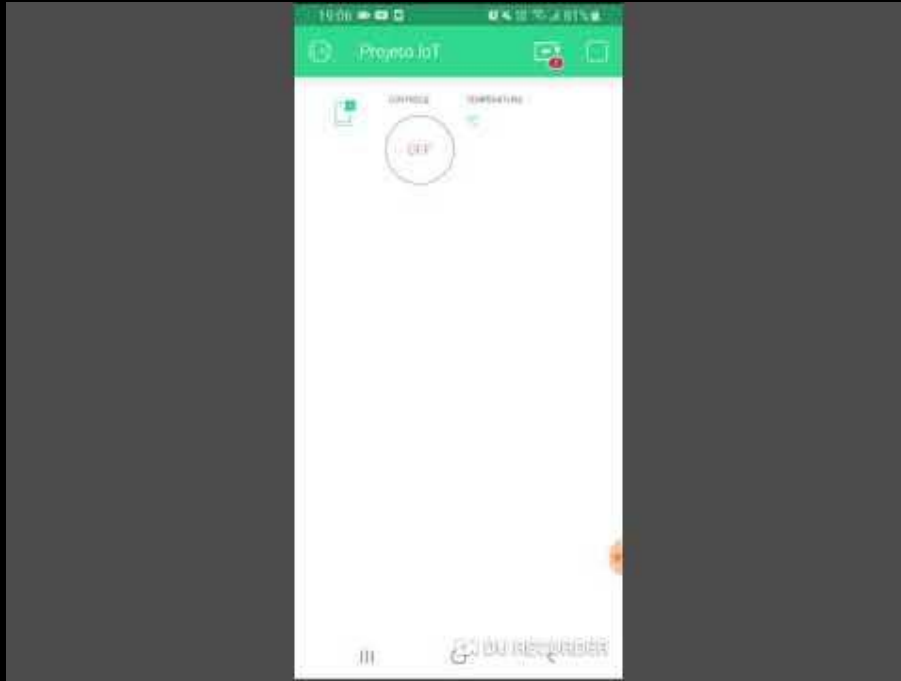
Tutoriais

1. Blynk: Cloud IoT com interface Mobile
 - a. Mobile App Android/iOS + HTTP request

1A Blynk: Cloud IoT com interface Mobile



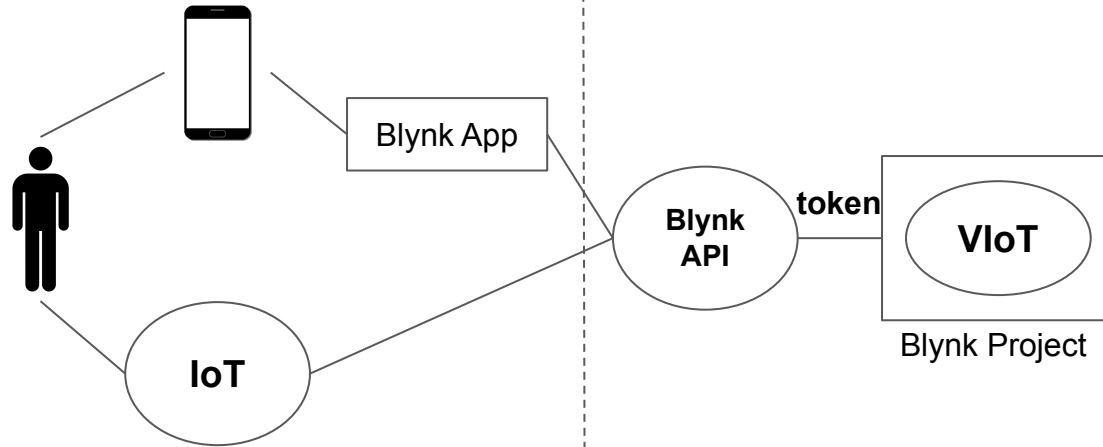
1A Blynk: Cloud IoT com interface Mobile e API



Arquitetura

LOCAL

CLOUD



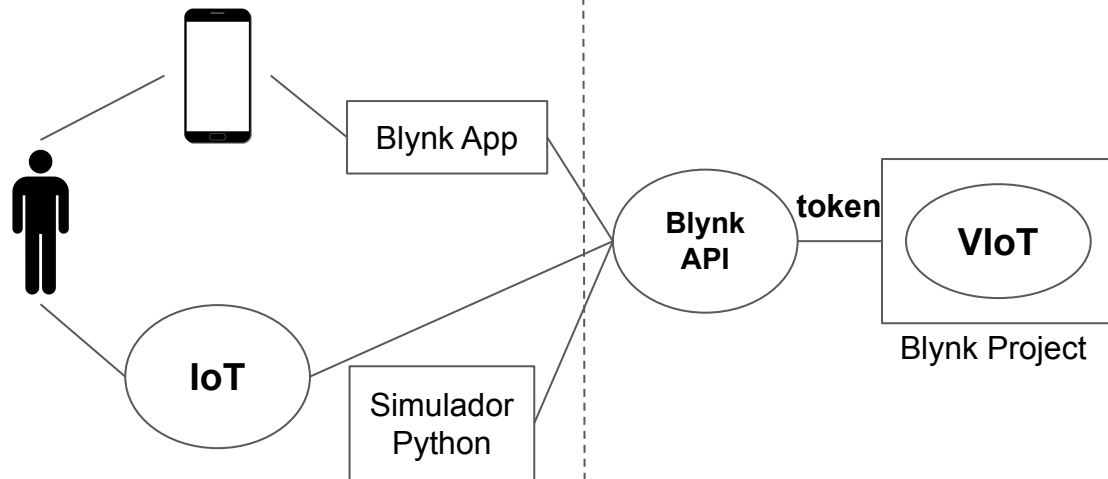
Tutoriais

1. Blynk: Cloud IoT com interface Mobile
 - a. Mobile App Android/iOS + HTTP request

Arquitetura

LOCAL

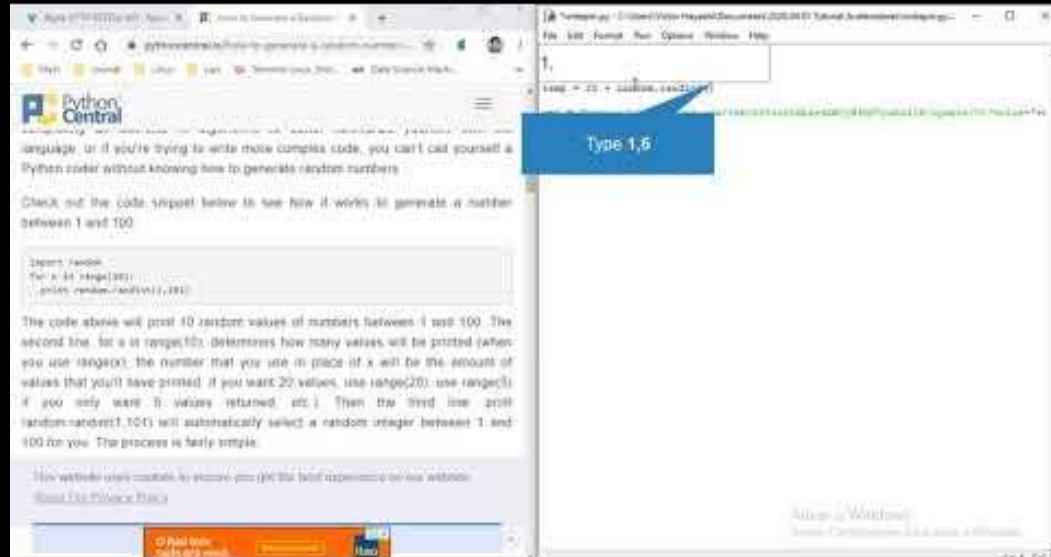
CLOUD



Tutoriais

1. Blynk: Cloud IoT com interface Mobile
 - a. Mobile App Android/iOS + HTTP request
 - b. Simular comportamento IoT com Python

1B Blynk: Simular comportamento IoT com Python



The image shows a side-by-side comparison of a web browser and a Python IDE. The browser window on the left displays a Python Central article titled "How to generate a random number in Python". The article explains that to generate random numbers in Python, one can use the `random` module. It provides a code snippet:

```
import random
for i in range(10):
    print(random.randint(1,100))
```

 The article further explains that `range(x)` determines the number of values to be printed, and `randint(1,100)` selects a random integer between 1 and 100. The IDE window on the right shows the same code snippet being typed into a text area. A blue callout box with the text "Type 1,6" points to the first parameter of the `randint` function in the code.

Python Central

language, or if you're trying to write more complex code, you can't call yourself a Python coder without knowing how to generate random numbers.

Check out the code snippet below to see how it works to generate a number between 1 and 100.

```
import random
for i in range(10):
    print(random.randint(1,100))
```

The code above will print 10 random values of numbers between 1 and 100. The second line, for `i in range(10)`, determines how many values will be printed (when you use `range(x)`, the number that you use in place of `x` will be the amount of values that you'll have printed; if you want 20 values, use `range(20)`; use `range()` if you only want 0 values returned, etc.). Then the third line, `print(random.randint(1,100))` will automatically select a random integer between 1 and 100 for you. The process is fairly simple.

This website uses cookies to enhance your browsing experience. [View Our Privacy Policy](#)

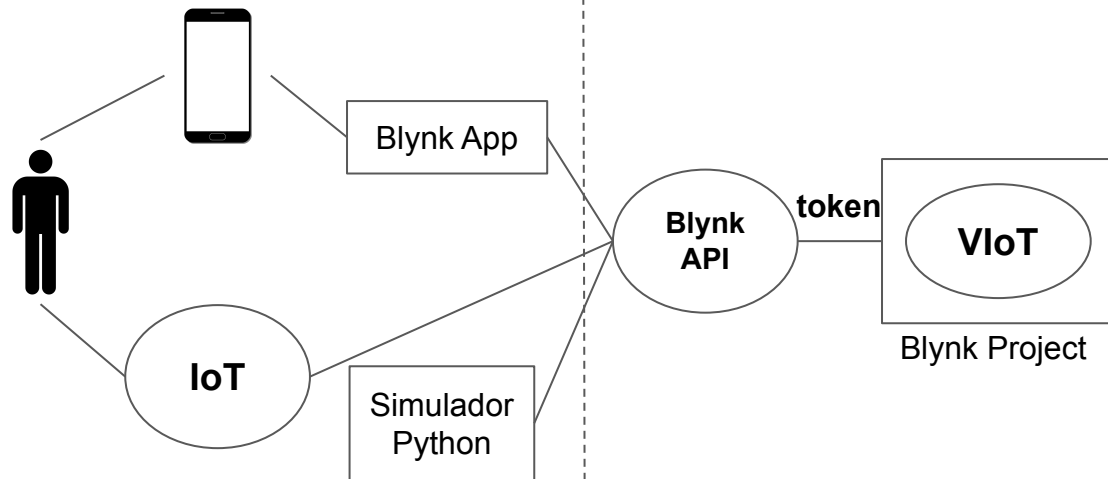
Allegro & Writings

Python Central

Arquitetura

LOCAL

CLOUD



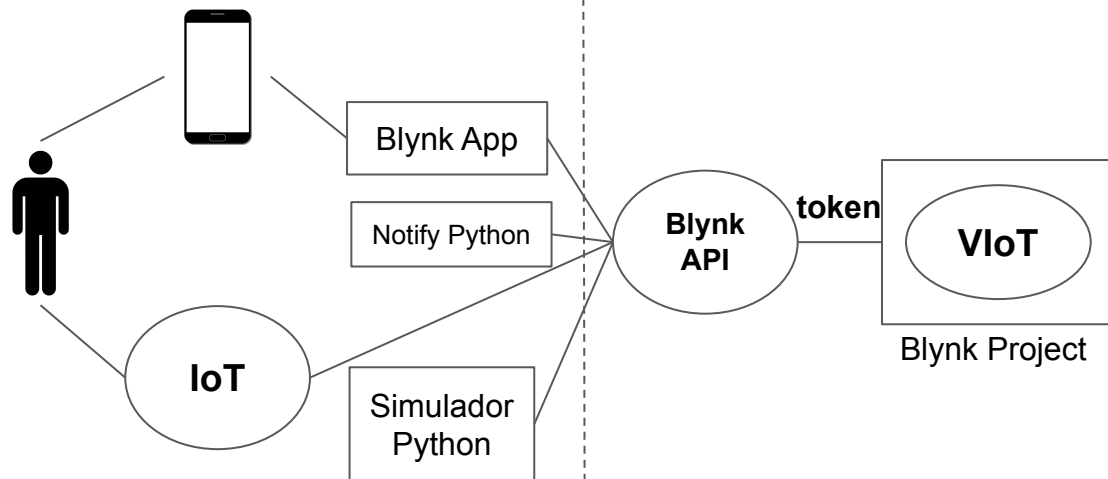
Tutoriais

1. Blynk: Cloud IoT com interface Mobile
 - a. Mobile App Android/iOS + HTTP request
 - b. Simular comportamento IoT com Python

Arquitetura

LOCAL

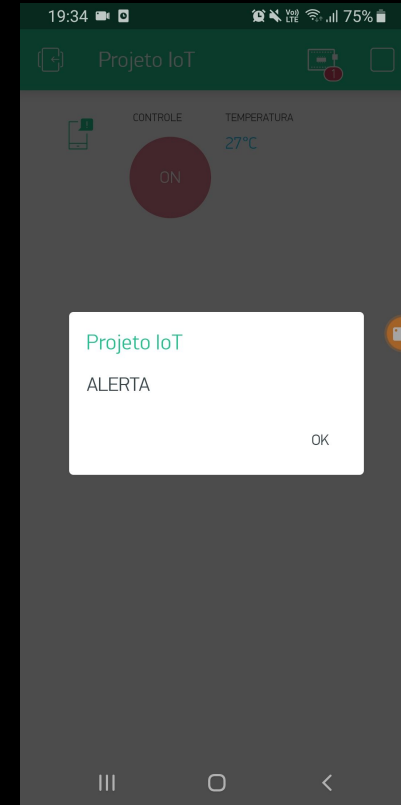
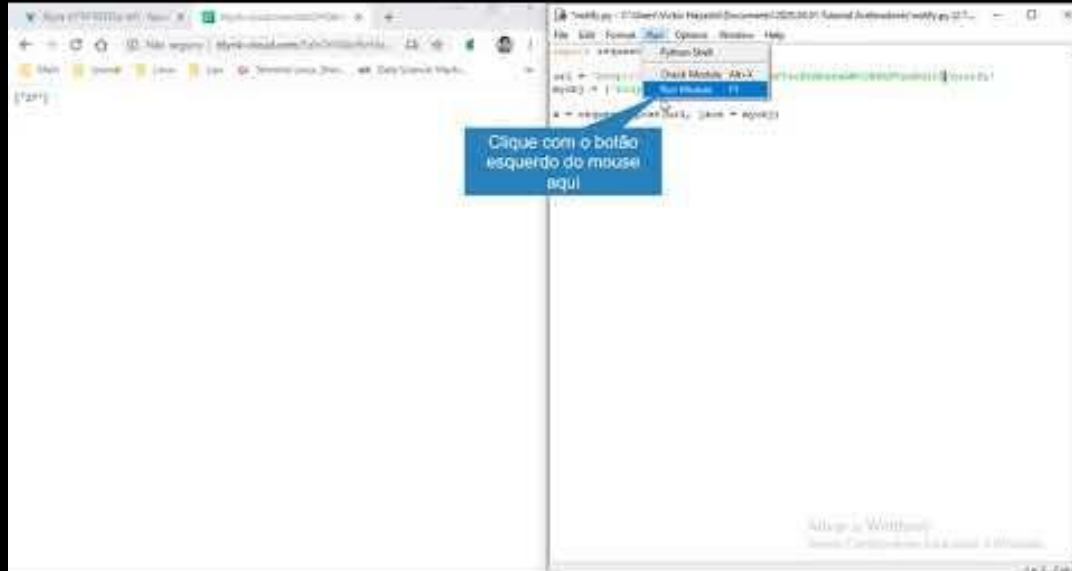
CLOUD



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

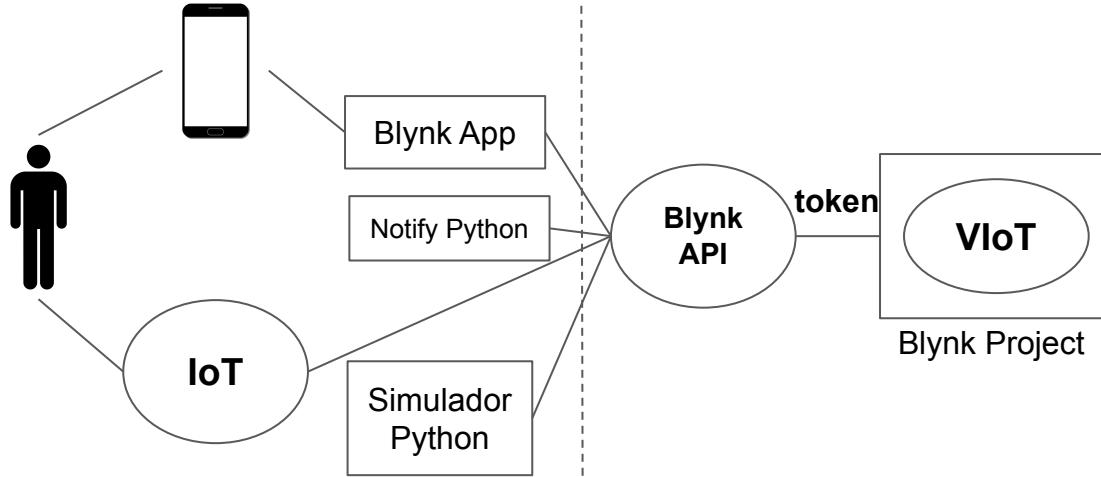
1C Usar API para notificação no celular



Arquitetura

LOCAL

CLOUD



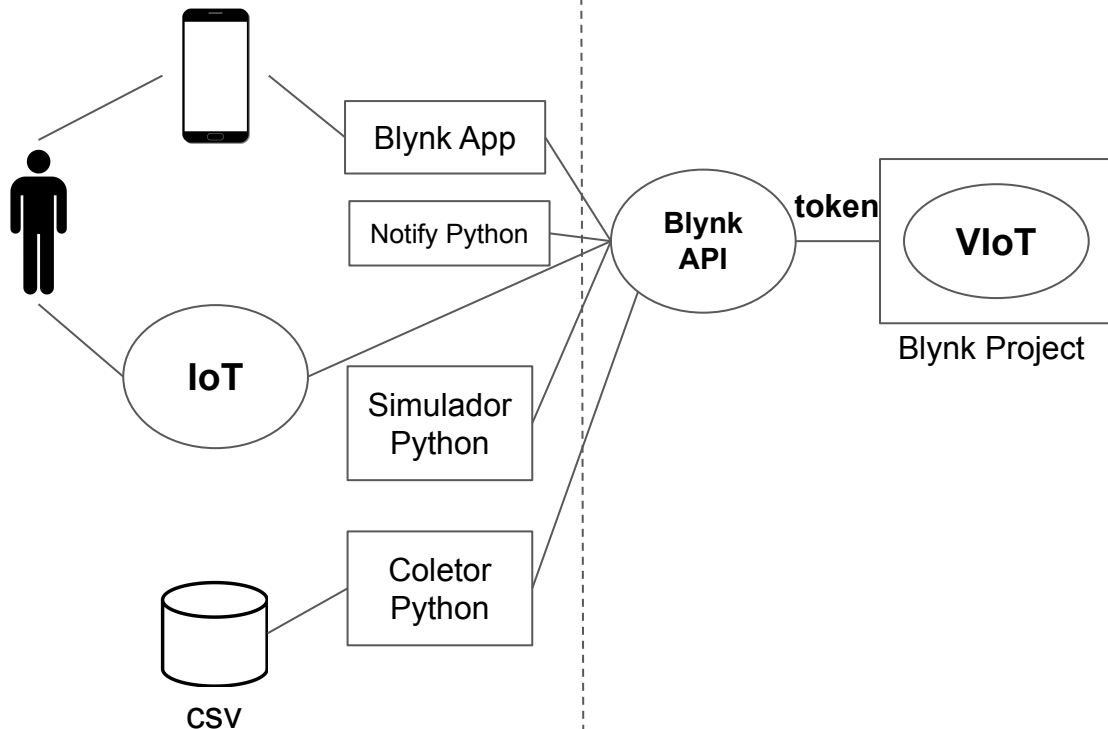
Tutoriais

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

Arquitetura

LOCAL

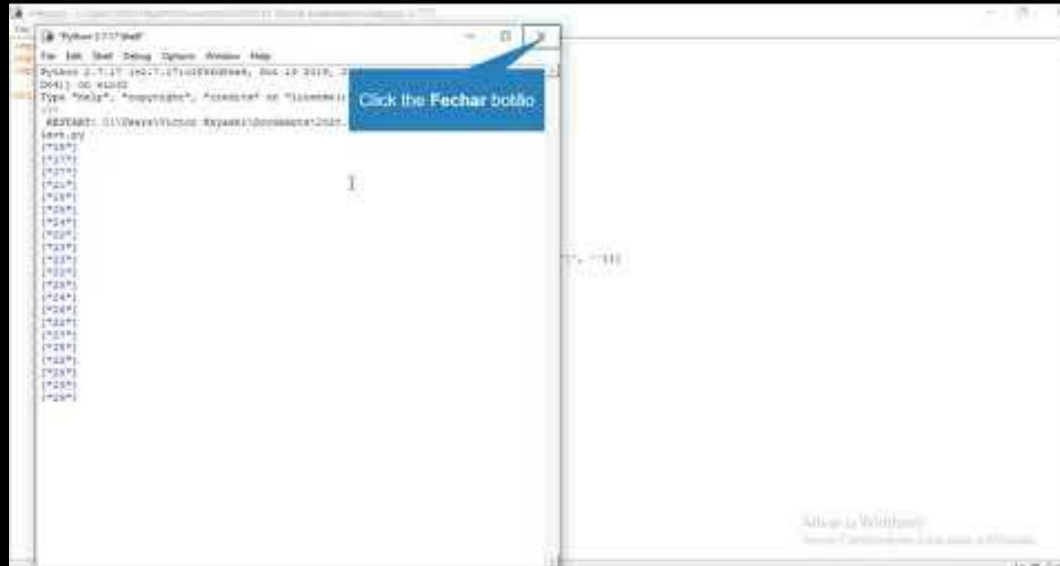
CLOUD



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

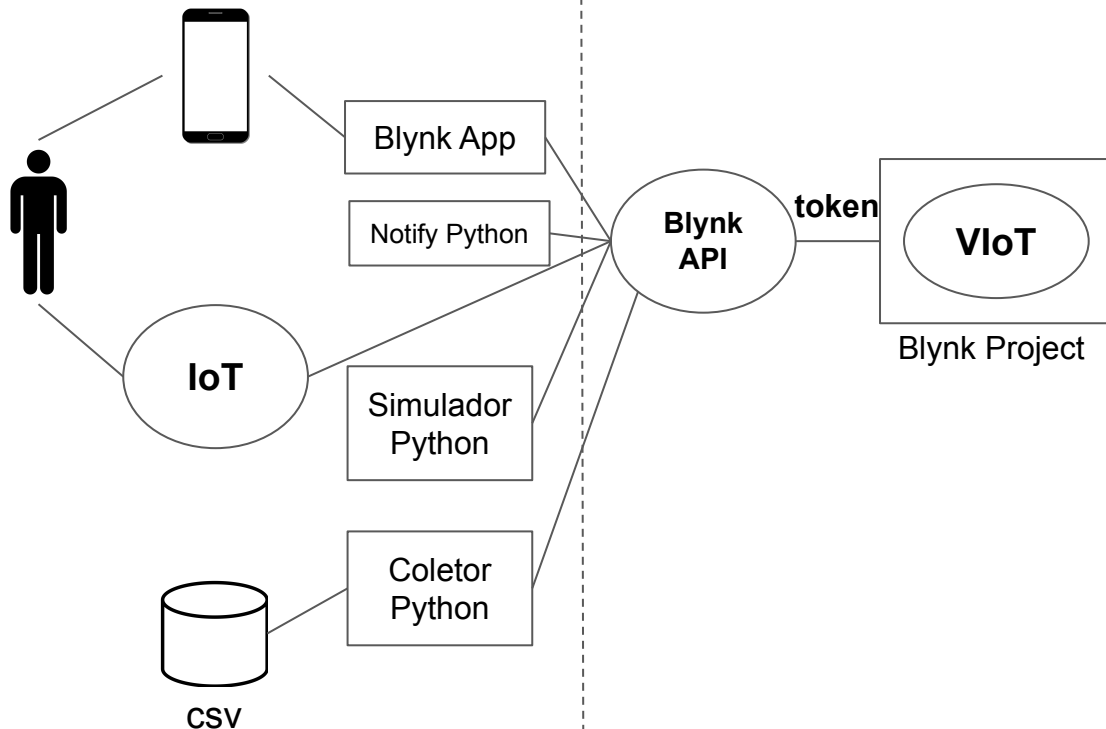
2A Dados: Coletor Python



Arquitetura

LOCAL

CLOUD



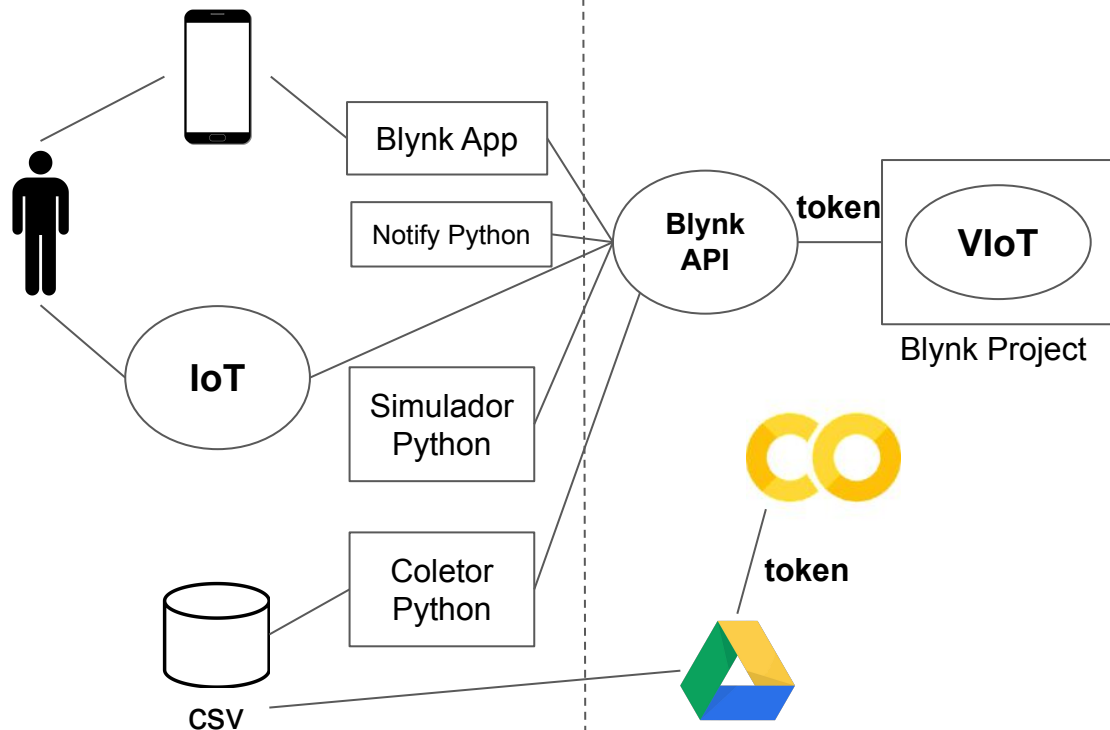
Tutoriais

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

Arquitetura

LOCAL

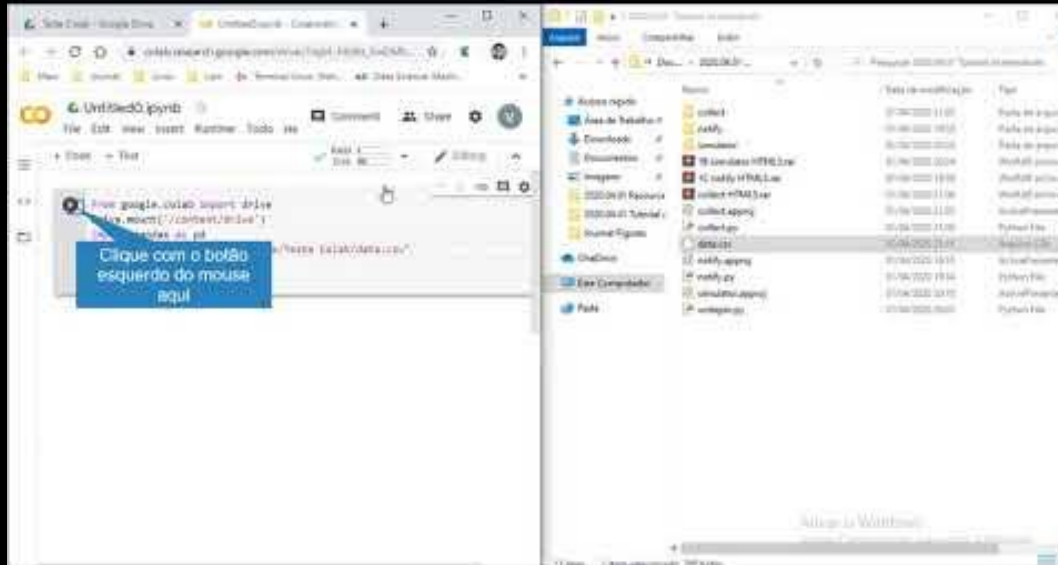
CLOUD



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

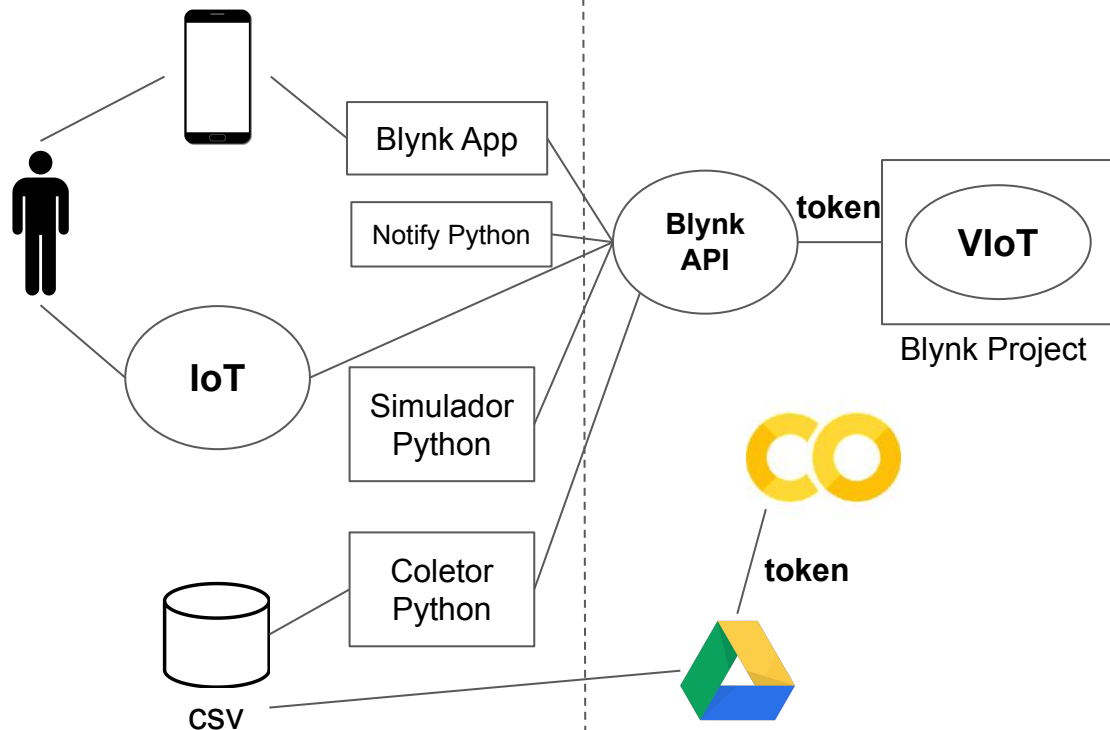
2B Dados: Google Colab e Google Drive



Arquitetura

LOCAL

CLOUD



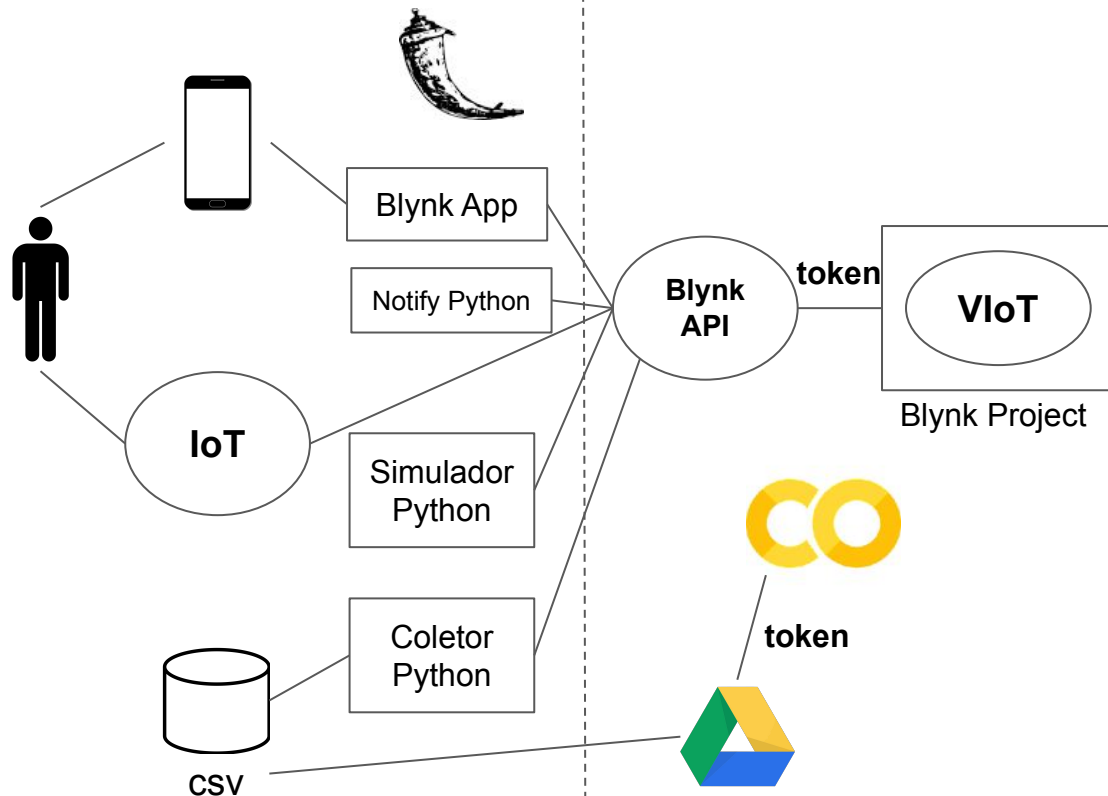
Tutoriais

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

Arquitetura

LOCAL

CLOUD



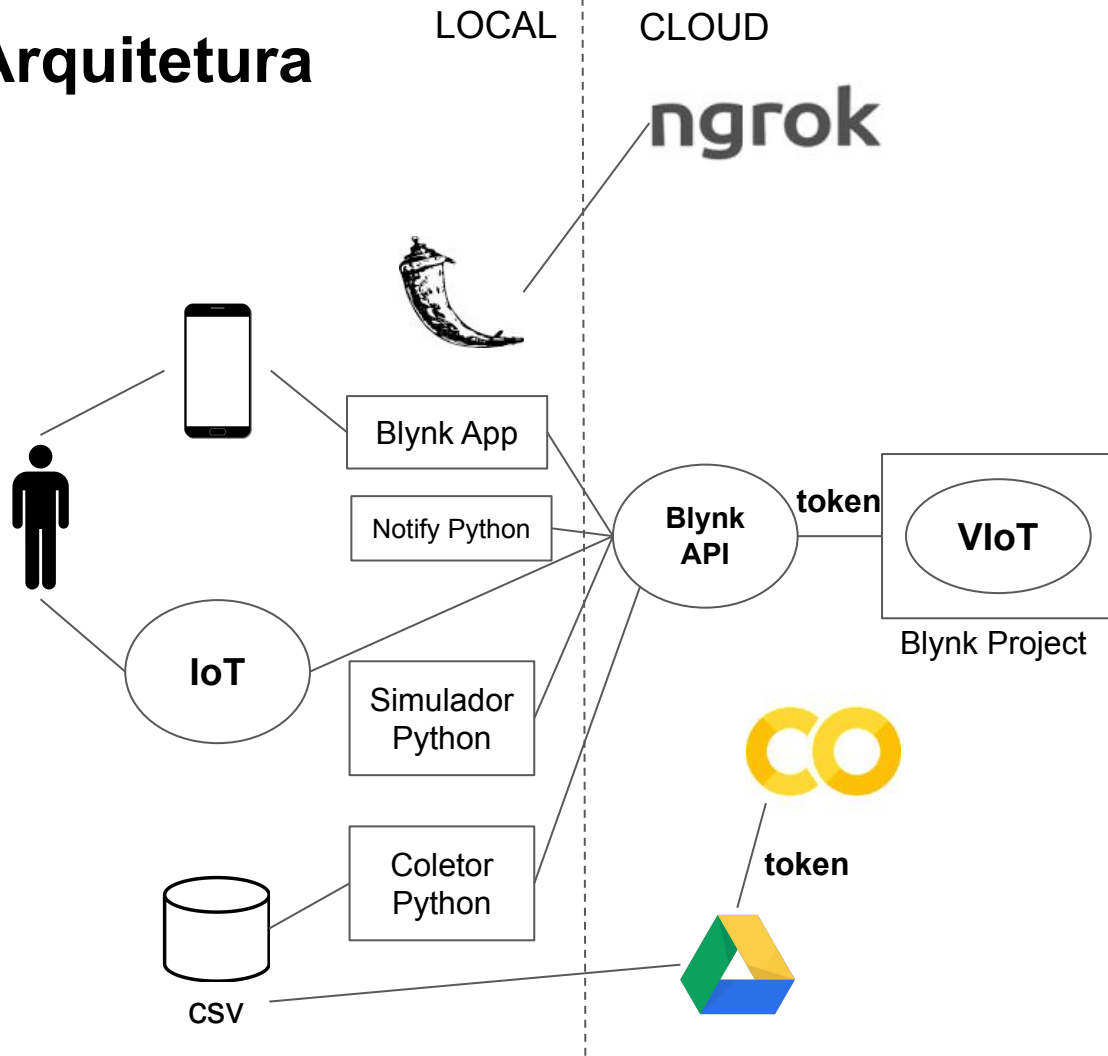
Tutoriais

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

Arquitetura

LOCAL

CLOUD



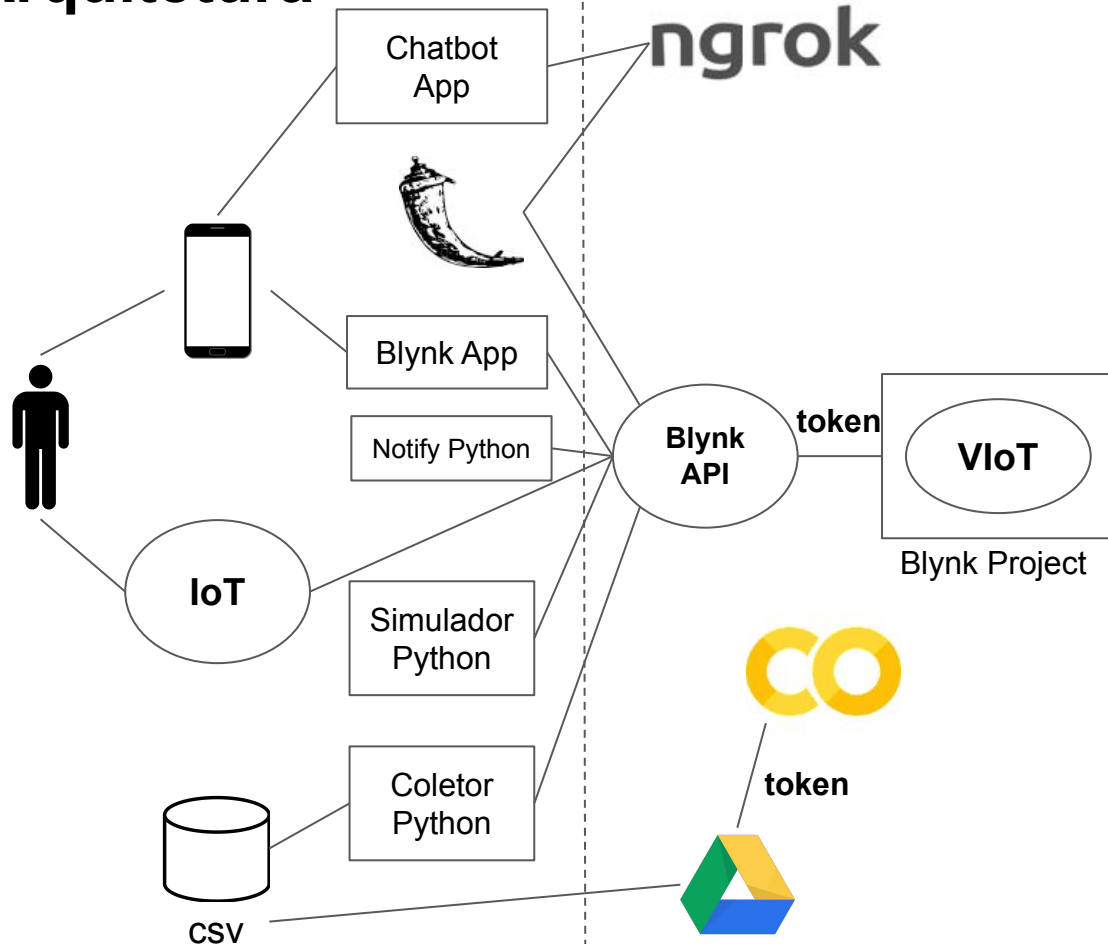
Tutoriais

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

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

Tutoriais Passo a Passo e Códigos Python

