Agendamento de tarefas com Rocketry

Live de Python # 214

Roteiro

1. Agendamento de tarefas

Alguns conceitos iniciais

2. Rocketry

Conhecendo a biblioteca

3. Pipelines

Como encadear operações

4. Um projetinho

A hora que a gente faz qualquer coisa sem pensar

picpay.me/dunossauro

apoia.se/livedepython

pix.dunossauro@gmail.com

Ajude o projeto <3

Acássio Anjos, Ademar Peixoto, Adilson Herculano, Alexandre Harano, Alexandre Lima, Alexandre Souza, Alexandre Takahashi, Alexandre Villares, Alex Lima, Alynne Ferreira, Alysson Oliveira, Ana Carneiro, Ana Padovan, Andre Azevedo, André Rafael, Aquiles Coutinho, Arnaldo Turque, Aurelio Costa, Bruno Freitas, Bruno Guizi, Bruno Oliveira, Bruno Ramos, Caio Nascimento, Carlos Alipio, Christiano Morais, Clara Battesini, Daniel Freitas, Daniel Haas, Danilo Segura, David Kwast, Delton Porfiro, Dhyeives Rodovalho, Diego Farias, Diego Guimarães, Dilenon Delfino, Dino Aguilar, Diogo Albuquerque, Diogo Paschoal, Douglas Bastos, Douglas Braga, Douglas Zickuhr, Dutofanim Dutofanim, Eliel Lima, Elton Silva, Emerson Rafael, Érico Andrei, Eugenio Mazzini, Euripedes Borges, Evandro Avellar, Everton Silva, Fabiano Tomita, Fabio Barros, Fábio Barros, Fabio Castro, Fábio Thomaz, Felipe Rodrigues, Fernanda Prado, Fernando Rozas, Flávio Meira, Flavkaze Flavkaze, Gabriel Barbosa, Gabriel Nascimento, Gabriel Simonetto, Geandreson Costa, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Guilherme Piccioni, Gustavo Dettenborn, Gustavo Suto, Heitor Fernandes, Henrique Junqueira, Hugo Cosme, Israel Gomes, Italo Silva, Jair Andrade, Jairo Lenfers, Janael Pinheiro, João Lugão, João Paulo, João Rodrigues, Joelson Sartori, Johnny Tardin, Jonatas Leon, Jonatas Oliveira, Jônatas Silva, José Gomes, Joseíto Júnior, Jose Mazolini, José Pedro, Juan Gutierrez, Juliana Machado, Júlio Gazeta, Julio Silva, Kaio Peixoto, Leandro Miranda, Leonardo Mello, Leonardo Nazareth, Leonardo Rodrigues, Lucas Mello, Lucas Mendes, Lucas Oliveira, Lucas Polo, Lucas Simon, Lucas Teixeira, Lucas Valino, Luciano Silva, Luciano Teixeira, Luiz Junior, Luiz Lima, Luiz Paula, Luiz Perciliano, Maicon Pantoja, Maiquel Leonel, Marcelino Pinheiro, Marcelo Matte, Márcio Martignoni, Marcio Moises, Marco Mello, Marcos Gomes, Marco Yamada, Maria Clara, Marina Passos, Mateus Lisboa, Matheus Cortezi, Matheus Silva, Matheus Vian, Mírian Batista, Murilo Andrade, Murilo Cunha, Murilo Viana, Natan Cervinski, Nathan Branco, Nicolas Teodosio, Osvaldo Neto, Patricia Minamizawa, Paulo Braga, Paulo Tadei, Pedro Henrique, Pedro Pereira, Peterson Santos, P Muniz, Priscila Santos, Rafael Lopes, Rafael Rodrigues, Rafael Romão, Ramayana Menezes, Regis Tomkiel, Renato Veirich, Ricardo Silva, Riverfount Riverfount, Robson Maciel, Rodrigo Alves, Rodrigo Cardoso, Rodrigo Freire, Rodrigo Oliveira, Rodrigo Quiles, Rodrigo Vaccari, Rodrigo Vieira, Rogério Nogueira, Rogério Sousa, Ronaldo Silva, Ronaldo Silveira, Rui Jr, Samanta Cicilia, Thalles Rosa, Thiago Araujo, Thiago Bueno, Thiago Curvelo, Thiago Moraes, Thiago Oliveira, Thiago Salgado, Thiago Souza, Tiago Minuzzi, Tony Dias, Valcilon Silva, Valdir Tegon, Victor Wildner, Vinícius Bastos, Vitor Luz, Vladimir Lemos, Walter Reis, Wellington Abreu, Wesley Mendes, William Alves, Willian Lopes, Wilson Neto, Wilson Rocha, Xico Silvério, Yury Barros

Obrigado você

Agenda mento

De tarefas

Agendamento

Agendar uma tarefa é a arte de dizer quem **em determinando momento** um **código será executado**.

Por exemplo:

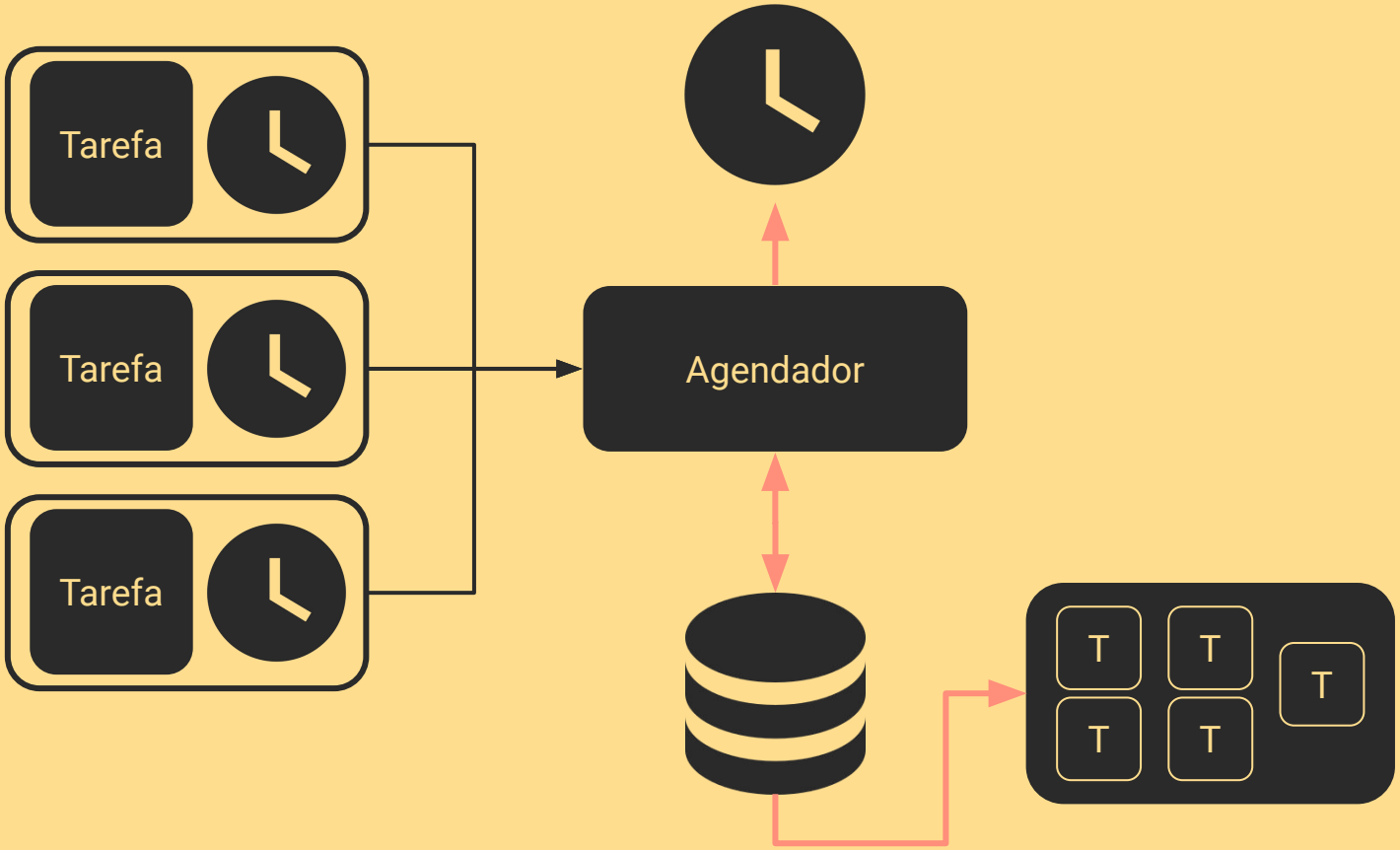
Execute o arquivo **tarefas.py** às **22:30** do **sábado**

Agendamento

Como geralmente funciona

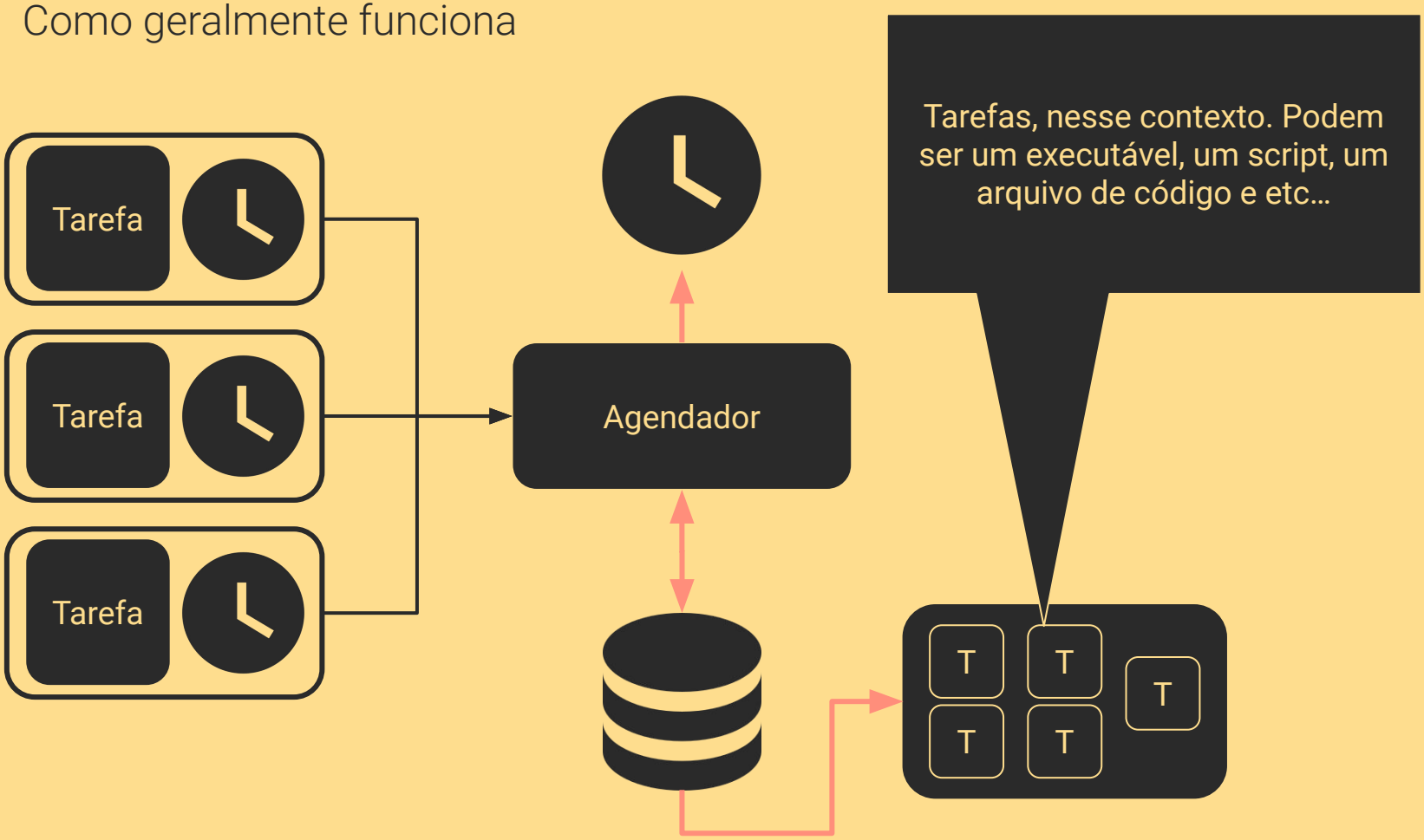
Agendamento

Como geralmente funciona



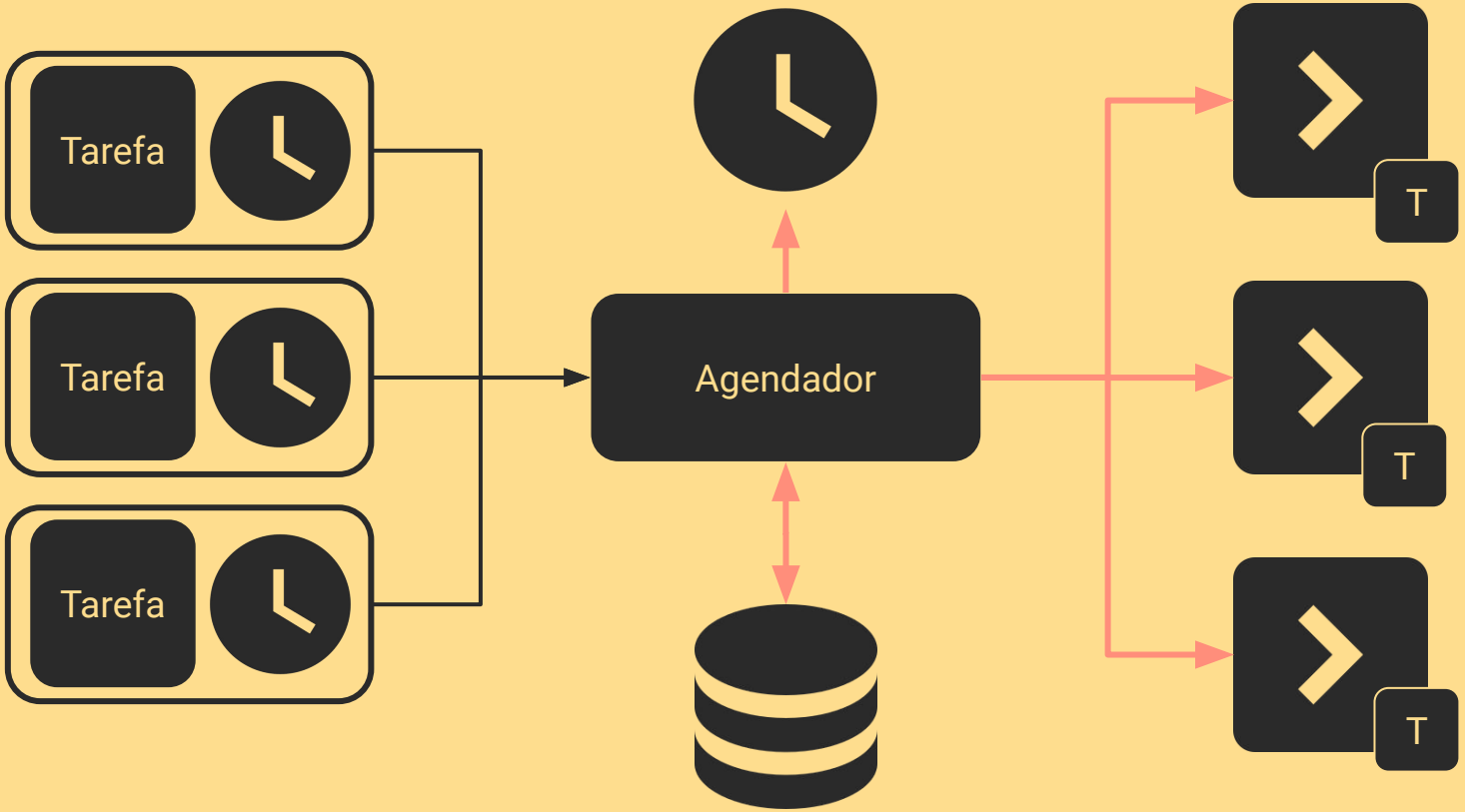
Agendamento

Como geralmente funciona



Agendamento

Como geralmente funciona



Agendamento de tarefas

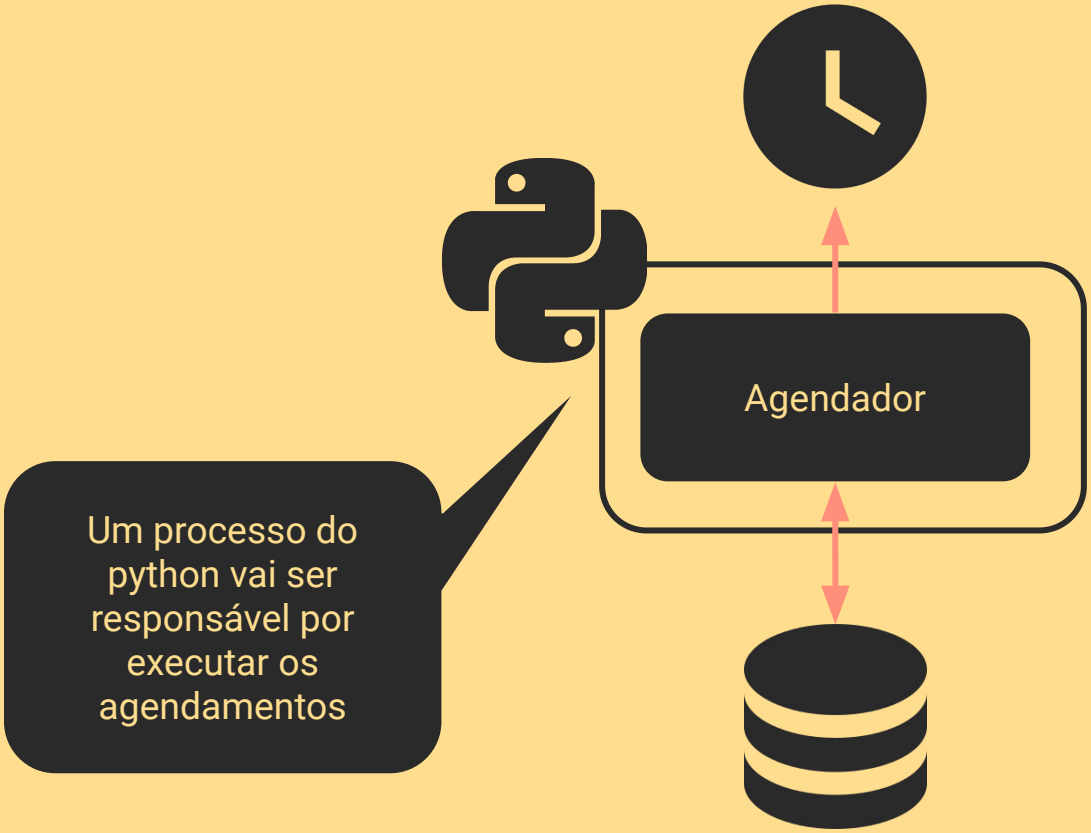
Temos diversas formas de agendar tarefas. Podemos usar um agendador de tarefas pronto, disponibilizado pelo sistema operacional.

- **TaskScheduler**: Windows
- **Cron + Contab**: GNU/Linux, BSDs e MacOS
- **Launchd**: MacOS

Ponto negativo: Por fazem parte do sistema, não existe uma maneira fácil de configurar agendamento multiplataforma.

Agendamento com Python

Se pensarmos em agendamento com python



Bibliotecas agendadoras

Existem diversas bibliotecas prontas para esse trabalho:

- Aplicações baseadas em agendamentos (**E**xtract, **T**ransform, **L**oad)
  - **Apache AirFlow** (Live de Python # 123)
  - **Prefect**

- Agendamento de tarefas distribuídas (executam em outro lugar)
  - **Celery Beat** (Live de Python # 159, **mas não falamos sobre agendamento**)

- Agendamento ao nível de sistema
  - **Supervisor** (trabalha com eventos entre processos)

Criação de agendadores

Em outros momentos, não precisamos construir uma grande aplicação que depende de agendamento. Ou como as ferramentas foram pensadas não atendem nossos problemas.

Temos bibliotecas a nosso favor:

- **Sched**: Biblioteca nativa para agendamento (Live de Python #50)
- **APScheduler**: Biblioteca muito utilizada. Sua API, porém, é muito complexa e carece de documentação
- **Rocketry**: Biblioteca nova, com um potencial incrível e bem documentada. Pode trabalhar com agendamentos simples como o **sched** e também é flexível para fluxos complexos. Como **airflow** e **prefect**

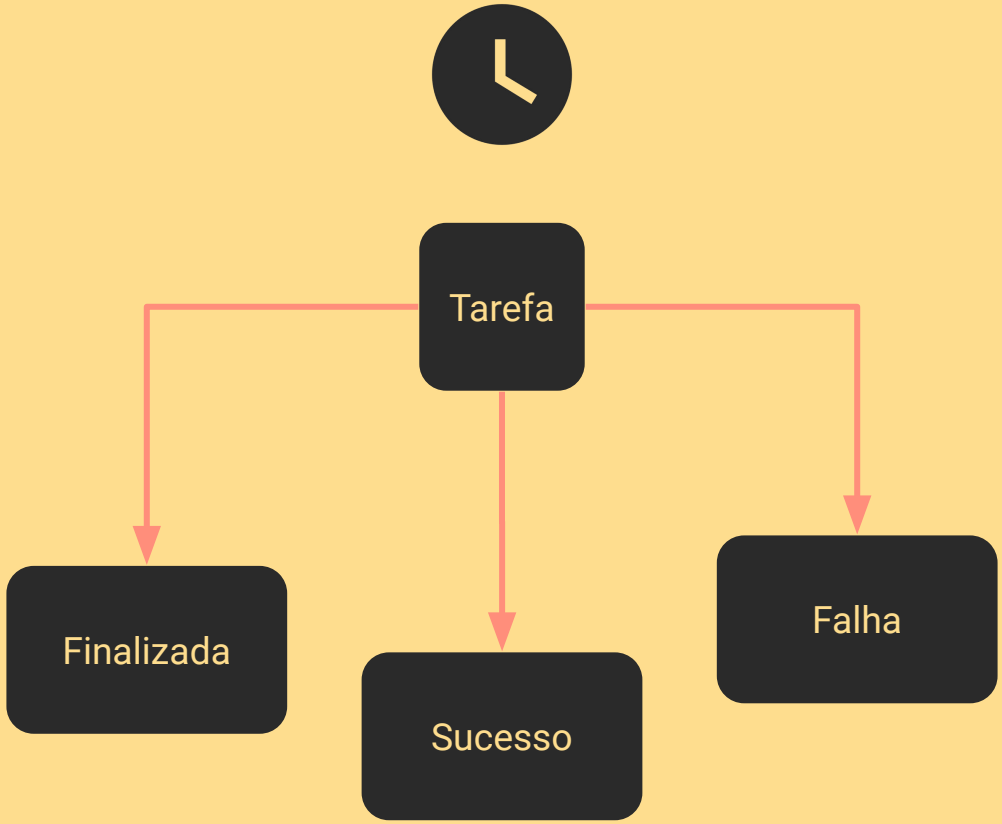
Agendamento vs Periodicidade

Uma tarefa tem duas formas de ser agendada. **Periodicamente** e **agendada**:

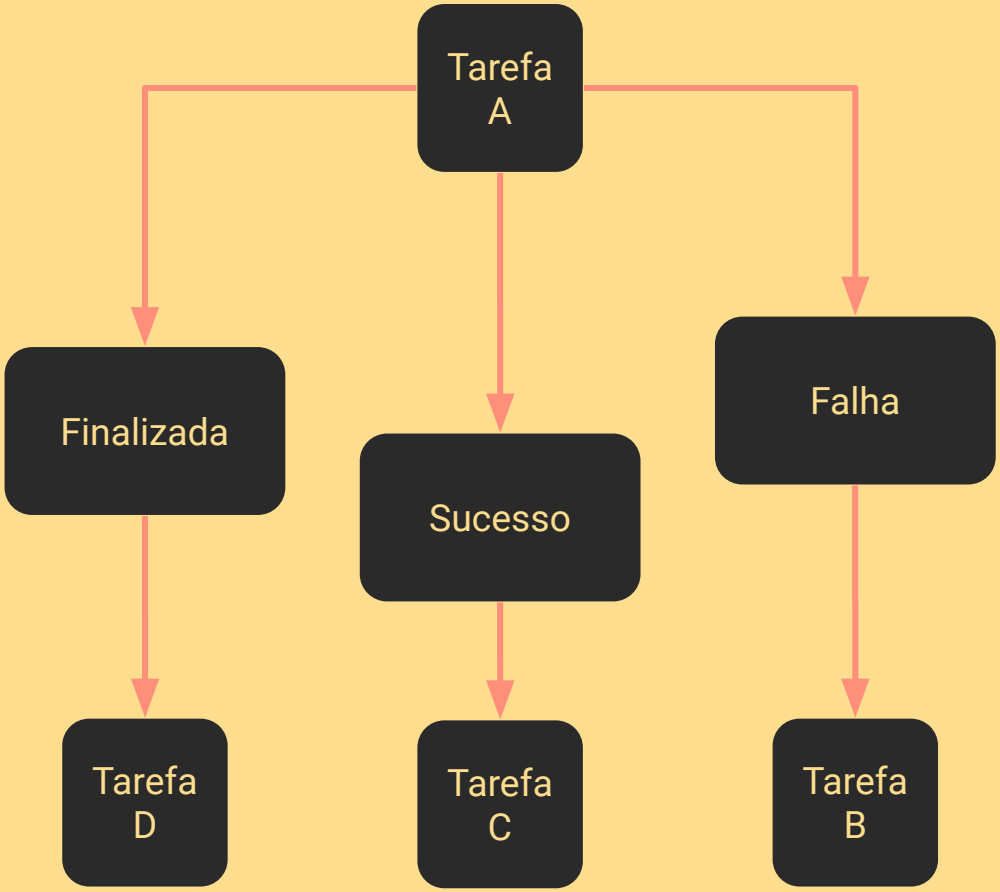
- Agendada:
  - Amanhã ao meio-dia
  - Este sábado as 3 da tarde
  - Dia 08 de agosto de 2022, entre meio-dia e 3 da tarde

- Periódica:
  - A cada minuto
  - Todos os dias, ao meio-dia
  - De segunda a sexta, às 14:34

Encadeamento de tarefas (pipeline)



Encadeamento de tarefas (pipeline)



Rocketry

A biblioteca que vamos usar

História

Rocketry é um projeto moderno para agendamento de tarefas. Criado em Out de 2021 por Mikael Koli.

Seu nome inicial era red-engine sendo alterado para Rocketry na versão 2.1 da biblioteca. Atualmente está na versão 2.2.0. A que usaremos nessa live.

pip install rocketry

Instalação

Estrutura básica

```python
from rocketry import Rocketry

app = Rocketry()

app.run()
```

Agendando nossa primeira tarefa

```python
# Exemplo_00.py
"""Rocketry executando `a_cada_minuto` todos os minutos."""
from rocketry import Rocketry

app = Rocketry()

@app.task('minutely')
def a_cada_minuto():
    print('Executando tarefa a cada minuto')

app.run()
```

Agendando nossa primeira tarefa

```python
# Exemplo_00.py
"""Rocketry executando `a_cada_minuto` todos os minutos."""
from rocketry import Rocketry

app = Rocketry()

@app.task('minutely')
def a_cada_minuto():
    print('Executando tarefa a cada minuto')

app.run()
```

Agendando nossa primeira tarefa

```python
# Exemplo_00.py
"""Rocketry executando `a_cada_minuto` todos os minutos."""
from rocketry import Rocketry

app = Rocketry()

@app.task('minutely')
def a_cada_minuto():
    print('Executando tarefa a cada minuto')

app.run()
```

Parâmetros de agendamento

```python
@app.task('minutely')
def a_cada_minuto(): ...

@app.task('hourly')
def a_cada_hora(): ...

@app.task('daily')
def a_cada_dia(): ...

@app.task('weekly')
def a_cada_semana(): ...

@app.task('monthly')
def a_cada_mes(): ...
```

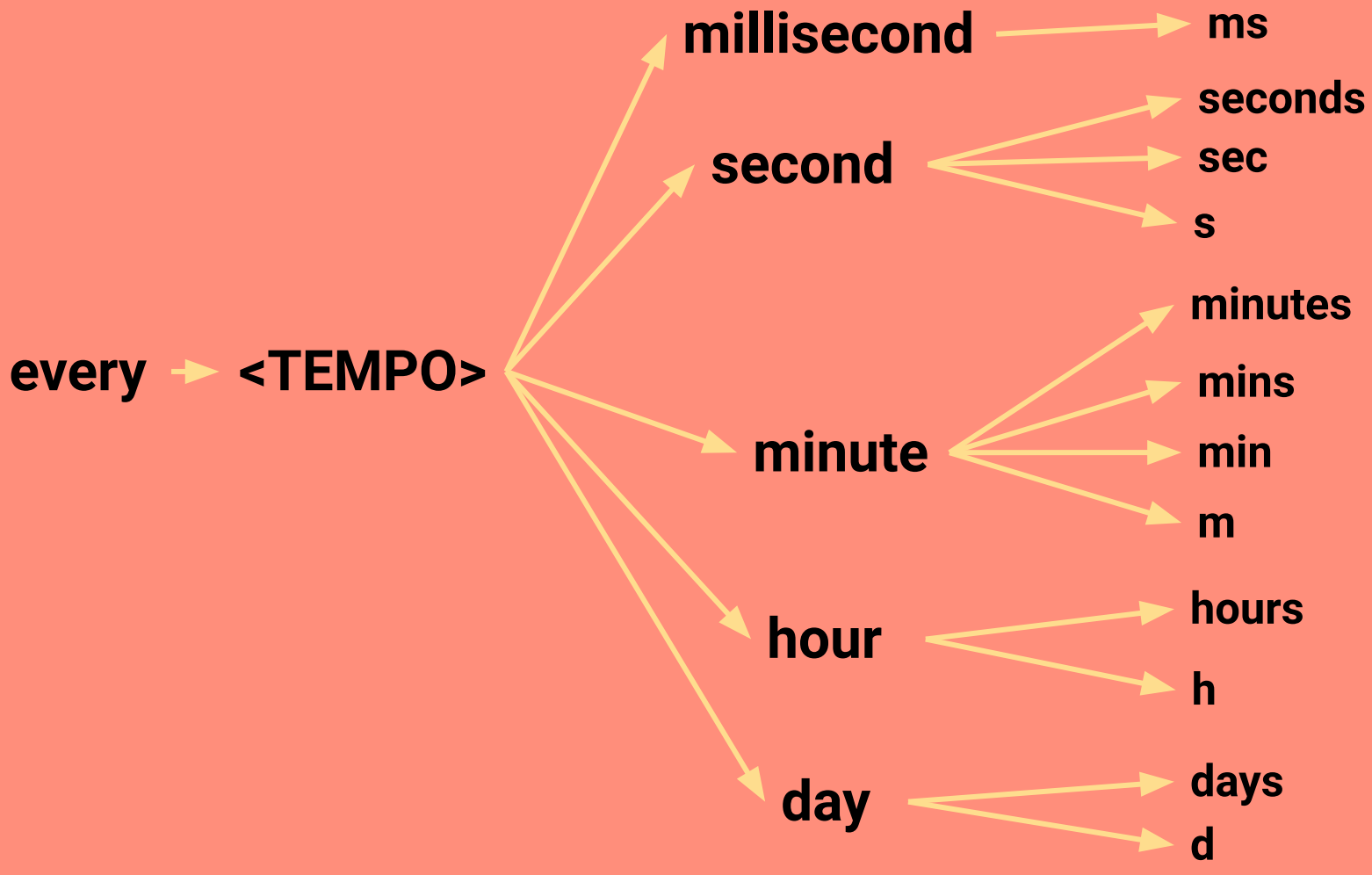
Caso não goste da sintaxe com strings

```python
# exemplo_01.py
"""Api de condicionais, para quem não gosta de strings"""
from rocketry import Rocketry
from rocketry.conds import minutely

app = Rocketry()

@app.task(minutely)
def a_cada_minuto():
    print('Executando tarefa a cada minuto')

app.run()
```

Sintaxe de conectivos periódicos

every	<numero>	<medida_de_tempo>
every	1	second
every	10	seconds
every	2	days

Sintaxe de conectivos periódicos (every)



Exemplos

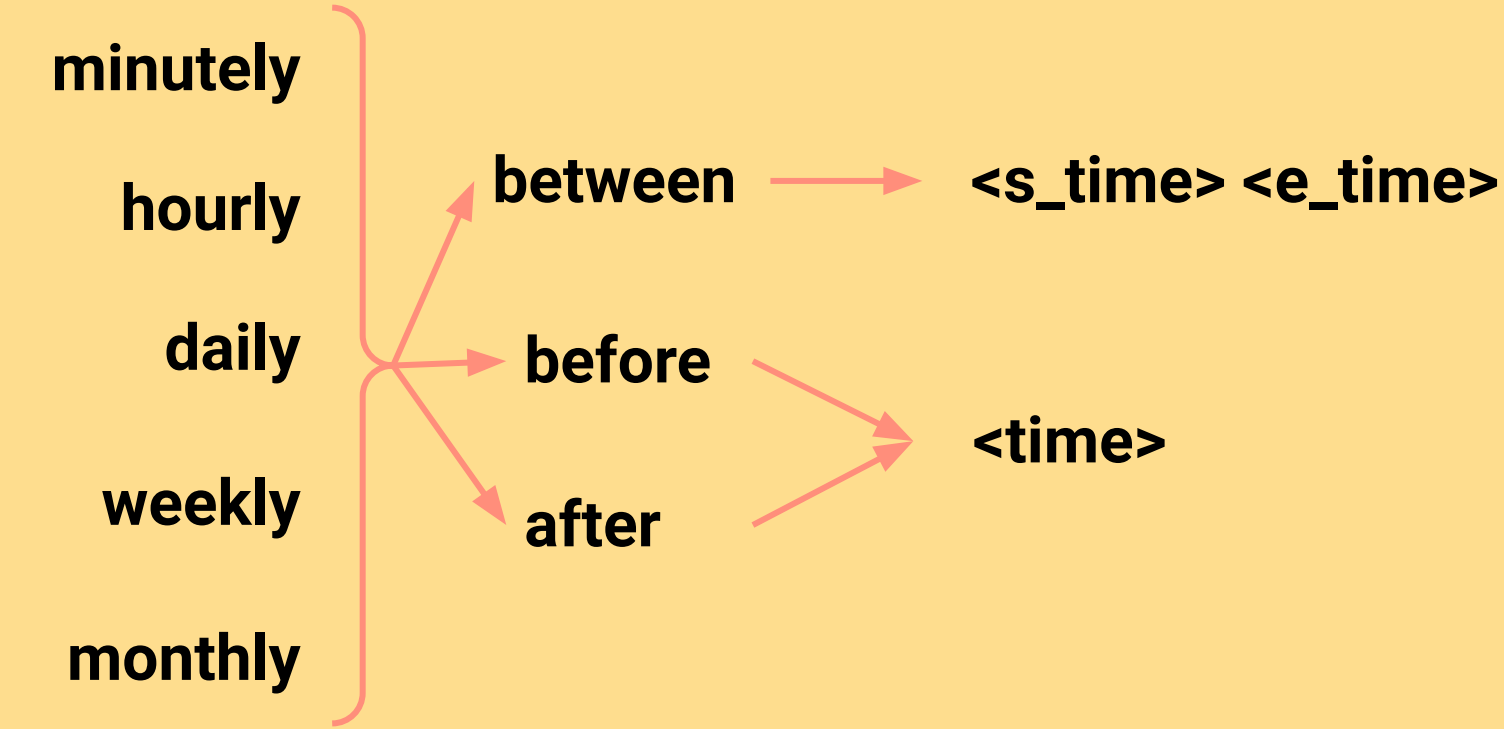
```python
# exemplo_02.py


@app.task('every 1s')
def a_cada_segundo(): ...

@app.task('every 1 day')
def a_cada_dia(): ...
```

```python
# exemplo_02.py
from rocketry.conds import every

@app.task(every('1d'))
def a_cada_dia(): ...


@app.task(every('1s'))
def a_cada_segundo(): ...
```

Sintaxe para restrições

objeto	<restrição>	<tempo>
every 1m	before	45
minutely	after	13
daily	between	13:00, 18:00

Sintaxe de restrições



Exemplos

```python
# exemplo_03.py
@app.task('hourly before 25:00')
def toda_hora_antes_do_minuto_25(): ...

@app.task('hourly after 01:00')
def toda_hora_depois_do_minuto_1(): ...

@app.task('daily after 23')
def diariamente_apos_as_23(): ...

@app.task('daily between 23:00 and 23:59')
def diariamente_entre_23_e_00(): ...

@app.task('weekly between Sunday and Monday')
def semanalmente_entre_domingo_e_segunda(): ...
```

```python
# exemplo_03.py
from rocketry.conds import hourly, daily, weekly

@app.task(hourly.before('25:00'))
def toda_hora_antes_do_minuto_25(): ...

@app.task(hourly.after('01:00'))
def toda_hora_depois_do_minuto_1(): ...

@app.task(daily.after('23:00'))
def diariamente_apos_as_23(): ...

@app.task(daily.between('23:00', '23:59'))
def diariamente_entre_23_e_00(): ...

@app.task(weekly.between('Monday', 'Sunday'))
def semanalmente_entre_domingo_e_segunda(): ...
```

Tempo exato (on, strating)

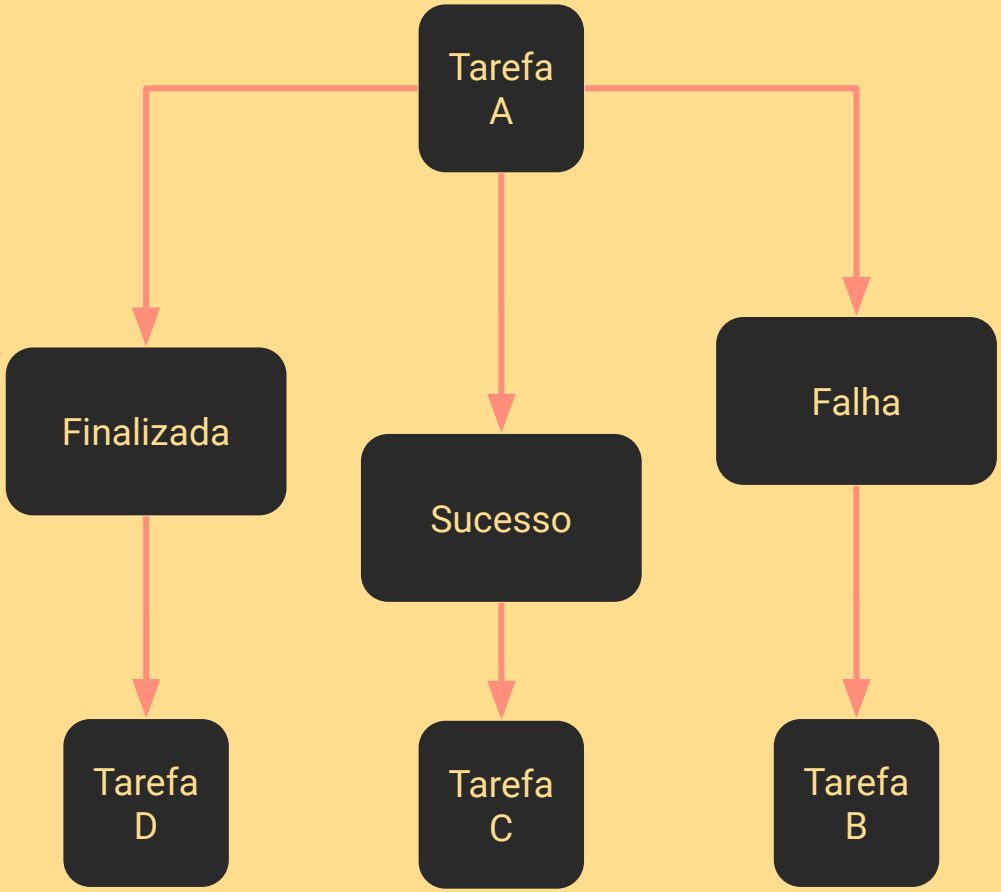
```python
# exemplo_04.py
from rocketry.conds import weekly, monthly

@app.task(weekly.on('Monday'))
def toda_segunda(): ...


@app.task(monthly.starting('8rd'))
def mensal_depois_do_dia_8(): ...
```

```python
# exemplo_04.py
@app.task('weekly on Monday')
def toda_segunda(): ...


@app.task('monthly starting 8rd')
def mensal_depois_do_dia_8(): ...
```

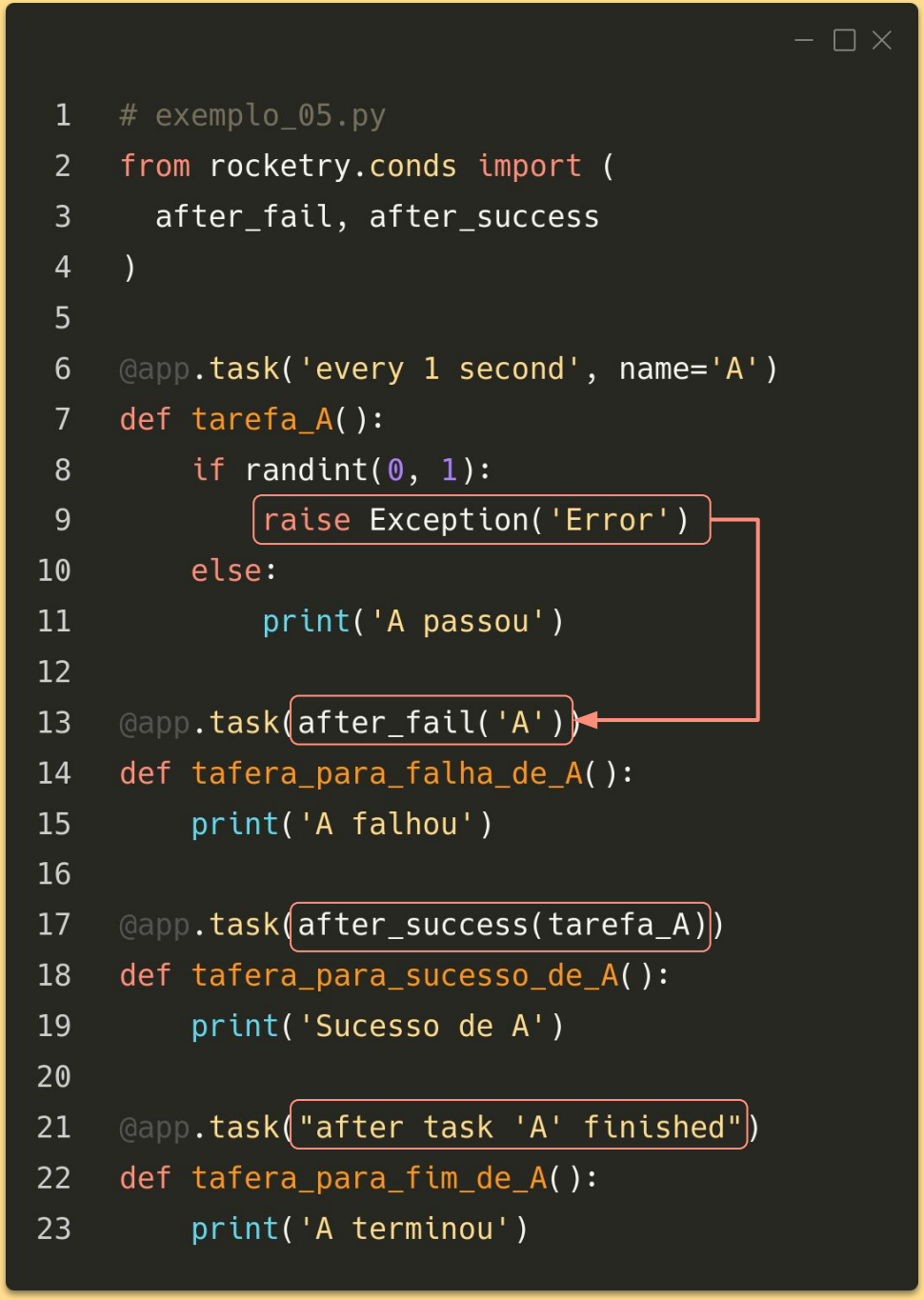
Pipes

Encadeando agendamentos

Criando pipelines




```python
# exemplo_05.py
from rocketry.conds import (
  after_fail, after_success
)

@app.task('every 1 second', name='A')
def tarefa_A():
    if randint(0, 1):
        raise Exception('Error')
    else:
        print('A passou')

@app.task(after_fail('A'))
def tafera_para_falha_de_A():
    print('A falhou')

@app.task(after_success(tarefa_A))
def tafera_para_sucesso_de_A():
    print('Sucesso de A')

@app.task("after task 'A' finished")
def tafera_para_fim_de_A():
    print('A terminou')
```


# exemplo_05.py
from rocketry.conds import (
after_fail, after_success
)
@app.task('every 1 second', name='A')
def tarefa_A():
if randint(0, 1):
raise Exception('Error')
else:
print('A passou')
@app.task(after_fail('A'))
def tafera_para_falha_de_A():
print('A falhou')
@app.task(after_success(tarefa_A))
def tafera_para_sucesso_de_A():
print('Sucesso de A')
@app.task("after task 'A' finished")
def tafera_para_fim_de_A():
print('A terminou')

Adicionar Logs [exemplo_06.py]

Entender o que deu errado na tarefa com logs

```
1   import logging
2
3   # Caso nunca tenham usado loggin - Lives 48 e 198
4   handler = logging.StreamHandler()
5   handler.setLevel(logging.DEBUG)
6   task_logger = logging.getLogger('rocketry.task')
7   task_logger.addHandler(handler)
```

Passando a resposta de uma task pra outra

```python
from rocketry.args import Return

@app.task('every 1 second')
def tarefa_A():
    return 'Sucesso'

@app.task(after_success(tarefa_A))
def tafera_para_sucesso_de_A(value=Return(tarefa_A)):
    print('Retorno de A: ', value)
```

Passando a resposta de uma task pra outra

```python
from rocketry.args import Return

@app.task('every 1 second')
def tarefa_A():
    return 'Sucesso'

@app.task(after_success(tarefa_A))
def tafera_para_sucesso_de_A(value=Return(tarefa_A)):
    print('Retorno de A: ', value)
```

Parametrizando tarefas

```python
from datetime import date
from rocketry import Rocketry
from rocketry.args import Arg

app = Rocketry()

@app.param()
def data_de_hoje():
    return date.today()

@app.task('daily')
def tarefa(dia_corrente=Arg(data_de_hoje)):
    print(dia_corrente)
```

Uso de funções comuns como parâmetros

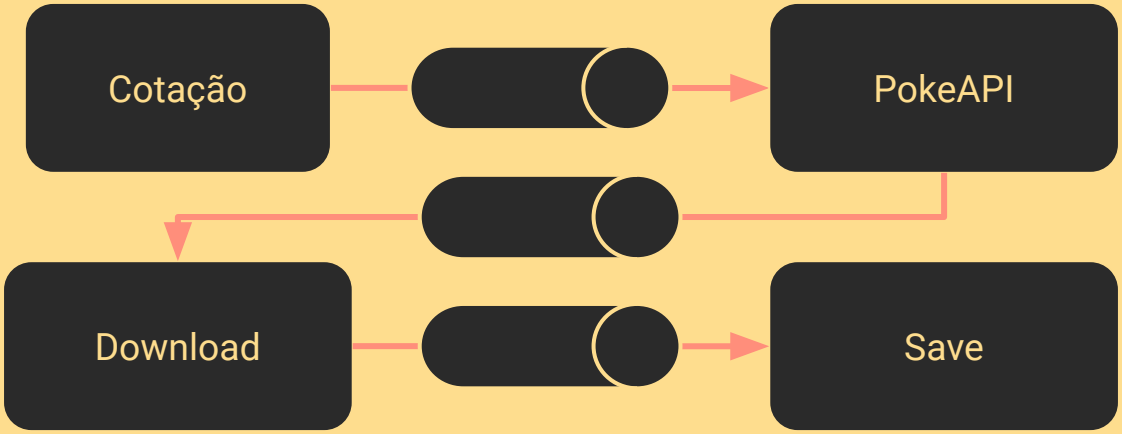
```python
from datetime import date
from rocketry import Rocketry
from rocketry.args import FuncArg

app = Rocketry()

def data_de_hoje():
    return date.today()

@app.task('daily')
def tarefa(dia_corrente=FuncArg(data_de_hoje)):
    print(dia_corrente)
```

Bora criar um pokedolar?

Mão na massa

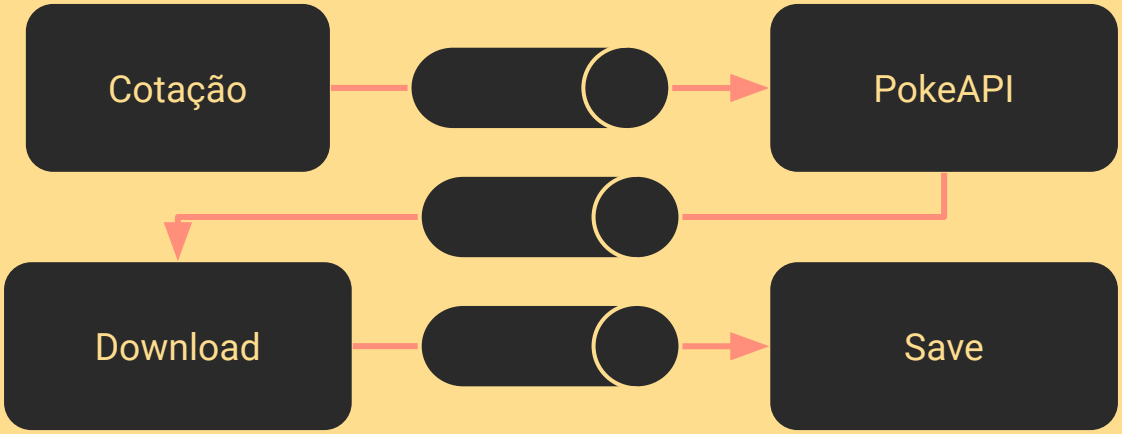
Objetivos do projeto Pokedolar

- Pegar a cotação do dólar do dia (USD -> BRL)
- Juntar as casas decimais (5,50 -> 550)
- Pegar um pokemon na pokeAPI (pokemon/550)
  - Pegar o nome do pokemon
  - Pegar a URL do sprite frontal
- Baixar o sprite frontal e salvar em uma pasta



Links para ajudar

- https://pokeapi.co/api/v2/pokemon
- https://economia.awesomeapi.com.br/json/daily/USD-BRL/?start_date={}&end_date={}



Coisas que não falamos

- Manipulação da sessão
- Tipos diferentes de executores
  - async
  - Processo
  - Main
- Outros tipos de task
  - command
  - imports dinâmicos
  - code

picpay.me/dunossauro

apoia.se/livedepython

pix.dunossauro@gmail.com

Ajude o projeto <3