

Live de
Python

WSGI + Gunicorn + Supervisor + Flask



Obrigado!

apoia.se/livedepython

```
$ python apoiadores.py
```

Alexandre Possebom	Alysson Oliveira	Amaziles Carvalho	Andre Machado
Andre Rodrigues	Bruno Guizi	Carlos Augusto	Cleber Santos
David Reis	Dayham Soares	Diego Ubirajara	Edimar Fardim
Eliabe Silva	Eliakim Moraes	Elias Soares	Emerson Lara
Fabiano Silos	Fabiano Teichmann	Fabiano Gomes	Fernando Furtado
Franklin Silva	Fábio Serrão	Gleison Oliveira	Hemilio Lauro
Humberto Rocha	JOAO COELHO	JONATHAN DOMINGOS	Jean Vetorello
Johnny Tardin	Jonatas Oliveira	Jonatas Simões	João Lugão
Jucélio Silva	Júlia Kastrup	Leon Teixeira	Magno Malkut
Maria Boladona	Matheus Francisco	Nilo Pereira	Pablo Henrique
Paulo Tadei	Pedro Alves	Rafael Galleani	Regis Santos
Renan Moura	Renato Santos	Rennan Almeida	Rodrigo Vaccari
Sérgio Passos	Thiago Araujo	Tiago Cordeiro	Tyrone Damasceno
Vergil Valverde	Vicente Marcal	Wander Silva	Wellington Carlos
Wellington Camargo	Welton Souza	William Oliveira	Willian Gl
Yros Aguiar	Falta você	Falta você	Falta você

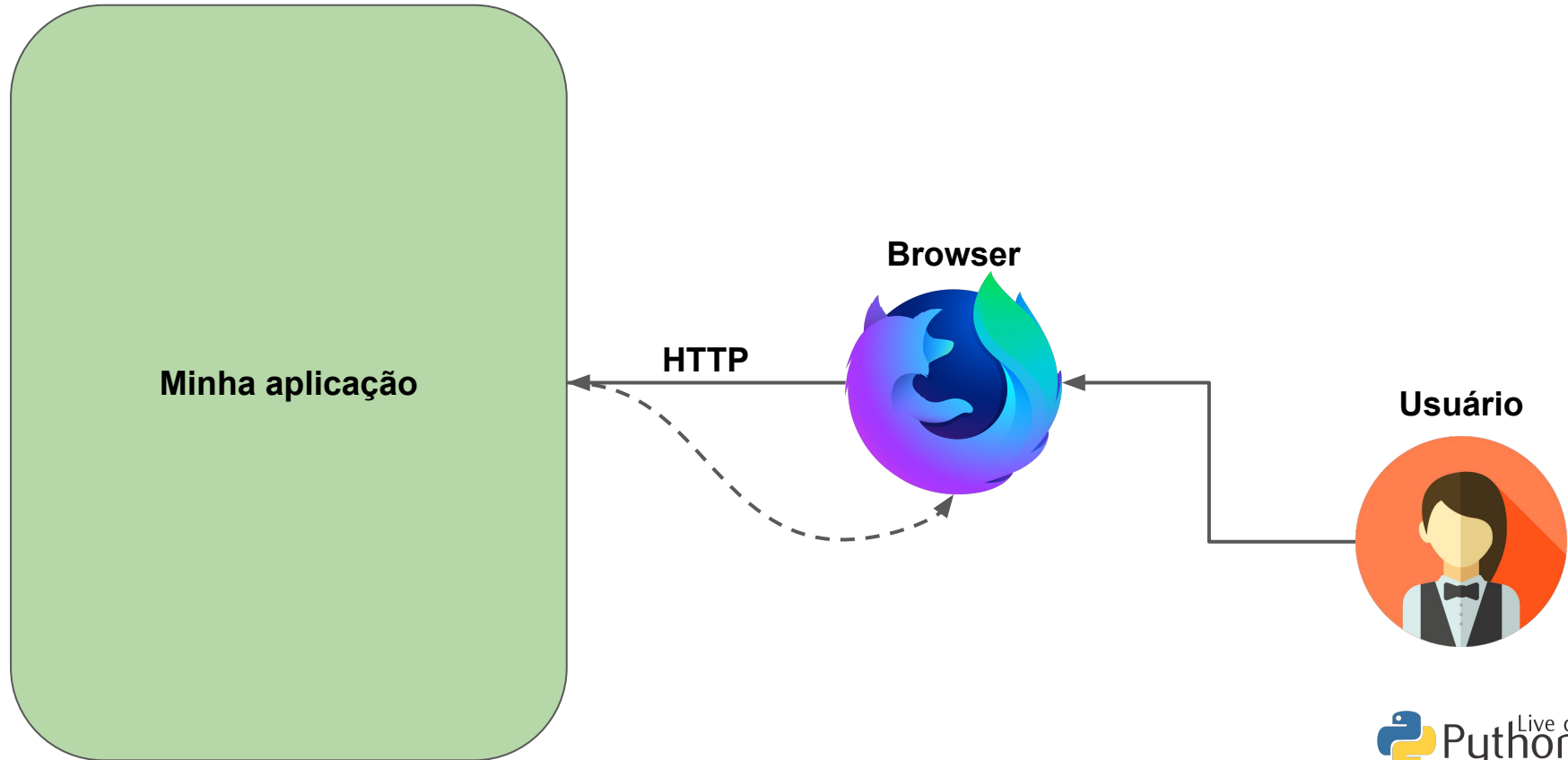
Roteiro

- Senta que lá vem história...
- PEP 333 (3)
- Nosso app
- Gunicorn

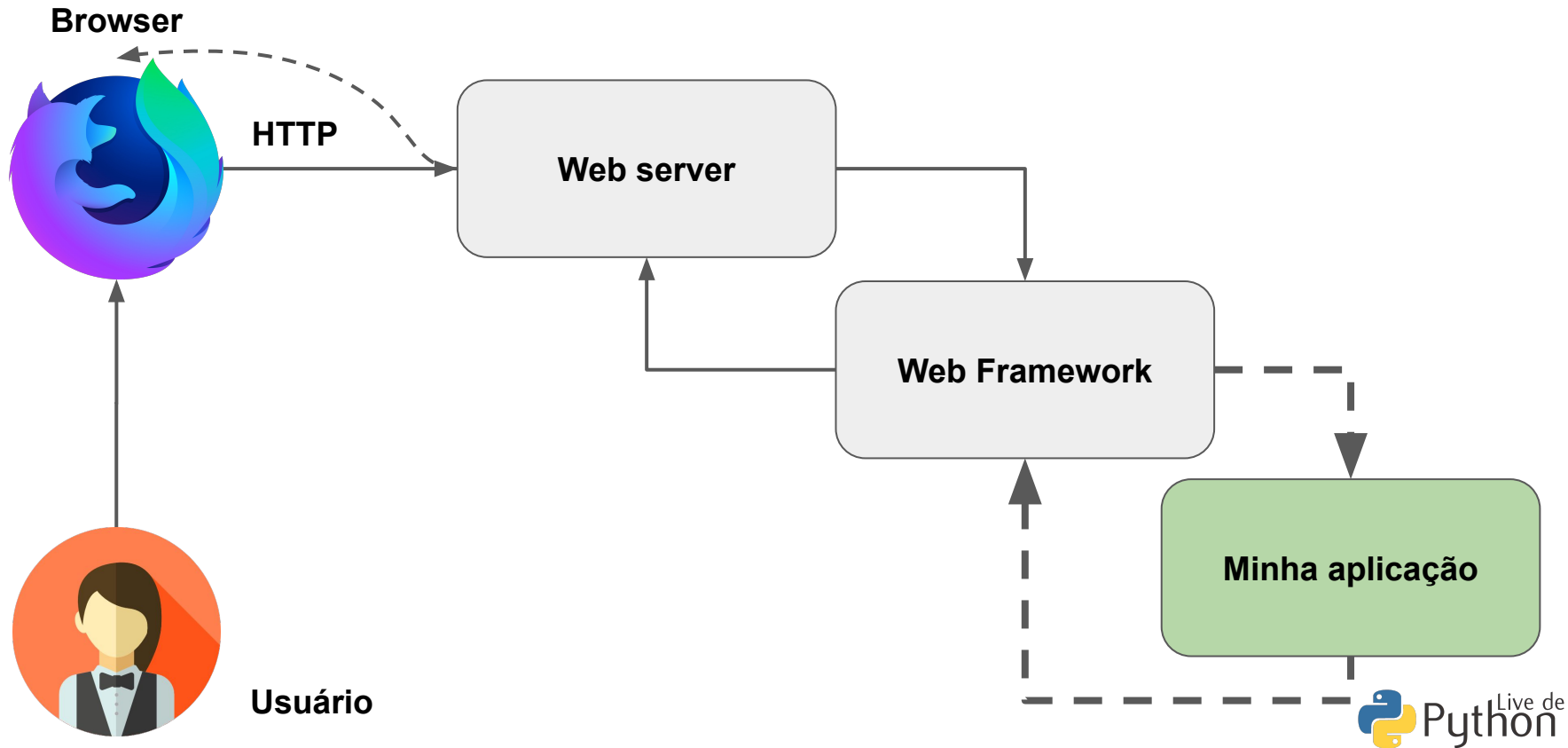
Vão ficar pra outro dia...

- Supervisor
- Proxy reverso
- Docker
- Testes de carga

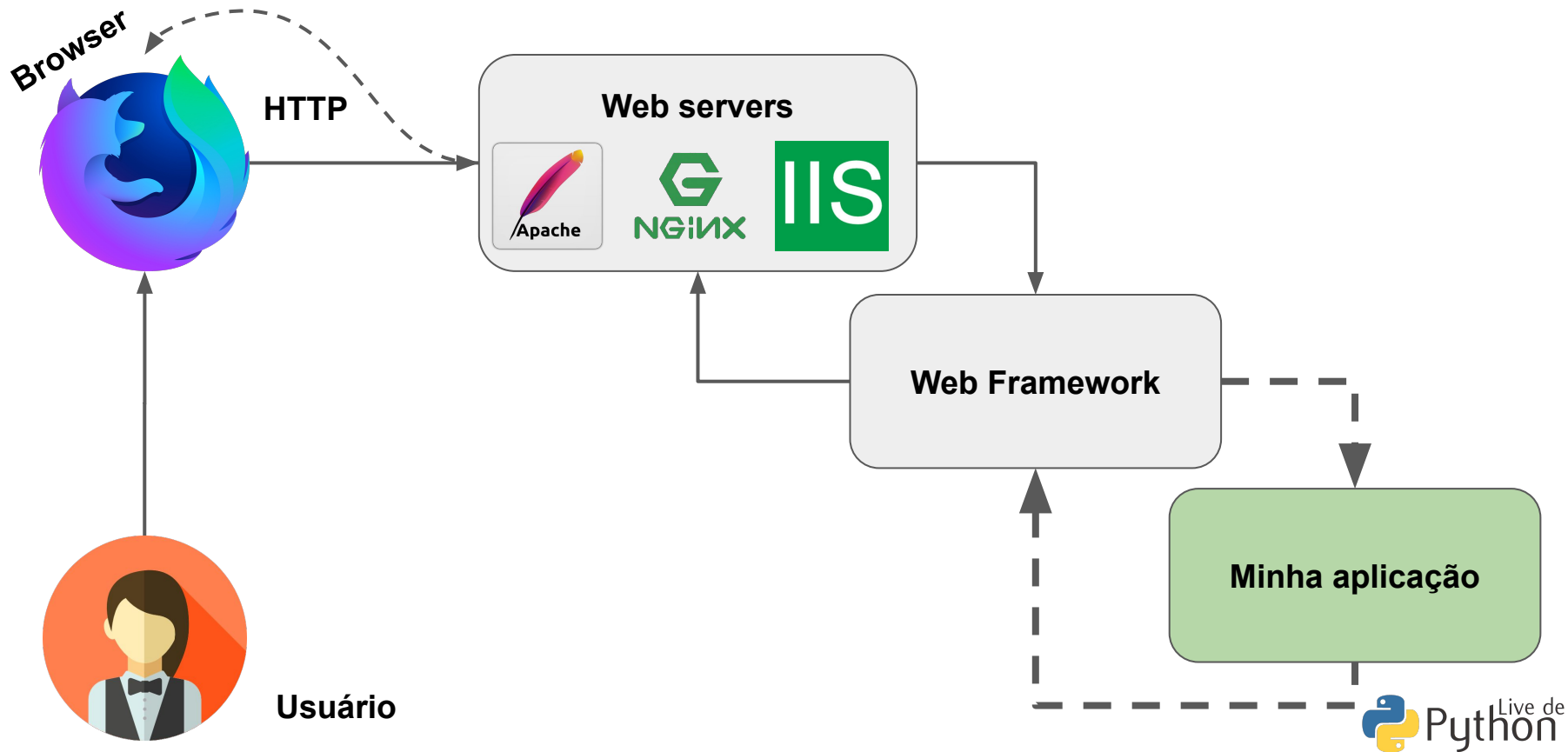
Desmembrando uma aplicação web



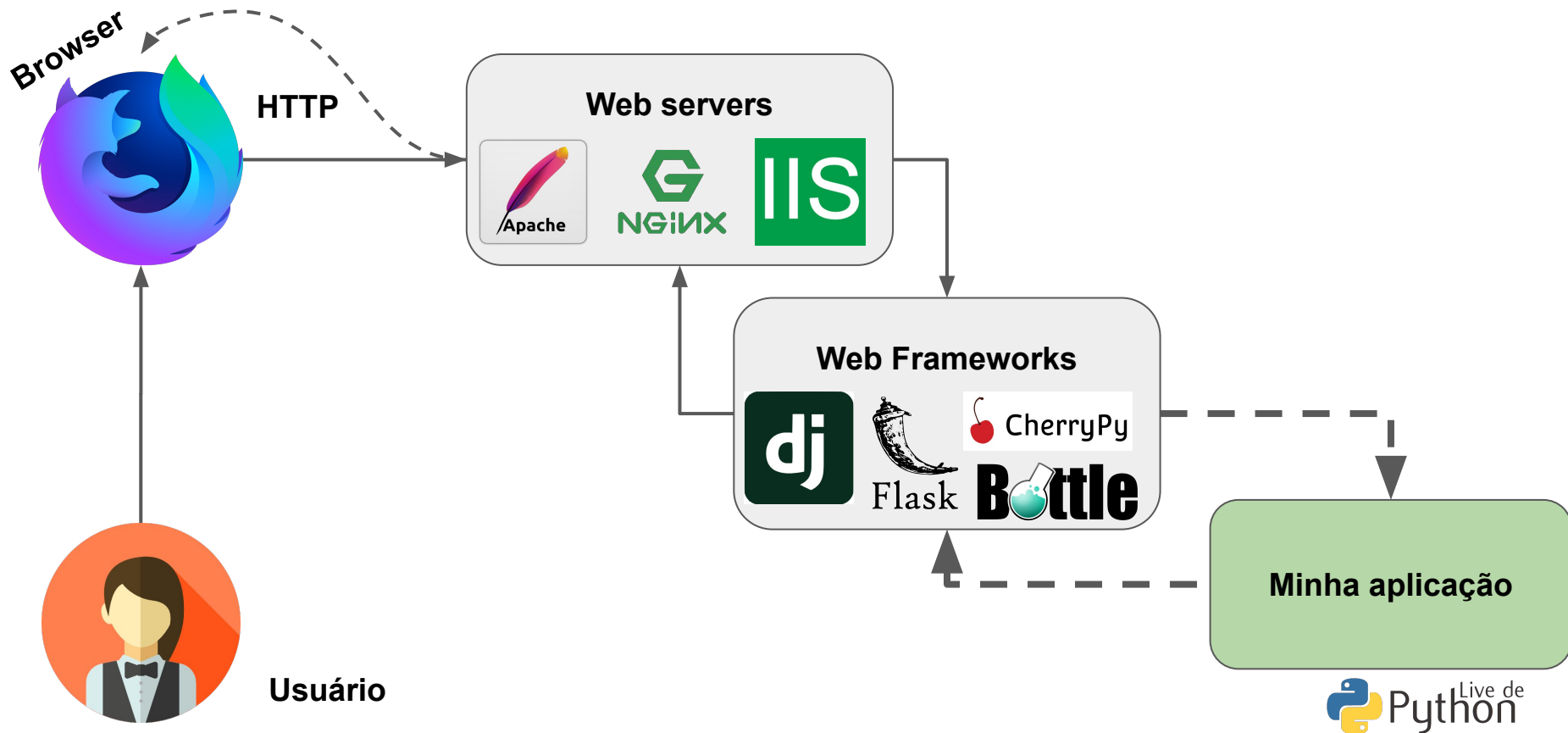
Desmembrando uma aplicação web



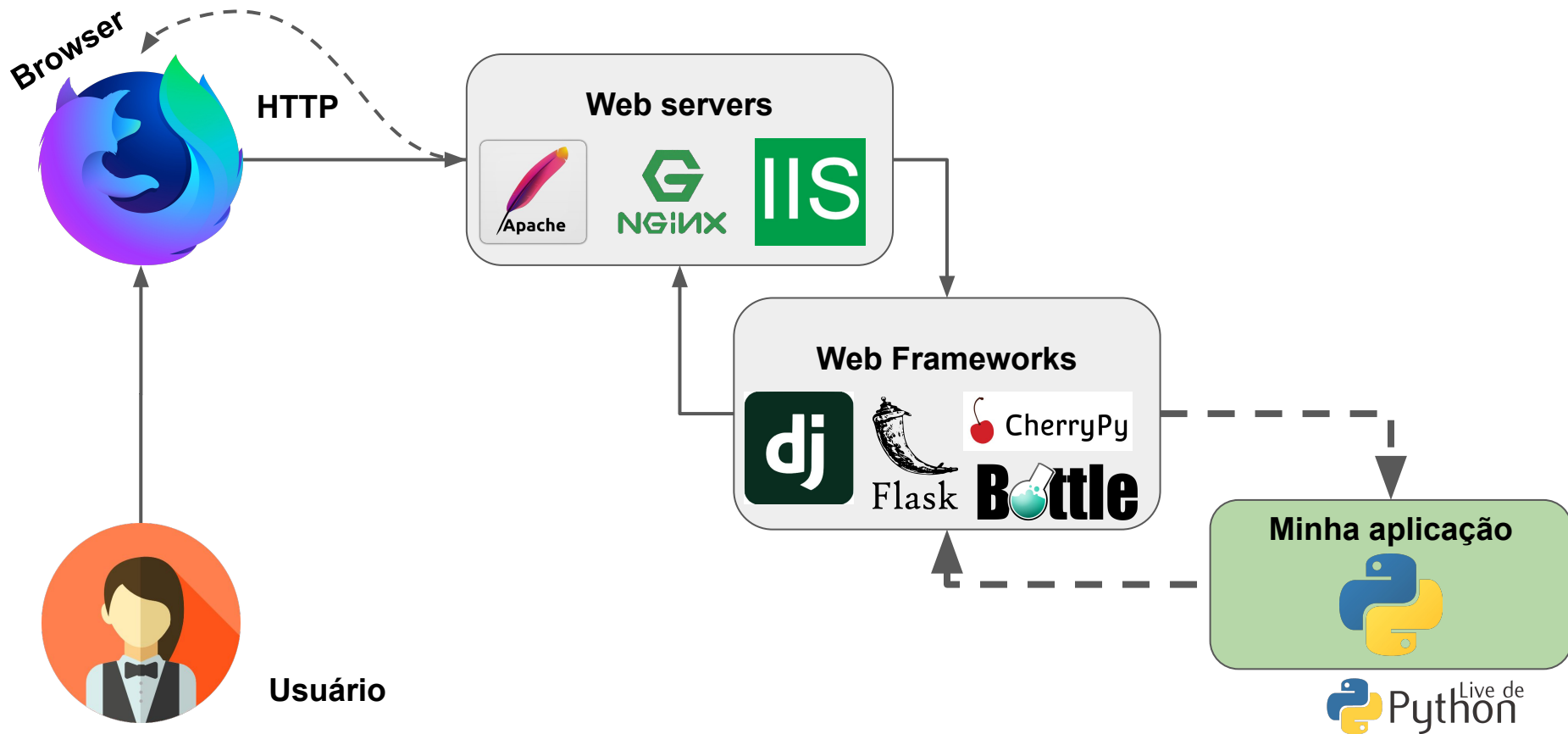
Desmembrando uma aplicação web



Desmembrando uma aplicação web



Desmembrando uma aplicação web



Mas será que todos os frameworks sabem trocar informações com todos os servidores?

NÃO

Desmembrando uma aplicação web



WSGI

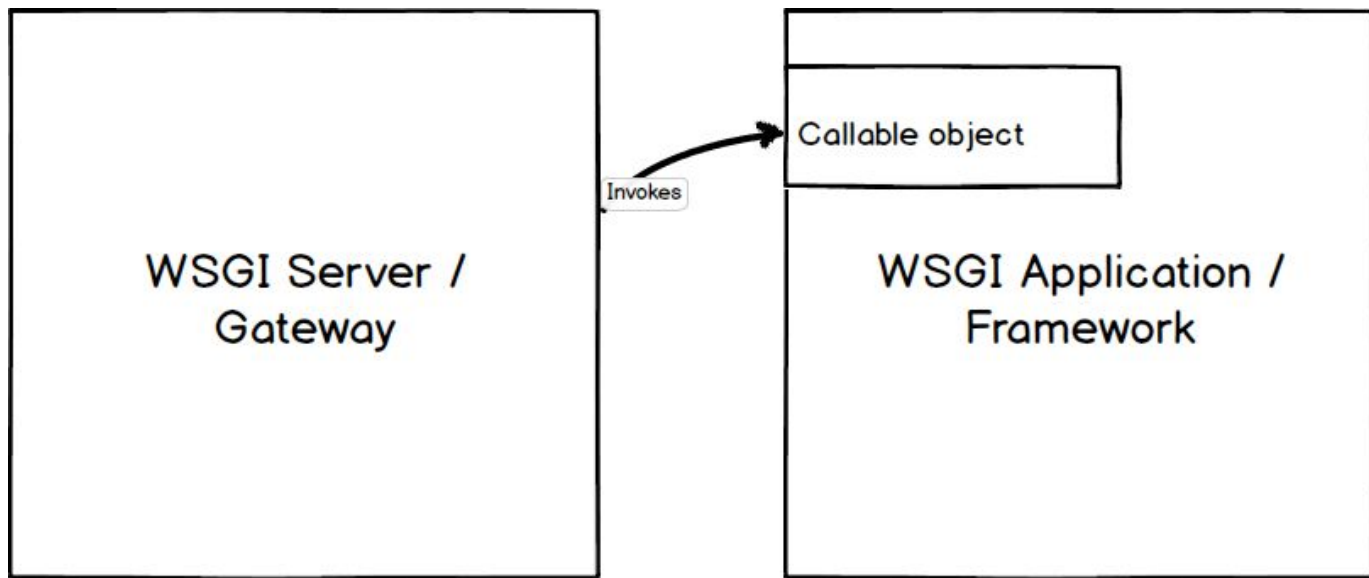
Web Server Gateway Interface, “uisgui”

(...) Uma interface padrão proposta entre servidores Web e aplicativos da Web ou estruturas Python, para promover a portabilidade de aplicativos da Web em uma variedade de servidores da Web. [\[PEP 333\]](#)

WSGI [2] - Porquê?

- Flexibilidade
 - O framework não precisa aprender a conversar com todos os webserver
 - O desenvolvedor pode escolher a ferramenta, WSGI, que lhe atende melhor
- Segregação de responsabilidades
 - O framework não é responsável por escalar
 - Dimensionar a quantidade de requests fica a cargo do WSGI

PEP 333 (3)



os desenvolvedores de framework só precisam fornecer um invocável

PEP 333 (3) - framework

```
def simple_app(environ, start_response):  
    status = '200 OK'  
    response_headers = [('Content-type', 'text/plain')]  
    start_response(status, response_headers)  
    return [b'Hello world!\n']
```

WSGIs

- Gunicorn
 - Baseado em Pré-forks
 - Simples de implementar
- uWSGI
 - Escrito em C, é mais rápido (pode ter problemas de integração)
 - Totalmente configurável (mais complexo)
- Waitress
 - Fork do gunicorn pra rodar no windows
- mod_WSGI
 - Mod iniciado na década de 90 e foi evoluindo com o padrão WSGI

Flask + Gunicorn



Flask APP

Uma aplicação bem simples.

```
__init__.py
1  • from flask import Flask
2
3  app = Flask(__name__)
4
5
6  • def create_app():
7      from .hello import bp_hello
8      app.register_blueprint(bp_hello)
9      return app
```

```
hello.py
1  • from flask import Blueprint, current_app
2
3  • from random import randint
4
5  bp_hello = Blueprint('Hello', __name__)
6
7
8  @bp_hello.route('/ping')
9  • def ping():
10     return "pong"
```

Vai, vamos escrever
essa aplicação

Flask + Gunicorn

```
dunossauro at bouman in ~/Live95
```

```
$ gunicorn "app:create_app()"
```

```
[2019-07-01 12:48:23 -0300] [5541] [INFO] Starting gunicorn 19.9.0
```

```
[2019-07-01 12:48:23 -0300] [5541] [INFO] Listening at: http://127.0.0.1:8000 (5541)
```

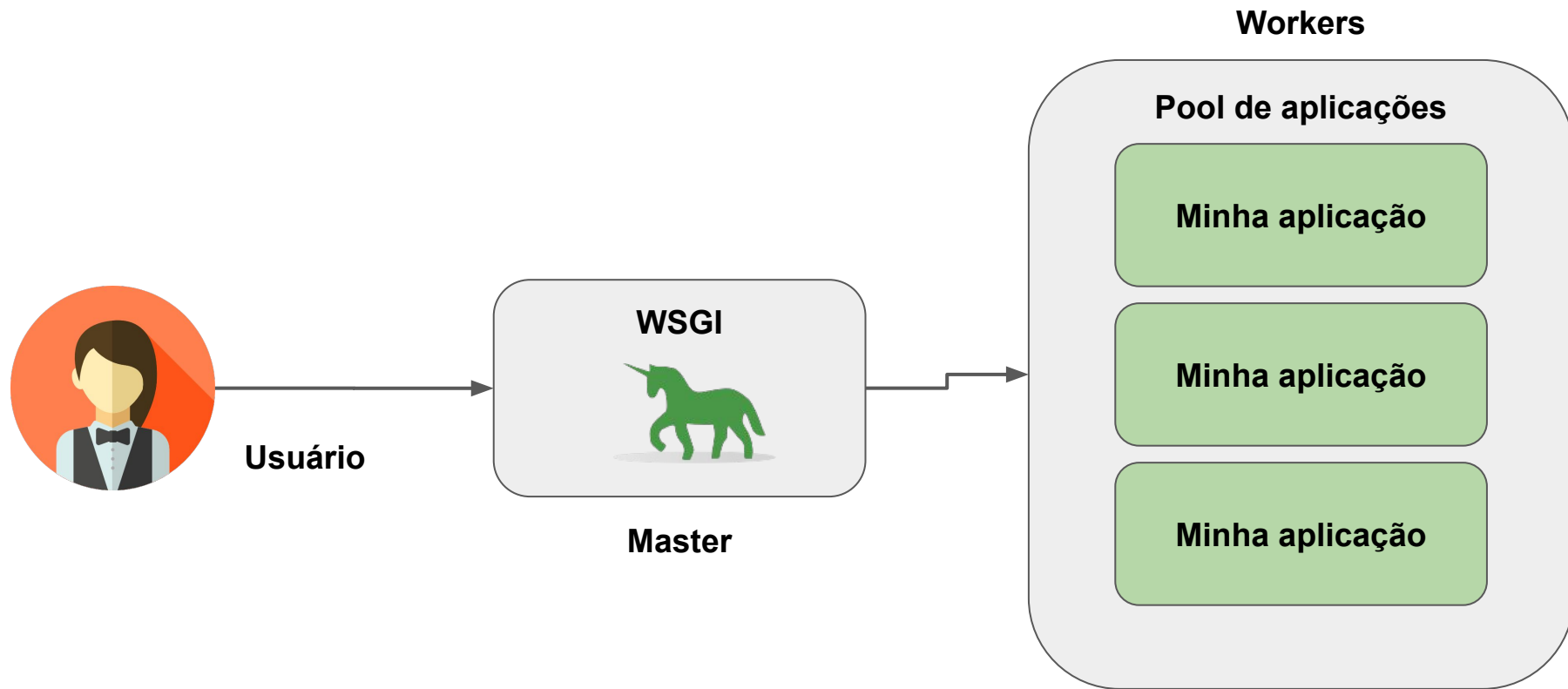
```
[2019-07-01 12:48:23 -0300] [5541] [INFO] Using worker: sync
```

```
[2019-07-01 12:48:23 -0300] [5544] [INFO] Booting worker with pid: 5544
```

A integração com Gunicorn se dá de maneira simples, praticamente transparente.

Agora vamos entender
o que vai rolar

Pre-Fork model



Master node

Master

The master process is a simple loop that listens for various process signals and reacts accordingly. It manages the list of running workers by listening for signals like TTIN, TTOU, and CHLD. TTIN and TTOU tell the master to increase or decrease the number of running workers. CHLD indicates that a child process has terminated, in this case the master process automatically restarts the failed worker.

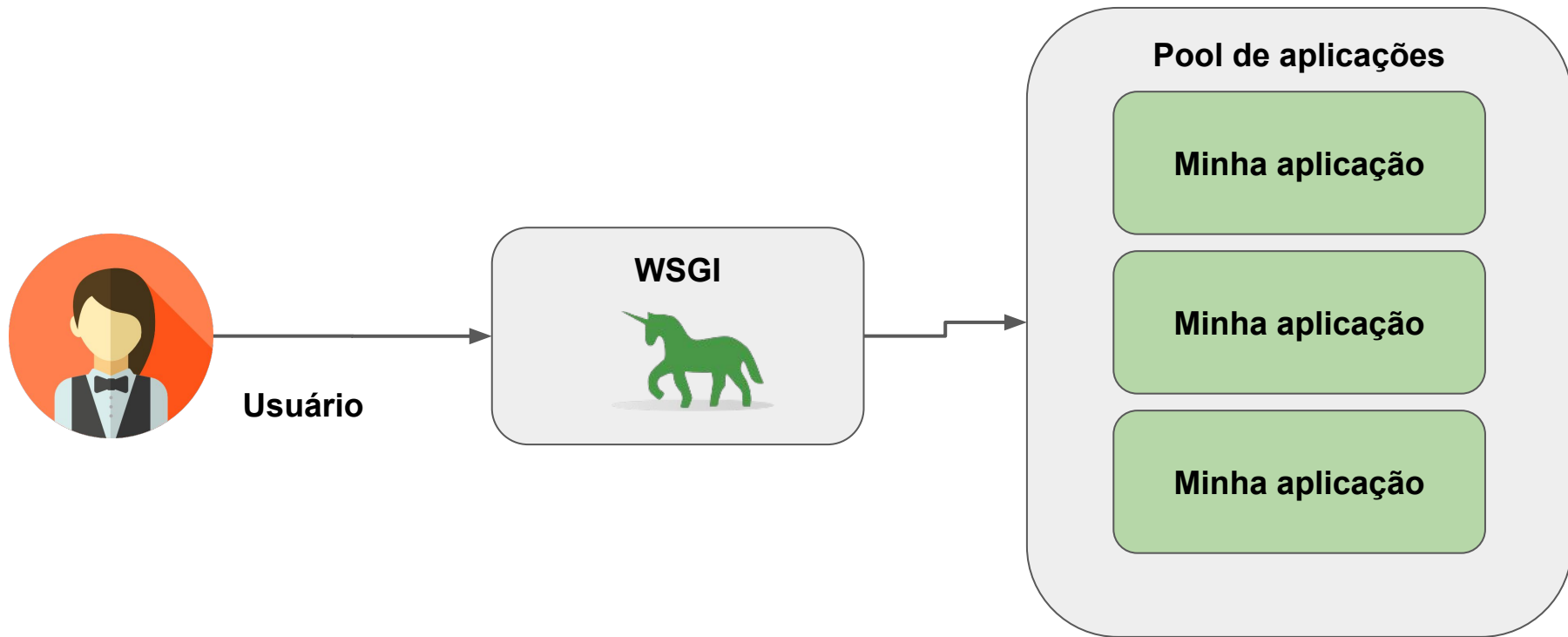
Master node

Sim, aqui temos várias siglas complicadas que ninguém nunca explica.

Sinais:

- **TTIN**
 - Quando um processo em segundo plano tenta ler algo do tty
- **TTOUT**
 - Quando um processo em segundo plano precisa escrever algo do tty
- **CHLD**
 - Um aviso de quando um processo filho morre. Nesse caso o unicorn faz o restart do processo

Flask + Gunicorn com Workers



Flask + Gunicorn com Workers

```
dunossauro at bouman in ~/Live101
```

```
$ gunicorn "app:create_app()" -w 5
```

```
[2019-08-19 19:18:43 -0300] [3304] [INFO] Starting gunicorn 19.9.0
```

```
[2019-08-19 19:18:43 -0300] [3304] [INFO] Listening at: http://127.0.0.1:8000 (3304)
```

```
[2019-08-19 19:18:43 -0300] [3304] [INFO] Using worker: sync
```

```
[2019-08-19 19:18:43 -0300] [3307] [INFO] Booting worker with pid: 3307
```

```
[2019-08-19 19:18:43 -0300] [3308] [INFO] Booting worker with pid: 3308
```

```
[2019-08-19 19:18:43 -0300] [3309] [INFO] Booting worker with pid: 3309
```

```
[2019-08-19 19:18:43 -0300] [3310] [INFO] Booting worker with pid: 3310
```

```
[2019-08-19 19:18:43 -0300] [3311] [INFO] Booting worker with pid: 3311
```

Flask + Gunicorn com Workers

```
dunossauro at bouman in ~/Live101
```

```
$ gunicorn "app:create_app()" -w 5
```

```
[2019-08-19 19:18:43 -0300] [3304] [INFO] Starting gunicorn 19.9.0
```

```
[2019-08-19 19:18:43 -0300] [3304] [INFO] Listening at: http://127.0.0.1:8000 (3304)
```

```
[2019-08-19 19:18:43 -0300] [3304] [INFO] Using worker: sync
```

```
[2019-08-19 19:18:43 -0300] [3307] [INFO] Booting worker with pid: 3307
```

```
[2019-08-19 19:18:43 -0300] [3308] [INFO] Booting worker with pid: 3308
```

```
[2019-08-19 19:18:43 -0300] [3309] [INFO] Booting worker with pid: 3309
```

```
[2019-08-19 19:18:43 -0300] [3310] [INFO] Booting worker with pid: 3310
```

```
[2019-08-19 19:18:43 -0300] [3311] [INFO] Booting worker with pid: 3311
```

Flask + Gunicorn com Workers

```
dunossauro at bouman in ~/Live101
```

```
$ gunicorn "app:create_app()" -w 5
```

```
[2019-08-19 19:18:43 -0300] [3304] [INFO] Starting gunicorn 19.9.0
[2019-08-19 19:18:43 -0300] [3304] [INFO] Listening at: http://127.0.0.1:8000 (3304)
[2019-08-19 19:18:43 -0300] [3304] [INFO] Using worker: sync
[2019-08-19 19:18:43 -0300] [3307] [INFO] Booting worker with pid: 3307
[2019-08-19 19:18:43 -0300] [3308] [INFO] Booting worker with pid: 3308
[2019-08-19 19:18:43 -0300] [3309] [INFO] Booting worker with pid: 3309
[2019-08-19 19:18:43 -0300] [3310] [INFO] Booting worker with pid: 3310
[2019-08-19 19:18:43 -0300] [3311] [INFO] Booting worker with pid: 3311
```

```
dunossauro at bouman in ~/L
```

```
$ ps a | grep gunicorn | cu
```

```
3304 pts/2 S+ 0:00 /home/dunossauro/.cache/pypoetry/virtualenvs/live101-py3.7/bin/gunicorn app:create_app() -w 5
3307 pts/2 S+ 0:00 /home/dunossauro/.cache/pypoetry/virtualenvs/live101-py3.7/bin/gunicorn app:create_app() -w 5
3308 pts/2 S+ 0:00 /home/dunossauro/.cache/pypoetry/virtualenvs/live101-py3.7/bin/gunicorn app:create_app() -w 5
3309 pts/2 S+ 0:00 /home/dunossauro/.cache/pypoetry/virtualenvs/live101-py3.7/bin/gunicorn app:create_app() -w 5
3310 pts/2 S+ 0:00 /home/dunossauro/.cache/pypoetry/virtualenvs/live101-py3.7/bin/gunicorn app:create_app() -w 5
3311 pts/2 S+ 0:00 /home/dunossauro/.cache/pypoetry/virtualenvs/live101-py3.7/bin/gunicorn app:create_app() -w 5
3900 pts/0 S+ 0:00
```

Vish, agora vai ficar
doido

Flask + Gunicorn com Threads

```
dunossauro at bouman in ~/Live101
```

```
$ gunicorn "app:create_app()" -t 10
```

```
[2019-08-19 19:35:29 -0300] [4852] [INFO] Starting gunicorn 19.9.0
```

```
[2019-08-19 19:35:29 -0300] [4852] [INFO] Listening at: http://127.0.0.1:8000 (4852)
```

```
[2019-08-19 19:35:29 -0300] [4852] [INFO] Using worker: sync
```

```
[2019-08-19 19:35:29 -0300] [4855] [INFO] Booting worker with pid: 4855
```

```
[2019-08-19 19:35:36 -0300] [4852] [INFO] Handling signal: winch
```

```
[2019-08-19 19:35:36 -0300] [4852] [INFO] Handling signal: winch
```

```
[2019-08-19 19:35:37 -0300] [4852] [INFO] Handling signal: winch
```


Flask + Gunicorn + Async (gevent)

```
dunossauro at bouman in ~/Live101
```

```
$ gunicorn "app:create_app()" -t 10 -k gevent
```

```
[2019-08-19 19:36:40 -0300] [4890] [INFO] Starting gunicorn 19.9.0
```

```
[2019-08-19 19:36:40 -0300] [4890] [INFO] Listening at: http://127.0.0.1:8000 (4890)
```

```
[2019-08-19 19:36:40 -0300] [4890] [INFO] Using worker: gevent
```

```
[2019-08-19 19:36:40 -0300] [4893] [INFO] Booting worker with pid: 4893
```