# APIs Pythonicas con GraphQL

# whoami



Python Dev



API Design



Ambro17



Nahuel Ambrosini

## AGENDA

OI APIS PYTHONICAS

Qué es una API Pythonica

**O2 GRAPHQL BASICS** Qué es GraphQL

O3 GRAPHQL API DEMO

Cómo luce una API Pythonica con GraphQL



Diseñar una API es el arte de hacer simple lo complejo

All programmers are API Designers

Joshua Bloch

## Tipos de APIs







Librerías



## Cómo es una API Pythonica



Fácil de Usar



Documentada



Difícil de Usar Mal



Errores Informativos



**Explorable** 



Integrable

```
import requests
resp = requests.get('https://www.google.com', auth=('User', 'Password'))
```



Fácil de Usar

```
import urllib
from base64 import b64encode

request = urllib.request.Request('https://www.google.com', headers={
    "Authorization": "Basic %s" % b64encode(b'User:Password')
})
resp = urllib.request.urlopen(request)
```

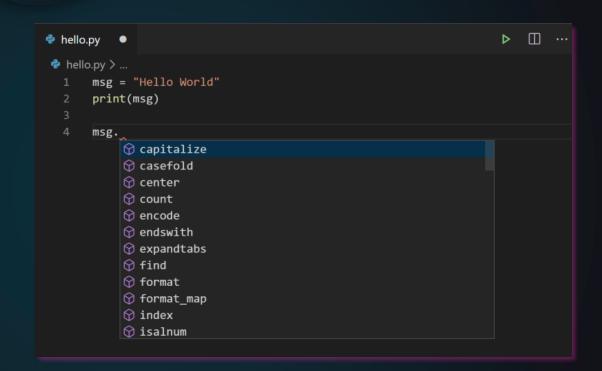


Difícil de Usar Mal 30 \* \* \* \* ~/my\_script.py

import schedule

schedule.every(30).minutes.do(my\_script)







#### Documentación de Python - 3.9.0

¡Te damos la bienvenida! Esta es la documentación de Python 3.9.0.

#### Áreas de la documentación:

¿Qué hay de nuevo en Python 3.9? o todos los "Qué hay de nuevo" desde 2.0

#### **Tutorial**

empieza aquí

#### Referencia de la Biblioteca

Mantén esto bajo tu almohada

#### Referencia del lenguaje

descripción de sintaxis y los elementos del lenguaje

#### Configuración y uso de Python

como usar Python en diferentes plataformas

#### Cómos (HOWTOs) de Python

documentos detallados sobre temas específicos

#### Instalación de módulos de Python

instalación desde el Índice de paquete de Python (Python Package Index) y otras fuentes

#### Distribuir módulos de Python

publicación de módulos para que otros los instalen

#### Extender e incrustar

tutorial para programadores y programadoras de C/C++

#### Python/C API

referencia para programadores y programadoras de C/C++

#### Preguntas frecuentes

preguntas frecuentes (¡con respuestas!)

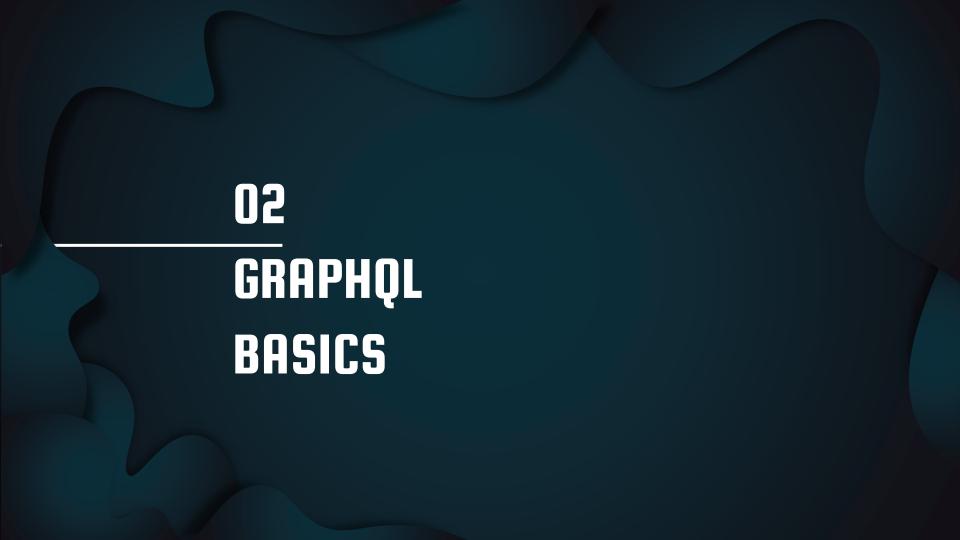


Errores Informativos

```
Dec 13, 2018 9:24:15 PM com.sun.jersev.server.impl.application.WebApplicationImpl initiate
                                                                                         Java 11 and JAXB Error
INFO: Initiating Jersey application, version 'Jersey: 1.19.4 05/24/2017 03:20 PM'
Dec 13, 2018 9:24:15 PM org.apache.catalina.core.ApplicationContext log
SEVERE: StandardWrapper.Throwable
java.lang.TypeNotPresentException: Type javax.xml.bind.JAXBContext not present
       at java.base/sun.reflect.generics.factory.CoreReflectionFactory.makeNamedType(CoreReflectionFactory.java:117)
       at java.base/sun.reflect.generics.visitor.Reifier.visitClassTypeSignature(Reifier.java:125)
       at java.base/sun.reflect.generics.tree.ClassTypeSignature.accept(ClassTypeSignature.java:49)
       at java.base/sun.reflect.generics.visitor.Reifier.reifyTypeArguments(Reifier.java:68)
       at java.base/sun.reflect.generics.visitor.Reifier.visitClassTypeSignature(Reifier.java:138)
       at java.base/sun.reflect.generics.tree.ClassTypeSignature.accept(ClassTypeSignature.java:49)
        >>> 2 + '2'
           File "<stdin>", line 1, in <module>
            File "<stdin>", line 2, in hello
        TypeError: unsupported operand type(s) for +: 'int' and 'str'
       at org.apache.catalina.core.StandardWrapper.loadServlet(StandardWrapper.java:1079)
       at org.apache.catalina.core.StandardWrapper.load(StandardWrapper.java:971)
       at org.apache.catalina.core.StandardContext.loadOnStartup(StandardContext.java:4829)
       at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5143)
       at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:183)
       at org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.java:1432)
       at org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.java:1422)
       at iava.base/iava.util.concurrent.FutureTask.run(FutureTask.iava:264)
       at org.apache.tomcat.util.threads.InlineExecutorService.execute(InlineExecutorService.java:75)
       at java.base/java.util.concurrent.AbstractExecutorService.submit(AbstractExecutorService.java:140)
                                                                                                                    crunchify.com
       at org.apache.catalina.core.ContainerBase.startInternal(ContainerBase.java:944)
```



```
$ history | awk '{print $2}' | sort | uniq -c | sort -nr | head -5
45 sudo
22 ls
14 cd
10 history
7 git
```



# GraphQL

Es un nuevo paradigma de APIs creado por Facebook Define un Lenguaje de Consultas Declarativo La API es un Grafo de Datos Tipado Los recursos son generados a partir de las consultas al Grafo

```
type Query {
  talks: [Talk!]!
type Talk {
  title: String!
 description: String!
  speaker: Speaker!
type Speaker {
  name: String!
  email: String!
```

## **Contrato Legible**

Nuestra API es legible por humanos

```
talks {
  title
  speaker {
    name
```

## **Consultas Declarativas**

Pido sólo lo que necesito

```
{
  "talks": {
    "title": "PyconAr GraphQL",
    "speaker": {
        "name": "John Doe"
    }
}
```

## **Respuestas Predecibles**

La respuesta tiene la forma de mi consulta

## Empresas usando GraphQL

















.. y muchas más!





## PyconAr API



¿Qué charlas puedo ver esta PyconAr?

¿Quienes son los sponsors de esta PyConAr? ¿Están buscando gente?

¿Qué personas están abiertas a ofertas?

¿Hay gente interesada en GraphQL?

## **GraphQL APIs**



Fácil de Usar



Documentada



Difícil de Usar Mal



Errores Informativos



**Explorable** 



Integrable

### **Conclusiones**

Las APIs Pythonicas son fáciles de usar y difíciles de usar mal

GraphQL nos puede ayudar a diseñar APIs Pythonicas

Aunque las APIs Pythonicas también pueden ser REST

GraphQL se puede combinar con REST, Microservicios o un Monolito

