

# Creando APIs con Python

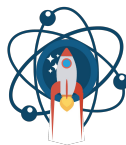
Rodolfo Ferro Pérez

Co-Founder & CTO @ Future Lab  
Dpto. de Matemáticas, UG - CIMAT  
GitHub Campus Expert





UNIVERSIDAD DE  
GUANAJUATO



**FUTURE  
LAB**

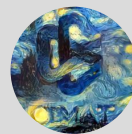


**EMBA  
JADO  
RES**  
León Joven



Clubes de Ciencia  
México

@FerroRodolfo



Áreas  
de interés



# Contenido:

1. Servicios web: APIs
2. Pre-requisitos e instalación
3. Requests: Consumiendo APIs con Python
4. Resolviendo el problema de Iris con scikit-learn
5. Flask: Microframework web de Python
6. IA como servicio: API vs Web App

# Servicios web

API = Application Programming Interface

Una API es un **conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas**: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software.

Fuente: “[¿Qué es una API?](#)”, TICbeat, 2014.

# Servicios web

HTTP Method	URI	Action
GET	http://[hostname]/todo/api/v1.0/tasks	Retrieve list of tasks
GET	http://[hostname]/todo/api/v1.0/tasks/[task_id]	Retrieve a task
POST	http://[hostname]/todo/api/v1.0/tasks	Create a new task
PUT	http://[hostname]/todo/api/v1.0/tasks/[task_id]	Update an existing task
DELETE	http://[hostname]/todo/api/v1.0/tasks/[task_id]	Delete a task

Fuente: <https://blog.miguelgrinberg.com/post/designing-a-restful-api-with-python-and-flask>

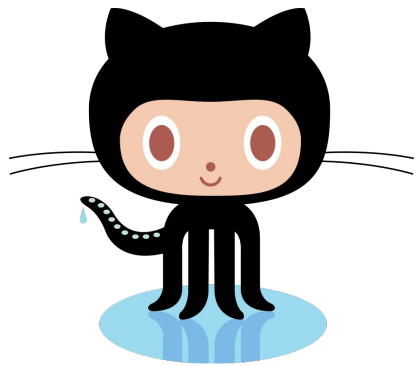
# Pre-requisitos e instalación

- Cuenta de GitHub (*recomendable*)
- Atom (*recomendable*)
- PIP - *Pip Installs Packages*
- Python 3.6:
  - scikit-learn
  - Unidecode
  - Requests
  - graphviz
  - NumPy
  - Flask

Pueden seguir las instrucciones del archivo README en el repositorio del taller para la instalación.

# Pre-requisitos e instalación

- Cuenta de **GitHub** (*recomendable*)
- Atom (*recomendable*)
- PIP - *Pip Installs Packages*
- Python 3.6:
  - scikit-learn
  - Unicode
  - Requests
  - graphviz
  - NumPy
  - Flask

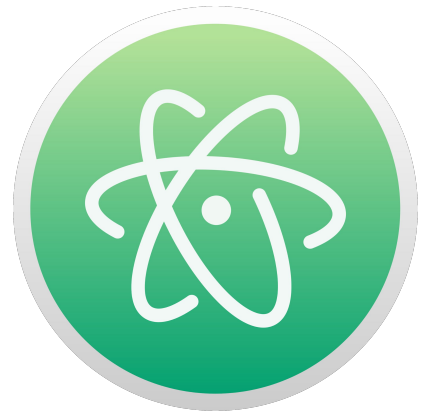


<http://github.com/>

Pueden seguir las instrucciones del archivo README en el repositorio del taller para la instalación.

# Pre-requisitos e instalación

- Cuenta de GitHub (*recomendable*)
- **Atom** (*recomendable*)
- PIP - *Pip Installs Packages*
- Python 3.6:
  - scikit-learn
  - Unidecode
  - Requests
  - graphviz
  - NumPy
  - Flask



<https://atom.io/>

Pueden seguir las instrucciones del archivo README en el repositorio del taller para la instalación.



# Pre-requisitos e instalación

- Cuenta de GitHub (*recomendable*)
- Atom (*recomendable*)
- **PIP** - *Pip Installs Packages*
- Python 3.6:
  - scikit-learn
  - Unicodecode
  - Requests
  - graphviz
  - NumPy
  - Flask



<https://pip.pypa.io/en/stable/installing/>

Pueden seguir las instrucciones del archivo README en el repositorio del taller para la instalación.

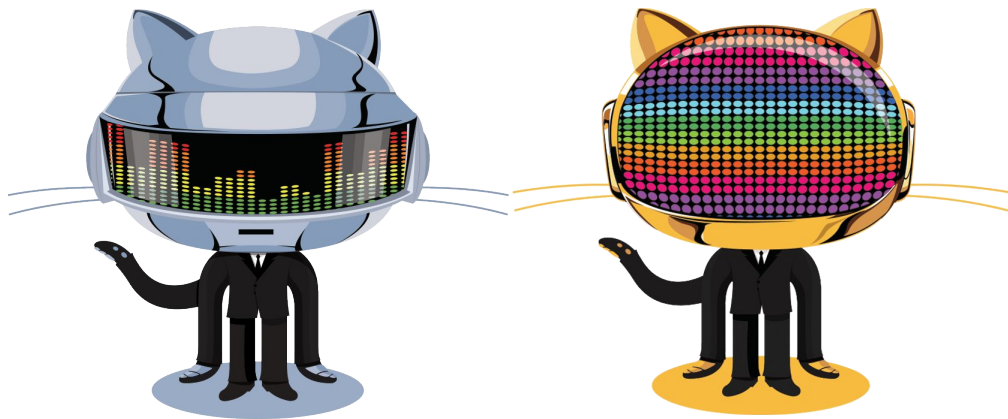
# Pre-requisitos e instalación

- Cuenta de GitHub (*recomendable*)
- Atom (*recomendable*)
- PIP - *Pip Installs Packages*
- **Python 3.6:**
  - scikit-learn
  - Unidecode
  - Requests
  - graphviz
  - NumPy
  - Flask



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

Pueden seguir las instrucciones del archivo README en el repositorio del taller para la instalación.



<https://github.com/RodolfoFerro/PythonBaseAPI>

Fork it, clone it, cut it, paste it, load it, check it, quick, rewrite it...

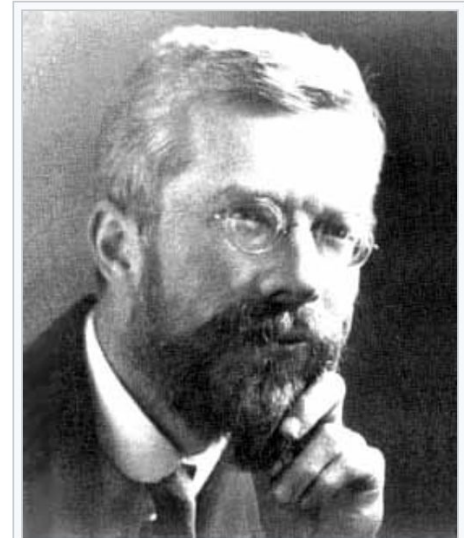
*– Daft Punk (ft. Rodolfo Ferro)*

# Problema de Iris

## Iris flor conjunto de datos

El **Iris flor conjunto de datos** o **Fisher's Iris conjunto de datos** es un **conjunto de datos multivariante** introducido por **Ronald Fisher** en su papel de 1936 *The use of multiple measurements in taxonomic problems* (*El uso de medidas múltiples en problemas taxonómicos*) como un ejemplo de **análisis discriminante lineal**.<sup>1</sup> A veces, se llama **Anderson's Iris conjunto de datos** porque **Edgar Anderson** coleccionó la data usada para cuantificar la variación **morfológica** del **Iris** con las flores de tres especies relacionadas.<sup>2</sup> Dos de las tres **especies** se coleccionaron en la **Península de la Gaspesia** «todos son de la misma pastura, y recolectado el mismo día y medidos al mismo tiempo por la misma persona con el mismo aparato».<sup>3</sup>

El conjunto de datos contiene 50 muestras de cada una de tres especies de **Iris** (**Iris setosa**, **Iris virginica** e **Iris versicolor**). Se midió cuatro rasgos de cada muestra: lo largo y lo ancho del **sépalos** y **pétalos**, en centímetros. Basado en la combinación de estos cuatro rasgos, Fisher se desarrolló un modelo discriminante lineal para distinguir entre una especie y otra.

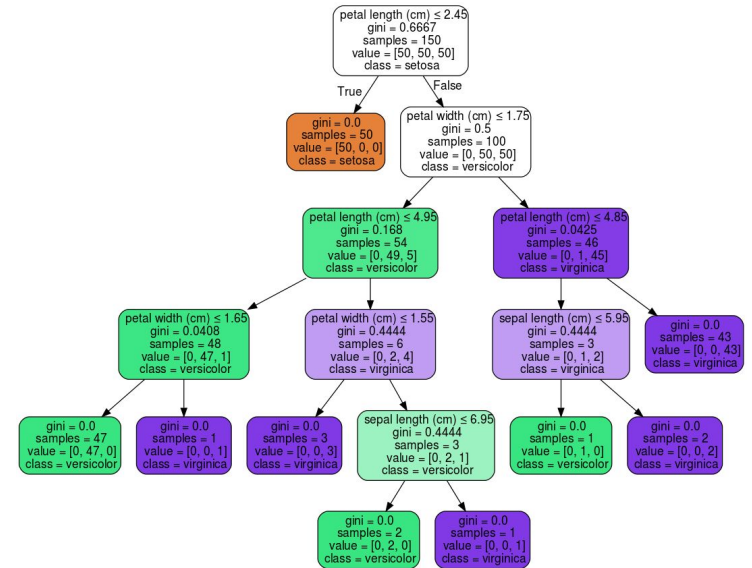
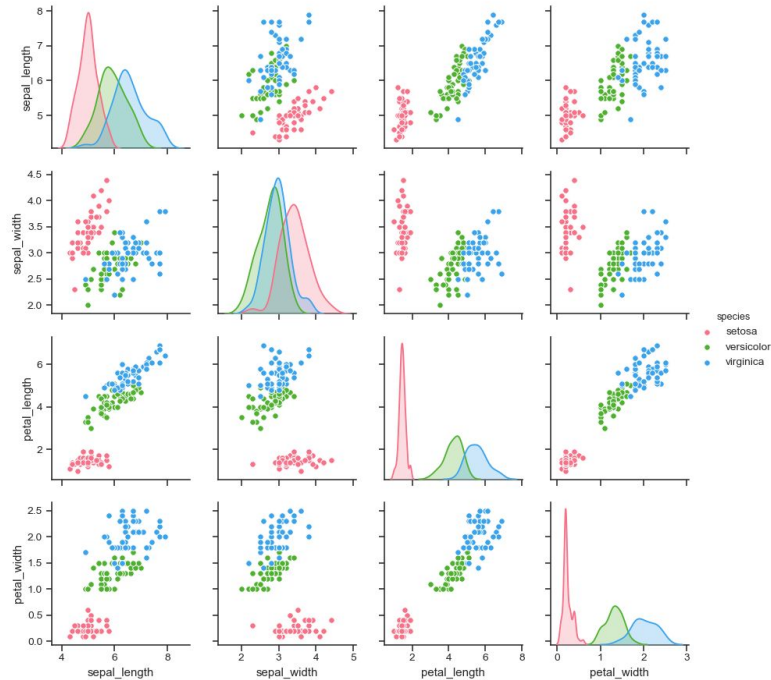


Ronald Fisher

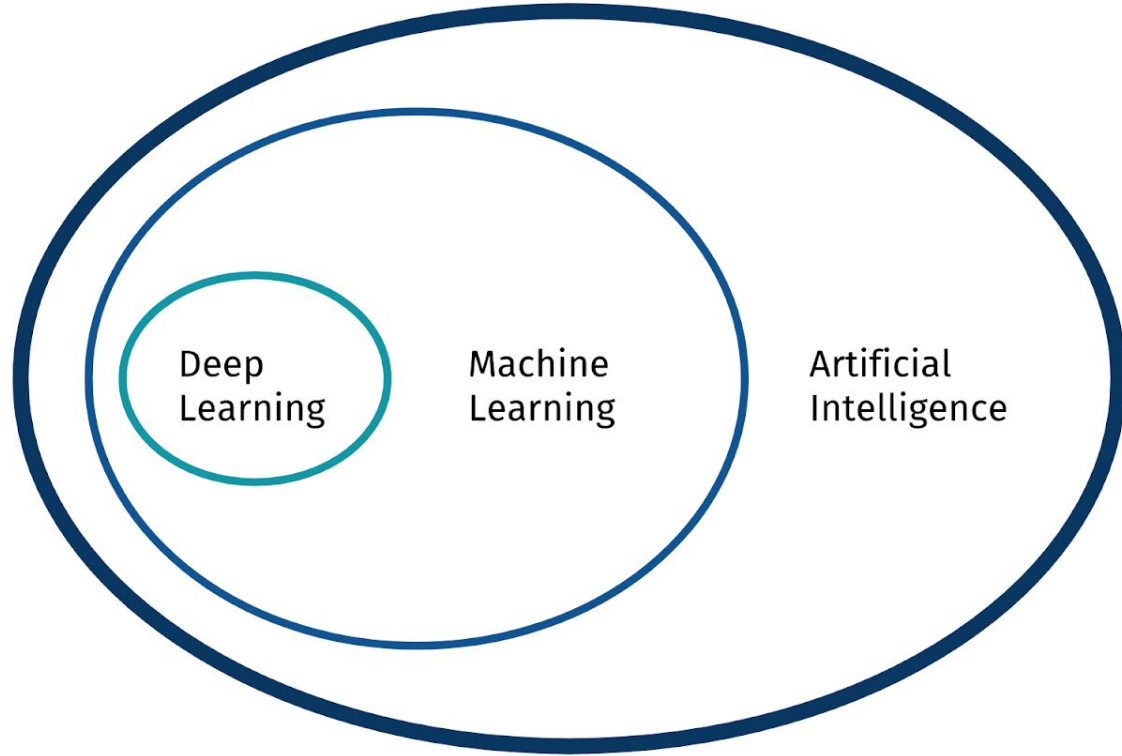


Fuente: [https://en.wikipedia.org/wiki/Iris\\_flower\\_data\\_set](https://en.wikipedia.org/wiki/Iris_flower_data_set)

# Problema de Iris



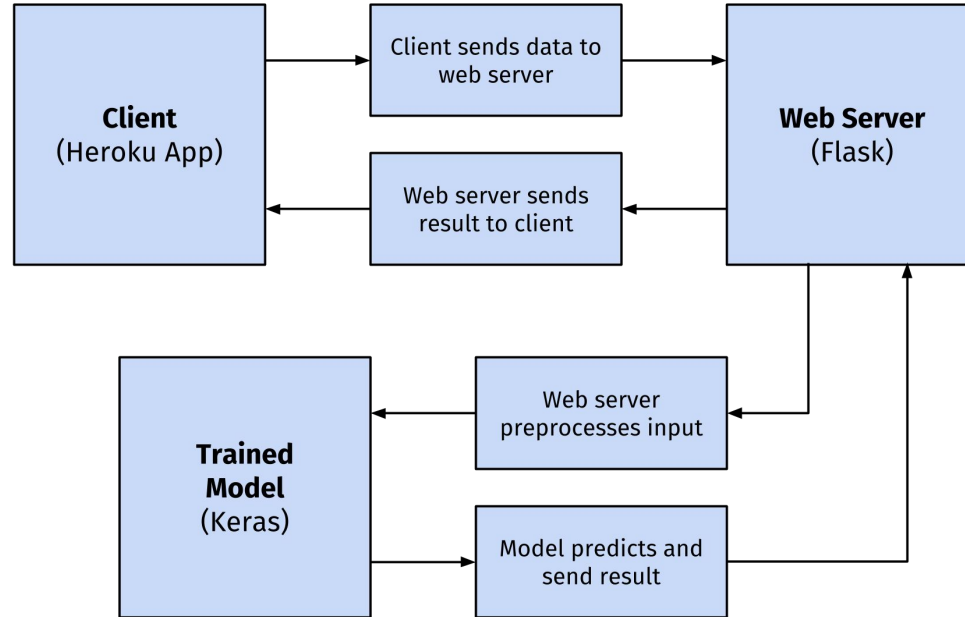
Fuente: <https://seaborn.pydata.org/generated/seaborn.pairplot.html>



# IA como servicio: API vs Web App

<https://kerasmnist.herokuapp.com>

# Diagrama de flujo de datos







UNIVERSIDAD DE  
GUANAJUATO



A nombre del **Future Lab**  
& **GitHub Campus Experts:**

**¡Gracias!**



**@FerroRodolfo**

Contacto: [ferro@cimat.mx](mailto:ferro@cimat.mx)

Sitio personal: [rodolfoferro.xyz](http://rodolfoferro.xyz)

Repositorio: <http://github.com/RodolfoFerro/PythonBaseAPI>