



Plomería con R

Desarrollo de API con `plumber.r`

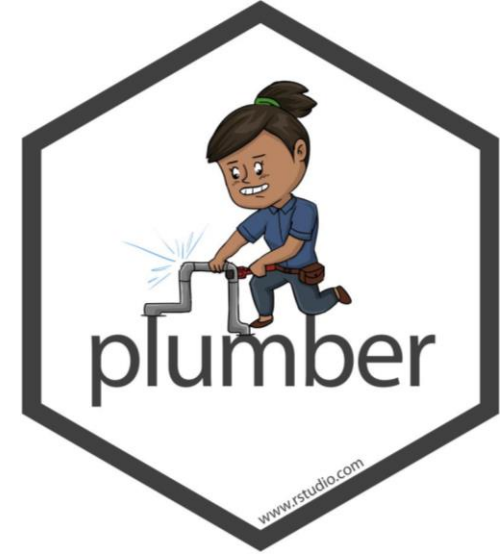
Gerardo Pirela

Data Scientist, 4ToldFinTech

gpirela@gmail.com

Visión general

- Preliminares:
 - API – Application Programming Interface
 - Swagger & Postman
- Introducción a librería Plumber
- Despliegue de modelos
- Hosting e integración continua con Docker
- Comentarios finales y referencias

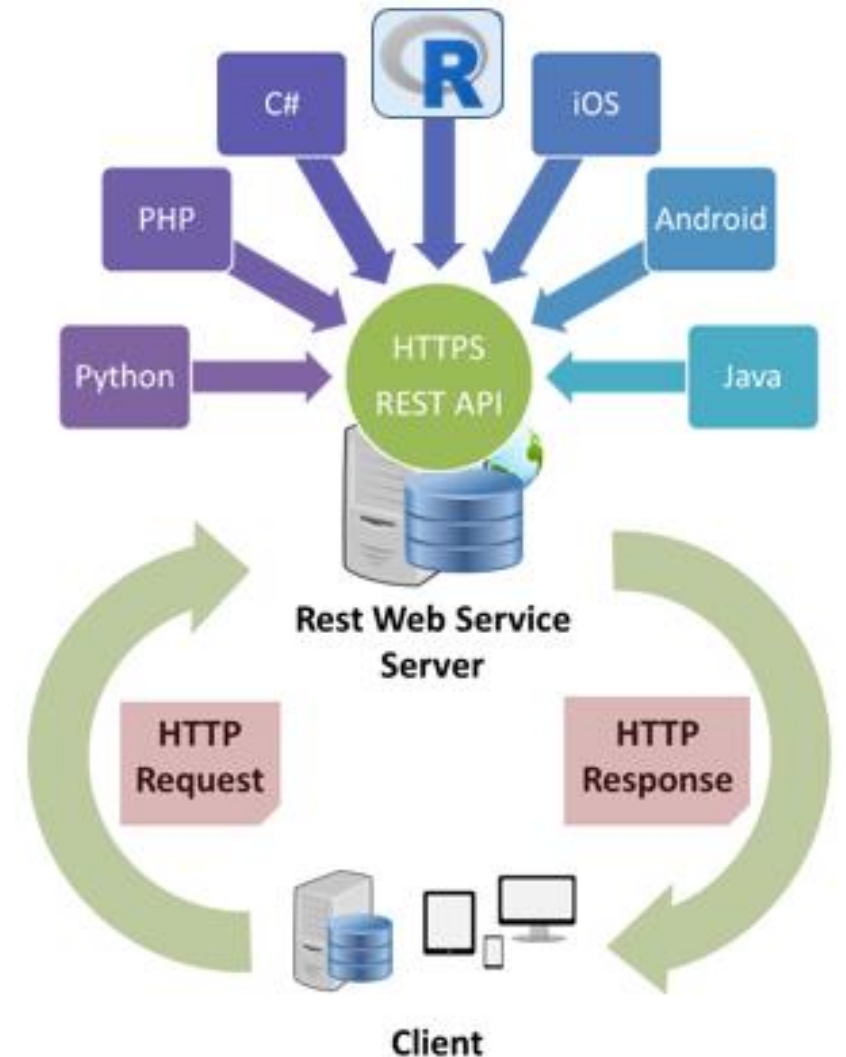




Preliminares:

Application Programming Interface

- Interfaces de programación de aplicaciones
 - Abstracción: biblioteca de funciones
 - Protocolo de comunicación
 - Envío parámetros y recepción de respuesta.
- WEB API
 - Protocolo de comunicación: HTTP
 - “Verbos” para invocar funciones: GET, PUT, ...
 - Envío y recepción de datos: *query string*, JSON, XML
 - REST-full – Arquitectura de microservicios

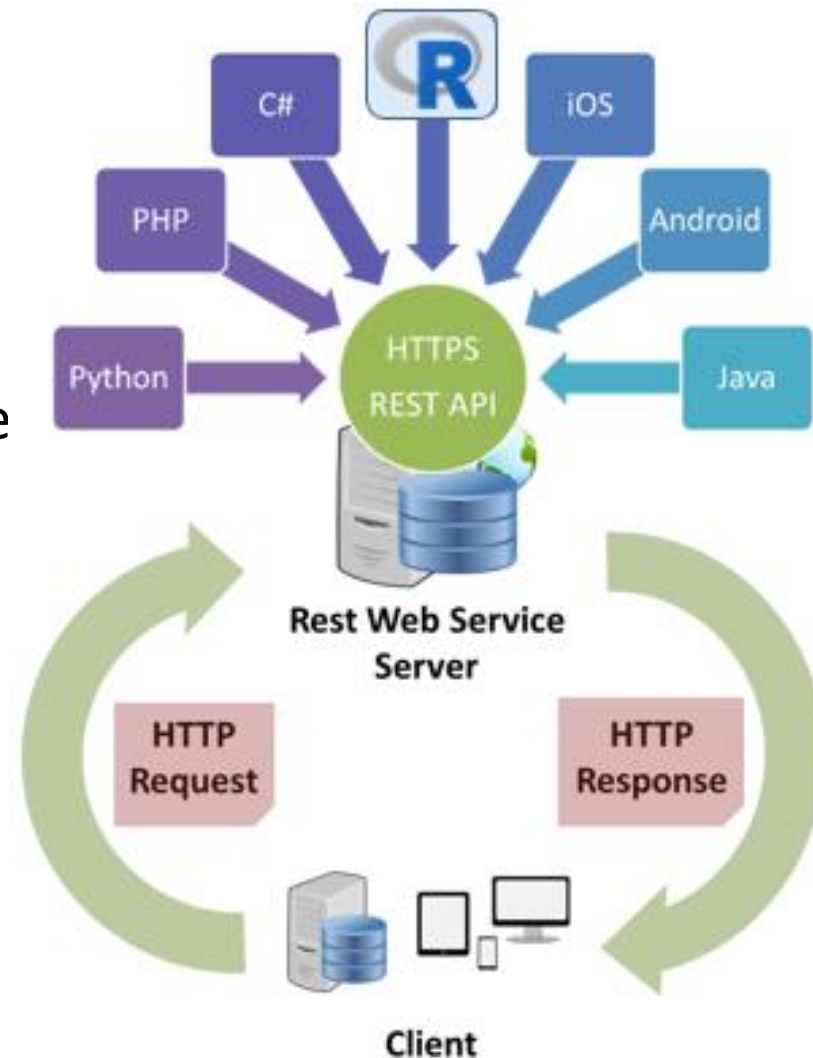


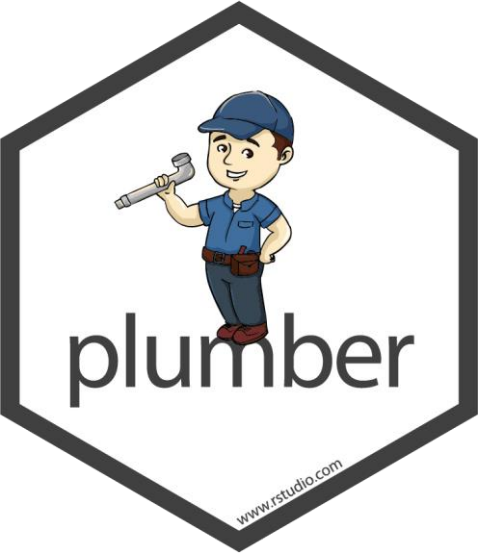
Preliminares:

Application Programming Interface

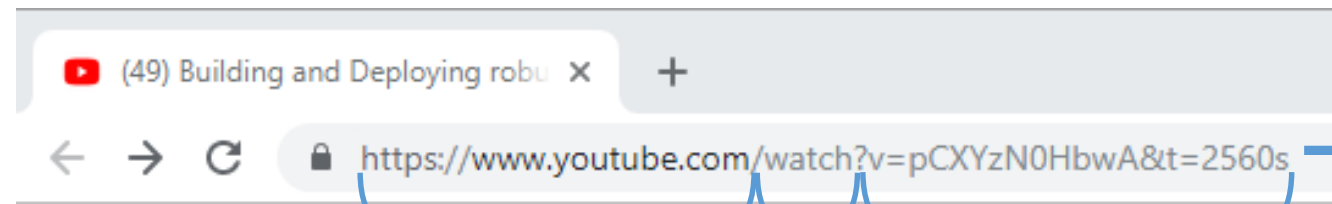


- Beneficios:
 - Flexibilidad de diseño
 - Acceso al servicio, agnóstico del lenguaje
 - Soporte multiplataforma y escalable
 - **Data Science:** Despliegue rápido de modelos





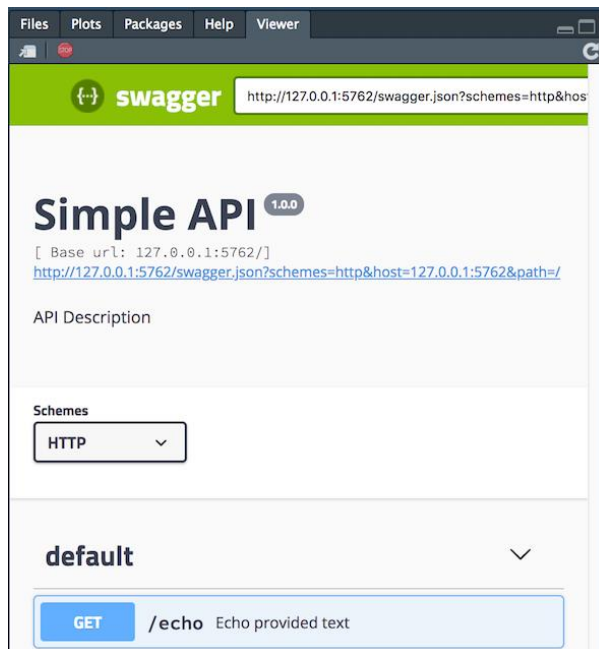
Preliminares:



Dirección del servidor

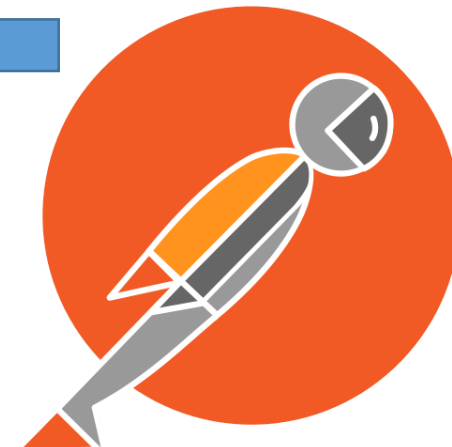
Parámetros

Nombre del servicio
(función)

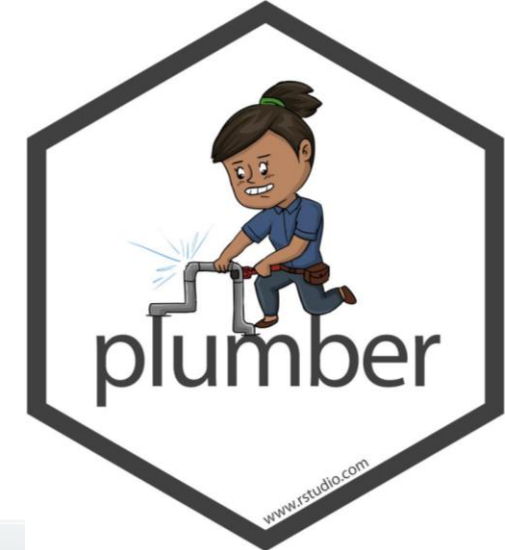


- Conjunto de herramientas para documentar e interactuar con API's, generar código y casos de uso.

- Entorno para organizar y probar con casos de uso, monitorizar y publicar API's.



Introducción a plumber.r



- Comentarios especiales para convertir las funciones de R en servicios web (“endpoints”).
 - `##` Comentarios en plumber
 - `@` Atributos del endpoint
 - Tipo (verbo http: get, post, ...)
 - Parámetros
 - “serializador” de la salida (JSON por defecto).

A screenshot of the RStudio integrated development environment. The window title is 'RStudio'. The menu bar includes 'File', 'Edit', 'Code', 'View', 'Plots', 'Session', 'Build', 'Debug', 'Profile', 'Tools', and 'Help'. The toolbar shows various icons for file operations and running code. The file explorer on the left shows two files: 'plomeria.r' and 'entrypoint.R'. The main editor window displays the code for 'plomeria.r'. The code is as follows:

```
1 library(plumber)
2
3 ## @apiTitle API de ejemplo
4 ## @apiDescription Endpoints para exponer funciones de ejemplo con Plumber.r
5
6
7 ## Replica cadena de texto recibida
8 ## @param string La cadena a replicar
9 ## @get /eco
10 function(string = 'Hola mundo!') {
11   list(paste0('Replico: ', string))
12 }
```



- Persistencia del modelo
- Pay load para enviar parámetros
 - Estructura de la respuesta

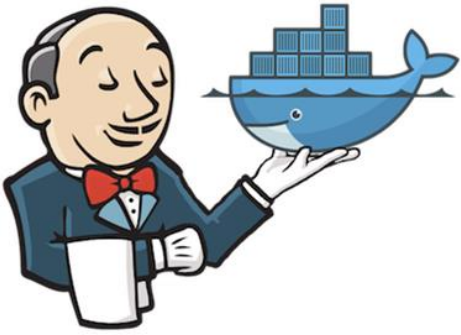
```
{  
  "features": {  
    "Sepal.Length": 3,  
    "Sepal.Width": 3,  
    "Petal.Length": 3,  
    "Petal.Width": 1  
  }  
}
```

```
{  
  "model": "irisVersicolorMDL",  
  "features": {  
    "Sepal.Length": 3,  
    "Sepal.Width": 3,  
    "Petal.Length": 3,  
    "Petal.Width": 1  
  },  
  "prediction": 0.3579  
}
```

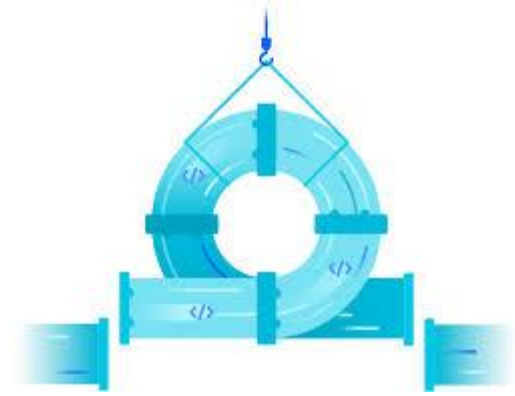
Notación JSON

Despliegue de modelos

```
plomeria.r* x  entrypt.R x  
← →  Source on Save  Run  Source  ▾  
1 library(jsonlite)  
2  
3  ## @apiTitle API de ejemplo  
4  ## @apiDescription Endpoints para exponer funciones de ejemplo con  
5  
6  
7  ## Replica cadena de texto recibida  
8  ## @param string La cadena a replicar  
9  ## @get /eco  
10 function(string = 'Hola mundo!') {  
11   list(paste0('Replico: ', string))  
12 }  
13  
14 ## Predictor con payload: json in, json out  
15 ## @serializer unboxedJSON  
16 ## @post /prediceVersicolor  
17 function(req) {  
18   injson <- fromJSON(req$postBody, simplifyVector = FALSE)  
19   model <- readRDS('models/irisVersicolorMDL.rds')  
20   predictores <- as.data.frame(injson$features)  
21   respuesta <- predict(model, predictores, type = 'response')  
22   list(timestamp = format(as.POSIXlt(Sys.time(), tz = "UTC"), '%Y-  
23     model = 'irisVersicolorMDL',  
24     features = injson$features,  
25     prediction = pred)  
26 }
```

Hosting e integración continua



```
>   GitHub, Inc. [US] | https://github.com/trestletech/plumber/blob/master/Dockerfile    
```

mtoto Install plumber from CRAN in top level Docker file (#292)

2 contributors

16 lines (12 sloc) | 486 Bytes Raw

```
1 FROM rocker/r-base
2 MAINTAINER Jeff Allen <docker@trestletech.com>
3
4 RUN apt-get update -qq && apt-get install -y \
5     git-core \
6     libssl-dev \
7     libcurl4-gnutls-dev
8
9 ## RUN R -e 'install.packages(c("devtools"))'
10 ## RUN R -e 'devtools::install_github("trestletech/plumber")'
11 RUN install2.r plumber
12
13 EXPOSE 8000
14 ENTRYPOINT ["R", "-e", "pr <- plumber::plumb(commandArgs()[4]); pr$run(host='0.0.0.0', port=8000)"]
15 CMD ["/usr/local/lib/R/site-library/plumber/examples/04-mean-sum/plumber.R"]
```





Trabajo futuro

- Rounting
- Filtros
- Debugging
- Manejo comprensivo de errores
- Manejo de información recibida con la petición HTTP
 - Cookies, dirección y puerto remoto, ...

- Seguridad



Referencias



- <https://www.rplumber.io/>
- <https://rviews.rstudio.com/2018/07/23/rest-apis-and-plumber/>
- <https://www.youtube.com/watch?v=pCXYzN0HbwA>
- <https://github.com/trestletech/plumber>
 - <https://github.com/trestletech/plumber/blob/master/Dockerfile>
- <https://www.getpostman.com/about-postman>
- <https://docs.docker.com/engine/reference/builder/>

