

Desarrollo de una aplicación con Bluemix

José Javier González Ortiz Lucía Montero Sanchis

24 de mayo de 2017

1. Introducción

Se ha desarrollado una aplicación Flask basada en Python, con los siguientes cinco servicios:

- Language Translator
- Visual Recognition
- Text to Speech
- Natural Language Understanding
- Cloudant NoSQL DB

El objetivo de la aplicación es ofrecer un servicio de análisis de imágenes y de frases en diferentes idiomas.

La aplicación usa cookies para mantener historial de usuarios de forma independiente. Se soporta también url estáticas para búsquedas ya realizadas. Esto permite poder compartir o salvar búsquedas previas.

Se ha empleado AJAX para las pantallas de carga y para obtener el historial de los usuarios. La aplicación soporta cacheado de resultados de cara a evitar repetir peticiones en un espacio de tiempo corto debido al alto numero de comunicaciones de microservicios presentes. Esto agiliza mucho consultas del historial.

Figura 1: La aplicación está disponible tanto en Bluemix como en una página web personal. El código está compartido en GitHub, y se ha utilizado TravisCI para la Integración Continua.



2. Enlaces URL

Figura 2: Bluemix: <http://erittely.josejg.com>
<http://erittely.eu-gb.mybluemix.net/>

ERITTELY
EL MEJOR SERVICIO DE ANÁLISIS DE FRASES E IMÁGENES ONLINE

ANÁLISIS DE FRASE

ANÁLISIS DE IMAGEN

Figura 3: GitHub: <https://github.com/JJGO/bluemix-project>

Project using IBM Bluemix cloud infrastructure

Add topics

63 commits	2 branches	0 releases	2 contributors	Apache-2.0
lumosan Merge branch 'master' of github.com:JJGO/bluemix-project				
hooks	Create pre-commit hook			3 days ago
report	Write report			8 minutes ago
static	Make fonts bigger and ul without bullets			2 hours ago
templates	Make longer truncate			13 minutes ago
.cignore	Add new entries to cignore			2 hours ago
.gitattributes	Add credentials for Bluemix microservices			4 days ago
.gitignore	Remove tmp audios and add rule to avoid in future			2 days ago
.travis.yml	Disable travis build for 2.7			an hour ago
LICENSE	Add README and LICENSE			4 days ago
Procfile	Add Bluemix deployment configuration files			4 days ago
README.md	Change README url			an hour ago
bluemix-services.ipynb	Add tweaks to notebook			5 hours ago
manifest.yml	Change Bluemix app name			2 hours ago
requirements.txt	Add python package requirements			4 days ago
runtime.txt	Add Bluemix deployment configuration files			4 days ago
server.py	Add test for index, text and image			an hour ago
services.py	Add voice getter to services library			3 days ago
setup.py	Create Minimal Flask Server with DB and translator			4 days ago
test_server.py	Added history test			an hour ago
vcap-local.json	Fix issue with https prefix for CloudantDB			4 days ago

Figura 4: Travis: <https://travis-ci.org/JJGO/bluemix-project>

Travis CI Blog Status Help Sign in with GitHub

JJGO / bluemix-project

build passing

Current Branches Build History Pull Requests More options

✓ master Merge branch 'develop' **-o- #16 passed**

- Commit 0d7306b
- Compare 0d4c43b..0d7306b
- Branch master

14 minutes ago

Jose Javier authored and committed

Ran for 9 min
Total time 2 min 28 sec

Build Jobs

✓ # 16.1	⌚ Python: 3.3	📦 no environment variables set	⌚ 47 sec
✓ # 16.2	⌚ Python: 3.5	📦 no environment variables set	⌚ 1 min 41 sec

Figura 5: El uso de Integración Continua queda reflejado en GitHub, donde el código aparece como 'build:passing'

Erittely GitHub, Inc. [US] | https://github.com/JJGO/bluemix-project

setup.py Added history test 9 minutes ago
test_server.py Fix issue with https prefix for CloudantDB 4 days ago
vcap-local.json

README.md

Bluemix Microservices Demo

build passing

This is just a proof-of-concept demo web app that uses a number of Bluemix microservices and deploys them using the Python web framework Flask

Requisites

You'll need the following:

- Bluemix account
- Cloud Foundry CLI
- Git
- Python

Demo

The website can be found [here](#).
Also a `jupyter` notebook is provided with [many examples](#)

3. Capturas de pantalla de la aplicación

3.1. Responsividad

Figura 6: Pantalla de inicio (responsividad I)

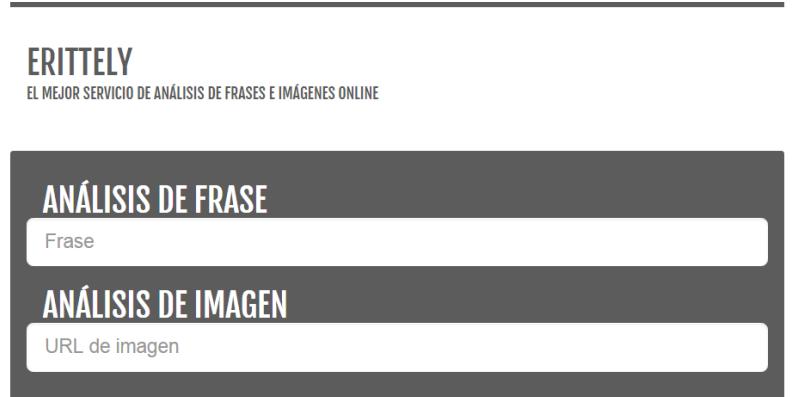


Figura 7: Pantalla de inicio (responsividad II)



Figura 8: Pantalla de inicio (responsividad III)

ERITTELY
EL MEJOR SERVICIO DE ANÁLISIS DE FRASES E
IMÁGENES ONLINE



3.2. Funcionamiento de la aplicación

Figura 9: (1) Introducir frase

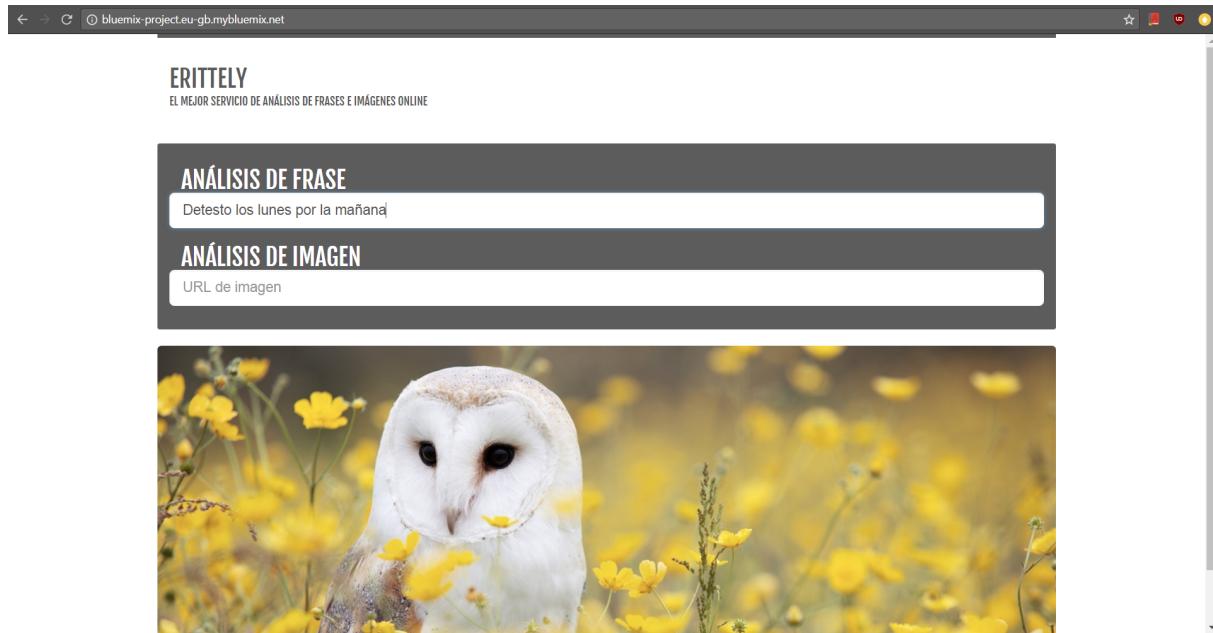


Figura 10: (2) Procesamiento de la frase

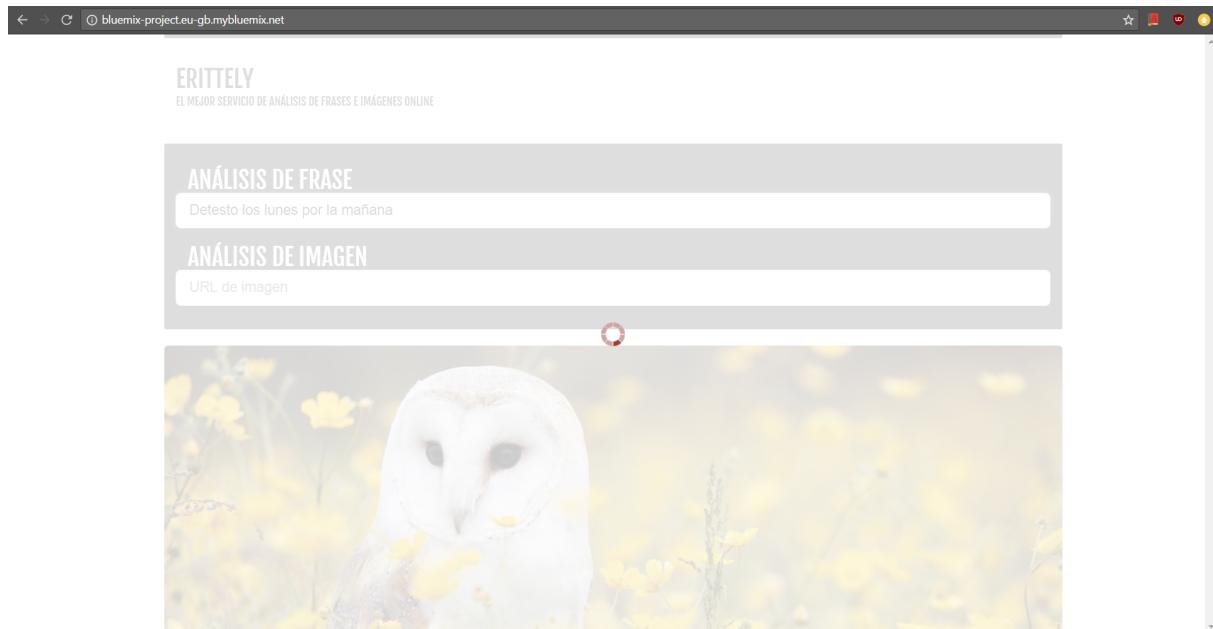


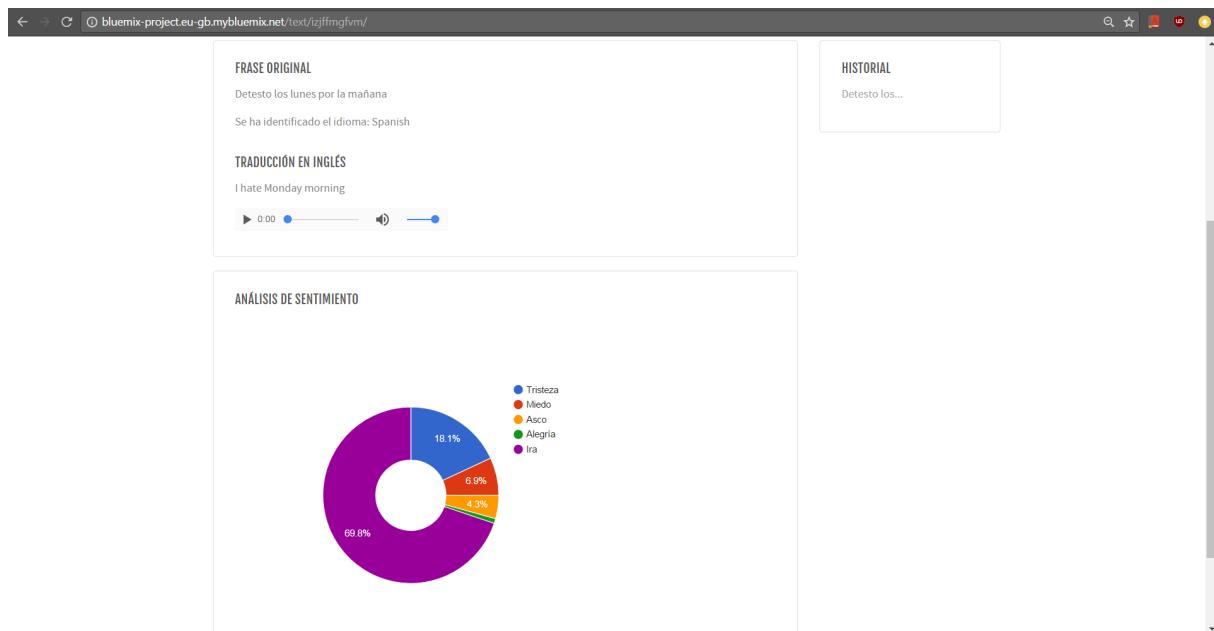
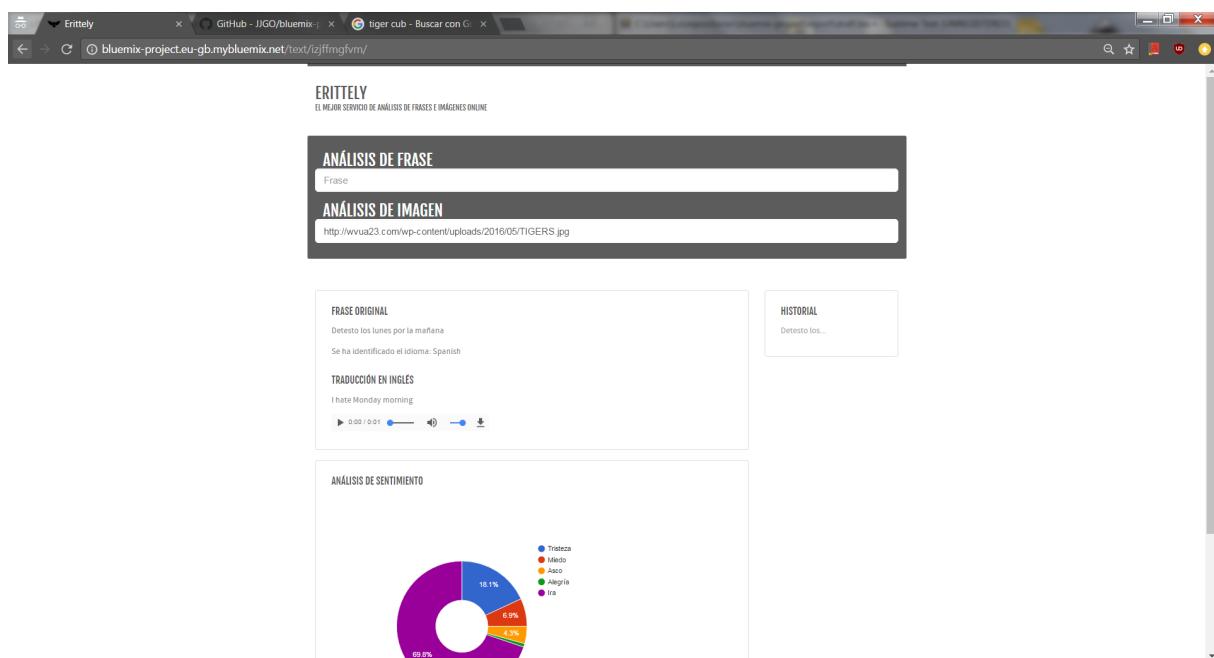
Figura 11: (3) Resultado. A la derecha aparece el historial, en el que ahora se incluye esta frase.**Figura 12:** (4) Ahora analizamos una imagen

Figura 13: (5) Resultado obtenido. Se ha añadido este elemento en el historial.

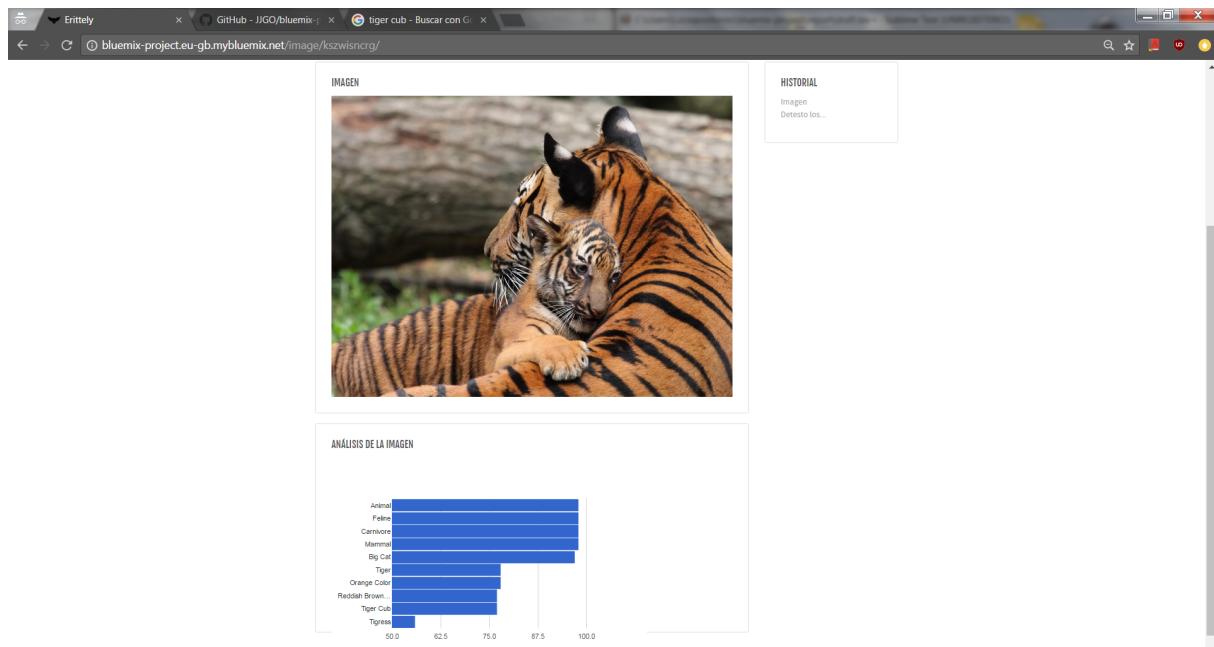


Figura 14: (6) Se puede hacer click en cualquier elemento del historial para repetir consultas anteriores

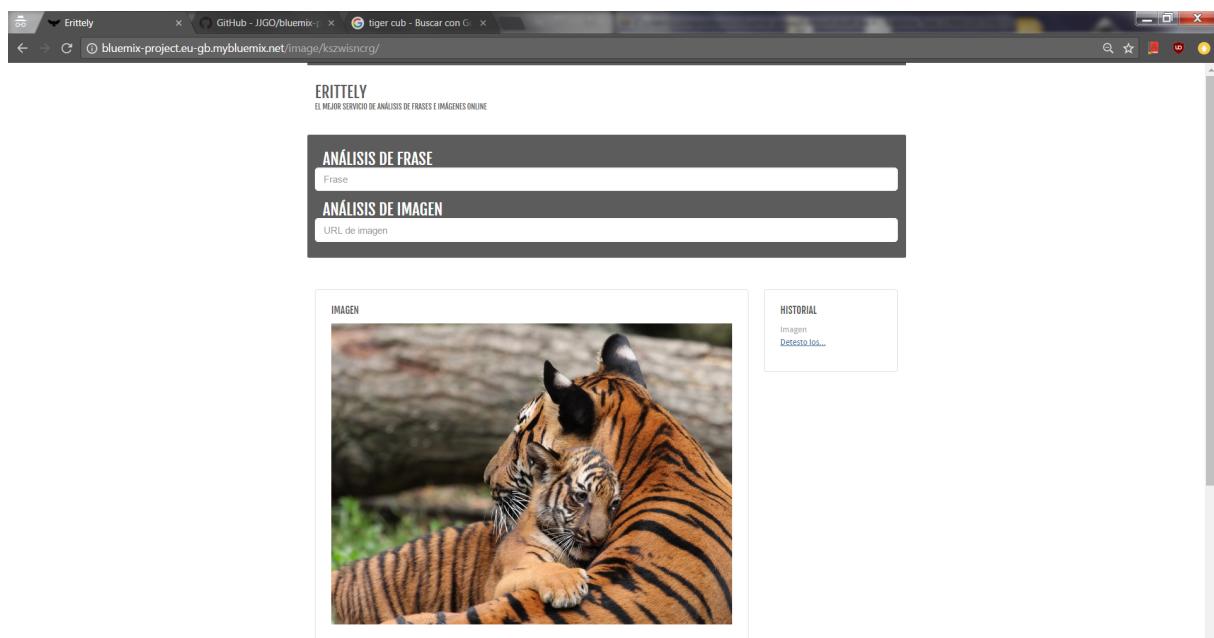


Figura 15: (7) Se repite la consulta anterior, y el elemento correspondiente del historial se coloca como el más reciente

