Meme Generator

Versión 0.0.1

Fernando D. Gómez

15 de enero de 2024

Contenido

1. Pre-requisitos **3**

2. Modo de uso **5**

3. API Reference **9**

3.1. `enums` 9

3.2. `generator` 10

3.3. `main` 11

Índice de Módulos Python **13**

Índice **15**

Aplicación para generar memes a partir de imágenes y texto.

CAPÍTULO 1

Pre-requisitos

Como primer paso es necesario instalar las dependencias:

```
python -m venv env
source env/bin/activate
pip install -r src/requirements.txt
```

CAPÍTULO 2

Modo de uso

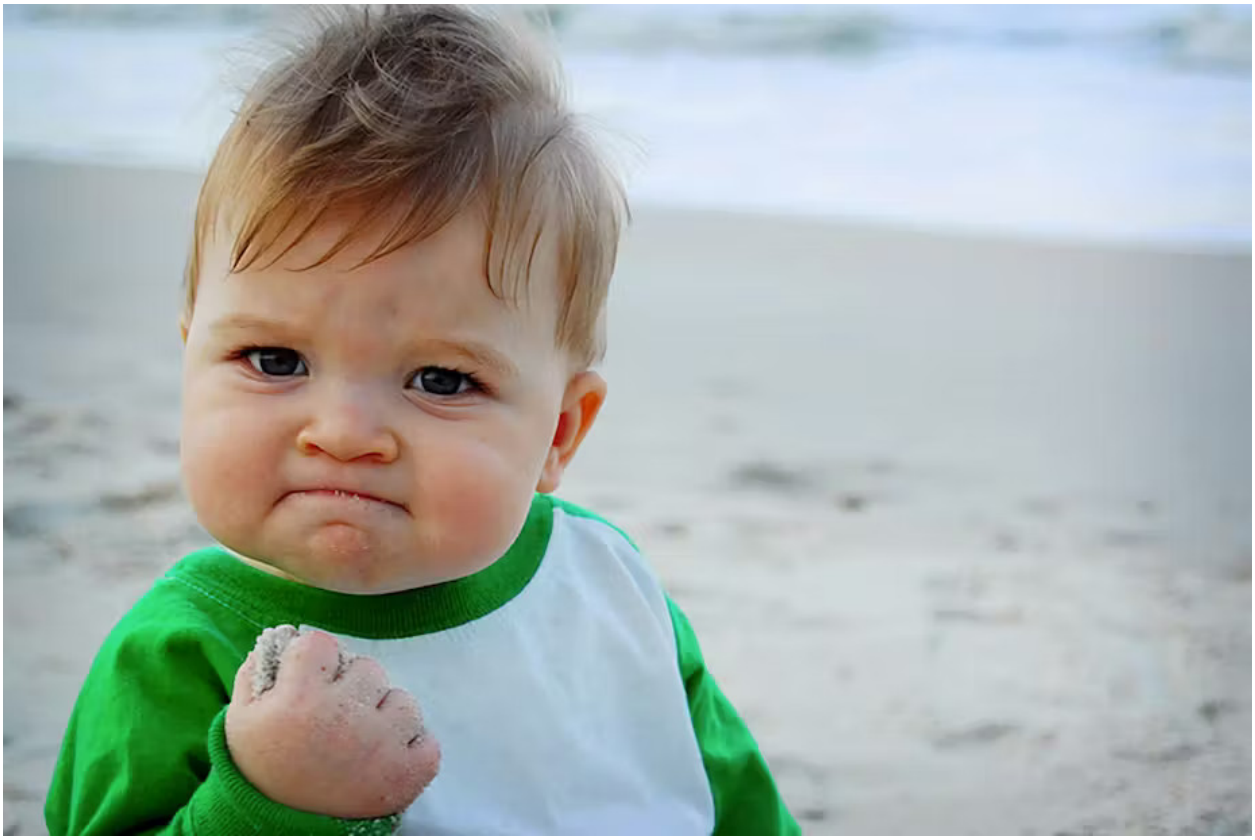
La generación de memes se realiza mediante la clase `Generator`, la cual toma como parámetros principales la imagen a utilizar, el texto a insertar en esa imagen y la orientación del texto.

El texto a insertar puede estar definido hasta en dos partes (**first_text**, **second_text**), para poder definir orientaciones independientes (**first_text_orientation**, **second_text_orientation**).:

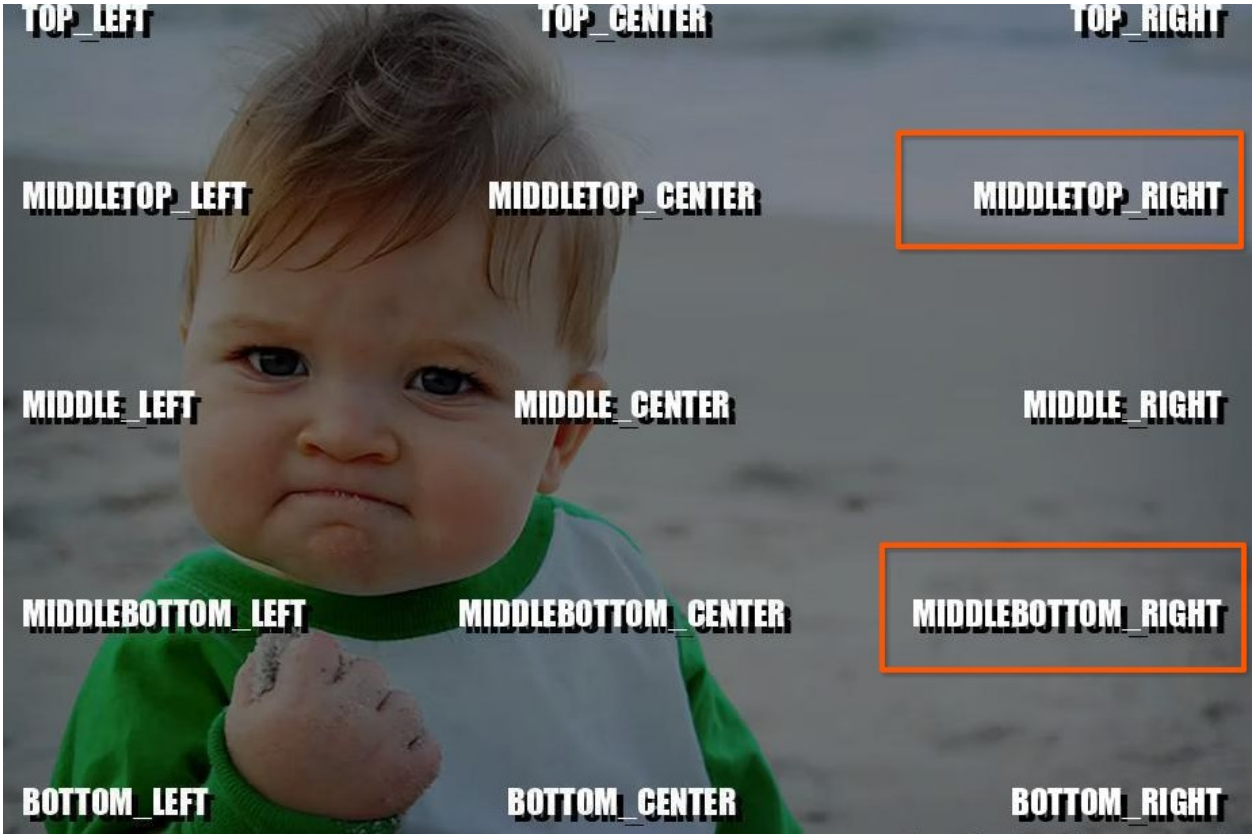
Para el parámetro **input_image_path** podemos proveer tanto una ruta local como una url de la imagen a utilizar.

Consejo: En el directorio `src/blank_images` del proyecto disponemos de algunas imagenes para utilizar.

Tomando como ejemplo la siguiente imagen:



Vamos a insertar dos textos, de acuerdo a las posiciones predeterminadas según la clase `TextPosition` tomaremos las posiciones **MIDDLETOP_RIGHT** y **MIDDLEBOTTOM_RIGHT**:



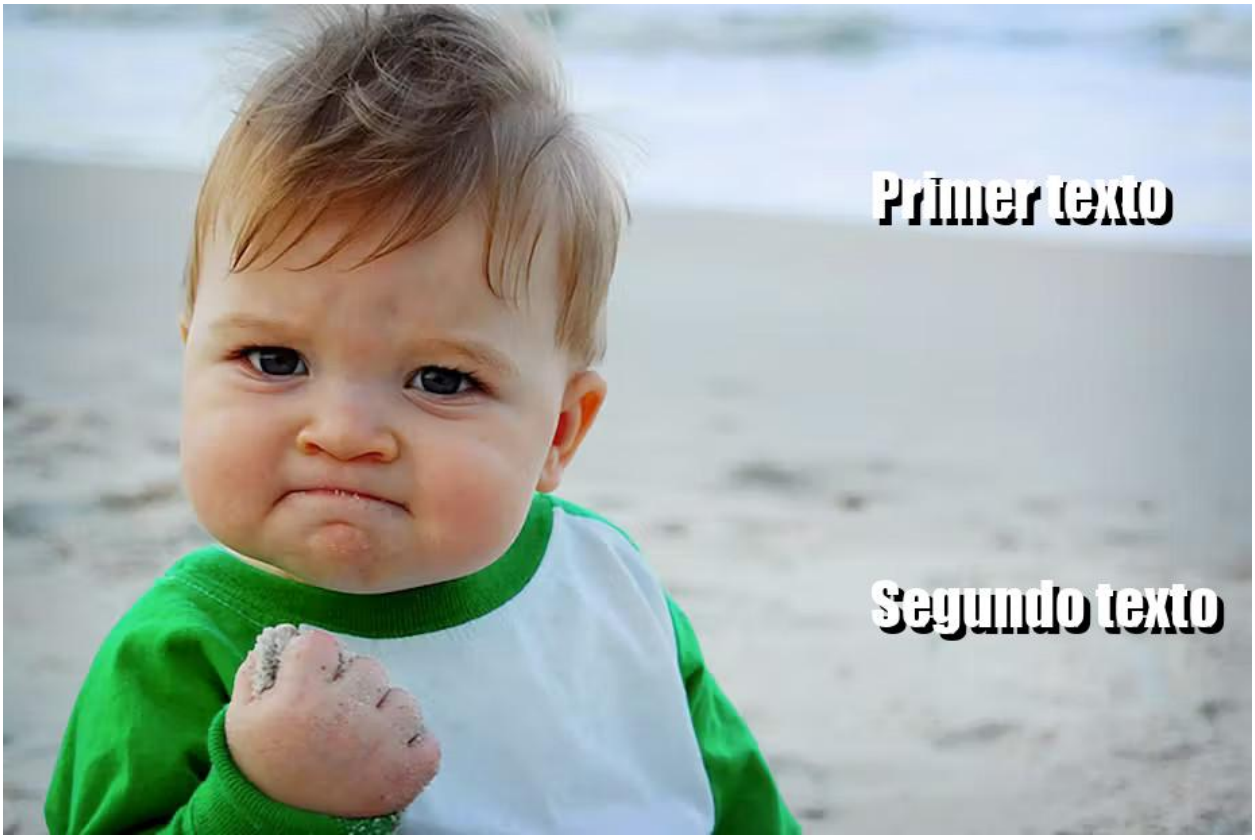

```
from src.generator import Generator
from src.generator import TextPosition

_generator = Generator()
_generator.generate_meme(
    input_image_path = './src/blank_images/bebe.jpg',
    alias = 'test_meme',
    first_text = 'Primer texto',
    first_text_orientation = TextPosition.MIDDLETOP_RIGHT,
    second_text = 'Segundo texto',
    second_text_orientation = TextPosition.MIDDLEBOTTOM_RIGHT
)
```

Nota: Si no definimos un valor para la constante `FONT_SIZE`, el tamaño de la letra será un poco reducido si definimos los dos grupos de texto en lugar de uno.

Podemos ejecutar el archivo `main.py` para generar el o los memes de prueba:

```
python ./src/main.py
```

Como resultado obtendremos la siguiente imagen:



CAPÍTULO 3

API Reference

This page contains auto-generated API reference documentation[1].

3.1 enums

Enumeraciones utilizadas en el proyecto.

3.1.1 Module Contents

Classes

TextPosition	Enumeración para las posiciones de los textos.

class enums.**TextPosition**(**args*, ***kwds*)

Bases: enum.Enum

Enumeración para las posiciones de los textos.

TOP_LEFT = 1

TOP_CENTER = 2

TOP_RIGHT = 3

MIDDLETOP_LEFT = 4

MIDDLETOP_CENTER = 5

MIDDLETOP_RIGHT = 6

[1] Created with sphinx-autoapi

MIDDLE_LEFT = 7

MIDDLE_CENTER = 8

MIDDLE_RIGHT = 9

MIDDLEBOTTOM_LEFT = 10

MIDDLEBOTTOM_CENTER = 11

MIDDLEBOTTOM_RIGHT = 12

BOTTOM_LEFT = 13

BOTTOM_CENTER = 14

BOTTOM_RIGHT = 15

3.2 generator

Módulo con la clase para generar memes a partir de una imagen y textos.

3.2.1 Module Contents

Classes

Generator	Clase del generador de memes

class generator.**Generator**

Clase del generador de memes

OUTPUT_DIR = './src/memes'

TEXT_COLOR = (255, 255, 255)

BORDER_COLOR = (0, 0, 0)

BORDER_SIZE = 5

FONT = 'impact.ttf'

FONT_SIZE

_get_image(*url_or_path: str*) → PIL.Image

Obtiene una imagen a partir de una url o path

Args:
url_or_path (str): url o path de la imagen

Raises:
Exception: Error al intentar obtener la imagen

Returns:
Image: Imagen obtenida

_draw_text(*image_draw_object: PIL.ImageDraw*, *text_position: tuple*, *text: str*, *font: PIL.ImageFont*, *text_color: tuple*, *border_color: tuple*, *border_size: int*)

Dibuja el texto en la imagen

Args:

image_draw_object (ImageDraw): instancia de la imagen. text_position (tuple): ubicación del texto en la imagen. text (str): texto para el meme. font (ImageFont): fuente a utilizar. text_color (tuple): color del texto. border_color (tuple): color del borde del texto. border_size (int): ancho del borde del texto.

_get_text_position(*image_width: int*, *image_heigth: int*, *text: str*, *font: PIL.ImageFont*, *text_position:* enums.TextPosition = *TextPosition.TOP_RIGHT*) → tuple

Obtiene la ubicación resultante del texto en la imagen según la posición definida

Args:

image_width (int): ancho de la imagen image_heigth (int): alto de la imagen text (str): texto utilizado. font (ImageFont): fuente utilizada. text_position (TextPosition, optional): posición definida para el texto. Defaults to TextPosition.TOP_RIGHT.

Returns:

tuple: ubicación del texto en la imagen. (x, y)

generate_meme(*input_image_path: str*, *output_image_path: str = ''*, *alias: str = ''*, *first_text: str = ''*, *first_text_orientation:* enums.TextPosition = *TextPosition.TOP_RIGHT*, *second_text: str = ''*, *second_text_orientation:* enums.TextPosition = *TextPosition.BOTTOM_RIGHT*, *text_color: tuple = TEXT_COLOR*, *border_color: tuple = BORDER_COLOR*, *border_size: int = BORDER_SIZE*)

Genera un meme a partir de una imagen, pueden proveerse hasta dos textos para agregar al meme

Args:

input_image_path (str): ruta de la imagen a utilizar, acepta también urls output_image_path (str, optional): Ruta destino para la imagen generada. Defaults to “”. alias (str, optional): nombre para el archivo. Defaults to “”. first_text (str, optional): primer texto a insertar. Defaults to “”. first_text_orientation (TextPosition, optional): posición donde se desea ubicar el texto. Defaults to TextPosition.TOP_RIGHT. second_text (str, optional): segundo texto a insertar. Defaults to “”. second_text_orientation (TextPosition, optional): posición donde se desea ubicar el texto. Defaults to TextPosition.BOTTOM_RIGHT. text_color (tuple, optional): Color a utilizar para el texto. Defaults to TEXT_COLOR. border_color (tuple, optional): Color a utilizar para los bordes del texto. Defaults to BORDER_COLOR. border_size (int, optional): Tamaño del borde del texto. Defaults to BORDER_SIZE.

Raises:

Exception: Error en caso de no definir textos.

3.3 main

Ejemplo de uso de la clase Generator.

3.3.1 Module Contents

main.**spacer**

Índice de Módulos Python

e

enums, 9

g

generator, 10

m

main, 11

Índice

Símbolos

_draw_text() (*método de generator.Generator*), 10
_get_image() (*método de generator.Generator*), 10
_get_text_position() (*método de generator.Generator*), 11

B

BORDER_COLOR (*atributo de generator.Generator*), 10
BORDER_SIZE (*atributo de generator.Generator*), 10
BOTTOM_CENTER (*atributo de enums.TextPosition*), 10
BOTTOM_LEFT (*atributo de enums.TextPosition*), 10
BOTTOM_RIGHT (*atributo de enums.TextPosition*), 10

E

enums
    module, 9

F

FONT (*atributo de generator.Generator*), 10
FONT_SIZE (*atributo de generator.Generator*), 10

G

generate_meme() (*método de generator.Generator*), 11
generator
    module, 10
Generator (*clase en generator*), 10

M

main
    module, 11
MIDDLE_CENTER (*atributo de enums.TextPosition*), 10
MIDDLE_LEFT (*atributo de enums.TextPosition*), 9
MIDDLE_RIGHT (*atributo de enums.TextPosition*), 10
MIDDLEBOTTOM_CENTER (*atributo de enums.TextPosition*), 10
MIDDLEBOTTOM_LEFT (*atributo de enums.TextPosition*), 10
MIDDLEBOTTOM_RIGHT (*atributo de enums.TextPosition*), 10
MIDDLETOP_CENTER (*atributo de enums.TextPosition*), 9
MIDDLETOP_LEFT (*atributo de enums.TextPosition*), 9
MIDDLETOP_RIGHT (*atributo de enums.TextPosition*), 9
module
    enums, 9
    generator, 10
    main, 11

O

OUTPUT_DIR (*atributo de generator.Generator*), 10

S

spacer (*en el módulo main*), 12

T

TEXT_COLOR (*atributo de generator.Generator*), 10
TextPosition (*clase en enums*), 9
TOP_CENTER (*atributo de enums.TextPosition*), 9
TOP_LEFT (*atributo de enums.TextPosition*), 9
TOP_RIGHT (*atributo de enums.TextPosition*), 9