



códigofacilito



# Proyecto final. 🦄

Bootcamp Python Avanzado.

Eduardo I. García Pérez.



## 📋 Objetivos.

- >\_ Implementar todos los conocimientos adquiridos a lo largo del Bootcamp.
- >\_ Fomentar el desarrollo e implementación de proyectos personales.
- >\_ Comenzar a desarrollar un portafolio de trabajo.





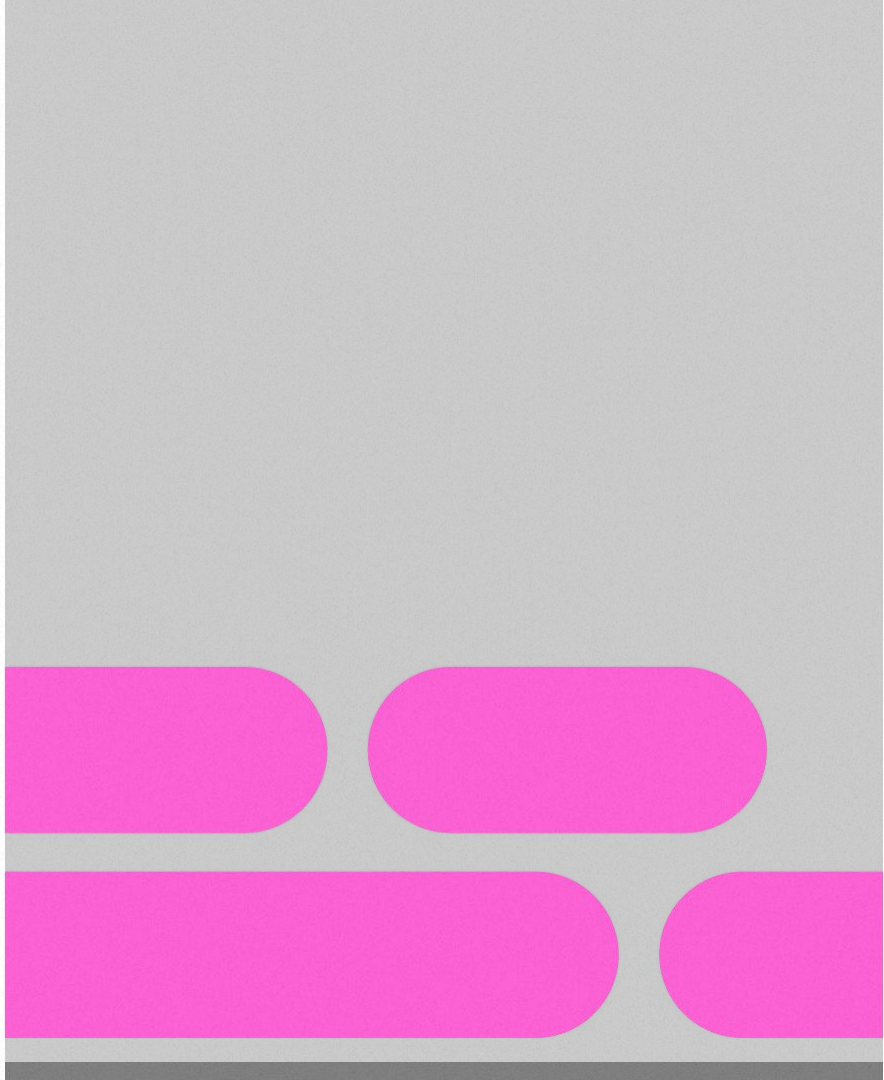
## >\_ Partes del proyecto.

- Desarrollo de Código.
- Repositorio en GitHub.
- Pruebas de software (Pruebas unitarias).
- Documentación del proceso de desarrollo.





**Detalles del proyecto.**







## >\_ Tecnologías a utilizar.

*Objetivo:* Implementar los conocimientos adquiridos a lo largo del bootcamp. No es obligatorio usar una o múltiples tecnologías en particular.

- Python en una versión superior a la 3.6





## >\_ Requerimientos Generales 1/2 🐍

- Python en una versión superior igual a la 3.6.
- Implementación de programación orientada a objetos.
  - Uso de interfaces, herencias y encapsulamiento.
- *Implementación de programación funcional\**
- Implementación de programación en concurrente y/o paralelo.
- Uso de decoradores.
- Uso de alguna estructura de datos.





## >\_ Requerimientos Generales 1/2. 🐍

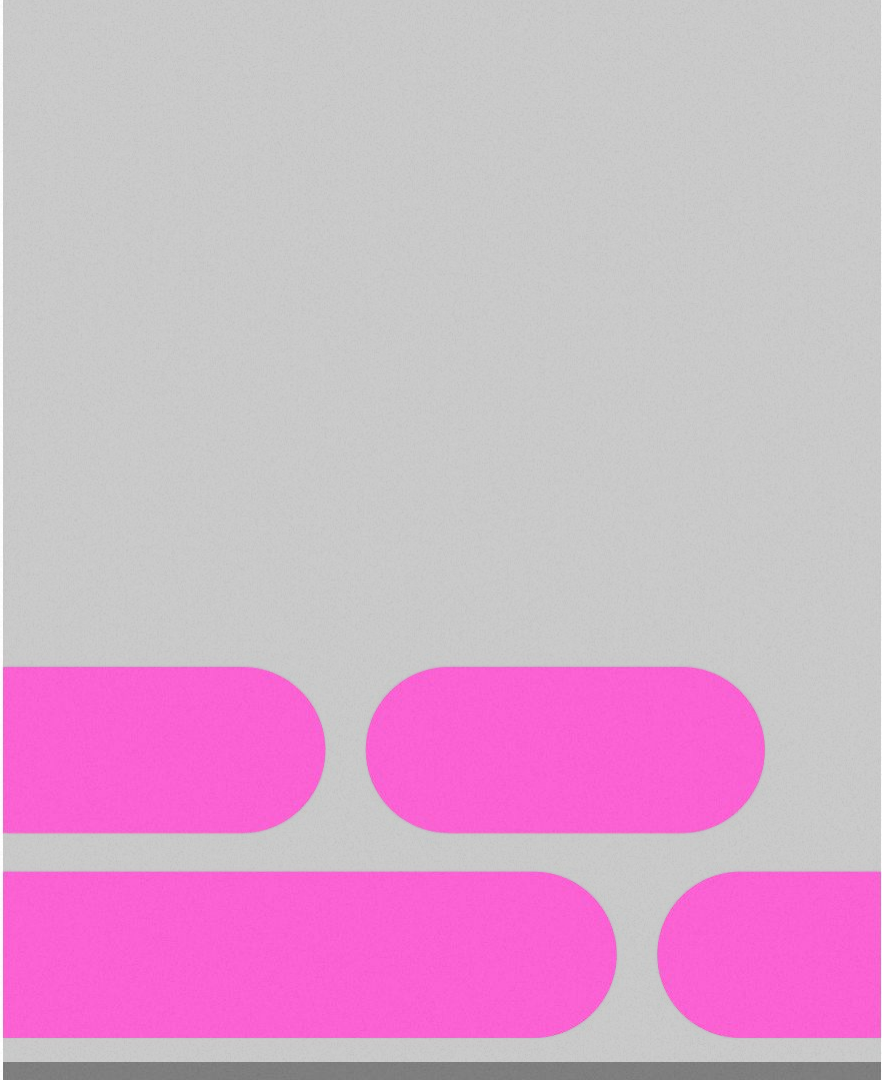
- Archivo(s) de pruebas unitarias.
- Uso de módulos y paquetes documentados mediante docstrings.
- Uso correcto de PEP8 y PEPs en general.







**Entregables.**







## >\_ Entregables.

- **Repositorio público en GitHub.**
  - Archivo README.md donde se detalla el cómo ejecutar el proyecto. (Formato Markdown).
  - Archivo(s) .py con el código de la implementación.
  - Archivo requirements.txt
- **Documentación.**
  - Archivo .pdf donde se detalle el funcionamiento del programa.
    - En el archivo se deben describir los retos presentados en la elaboración del proyecto y cómo fueron solucionados.
    - En caso de no terminar el proyecto, redacta las problemáticas que te impidieron finalizar.





## >\_ Formato de entrega. 🍞

- La entrega se hará mediante un formulario donde se solicitara:
  - Nombre completo para el certificado.
  - Link del repositorio de GitHub.
  - El archivo .PDF
  - Descripción del proyecto.





## >\_ Fecha máxima de entrega. 🕒

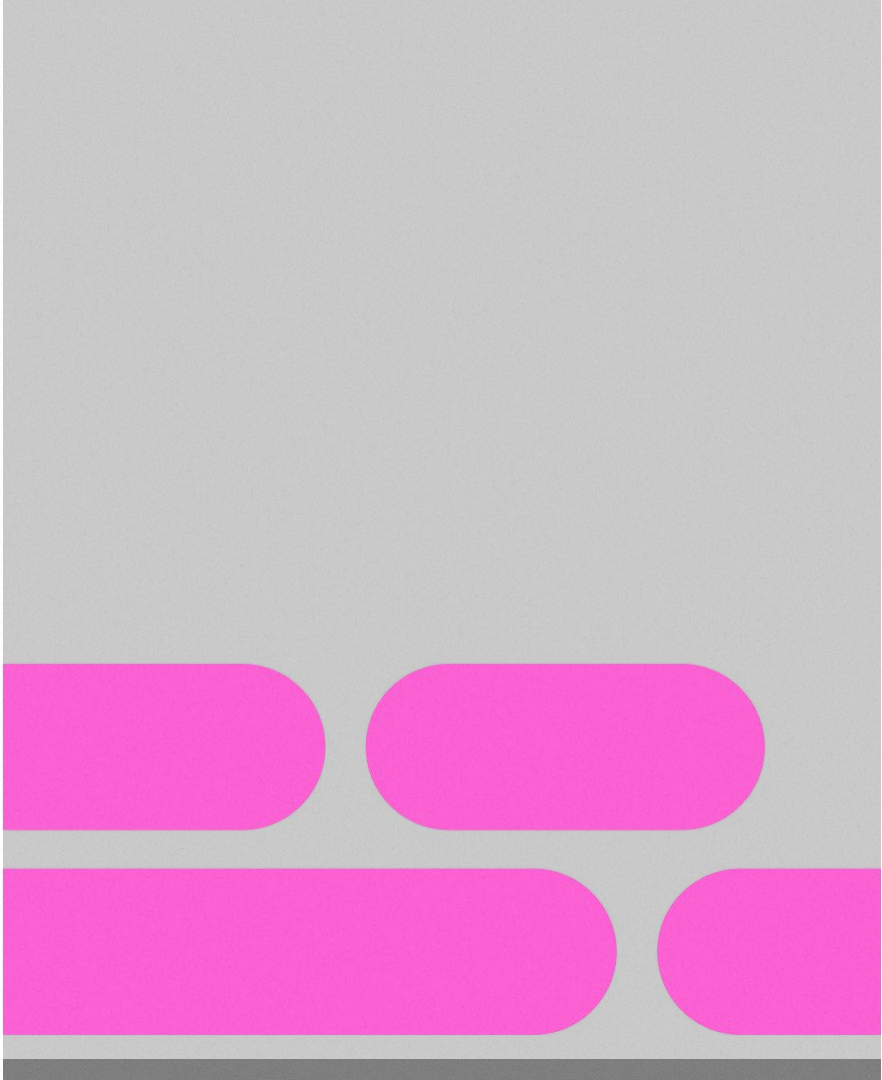
- 18 de Noviembre del 2024.







**Ideas de proyectos.**





## >- Sistema de alertas.

- El sistema debe notificar a los usuarios interesados sobre un producto, nota o tema de interesa de que algo ha llegado a su objetivo.
  - Ejemplos.
    - *Sistema de precios de criptomonedas.*
    - *Sistema de precios de productos.*
    - *Sistema de alerta de menciones por políticos.*
- El sistema debe recabar información de internet y llevar un registro de todos los datos recabados.
- En caso los usuarios no sean notificados de que su tema de interés no hay llegado a su objetivo en una N cantidad de tiempo, el sistema debe notificar de esto a los usuarios y expulsarlos de la cola de espera.





## >\_ Administrador de correos.

- El sitio web debe cumplir con los siguientes requerimientos.
  - Poder crear correos electrónicos.
  - Poder programar el envío de correos electrónicos.
  - Poder conocer la cantidad de personas que abrieron el correo y cuantas veces lo hicieron.
  - Después de que los correos han sido enviados, notificar al administrador un resumen de los correos enviados.
    - Cuantos se recibieron.
    - Cuantos se abrieron.
    - Cuandos se leyeron.







## >\_ Generador de informes. 🍭

- El sistema debe recabar información de internet y llevar un registro de todos los datos recabados.
- Los datos debe ser contrastados de, por lo menos, 2 fuentes diferentes.
- El sistema debe mostrar, mediante gráficos, la información recabada por periodos de tiempo.

Deseable: Mostrar en tiempo real los datos recabados a través de las gráficas.





## >\_ Simulación de threads y procesos.

- Crear un programa en Python, con interfaz gráfica, capaz de demostrar el funcionamiento de threads.
  - Cómo se crean.
  - Cómo duerme.
  - Cómo se destruyen
  - Cómo comparten información.
- Ejemplo: Simulador de física (Movimiento de partículas, caída libre, aceleración etc.. )





## >\_ Implementación de un algoritmo. 🌲

- Crear un programa en Python, con interfaz gráfica, capaz de demostrar el funcionamiento de un algoritmo de interés.
  - Cálculo en paralelo de números primos.
  - Procesamiento de imágenes con algoritmos como K neighbors, clasificando pixeles.
  - Algoritmo del viajero.
  - Algoritmo de Montecarlo.
  - Algoritmo de Polígonos.
  - Algoritmos de encriptación en paralelo.







## >\_ Otros.

- Programa a libre elección.
- Sistema de recomendación (Películas, Música, Libros).
- Plataforma de tienda en línea.
- Sistema de gestión de tareas.
- Plataforma de chat en tiempo real.
- Generador de texto automático.
- Juego en línea (Ajedrez, damas, batalla naval etc..)





códigofacilito



# Proyecto final. 🦄

Bootcamp Python Avanzado.

Eduardo I. García Pérez.

