

# TAREA 1

## OBJETIVOS

Los objetivos de esta primera tarea son:

1. Extraer los casos de uso de nuestra aplicación web de escritorio
2. A partir de estos casos de uso crear un diseño
3. Trasladar el prototipo del diseño a lenguaje UML con Mermaid
4. Implementar el diseño escogido en python GTK usando el patrón modelo,vista, presentador

## FINALIDAD

1. Familiarizarnos con python y con GTK
2. Aprender a implementar el patrón modelo, vista y presentador
3. Diseñar la parte gráfica de una interfaz
4. Trabajar en grupo usando una aplicación de control de versiones (github)

## RECURSOS

<https://www.youtube.com/watch?v=wrBgaP4r63E> → vídeo que explica un pouco o patrón MVP

<https://docs.gtk.org/gtk4/class.Paned.html> → instrucciones de como dividir una ventana en dos

<https://mermaid.js.org/syntax/classDiagram.html> → como se crea diagrama de clases en Mermaid

<https://www.thecocktaildb.com/api.php> → enlace a la API

<https://ellibrodepython.com/diccionarios-en-python> → diccionarios en python

<https://docs.gtk.org/gtk4/class.Image.html> → clase Gtk.Image y como se usa

<https://docs.gtk.org/gtk4/class.ListBox.html> → listas en GTK

<https://docs.gtk.org/gtk4/class.Box.html> → box

<https://docs.gtk.org/gtk4/class.SearchEntry.html> → barra de búsqueda

<https://docs.gtk.org/gtk4/class.ScrolledWindow.html> → scroller

<https://docs.gtk.org/gtk4/class.ComboBox.html> → desplegable

<https://docs.gtk.org/gdk-pixbuf/class.Pixbuf.html> → pixbuf (info) da foto

<https://realpython.com/python-requests/> → librería request de python para consultas html

<https://chat.openai.com/> → para consultar dudas y debuggear errores

## TAREA 2

### OBJETIVOS

- Detectar que operaciones pueden dar error y hacer cambios para que se avise al usuario de que se está produciendo un error
- Detectar operaciones que se puedan bloquear y modificar su código para que sea concurrente

### FINALIDAD

Con esta tarea hemos aprendido a elaborar una interfaz que sea concurrente y a gestionar sus errores para que el usuario tenga una mejor experiencia usando la interfaz

### RECURSOS

<https://docs.gtk.org/gtk4/class.MessageDialog.html> → para lanzar pop up de error

[https://github.com/sebp/PyGObject-Tutorial/blob/master/examples/messagedialog\\_example.py](https://github.com/sebp/PyGObject-Tutorial/blob/master/examples/messagedialog_example.py) → ejemplo de pop up

<https://docs.gtk.org/gtk4/class.Spinner.html> → uso del spinner

<https://es.stackoverflow.com/questions/412117/como-comprobar-conexi%C3%B3n-a-internet-en-python3> → para comprobar si tenemos conexión

<https://docs.python.org/es/3.8/library/threading.html> → threading en python

[https://docs.gtk.org/glib/func.idle\\_add.html](https://docs.gtk.org/glib/func.idle_add.html) → glib.idle\_add

<https://chat.openai.com/> → para consultar dudas y debuggear errores

## TAREA 3

### OBJETIVOS

- Internacionalizar la aplicación para que se adapte al locale escogido por el usuario
- Internacionalizar las unidades de medida para que se adapten al locale escogido

### FINALIDAD

Con esta tarea hemos aprendido a internacionalizar una aplicación a diferentes idiomas, en este caso la aplicación base está en inglés y la internacionalizamos al español y al alemán.

### RECURSOS

- <https://docs.python.org/3/library/locale.html> → uso del locale en python
- <https://translate.google.es/?hl=es&sl=en&tl=de&op=translate> → para traducir aplicación al alemán

