

Objetivos de aprendizaje:

- **Tarea 1:**

- Entender que patrones arquitectonicos existen a la hora de crear una IGU y cual es mejor utilizar en función de que aplicación se quiere desarrollar (en nuestro caso uno que separe la vista del estado/modelo).
- Aprender a programar con python, en concreto usando la librería GTK y separando los distintos componentes de la interfaz en distintos modulos.
- Aprender a diseñar, mediante buenas prácticas (botones con localización y tamaño correctos, textos visibles, usabilidad, etc), las diferentes pantallas que componen la aplicación con la herramienta Glade.
- Ser capaces de representar un diagrama UML tanto dinámico como estático a traves de mermaid e integrarlo en un archivo markdown.

-> Recursos:

- Patrón MVC: para usarlo se leen los distintos apartados de la página, nos ayuda a entender como funciona el patrón MVC, enlace: <https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/>
- Documentación GTK: para usarlo busca el campo de interés en el índice, nos ayuda a consultar los diferentes widgets existentes en GTK y para consultas relacionadas con esta librería, enlace: <https://python-gtk-3-tutorial.readthedocs.io/en/latest/index.html>
- Mermaid Live Editor: para usarlo tienes que poner tus diagramas en formato mermaid y podras previsualizar como quedarían en el markdown, lo usamos para ayudarnos a hacer los diagramas y comprobar que los hacemos correctamente, enlace: <https://mermaid.live/>

- **Tarea 2:**

- Entender como nuestra aplicación en la E/S debe funcionar de forma concurrente para evitar que esta se bloquee al realizar muchas peticiones.
- Identificar que casos de error puede tener nuestra aplicación y como debemos mostrar estos al usuario.

-> Recursos:

- Librería threading: para usarlo se identifican los campos interesantes al leerlo y se ve como se utilizan los elementos de la librería en los ejemplos, nos ayuda a entender como utilizar esta librería para gestionar las operaciones de E/S de forma concurrente en nuestra aplicación, enlace: <https://www.codigopiton.com/como-usar-hilos-o-threads-en-python/>
- Tutoría: mediante una tutoría con el profesor concretamos los elementos que faltaban en nuestra pantalla de error para que fuera correcta a nivel de diseño y proporcionase al usuario formas de retroceder en caso de error de forma clara.

- **Tarea 3:**

- Aprender como internacionalizar una aplicación de forma que un usuario que utilice su equipo en otro idioma pueda utilizar nuestra aplicación en este.

-> Recursos:

- Librería locale: se consulta la documentación para entender como funciona y por que funciones esta compuesta, nos ayudo para informarnos de como conseguir el idioma en el que estaba la máquina desde la que se ejecuta nuestra aplicación, enlace: <https://docs.python.org/3/library/locale.html>