

# Práctica 1 - Interfaces Gráficas de Usuario con Gtk+ y Python

Interfaces Persona Máquina

Curso 16/17

## Resumen

Aplicar los conocimientos adquiridos sobre el desarrollo de interfaces gráficas de usuario creando una aplicación cuya interfaz gráfica de escritorio se implementará utilizando el lenguaje de programación Python y la librería gráfica Gtk+.

## 1. Descripción

La aplicación que desarrollarás será una pequeña base de datos de películas que mostrará a la usuaria las películas que ya ha visto, las que tiene pendientes de ver, y, usando un servicio en red, ofrecerá recomendaciones a partir de películas ya vistas.

Los siguientes apartados describen los *sprints* que debes realizar según el orden de realización.

### 1.1. Requisitos no funcionales

- La implementación se realizará con python y GTK+.

## 2. Sprint 1

En este *sprint* debes realizar los siguientes pasos:

1. Diseña una IU que permita ver una lista de películas y realizar las tareas básicas: añadir, borrar, editar.

Puedes emplear el formato de tu elección para documentar el diseño.

2. Haz un diseño software de la aplicación para dar soporte a la interface que acabas de diseñar.

El diseño debe ajustarse a la idea básica del MVC: separar la vista del modelo.

Para documentar el diseño debes usar los diagramas UML necesarios.

3. Implementa el diseño.
4. Valida todos los pasos anteriores, en especial el funcionamiento de tu implementación. A continuación asigne al último commit del repositorio la etiqueta `sprint1`.
5. Valida el contenido del repositorio remoto<sup>1</sup>.

### 3. Sprint 2

En este *sprint* debes realizar los siguientes pasos:

1. Internacionaliza la aplicación.  
Usa `gettext` para el soporte de idiomas.
2. Localiza el idioma de la aplicación a dos idiomas de tu preferencia.
3. Valida todos los pasos anteriores, en especial el funcionamiento de tu implementación. A continuación asigne al último commit del repositorio la etiqueta `sprint2`.
4. Valida el contenido del repositorio remoto.

### 4. Sprint 3

En este *sprint* debes realizar los siguientes pasos:

1. Incrementa el diseño de la IU de forma que permita marcar las películas que ya hayan sido vistas, y poder ver por separado la lista de películas que han sido vistas y las que están pendientes.
2. Incrementa el diseño software de la aplicación para dar soporte a los cambios que estas introduciendo.
3. Implementa el diseño.
4. Valida todos los pasos anteriores, en especial el funcionamiento de tu implementación. A continuación asigne al último commit del repositorio la etiqueta `sprint3`.
5. Valida el contenido del repositorio remoto.

---

<sup>1</sup>HINT: después de clonarlo, puedes hacer un reset a la etiqueta `sprint1` (`git clone --hard sprint1`)

## 5. Sprint 4

En este *sprint* debes realizar los siguientes pasos:

1. Incrementa el diseño de la IU para incluir un mecanismo que ofrezca a la usuaria recomendaciones a partir de las películas ya vistas.  
IMPORTANTE: Las recomendaciones se obtienen de un servicio en red.
2. Incrementa el diseño software de la aplicación para dar soporte a los cambios que estas introduciendo.
3. Implementa el diseño.  
Puedes usar el servicio en red de tu preferencia. Si no conoces ninguno, te sugerimos el ofrecido en <https://www.themoviedb.org/>
4. Valida todos los pasos anteriores, en especial el funcionamiento de tu implementación. A continuación asigne al último commit del repositorio la etiqueta `sprint4`.
5. Valida el contenido del repositorio remoto.

## 6. Sprint 5

En este *sprint* debes realizar los siguientes pasos:

1. Documenta y corrige los casos en que la interface no cumple el principio “principle of least astonishment”.
2. Documenta y corrige los casos en que la interface:
  - no gestiona los errores,
  - no proporciona *feedback* cuando es necesario,
  - no la concurrencia, i.e. se bloquea.
3. Documenta y corrige los casos en que la interface no cumple las *Gnome Human Interface Guidelines*.
4. Valida todos los pasos anteriores, en especial el funcionamiento de tu implementación. A continuación asigne al último commit del repositorio la etiqueta `sprint5`.
5. Valida el contenido del repositorio remoto.

## 7. Evaluación

Para que la práctica pueda ser evaluada como apta, debe cumplir las siguientes condiciones:

- Ingeniería Software
  - La aplicación se ejecuta sin fallos.
  - Los commits reflejan el orden indicado para las tareas.
  - Los commits correspondientes al final de una tarea tienen la etiqueta correspondiente.
  - Se han completado todas las tareas.
  - El diseño software se ajusta al patrón MVC.
  - La parte del modelo se corresponde con las instrucciones adjuntas a la práctica.
- Experiencia de usuario
  - La interfaz cumple el principio “principle of least astonishment”.
  - La interfaz se ajusta a las *Gnome Human Interface Guidelines*.