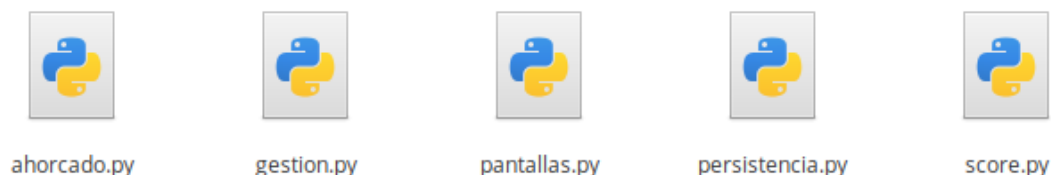


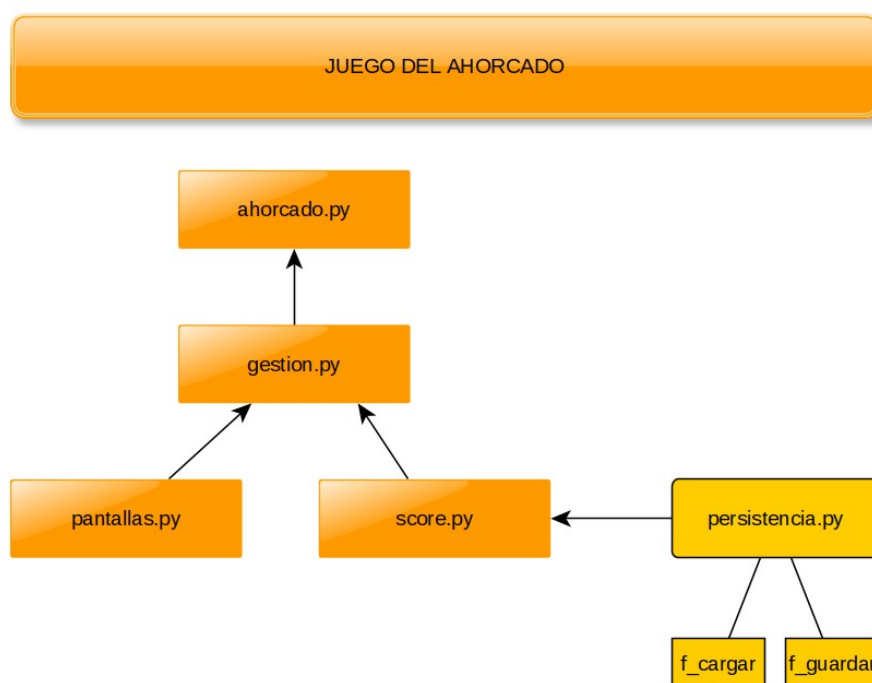
2º BTO	PROGRAMACIÓN Y COMPUTACIÓN	
CURSO 2021 - 2022	ACTIVIDAD 3.1 – MI PRIMER PROYECTO	
UD 3. DISEÑO DE PROGRAMAS		
ACTIVIDAD EVALUABLE	CRITERIOS DE EVALUACIÓN	Peso en %
	2.6. Aplicar la creatividad al proceso de desarrollo de software, transformando ideas en aplicaciones.	100%
	2.7. Elegir y utilizar Ide's, depuradores y herramientas de control de versiones de código.	100%
	2.8. Diseñar y probar programas propios o ajenos, elaborando la correspondiente documentación.	100%

En esta actividad **GRUPAL** se os va a dar un esqueleto de una aplicación funcional. Se trata del **Juego del Ahorcado**, para jugar en modo consola de comandos, en sistemas Windows y Linux. El programa está en la plataforma Moodle de la asignatura, comprimido con el nombre de **ahorcado.zip**.

La aplicación consta de 5 ficheros:



Cada uno de ellos tiene una funcionalidad diferente. Un esquema parcialmente resuelto de la estructura de la aplicación es el siguiente:



Vamos a analizar la situación del proyecto...

- ✓ Como puedes observar, la aplicación comienza con el fichero **ahorcado.py**, que contiene el punto de inicio del programa (es decir, para poder ejecutarlo en la consola de comandos basta con indicar *python3 ahorcado.py*). Este módulo llama a una función del módulo de gestión.
- ✓ El módulo **pantallas.py** incluye TODAS las funciones necesarias para visualizar y pedir datos por pantalla. Y así debe de seguir siendo... es decir, un *input* o un *print* únicamente puede estar en modo de función y dentro de este módulo.
- ✓ El módulo **score.py** contiene la gestión de los usuarios que pueden jugar al juego, sus puntuaciones, etc. Y solo puede usarse para este fin. Como podrás comprobar está vacío.
- ✓ El módulo **gestion.py** es el módulo de gestión del juego y contiene las funciones principales de la aplicación, es decir, el motor del juego y la gestión del menú del juego. Este módulo hace uso de varias funciones del módulo de pantallas y del módulo de score. Está parcialmente resuelto, como podrás comprobar.
- ✓ Finalmente, está el módulo **persistencia.py**, que contiene las funciones *f\_cargar* y *f\_guardar*, para gestionar la persistencia de los datos de los jugadores y de sus victorias, en disco. Este fichero también está vacío, pero se indica claramente que solo se van a crear esas dos funciones. Este módulo tiene que ser genérico, no importando la estructura a cargar o guardar, es decir, es independiente de la aplicación que lo use.

## ¿Qué se os pide?

De lo que trata esta actividad es de dar una funcionalidad completa al juego. Para ello habrá que:

- a) Implementar la gestión de jugadores y sus puntuaciones, de manera que en cada partida se indique el jugador que participa. Es decir para cada jugador se tiene que registrar el número de intentos erróneos. Por tanto, cada jugador debe tener una horca distinta, acorde a sus fallos. → Esto es, dar funcionalidad a la opción 1 y 2 del menú principal, y modificar la opción 3 para incluir a los jugadores.
- b) Implementar la persistencia de los datos de los jugadores, es decir, los jugadores que están dados de alta, que pueden darse de alta, y al finalizar el programa, guardar esos jugadores con sus puntuaciones en disco. → Dar funcionalidad a la opción 1 y a la opción 0, para cargar y guardar datos en disco.
- c) Existe una alternativa al uso del módulo *getpass*. Investiga sobre el módulo *pwininput*, el cual mejora a *getpass*, y úsalo en su lugar. **(PISTA: pip3 install pwininput)**.
- d) Implementa la funcionalidad de que cuando se introduzca una letra que ya ha sido introducida anteriormente y forma parte de la solución, lo tome como un error.
- e) Implementar una opción para visualizar los resultados de todos los jugadores. **PISTA:** Tienes en el módulo *pantalla.py* una función que ya hace esto...
- f) Implementar la funcionalidad para poder resolver el juego introduciendo la palabra directamente. **PISTA:** Ya existe la función de pedir datos en este caso.
- g) Implementar la funcionalidad para poder abandonar la partida a la mitad de la misma, independientemente de los jugadores que haya y del jugador que sea. Cuando un jugador finaliza la partida, se finaliza para todos.

- h) Crear una función **f\_finalizar(opcion)**, que incluya las funcionalidades de las funciones **f\_ganador** y **f\_perdedor**. Es decir, que dependiendo del parámetro opcion, haga una cosa u otra. En el código deberán comentarse las funciones **f\_ganador** y **f\_perdedor** (no borrarlas). Habrá que utilizar esta función nueva en lugar de las dos originales.
- i) Crear la documentación con **pydoc** del aplicativo.
- j) Utilizar **git** como sistema de control de versiones para alojar el proyecto en **GitHub**.

## Rúbrica

En esta sección se presenta la Rúbrica, que es el documento donde se especifica como se valora y se puntúa tu actividad.

<b>Rúbrica de la tarea para CE 2.6 - Aplicar la creatividad al proceso de desarrollo de software, transformando ideas en aplicaciones.</b>	<b>Puntuación máxima</b>
Implementa la gestión de jugadores y sus puntuaciones en el aplicativo, tanto en el alta de usuarios, como en su implementación en el juego por turnos de jugadores.	2,5
Implementa la persistencia de datos, pudiendo cargarse desde el menú principal y guardarse en disco al salir del aplicativo, de los jugadores junto con sus número de victorias.	2
Utiliza el módulo pwininput para la petición de la palabra secreta.	1
Implementa el error de inserción de letras que ya han sido dadas como válidas dentro de la palabra secreta.	1
Implementa la visualización de los resultados de todos los jugadores.	0,5
Implementa la funcionalidad para poder resolver el juego introduciendo la palabra directamente.	2
Implementa la funcionalidad para poder abandonar la partida a la mitad de la misma.	0,5
Crea la función f_finalizar y la utiliza en el programa, con una funcionalidad correcta.	0,5
<b>Rúbrica de la tarea para CE 2.7. Elegir y utilizar Ide's, depuradores y herramientas de control de versiones de código.</b>	<b>Puntuación máxima</b>
Usa un IDE para el desarrollo del proyecto.	3
Crea un proyecto en GitHub, sube el proyecto y usa el sistema de control de versiones git para realizarlo.	7
<b>Rúbrica de la tarea para CE 2.8. Diseñar y probar programas propios o ajenos, elaborando la correspondiente documentación.</b>	<b>Puntuación máxima</b>
Crea la documentación técnica del proyecto mediante pydoc y la entrega en formato HTML. Se tendrá en cuenta los comentarios internos del código fuente.	10

## Entrega

Se deberán entregar un fichero llamado **proyecto.zip** que contendrá:

- ✓ todos los ficheros Python del proyecto.
- ✓ Un fichero de texto, llamado **mi\_proyecto.txt**, que contendrá la URL del proyecto en GitHub.
- ✓ Una carpeta con la documentación generada por pydoc.

El fichero ZIP se entregará, atendiendo a todos los elementos de la Rúbrica, en la fecha límite indicada en la plataforma Moodle del centro.