

Hacer un programa principal *examen1.py* que llame a las funciones que incluyas en *examen1\_lib.py*. Las funciones de la librería serán:

1. **(1 pto) Menú.** Función desarrollada en *examen1\_lib.py* y llamada desde *examen1.py*, que imprimirá las opciones por pantalla y devolverá como parámetro la opción elegida por el usuario. Se mostrará:
  1. Jugar al adivinar palabras – Nivel fácil
  2. Jugar al adivinar palabras – Nivel medio
  3. Jugar al adivinar palabras – Nivel difícil
  4. Salir.

**Nota:** puedes no hacer este apartado y hacer las siguientes funciones.

2. **(2 ptos) Ordenar\_palabras.** Función que recibirá desde el programa principal un parámetro "listado" que contiene una colección de palabras sin ordenar y las guardará clasificadas en "palabras", que será una estructura de 3 filas: la primera fila aquellas palabras de longitud menor o igual a 4 (nivel fácil); la segunda fila las de longitud entre 5-7 (nivel medio); en la última fila las de longitud 8 o más (nivel difícil). La función devolverá la matriz de palabras.

**Nota:** puedes no hacer este apartado, pasándole a la siguiente función Jugar una matriz ordenada.

3. **(7 ptos) Jugar.** Función que:
  - Recibirá 2 parámetros: matriz de palabras ordenadas y un nivel de dificultad (0-bajo, 1-medio, 2-difícil)
  - Al principio, se mostrará la palabra oculta mediante guiones-bajos.
  - El usuario podrá cometer 5 fallos máximo para adivinar la palabra, informándole siempre sobre el número de intentos que le restan.
  - Si acierta una letra, se avisará con un mensaje "¡Bien! La Letra está presente", y se mostrará las letras que haya acertado en la posición que corresponda.
  - Si falla una letra, se avisará con un mensaje "Incorrecto, la letra no está presente". Igualmente se mostrará la parte de palabra que lleve resuelta.
  - Se deberá validar que el usuario introduce una letra (y solo una).
  - Al final del juego, si la palabra secreta es igual a la revelada (y no ha agotado el número de intentos), se mostrará un mensaje de "ENHORABUENA, HAS GANADO EL JUEGO".

**Puntuación de este apartado:**

- A) Si el programa funciona y muestra las que se van acertando en su posición y va descartando las incorrectas: 2 ptos.
- B) Si además de lo anterior, se controlan las validan los datos introducidos: 4 ptos
- C) Si además de lo anterior, muestra el número de intentos restantes: 5 ptos.
- D) Si funciona todo, 7 puntos.

**Ejemplo:** para la palabra secreta "móvil", donde el usuario a indicado la "a", "o", "r", "m", "w"

-----  
Introduzca una letra: "W"

Error, la letra no está presente. Te quedan 2 intentos

M O \_ \_ \_  
-----

**Criterios de entrega:**

Debes entregar 2 ficheros de python con los nombre indicados. Además incluirás un leeme.txt donde indiques los apartados realizados para que se puedan valorar independientemente (ejemplo: apartado 1, 2, 3B)

| <b>Criterios de calificación</b>   | <b>Punt.</b> |
|--|--------------|
| Si una función no compila y no funciona, no se califica  |              |
| ESTILO: comentarios, tabulaciones, nombres apropiados de variables, etc.   | Hasta -2     |
| “Acabado final”, código eficiente y bien estructurado, apreciación del profesor  | Hasta -2     |
| * El alumno debe ser capaz de contestar apropiadamente a las preguntas que se le realicen sobre el código y la funcionalidad de su aplicación, así como realizar pequeñas modificaciones que el profesor le solicite durante la POSIBLE revisión de su prueba de evaluación. | -10          |