UD5 – Caso Práctico 3: Juego del ahorcado

OBJETIVO

Implementar un programa que permita jugar al juego del ahorcado según las indicaciones indicadas abajo.

DINÁMICA DEL JUEGO (adaptada)

- El ahorcado es un juego en el que el jugador tiene que intentar adivinar una palabra proponiendo letras. Al principio del juego se elije una palabra aleatoria que el usuario desconoce y se le muestra 'oculta', sustituyendo las letras por guiones.
- A continuación el jugador propone una letra y si está en la palabra entonces acierta y se le muestra la palabra con dicha letra 'visible'. Sin embargo, si ha propuesto una letra que no está en la palabra entonces el jugador pierde una vida.
- El jugador debe seguir proponiendo letras hasta conseguir adivinar todas las letras de la palabra, es decir, la palabra completa (en tal caso gana la partida) o hasta que se quede sin vidas (pierde la partida y se le muestra de qué palabra se trataba).
- En cada partida se utiliza una palabra elegida al azar y el jugador tiene 5 vidas. Debe llevarse la cuenta de qué letras se han usado para que el usuario lo sepa.

Por ejemplo, supongamos que la palabra es "CASA", que tiene 4 letras. Se le mostrará al usuario esta información y esperará a que introduzca una letra:

```
¡COMIENZA EL JUEGO DEL AHORCADO!
Palabra: - - - -
Vidas: 5   Letras usadas:
```

Supongamos que el usuario intropduce una A, que sí existe en la palabra. Esa letra pasará a estar 'visible' y se añade la A a las letras usadas.

```
¡HAS ACERTADO! :)
Palabra: - A - A
Vidas: 5   Letras usadas: A
```

Supongamos que luego el usuario introduce una E, que no existe en la palabra. El usuario pierde una vida y se añade la E a las letras usadas.

```
¡HAS FALLADO! :(
Palabra: - A - A
Vidas: 4   Letras usadas: A E
```

Y así una y otra vez hasta que el usuario acierte todas las letras de la palabra (se mostrará el mensaje `¡HAS GANADO!`) o hasta que se quede sin vidas (se mostrará el mensaje `¡HAS PERDIDO!` y se mostrará qué palabra era).

CONSIDERACIONES

- El programa debe tener almacenadas al menos 50 palabras para jugar.
- Todas las palabras a adivinar estarán en mayúscula y no se permiten palabras compuestas ni otros signos de puntuación. No se usarán tildes. Solo letras de A a Z.
- Cuando el usuario elija una letra la puede introducir tanto en mayúscula como en minúscuyla indistintamente.
- No se permite que el usuario proponga una letra que ya ha utilizado antes. Si lo hacem esto no le penaliza pero se le volverá a pedir que proponga una letra.

PASOS

1. **Realiza el diseño descendente (top-down)** del programa, dividiendo el problema en subproblemas más pequeños, sucesivamente hasta que sean simples y no sea necesario dividirlos más.
2. **Piensa de qué forma puedes almacenar la información más importante**. ¿Qué variables o estructuras de datos puedes necesitar? ¿De qué tipo serían?
3. **Identifica qué subproblemas convendría programar como una función**. Identifica también si hay subproblemas parecidos que puedan ser resueltos por la misma función mediante parámetros, o si conviene crear funciones sobrecargadas.
4. **Haz una lista con las funciones** que necesitarás programar (no todos los subproblemas necesitan ser funciones). Incluye la cabecera completa (tipo devuelto, nombre y parámetros) y una comentario explicando qué hace cada una de ellas.
5. **Implementa el programa**: primero todas las funciones y por último el main.