

## TDA Ahorcado

**Nombre:** Ahorcado.

**Invariante:** -

**Operaciones:**

- ahorcado: string x entero → Ahorcado
- ~ahorcado: Ahorcado → !Ahorcado
- arriesgar: string x letra → Ahorcado
- arriesgar: string x string → Ahorcado
- adivino\_letra: string x char → bool
- adivino\_palabra: string x string → bool
- obtener\_intentos: → entero
- iniciar\_juego: (entero x char) U (entero x string) → Ahorcado
- finalizar\_juego: (entero x entero) U NULL → Ahorcado
- termino: (entero x entero) U (vector x vector) → Ahorcado
- imprimir\_vec\_guiones: → void
- imprimir\_letras\_arriesgadas: → void
- arriesgo\_esa\_letra: char x char → bool
- vector\_a\_str: char x vector x string → string

**Detalles de las operaciones:**

- Operación: Ahorcado
  - Descripción: construye un juego Ahorcado, con dos parámetros. Uno es la palabra a adivinar y otro es la cantidad de intentos para adivinar dicha palabra.
  - Pre condición: La palabra debe tener longitud mínima 3 y máxima 15. Cantidad de intentos es un entero que como mínimo debe ser 3 y máximo 12.
  - Post condición: construye a Ahorcado, inicializando 2 vectores dinámicos de caracteres (palabra a adivinar y vector con guiones), cantidad de intentos y cantidad de letras arriesgadas.
- Operación: ~Ahorcado
  - Descripción: destruye a Ahorcado, no recibe parámetros.
  - Pre condición: -
  - Post condición: libera la memoria dinámica que se uso en Ahorcado, destruye a Ahorcado.
- Operación: arriesgar
  - Descripción: se recibe un char por parámetro, se llama a la función adivino\_letra para verificar si está en la palabra a adivinar. Se modifican el vector de letras arriesgadas y cantidad de vectores. Se verifica si el juego termino.
  - Pre condición: el char debe estar en mayúscula.
  - Post condición: si el char no está en la palabra se le resta 1 a cantidad de intentos. Luego se ve si el juego sigue o termino.
- Operación arriesgar
  - Descripción: se recibe un string por parámetro, se llama a la función adivino\_palabra. Se verifica si el juego termino.

- Pre condición: el string debe estar todo en mayúscula.
  - Post condición: si el string es igual a la palabra a adivinar se llama a finalizar\_juego, en caso contrario se le resta 2 a cantidad de intentos. Luego se verifica si el juego sigue o termino.
- Operación: adivino\_letra
  - Descripción: se recibe un char por parámetro y se comprueba si está en la palabra a adivinar.
  - Pre condición: el char debe estar en mayúscula.
  - Post condición: si el char se encuentra en la palabra a adivinar se devuelve TRUE y se modifica el vector con guiones, en caso contrario se devuelve FALSE y no se hace nada.
- Operación: adivino\_palabra
  - Descripción: se recibe un string por parámetro y se comprueba si es igual a la palabra a adivinar
  - Pre condición: la palabra debe estar toda en mayúsculas y debe tener la misma longitud que la palabra a adivinar.
  - Post condición: si el string pasado por parámetro es igual a la palabra a adivinar se devuelve TRUE, caso contrario se devuelve FALSE.
- Operación: obtener\_intentos
  - Descripción: se devuelve un entero que es la cantidad de intentos que tiene Ahorcado, como mínimo puede ser -1 y máximo 12.
  - Pre condición: -
  - Post condición: devuelve un entero, que es la cantidad de intentos.
- Operación: iniciar\_juego
  - Descripción: se pregunta al usuario si va a arriesgar una letra o una palabra. Depende el caso se pide dicho ingreso y se llama al arriesgar() correspondiente.
  - Pre condición: -
  - Post condición: depende a las opciones que el usuario elige se pide un char y se llama a arriesgar(char) o se pide un string y se llama a arriesgar(string).
- Operación: finalizar\_juego
  - Descripción: se recibe un char por parámetro, se finaliza el juego.
  - Pre condición: para indicar que gano char == 'G', si perdió el char puede tener cualquier otro valor.
  - Post condición: si el char == 'G' se modifica el valor de cantidad de intentos a 0 y se le indica al usuario que gano. Si char != 'G' se le indica al usuario que perdió.
- Operación: termino
  - Descripción: se verifica si al usuario le quedan intentos y si adivino la palabra (letra x letra).
  - Pre condición: -
  - Post condición: si cantidad de intentos = 0 se llama a finalizar\_juego('P'). Si lo anterior no se cumple, veo si palabra\_a\_adivinar == vector\_con\_guiones y si es así llamo a finalizar\_juego('G'). Si no se cumple nada de lo anterior no se hace nada.
- Operación: imprimir\_vec\_guiones
  - Descripción: imprime por pantalla las letras adivinadas en su posición correspondiente, y las que falta adivinar con guiones

- Pre condición: -
  - Post condición: Imprime por pantalla el contenido de cada posicion de vector\_guiones.
- Operación: imprimir\_vec\_guiones
  - Descripción: imprime por pantalla las letras que el usuario ya arriesgo.
  - Pre condición: -
  - Post condición: Imprime por pantalla el contenido de cada posicion de letras\_arriesgadas.
- Operación: arriesgo\_esa\_letra
  - Descripción: se ve si el char pasado por parámetro ya fue arriesgada por el usuario anteriormente.
  - Pre condición: el char debe estar en mayúscula.
  - Post condición: si el char se encuentra en el vector letras\_arriesgadas se devuelve TRUE, en caso contrario se devuelve FALSE.
- Operación: vector\_a\_str
  - Descripción: se pasa el contenido de un vector a un string. Se recibe un char como parámetro para indicar a que vector se pasa a string. Los vectores pueden ser palabra\_a\_adivinar o vector\_guiones
  - Pre condición: char == 'G' para referirse al vector\_guiones, para referirse al vector palabra\_a\_adivinar el char puede tomar cualquier otro valor.
  - Post condición: devuelve un string con el contenido del vector correspondiente.