

## PROGRAMACIÓN I - GRUPO B

### PRÁCTICA Nº3

16 de noviembre 2023 18:30-21:30

#### INSTRUCCIONES

---

1. Se debe acceder a la **SALA DE GRUPO** del campus virtual de vuestro equipo de trabajo.
2. Durante el desarrollo de la sesión práctica se debe seguir el procedimiento de trabajo en la sala asignada para realizar las actividades evaluables. Recordad que es **imprescindible GRABAR toda la sesión y COMPARTIR la pantalla completa**.
3. La entrega de la práctica sólo se admitirá a través de la **actividad** disponible en el campus virtual de la asignatura de **Prácticas de Programación I antes de la hora de finalización de la sesión de prácticas**.
4. Se debe entregar **un único fichero en formato .txt o .cpp, sin comprimir**, con el nombre de **P3Ejercicio**.
5. El fichero entregado debe **incluir el nombre de los integrantes del equipo**.
6. Aunque las prácticas se realizan en grupos de dos integrantes, para su evaluación, **ambos deberán hacer la entrega a través de su campus virtual**. En otro caso, la práctica quedará **sin evaluar** y supondrá un **0 en su calificación**.
7. **El incumplimiento de alguna de las instrucciones sobre la realización/entrega de la Práctica supondrá su descalificación**.

Al final del enunciado de la práctica tenéis la puntuación de los diferentes hitos que debéis cumplir durante la sesión. Se recomienda que realicéis de forma previa al desarrollo una lectura comprensiva del enunciado, así como un debate en cada equipo con el que podáis planificar la estrategia a seguir durante la práctica.

Dicha estrategia deberá quedar plasmada en un flujograma que deberá ser entregado en papel al finalizar la práctica, indicando en el los nombres de los integrantes del equipo.

**EJERCICIO PRÁCTICO [ 100% puntuación ] – 160minutos**

---

Diseñar y escribir un programa en C++11, **P3Ejercicio**, que permita jugar a una persona al juego del ahorcado con palabras de 5 letras. Dicho programa debe cumplir los siguientes requisitos:

- El desarrollo debe incorporar el mecanismo que consideres oportuno para poder **representar variables de tipo Jugador** en la memoria del computador. Para este ejercicio un jugador consistirá en:
  - Un nombre completo.
  - Una variable vidas que deberá estar inicializado por defecto al valor 5.
  - Una variable partidas ganadas.
  - Una variable partidas perdidas.
  - Una variable de tipo contenedor que pueda almacenar hasta 5 palabras cuya etiqueta sea **PalabrasUsadas**.
- En el desarrollo debes incorporar el mecanismo que consideres oportuno para poder **representar** en la memoria del computador **variables de tipo PalabraEnJuego**. Para este ejercicio dicho tipo de datos consistirá en una agrupación de:
  - Una palabra.
  - Una variable de tipo contenedor de 5 elementos de tipo **bool** inicializados al valor **false** cuya etiqueta sea **Mascara**.
    - Utilizaremos esta variable para controlar que caracteres han sido acertados y cuales están pendientes de averiguar.
- En la **función principal del programa** debes incorporar una **variable contenedor** del tipo que consideres oportuno **que pueda agrupar 9 variables de tipo PalabraEnJuego** cuya etiqueta será **PalabrasSecretas**.
- En la **función principal del programa** debes incorporar una **variable de tipo Jugador** donde poder almacenar la información de la persona que vaya a jugar al ahorcado.

- **Dinámica del juego:**

- Al inicio del juego **el programa solicitará** al jugador y almacenará en el campo correspondiente de la variable ***PalabrasSecretas*** tantas palabras como elementos tenga dicha variable contenedor, comprobando antes del guardado que todas ellas están compuestas por 5 caracteres. En caso de no tener la palabra 5 caracteres el programa la solicitará indefinidamente hasta que el usuario introduzca una palabra válida. En ese momento se almacenará la palabra en el campo adecuado de la variable ***PalabrasSecretas*** hasta completar su tamaño máximo.
- Del mismo modo el programa solicitará el nombre completo del jugador, almacenándolo en el campo correspondiente de la variable de tipo ***Jugador***.
- A continuación, el juego debe implementar un bucle de control en el que mientras el jugador quiera continuar con otra ronda de juego se realizarán las siguientes acciones:
  - Seleccionar de forma aleatoria una de las 9 palabras secretas (dicha palabra será la que deba averiguar el jugador).
  - De manera cíclica mientras que la palabra no haya sido descubierta o el jugador continúen teniendo vidas disponibles:
    - Se visualizará por pantalla tantos guiones bajos como caracteres tenga la palabra secreta en juego sin adivinar aún por el jugador. En el caso de haber caracteres averiguados se visualizarán en la posición correspondiente.
    - El programa preguntará al jugador si quiere intentar adivinar la palabra o si desea probar con una letra.
      - En caso de elegir intento de resolver palabra completa solicitará dicha palabra. En caso de no estar registrada en el histórico ***PalabrasUsadas*** de la variable tipo ***Jugador*** la almacenará en la primera posición libre. Si dicho campo no tuviera ningún elemento libre la nueva palabra no se almacenará:

- En caso de acertar la palabra anotará una partida ganada en el campo correspondiente de la variable tipo **Jugador**, se mostrará por terminal el mensaje “**Palabra adivinada!!!**” y finalizará la ronda.
- En caso de errar la palabra se mostrará por terminal el mensaje “**Error. Palabra incorrecta**” y se restará una vida al jugador y continuará.
- En caso de elegir probar con una letra solicitará la letra al jugador:
  - Si la letra está contenida en la palabra secreta y no ha sido revelada aún, el programa mostrará el mensaje “**Letra acertada!**” y se marcará en el/las variables de tipo **bool** correspondientes a la posición o posiciones de la letra en la palabra con el valor **true**.
  - En el caso de tratarse de la última letra pendiente de averiguar (por tanto, todos los **bool** de la máscara estarán a valor **true**) el programa mostrará por terminal el mensaje “**Palabra correcta!**” y se anotará una partida ganada.
  - Si la letra no pertenece a la palabra secreta o ya ha sido usada en la ronda de juego se mostrará el mensaje “**Error. Letra no adivinada o ya ha sido usada**” y se restará una vida al jugador.
- En el momento que el jugador haya consumido todas sus vidas en una misma ronda se mostrará por pantalla el mensaje “**Has sido ahorcado**” y se anotará una partida perdida

- Al final de cada ronda el programa debe preguntar si se desea seguir jugando o no:
  - En caso afirmativo debe seleccionar de nuevo una palabra de manera aleatoria y repetir la secuencia de juego indicada anteriormente.
  - En caso negativo el programa finalizará, mostrando previamente el resultado de la sesión de juego, y todas las palabras secretas que el usuario había introducido inicialmente.

### RÚBRICA DE CALIFICACIÓN

---

- El 30% de la nota de la práctica, se obtendrá sobre trabajo realizado durante la sesión (resolución de preguntas, dominio de la materia, etc.). Esta nota será individual, y se considerarán para su valoración aspectos como:
  - **Defensa individual** a cuestiones que plantee el profesor durante la sesión a cada integrante del equipo.
  - **Trabajo/debate en equipo** sobre el desarrollo a realizar en la práctica. Quedando representado en un flujograma en el que se visualice el concepto del programa a realizar. **Deberá desarrollarse al inicio de la práctica y se entregará en papel a la finalización de la misma.**
- El 70% restante será sobre la puntuación del trabajo entregado antes de la finalización de cada práctica (misma nota para cada integrante del equipo), siguiendo la siguiente rúbrica general para cada apartado puntuable:

	%máx. (*)
El elemento evaluable no compila o no se asemeja a lo que se pide	0%
El elemento evaluable no se aproxima suficientemente a lo pedido	40%
El elemento evaluable se aproxima suficientemente a lo pedido	60%
El elemento evaluable funciona correctamente y las estrategias y elementos de código elegido son adecuados.	100%

(\*) El porcentaje (% máx.) representa el valor máximo sobre el que se evalúa el elemento indicado.

Puntuación por apartados. El 70% de la calificación de la práctica será obtenida sobre la corrección de la práctica entregada siguiendo el siguiente reparto de puntos para cada apartado:

Escritura correcta del código con las tabulaciones adecuadas, nombres significativos para las variables, comentarios, etc. (Limpieza del trabajo desarrollado)	1 punto
Definición de los tipos de datos <b>Jugador</b> y <b>PalabraEnJuego</b>	0,25 puntos
Declaración variable <b>PalabrasSecretas</b>	0,25 puntos
Petición, validación y almacenamiento del banco de palabras para jugar al ahorcado en la variable <b>PalabrasSecretas</b>	1 punto
Dinámica del juego:	
Bucle para que el jugador pueda jugar tantas partidas como quiera	0,25 puntos
Petición nombre completo al jugador	0,25 puntos
Selección aleatoria de una palabra	0,25 puntos
Bucle para gestionar las acciones de cada ronda	0,25 puntos
Visualización caracteres o guiones bajos según el estado de la palabra en juego	1 punto
Gestión del intento averiguar palabra	2 puntos
Gestión del intento encontrar caracteres	2 puntos
Gestión vidas y anotación partidas ganadas y perdidas	1 punto
Visualización de los resultados obtenidos por el jugador y de las palabras secretas antes de terminar el programa	0,5 puntos