

LABORATORIO N°2

Programación en C: Utilización de arreglo de caracteres y diseño modular.

Fecha de envío: Domingo 20 de octubre, 23:55 hrs. (vía aula virtual)

Modalidad: Trabajo en grupos de a lo más **dos** personas

I. Objetivos.

El objetivo del presente laboratorio es evaluar su capacidad para:

1. Llevar a cabo un programa completo en el lenguaje de programación C, utilizando todos los elementos básicos que provee el lenguaje, y sus tipos de datos básicos.
2. Aplicar las buenas prácticas en programación (orden, comentarios, identificadores representativos).
3. Resolver un problema, considerando que se cuenta con el procedimiento en lenguaje natural, que se debe implementar para resolverlos.
4. Hacer un correcto diseño modular para la resolución de un problema dado.

II. Enunciado.

Debe crear un programa que le permita al usuario jugar al AHORCADO, con la interfaz presentada en la Figura 1.

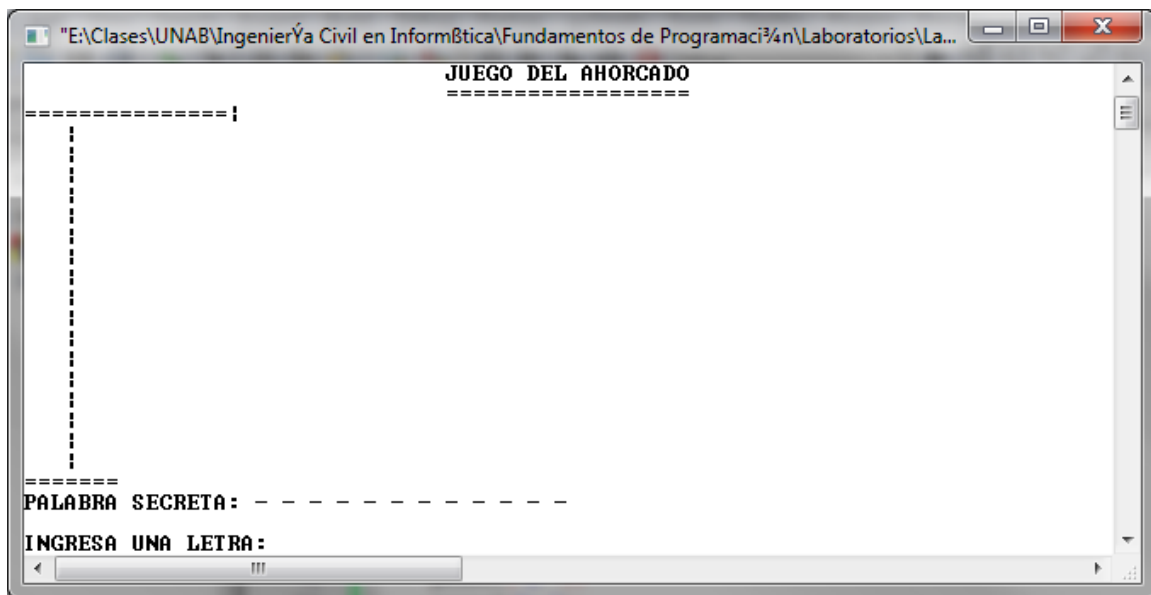


Figura 1: Interfaz principal

Si el usuario digita una letra que **pertenece** a la palabra a adivinar, debe aparecer tal letra en cada ocurrencia dentro de la palabra. En la Figura 2 se muestra un ejemplo para la palabra secreta “ornitorrinco”, cuando el usuario digita una “r”.

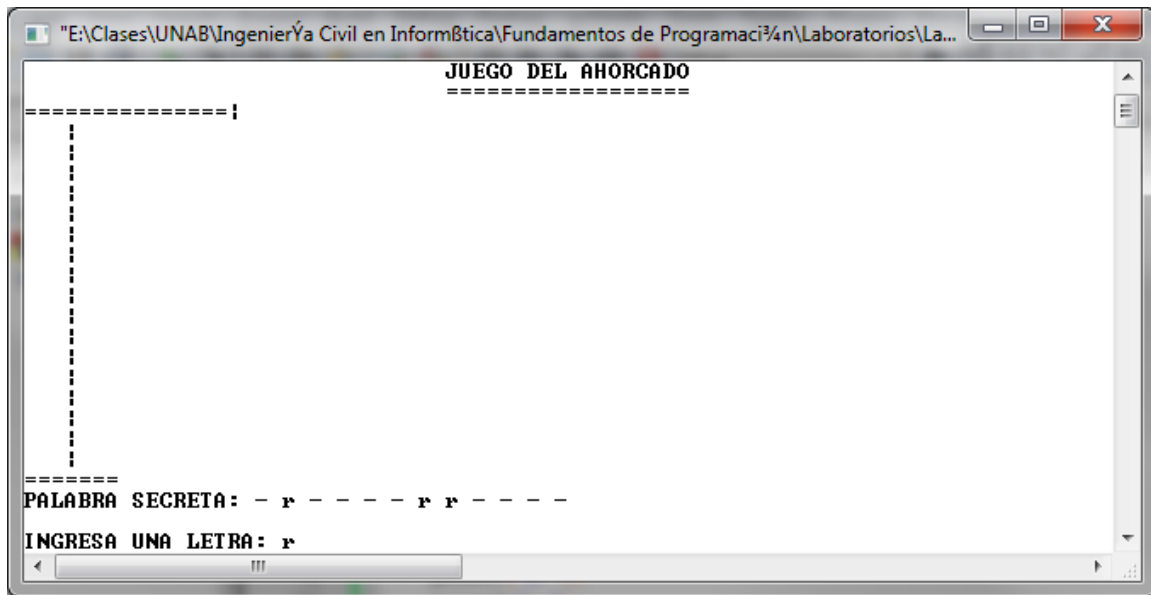


Figura 2: Interfaz cuando el usuario acierta en una letra de la palabra secreta.

Si el usuario digita una letra que **no pertenece** a la palabra a adivinar, debe aparecer una parte del “ahorcado” en la interfaz. En la Figura 3 se muestra un ejemplo para la palabra secreta “ornitorrinco”, cuando el usuario digita una “x”.

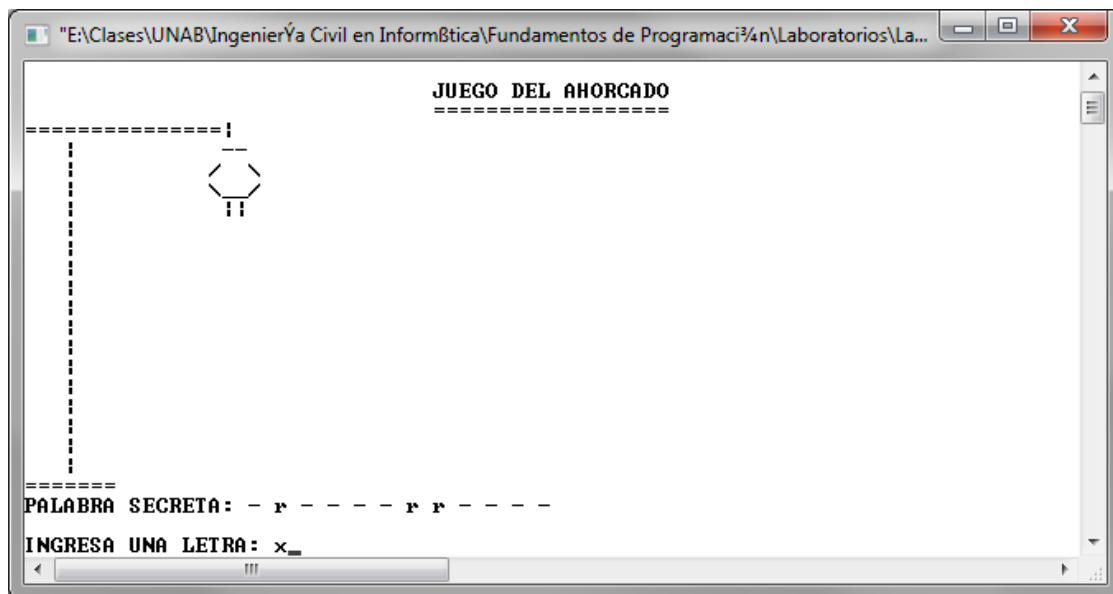


Figura 3: Interfaz cuando el usuario erra en una letra de la palabra secreta.

El usuario tiene 6 posibilidades de equivocarse. Por cada una de ellas, debe aparecer una parte del ahorcado. En la Figura 4 aparecen las 6 partes en las que se debe dividir el ahorcado.

La secuencia de aparición es:

- 1° Cabeza.
- 2° Tronco.
- 3° Brazo Izquierdo
- 4° Brazo Derecho.
- 5° Pierna Izquierda.
- 6° Pierna Derecha.

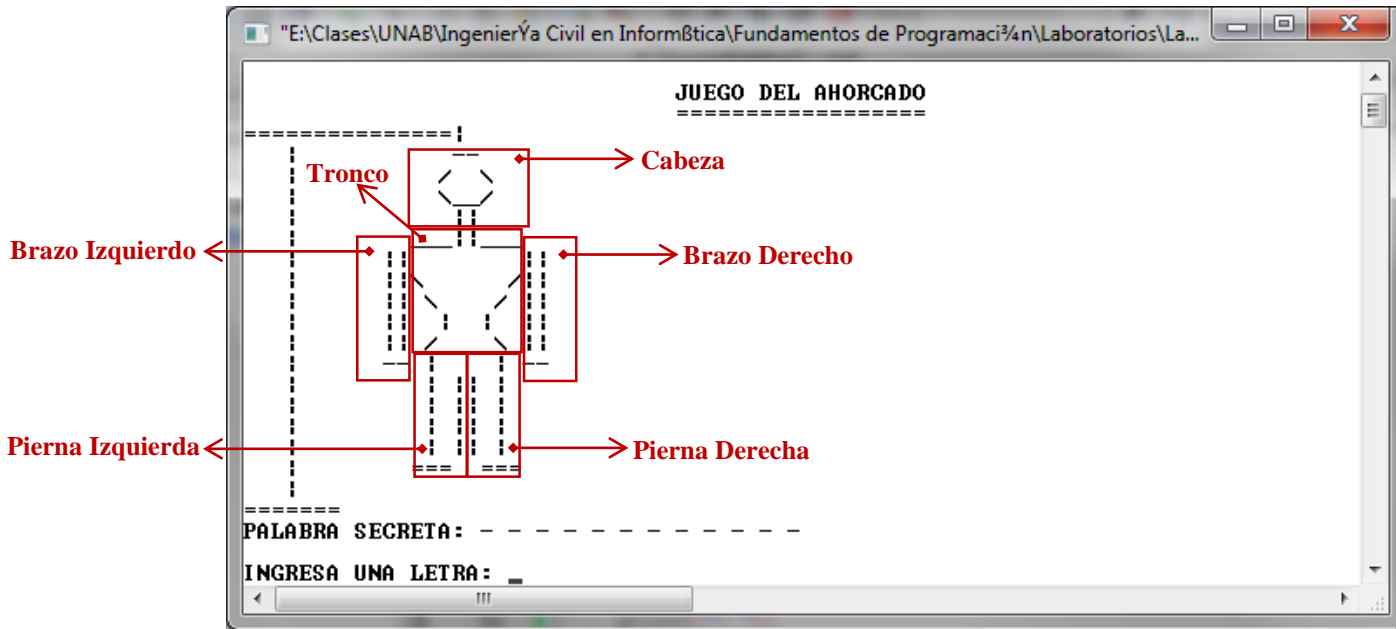


Figura 4: Partes del ahorcado.

Si el usuario adivina la palabra, antes que se dibuje el ahorcado completamente, debe aparecer el aviso de que ganó, y la palabra que adivinó. En la Figura 5 se muestra un ejemplo para la palabra “**ornitorrinco**”.

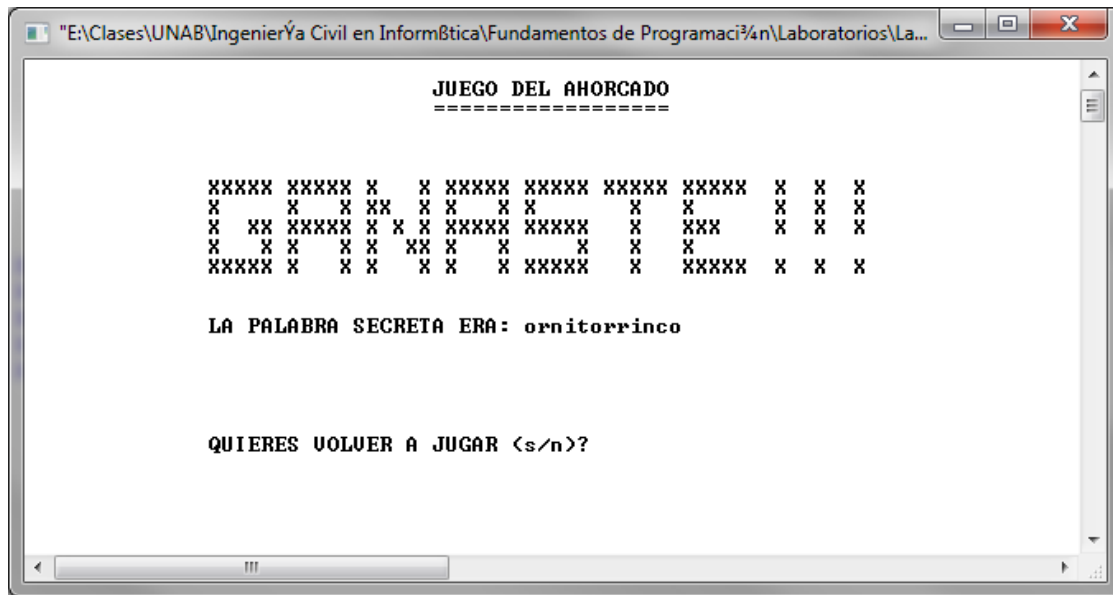


Figura 5: Interfaz cuando el usuario adivina la palabra secreta.

Si el usuario **no adivina** la palabra, y se completan los intentos, debe aparecer el aviso de que perdió, y la palabra que no logró adivinar. En la Figura 6 se muestra un ejemplo para la palabra “ornitorrinco”.

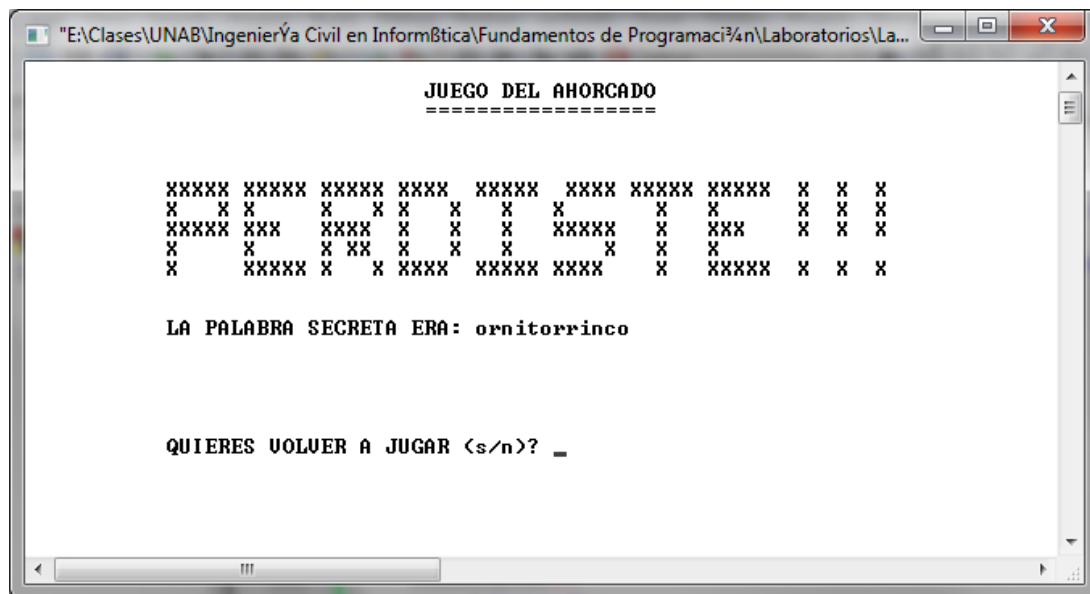


Figura 6: Interfaz cuando el usuario no adivina la palabra secreta.

Ya sea que pierda o gane, el programa le debe mostrar la opción al usuario de volver a jugar.

II.1 Consideraciones en la revisión.

Validaciones e interfaz con el usuario:

1. Su programa debe ser capaz de reconocer la letra ingresada por el usuario, independiente si digitó una mayúscula o una minúscula.
2. Su programa debe ser capaz de reconocer la vocal ingresada por el usuario, independiente si lo digitó con algún tipo de tilde o no.
3. Su programa debe ser capaz de distinguir letras de otros caracteres: si el usuario no ingresa una letra, el programa no debe ocupar uno de los 6 intentos del juego.

Sobre el código:

1. El programa debe poseer 20 palabras distintas escogidas por usted, para que el usuario pueda jugar.
2. Si el usuario decide volver a jugar, su programa debe ser capaz de escoger una palabra que no haya sido jugada antes en esa partida.
3. Sólo si el usuario ya jugó las 20 palabras, su programa puede volver a escoger una palabra ya jugada.
4. Su programa debe ser capaz de no castigar al usuario si vuelve a ingresar una letra incorrecta, que ya hubiese ingresado en la partida en curso.
5. Cada vez que el usuario vuelva a ingresar una letra, usted debe “limpiar” la pantalla y volver a escribir desde la primera posición, en la primera línea.
6. Debe usar identificadores representativos para sus variables y constantes.
7. Debe comentar cada una de las funciones que posea su programa, indicando para ellas:
 - Parámetros de entrada.
 - Salida.
 - Descripción.
8. Su código debe estar correctamente *indentado* (uso de sangrías para cada sub-bloque de instrucciones).

Sobre el Manual de Usuario: La interfaz del programa solicitado está totalmente establecida por lo que **no se pedirá Manual de Usuario**. Esto quiere decir que usted **debe cumplir al pie de la letra con la interfaz declarada en este enunciado, sino tendrá descuentos en su nota**.

IMPORTANTE: Archivos que se podrán a su disposición:

Para facilitarles el trabajo de la interfaz, se pondrá a su disposición junto a este enunciado el archivo **dibujos.h**. Este archivo contiene las diferentes funciones para “dibujar” las diferentes

interfaces con el usuario. Usted solo deberá usarlas en su programa. Las funciones que posee este archivo son:

- TituloJuego();
- SinAhorcado();
- Cabeza();
- Tronco();
- BrazoIzquierdo();
- BrazoDerecho();
- PiernaIzquierda();
- PiernaDerecha();
- MensajeGanaste();
- MensajePerdiste();
- LimpiaPantalla();

II.2 Sobre la entrega, atrasos y faltas a la ética.

1. Debe subir su trabajo a la plataforma de unabvirtual.cl, en una casilla que se habilitará especialmente para esto.
2. Su trabajo debe ser subido al aula virtual por **SOLO UNO de los integrantes del grupo**.
3. Debe subir un archivo comprimido (.rar o .zip) que incluya todos los códigos fuente de su programa.

El nombre de su archivo debe ser el nombre y apellido paterno de cada integrante del grupo. **Ejemplo:** Si trabajaron juntos los alumnos Rosa González y Santiago Fuenzalida, el archivo se debe llamar: **RosaGonzalezSantiagoFuenzalida.zip** (o **.rar**)

4. Ante la ausencia de alguno de los archivos mencionados en el punto anterior, la nota del trabajo completo será un **1.0**.
5. Si el programa no se puede ejecutar, tendrá la nota mínima: **1.0**. Si su programa funciona, se evaluará su ejecución, y también el código fuente.
6. Ante el escenario de existir sospecha de copia (con otros compañeros, o desde internet), será interrogado acerca de su trabajo, para aclarar dudas de su entendimiento y autoría. Si se confirma la sospecha, el trabajo será evaluado con nota **1.0**.
7. Las consultas las debe realizar directamente a los ayudantes, de lunes a viernes a los correos y horarios que ellos establezcan.