



INFORME JUEGO CATEGORIZADO

ALGORITMO Y ESTRUCTURA DE DATOS



1 DE AGOSTO DE 2020

GRUPO N°3

JACAS LEILA, DAVID

|

JUAREZ, PATRICIO LUCAS

INTRODUCCION

En convenio con el grupo realizamos la elección como temática el desarrollo de los juegos categorizados debido a que consideramos que la opción era la más compleja y representaba un reto para nosotros y escapaba de los trabajos que realizábamos hasta el momento.

Nuestro propósito fue siempre y en todo momento aprender y hacer un trabajo lo más efectivo y que englobe distintas posiciones con respecto a las elecciones del usuario.

Trabajamos en un total de 5 fases prácticamente reciprocas:

1. **Análisis:** En esta etapa realizamos la lectura y posterior desarrollo en un borrador sobre como realizaríamos los juegos y sus respectivas estructuras.
2. **Desarrollo de librerías:** Escritura de las librerías de MayMen, Adivinium, WarCoin y Ahorcado (Este apartado se tratará más a profundidad posteriormente)
3. **Investigación:** Realizamos reiteradas búsquedas en internet sobre nuevas librerías y funciones que no habíamos utilizado anteriormente, pero que hacían que el programa funcionara de mejor manera.
4. **Desarrollo del menú:** Probablemente el apartado más fácil fue la creación de un menú de selección de los juegos.
5. **Testing:** Siendo esta la ultima etapa, realizamos distintas pruebas al programa para someterlo a brutas introducciones de datos intentando que este se rompa, lográndolo muchas veces y siempre con un posterior arreglo de este.

A pesar de que nuestro grupo fue más extenso en algún momento, tuvimos algunas discrepancias y únicamente terminamos siendo participes Lucas Juárez y David Leila. Aun así, el trabajo se hizo más fácil a la hora de ponerse de acuerdo en las estructuras, pero más lento al momento de hacer cambios o escribir partes extensas.

Pensamos que no existen roles distinguidos ya que al trabajo lo hicimos en conjunto en todo momento.

INFORME

El programa sigue la siguiente estructura definida:

- Menú principal
 - MayMen
 - Adivinuum
 - WarCoin
 - Ahorcado
 - Librería del Ahorcado

En el menú principal utilizamos una estructura de ciclo << Do While>> para ejecutar el programa mientras el usuario lo requiera. Luego con 4 estructuras de selección <<If>>, ejecutamos los respectivos main de los juegos dentro del archivo <<main.cpp>>, y además preguntamos si el usuario desea leer las instrucciones del respectivo juego.

```
=====
=====| BIENVENIDO AL MENU DEL JUEGO |=====
=====| ALT + F4 PARA CERRAR |=====
=====

=====
=====| ELIGE UN JUEGO |=====
=====| 1- MAYMEN |=====
=====| 2- ADIVINUUM |=====
=====| 3- WARCOIN |=====
=====| 4- AHORCADO |=====
=====
=====|> █
```

A partir de acá se empiezan a utilizar librerías. Cada juego es una librería y en el caso de el Ahorcado, parte de la estructura está en otra librería.

Algunas librerías utilizadas y su explicación corta:

string.h

Esta librería nos provee algunas funciones útiles para El Ahorcado ya que nos permite entre algunas cosas, comparar dos variables de tipo string o mostrar la cantidad de caracteres.

iostream

Esta librería es parecida a stdio.h en c, pero se utiliza en el lenguaje C++. La utilizamos debido a que nos proveía facilidad con la función Cin para leer variables de tipo string.

windows.h

Esta librería nos provee algunas funciones como setlocale que sirve para cambiar el idioma a español para poder incluir acentos y "ñ", o poner pantalla completa y cambiar de color.

Fuentes y Recursos

Para aprender mas sobre el tema recurrimos reiteradas veces a internet a fuentes como StackOverflow, GitHub, u otros foros de tipo libre.

Con respecto al trabajo a distancia utilizamos software libre para comunicarnos. Por ejemplo, hicimos uso de la plataforma Discord y un canal creado exclusivamente para el trabajo, como forma de comunicación de voz. Por otro lado, para utilizar la pantalla simultáneamente utilizamos AnyDesk y Microsoft Office Word para escribir el informe (Licencia proveída por la facultad).

Para incluir archivos .gif en el Informe, utilizamos un programa llamado Screen-To-Gif de uso gratuito y código abierto.

CASOS DE PRUEBA

MayMen

⊕ CASO 1:

Nro	Nombre/Identificador	Precondiciones	Entradas	Pasos	Resultados Esperados	Pos. Condiciones
MAYMEN						
1	CasoPrueba1Maymen	Debe estar creado el archivo main.exe con su código ya compilado y sin errores.	Número por probar: 300, 500, 700, 650, 550 580, 610, 603, 597, 600	1. Ejecutar el archivo .exe. 2. Ingresar los datos de entrada.	Mostrar correctamente el rango entre un número y el otro para encontrar el número secreto.	Se mantiene correctamente el rango, dándonos a entender donde se encuentra el número secreto.
2	CasoPrueba2Maymen	Ejecutamos el archivo .exe del respectivo programa.	Número por probar: 10, 30, 40, 200, 210	1. Ejecutamos el archivo .exe. 2. Ingresamos los datos de entrada correspondientes 3. Buscamos acertar al número secreto.	Mostrar correctamente El puntaje al ganar	Puntaje obtenido acorde al resultado.
3	CasoPrueba3Maymen	Ejecutamos el archivo .exe del respectivo Programa.	Numero por probar: 10, 20, 30, 40, 50, 60, 70, 80, 90,100	1. Ejecutamos el archivo .exe. 2. Ingresamos los datos de entrada correspondientes 3. Buscamos perder en el juego.	Mostrar correctamente al momento de perder su puntaje, además mostrar cual fue el número secreto.	Puntaje obtenido en la perdida obtenido correctamente, El número secreto mostrado coincide con los rangos mostrados anteriormente.

CasoPrueba1MayMen

```
=====
=====| Intento 1 |=====
=====

300
El numero secreto es mayor y se encuentra entre 301 y 999
=====
=====| Intento 2 |=====
=====

500
El numero secreto es mayor y se encuentra entre 501 y 999
=====
=====| Intento 3 |=====
=====

700
El numero secreto es menor y se encuentra entre 501 y 699
=====
=====| Intento 4 |=====
=====

650
El numero secreto es menor y se encuentra entre 501 y 649
=====
=====| Intento 5 |=====
=====

550
El numero secreto es mayor y se encuentra entre 551 y 649
=====
=====| Intento 6 |=====
=====
```

580

El numero secreto es mayor y se encuentra entre 581 y 649

=====

=====| Intento 7 |=====

=====

610

El numero secreto es menor y se encuentra entre 581 y 609

=====

=====| Intento 8 |=====

=====

603

El numero secreto es menor y se encuentra entre 581 y 602

=====

=====| Intento 9 |=====

=====

597

El numero secreto es mayor y se encuentra entre 598 y 602

=====

=====| Intento 10 |=====

=====

600

=====

=====Acertaste! El puntaje obtenido es 1 =====

=====

CasoPrueba2MayMen

```
=====
=====| Intento 1 |=====
=====

10
El numero secreto es mayor y se encuentra entre 11 y 999
=====
=====| Intento 2 |=====
=====

30
El numero secreto es mayor y se encuentra entre 31 y 999
=====
=====| Intento 3 |=====
=====

40
El numero secreto es mayor y se encuentra entre 41 y 999
=====
=====| Intento 4 |=====
=====

200
El numero secreto es mayor y se encuentra entre 201 y 999
=====
=====| Intento 5 |=====
=====

210
=====
=====Acertaste! El puntaje obtenido es 6 =====
=====
```


CasoPrueba3MayMen

```
=====
=====| Intento 1 |=====
=====
```

```
10
El numero secreto es mayor y se encuentra entre 11 y 999
```

```
=====
=====| Intento 2 |=====
=====
```

```
20
El numero secreto es mayor y se encuentra entre 21 y 999
```

```
=====
=====| Intento 3 |=====
=====
```

```
30
El numero secreto es mayor y se encuentra entre 31 y 999
```

```
=====
=====| Intento 4 |=====
=====
```

```
40
El numero secreto es mayor y se encuentra entre 41 y 999
```

```
=====
=====| Intento 5 |=====
=====
```

```
50
El numero secreto es mayor y se encuentra entre 51 y 999
```

```
=====
=====| Intento 6 |=====
=====
```

60

El numero secreto es mayor y se encuentra entre 61 y 999

=====

=====| Intento 7 |=====

=====

70

El numero secreto es mayor y se encuentra entre 71 y 999

=====

=====| Intento 8 |=====

=====

80

El numero secreto es mayor y se encuentra entre 81 y 999

=====

=====| Intento 9 |=====

=====

90

El numero secreto es mayor y se encuentra entre 91 y 999

=====

=====| Intento 10 |=====

=====

100

El numero secreto es mayor y se encuentra entre 101 y 999

=====

=====No acertaste! El puntaje obtenido es: 0=====

=====El numero secreto fue 210 =====

CASOS DE PRUEBA

ADIVINUM

CASO 1:

Nro.	Nombre/Identificador	Precondiciones	Entradas	Pasos	Resultados Esperados	Pos. Condiciones
ADIVINUM						
1	CasoPrueba1Adivinum	Debe estar creado el archivo main.exe con su código ya compilado y sin errores.	Número por probar: 2424-3438-5825-2342-3482	1. Ejecutar el archivo .exe. 2. Ingresar los datos de entrada. 3.Comprobamos los mensajes que nos devuelven.	Mostrar que ninguna cifra debe ser igual al ingresar un numero de 4 dígitos.	Se mantiene Correctamente el funcionamiento al realizar la prueba de números con dígitos iguales, sin sumar ningún intento al realizar la prueba de estos.
2	CasoPrueba2Adivinum	Ejecutamos el archivo .exe del respectivo programa.	Numero por probar: 14-23-4-3-4	1.Ejecutar el archivo .exe 2.Ingresar los datos de entrada. 3. Comprobamos que el mensaje sea acorde al valor de entrada ingresado.	Mostrar que el numero debe ser de 4 cifras cuando se ingresa un número menor a lo indicado.	El funcionamiento cumple con los datos ingresados.
3	CasoPrueba3Adivinum	Modificamos el archivo para que el número secreto sea 3456, compilamos y Ejecutamos el archivo .exe del respectivo programa.	Numero por probar: 3489-8956-5643-3456	1. Ejecutamos el archivo .exe. 2. Ingresamos los datos de entrada teniendo en cuenta que el número secreto ya lo sabemos de antemano. 3. Comprobamos el funcionamiento de la posición del número.	Mostrar la cantidad de números iguales Mostrar la cantidad de números en la misma posición. Mostrar la cantidad de números que coinciden en distintas posiciones.	El funcionamiento es correcto, al realizar las 3 pruebas. Además, el puntaje al ganar es adecuado al número de intentos.

CasoPrueba1Adivinum

C:\Users\lucas\OneDrive\Escritorio\Adivinum.exe

```
=====
=====| Intento 1 |=====
=====
```

14

El numero debe tener 4 cifras. Vuelva a intentarlo.

```
=====
=====| Intento 1 |=====
=====
```

23

El numero debe tener 4 cifras. Vuelva a intentarlo.

```
=====
=====| Intento 1 |=====
=====
```

4

El numero debe tener 4 cifras. Vuelva a intentarlo.

```
=====
=====| Intento 1 |=====
=====
```

3

El numero debe tener 4 cifras. Vuelva a intentarlo.

```
=====
=====| Intento 1 |=====
=====
```

4

El numero debe tener 4 cifras. Vuelva a intentarlo.

```
=====
```

INTENTO 1:

C:\Users\lucas\OneDrive\Escritorio\Adivinum.exe

```
=====
=====| Intento 1 |=====
=====
```

2424

Ninguna cifra debe ser igual.

```
=====
=====| Intento 1 |=====
=====
```

3438

Ninguna cifra debe ser igual.

```
=====
=====| Intento 1 |=====
=====
```

5825

Ninguna cifra debe ser igual.

```
=====
=====| Intento 1 |=====
=====
```

2342

Ninguna cifra debe ser igual.

```
=====
=====| Intento 1 |=====
=====
```

INTENTO 2

C:\Users\lucas\OneDrive\Escritorio\Adivinum.exe

=====

3438

Ninguna cifra debe ser igual.

=====

=====| Intento 1 |=====

=====

5825

Ninguna cifra debe ser igual.

=====

=====| Intento 1 |=====

=====

2342

Ninguna cifra debe ser igual.

=====

=====| Intento 1 |=====

=====

3482

Cant. Misma Posicion: 0 - Cant. Otra Posicion: 4

=====

CasoPrueba2Adivinum

C:\Users\lucas\OneDrive\Escritorio\Adivinum.exe

```
=====
=====| Intento 1 |=====
=====
```

14

El numero debe tener 4 cifras. Vuelva a intentarlo.

```
=====
=====| Intento 1 |=====
=====
```

23

El numero debe tener 4 cifras. Vuelva a intentarlo.

```
=====
=====| Intento 1 |=====
=====
```

4

El numero debe tener 4 cifras. Vuelva a intentarlo.

```
=====
=====| Intento 1 |=====
=====
```

3

El numero debe tener 4 cifras. Vuelva a intentarlo.

```
=====
=====| Intento 1 |=====
=====
```

4

El numero debe tener 4 cifras. Vuelva a intentarlo.

```
=====
```

CasoPrueba3Adivinum

INTENTO 1

C:\Users\lucas\OneDrive\Escritorio\Adivinum.exe

```
=====
=====| Intento 1 |=====
=====
```

3489

Cant. Misma Posicion: 2 - Cant. Otra Posicion: 2

```
=====
=====| Intento 2 |=====
=====
```

INTENTO 2

C:\Users\lucas\OneDrive\Escritorio\Adivinum.exe

```
=====
=====| Intento 1 |=====
=====
```

3489

Cant. Misma Posicion: 2 - Cant. Otra Posicion: 2

```
=====
=====| Intento 2 |=====
=====
```

8956

Cant. Misma Posicion: 2 - Cant. Otra Posicion: 2

```
=====
=====| Intento 3 |=====
=====
```


INTENTO 3

C:\Users\lucas\OneDrive\Escritorio\Adivinum.exe

```
=====
=====| Intento 1 |=====
=====
```

3489

Cant. Misma Posicion: 2 - Cant. Otra Posicion: 2

```
=====
=====| Intento 2 |=====
=====
```

8956

Cant. Misma Posicion: 2 - Cant. Otra Posicion: 2

```
=====
=====| Intento 3 |=====
=====
```

5643

Cant. Misma Posicion: 0 - Cant. Otra Posicion: 4

```
=====
=====| Intento 4 |=====
=====
```

■

INTENTO 4

C:\Users\lucas\OneDrive\Escritorio\Adivinum.exe

```
=====
=====| Intento 1 |=====
=====
```

4389

Cant. Misma Posicion: 0 - Cant. Otra Posicion: 4

```
=====
=====| Intento 2 |=====
=====
```

8956

Cant. Misma Posicion: 2 - Cant. Otra Posicion: 2

```
=====
=====| Intento 3 |=====
=====
```

5643

Cant. Misma Posicion: 0 - Cant. Otra Posicion: 4

```
=====
=====| Intento 4 |=====
=====
```

3456

```
=====
=====Felicitaciones! Acertaste el numero. Puntaje obtenido: 6 =====
=====
```

Process exited after 8.428 seconds with return value 0

Presione una tecla para continuar . . . ■

CASOS DE PRUEBA

WARCOIN

CASO 1:

Nro.	Nombre/Identificador	Precondiciones	Entradas	Pasos	Resultados Esperados	Pos. Condiciones
WARCOIN						
1	CasoPrueba1WarCoin	Debe estar creado el archivo main.exe con su código ya compilado y sin errores.	Número por probar: 30-20	1. Ejecutar el archivo .exe. 2. Ingresar los datos de entrada. 3. Verificamos el resultado al ingresar los valores de entrada.	Marcar el error al ingresar los datos erróneos.	Se mantiene correctamente el rango de entrada de monedas menor y mayor. Además, cumple la función de volver a pedir los números a ingresar correctamente.
2	CasoPrueba2WarCoin	Ejecutamos el archivo .exe del respectivo programa.	Números por probar: 10(menor)-15(mayor). 10	1. Ejecutar el archivo .exe. 2. Ingresar los datos de entrada. 3. Jugamos una partida para comprobar el funcionamiento de este.	Marca del puntaje correctamente, Descuentos de monedas, descontados adecuadamente.	Comprobamos que al ganar la CPU el puntaje mostrado es 0. El descuento de monedas es acorde al rango ingresado.
3	CasoPrueba3WarCoin	Ejecutamos el archivo .exe del respectivo programa 3 veces.	Números por probar: 10(menor)-15(mayor) 1er intento 8(menor)-12(mayor) 2do intento 5(menor)-10(mayor). 3er intento.	1. Ejecutar el archivo .exe. 2. Comprobar el funcionamiento de la función usuario computadora. 3. Cerra y comprobar nuevamente.	Funcionamiento de la función usuario-computadora de manera aleatoria.	Comprobamos que el juego nos muestra correctamente de manera aleatoria el comienzo ya sea CPU o Jugador. Se muestra correctamente el intento

CasoPrueba1WarCoin

C:\Users\lucas\OneDrive\Escritorio\Warcoin.exe

```
=====
=====INGRESE LA CANTIDAD MINIMA DE MONEDAS QUE SE PUEDEN RETIRAR=====
=====
```

30

```
=====
=====INGRESE LA CANTIDAD MAXIMA DE MONEDAS QUE SE PUEDEN RETIRAR=====
=====
```

20

El valor Maximo no puede ser Menor al valor Menor...

Presione una tecla para continuar . . . ■

CasoPrueba2WarCoin

```
C:\Users\lucas\OneDrive\Escritorio\Warcoin.exe
=====
=====INGRESE LA CANTIDAD MINIMA DE MONEDAS QUE SE PUEDEN RETIRAR=====
=====
10
=====
=====INGRESE LA CANTIDAD MAXIMA DE MONEDAS QUE SE PUEDEN RETIRAR=====
=====
15
=====
===== |Es el turno del: Usuario |=====
=====
=====
=====|LA CANTIDAD INICIAL DE MONEDAS EN LA PILA ES: 22 |=====
=====
=====
=====| INGRESE LA CANTIDAD DE MONEDAS A ELIMINAR |=====
=====
10
=====
==> Quedan 12 monedas
==> La computadora saco 10 monedas de la pila
==> Quedan 0 monedas
=====
=====| EL CPU GANO!! PUNTAJE OBTENIDO: 0 |=====
=====
-----
Process exited after 13.35 seconds with return value 0
Presione una tecla para continuar . . .
```

CasoPrueba3WarCoin

INTENTO 1

C:\Users\lucas\OneDrive\Escritorio\Warcoin.exe

```
=====
=====INGRESE LA CANTIDAD MINIMA DE MONEDAS QUE SE PUEDEN RETIRAR=====
=====
10
=====
=====INGRESE LA CANTIDAD MAXIMA DE MONEDAS QUE SE PUEDEN RETIRAR=====
=====
15
=====
===== |Es el turno del: Usuario |=====
=====
=====|LA CANTIDAD INICIAL DE MONEDAS EN LA PILA ES: 22 |=====
=====
=====| INGRESE LA CANTIDAD DE MONEDAS A ELIMINAR |=====
=====
10

==> Quedan 12 monedas

==> La computadora saco 10 monedas de la pila

==> Quedan 0 monedas

=====
=====| EL CPU GANO!! PUNTAJE OBTENIDO: 0 |=====
=====

-----
Process exited after 13.35 seconds with return value 0
Presione una tecla para continuar . . .
```

CasoPrueba3WarCoin

INTENTO 2

C:\Users\lucas\OneDrive\Escritorio\Warcoin.exe

```
=====
=====INGRESE LA CANTIDAD MINIMA DE MONEDAS QUE SE PUEDEN RETIRAR=====
=====
```

8

```
=====
=====INGRESE LA CANTIDAD MAXIMA DE MONEDAS QUE SE PUEDEN RETIRAR=====
=====
```

12

```
=====
=====| Es el turno de: CPU |=====
=====
```

```
=====
=====|LA CANTIDAD INICIAL DE MONEDAS EN LA PILA ES: 23 |=====
=====
```

=> La computadora saco 11 monedas de la pila

=> Quedan 12 monedas

Ingresar la cantidad de monedas a eliminar:

CasoPrueba3WarCoin

INTENTO 3

C:\Users\lucas\OneDrive\Escritorio\Warcoin.exe

```
=====
=====INGRESE LA CANTIDAD MINIMA DE MONEDAS QUE SE PUEDEN RETIRAR=====
=====
```

5

```
=====
=====INGRESE LA CANTIDAD MAXIMA DE MONEDAS QUE SE PUEDEN RETIRAR=====
=====
```

10

```
=====
===== |Es el turno del: Usuario |=====
=====
```

```
=====
=====|LA CANTIDAD INICIAL DE MONEDAS EN LA PILA ES: 50 |=====
=====
```

```
=====
=====| INGRESE LA CANTIDAD DE MONEDAS A ELIMINAR |=====
=====
```


CASOS DE PRUEBA

AHORCADO

CASO 1:

Nro	Nombre/Identificador	Precondiciones	Entradas	Pasos	Resultados Esperados	Pos. Condiciones
AHORCADO						
1	CasoPrueba1Ahorcado	Debe estar creado el archivo main.exe con su código ya compilado y sin errores.	Cant de Jugadores: Op1 Palabra Elegida: 13 Letra: A, E, I, O, U, W, C, H, L, R	1. Ejecutar el archivo .exe. 2. Ingresar los datos de entrada.	Mostrar correctamente las partes del cuerpo y el número de intento	Se muestra correctamente el intento
2	CasoPrueba2Ahorcado	Debe estar creado el archivo main.exe con su código ya compilado y sin errores.	Cant de Jugadores: Op0 Letra: A, E, I, O, U, D, C, J, W, H	1. Ejecutar el archivo .exe. 2. Ingresar los datos de entrada.	Debería perder	Muestra correctamente que hemos perdido y nos da 0 puntos
3	CasoPrueba3Ahorcado	Debe estar creado el archivo main.exe con su código ya compilado y sin errores.	Cant de Jugadores: Op4	1. Ejecutar el archivo .exe. 2. Ingresar los datos de entrada.	Error	Debe ofrecernos instrucciones

CasoPrueba1Ahorcado

INTENTO 1

```
Palabra a adivinar: _____

=====
=====JUGADA #1=====
=====

=====
=====Ingrese una letra=====
=====
=====> a

Palabra a adivinar: __A_

=====
=====JUGADA #2=====
=====

=====
=====Ingrese una letra=====
=====
=====> e

Estado del muñeco: cabeza -
Palabra a adivinar: __A_

=====
=====JUGADA #3=====
=====

=====
=====Ingrese una letra=====
=====
=====> i

Estado del muñeco: cabeza - tronco
Palabra a adivinar: __A_

=====
=====JUGADA #4=====
=====
```

CasoPrueba1Ahorcado

INTENTO 2

Estado del muñeco: cabeza - tronco - brazo derecho

Palabra a adivinar: __A_

=====
=====JUGADA #5=====

=====
=====Ingrese una letra=====

=====> u

Estado del muñeco: cabeza - tronco - brazo derecho - brazo izquierdo

Palabra a adivinar: __A_

=====
=====JUGADA #6=====

=====
=====Ingrese una letra=====

=====> w

Estado del muñeco: cabeza - tronco - brazo derecho - brazo izquierdo - pierna derecha

Palabra a adivinar: __A_

=====
=====JUGADA #7=====

=====
=====Ingrese una letra=====

=====> c

CasoPrueba1Ahorcado

INTENTO 3

```
Estado del muñeco: cabeza - tronco - brazo derecho - brazo izquierdo - pierna derecha - pierna izquierda
Palabra a adivinar: CHA_

=====
=====JUGADA #10=====
=====

=====
=====Ingrese una letra=====
=====
======> r

Palabra a adivinar: CHAR
Ganaste!!! Tu puntaje es 38
Presione una tecla para continuar . . . █
```

INTENTO 4

```
Palabra a adivinar: C_A_

=====
=====JUGADA #8=====
=====

=====
=====Ingrese una letra=====
=====
======> h

Palabra a adivinar: CHA_

=====
=====JUGADA #9=====
=====

=====
=====Ingrese una letra=====
=====
======> l

Estado del muñeco: cabeza - tronco - brazo derecho - brazo izquierdo - pierna derecha - pierna izquierda
Palabra a adivinar: CHA_

=====
=====JUGADA #10=====
=====

=====
=====Ingrese una letra=====
=====
======>
```

CasoPrueba2Ahorcado

INTENTO 1

```
=====
=====Ingrese una letra=====
=====
=====> c

Palabra a adivinar: CO_I_A_O_

=====
=====JUGADA #7=====
=====

=====
=====Ingrese una letra=====
=====
=====> h

Estado del muñeco: cabeza - tronco - brazo derecho - brazo izquierdo
Palabra a adivinar: CO_I_A_O_

=====
=====JUGADA #8=====
=====

=====
=====Ingrese una letra=====
=====
=====> l

Palabra a adivinar: CO_ILA_O_

=====
=====JUGADA #8=====
=====

=====
=====Ingrese una letra=====
=====
=====> r

Palabra a adivinar: CO_ILA_OR
```

CasoPrueba2Ahorcado

INTENTO 2

```
Palabra a adivinar: _____
=====
=====JUGADA #1=====
=====
=====Ingrese una letra=====
=====> a

Palabra a adivinar: _____A_____
=====
=====JUGADA #1=====
=====
=====Ingrese una letra=====
=====> e

Estado del muñeco: cabeza -
Palabra a adivinar: _____A_____
=====
=====JUGADA #2=====
=====
=====Ingrese una letra=====
=====> i

Palabra a adivinar: _____I_A_____
=====
=====JUGADA #3=====
=====
=====
```


INTENTO 3

```
=====
=====Ingrese una letra=====
=====> o

Palabra a adivinar: _O_I_A_O_
=====
=====JUGADA #4=====
=====

=====Ingrese una letra=====
=====> u

Estado del muñeco: cabeza - tronco
Palabra a adivinar: _O_I_A_O_
=====
=====JUGADA #5=====
=====

=====Ingrese una letra=====
=====> w

Estado del muñeco: cabeza - tronco - brazo derecho
Palabra a adivinar: _O_I_A_O_
=====
=====JUGADA #6=====
=====
```

INTENTO 4

```
Palabra a adivinar: CO_ILA_OR
=====
=====JUGADA #8=====
=====

=====Ingrese una letra=====
=====>
```

CONCLUSION

Nos parece que el trabajo estuvo bastante interesante ya que a primera vista parece fácil pero cuando uno va a escribir el código, va contemplando dificultades y posibles opciones. Claramente el juego que mas nos costo es el juego del Ahorcado, ya que no teníamos prácticamente nada de experiencias usando variables de tipo string y descubrimos también que el lenguaje C no es el mejor para trabajar con este tipo de variables. Nos fuimos enfrentando con muchos bugs del programa y limitaciones de funciones de este. Sin embargo, es cierto que nos sirvió como practica y fuimos autodidactas en este caso.