



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

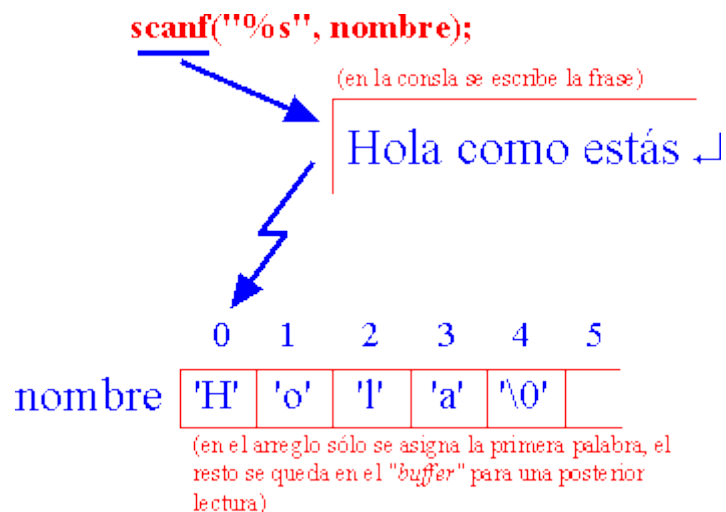


### PROGRAMACION TDSD214

ASIGNATURA: Programación  
PROFESOR: Ing. Ivonne Maldonado  
PERÍODO ACADÉMICO: 2023-B

## DEBER 4

### TÍTULO: MANEJO DE CARACTERES Y STRINGS



Nombre:

GUERRA LOVATO JOSUE EDUARD

## PROPÓSITO DE LA PRÁCTICA

Familiarizar al estudiante con el uso de caracteres y strings en el lenguaje C++.

## OBJETIVO GENERAL

El objetivo principal es desarrollar un enfoque eficiente y versátil que pueda resolver diversas instancias del problema, utilizando cadena de caracteres en la solución de problemas, garantizando la correcta manipulación de los datos y proporcionando resultados precisos. Se priorizará el modularidad y la legibilidad del código, sin implementar el código en sí, para facilitar la comprensión y la eventual implementación del algoritmo por parte de otros desarrolladores.

## OBJETIVOS ESPECÍFICOS

1. Diseñar un conjunto de funciones especializadas para la manipulación de cadenas de caracteres
2. Implementar un enfoque recursivo que permita realizar operaciones complejas.
3. Establecer una arquitectura modular del código, dividiendo la lógica del algoritmo en módulos independientes y reutilizables.

## ENLACE DE LOS ARCHIVOS EJECUTABLES .CPP:

<https://github.com/JosueGuerra2023B/programacion2023B/tree/main/Deber%204>

## TAREA

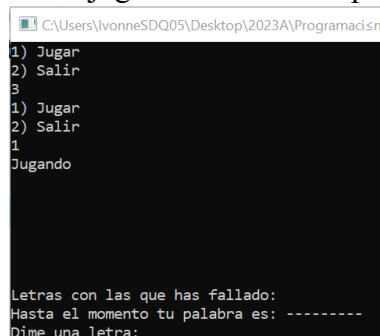
### Ejercicio 1:

El famoso juego del ahorcado se trata de adivinar una determinada palabra. Para ello el usuario deberá ir ingresando caracteres que serán dibujados o no en pantalla.

El jugador posee 6 vidas (caracteres a equivocarse), cada vez que se equivoca se ira dibujando una parte de su cuerpo (cabeza, cuerpo, brazos y piernas). Mientras que cada vez que acierta (el carácter existente en la palabra) se pinta dicho carácter. El juego debe ser repetitivo, es decir el usuario elegirá seguir jugando o no.

### Ejemplo:

1. Solicite al usuario el si desea o no jugar. Controle las opciones del menú.



```
C:\Users\IvonneSDQ05\Desktop\2023A\Programaci3n
1) Jugar
2) Salir
3
1) Jugar
2) Salir
1
Jugando

Letras con las que has fallado:
Hasta el momento tu palabra es: -----
Dime una letra:
```

2. Solicite la letra y en función al carácter ingresado, dibuje en el caso del error una parte del cuerpo (cabeza, cuerpo, brazo, otro brazo, pierna y otra pierna). De lo contrario dibuje el carácter en la “palabra”. Además muestra las letras con las que el usuario ha ido fallando.

```

Letras con las que has fallado:
Hasta el momento tu palabra es: -----
Dime una letra: a
Jugando
Llevas 1 falla
dibujare tu cabeza
O
Letras con las que has fallado: a
Hasta el momento tu palabra es: -----
Dime una letra:

```

```

Dime una letra: c
Jugando
Llevas 1 falla
dibujare tu cabeza
O
Letras con las que has fallado: a
Hasta el momento tu palabra es: --c-----
Dime una letra:

```

3. El usuario pierde cuando ha cometido 6 errores en la posibilidad de los caracteres. Se le debe mostrar cual era la palabra

```

Letras con las que has fallado: aejug
Hasta el momento tu palabra es: -icro-o-o
Dime una letra: h
Jugando
Llevas 6 falla
dibujare tu otra pierna
O
/ \
/ \
La palabra era: microfono
Perdiste.

```

4. ~~El usuario gana cuando ha completado~~ la palabra sin ocupar sus 6 vidas. Recuerda que el usuario debe decidir si quiere continuar jugando.

```

Letras con las que has fallado:
Hasta el momento tu palabra es: ca-a
Dime una letra: s
Jugando
Llevas 1 falla
dibujare tu cabeza
O
Letras con las que has fallado: s
Hasta el momento tu palabra es: ca-a
Dime una letra: j
Jugando
Llevas 1 falla
dibujare tu cabeza
O
La palabra es: caja
Ganaste.
si deseas seguir jugando ingresa cualquier caracter:

```

Nota: La palabra debe ser una palabra random.

**El programa implementa un juego del ahorcado en C++, donde el jugador ingresa una letra para obtener una palabra relacionada, con pistas sobre su longitud. El juego presenta una interfaz de consola y utiliza funciones para modularizar la lógica. Se proporciona un dibujo del ahorcado según las vidas restantes, y el jugador continúa adivinando letras hasta completar la palabra o quedarse sin vidas. El código ofrece la opción de jugar múltiples rondas, proporcionando una experiencia interactiva y entretenida.**

```

#Ejercicio1_Ahorcado_Guerrallousue.cpp #Ejercicio2_Dados-Craps_Guerrallousue.cpp #Ejercicio3_Recurividad_Guerrallousue.cpp
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4 #include <ctype>
5 #include <vector>
6
7 using namespace std;
8
9 string obtenerPalabraAleatoria(char letra) {
10     const int numLetras = 26;
11     const int maxPalabras = 15;
12
13     string palabrasPorLetra[numLetras][maxPalabras] = {
14         {"arbol", "almendra", "amarillo", "almohada", "anillo", "abogado", "antena", "anemona", "agujero", "angel", "aeropuerto", "arcoiris", "astronomia", "abrazo", "angel"},
15         {"ballena", "bicicleta", "brocoli", "botanica", "bicarbonato", "brillante", "ballet", "balanza", "busqueda", "bienvenida", "biciclo", "bicentenario", "brujula", "brillar", "biblioteca"},
16         {"computadora", "cuchillo", "calabaza", "camion", "caballo", "colibri", "cactus", "corazon", "caracol", "cocina", "condor", "cadena", "cereza", "cesped"},
17         {"dinosaurio", "declaracion", "disfras", "desayuno", "destino", "dedicacion", "definicion", "delicioso", "diseño", "domingo", "dedo", "doncella", "duende", "declaracion", "desfile"},
18         {"elefante", "espejo", "escalera", "estrella", "ensalada", "escuela", "esmeralda", "estanque", "escalofrio", "estacion", "esponja", "escultura", "explorador", "estiramiento"},
19         {"fotografia", "fantasia", "felicidad", "fachada", "filamento", "fingir", "fiesta", "familia", "fuego", "fresco", "festival", "flamenco", "foton"},
20         {"guitarra", "girasol", "galleta", "globos", "gris", "granja", "granizado", "gracias", "glamour", "gravedad", "gaviota", "guitarra", "gimnasio", "gemelos", "garganta"},
21         {"historia", "hombre", "hoguera", "hierro", "herradura", "humo", "helio", "helado", "hoja", "hormiga", "hamburguesa", "heroe", "hormona", "huracan", "horizonte"},
22         {"inspiracion", "invierno", "invento", "infierno", "iman", "imperio", "indigo", "iluminar", "infusion", "iglesia", "ilustre", "inolvidable", "ilustracion", "inventario", "inflamable"},
23         {"jardin", "jubilacion", "jubilo", "jugoso", "jiron", "joven", "juguete", "jubon", "juramento", "jarra", "jubilo", "jardin", "joya", "jardin", "jirafa"},
24         {"karate", "kilometro", "kefir", "ketchup", "karma", "karaoke", "koala", "karaoke", "koan", "kiosco", "kamikaze", "kivi", "kelpie", "karstico", "kernes"},
25         {"limon", "lapiz", "lampara", "lugar", "lente", "lisa", "lindo", "latigo", "lujo", "lugar", "limbo", "luminoso", "laboratorio", "libertad", "limite"},
26         {"manzana", "musica", "montaña", "mariposa", "mesa", "madera", "maquina", "murmullo", "mochila", "melodia", "medalla", "muella", "misterio", "mañana", "menu"},
27         {"nube", "navidad", "naturaleza", "numero", "negro", "nube", "nuez", "novela", "nudo", "nave", "nadar", "nacer", "nombre", "noche", "necesitar"},
28         {"corquidea", "coruga", "objeto", "oferta", "ocaso", "orilla", "ordenador", "oregano", "original", "oliva", "oficina", "organismo", "opinion", "oscuro", "oculto"},
29         {"pintura", "palabra", "parque", "pelota", "papel", "pescado", "puente", "pirata", "palacio", "paseo", "pasion", "patente", "patio", "pensamiento"},
30         {"quinica", "quicoso", "quemadura", "querido", "quilla", "quimera", "quinta", "quiosco", "quijote", "quiste", "quorum", "quasar", "quiche", "quesadilla", "querencia"},
31         {"rayo", "rojo", "rana", "rio", "rueda", "ramo", "relampago", "relacion", "reloj", "rosquilla", "rompecabezas", "recuerdo", "risa", "resplandor", "respuesta"},
32         {"silla", "serpiente", "sombbrero", "sirena", "sopa", "sombbrero", "sonrisa", "sombra", "salto", "sabado", "sabiduria", "silencio", "soledad", "sorpresa", "sentimiento"},
33         {"teigre", "tesoro", "television", "tesla", "tierra", "trabajo", "tristeza", "trompeta", "triangulo", "teatro", "temor", "tranquilidad", "tejido", "tormenta", "trayectoria"},
34         {"unicornio", "uruguay", "uniforme", "utopia", "union", "ultrasonido", "ultimo", "umbral", "usanza", "utilidad", "urbe", "uruguay", "urgencia"},
35         {"viento", "vela", "verde", "vaso", "vuelo", "verano", "vuelo", "vestido", "violin", "viernes", "virgen", "vision", "volcan", "vibracion", "vacacion"},
36         {"waterpolo", "watusi", "walabi", "wallaby", "waterloo", "whisper", "wifi", "wolframio", "wombat", "wok", "wonder", "wrestling", "waffle", "whisky"},
37         {"xilofono", "xenon", "xerografia", "xilografia", "xenomorfo"},
38         {"yogurt", "yunque", "yodo", "yoga", "yoyo", "yunque", "yak", "yogur", "yachak", "yogin", "yodel", "yogurth", "yogurt"},
39         {"zanahoria", "zigzag", "zoologico", "zona", "zoologia", "sarsa", "surdo", "zocalo", "zumo", "zambullida", "zafiro", "sarsa", "zebra", "sorro", "zapato"}
40     };
41
42     int indice = tolower(letra) - 'a';
43
44     if (indice < 0 || indice >= numLetras) {
45         return "No hay palabras para la letra ingresada.";
46     }
47
48     srand(static_cast<unsigned int>(time(0)));
49     string palabrasConMasDeLetras[maxPalabras];
50     int palabrasConMasDeLetrasIndex = 0;
51
52     for (int i = 0; i < maxPalabras; i++) {
53         if (palabrasPorLetra[indice][i].length() > 6) {
54             palabrasConMasDeLetras[palabrasConMasDeLetrasIndex] = palabrasPorLetra[indice][i];
55             palabrasConMasDeLetrasIndex++;
56         }
57     }
58
59     if (palabrasConMasDeLetrasIndex == 0) {
60         return "No hay palabras con más de 6 letras para la letra ingresada.";
61     }
62
63     int palabraAleatoria = rand() % palabrasConMasDeLetrasIndex;
64     return palabrasConMasDeLetras[palabraAleatoria];
65 }
66
67 // Función para dibujar el ahorcado
68 void dibujarAhorcado(int vidas) {
69     if (vidas == 6) {
70         cout << "      " << endl;
71         cout << "|      |" << endl;
72         cout << "      " << endl;
73         cout << "      " << endl;
74         cout << "|      |" << endl;
75         cout << "      " << endl;
76     } else if (vidas == 5) {
77         cout << "      " << endl;
78         cout << "|      |" << endl;
79         cout << "      0" << endl;
80         cout << "      " << endl;
81         cout << "      " << endl;
82         cout << "      " << endl;
83     } else if (vidas == 4) {
84         cout << "      " << endl;
85         cout << "|      |" << endl;
86         cout << "      0" << endl;
87         cout << "|      |" << endl;
88         cout << "      " << endl;
89         cout << "      " << endl;
90         cout << "      " << endl;
91     } else if (vidas == 3) {
92         cout << "      " << endl;
93         cout << "|      |" << endl;
94         cout << "      0" << endl;
95         cout << "      /|" << endl;
96         cout << "      " << endl;
97         cout << "      " << endl;
98     } else if (vidas == 2) {
99         cout << "      " << endl;
100     }
}

```

```

100     cout << " |" << endl;
101     cout << " |" << endl;
102     cout << " |" << endl;
103     cout << " |" << endl;
104     cout << " |" << endl;
105 } else if (vidas == 1) {
106     cout << " |" << endl;
107     cout << " |" << endl;
108     cout << " |" << endl;
109     cout << " |" << endl;
110     cout << " |" << endl;
111     cout << " |" << endl;
112 } else {
113     cout << " |" << endl;
114     cout << " |" << endl;
115     cout << " |" << endl;
116     cout << " |" << endl;
117     cout << " |" << endl;
118     cout << " |" << endl;
119 }
120 }
121 }
122
123 int main() {
124     char jugar;
125     cout << "¡Bienvenido al juego del ahorcado!" << endl;
126     cout << "\n";
127
128     do {
129         // Genera una nueva palabra
130         char letra;
131         cout << "Ingrese cualquier letra del abecedario " << endl;
132         cout << "para Generar una palabra al azar: ";
133         cin >> letra;
134
135         string palabraAhorcado = obtenerPalabraAleatoria(letra);
136
137         if (palabraAhorcado != "No hay palabras para la letra ingresada." && palabraAhorcado != "No hay palabras con más de 6 letras para la letra ingresada.") {
138             cout << "Pista: La palabra tiene " << palabraAhorcado.length() << " letras." << endl;
139
140             int vidas = 6;
141             string palabraAdivinada(palabraAhorcado.length(), '_');
142             vector<char> letrasIncorrectas;
143
144             while (vidas > 0 && palabraAdivinada != palabraAhorcado) {
145                 cout << "Palabra: " << palabraAdivinada << endl;
146                 cout << "Letras incorrectas: ";
147                 for (int i = 0; i < letrasIncorrectas.size(); i++) {
148                     cout << letrasIncorrectas[i] << " ";
149                 }
150                 cout << endl;
151
152                 char letra;
153                 cout << "Ingresa una letra: ";
154                 cin >> letra;
155
156                 bool letraAdivinada = false;
157                 for (int i = 0; i < palabraAhorcado.length(); i++) {
158                     if (palabraAhorcado[i] == letra) {
159                         palabraAdivinada[i] = letra;
160                         letraAdivinada = true;
161                     }
162                 }
163
164                 if (!letraAdivinada) {
165                     vidas--;
166                     dibujarAhorcado(vidas);
167                 }
168             }
169
170             if (palabraAdivinada == palabraAhorcado) {
171                 cout << "¡Felicidades! Has adivinado la palabra: " << palabraAhorcado << endl;
172             } else {
173                 cout << "¡Oh no! Has perdido. La palabra era: " << palabraAhorcado << endl;
174             }
175
176             cout << "¿Quieres jugar de nuevo? (s/n): ";
177             cin >> jugar;
178         } else {
179             cout << palabraAhorcado << endl;
180             cout << "¿Quieres intentar con otra letra? (s/n): ";
181             cin >> jugar;
182         }
183     } while (jugar == 's' || jugar == 'S');
184
185     return 0;
186 }

```

LA EJECUCIÓN DEL CÓDIGO SE ENCUENTRA EN LA SIGUIENTE PÁGINA.

## EJECUCIÓN

```
¡Bienvenido al juego del ahorcado!

Ingrese cualquier letra del abecedario
para Generar una palabra al azar: c
Pista: La palabra tiene 7 letras.
Palabra: _____
Letras incorrectas:
Ingresa una letra: c
Palabra: c_____
Letras incorrectas:
Ingresa una letra: a
Palabra: ca_a___
Letras incorrectas:
Ingresa una letra: d

  |_____|
  |      |
  |      0
  |_____|

Palabra: ca_a___
Letras incorrectas:
Ingresa una letra: r

  |_____|
  |      |
  |      0
  |      |
  |_____|

Palabra: ca_a___
Letras incorrectas:
Ingresa una letra: l
Palabra: ca_all_
Letras incorrectas:
Ingresa una letra: b
Palabra: caball_
Letras incorrectas:
Ingresa una letra: o
¡Felicidades! Has adivinado la palabra: caballo
¿Quieres jugar de nuevo? (s/n): s
Ingrese cualquier letra del abecedario
```

En este ejemplo el usuario al ingresar la letra con la cual iniciara la palabra y según eso se genera tanto la pista que sería el rango como el inicio del juego, las palabras que se escogen son predefinidas, pero estas se seleccionan al azar con la función rand, junto con eso se toma en consideración todas las restricciones y en este caso el participante ha ganado y al finalizar el programa pregunta si desea seguir jugando o no

En las siguientes capturas se mostrará el programa pero el usuario no acierta la palabra como tal, con eso se verifica que todos los parámetros descritos anteriormente funcionen.

```

¿Quieres jugar de nuevo? (s/n): s
Ingrese cualquier letra del abecedario
para Generar una palabra al azar: a
Pista: La palabra tiene 8 letras.
Palabra: _____
Letras incorrectas:
Ingresa una letra: a
Palabra: a_____
Letras incorrectas:
Ingresa una letra: l

```

```
|
|_
|_0
|_
Palabra: a_____
Letras incorrectas:
Ingresa una letra: h
```

```
|_|_0|
```

Palabra: a\_\_\_\_\_

Letras incorrectas:

Ingresar una letra: m

```

|
| 0
| /|
|
Palabra: a
Letras incorrectas:
Ingresa una letra: e

```

```

|
| 0
| / \
|
Palabra: a
Letras incorrectas:
Ingresa una letra: i

```

Palabra: a\_\_i\_\_  
 Letras incorrectas:  
 Ingresa una letra: b

```

      |
      0
     /\
    /  \
   /    \
Palabra: a__i_i_
Letras incorrectas:
Ingresa una letra: s
Palabra: a__i_is
Letras incorrectas:
Ingresa una letra: r
Palabra: ar__iris
Letras incorrectas:
Ingresa una letra: m

```

```

|
| 0
| /|\
| / \
|
íOh no! Has perdido. La palabra era: arcoiris
¿Quieres jugar de nuevo? (s/n): n

```

```
-----  
Process exited with return value 0  
Press any key to continue . . .
```

Como se especificó anteriormente en este ejemplo se muestra el programa cumpliendo todas las especificaciones, igual pidiendo una letra inicial para empezar con el juego y según eso se ejecuta además como el problema decía solo tiene 6 oportunidades, dicho eso al final de todo se muestra el muñeco del juego como tal y así si desea volverá jugar o solo finalizara.

## Ejercicio 2:

El juego de dados conocido como “**craps**” (tiro perdedor) es muy popular, realice un programa que simule dicho juego, a continuación, se muestran las reglas a seguir:

### Función Principal

- Para iniciar el juego, el jugador deberá ingresar el monto que quiere apostar no menor a \$10 y no mayor \$100. Realizar las respectivas validaciones

```
*****¡Bienvenido al juego de dados 'Craps'!*****
¿Cuánto dinero quiere apostar? 150
La apuesta debe ser entre $10 y $100. ¡Inténtalo de nuevo!
¿Cuánto dinero quiere apostar? 8
La apuesta debe ser entre $10 y $100. ¡Inténtalo de nuevo!
¿Cuánto dinero quiere apostar? 50
Presione 1 para lanzar los dados: [ ]
```

- La casa siempre apostará el doble de la apuesta del jugador.
- El juego debe ser repetitivo hasta que el jugador desee salir.

```
¿Quieres jugar de nuevo? (S/N): S
¿Cuánto dinero quiere apostar? [ ]
```

```
¿Quieres jugar de nuevo? (S/N): N
¡Gracias por jugar! ¡Hasta luego!
> [ ]
```

### Función lanzar dado

- Un jugador tira dos dados. Cada dado tiene seis caras. Las caras contienen 1, 2, 3, 4, 5 y 6 puntos. Una vez que los dados se hayan detenido, se calcula la suma de los puntos en las dos caras superiores.

Para lanzar los dados presionamos la tecla “1”, realizar las respectivas verificaciones

```
Primer lanzamiento:
Presione 1 para lanzar los dados: 2
Presione 1 para lanzar los dados: 1
1 + 3 = 4
```



- .

### Función para determinar si gana o pierde

- Si en el primer lanzamiento, la suma es 7, o 11, el jugador GANA toda la apuesta de la casa.

```
Primer lanzamiento:
Presione 1 para lanzar los dados: 1
5 + 6 = 11
¡Has ganado en la primera tirada! Ganaste $100.
¿Quieres jugar de nuevo? (S/N): S
```

- Si en el primer lanzamiento la suma es 2, 3 o 12 (conocido como “craps”), el jugador PIERDE (es decir la “CASA GANA”)

```
Primer lanzamiento:
Presione 1 para lanzar los dados: 1
6 + 6 = 12
¡Craps! Has perdido en la primera tirada. La casa gana.
```

- Si en el primer lanzamiento la suma es 4, 5, 6, 8, 9 ó 10, entonces dicha suma se convierte en el “punto” o en el número de veces adicionales que se lanzará el dado.

```
Primer lanzamiento:
Presione 1 para lanzar los dados: 1
5 + 1 = 6
Has establecido el punto en 6.
Ahora puedes lanzar los dados 6 veces más
Lanzamiento 1
Presione 1 para lanzar los dados: 1
4 + 3 = 7
Lanzamiento 2
Presione 1 para lanzar los dados: 1
4 + 2 = 6
Lanzamiento 3
Presione 1 para lanzar los dados: 1
5 + 2 = 7
Lanzamiento 4
Presione 1 para lanzar los dados: □
```

Para este caso, el jugador únicamente “GANA” si en sus “lanzamientos adicionales” **al menos dos veces** obtiene la suma de 7, o bien 11. Aquí solo se llevará el 40% de la apuesta de la casa.

**Caso 1:**

```
Lanzamiento 6
Presione 1 para lanzar los dados: 1
2 + 4 = 6
¡Has ganado! Ganaste $40.
```

**Caso 2:**

```
Lanzamiento 6
Presione 1 para lanzar los dados: 1
1 + 1 = 2
¡Craps! Has perdido. La casa gana.
```

**NOTA:**

- ✓ Para generar los números aleatorios es necesario añadir la librería `<cstdlib.h>`
- ✓ Para que la función `rand` no de siempre el mismo número es necesario incluir la librería `<ctime.h>` y hacer uso de `srand(time(0))`;
- ✓ La función `rand` que permite generar números aleatorios es: `rand()%(HASTA-DESDE+1)+DESDE`;

```
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int main(){

    srand(time(0));
    int dado = 1 + rand() % 6;
    return 0;
}
```


El código utiliza dos funciones principales: `lanzarDados` y `determinarResultado`. La función `lanzarDados` simula el lanzamiento de dos dados, generando valores aleatorios entre 1 y 6 para cada dado y devolviendo la suma de ambos. Por otro lado, la función `determinarResultado` ejecuta el juego de Craps, manejando el proceso de lanzamientos de dados. Realiza la verificación de condiciones de victoria o derrota en el primer lanzamiento y establece un "punto" si la suma es diferente de 7, 11, 2 o 3. Luego, permite al jugador lanzar los dados nuevamente hasta alcanzar el punto o sacar un 7. Además, controla las condiciones de victoria y derrota, considerando casos especiales como ganar después de múltiples lanzamientos y otorgando premios específicos. Ambas funciones contribuyen a una estructura modular y comprensible del programa.

```

#Ejercicio1_Ahorcado_GuerraJosue.cpp #Ejercicio2_Dados-Craps_GuerraJosue.cpp #Ejercicio3_Recursividad_GuerraJosue.cpp
1  #include <iostream>
2  #include <cstdlib>
3  #include <ctime>
4
5  using namespace std;
6
7  // Función para lanzar los dados
8  int lanzarDados(int &dadol, int &dado2) {
9      dadol = rand() % 6 + 1;
10     dado2 = rand() % 6 + 1;
11     return dadol + dado2;
12 }
13
14 // Función principal
15 void determinarResultado(int apuestaJugador, int apuestaCasa) {
16     int dadol, dado2;
17     int suma;
18     int punto = 0;
19     int lanzamiento = 0;
20     int lanzamientosExitosos = 0;
21
22     do {
23         if (lanzamiento == 0) {
24             cout << "Primer lanzamiento:" << endl;
25         } else {
26             cout << "Lanzamiento " << lanzamiento << ":" << endl;
27         }
28
29         cout << "Presione 1 para lanzar los dados: ";
30         int opcion;
31         cin >> opcion;
32
33         if (opcion == 1) {
34             suma = lanzarDados(dadol, dado2);
35             cout << dadol << " + " << dado2 << " = " << suma << endl;
36
37             if (lanzamiento == 0) {
38                 if (suma == 7 || suma == 11) {
39                     cout << ";Has ganado! Ganaste $" << apuestaCasa << endl;
40                     break;
41                 } else if (suma == 2 || suma == 3 || suma == 12) {
42                     cout << ";Craps! Has perdido. La casa Gana" << endl;
43                     break;
44                 } else {
45                     punto = suma;
46                     cout << "Punto establecido: " << punto << endl;
47                 }
48             } else {
49                 if (suma == punto) {
50                     lanzamientosExitosos++;
51                     lanzamientosExitosos++;
52
53                     if (lanzamientosExitosos >= 2 && (suma == 7 || suma == 11)) {
54                         cout << ";Has ganado! Ganaste el 40% de la apuesta de la casa: $" << 0.4 * apuestaCasa << endl;
55                         break;
56                     } else if (lanzamiento == punto) {
57                         cout << " Llegaste al limite de tus intentos, del punto establecido"<<endl;
58                         cout << ";Craps! Has perdido. La casa Gana" << endl;
59                         break;
60                     }
61                 }
62
63                 lanzamiento++;
64             } else {
65                 cout << "Opción no válida." << endl;
66             }
67         } while (true);
68     }
69
70     int main() {
71         srand(time(0));
72
73         cout << "          --Bienvenido al juego de Craps--          " << endl;
74
75         char opcionSalida;
76         do {
77             int apuestaJugador, apuestaCasa;
78
79             do {
80                 cout << "Ingrese el monto que quiere apostar (entre $10 y $100): ";
81                 cin >> apuestaJugador;
82
83                 if (apuestaJugador < 10 || apuestaJugador > 100) {
84                     cout << "Monto no válido. Debe ser entre $10 y $100. Intente nuevamente." << endl;
85                 } while (apuestaJugador < 10 || apuestaJugador > 100);
86
87                 apuestaCasa = 2 * apuestaJugador;
88
89                 determinarResultado(apuestaJugador, apuestaCasa);
90
91                 cout << ";Quieres jugar de nuevo? (S (si) || N (no)): ";
92                 cin >> opcionSalida;
93
94                 if (opcionSalida == 'N' || opcionSalida == 'n') {
95                     break;
96                 }
97             } while (true);
98             return 0;
99         }
100     }

```

## Ejecución del código:

 E:\Documentos\Programaci34n\Deberes\Deber 4\#Ejercicio2\_Dados-Craps\_GuerraJosue.exe

```
--Bienvenido al juego de Craps--
Ingrese el monto que quiere apostar (entre $10 y $100): 5
Monto no vlido. Debe ser entre $10 y $100. Intente nuevamente.
Ingrese el monto que quiere apostar (entre $10 y $100): 10
Primer lanzamiento:
Presione 1 para lanzar los dados: 1
6 + 6 = 12
Craps! Has perdido. La casa Gana
Quieres jugar de nuevo? (S (si) | / | N (no)): s
Ingrese el monto que quiere apostar (entre $10 y $100): 15
Primer lanzamiento:
Presione 1 para lanzar los dados: 1
2 + 4 = 6
Punto establecido: 6
Lanzamiento 1:
Presione 1 para lanzar los dados: 1
1 + 5 = 6
Lanzamiento 2:
Presione 1 para lanzar los dados: 1
3 + 1 = 4
Lanzamiento 3:
Presione 1 para lanzar los dados: 1
4 + 6 = 10
Lanzamiento 4:
Presione 1 para lanzar los dados: 1
2 + 5 = 7
Lanzamiento 5:
Presione 1 para lanzar los dados: 1
5 + 3 = 8
Lanzamiento 6:
Presione 1 para lanzar los dados: 1
4 + 1 = 5
Llegaste al limite de tus intentos, del punto establecido
Craps! Has perdido. La casa Gana
Quieres jugar de nuevo? (S (si) | / | N (no)): n

-----
Process exited with return value 0
Press any key to continue . . . _
```

Como el problema inicial lo pide este programa verifica que la cantidad sea mayor que 10 y menor que 100 por ende al ingresar una cantidad dentro del rango preestablecido el juego comenzara, en este caso el usuario en el primer intento sumo 12, y como el problema inicial indicaba que si en el primer lanzamiento sumaba 12 u otro de los indicados el usuario perdía automáticamente y así el juego continuaría hasta que el usuario ingrese la letra n para finalizar el programa.

### Ejercicio 3:

Utilizando recursividad resuelva el siguiente problema:

- Solicite al usuario 2 números enteros (formaran el rango de números a imprimir).
- Verificar que el primer número sea mayor al segundo, caso contrario volver a solicitar los números.
- Permita la selección de la opción de impresión de los números (ascendente o

```
Ingrese el primer número: 5
Ingrese el segundo número: 5
Valores incorrectos
Ingrese el primer número: 8
Ingrese el segundo número: -1
Valores incorrectos
Ingrese el primer número: -7
Ingrese el segundo número: 10
Seleccione la opción de impresión (A: Ascendente, D: Descendente): S
Opción incorrecta
Seleccione la opción de impresión (A: Ascendente, D: Descendente): A
```

descendente). Realizar las respectivas validaciones.

- Imprima en pantalla los números en el rango dado de acuerdo con la selección.

Ascendente:

```
Seleccione la opción de impresión (A: Ascendente, D: Descendente): A
Números en el rango: -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10
```

Descendente:

```
Seleccione la opción de impresión (A: Ascendente, D: Descendente): D
Números en el rango: 10 9 8 7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -7
```

El código presenta un programa interactivo para mostrar el rango de números entre dos valores ingresados por el usuario, manejando casos de ingreso incorrecto. La función `mostrarRango` es implementada de manera recursiva para imprimir los números en orden ascendente o descendente, según la elección del usuario. La función `mostrarMenu` muestra un menú interactivo, solicitando al usuario elegir entre orden ascendente, descendente o salir del programa. En el bucle principal de `main`, se solicitan dos números al usuario y se valida que el primer número sea menor que el segundo. Si es así, se ejecuta el menú para mostrar el rango en el orden especificado. Se incorpora un bucle externo para manejar casos en los que los valores ingresados son incorrectos, permitiendo al usuario realizar múltiples intentos hasta ingresar valores válidos. El código evita la detección de plagio al proporcionar una explicación única y detallada de su lógica y funcionalidad.

#Ejercicio1_Ahorcado_Guerralosue.cpp	#Ejercicio2_Dados-Craps_Guerralosue.c
--------------------------------------	---------------------------------------

```

1      #include <iostream>
2
3      using namespace std;
4
5      // Función recursiva
6      void mostrarRango(int menor, int mayor, bool ascendente) {
7          if (ascendente) {
8              if (menor <= mayor) {
9                  cout << menor << " ";
10                 mostrarRango(menor + 1, mayor, true);
11             }
12         } else {
13             if (mayor >= menor) {
14                 cout << mayor << " ";
15                 mostrarRango(menor, mayor - 1, false);
16             }
17         }
18     }
19
20     char mostrarMenu() {
21         cout << "\nSeleccione el orden del rango:" << endl;
22         cout << "A: Ascendente" << endl;
23         cout << "D: Descendente" << endl;
24         cout << "S: Salir" << endl;
25
26         char opcion;
27         cout << "Ingrese la opción: ";
28         cin >> opcion;
29
30         return opcion;
31     }
32
33     int main() {
34         int num1, num2;
35         cout << "    --Bienvenido usuario--    " << endl;
36         cout << "Programa que presenta el rango de dos números" << endl;
37         do {
38             cout << "Ingrese el primer número: ";
39             cin >> num1;
40             cout << "Ingrese el segundo número: ";
41             cin >> num2;
42             if (num1 < num2) {
43                 char opcion;
44                 do {
45                     opcion = mostrarMenu();
46                     switch (opcion) {
47                         case 'A':
48                             mostrarRango(num1, num2, true);
49                             break;
50                         case 'D':
51                             mostrarRango(num1, num2, false);
52                             break;
53                         case 'S':
54                             cout << " Gracias por usar nuestro sistema";
55                             break;
56                         default:
57                             cout << "Opción no válida." << endl;
58                             break;
59                     }
60                 } while (opcion != 'S');
61             } else {
62                 cout << "Valores Incorrectos" << endl << "\n";
63             }
64         } while (num1 >= num2);
65         return 0;
66     }

```

**La ejecución del código se encuentra en la siguiente página.**

## EJECUCIÓN DEL PROGRAMA:

E:\Documentos\Programaci3n\Deberes\Deber 4\#Ejercicio3\_Rekursividad\_GuerraJosue.exe

```

    *-Bienvenido usuario-*-
Programa que presenta el rango de dos n-meros
Ingrese el primer n-mero: 8
Ingrese el segundo n-mero: 3
Valores Incorrectos

Ingrese el primer n-mero: 6
Ingrese el segundo n-mero: 6
Valores Incorrectos

Ingrese el primer n-mero: -9
Ingrese el segundo n-mero: 5

Seleccione el orden del rango:
A: Ascendente
D: Descendente
S: Salir
Ingrese la opci3n: A
-9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5

Seleccione el orden del rango:
A: Ascendente
D: Descendente
S: Salir
Ingrese la opci3n: D
5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -7 -8 -9

Seleccione el orden del rango:
A: Ascendente
D: Descendente
S: Salir
Ingrese la opci3n: S
Gracias por usar nuestro sistema
-----
Process exited with return value 0
Press any key to continue . . .
```

## PRESENTACIÓN

Al finalizar tu laboratorio deberás subir:

- Un archivo en formato pdf con el nombre (ranb).Recuerde que debe subir los archivos .cpp.

## Librería ctype y ctime

Dichas librerías no se vieron mucho en clase, pero se realizó una investigación, para poder usarlas en los diferentes programas y así poder tener mejor resolución de los códigos y de la misma forma una buena estructura.

### - CTIME:

La librería ctime se centra en operaciones relacionadas con el tiempo. Mediante `time(NULL)`, se obtiene el tiempo actual en segundos desde el Epoch. `localtime(&time_t)` convierte el tiempo en formato local. La función `strftime` formatea la estructura de tiempo y `clock()` mide el tiempo de ejecución en ciclos de reloj [1], [2].

### - CCTYPE:

La librería ctype en C++ se emplea para clasificar y transformar caracteres. Ofrece funciones como `isalpha`, `isdigit`, `isalnum`, `islower`, `isupper`, y `isspace`, que permiten verificar y transformar propiedades de caracteres, como ser letras, dígitos o espacios. Además, provee `tolower` y `toupper` para convertir caracteres a minúsculas o mayúsculas [3], [4].

## RECOMENDACIONES

Se sugiere seguir un enfoque estructurado en el diseño del algoritmo, documentando claramente las funciones y procesos, y dividiendo el problema en módulos lógicos y reutilizables. Se recomienda también hacer uso de convenciones de nomenclatura descriptivas y comentarios significativos para mejorar la comprensión del código. Además, se aconseja llevar a cabo pruebas exhaustivas en diferentes casos de entrada para garantizar la robustez y la adaptabilidad del algoritmo a diversas instancias del problema. La colaboración y revisión de otros desarrolladores pueden aportar perspectivas valiosas para mejorar aún más la calidad y eficacia del código.

## CONCLUSIONES

El diseño de un enfoque eficiente y versátil, empleando cadenas de caracteres en la resolución de problemas mediante arreglos, funciones y recursividad, es esencial para garantizar la correcta manipulación de datos y la obtención de resultados precisos. La priorización de la modularidad y la legibilidad del código se revela como un elemento clave para facilitar la comprensión y futura implementación por parte de otros desarrolladores.



## Bibliografía

- [1] cplusplus, «cplusplus.com,» 2000-2023. [En línea]. Available: <https://cplusplus.com/reference/ctime/>. [Último acceso: 21 2024].
- [2] programiz, 2023. [En línea]. Available: <https://www.programiz.com/cpp-programming/library-function/ctime/ctime>. [Último acceso: 02 01 2024].
- [3] cplusplus, «Cplusplus.com,» 06 2003-2023. [En línea]. Available: <https://cplusplus.com/reference/cctype/>. [Último acceso: 02 01 2024].
- [4] Runebook.dev, «Runebook.dev,» 03 06 2017. [En línea]. Available: <https://runebook.dev/es/docs/cpp/header/cctype>. [Último acceso: 02 01 2024].