

# Senku

Integrantes:

- Walter Diaz Ysla (201810141)
- Lizbet Condo Santi (201810360)
- Andrea Haro Farfan (201810173)

# Funciones.h

En este header se encuentran los nombres de las funciones que se usarán para el correcto funcionamiento del programa

```
1  #include <iostream>
2  #include<stdio.h>
3  #include <stdlib.h>
4  using namespace std;
5  typedef char tablero[100][100];
6
7  void inicializarTablero_ingles(tablero tab,int fila,int columna);
8
9  void mostrarTablero(tablero tab, int fila, int columna);
10
11 int validar_mov(int origenFila, int origenColumna, int destinoFila, int destinoColumna, tablero tab);
12
13 void mover(int origenFila, int origenColumna, int destinoFila, int destinoColumna, tablero tab);
14
15 void perdiste(tablero tab,int fila,int columna);
16
17 void final_juego(tablero tab,int fila,int columna);
18
19 void inicializarTablero_aleman(tablero tab,int fila, int columna);
20
21 void inicializarTablero_asimetrico(tablero tab, int fila, int columna);
22
23 void jugar_ingles();
24
25 void jugar_aleman();
26
27 void jugar_asimetrico();
28
29 void menu();
30
31
```

# Funciones.cpp

Esta función describe la ubicación del espacio vacío (+) y las fichas (O) para el tablero inglés

```
#include <funciones.h>
```

```
void inicializarTablero_ingles(tablero tab,int fila,int
columna){

    for (int f = 0; f < fila; f++){
        for (int c = 0; c < columna; c++){
            if ((f == 3 && c == 3) || (f == 3 && c == 4) || (f == 3 &
& c == 5)){
                tab[f][c] = 'O';
            }

            else{
                if( (f < 2 || f > 4) && (c < 2 || c > 4) ){
                    tab[f][c] = ' ';
                }else{
                    tab[f][c] = '+';
                }
            }
        }
    }
}
```

# Funciones.cpp

Esta función se encarga de mostrar los números del tablero, para que el usuario pueda ubicarse en el tablero al momento de jugar.

```
22 void mostrarTablero(tablero tab, int fila, int columna){
23     int numeros;
24
25     for (int f = 0; f < fila; f++){
26         if (f == 0){
27             for (numeros = 0; numeros <= fila; numeros++){
28                 if (numeros == 0){
29                     cout<<" ";
30                 }else{
31                     cout<<" "<<numeros;
32                 }
33             }cout<<endl;
34         }
35         cout<<" "<<(f+1);
36         for (int c = 0; c < columna; c++){
37             cout<<" "<<tab[f][c];
38         }
39         cout<<endl;
40     }
41 }
42
```

# Funciones.cpp

Esta función es la que se encarga de validar los movimientos al momento de jugar, retorna 1 si el movimiento es válido y retorna 0 si es inválido.

-Que mínimo exista una ficha (O) al costado de la ficha a mover.

-Que la posición de origen sea (O) y la de destino vacío (+).

-Que solo se pueda mover 2 espacios.

```
43 int validar_mov(int origenFila, int origenColumna, int
destinoFila, int destinoColumna, tablero tab){
44     //Que la ficha a mover tenga a su costado alguna ficha
45     if( (tab[origenFila-2][origenColumna-1]=='O' || tab
[origenFila][origenColumna-1]=='O') || (tab[origenFila-1]
[origenColumna-2]=='O' || tab[origenFila-1][origenColumna]
=='O')){
46         cout<<"\nHay un 0 al costado de la ficha\n";
47
48     //Que la posicion de origen sea 0 y la de destino +
49     if(tab[origenFila - 1][origenColumna - 1] == 'O' && tab
[destinoFila - 1][destinoColumna - 1] == '+'){
50         cout<<"\nLa posicion de origen es 0 y de destino +";
51     //Que solo se mueva dos posiciones ya sea vertical,
horizontal, arriba o abajo
52     if(abs(origenFila-destinoFila)== 2 || abs
(origenColumna-destinoColumna)==2){
53         cout<<"\nSolo se mueve dos posiciones\n";
54     //Movimiento horizontal
```

# Funciones.cpp

Esta es la continuación de la función anterior

-Si el movimiento es horizontal y al costado de la ficha no hay un espacio vacío (+), esta se puede mover a la derecha o la izquierda.

-Si el movimiento es vertical y al costado de la ficha no hay un espacio vacío (+), esta se puede mover hacia abajo o hacia arriba.

```
55  if (origenFila == destinoFila){
56      //movimiento derecha ---->
57      if ((origenColumna < destinoColumna) && (tab
58          [origenFila-1][origenColumna]!='+')){
59          return 1;
60          cout<<"\nSe puede mover a la derecha\n";
61      }else{
62          //movimiento izquierda <----
63          if(tab[origenFila-1][origenColumna-2] !='+'){
64              cout<<"\nSe puede mover a la izquierda\n";
65              return 1;
66          }
67      }
68      //Movimiento vertical
69      else{
70          if (origenColumna == destinoColumna){
71              //movimiento hacia abajo
72              if ((origenFila < destinoFila) && (tab[origenFila]
73                  [origenColumna-1]!='+')){
74                  cout<<"\nSe puede mover hacia abajo\n";
75                  return 1;
76              }else{
77                  //movimiento hacia arriba
78                  if(tab[origenFila-2][origenColumna-1]!='+'){
79                      cout<<"\nSe puede mover hacia arriba\n";
80                      return 1;
81                  }
82              }
83          }
84      }
85  }
```

# Funciones.cpp

Esta función se encarga de realizar los movimientos en el juego.

-Se cambia la ficha que se mueve por un espacio vacío (+) y la posición de destino por una ficha (O).

-Esto ocurre solo para el movimiento horizontal (hacia la derecha o hacia la izquierda) y para el movimiento vertical (hacia abajo o hacia arriba).

```
98 void mover(int origenFila, int origenColumna, int
destinoFila, int destinoColumna, tablero tab){
99
100     tab[origenFila - 1][origenColumna - 1] = '+';
101     tab[destinoFila - 1][destinoColumna - 1] = 'O';
102     //Movimiento horizontal
103     if (origenFila == destinoFila){
104         if (origenColumna < destinoColumna ){
105             //movimiento derecha
106             tab[origenFila - 1][origenColumna] = '+';
107         }
108         //movimiento izquierda
109         else{
110             tab[origenFila - 1][origenColumna - 2] = '+';
111         }
112     }
113     //Movimiento vertical
114     else{
115         if (origenColumna == destinoColumna){
116             //movimiento hacia abajo
117             if (origenFila < destinoFila){
118                 tab[origenFila][origenColumna - 1] = '+';
119             }
120             //movimiento hacia arriba
121             else{
122                 tab[origenFila - 2][origenColumna - 1] = '+';
123             }
124         }
125     }
```



# Funciones.cpp

Esta función es la que se encarga de verificar si ya no hay más movimientos válidos para realizar en el juego, por lo que el usuario perdería.

```
127 void perdiste(tablero tab,int fila,int columna){
128     int cuentaFichas=0;
129     for (int f = 0; f < fila; f++){
130         for (int c = 0; c < columna; c++){
131             if(tab[f][c]=='O' & cuentaFichas==1){
132                 if(tab[f-1][c]=='O' || tab[f+1][c]=='O' || tab[f][c-1] == 'O' || tab[f][c+1]=='O'){
133                     break;
134                 }
135                 else{
136                     cuentaFichas++;
137                 }
138             } if (cuentaFichas>1){
139                 cout<<"Perdiste";
140                 break;
141             }
142         }
143     }
144 }
145
```



# Funciones.cpp

Esta función es la que se encarga de recorrer todo el tablero y verificar si:

-Hay una sola ficha: Ganaste

-Hay más de una ficha:  
Perdiste

```
149 void final_juego(tablero tab,int fila,int columna){
150     int contador = 0;
151
152     for (int f = 0; f < fila; f++){
153         for (int c = 0; c < columna; c++){
154             if(tab[f][c] == '0'){
155                 contador++;
156             }
157         }
158     }
159     if (contador == 1){
160         cout<<"Ganaste!!!! :) :) :) ";
161     }else{
162         if(contador>1){
163             perdiste(tab,fila,columna);
164         }
165     }
166 }
```

# Funciones.cpp

En esta función se muestra la posición del espacio vacío (+) y la ubicación de las fichas (O) para el tablero alemán.

```
167
168 void inicializarTablero_aleman(tablero tab,int fila, int
columna){
169
170     for (int f = 0; f < fila; f++){
171         for (int c = 0; c < columna; c++){
172             if (f == 4 && c == 4){
173                 tab[f][c] = '+';
174             }else{
175                 if( (f < 3 || f > 5)&&(c < 3 || c > 5) ){
176                     tab[f][c] = ' ';
177                 }else{
178                     tab[f][c] = 'O';
179                 }
180             }
181         }
182     }
183 }
```

# Funciones.cpp

En esta función muestra la posición del espacio vacío (+) y la ubicación de la ficha (O) para el tablero asimétrico.

```
185 void inicializarTablero_asimetrico(tablero tab, int fila,  
186 int columna){  
187     for (int f = 0; f < fila; f++){  
188         for (int c = 0; c < columna; c++){  
189             if (f == 4 && c == 3){  
190                 tab[f][c] = '+';  
191             }else{  
192                 if( (f < 3 || f > 5)&&(c < 2 || c > 4) ){  
193                     tab[f][c] = ' ';  
194                 }else{  
195                     tab[f][c] = 'O';  
196                 }  
197             }  
198         }  
199     }  
200 }
```

# Funciones.cpp

En esta función se pide al usuario el ingreso de la posición original y la posición de destino de la ficha a mover para el tablero ingles.

```
10 void jugar_ingles(){
11     tablero tab;
12     inicializarTablero_ingles(tab,7,7);
13     mostrarTablero(tab,7,7);
14
15     int origenFila, origenColumna, destinoFila, destinoColumna,
        movimientoValido, entrada=1;
16
17     while(entrada != 0){
18         cout<<"\nIngresa la posición de origen (fila):";
19         cin>>origenFila;
20         cout<<"\nIngresa la posición de origen (columna):";
21         cin>>origenColumna;
22         cout<<endl;
23
24         cout<<"Ingresa la posición de destino (fila):";
25         cin>>destinoFila;
26         cout<<"\nIngresa la posición de destino (columna):";
27         cin>>destinoColumna;
28         cout<<endl;
29     }
```

# Funciones.cpp

A continuación, se restringe los movimientos de los jugadores, se supervisa si el movimiento es válido según las reglas del juego

```
29
30 movimientoValido = validar_mov(origenFila,origenColumna,
    destinoFila,destinoColumna,tab);
31 if(movimientoValido==1){
32     mover(origenFila,origenColumna,destinoFila,
        destinoColumna,tab);
33     cout<<endl;
34     mostrarTablero(tab,7,7);
35 }else{
36     cout<<"\nMOVIMIENTO INVÁLIDO\n";
37     cout<<endl;
38     mostrarTablero(tab,7,7);
39 }
40
41 final_juego(tab,7,7);
42 }
43 }
44
```

# Funciones.cpp

En esta función se pide al usuario el ingreso de la posición original y la posición de destino de la ficha a mover para el tablero Alemán.

```
47 void jugar_aleman(){
48     tablero tab;
49     inicializarTablero_aleman(tab,9,9);
50     mostrarTablero(tab,9,9);
51     int origenFila, origenColumna, destinoFila, destinoColumna,
        movimientoValido, entrada=1;
52
53     while(entrada != 0){
54         cout<<"\nIngresa la posición de origen (fila):";
55         cin>>origenFila;
56         cout<<"\nIngresa la posición de origen (columna):";
57         cin>>origenColumna;
58         cout<<endl;
59
60         cout<<"Ingresa la posición de destino (fila):";
61         cin>>destinoFila;
62         cout<<"\nIngresa la posición de destino (columna):";
63         cin>>destinoColumna;
64         cout<<endl;
```

# Funciones.cpp

A continuación, se restringe los movimientos de los jugadores, se supervisa si el movimiento es válido según las reglas del juego

```
65
66 movimientoValido = validar_mov(origenFila,origenColumna,
destinoFila,destinoColumna,tab);
67
68     if(movimientoValido==1){
69         mover(origenFila,origenColumna,destinoFila,
destinoColumna,tab);
70         cout<<endl;
71         mostrarTablero(tab,9,9);
72     }else{
73         cout<<"\nMOVIMIENTO INVÁLIDO\n";
74         cout<<endl;
75         mostrarTablero(tab,9,9);
76     }
77     final_juego(tab,9,9);
78 }
79 }
80
```



# Funciones.cpp

En esta función se pide al usuario el ingreso de la posición original y la posición de destino de la ficha a mover para el tablero Asimétrico.

```
82
83 void jugar_asimetrico(){
84     tablero tab;
85     inicializarTablero_asimetrico(tab,8,8);
86     mostrarTablero(tab,8,8);
87     int origenFila, origenColumna, destinoFila, destinoColumna,
        movimientoValido, entrada=1;
88
89     while(entrada != 0){
90         cout<<"\nIngresa la posición de origen (fila):";
91         cin>>origenFila;
92         cout<<"\nIngresa la posición de origen (columna):";
93         cin>>origenColumna;
94         cout<<endl;
95
96         cout<<"Ingresa la posición de destino (fila):";
97         cin>>destinoFila;
98         cout<<"\nIngresa la posición de destino (columna):";
99         cin>>destinoColumna;
100        cout<<endl;
101    }
```

# Funciones.cpp

A continuación, se restringe los movimientos de los jugadores, se supervisa si el movimiento es válido según las reglas del juego

```
101
102 movimientoValido = validar_mov(origenFila,origenColumna,
destinoFila,destinoColumna,tab);
103     if(movimientoValido==1){
104         mover(origenFila,origenColumna,destinoFila,
destinoColumna,tab);
105         cout<<endl;
106         mostrarTablero(tab,8,8);
107     }else{
108         cout<<"\nMOVIMIENTO INVÁLIDO\n";
109         cout<<endl;
110         mostrarTablero(tab,8,8);
111     }
112
113     final_juego(tab,8,8);
114 }
115 }
116
117
```

# Funciones.cpp

Esta función menú es la encargada de mostrar y ejecutar las opciones según lo que el usuario escoja.

```
118 void menu(){
119     int opcion;
120
121     cout<<"\n Menú:          JUEGO    SENKU    ";cout<<endl;
122     cout<<"-----"
123     -----";
124     cout<<endl;
125     cout<<"1.Estilo Inglés";
126     cout<<"\n2.Estilo Alemán";
127     cout<<"\n3.Estilo Asimétrico";
128     cout<<endl;
129     cout<<"-----"
130     -----";
131     cout<<"\n0.Salir del programa";
132     cout<<endl;
133     cout<<endl;
134     cout<<"Seleccionar una opción:";cin>>opcion;
135     if(opcion==1){
136         cout<<endl;
137         jugar_ingles();
138     }if(opcion==2){
139         cout<<endl;
140         jugar_aleman();
141     }if(opcion==3){
142         cout<<endl;
143         jugar_asimetrico();
144     }if(opcion==0){
145         cout<<"\nVuelve cuando quieras!!!!!!  :) :) :) ";
146     }
147     }else
148     cout<<"Entrada inválida";cout<<endl;
149 }
```

# Main

Esta función nos permite acceder al programa entero(Conjunto de funciones).

```
147  
148 int main(){  
149     menu();  
150     return 0;  
151 }  
152
```

# GRACIAS

Github: <https://github.com/walterdiaz11/Senku/tree/master>