

Proyecto 2

```
using Proyecto_2;

class Program
{
    static void Main(string[] args)
    {
        Tablero tablero = new Tablero(); //Se creo el objeto de la clase
        tablero
        Pieza pieza= new Pieza(); // Se creo el objeto de la clase pieza

        int opcion = '0'; // Se declaró la variable opcion como int para
        realizar el menú

        while (opcion != '5') //Se creo un bucle de while para poder
        realizar el switch
        {
            Console.WriteLine("");
            Console.WriteLine(" ♟️ Juego de ajedrez ♟️");
            Console.WriteLine("Siga las instrucciones en el orden
correspondiente");
            Console.WriteLine("1) Agregar las piezas ♟️ ♜ ♝"); //Se crearon
las instrucciones para buscar los posibles movimientos
            Console.WriteLine("2) Agregar la dama ♚");
            Console.WriteLine("3) Mostrar los posibles movimientos
↔️ ↕️ ↗️ ↘️");
            Console.WriteLine("4) Imprimir tablero □");
            Console.WriteLine("5) Salir del juego 🏹");

            opcion = Console.ReadLine()[0];

            int opcionNum = int.Parse(opcion.ToString());

            switch (opcionNum)
            {
                case '1':
                    Console.WriteLine("¿Cuántas piezas desea agregar?");
                    int CantidadPieza= int.Parse(Console.ReadLine());
                    tablero.GuardarTablero(CantidadPieza); //Se mandó a
llamar a la función guardar tablero, que contiene la información de las
piezas

                    break;
                case '2':
```

```
        tablero.GuardarDama(); //Se mandó a llamar a la función
guardar dama, que contiene la información de la dama
        break;
    case '3':
        tablero.Movimientos(); //Se mandó a llamar a la función
movimientos, que contiene la información de los movimientos del tablero
        break;
    case '4':
        tablero.MostrarTablero(); //Se mandó a llamar a la
función mostrar tablero, que contiene la información del tablero
        break;
    case '5': //Se creó esta opción para poder salir del juego
        break;

    default: //Se creo un default para verificar que se ingrese
la opción indicada
        Console.WriteLine("Opción inválida");
        break;
    }
}
}
```

Clase pieza

```
public class Pieza
{
    public int cantidadPieza; // Se declaró la variable cantidadPieza
    public string colorPieza; // Se declaró la variable colorPieza
    public string tipoPieza; // Se declaró la variable tipoPieza
    public string posicionPieza; // Se declaró la variable posicionPieza
    public int Filas; // Se declaró la variable Filas
    public int Columna; // Se declaró la variable Columnas

    public Pieza() //Se creo una función para declara los valores que
    contienen la información de la pieza
    {
        colorPieza = ""; //La variables se asignaron con valores vacíos para
        poder aceptar la información que ingrese el usuario
        tipoPieza = "";
        posicionPieza = "";
    }

    public void TipoPieza() //Se creo una función para solicitar la
    información del tipo pieza
    {
        Tablero objtablero = new Tablero(); //Se creo el objeto tablero
        Console.WriteLine("¿Tipo de pieza que desea agregar? " + "(alfil,
        peon, torre, caballo)");
        tipoPieza = Console.ReadLine(); //Se guarda la información en el
        tipo pieza

        do //Se utilizó un ciclo do while para validar los colores blanco y
        negro; de otra manera no acepta el color ingresado
        {
            Console.WriteLine("¿Tipo de color para el/la " + tipoPieza + "?
            " + "(blanco o negro)");
            colorPieza = Console.ReadLine(); // Se guarda el color ingresado
            en la variable
            if (colorPieza == "blanco" || colorPieza == "negro") // Se
            utiliza la condición if para validar que el color sea adecuado y el código
            pueda continuar
            {
                break; //Se termina el ciclo
            }
        }
    }
}
```

```

        if (colorPieza != "blanco" && colorPieza != "negro") //Se
        utiliza la condición if para mostrar un mensaje de color inválido si se
        ingresa un color no aceptado
        {
            Console.WriteLine("Color inválido");
        }

    } while (true); //Indica que el ciclo se repetirá hasta que se
    ingrese el color adecuado

    Console.WriteLine("¿Posición para el/la " + tipoPieza + "?");
    posicionPieza = Console.ReadLine(); //Se guarda la posición de pieza
    ingresada

    for (int x = 0; x < objtablero.tablero.GetLength(0); x++) //Se
    recorren las filas de la matriz para encontrar la posición
    {
        for (int j = 0; j < objtablero.tablero.GetLength(1); j++) //Se
        recorren las columnas de la matriz para encontrar la posición
        {
            if (posicionPieza == objtablero.tablero[x, j]) //Se utilizó
            la condición para verificar que la condición que se ingresó tiene relación
            con el tablero
            {
                Filas = x; // Se asignaron los valores para las
                Columna = j;
            }
        }
    }

}

public void Dama() //Se creo la función dama donde se almacenará la
información de la dama
{
    Tablero objtablero = new Tablero(); //Se creo el objeto tablero
    tipoPieza = "dama"; //Se asignó el un valor específico a la variable
    tipoPieza, ya que, solo es para la dama

    do //Se creo un ciclo do while para verificar el color de la dama
    {
        Console.WriteLine("¿Tipo de color de la dama?" + "(blanco o
negro)");
    }
}

```

```
        colorPieza = Console.ReadLine(); //El color escogido se guardará
en la variable colorPieza
        if (colorPieza == "blanco" || colorPieza == "negro") //Se
utiliza la condición if para validar que el color sea adecuado y el código
pueda continuar
        {
            break; //Se termina el ciclo
        }
        if (colorPieza != "blanco" && colorPieza != "negro") //Se
utiliza la condición if para mostrar un mensaje de color inválido si se
ingresa un color no aceptado
        {
            Console.WriteLine("Color inválido");
        }
    }
    while (true); //Indica que el ciclo se repetirá hasta que se ingrese
el color adecuado

    Console.WriteLine("¿Posición de la dama?");
    posicionPieza = Console.ReadLine(); //Se guardará la posición
ingresada en la variable posicionPieza

    for (int i = 0; i < objtablero.tablero.GetLength(0); i++) //Se
recorren las filas de la matriz para encontrar la posición
    {
        for (int j = 0; j < objtablero.tablero.GetLength(1); j++) //Se
recorren las columnas de la matriz para encontrar la posición
        {
            if (posicionPieza == objtablero.tablero[i, j]) //Se utilizó
la condición para verificar que la condición que se ingresó tiene relación
con el tablero
            {
                Filas = i; // Se asignaron los valores para las
variables filas y columnas
                Columna = j;
            }
        }
    }
}
```

Clase tablero

```
public class Tablero
{
    int FilaDama; //Se declaró la variable para la fila de la dama
    int ColumnaDama; //Se declaró la variable para la columna de la dama
    public string[,] tablero; //Se creo la matriz de dos dimensiones para el
    tablero
    public Pieza[,] MTablero = new Pieza[8, 8]; //Se creo la matriz donde se
    guardaran todos los datos(matriz de la clase pieza)
    public Tablero() //Se creo la función del tablero para crear la matriz
    del tablero
    {
        tablero = new string[8, 8]; //Se creo la matriz del tablero con dos
        dimensiones de 8 filas y 8 columnas
        CrearTablero(); // Se crea el método CrearTablero
    }

    public void GuardarTablero(int cantidadPieza) //Se creo la función de
    GuardarTablero
    {
        for (int i = 0; i < cantidadPieza; i++) // Se repite hasta cumplir
        con la cantidad de piezas establecidas
        {
            Pieza objPieza = new Pieza(); //Se creo un objeto de la clase
            pieza
            objPieza.TipoPieza(); // Se creo un método para el objeto de
            tipoPieza para indicar el tipo de pieza que el usuario ingrese

            if (MTablero[objPieza.Filas, objPieza.Columna] == null) // Se
            creo una condición para verificar que el lugar donde se colocará la pieza se
            encuentre disponible
            {
                MTablero[objPieza.Filas, objPieza.Columna] = objPieza; //Se
                coloca la pieza en el tablero utilizando objPieza.Filas y objPieza.Columna
                para indicar el lugar específico en el que se colocarán
            }
            else
            {
                Console.WriteLine("La posición ya esta siendo usada por " +
                MTablero[objPieza.Filas, objPieza.Columna].tipoPieza + " ")
            }
        }
    }
}
```

```
+ MTablero[objPieza.Filas, objPieza.Columna].colorPieza); //Este mensaje
aparecerá, si la posición ya se encuentra ocupada y la clase de pieza que lo
está ocupando
```

```
    }
  }
}

public void GuardarDama() //Se creo la función guardardama para guardar
la información de su posición
```

```
{
    Pieza objPieza = new Pieza(); //Se creo el objeto pieza
    objPieza.Dama(); //Se creo un método para el objeto pieza para
indicar que solo es para la dama
    if (MTablero[objPieza.Filas, objPieza.Columna] == null) // Se creo
una condición para verificar que el lugar donde se colocará la dama se
encuentre disponible
```

```
{
    MTablero[objPieza.Filas, objPieza.Columna] = objPieza; //Se
coloca la dama en el tablero utilizando objPieza.Filas y objPieza.Columna
para indicar el lugar específico en el que se colocarán
```

```
}
else
{
    Console.WriteLine("La posición ya está siendo usada por "+
MTablero[objPieza.Filas, objPieza.Columna].tipoPieza +
MTablero[objPieza.Filas, objPieza.Columna].colorPieza); //Este mensaje
aparecerá, si la posición ya se encuentra ocupada y mostrará el tipo de
pieza que lo está ocupando
```

```
}

public void Movimientos() //Se creo la función movimientos
{
    //En esta parte se busca la posiccion de la dama
    for (int i = 0; i < MTablero.GetLength(0); i++) //Recorre la matriz
del tablero, esto es para las filas
    {
        for (int j = 0; j < MTablero.GetLength(1); j++) //Recorre la
matriz del tablero, esto es para las columnas
        {
            if (MTablero[i, j] == null) //Se creo la condición para
verificar si cada cuadro del tablero se encuentra vacio o no
            {
                Pieza objPieza = new Pieza(); //Se creo el objeto pieza
```

```

        objPieza.tipoPieza = "N/P"; //Asigna el valor no hay
        pieza a cada celda de tipoPieza que se encuentra vacía
        MTablero[i, j] = objPieza; //Asigna un objeto a objPieza
        para que cada cuadro del tablero no se encuentre vacío
    }
    if (MTablero[i, j].tipoPieza == "dama") // Se creo la
    condición if para verificar la posición de la dama
    {
        FilaDama = i; //Se guarda la posición de filas de la dama
        ColumnaDama = j; //Se guarda la posición de columnas de
        la dama
    }
}

Console.WriteLine("Movimientos horizontales hacia la derecha");
for (int j = ColumnaDama + 1; j < 8; j++) //Se recorre la matriz
para verificar las piezas que se encuentran de manera horizontal hacia la
derecha de la dama
{
    if (MTablero[FilaDama, j].tipoPieza == "N/P") //Se crea una
    condición para verificar si no hay pieza
    {
        Console.WriteLine("Posición vacía " + tablero[FilaDama, j]);
        //Se escribe posición vacía si no hay pieza
    }
    if (MTablero[FilaDama, j].tipoPieza != "N/P" &&
    MTablero[FilaDama, j].colorPieza != MTablero[FilaDama,
    ColumnaDama].colorPieza) //Se creo la condición para validar lo que sucede
    si existe una pieza de color distinto a la dama
    {
        Console.WriteLine(MTablero[FilaDama, j].tipoPieza + " en la
        posición " + MTablero[FilaDama, j].posicionPieza); //Se imprime la pieza y
        el color, solo llega hasta la posición donde se encuentre la pieza, ya que,
        no puede ir más lejos
        break;
    }
    else
    {
        if (MTablero[FilaDama, j].tipoPieza != "N/P") // Se creo una
        condición para verificar si la pieza es del mismo color de la dama
        {
            break; //Se rompe el bucle, ya que, la dama no puede
            comerse a esa pieza
        }
    }
}

```



```

    }
}

Console.WriteLine("");
Console.WriteLine("Movimientos horizontales hacia la izquierda");
for (int j = ColumnaDama - 1; j >= 0; j--) //Se recorre la matriz
para verificar las piezas que se encuentran de manera horizontal hacia la
izquierda de la dama
{
    if (MTablero[FilaDama, j].tipoPieza == "N/P") //Se crea una
condición para verificar si no hay pieza
    {
        Console.WriteLine("Posición vacía " + tablero[FilaDama, j]);
//Se escribe posición vacía si no hay pieza
    }
    if (MTablero[FilaDama, j].tipoPieza != "N/P" &&
MTablero[FilaDama, j].colorPieza != MTablero[FilaDama,
ColumnaDama].colorPieza) //Se creo la condición para validar lo que sucede
si existe una pieza de color distinto a la dama
    {
        Console.WriteLine(MTablero[FilaDama, j].tipoPieza + " en la
posición " + MTablero[FilaDama, j].posicionPieza); //Se imprime la pieza y
el color, solo llega hasta la posición donde se encuentre la pieza, ya que,
no puede ir más lejos
        break;
    }
    else
    {
        if (MTablero[FilaDama, j].tipoPieza != "N/P") // Se creo una
condición para verificar si la pieza es del mismo color de la dama
        {
            break; //Se rompe el bucle, ya que, la dama no puede
comerse a esa pieza
        }
    }
}

Console.WriteLine("");
Console.WriteLine("Movimientos verticales hacia arriba");
for (int j = FilaDama - 1; j >= 0; j--) //Se recorre la matriz para
verificar las piezas que se encuentran de manera vertical hacia arriba
{
    if (MTablero[j, ColumnaDama].tipoPieza == "N/P") //Se crea una
condición para verificar si no hay pieza
    {

```

```

        Console.WriteLine("Posición vacía " + tablero[j,
ColumnaDama]); //Se escribe posición vacía si no hay pieza
    }
    if (MTablero[j, ColumnaDama].tipoPieza != "N/P" && MTablero[j,
ColumnaDama].colorPieza != MTablero[FilaDama, ColumnaDama].colorPieza) //Se
creo la condición para validar lo que sucede si existe una pieza de color
distinto a la dama
    {
        Console.WriteLine(MTablero[j, ColumnaDama].tipoPieza + " en
la posición " + MTablero[j, ColumnaDama].posicionPieza); //Se imprime la
pieza y el color, solo llega hasta la posición donde se encuentre la pieza,
ya que, no puede ir más lejos
        break;
    }
    else
    {
        if (MTablero[j, ColumnaDama].tipoPieza != "N/P") // Se creo
una condición para verificar si la pieza es del mismo color de la dama
        {
            break; //Se rompe el bucle, ya que, la dama no puede
comerse a esa pieza
        }
    }
}

Console.WriteLine("");
Console.WriteLine("Movimientos verticales hacia abajo");
for (int j = FilaDama + 1; j < 8; j++) //Se recorre la matriz para
verificar las piezas que se encuentran de manera vertical hacia abajo
{
    if (MTablero[j, ColumnaDama].tipoPieza == "N/P") //Se crea una
condición para verificar si no hay pieza
    {
        Console.WriteLine("Posición vacía " + tablero[j,
ColumnaDama]); //Se escribe posición vacía si no hay pieza
    }
    if (MTablero[j, ColumnaDama].tipoPieza != "N/P" && MTablero[j,
ColumnaDama].colorPieza != MTablero[FilaDama, ColumnaDama].colorPieza) //Se
creo la condición para validar lo que sucede si existe una pieza de color
distinto a la dama
    {
        Console.WriteLine(MTablero[j, ColumnaDama].tipoPieza + " en
la posición " + MTablero[j, ColumnaDama].posicionPieza); //Se imprime la
pieza y el color, solo llega hasta la posición donde se encuentre la pieza,
ya que, no puede ir más lejos
    }
}

```

```

        break;
    }
    else
    {
        if (MTablero[j, ColumnaDama].tipoPieza != "N/P") // Se creo
una condición para verificar si la pieza es del mismo color de la dama
        {
            break; //Se rompe el bucle, ya que, la dama no puede
comerse a esa pieza
        }
    }
}

int BordeC = ColumnaDama; //Se declara la variable y se inicializa
dándole el valor de ColumnaDama
int BordeF = FilaDama; //Se declara la variable y se inicializa
dándole el valor de ColumnaDama
Console.WriteLine("");
Console.WriteLine("Movimientos diagonales hacia la derecha hacia
abajo");
if (BordeC != 7 && BordeF != 7) //Se crea una condición para
verificar que la dama no se encuentre en el borde inferior derecho
{
    int DAbajo = ColumnaDama; //Se declara la variable y se
inicializa con ColumnaDama
    for (int j = FilaDama + 1; j < 8 && DAbajo < 7; j++) //Se
recorre la matriz para verificar las piezas que se encuentran de manera
diagonal hacia abajo hacia la derecha
    {
        DAbajo++; //Se incrementa la variable para ver los
movimientos de la dama
        if (MTablero[j, DAbajo].tipoPieza == "N/P") //Se crea una
condición para verificar si no hay pieza
        {
            Console.WriteLine("Posición vacía " + tablero[j,
DAbajo]); //Se escribe posición vacía si no hay pieza
        }
        if (MTablero[j, DAbajo].tipoPieza != "N/P" && MTablero[j,
DAbajo].colorPieza != MTablero[FilaDama, ColumnaDama].colorPieza) //Se creo
la condición para validar lo que sucede si existe una pieza de color
distinto a la dama
        {
            Console.WriteLine(MTablero[j, DAbajo].tipoPieza + " en
la posición " + MTablero[j, DAbajo].posicionPieza); //Se imprime la pieza y

```

el color, solo llega hasta la posición donde se encuentre la pieza, ya que, no puede ir más lejos

```

        break;
    }
    else
    {
        if (MTablero[j, DAbajo].tipoPieza != "N/P") // Se creo
una condición para verificar si la pieza es del mismo color de la dama
        {
            break; //Se rompe el bucle, ya que, la dama no puede
comerse a esa pieza
        }
    }
}
}
Console.WriteLine("");
Console.WriteLine("Movimientos diagonales hacia la derecha hacia
arriba");
if (BordeC != 7 && BordeF != 0) //Se crea una condición para
verificar que la dama no se encuentre en el borde superior derecho
{
    int DArriba = ColumnaDama; //Se declara la variable y se
inicializa con ColumnaDama
    for (int j = FilaDama - 1; j >= 0 && DArriba < 7; j--) //Se
recorre la matriz para verificar las piezas que se encuentran de manera
diagonal hacia arriba hacia la derecha
    {
        DArriba++; //Se incrementa la variable para ver los
movimientos de la dama
        if (MTablero[j, DArriba].tipoPieza == "N/P") //Se crea una
condición para verificar si no hay pieza
        {
            Console.WriteLine("Posición vacía " + tablero[j,
DArriba]); //Se escribe posición vacía si no hay pieza
        }
        if (MTablero[j, DArriba].tipoPieza != "N/P" && MTablero[j,
DArriba].colorPieza != MTablero[FilaDama, ColumnaDama].colorPieza) //Se creo
la condición para validar lo que sucede si existe una pieza de color
distinto a la dama
        {
            Console.WriteLine(MTablero[j, DArriba].tipoPieza + " en
la posición " + MTablero[j, DArriba].posicionPieza); //Se imprime la pieza y
el color, solo llega hasta la posición donde se encuentre la pieza, ya que,
no puede ir más lejos
            break;
        }
    }
}

```

```

    }
    else
    {
        if (MTablero[j, DArriba].tipoPieza != "N/P") // Se creo
una condición para verificar si la pieza es del mismo color de la dama
        {
            break; //Se rompe el bucle, ya que, la dama no puede
comerse a esa pieza
        }
    }
}

Console.WriteLine("");
Console.WriteLine("Movimientos diagonales hacia la izquierda hacia
abajo");
if (BordeC != 0 && BordeF != 0) //Se crea una condición para
verificar que la dama no se encuentre en el borde inferior izquierdo
{
    int IAbajo = ColumnaDama; //Se declara la variable y se
inicializa con ColumnaDama
    for (int j = FilaDama + 1; j < 8 && IAbajo > 0; j++) //Se
recorre la matriz para verificar las piezas que se encuentran de manera
diagonal hacia abajo hacia la derecha
    {
        IAbajo--; //Se decrementa la variable para ver los
movimientos de la dama
        if (MTablero[j, IAbajo].tipoPieza == "N/P") //Se crea una
condición para verificar si no hay pieza
        {
            Console.WriteLine("Posición vacía " + tablero[j,
IAbajo]); //Se escribe posición vacía si no hay pieza
        }
        if (MTablero[j, IAbajo].tipoPieza != "N/P" && MTablero[j,
IAbajo].colorPieza != MTablero[FilaDama, ColumnaDama].colorPieza) //Se creo
la condición para validar lo que sucede si existe una pieza de color
distinto a la dama
        {
            Console.WriteLine(MTablero[j, IAbajo].tipoPieza + " en
la posición " + MTablero[j, IAbajo].posicionPieza); //Se imprime la pieza y
el color, solo llega hasta la posición donde se encuentre la pieza, ya que,
no puede ir más lejos
            break;
        }
    }
    else

```

```

        {
            if (MTablero[j, IAbajo].tipoPieza != "N/P") // Se creo
una condición para verificar si la pieza es del mismo color de la dama
            {
                break; //Se rompe el bucle, ya que, la dama no puede
comerse a esa pieza
            }
        }
    }
}

Console.WriteLine("");
Console.WriteLine("Movimientos diagonales hacia la izquierda hacia
arriba");
if (BordeC != 0 && BordeF != 7) //Se crea una condición para
verificar que la dama no se encuentre en el borde inferior izquierdo
{
    int IArriba = ColumnaDama; //Se declara la variable y se
inicializa con ColumnaDama
    for (int j = FilaDama - 1; j >= 0 && IArriba > 0; j--) //Se
recorre la matriz para verificar las piezas que se encuentran de manera
diagonal hacia abajo hacia la derecha
    {
        IArriba--; //Se decrementa la variable para ver los
movimientos de la dama
        if (MTablero[j, IArriba].tipoPieza == "N/P") //Se crea una
condición para verificar si no hay pieza
        {
            Console.WriteLine("Posición vacía " + tablero[j,
IArriba]); //Se escribe posición vacía si no hay pieza
        }
        if (MTablero[j, IArriba].tipoPieza != "N/P" && MTablero[j,
IArriba].colorPieza != MTablero[FilaDama, ColumnaDama].colorPieza) //Se creo
la condición para validar lo que sucede si existe una pieza de color
distinto a la dama
        {
            Console.WriteLine(MTablero[j, IArriba].tipoPieza + " en
la posición " + MTablero[j, IArriba].posicionPieza); //Se imprime la pieza y
el color, solo llega hasta la posición donde se encuentre la pieza, ya que,
no puede ir más lejos
            break;
        }
    }
    else
    {

```

```

        if (MTablero[j, IArriba].tipoPieza != "N/P") // Se creo
una condición para verificar si la pieza es del mismo color de la dama
        {
            break; //Se rompe el bucle, ya que, la dama no puede
comerse a esa pieza
        }
    }
}

public void CrearTablero() //Se creo una función para poder crear el
tablero
{
    char[] letraColumna = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h' }; //
Se creo un arreglo de char para las letras del tablero
    for (int fila = 0; fila < 8; fila++) //El bucle recorre las filas
del tablero
    {
        for (int columna = 0; columna < 8; columna++) //El bucle recorre
las columnas del tablero
        {
            tablero[fila, columna] = letraColumna[columna] + (8 -
fila).ToString(); //Se sustituyen los números de las columnas por letras y
todo el tablero se convierte a string
        }
    }
}

public void MostrarTablero() // Se creo la función para mostrar lo que
sucede en el tablero
{
    for (int fila = 0; fila < 8; fila++) //Se recorren las filas del
tablero
    {
        for (int columna = 0; columna < 8; columna++) //Se recorren las
columnas del tablero
        {
            if (MTablero[fila, columna].colorPieza == "blanco") //Se
creo una condición para verificar si la pieza ingresada es de color blanco
            {
                Console.ForegroundColor = ConsoleColor.White; //Cambia
el color de letra a blanco
                Console.Write(MTablero[fila, columna].tipoPieza + " B "
+ "\t"); //Se imprime el tipo de pieza con el color en el tablero
            }
        }
    }
}

```

```
        if (MTablero[fila, columna].colorPieza == "negro") //Se creo
una condición para verificar si la pieza ingresada es de color negro
        {
            Console.ForegroundColor = ConsoleColor.Black; //Cambia
el color de letra a negro
            Console.Write(MTablero[fila, columna].tipoPieza + " N "
+ "\t"); //Se imprime el tipo de pieza con el color en el tablero
        }
        if (MTablero[fila, columna].tipoPieza == "N/P") //Se creo
una condición para verificar si la pieza ingresada no es de color blanco o
negro
        {
            Console.ForegroundColor = ConsoleColor.DarkBlue;
//Cambia el color de letra a gris
            Console.Write(MTablero[fila, columna].tipoPieza + " " +
"\t"); //Se imprime el tipo de pieza en el tablero
        }
    }
    Console.ForegroundColor = ConsoleColor.Black; //Deja el color
negro como predeterminado
    Console.WriteLine();
}
}
```