**Estructura del Código**

El proyecto está organizado en archivos separados para mantener una estructura clara y modular:

1. **main.cpp**: Archivo principal que controla el flujo del juego.
2. **Tablero.h / Tablero.cpp**: Clase que gestiona el estado del tablero.
3. **IA\_Minimax.h / IA\_Minimax.cpp**: Clase que implementa el algoritmo Minimax con poda alfa-beta.

**Explicación de las Clases**

**1. Clase Tablero**

Esta clase representa el tablero del juego y gestiona todas las operaciones relacionadas con él.

* **Atributos**:
  + tablero[3][3]: Matriz que almacena el estado del tablero. Los valores pueden ser:
    - 'X': Representa al jugador humano.
    - 'O': Representa a la máquina.
    - ' ': Representa una celda vacía.
* **Métodos principales**:
  + mostrar(): Imprime el tablero en la consola.
  + hayMovimientos(): Verifica si quedan celdas vacías en el tablero.
  + evaluar(): Evalúa el estado actual del tablero y devuelve:
    - 10 si la máquina gana.
    - -10 si el humano gana.
    - 0 si no hay un ganador aún.
  + realizarMovimiento(fila, columna, jugador): Coloca una ficha en la posición indicada si es válida.
  + deshacerMovimiento(fila, columna): Elimina una ficha de una posición específica (para backtracking).

**2. Clase IA\_Minimax**

Esta clase implementa el algoritmo **Minimax con poda alfa-beta** para decidir el mejor movimiento para la máquina.

* **Métodos principales**:
  + minimax(tablero, profundidad, esMaximizador, alfa, beta): Realiza la búsqueda del mejor movimiento utilizando:
    - **Alfa**: El mejor puntaje que el jugador maximizador puede garantizar en ese momento.
    - **Beta**: El mejor puntaje que el jugador minimizador puede garantizar.
    - Poda ocurre si alfa >= beta.
  + mejorMovimiento(tablero): Itera sobre todas las posibles jugadas de la máquina, evalúa cada una con minimax y devuelve la posición óptima como un par (fila, columna).

**3. Archivo main.cpp**

Controla el flujo del juego, alternando entre los turnos del humano y la máquina.

* **Flujo del programa**:
  1. Inicializa un tablero vacío.
  2. Alterna entre los turnos del humano y la máquina:
     + El humano ingresa las coordenadas de su movimiento.
     + La máquina utiliza el método mejorMovimiento() de la clase IA\_Minimax para jugar.
  3. Finaliza cuando:
     + Algún jugador gana.
     + No hay más movimientos posibles (empate).
  4. Muestra el resultado del juego.

**Algoritmo Minimax con Poda Alfa-Beta**

* **Idea básica**: El algoritmo evalúa todas las posibles configuraciones del tablero y selecciona la jugada que maximiza las posibilidades de ganar (para la máquina) y minimiza las opciones del oponente.
* **Optimización con poda alfa-beta**:
  + Durante la evaluación del árbol de búsqueda, se eliminan ramas que no pueden mejorar el resultado actual.
  + Reduce significativamente la complejidad computacional del algoritmo.