



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Facultad de Ingeniería en Electricidad y Computación**  
**“DESARROLLO DE APLICACIONES WEB AVANZADAS CON AJAX”**

**TESINA DE SEMINARIO**

Previo a la obtención del Título de:

**INGENIERO EN CIENCIAS COMPUTACIONALES**  
**ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS**

Autores:

**LUIS MANUEL TOAPANTA WILCHES**

**HEIDI ARACELY SALAZAR PATIN**

**GUAYAQUIL – ECUADOR**

## **DEDICATORIA**

A Dios quien me ha ayudado e inspirado a tomar buenas decisiones a lo largo de mi vida.

A mis padres, quienes me han apoyado constantemente en todo, en la medida de sus posibilidades. A mis hermanos por ser mi ejemplo a seguir.

**HEIDI ARACELY SALAZAR PATÍN**

Le dedico este proyecto a mi hermana Elizabeth.

**LUIS MANUEL TOAPANTA WILCHES**

## **AGRADECIMIENTO**

Agradezco a Dios por guiarme todos los días y a mi familia por apoyarme a lo largo de mi carrera. A mi Facultad de Ingeniería en Electricidad y Computación de la ESPOL por prepararme para ser una buena profesional. A mi compañero de proyecto Luis Toapanta por su colaboración y empuje en la realización de esta materia de grado.

Un agradecimiento especial a los Ingenieros Carlos Martín y Daniel Ochoa por haberme guiado con sus conocimientos y consejos para hacer posible la culminación de nuestro Proyecto de Grado.

**HEIDI ARACELY SALAZAR PATÍN**

## **AGRADECIMIENTO**

Agradezco a mi familia, especialmente a mis tíos Rosario, Christian y Matilde quienes me apoyaron en la carrera y a mi hermana Elizabeth que me compró mi primera computadora y siempre me dio apoyo moral; a mi amigo Mario que me enseñó a programar en lenguaje C, a mis amigos y compañeros de la universidad, y a todos aquellos que de alguna forma me apoyaron y guiaron con sus consejos y sugerencias a nivel personal y académico. Agradezco también a la ESPOL y a la FIEC por su impulso al conocimiento de sus estudiantes.

De corazón os lo agradezco

**LUIS MANUEL TOAPANTA WILCHES**

## **TRIBUNAL DE SUSTENTACIÓN**

---

Ing. Carlos Martín

PROFESOR DEL SEMINARIO

---

Ing. Daniel Ochoa

PROFESOR DELEGADO DEL DECANO

## DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Trabajo de Grado, nos corresponden exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL)

---

Luis M. Toapanta Wilches

---

Heidi A. Salazar Patín

## RESUMEN

El presente informe describe el desarrollo de una versión en línea de uno de los juegos de mesa más populares, el cual es conocido como el deporte ciencia: El Ajedrez.

Es reconocido que el ajedrez fomenta con su práctica algunas habilidades tales como la concentración, la imaginación, la creatividad, la capacidad de cálculo, etc.. La ESPOL promueve este deporte y cuenta con un club y una sala de juego equipada para este fin.

Este desarrollo se realizó, por la necesidad del club de ajedrez de la ESPOL de contar con una aplicación propia que permita realizar una buena estimación de nivel relativo entre los jugadores.

Como sabemos el ajedrez ha sido un antiguo juego-ciencia, que viene desde tiempos muy remotos y que hace poco el hombre se limitaba a estar solo sentado frente a un cuadriculado tablero, viendo sus movimientos y lances en el transcurso del juego.

Hoy en día el ajedrez ha ido evolucionando de tal manera que en la actualidad ya existen programas que ayudan a mejorar las técnicas de juego y comprobar

nuestros avances en el mismo, pero siempre llevado de un buen plan que permita desarrollar nuestras ideas y evitar que el contrincante gane la partida.

Este juego permite mejorar poco a poco la comprensión de lectura, cosa que lo ayudará en el desarrollo de sus estudios escolares y sin duda alguna este juego acompañará al hombre por mucho tiempo más, ya sea como simple distracción en los momentos libres o bien visto como un verdadero arte en donde se puede observar el afán eterno del hombre en la búsqueda de la perfección.



## ÍNDICE GENERAL

<b>TRIBUNAL DE SUSTENTACIÓN</b>	<b>IV</b>
<b>DECLARACIÓN EXPRESA</b>	<b>V</b>
<b>RESUMEN</b>	<b>VI</b>
<b>ÍNDICE GENERAL</b>	<b>VIII</b>
<b>INDICE DE FIGURAS Y GRÁFICOS</b>	<b>XI</b>
<b>ABREVIATURAS</b>	<b>XII</b>
<b>INTRODUCCIÓN</b>	<b>XIV</b>
<b>CAPITULO I</b>	<b>1</b>
<b>1. ANTECEDENTES Y JUSTIFICACIÓN</b>	<b>1</b>
1.1. Descripción del problema	1
1.2. Objetivos	2
1.2.1. Objetivos generales	2

1.2.2. Objetivos específicos	2
<b>1.3. Justificación</b>	<b>3</b>
<b>1.4. Alcance</b>	<b>4</b>
<b>1.5. Limitaciones</b>	<b>6</b>
<b>CAPITULO II</b>	<b>7</b>
<b>2. EL AJEDREZ Y LOS JUEGOS EN LÍNEA</b>	<b>7</b>
<b>2.1. Perspectiva histórica de los juegos de ajedrez en computador</b>	<b>7</b>
2.1.1. Estrategia contra fuerza bruta	8
2.1.2. Computadoras contra humanos	13
<b>2.2. Algoritmo para calculo de ELO</b>	<b>13</b>
<b>CAPITULO III</b>	<b>17</b>
<b>3. ANÁLISIS DE REQUERIMIENTOS Y DISEÑO</b>	<b>17</b>
<b>3.1. Definición del diseño</b>	<b>17</b>
3.1.1. Requerimientos Funcionales	17
3.1.2. Requerimientos NO Funcionales	21
<b>3.2. Diseño de la aplicación e interacción del juego</b>	<b>22</b>
3.2.1. Arquitectura de la Aplicación	22

<b>3.3. Herramientas de desarrollo</b>	<b>24</b>
<b>3.4. Herramienta de Software</b>	<b>24</b>
3.4.1. Visual Studio 2010	24
3.4.2. SQL Server 2008	25
3.4.3. Ajax Control Toolkit	27
3.4.4. PokelIn	27
3.4.5. Entity FrameWork	28
3.4.6. JQuery	29
<b>CAPITULO IV</b>	<b>30</b>
<b>4. IMPLEMENTACIÓN DEL SISTEMA</b>	<b>30</b>
<b>4.1. El tablero y las piezas</b>	<b>30</b>
<b>4.2. El envío de información desde el servidor hacia los clientes</b>	<b>31</b>
<b>4.3. Los Movimientos y el reloj</b>	<b>32</b>
<b>4.4. Las distintas partidas y la concurrencia</b>	<b>34</b>
<b>CONCLUSIONES Y RECOMENDACIONES</b>	<b>38</b>
<b>ANEXO A.- REGLAS DEL JUEGO DE AJEDREZ Y EVOLUCIÓN DEL AJEDREZ</b>	<b>42</b>

<b>ANEXO B.- TECNOLOGÍA UTILIZADA Y CARACTERÍSTICAS</b>	<b>60</b>
<b>REFERENCIA BIBLIOGRÁFICA</b>	<b>70</b>

## **INDICE DE FIGURAS Y GRÁFICOS**

**Figura 3.2.1:** Arquitectura de Sistema “*Modelo 3 – Tier*”

**Figura 3.4.4:** Diagrama Pokeln

**Figura 4.1.:** Diagrama de clases

**Figura 1.** Posición inicial de las piezas

**Figura a:** El Alfil

**Figura b:** La torre

**Figura c:** Dama o Reyna

**Figura d:** El Caballo

**Figura e:** El Peón

**Figura f:** Captura “Al paso”

**Figura i:** Desplazamiento del Rey

**Figura ii:** Enroque del Rey

**Figura iii:** Top 100 Raking de Jugadores de Enero 2011

## ABREVIATURAS

Las abreviaturas presentadas en la siguiente tesis son las siguientes:

**AJAX:** Asynchronous JavaScript and XML.

**OMG:** Object Management Group

**JQUERY:** Librería o framework de Javascript.

**FSM:** Finite State Machines

**JSP:** Java Server Pages

**JSON:** JavaScript Object Notation

**XML:** Extensible Markup Language

**PGN:** Portable Game Notation

**FIDE:** Federación Internacional de Ajedrez (por su nombre en francés:  
Fédération Internationale des Échecs)

**USCF:** Federación de Ajedrez de Estados Unidos (por su nombre en  
inglés: United States Chess Federation)



## INTRODUCCIÓN

El ajedrez, considerado por muchos como el deporte ciencia, tiene una gran acogida en la ESPOL. La universidad ha ganado medallas de oro en algunos torneos interuniversitarios y esto se debe en gran parte al entrenamiento continuo que realizan sus estudiantes, apoyados por la institución.

El entrenador de ajedrez de la ESPOL considera adecuado contar con una aplicación propia para realizar juegos de ajedrez en línea entre los estudiantes de la universidad, con el fin de poder tener una buena estimación del nivel relativo entre los jugadores aunque no se encuentren registrados en el Ranking de la FIDE, y propiciar el entrenamiento continuo antes mencionado.

Los juegos en línea son aquellos que pueden ser ejecutados desde un navegador web y requieren de una conexión a Internet. En la mayoría de los casos los juegos en línea son programas gratuitos que pueden ser usados por un tiempo ilimitado y están disponibles libremente.

Hay algunos sistemas de juego de ajedrez en línea, los cuales son usados por la comunidad politécnica, pero tienen algunas limitaciones como el

exceso de publicidad en las versiones gratuitas o el costo de suscripción en algunos casos.

Una ventaja de tener un sistema propio es que se puede construir estadísticas a la medida, de tal forma que el entrenador de juego pueda mejorar su trabajo con los estudiantes y a su vez ellos puedan comprobar sus avances en el juego.



# **CAPITULO I**

## **1. ANTECEDENTES Y JUSTIFICACIÓN**

### **1.1. Descripción del problema**

La ESPOL a través de sus estudiantes participa activamente en torneos de ajedrez a nivel nacional, de manera personal y como club. El Club ESPOL ha obtenido algunas medallas en competiciones interuniversitarias.

Esto se ha conseguido gracias al impulso que la universidad ha realizado, ya que cuenta con una sala de Ajedrez en la Biblioteca de Ingenierías y con un entrenador a cargo de dicha sala.

Es necesario que los jugadores de la universidad puedan realizar partidas de entrenamiento entre ellos para participar en torneos, por ende la sala de ajedrez dispone de varios tableros y relojes para cumplir este

propósito. Sin embargo, es difícil realizar el seguimiento del progreso de cada estudiante dado que algunos de ellos no pueden asistir a la sala todos los días, y cuando juegan no acostumbran a registrar los movimientos realizados en cada partida, siendo difícil su análisis posterior. Tampoco se tiene una medición del nivel relativo de juego entre los distintos jugadores, por lo que no se puede saber con precisión qué jugadores son mejores que otros.

## **1.2. Objetivos**

### **1.2.1. Objetivos generales**

- Desarrollar un sistema para jugar partidas en línea.
- Ayudar al entrenador en la evaluación del progreso de cada jugador.

### **1.2.2. Objetivos específicos**

- Registrar a los estudiantes de ESPOL que quieran aprender o practicar el ajedrez.
- Registrar los movimientos de cada partida jugada en línea.
- Implementar un procedimiento que nos permita conocer el nuevo nivel de juego relativo al final de cada partida.
- Permitir las consultas por parte del entrenador y de los jugadores en general.

### **1.3. Justificación**

Es conocido que el ajedrez se viene jugando desde hace siglos y que se han ideado algunos modos de representar gráficamente las piezas y anotar las jugadas con el fin de preservar problemas y partidas.

Gracias a las pruebas y errores de muchos jugadores en sus respectivas partidas, y al análisis posterior a ellas por parte de maestros y entrenadores, se han conseguido muchos de los progresos en la comprensión del juego. El análisis posterior de las partidas propias es muy importante para mejorar el nivel de juego de cada participante, y en esto coinciden muchos entrenadores.

Debido a la emoción del juego algunas veces los jugadores anotan mal sus jugadas en la planilla destinada para ello. Además, el trabajo de ingresar las planillas a la PC para poder compartir las partidas con amigos o compañeros de juego no es una tarea agradable para muchos jugadores.

De igual manera, tener una medida del nivel de fuerza de juego relativo no es una tarea trivial si no se cuenta con un sistema que pueda medir dicho nivel en función de las partidas jugadas por los mismos.

Un sistema que permita a los jugadores entrenar con sus compañeros desde cualquier lugar dentro de la ESPOL o del mundo, y que al mismo tiempo registre automáticamente sus movimientos y que les otorgue o les quite puntos en función del resultado de cada partida y del oponente al que se enfrentan, es una necesidad para cualquier club de ajedrez.

Cabe mencionar que ya existen sistemas similares que algunos estudiantes de ESPOL utilizan, pero siendo sistemas de empresas particulares, no se puede acceder a consultas particulares por parte del entrenador, además, no se tienen garantías de que aquellos sistemas funcionarán durante mucho tiempo.

#### **1.4. Alcance**

El sistema tiene las siguientes opciones:

- Permite registrar a un miembro de la comunidad politécnica.
- Permite desafiar a la comunidad o a un miembro específico a jugar una partida.
- Permite a los jugadores configurar las opciones antes de cada partida. Entre las opciones se tienen:
  - Tiempo de juego
  - Color de las piezas (blancas, negras o alternadas)
  - Con puntuación o sin puntuación
  - Guardar la partida automáticamente

- Guarda la configuración de opciones del usuario de tal forma que se establezcan como valores por defecto para que así el mismo usuario pueda jugar desde diferentes computadoras con las mismas opciones.
- Calcular el nuevo puntaje del jugador al finalizar cada partida con puntuación.
- Permitir consultar estadísticas personales como el número de partidas ganadas, perdidas y empatadas; y el máximo puntaje obtenido históricamente.
- Compartir partidas finalizadas por correo electrónico o tal vez en redes sociales más usadas.
- Consultar las partidas guardadas, permitir reproducirlas y exportarlas usando el formato PGN #, de tal forma que puedan abrirse en algún programa externo de análisis de partidas, con el fin de que el usuario interesado en mejorar su nivel, y/o el entrenador puedan estudiar la misma.
- Controlar la desconexión de los jugadores, por ejemplo, otorgándole la victoria a un jugador en caso de que su oponente se desconecte durante la partida.
- Permitir al administrador consultar estadísticas:
  - Gráfico con el número de partidas por fecha.
  - Número de jugadores por rangos de puntuación.
  - Top N de jugadores por puntuación.

- Jugadores con más partidas jugadas, ganadas y perdidas

### **1.5. Limitaciones**

- El máximo número de jugadores que pueden conectarse antes de la denegación de servicio no ha sido estimado, por lo que inicialmente el registro debe ser realizado por el entrenador.
- El reloj de ajedrez simulado por el sistema no será tan preciso como un reloj de ajedrez físico.

## **CAPITULO II**

### **2. EL AJEDREZ Y LOS JUEGOS EN LÍNEA**

La aparición del ajedrez por correspondencia fomentó enormemente el desarrollo del juego, facilitando nuevas aperturas y defensas. Esta modalidad de juego consiste en que los jugadores transmiten sus movimientos a través del correo. En un encuentro de ajedrez se enviaban las cartas con las jugadas por medio de un carruaje y esto se extendía notablemente en el tiempo, pudiendo llegar a durar varios años la disputa de las mismas. Todas las jugadas eran publicadas en los periódicos.

#### **2.1. Perspectiva histórica de los juegos de ajedrez en computador**

Antes de la era de internet y de los juegos en línea, en el siglo XVIII se difundió la idea de crear una computadora que sea capaz de jugar al ajedrez, aunque esto no era totalmente cierto en el año 1979, pero en 1912 el español Leonardo Torres y Quevedo construyó un autómata capaz de jugar al ajedrez, llamado El Ajedrecista, aunque esto no fue lo suficientemente eficaz, de tal manera que en la década de los 50 apareció

la computadora y desde entonces la ingeniería informática han construido maquinas y programas que juegan al ajedrez.

Actualmente, hay numerosos programa o software libre (GNU Chess, Amy, o Crafty) que pueden jugar al ajedrez en cualquier ordenador personal y derrotar a jugadores profesionales, bajo condiciones de torneo de igual manera hay otros programas mejores (Shredder, Fritz, Rybka o Fruit) han vencido a muchos jugadores campeones.

Una de causas que motivó la existencia del ajedrez computarizado fue el entretenimiento propio, también tenerlo como herramienta o análisis, para competiciones entre computadoras de ajedrez y como investigación del conocimiento humano.

En algunos juegos de estrategia las computadoras suelen vencer las partidas, mientras que en otros, los principiantes vencen a las máquinas. En el ajedrez, el resultado de la fusión de las habilidades de los expertos, con los programas de ajedrez, es mayor que el de cualquiera de los dos a solas.

### **2.1.1. Estrategia contra fuerza bruta**



Antes de los juegos de ajedrez por computadora, se predijo las dos posibles principales formas de búsqueda de cualquier programa, a las que nombró de 'Tipo A', y 'Tipo B'.

Los programas 'Tipo A', utilizarían una búsqueda basada en la " Fuerza bruta", los cuales examinarían todas las posibles posiciones o movimientos usando el algoritmo minimax, aunque esto no fue muy práctico por dos razones:

Primero, con 30 movimientos posibles en una posición típica, Shannon predijo que buscando las 306 (más de 700.000.000) posiciones contenidas en los primeros tres movimientos, tardaría cerca de 16 minutos y aún sienta optimista que el programa evaluara un millón de posiciones por segundo. Después de esta conjetura, se tardó alrededor de 40 años para conseguir esa velocidad.

Segundo, se ignoraba el problema de la latencia (suma de retardos temporales dentro de una red), ya que el programa evalúa la posición resultante después de todo el intercambio de piezas ocurrido durante todos esos movimientos al final de cada partida. Los programas de 'Tipo A' funcionan así, pero el inconveniente es que se incrementa enormemente el número de posiciones

necesarias para el análisis, y de este modo el programa se ralentizaba (proceso lento) todavía más.

De este modo se sugirió que los programas “Tipo B” utilizarían una especie de “inteligencia artificial estratégica” para analizar solo las mejores jugadas de cada posición, en un tiempo razonable muy similar a lo que hacen los jugadores humanos, pero el problema de los programas 'Tipo B' es que se confía en que el programa puede decidir qué movimientos son buenos para cualquier posición, siendo un problema para los programas 'Tipo A'.

Un gran defensor de las computadoras de ajedrez fue el Campeón del mundo de ajedrez Mijail Botvinnik, trabajó con hardware a principios de los años 1960, en ese momento los ordenadores más potentes podían conseguir tres plies<sup>[1]</sup> (movimientos) por búsqueda, pero Botvinnik no tenía tales máquinas.

En 1973, la Universidad de Northwestern, creadora del programas de Tipo B, se pasó a los programas de Tipo A, porque encontraban a los programas de Tipo B poco estimulantes durante los torneos, y así era difícil de predecir lo que van a mover. Otra razón fue que en los programas de Tipo A era mucho más fácil detectar los fallos del

programa y depurarlos, haciendo de esto un programa más rápido: en el tiempo que solían tomar para decidir los movimientos.

De hecho, Chess 4.0 estableció un paradigma que era y continúa utilizándose en todos los programas de ajedrez actuales. Los programas tipo Chess 4.0 ganaban por sus programas que jugaban un mejor ajedrez. Tales programas no intentaban imitar los procesos de pensamiento humanos, pero confiaban completamente en búsquedas alfa-beta<sup>[2]</sup> y negascout<sup>[3]</sup>. Muchos de tales programas incluyendo los actuales, incluyen una parte selectiva bastante limitada de la búsqueda basada en búsquedas latentes que eran lanzadas en ciertas condiciones en un intento de eliminar o reducir los movimientos malos o investigar nodos interesantes.

El desarrollo y los avances tecnológicos hicieron que el sistema de fuerza bruta continuara en alza y se intensificara mucho más en los años 90. El resultado ha sido la creación de programas mucho más sólidos, con una IA táctica asombrosa, sin apenas errores, y conducidos hacia el límite de su profundidad de búsqueda. Esto ha producido resultados extraordinarios, por lo menos en lo referente al ajedrez, dejando que las computadoras hagan lo que mejor saben hacer, calcular en vez de intentar emular la inteligencia y conocimiento humanos. En 1997, Deep Blue, una computadora de

Tipo A, derrotó por primera vez al Campeón del Mundo Garry Kasparov.

Sin embargo, a finales de los años 1990, los programadores prefirieron los programas de Tipo B, sustituyendo a los de Tipo A.

En 1998 se publica Rebel 10, un programa comercial de Tipo B, quien derrotó a Viswanathan Anand, y se proclamó el segundo motor de ajedrez más fuerte del mundo aquel año, pero cabe decir que de las cuatro partidas de ajedrez rápido que se jugaron, Rebel ganó 3 de ellas, en las dos partidas semirrápidas, quedaron 1.5-0.5 a favor de Rebel, y en la partida con tiempo de control más largo, fue Anand quien venció. De esto se pudo deducir que las computadoras juegan mejor que los humanos en tiempos de control más rápidos, pero que la fuerza de los jugadores se mide con tiempos de control más largos, donde Anand demostró que los humanos siguen siendo mejores.

En el 2005, Hydra, una computadora de ajedrez del Tipo B, volvió a derrotar al mejor jugador británico y séptimo mejor clasificado del mundo, Michael Adams.

### **2.1.2. Computadoras contra humanos**

En 1996, la computadora Deep Blue de IBM ganó a Garry Kasparov en tiempos de control de torneo, con ritmo de juego lento.

IBM desmanteló a Deep Blue después del match y no ha vuelto a jugar desde entonces. Sin embargo, se han seguido jugando matches entre humanos y computadoras. Con el incremento de la potencia de procesado, los programas de ajedrez ejecutándose en ordenadores empiezan a superar a los jugadores más fuertes. Al menos en las partidas rápidas las computadoras juegan mejor que los humanos pero en controles de tiempo clásicos, en los que se determina la clasificación de un jugador, la ventaja no está tan clara.

### **2.2. Algoritmo para calculo de ELO**

El sistema de puntuación Elo es un método para calcular la fuerza relativa de los jugadores de juegos como el ajedrez. Fue inventado por el profesor Árpád Élő para mejorar el sistema de clasificación vigente de los jugadores de ajedrez. La USCF lo adoptó en 1960 y La FIDE en 1970.

El rendimiento de los jugadores no se puede medir de forma absoluta, solo se puede deducir de las victorias, derrotas y empates contra otros jugadores. La puntuación de un jugador depende de las puntuaciones de sus oponentes, y los resultados obtenidos en su contra. La diferencia relativa en la puntuación entre dos jugadores determina una estimación de

los resultados esperados entre ellos. Elo sugirió establecer las puntuaciones de manera que una diferencia de 200 puntos en el ajedrez podría significar que el jugador más fuerte tiene una puntuación esperada de aproximadamente 0.75.

La puntuación que se espera del jugador A es su probabilidad de ganar, más la mitad de su probabilidad de empatar. Así, una puntuación esperada de 0.75 podría representar 75% de probabilidades de ganar, el 25% de posibilidades de perder, y 0% de probabilidad de empatar. En el otro extremo que podría representar un 50% de posibilidades de ganar, 0% de probabilidad de perder, y el 50% de probabilidad de empatar. La probabilidad de empatar, en lugar de tener un resultado decisivo, no se especifica en el sistema Elo, En cambio un empate se considera una media victoria y una media pérdida.

Si el jugador A tiene fuerza  $R_A$  y el jugador B tiene fuerza de  $R_B$ , la fórmula exacta (usando la curva logística ) para la calificación esperada del Jugador A es:

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}}.$$

Del mismo modo el puntaje esperado para el jugador B es:

$$E_B = \frac{1}{1 + 10^{(R_A - R_B)/400}}.$$

Esto también podría ser expresada por:

$$E_A = \frac{Q_A}{Q_A + Q_B}$$

y

$$E_B = \frac{Q_B}{Q_A + Q_B}$$

En este último caso, el denominador se aplica a ambas expresiones, es decir que solo mediante el estudio de los numeradores, sabremos el resultado esperado para el jugador A que es  $Q_A/Q_B$  veces mayor que la puntuación esperada para el jugador B. En conclusión se deduce que por cada 400 puntos de rating de ventaja sobre el oponente, la oportunidad de ganar es diez veces más en comparación con la oportunidad del oponente a ganar.

Cuando en un torneo los marcadores reales de un jugador superan los resultados esperados, el sistema Elo toma esto como evidencia de que la puntuación del jugador es muy baja, y necesita ser ajustada hacia arriba. De igual manera, cuando los marcadores están por debajo de los resultados esperados, la puntuación de ese jugador se ajusta a la baja. La sugerencia inicial de Elo, la cual es aún ampliamente usada, fue un simple ajuste lineal proporcional a la cantidad excedente o faltante respecto a su resultado esperado. El máximo ajuste por juego (llamado el valor K) se fijó en  $K = 16$  para maestros y  $K = 32$  para los jugadores más débiles.

La FIDE usa los siguientes rangos para el factor K:

- $K = 25$  para un jugador nuevo en la lista hasta que haya completado eventos con un total de al menos 30 juegos.
- $K = 15$  tanto como la puntuación del jugador permanece por debajo de 2400.
- $K = 10$  una vez que la puntuación publicada ha alcanzado los 2400 puntos y el ha completado un total eventos con un total de al menos 30 juegos. Luego de esto,  $K$  permanece constante con un valor de 10.

Suponiendo que del jugador A se esperaba un marcador total de  $E_A$ , pero actualmente obtiene  $S_A$  puntos. La fórmula para actualizar su puntuación es:

La actualización puede ser realizada después de cada juego o cada torneo, o cualquier período de de puntuación aplicable.



## **CAPITULO III**

### **3. ANÁLISIS DE REQUERIMIENTOS Y DISEÑO**

#### **3.1. Definición del diseño**

Al utilizar herramientas nuevas o desarrolladas por otras compañías hemos elegido para un óptimo desarrollo del proyecto utilizar librerías que nos permitan una interfaz más sencilla en el manejo de las coordenadas de movimiento.

Entre los requerimientos que se cumplirán dentro del alcance del juego se tiene:

- Requerimientos Funcionales
- Requerimientos NO Funcionales

#### **3.1.1. Requerimientos Funcionales**

Los requerimientos funcionales definen cuales son las funciones que el jugador necesita realizar a medida que vaya ejecutando el juego. Estos requerimientos son:

- Ingresar al Juego

- Desafiar a Jugar
- Configurar opciones del Juego
- Guardar Configuración de Juego
- Obtener nuevo Puntaje
- Consultar Estadísticas Personales de Juego
- Consultar Partidas Finalizadas
- Consultar Partidas Guardadas
- Exportar Partidas Guardadas
- Controlar la desconexión de los Jugadores
- Consultar gráficamente partidas por fecha
- Consultar Jugadores por rangos de puntuación
- Consultar Jugadores por Puntuación

### **Ingresar al Juego**

Permite registrar a un miembro de la comunidad politécnica mediante un identificador de usuario de ESPOL a ingresar al juego.

### **Desafiar a Jugar**

Permite desafiar a la comunidad o a un miembro específico a jugar una partida.

### **Configurar opciones del Juego**

Permite a los jugadores configurar las opciones antes de cada partida. Entre las opciones se tienen:

- Tiempo de juego
- Color de las piezas (blancas o negras)
- Con puntuación o sin puntuación
- Guardar la partida automáticamente

### **Guardar Configuración de Juego**

Guarda la configuración de opciones del usuario de tal forma que se establezcan como valores por defecto para que así el mismo usuario pueda jugar desde diferentes computadoras con las mismas opciones.

### **Obtener nuevo Puntaje**

Calcular el nuevo puntaje del jugador al finalizar cada partida con puntuación.

### **Consultar Estadísticas Personales de Juego**

Permite consultar estadísticas personales como el número de partidas ganadas, perdidas y empatadas; y el máximo puntaje obtenido históricamente.

**Consultar Partidas Finalizadas**

Compartir partidas finalizadas por correo electrónico o tal vez en redes sociales más usadas.

**Consultar Partidas Guardadas**

Para consultar las partidas guardadas se deberá reproducir y exportar las partidas usando el formato PGN #, de tal forma que puedan abrirse en algún programa externo de análisis de partidas, con el fin de que el usuario interesado en mejorar su nivel, y/o el entrenador puedan estudiar la misma.

**Controlar la desconexión de los Jugadores**

Controlar la desconexión de los jugadores, por ejemplo, otorgándole la victoria a un jugador en caso de que su oponente se desconecte durante la partida.

**Consultar partidas por fecha**

Permitirá consultar por medio de un gráfico el número de partidas vs. fecha

**Consultar rangos de puntuación.**

Este requerimiento permitirá consultar el número de jugadores

### **Consultar Jugadores por Puntuación**

Permitirá consultar el Top N de jugadores por puntuación

#### **3.1.2. Requerimientos NO Funcionales**

Los requerimientos NO funcionales influyen en la operatividad del sistema. Para el desarrollo del juego consideramos los siguientes:

##### **Rendimiento y Escalabilidad**

El sistema está en capacidad de mejorar el desarrollo de nuevas funcionalidades, como permitir consultar las partidas guardadas, reproducirlas y exportarlas usando el formato PGN (<http://www.chessclub.com/help/PGN-spec>), de tal forma que puedan abrirse en algún programa externo de análisis de partidas, con el fin de que el usuario interesado en mejorar su nivel pueda estudiar la misma.

##### **Normas técnicas y Reglas del Juego**

Este requerimiento es otra parte importante dentro del mismo. Son las que se deben cumplir para saber si al momento de jugar se realiza las jugadas correctamente, según sus técnicas y reglas. Indica también el tiempo que utiliza cada jugador, para llevar un mejor control de movimiento.

## **Apariencia**

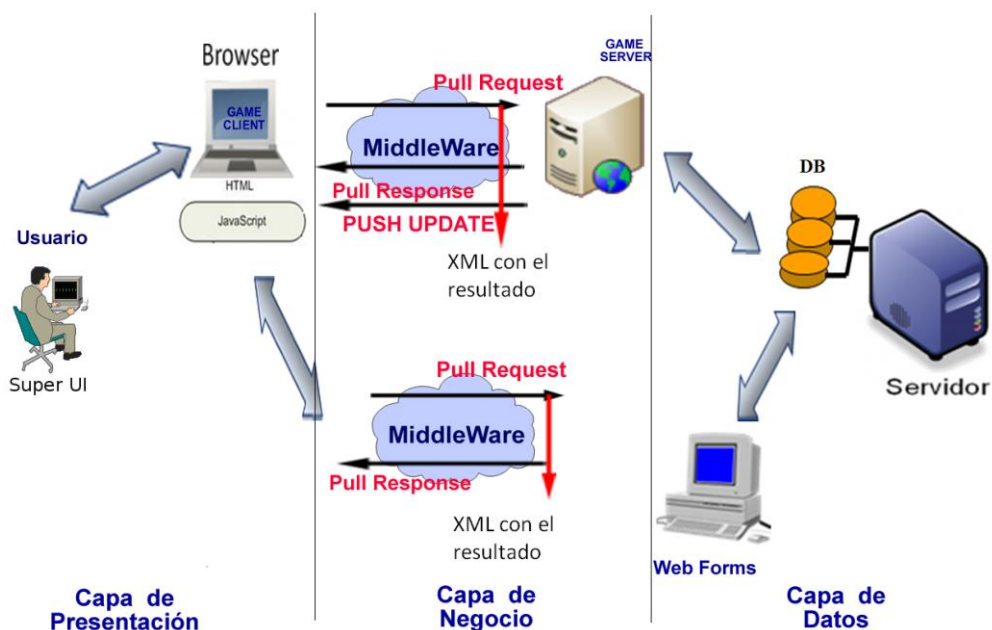
El juego es una interfaz realmente de uso intuitiva, sencilla y gráfica. Debe usar los colores adecuados. Como se trata de un juego de ajedrez para la comunidad politécnica, utilizar colores bajos pero sin dejar a un lado el contraste y brillo de los mismos. No se deben de utilizar colores fuertes y oscuros.

## **3.2. Diseño de la aplicación e interacción del juego**

### **3.2.1. Arquitectura de la Aplicación**

La arquitectura que utilizamos en nuestro sistema es el “Modelo de tres Capas” con una conexión abierta para que puedan interactuar los participantes junto con el servidor.

Aquí detallamos los procesos que se realizarán en cada capa, y se muestra la arquitectura del sistema en la figura 3.2.1.



**Figura 3.2.1** Arquitectura de Sistema “Modelo 3 - Tier” [6]

La capa de entrada se define como los datos de entrada al juego, es decir cuando el jugador realiza una jugada o movimiento de una pieza. Luego, este movimiento es cargado y pasan a la siguiente capa de procesamiento, sin cerrar la conexión del servidor.

La capa de procesamiento carga los objetos en la escena incluyendo los datos del usuario, controla el tiempo de duración del juego, guarda la puntuación que obtuvo en el juego, maneja los eventos que ocurren en la escena y envía los datos a la última capa.

La capa de salida muestra la escena que está siendo procesada por la capa anterior, guarda en la base de datos el puntaje obtenido en la partida de ambos competidores.

### **3.3. Herramientas de desarrollo**

En este capítulo trataremos las nuevas herramientas que utilizamos para el desarrollo de nuestro proyecto. Estas herramientas fueron seleccionadas con la finalidad de manejar librerías que nos permita un manejo más sencillo y eficiente para el programador y para tener una buena interacción con el usuario, permitiendo una interfaz más sencilla.

### **3.4. Herramienta de Software**

Dentro de las herramientas que se utilizaron en el desarrollo del proyecto se incluyen las siguientes:

- Visual Studio 2010
- Sql Server 2008
- AjaxControlToolkit
- Pokeln
- Entity FrameWork
- JQuery

#### **3.4.1. Visual Studio 2010**

Permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporta la



plataforma.NET. Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

Visual Studio 2010 es la versión más reciente de esta herramienta, acompañada por .NET Framework 4.0.

Entre sus más destacables características, se encuentran la capacidad para utilizar múltiples monitores, así como la posibilidad de desacoplar las ventanas de su sitio original y acoplarlas en otros sitios de la interfaz de trabajo.

### 3.4.2. SQL Server 2008

Es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o PostgreSQL o MySQL.

SQL Server nos permite ejecutar aplicaciones de misión crítica, reduciendo costos de administración de infraestructura de datos y brinda introspectiva e información a todos los usuarios.

- **Confiable:** Permite a las organizaciones ejecutar sus aplicaciones más críticas con niveles de seguridad, confiabilidad y escalabilidad muy altos.
- **Productivo:** Permite reducir el tiempo y los costos requeridos para desarrollar y administrar sus infraestructuras de datos.

- **Inteligente:** Ofrece una plataforma integral que brinda introspectiva e información donde sus usuarios lo desean.

SQL Server 2008 nos permite crear rápidamente aplicaciones conectadas a la base de datos con la funcionalidad de funcionar en forma desconectada y después sincronizarlos con la base de datos central sin perder la línea de negocio y manteniendo los datos validados.

Elegimos trabajar con esta base de datos porque contiene mayor seguridad e integración más sencilla con el programa Visual Studio 2010, además encripta de manera transparente los datos, comprime los datos, tiene correctores de sintaxis del lenguaje Transact-SQL e IntelliSense (una característica del visual studio que permite a la base de datos sugerir objetos existentes mientras uno escribe la mitad de la palabra). Así mismo incluye nuevos tipos de datos y funciones. Entre ellos, datos espaciales, nuevos datos de tiempo (datetime2 y Datetimeoffset), tipos de datos jerárquicos.

### **Características de Microsoft SQL Server**

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.

- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

### **3.4.3. Ajax Control Toolkit**

Está desarrollada en base a ASP.NET AJAX y contiene un amplio conjunto de controles que se pueden utilizar para construir aplicaciones Web en AJAX altamente sencillas e interactivas para Web y extensores de Microsoft AJAX mejorando la funcionalidad de cliente de los controladores del servidor Web ASP.NET como los controles TextBox, Button y Panel. Mediante los extensores, puede ofrecer a los usuarios una mejor experiencia basada en Web.

El Ajax Control Toolkit contiene más de 40 controles, incluyendo los controles de AutoComplete, CollapsiblePanel, ColorPicker, MaskedEdit, Calendar, Accordion, y control Watermark.

### **3.4.4. Pokeln**

Es una librería AJAX que ofrece Comet para ASP.NET, maneja una amplia gama de las necesidades del desarrollo de la aplicación web como:

Serializa objetos del lado de servidor en lado de cliente.

Maneja casos por cada cliente.

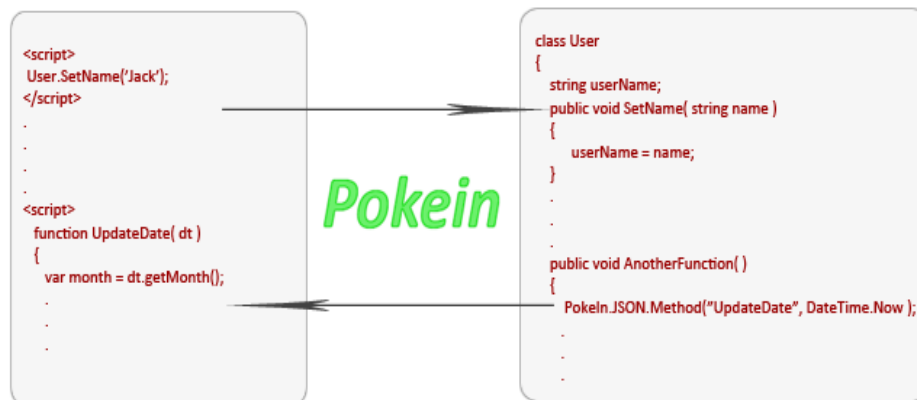
Manejan los monitores y recursos internos.

Define los métodos del objeto del servidor en lado de cliente.

Aplicaciones de visualización de los datos.

Juegos Onlines.

Aplicaciones del comercio electrónico.



**Figura 3.4.4** Diagrama Pokeln [7]

### 3.4.5. Entity FrameWork

Entity Framework es un conjunto de tecnologías de ADO.NET que permiten el desarrollo de aplicaciones de software orientadas a datos. Los arquitectos y programadores de aplicaciones orientadas a datos se han enfrentado a la necesidad de lograr dos objetivos muy diferentes. Deben modelar las entidades, las relaciones y la lógica de los problemas empresariales que resuelven, y también deben trabajar con los motores de datos que se usan para almacenar y recuperar los datos. Los datos pueden abarcar varios sistemas de almacenamiento, cada uno con sus propios protocolos; incluso las aplicaciones que funcionan con un único sistema de almacenamiento deben equilibrar los requisitos del sistema de almacenamiento con respecto a los requisitos de escribir un código de aplicación eficaz y fácil de mantener.

Entity Framework permite a los programadores trabajar con datos en forma de objetos y propiedades específicos del dominio, por ejemplo, con clientes y direcciones, sin tener que pensar en las tablas de las bases de datos subyacentes y en las columnas en las que se almacenan estos datos. Con Entity Framework, los desarrolladores de software pueden trabajar en un nivel más alto de abstracción cuando tratan con datos, y puede crear y mantener aplicaciones orientadas a datos con menos código. Dado que Entity Framework es un componente de .NET Framework, las aplicaciones de Entity Framework se pueden ejecutar en cualquier equipo en el que esté instalado .NET Framework a partir de la versión 3.5 SP1.

#### **3.4.6. JQuery**

JQuery es una librería o framework de JavaScript, permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.

jQuery es un software libre y de código abierto, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta librería se logran grandes resultados en menos tiempo y espacio.

## **CAPITULO IV**

### **4. IMPLEMENTACIÓN DEL SISTEMA**

En este capítulo se describirá el proceso de desarrollo de la aplicación y los desafíos encontrados, hasta llegar a la solución.

#### **4.1. El tablero y las piezas**

Los jugadores realizan los movimientos mediante operaciones de Drag&Drop. El estado debe mantenerse en memoria para realizar las validaciones respectivas.

#### **Desafío.-**

Una completa descripción de una posición de ajedrez, debería de contener los siguientes elementos:

- El lugar de cada pieza en el tablero
- El jugador que tiene el turno de mover

- El estado del enroque de ambos jugadores
- Si es posible que una pieza pueda capturar al paso, y de cual pieza se trata.

### Solución.-

Para conseguir la representación del tablero diseñamos las clases que se muestran en el siguiente diagrama:

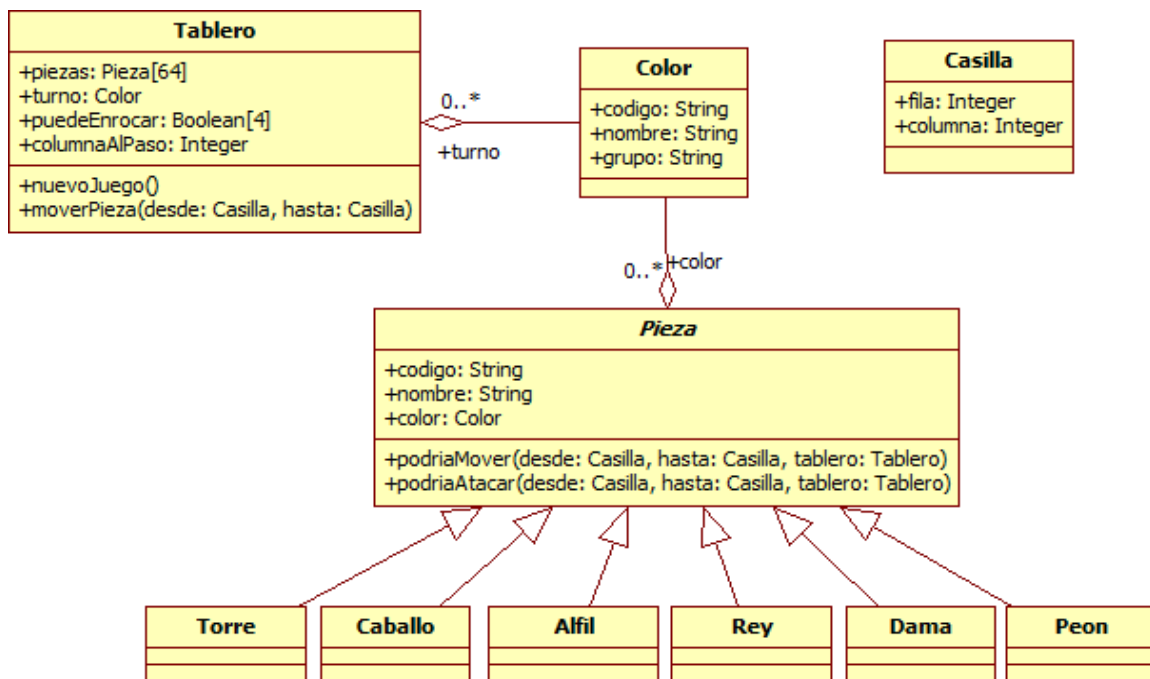


Figura 4.1 Diagrama de clases [8]

## 4.2. El envío de información desde el servidor hacia los clientes

En una aplicación de juego en línea, es necesario que el servidor comunique a los clientes sobre el acontecimiento de ciertos eventos.

#### **Desafío.-**

En el caso de nuestro juego de ajedrez por ejemplo, es necesario que cada vez que un jugador realice su movimiento, su oponente pueda visualizar dicho movimiento de manera casi instantánea, como si estuviera jugando en persona frente a su oponente.

#### **Solución.-**

Esta necesidad no la satisface AJAX, es necesario usar un concepto un poco más moderno: COMET.

Se hace uso de la librería POKEIN, que ofrece una implementación de COMET para ASP.Net fácil de usar.

### **4.3. Los Movimientos y el reloj**

Un movimiento o jugada se representa casi siempre por una casilla de origen y una de destino.



**Desafío.-**

Un movimiento debe reflejarse en cada usuario que juega o visualiza una partida.

**Solución.-**

Cuando un jugador realiza un movimiento sobre su representación del tablero, este movimiento se debe serializar y enviarse al servidor, el cual lo deserializa y lo procesa, para luego volverlo a serializar y así enviarlo al jugador oponente y a los otros jugadores que se encuentren observando la partida.

**Desafío.-**

Las partidas deben jugarse con reloj, el mismo que debe ser lo más realista y robusto posible.

**Solución.-**

El reloj que el sistema emula consiste en dos componentes que funcionan de forma diferente. Un componente se ejecuta en el lado del cliente en forma de dos contadores descendentes, los cuales el jugador

puede visualizar como su reloj y el de su oponente respectivamente. El otro componente se ejecuta del lado del servidor y permite que los contadores en los clientes se sincronicen con el servidor ante ciertos eventos como la realización de cada movimiento o la finalización del tiempo en alguno de los jugadores, etc. Durante la sincronización, el servidor envía el valor de los dos relojes, valores que son calculados en función de los milisegundos transcurridos desde la última vez que ocurrió un evento. Esta combinación de componentes es importante porque si sólo se controlara los relojes del lado del servidor, existiría la necesidad de crear múltiples hilos de ejecución, uno para cada reloj de cada partida; mientras que si sólo se controlara los relojes del lado del cliente, los jugadores malintencionados podrían encontrar la manera de entorpecer el conteo descendente del reloj, ganando tiempo para cada movimiento.

#### **4.4. Las distintas partidas y la concurrencia**

Es importante que cualquier servidor de juegos realice un buen manejo de la concurrencia.

En nuestro caso al igual que en muchos otros juegos y aplicaciones web, un mismo usuario no puede mantener más de una conexión abierta. La política

implementada finaliza la conexión anterior si la hubiere cada vez que un jugador se conecta al servidor.

Pero esto no nos libra de mantener la consistencia en cada partida.

### **Desafío.-**

Veamos el siguiente ejemplo:

Supongamos que el jugador A se enfrenta en este momento al jugador B; que es el turno de A; que en las pantallas de los dos jugadores los relojes marcan 0.1 segundos restantes para A y 0.1 segundos restantes para B (recordar que en cada pantalla se muestran ambos relojes). Como es el turno de A, su reloj está disminuyendo y si juega en este momento, el navegador enviará la jugada al servidor y en ese cliente comenzará a disminuir el reloj de B, por su parte, B está esperando que juegue A, y mientras tanto, debido a la latencia intrínseca en el envío de información, el reloj de A finaliza en la pantalla de B, pero al mismo tiempo, el reloj de B finaliza en la pantalla de A.

### **Solución.-**

Una vez que A juega, A debe esperar que el servidor le envíe una señal de que su jugada ha sido aceptada. Por su parte, B al enviar la

información de que el reloj de A ha finalizado, debe esperar que el servidor le confirme tal suceso. El servidor estará en capacidad de controlar esto porque sabe en todo momento de quién es el turno y mediante las técnicas de sincronización conocidas se pueden encolar los mensajes de los distintos clientes, de tal forma que se simplifica el problema si los mensajes se procesan en orden de llegada.



## **CONCLUSIONES Y RECOMENDACIONES**

En esta sección indicaremos las conclusiones obtenidas en el desarrollo del proyecto y al mismo tiempo daremos algunas recomendaciones para el uso efectivo del sistema.

### **Conclusiones**

1. Con el criterio de evaluación de algunos especialistas en el tema de ajedrez e informática se llegó a la conclusión de que el ajedrez y la tecnología informática son un binomio ideal, que se encuentran estrechamente ligados entre sí, para mejorar el proceso de aprendizaje constructivo del conocimiento por parte del usuario con la tecnología, transformando la información en conocimiento.
2. Contar con el desarrollo propio de un sistema en línea, permitirá a los estudiantes de ESPOL poner en práctica sus conocimientos sin necesidad de ir a un club o registrarse en alguna página de juego en línea, donde jugadores de ajedrez frecuentan.

3. Podemos decir que el jugar ajedrez en línea además permite mejorar, progresar y aumentar las habilidades intelectuales de cada participante con el objetivo de que cada jugador pueda ir evaluándose en el sistema, pues maneja la puntuación del juego al finalizar cada partida, es decir que se actualiza cada partida según se gane o pierda.
4. Con el análisis de pruebas realizadas en el desarrollo del sistema, podemos indicar que el jugador puede llegar a ganar una partida, sin tener que depender del movimiento de su contrincante.

### **Recomendaciones**

Las siguientes recomendaciones están basadas en el análisis de pruebas realizadas en el sistema y en experiencias adquiridas durante el desarrollo del mismo.

1. Para aquellos estudiantes de ESPOL que deseen desarrollar mejor sus técnicas de juego, se recomienda que utilicen el desarrollo del sistema de ajedrez en línea.
2. Sugerimos extender aún más el desarrollo del sistema de juego según los requerimientos necesarios, puesto que hemos visto que este tipo de sistemas es muy aplicable a esta clase de proyectos.
3. El número máximo de participantes que pueden estar conectados en el sistema es de 25 personas, de tal manera que si llegase a aumentar este

4. número de personas, se recomienda comprar la licencia original para que pueda soportar cantidades más fuertes.





# **ANEXO A.- Reglas del Juego de Ajedrez y Evolución del Ajedrez**

## **Artículo 1: Naturaleza y objetivos de la partida de ajedrez**

La partida de ajedrez se juega entre dos adversarios que mueven alternativamente sus propias piezas sobre un tablero cuadrado, llamado "tablero de ajedrez". El jugador con las piezas blancas comienza la partida. Se dice que un jugador "está en juego" cuando se ha realizado la jugada de su adversario.

El objetivo de cada jugador es situar al rey de su adversario "bajo ataque", de tal forma que el adversario no disponga de ninguna jugada (movimiento de pieza) legal que evite la "captura" del rey en la siguiente jugada. Del jugador que alcanza este objetivo se dice que ha dado "mate" al rey de su adversario y que ha ganado la partida. El adversario, cuyo rey ha recibido el mate, pierde la partida.


Si la posición es tal que ninguno de los jugadores puede dar mate, la partida es tablas.


## **Artículo 2: La posición inicial de las piezas sobre el tablero**

El tablero de ajedrez es un cuadrado dividido en 64 casillas cuadradas del mismo tamaño, con distribución 8 x 8, alternativamente claras (las casillas "blancas") y oscuras (las casillas "negras"). El tablero se coloca entre los jugadores de tal forma que la casilla de la esquina derecha más cercana a cada jugador sea blanca.

Al comienzo de la partida, un jugador dispone de 16 piezas de color claro (las piezas "blancas"); el otro tiene 16 piezas de color oscuro (las piezas "negras").

Estas piezas son las siguientes:

Un Rey..... 

Una Reina o Dama..... 

Dos Alfiles..... 

Dos Caballos..... 

Dos Torres o Roques..... 

Ocho Peones..... 

La posición inicial de las piezas en el tablero es la siguiente:

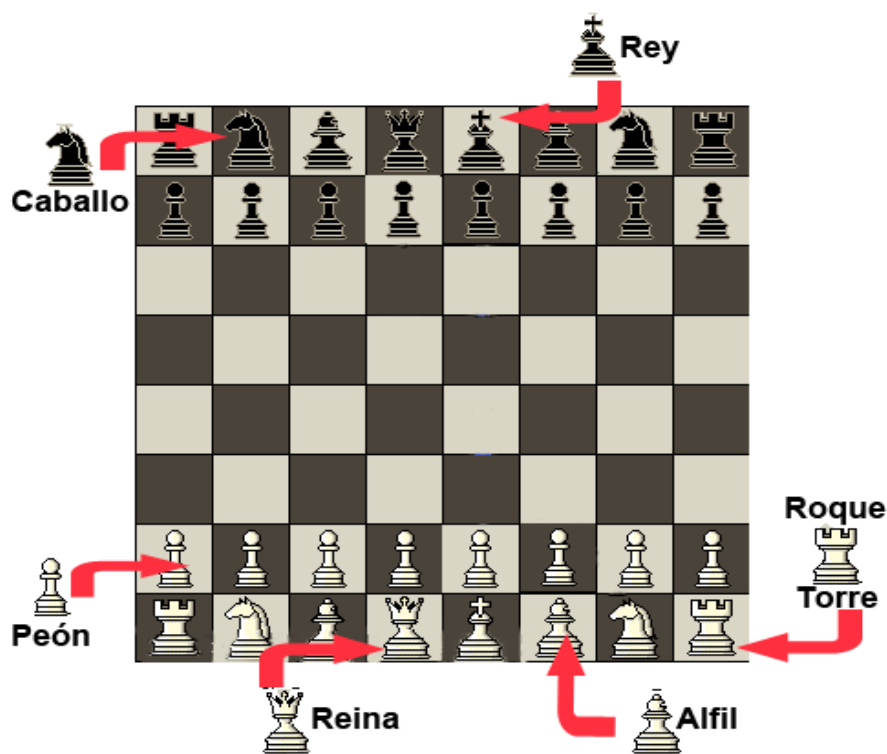


Figura 1. Posición inicial de las piezas

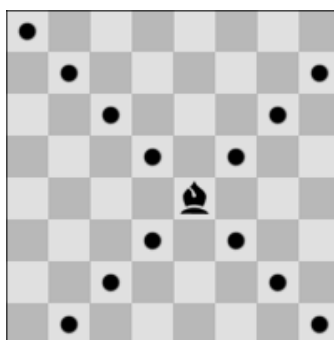
Las ocho hileras verticales se denominan "columnas". Las ocho hileras horizontales se denominan "filas". Una sucesión de casillas del mismo color en línea recta, tocándose por sus vértices, se denomina "diagonal".

### Artículo 3: El movimiento de las piezas

No está permitido mover una pieza a una casilla ocupada por una pieza del mismo color. Si una pieza se mueve a una casilla ocupada por una pieza de su

adversario, ésta es capturada y retirada del tablero como parte del mismo movimiento. Se dice que una pieza ataca a otra del adversario si puede efectuar una captura en esa casilla conforme al Artículo 3.

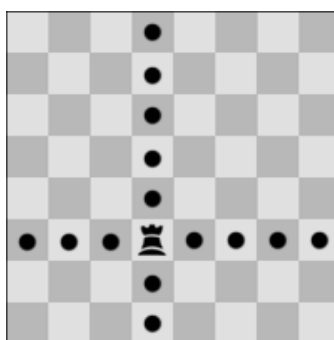
El alfil puede ser movido a cualquier casilla a lo largo de una de las diagonales sobre las que se encuentra.



**Figura a: El Alfil**

<http://www.ajedrezaranjuez.com/leyes.htm>

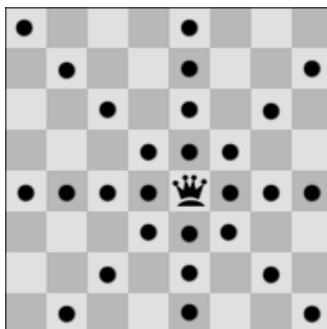
La torre puede ser movida a cualquier casilla a lo largo de la fila o columna en las que se encuentra.



**Figura b: La torre**

<http://www.ajedrezaranjuez.com/leyes.htm>

La dama puede ser movida a cualquier casilla a lo largo de la fila, columna o diagonal en las que se encuentra.

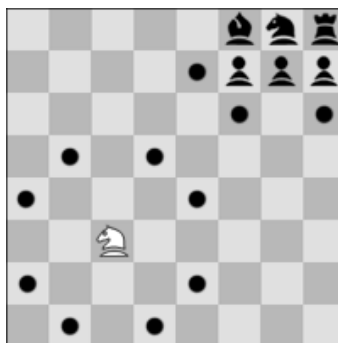


**Figura c: Dama o Reyna**

<http://www.ajedrezaranjuez.com/leyes.htm>

Al realizar estos movimientos, el alfil, la torre o la dama no pueden pasar sobre ninguna otra pieza.

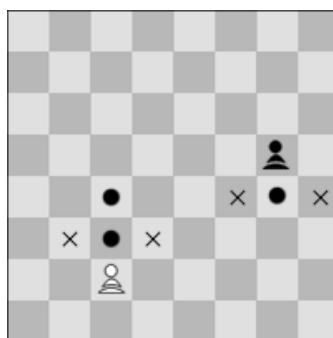
El caballo puede ser movido a una de las casillas más próximas a la que se encuentre, sin ser de la misma fila, columna o diagonal.



**Figura d: El Caballo**

<http://www.ajedrezaranjuez.com/leyes.htm>

- a. El peón puede ser movido hacia adelante a la casilla inmediatamente delante suyo en la misma columna, siempre que dicha casilla esté desocupada, o
- b. en su primer movimiento el peón puede ser movido como en (a); alternativamente, puede avanzar dos casillas a lo largo de la misma columna, siempre que ambas casillas estén desocupadas, o
- c. el peón puede ser movido a una casilla ocupada por una pieza del adversario que esté en diagonal delante suyo, sobre una columna adyacente, capturando dicha pieza.



**Figura e: El Peón**

<http://www.ajedrezaranjuez.com/leyes.htm>

- d. Un peón que ataca una casilla atravesada por un peón del adversario que ha avanzado dos casillas en un movimiento desde su casilla original, puede capturarlo como si sólo hubiera avanzado una casilla. Esta captura sólo puede efectuarse en el movimiento inmediatamente siguiente al citado avance y se denomina captura "al paso".



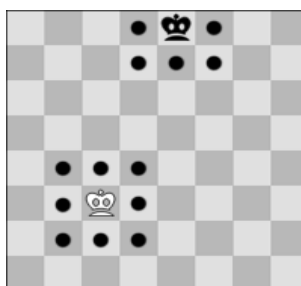
**Figura f: Captura "Al paso"**

<http://www.ajedrezaranjuez.com/leyes.htm>

- e. Cuando un peón alcanza la fila más alejada desde su posición inicial debe ser cambiado, como parte del mismo movimiento, por una dama, torre, alfil o caballo del mismo color. La elección del jugador no está limitada a piezas que hayan sido capturadas anteriormente. Este cambio de un peón por otra pieza se denomina "promoción", siendo inmediato el efecto de la nueva pieza.

**Hay dos formas diferentes de mover el rey:**

- i. Desplazándolo a cualquier casilla adyacente no atacada por una o más piezas del adversario.



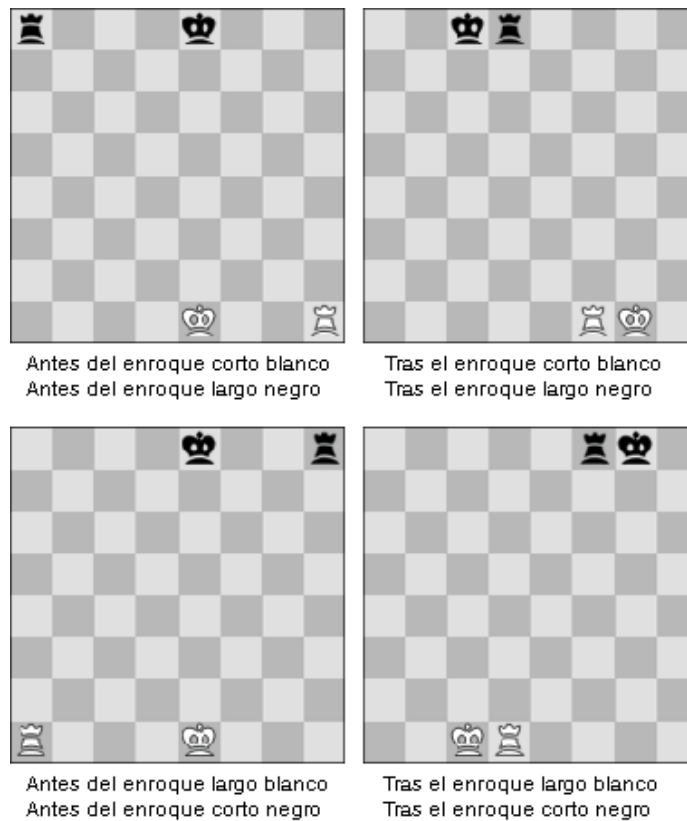
**Figura i: Desplazamiento del Rey**

<http://foro-mentesenblanco.web44.net/ajedrez.html>



Se considera que las piezas del adversario atacan una casilla incluso cuando dichas piezas no pueden ser movidas.

- ii. "Enrocando<sup>[4]</sup>". El enroque es un movimiento del rey y de una de las torres del mismo color y que esté en la misma fila, que cuenta como una simple jugada del rey y que se realiza como sigue: el rey es trasladado dos casillas desde su casilla original hacia la torre y luego dicha torre es trasladada a la casilla que acaba de cruzar el rey.



**Figura ii: Enroque del Rey**

<http://foro-mentesenblanco.web44.net/ajedrez.html>

(1) El enroque es ilegal:

- a. si el rey ya ha sido movido, o
- b. si se ha movido la torre con la que se quiere enrocar.

(2) El enroque está temporalmente impedido:

- c. si la casilla en la que se encuentra el rey, o la que debe cruzar, o la que finalmente va a ocupar, está atacada por una o más piezas del adversario, o
- d. si hay alguna pieza entre el rey y la torre con la que se va a efectuar el enroque.

Se dice que el rey está "en jaque", si está atacado por una ó más piezas del adversario, incluso aunque dichas piezas no pudieran ser movidas porque dejarían o situarían a su propio rey en Jaque. Ninguna pieza puede ser movida de manera tal que ponga o deje a su propio rey en jaque.

#### **Artículo 4: La acción de mover las piezas**

Cada jugada debe efectuarse con una sola mano.

El jugador que está en juego puede ajustar una o más piezas en sus casillas, siempre que previamente exprese su intención de hacerlo (por ejemplo, diciendo "compongo").

**(a)** Exceptuando lo previsto en el Artículo 4, si el jugador que está en juego toca deliberadamente sobre el tablero:

- a. una o más piezas propias, debe mover la primera pieza tocada que se pueda mover, o
- b. una o más piezas del adversario, debe capturar la primera pieza tocada que pueda ser capturada, o
- c. una pieza de cada color, debe capturar la pieza del adversario con la suya o, si ello es ilegal, mover o capturar la primera pieza tocada que se pueda mover o capturar. Si resulta imposible establecer qué pieza se tocó en primer lugar, será la pieza propia la que se considere como pieza tocada.

**(b)** Puede tomarse las siguientes consideraciones:

- a. Si un jugador toca deliberadamente su rey y torre, debe enrocar por ese lado si fuera legal hacerlo.
- b. Si un jugador toca deliberadamente una torre y luego su rey, no podrá enrocar por ese lado en esa jugada y se procederá según lo establecido en el Artículo 4(a).

- c. Si un jugador, con la intención de enrocar, toca el rey o rey y torre a la vez, siendo ilegal el enroque por ese lado, el jugador debe hacer otra jugada legal con su rey, lo que puede incluir el enroque por el otro lado. Si el rey no tiene ningún movimiento legal, el jugador es libre de realizar cualquier jugada legal.
- d. Si un jugador promociona un peón, la elección de la pieza se considerará definitiva cuando ésta toque la casilla de promoción.

Si ninguna de las piezas tocadas puede ser movida o capturada, el jugador puede realizar cualquier jugada legal.

Cuando se ha dejado una pieza en una casilla, como jugada legal o parte de una jugada legal, ya no puede ser movida a otra casilla. Se considera que el movimiento ha sido realizado cuando se han cumplido todos los requisitos pertinentes del Artículo 3.

- a. En el caso de captura, cuando la pieza capturada se ha retirado del tablero y el jugador ha soltado su propia pieza después de situarla en su propia casilla;
- b. en el caso del enroque, cuando el jugador ha soltado la torre en la casilla previamente cruzada por el rey. Cuando el jugador ha soltado el rey la jugada no se considera todavía completada, pero el jugador ya no tiene otra opción que realizar el enroque por ese lado, siempre que sea legal;

- c. en el caso de la promoción de un peón, cuando éste ha sido retirado del tablero y la mano del jugador ha soltado la nueva pieza después de situarla en la casilla de promoción. Cuando el jugador ha soltado el peón que ha alcanzado la casilla de promoción, la jugada no se considera todavía completada, pero el jugador no tiene ya el derecho de jugar el peón a otra casilla.

Un jugador pierde su derecho a reclamar la violación de los Artículos 4 (a) y (b) por parte de su adversario, una vez que él mismo haya tocado deliberadamente una pieza.

#### **Artículo 5: La finalización de la partida**

##### 5.1

- a. La partida es ganada por el jugador que ha dado mate al rey de su adversario. Esto finaliza inmediatamente la partida, siempre que la jugada que generó la posición de mate fuera una jugada legal.
- b. La partida es ganada por el jugador cuyo adversario declara que abandona. Esto finaliza inmediatamente la partida.

##### 5.2

- a. La partida es tablas cuando el jugador que está en juego no puede hacer ninguna jugada legal y su rey no está en jaque. Se dice entonces que el

- b. rey está "ahogado". Esto finaliza inmediatamente la partida, siempre que la jugada que generó la posición de rey ahogado fuera legal.
- c. La partida es tablas cuando se alcanza una posición en la que ningún jugador puede dar mate al rey del adversario con cualquier serie de jugadas legales. Se dice entonces que la partida termina en una "posición muerta". Esto finaliza inmediatamente la partida, siempre que la jugada que generó tal posición fuera legal.
- d. La partida es tablas por acuerdo entre los dos jugadores durante el desarrollo de la misma. Esto finaliza inmediatamente la partida.
- e. La partida puede ser tablas si se va a dar o ya se ha dado cualquier posición idéntica, al menos tres veces sobre el tablero.
- f. La partida puede ser tablas si cada jugador ha hecho los últimos 50 movimientos consecutivos sin que haya habido ningún movimiento de peón ni captura de pieza.

### **Top 100 Raking de Jugadores de Enero 2011**

Rank	Name	Title	Country	Rating	Games	B-Year
1	<a href="#">Carlsen, Magnus</a>	g	NOR	2814	17	1990
2	<a href="#">Anand, Viswanathan</a>	g	IND	2810	17	1969
3	<a href="#">Aronian, Levon</a>	g	ARM	2805	9	1982
4	<a href="#">Kramnik, Vladimir</a>	g	RUS	2784	16	1975
5	<a href="#">Karjakin, Sergey</a>	g	RUS	2776	20	1990

6	<a href="#">Topalov, Veselin</a>	g	BUL	2775	10	1975
7	<a href="#">Grischuk, Alexander</a>	g	RUS	2773	20	1983
8	<a href="#">Mamedyarov, Shakhriyar</a>	g	AZE	2772	9	1985
9	<a href="#">Ivanchuk, Vassily</a>	g	UKR	2764	0	1969
10	<a href="#">Nakamura, Hikaru</a>	g	USA	2751	16	1987
11	<a href="#">Ponomariov, Ruslan</a>	g	UKR	2744	0	1983
12	<a href="#">Radjabov, Teimour</a>	g	AZE	2744	0	1987
13	<a href="#">Gashimov, Vugar</a>	g	AZE	2736	10	1986
14	<a href="#">Wang, Yue</a>	g	CHN	2734	21	1987
15	<a href="#">Nepomniachtchi, Ian</a>	g	RUS	2733	11	1990
16	<a href="#">Gelfand, Boris</a>	g	ISR	2733	9	1968
17	<a href="#">Wang, Hao</a>	g	CHN	2731	33	1989
18	<a href="#">Svidler, Peter</a>	g	RUS	2730	11	1976
19	<a href="#">Kamsky, Gata</a>	g	USA	2730	6	1974
20	<a href="#">Wojtaszek, Radoslaw</a>	g	POL	2726	0	1987
21	<a href="#">Eljanov, Pavel</a>	g	UKR	2724	9	1983
22	<a href="#">Bacrot, Etienne</a>	g	FRA	2723	10	1983
23	<a href="#">Adams, Michael</a>	g	ENG	2723	7	1971
24	<a href="#">Shirov, Alexei</a>	g	ESP	2722	15	1972
25	<a href="#">Caruana, Fabiano</a>	g	ITA	2721	18	1992
26	<a href="#">Movsesian, Sergei</a>	g	ARM	2721	0	1978
27	<a href="#">Almasi, Zoltan</a>	g	HUN	2719	10	1976
28	<a href="#">Jakovenko, Dmitry</a>	g	RUS	2718	11	1983
29	<a href="#">Leko, Peter</a>	g	HUN	2717	0	1979
30	<a href="#">Dominguez Perez, Leinier</a>	g	CUB	2716	0	1983
31	<a href="#">Vachier-Lagrave, Maxime</a>	g	FRA	2715	6	1990
32	<a href="#">Malakhov, Vladimir</a>	g	RUS	2714	14	1980
33	<a href="#">Vitiugov, Nikita</a>	g	RUS	2709	11	1987
34	<a href="#">Navara, David</a>	g	CZE	2708	0	1985
35	<a href="#">Fressinet, Laurent</a>	g	FRA	2707	15	1981
36	<a href="#">Jobava, Baadur</a>	g	GEO	2707	0	1983

37	<a href="#">Alekseev, Evgeny</a>	g	RUS	2701	0	1985
38	<a href="#">Efimenko, Zahar</a>	g	UKR	2701	0	1985
39	<a href="#">Morozevich, Alexander</a>	g	RUS	2700	0	1977
40	<a href="#">Vallejo Pons, Francisco</a>	g	ESP	2698	0	1982
41	<a href="#">Tomashevsky, Evgeny</a>	g	RUS	2695	11	1987
42	<a href="#">Sutovsky, Emil</a>	g	ISR	2695	0	1977
43	<a href="#">Dreev, Aleksey</a>	g	RUS	2694	18	1969
44	<a href="#">Bologan, Viktor</a>	g	MDA	2693	7	1971
45	<a href="#">Sasikiran, Krishnan</a>	g	IND	2690	9	1981
46	<a href="#">Onischuk, Alexander</a>	g	USA	2689	13	1975
47	<a href="#">Andreikin, Dmitry</a>	g	RUS	2689	9	1990
48	<a href="#">Riazantsev, Alexander</a>	g	RUS	2689	0	1985
49	<a href="#">Laznicka, Viktor</a>	g	CZE	2688	0	1988
50	<a href="#">Motylev, Alexander</a>	g	RUS	2687	3	1979
51	<a href="#">Bruzon Batista, Lazaro</a>	g	CUB	2686	18	1982
52	<a href="#">Giri, Anish</a>	g	NED	2686	7	1994
53	<a href="#">Polgar, Judit</a>	g	HUN	2686	0	1976
54	<a href="#">Naiditsch, Arkadij</a>	g	GER	2685	0	1985
55	<a href="#">Kasimdzhanov, Rustam</a>	g	UZB	2681	7	1979
56	<a href="#">Timofeev, Artyom</a>	g	RUS	2681	0	1985
57	<a href="#">Berkes, Ferenc</a>	g	HUN	2678	9	1985
58	<a href="#">Volokitin, Andrei</a>	g	UKR	2678	2	1986
59	<a href="#">Nisipeanu, Liviu-Dieter</a>	g	ROU	2678	0	1976
60	<a href="#">Rublevsky, Sergei</a>	g	RUS	2678	0	1974
61	<a href="#">Bu, Xiangzhi</a>	g	CHN	2677	18	1985
62	<a href="#">Van Wely, Loek</a>	g	NED	2676	12	1972
63	<a href="#">Akopian, Vladimir</a>	g	ARM	2675	0	1971
64	<a href="#">So, Wesley</a>	g	PHI	2673	18	1993
65	<a href="#">Inarkiev, Ernesto</a>	g	RUS	2672	9	1985
66	<a href="#">Areshchenko, Alexander</a>	g	UKR	2671	9	1986
67	<a href="#">Zhigalko, Sergei</a>	g	BLR	2671	9	1989



68	<a href="#">Korobov, Anton</a>	g	UKR	2670	18	1985
69	<a href="#">Miroshnichenko, Evgenij</a>	g	UKR	2670	0	1978
70	<a href="#">Moiseenko, Alexander</a>	g	UKR	2670	0	1980
71	<a href="#">Nielsen, Peter Heine</a>	g	DEN	2670	0	1973
72	<a href="#">Georgiev, Kiril</a>	g	BUL	2669	0	1965
73	<a href="#">Meier, Georg</a>	g	GER	2667	10	1987
74	<a href="#">Harikrishna, P.</a>	g	IND	2667	9	1986
75	<a href="#">Sargissian, Gabriel</a>	g	ARM	2667	0	1983
76	<a href="#">Kobalia, Mikhail</a>	g	RUS	2666	0	1978
77	<a href="#">Cheparinov, Ivan</a>	g	BUL	2665	9	1986
78	<a href="#">McShane, Luke J</a>	g	ENG	2664	7	1984
79	<a href="#">Le, Quang Liem</a>	g	VIE	2664	5	1991
80	<a href="#">Bareev, Evgeny</a>	g	RUS	2663	0	1966
81	<a href="#">Kurnosov, Igor</a>	g	RUS	2662	18	1985
82	<a href="#">Smeets, Jan</a>	g	NED	2662	1	1985
83	<a href="#">Zvjaginsev, Vadim</a>	g	RUS	2660	20	1976
84	<a href="#">Grachev, Boris</a>	g	RUS	2660	12	1986
85	<a href="#">Smirin, Ilia</a>	g	ISR	2660	9	1968
86	<a href="#">Mamedov, Rauf</a>	g	AZE	2660	0	1988
87	<a href="#">Socko, Bartosz</a>	g	POL	2660	0	1978
88	<a href="#">Gharamian, Tigran</a>	g	FRA	2658	13	1984
89	<a href="#">Short, Nigel D</a>	g	ENG	2658	13	1965
90	<a href="#">Sokolov, Ivan</a>	g	NED	2657	9	1968
91	<a href="#">Feller, Sebastien</a>	g	FRA	2657	0	1991
92	<a href="#">Nyback, Tomi</a>	g	FIN	2656	2	1985
93	<a href="#">Zhou, Jianchao</a>	g	CHN	2655	26	1988
94	<a href="#">Fridman, Daniel</a>	g	GER	2655	9	1976
95	<a href="#">Milov, Vadim</a>	g	SUI	2653	9	1972
96	<a href="#">Gustafsson, Jan</a>	g	GER	2652	0	1979
97	<a href="#">Potkin, Vladimir</a>	g	RUS	2651	11	1982
98	<a href="#">Ganguly, Surya Shekhar</a>	g	IND	2651	8	1983

99	<a href="#">Fedorchuk, Sergey A.</a>	g	UKR	2650	22	1981
100	<a href="#">Khismatullin, Denis</a>	g	RUS	2649	20	1984
101	<a href="#">Li, Chao b</a>	g	CHN	2649	18	1989
102	<a href="#">Roiz, Michael</a>	g	ISR	2649	0	1983

**Figura iii:** Top 100 Raking de Jugadores de Enero 2011



## **ANEXO B.- Tecnología Utilizada y Características**

### **.NET Framework**

.NET Framework<sup>[5]</sup> es un componente integral de Windows que admite la compilación y la ejecución de aplicaciones y servicios Web XML. Los componentes clave de .NET Framework son Common Language Runtime (CLR) y la biblioteca de clases, que incluye ADO.NET, ASP.NET, formularios Windows Forms y Windows Presentation Foundation (WPF).

El diseño de .NET Framework cumple con los siguientes objetivos:

- Proporcionar una programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local, pero distribuida en Internet o ejecutar de forma remota.
- Facilitar un código que reduzca lo máximo posible la implementación de software y los conflictos de versiones.
- Ofrecer un código que inicie la ejecución segura del mismo.
- Proporcionar un código que elimine los problemas de rendimiento de los entornos en los que se utilizan scripts o intérpretes de comandos.

- Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en el Web.
- Establecer toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se puede integrar con otros tipos de código.

Common Language Runtime (CLR) es el fundamento de .NET Framework. El motor en tiempo de ejecución se puede considerar como un agente que administra el código en tiempo de ejecución y proporciona servicios centrales, como la administración de memoria, subprocesos y la comunicación remota, al tiempo que aplica una seguridad estricta a los tipos y otras formas de especificación del código que promueven su seguridad y solidez. La biblioteca de clases, el otro componente principal de .NET Framework, es una completa colección orientada a objetos de tipos reutilizables que se pueden emplear para desarrollar aplicaciones que abarcan desde las tradicionales herramientas de interfaz gráfica de usuario (GUI) o de línea de comandos hasta las aplicaciones basadas en las innovaciones más recientes proporcionadas por ASP.NET, como los formularios Web Forms y los servicios Web XML.

.NET Framework puede hospedarse en componentes no administrados que cargan Common Language Runtime (CLR) en sus procesos e inician la ejecución de código administrado, con lo que se crea un entorno de software en el que se

pueden utilizar características administradas y no administradas. En .NET Framework no sólo se ofrecen varios hosts de motor en tiempo de ejecución, sino que también se admite el desarrollo de estos hosts por parte de terceros.

Por ejemplo, ASP.NET hospeda el motor en tiempo de ejecución para proporcionar un entorno de servidor escalable para el código administrado. ASP.NET trabaja directamente con el motor en tiempo de ejecución para habilitar aplicaciones de ASP.NET y servicios Web XML.

Internet Explorer es un ejemplo de aplicación no administrada que hospeda el motor en tiempo de ejecución.

### **Active Server Pages**

Active Server Pages (ASP) es una tecnología de Microsoft del "lado del servidor" para páginas web generadas dinámicamente, que ha sido comercializada en muchas ocasiones como un anexo a Internet Information Server (IIS).

La tecnología ASP está estrechamente relacionada con el modelo tecnológico de su fabricante. Intenta ser solución para un modelo de programación rápida ya que programar en ASP es como programar en Visual Basic y C#, por supuesto con ciertas limitaciones y algunas ventajas en entornos web. El proyecto se lo realizó en ASP.Net que es la versión más reciente de esta herramienta, acompañada por .NET Framework 4.0.

Lo interesante de este modelo tecnológico es poder utilizar diversos componentes ya desarrollados como algunos controles ActiveX así como componentes del lado del servidor, tales como CDONTS, por ejemplo, que permite la interacción de los scripts con el servidor SMTP que integra IIS.

Se facilita la programación de sitios web mediante varios objetos integrados, como por ejemplo un objeto de sesión basada en cookies, que mantiene las variables mientras se pasa de página a página.

### **Características generales de ASP.NET**

Entre sus más destacables características, se encuentran la capacidad para utilizar múltiples monitores, así como la posibilidad de desacoplar las ventanas de su sitio original y acoplarlas en otros sitios de la interfaz de trabajo. Además de esto, aparece una edición que compila las características de todas las ediciones comunes de Visual Studio: Professional, Team Studio, Test, conocida como Visual Studio Ultimate.

**Las páginas** de ASP.NET, conocidas como "web forms", son el principal medio de construcción para el desarrollo de aplicaciones web. Los formularios web están contenidos en archivos con una extensión ASPX; en jerga de programación, estos archivos contienen etiquetas HTML o XHTML estático, y también etiquetas definiendo Controles Web que se procesan del lado del servidor y Controles de Usuario donde los desarrolladores colocan todo el código estático y dinámico requerido por la página web. Adicionalmente, el código

dinámico que se ejecuta en el servidor puede ser colocado en una página dentro de un bloque que es muy similar a otras tecnologías de desarrollo como PHP, JSP y ASP, pero no es aconsejada excepto para propósitos de enlace de datos pues requiere más llamadas cuando se genera la página. ASP.NET Es limitado a solo funcionar con IIS.

ASP.NET permite la creación de componentes reutilizables a través de la creación de **Controles de Usuario** (User Controls).

Las aplicaciones ASP.NET son alojadas en un servidor web y se tiene acceso a ellas mediante el protocolo sin estado HTTP, que no guarda ninguna información sobre conexiones anteriores. Por lo tanto, si la aplicación requiere interacción entre conexiones, tiene que implementar su propia **administración del estado**.

El **estado de la aplicación** (Application state) es una colección de variables definidas por el usuario que son compartidas por todas las invocaciones de una aplicación ASP.NET.

El **estado de la sesión** (Session state) es una colección de variables definidas por el usuario, las cuales persisten durante la sesión de un usuario.

El **estado de la vista** (View state) permite el mecanismo de administración de estado a nivel de página, que es utilizado por las páginas HTML generadas por las aplicaciones ASP.NET para mantener el estado de los controles de los formularios web y los widgets.



## **Tecnología ADO.NET Entity Framework**

ADO.NET Entity Framework permite crear aplicaciones de acceso a datos programando con un modelo de la aplicación conceptual en lugar de un esquema de almacenamiento relacional. El objetivo es reducir la cantidad de código y mantenimiento que se necesita para las aplicaciones orientadas a datos. Las aplicaciones de Entity Framework ofrecen las siguientes ventajas:

- Las aplicaciones pueden funcionar en términos de un modelo conceptual más centrado en la aplicación, que incluye tipos con herencia, miembros complejos y relaciones.
- Las aplicaciones están libres de dependencias de codificación rígida de un motor de datos o de un esquema de almacenamiento.
- Las asignaciones entre el modelo conceptual y el esquema específico de almacenamiento pueden cambiar sin tener que cambiar el código de la aplicación.
- Los programadores pueden trabajar con un modelo de objeto de aplicación coherente que se puede asignar a diversos esquemas de almacenamiento, posiblemente implementados en sistemas de administración de base de datos diferentes.

- Se pueden asignar varios modelos conceptuales a un único esquema de almacenamiento.
- La compatibilidad con Language Integrated Query (LINQ) proporciona validación de la sintaxis en el momento de la compilación para consultas en un modelo conceptual.

### **ADO.NET Entity Data Model Designer**

ADO.NET Entity Data Model Designer (Entity Designer) es una herramienta que permite la modificación de un archivo .edmx con el mecanismo de apuntar y hacer clic. Se puede utilizar Entity Designer para crear y modificar entidades, asociaciones, asignaciones y relaciones de herencia, además de validar un archivo .edmx.

Entity Designer funciona junto con el Asistente para Entity Data Model, el Asistente para actualizar modelo y el Asistente para crear base de datos para poder generar, editar y actualizar un archivo .edmx.

Entity Designer consta de los siguientes componentes:

- Una superficie de diseño visual para modificar el modelo conceptual. Puede crear, modificar o eliminar entidades y asociaciones.

- Una ventana Detalles de Mapping para ver y editar asignaciones. Puede asignar tipos de entidad o asociaciones a tablas, columnas y procedimientos almacenados de base de datos.
- Una ventana Explorador de modelos que proporciona vistas de árbol del modelo conceptual y del modelo de almacenamiento.
- Controles del cuadro de herramientas para crear entidades, asociaciones y relaciones de herencia.

Entity Designer trabaja con un archivo .edmx el cual es la combinación de tres archivos de metadatos: el lenguaje de definición de esquemas conceptuales (CSDL), el lenguaje de definición de esquemas de almacenamiento (SSDL) y el lenguaje de especificación de asignaciones (MSL). Al ejecutar el Asistente para Entity Data Model, se crea y agrega un archivo .edmx a la solución. Este archivo se abre automáticamente en Entity Designer cuando finaliza el Asistente para Entity Data Model.

### **Internet Information Server**

Internet Information Services (IIS) es un servidor web y un conjunto de servicios para el sistema operativo. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a una PC en un servidor web para Internet o una intranet, es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente.

Los servicios de IIS proporcionan las herramientas y funciones necesarias para administrar de forma sencilla un servidor web seguro.

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas. Por ejemplo, Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

IIS, ha evolucionado en varias versiones respecto a los sistemas

operativos propios de cada época, entre los cuales tenemos:

IIS 1.0, Windows NT 3.51 Service Pack 3

IIS 2.0, Windows NT 4.0

IIS 3.0, Windows NT 4.0 Service Pack 3

IIS 4.0, Windows NT 4.0 Option Pack

IIS 5.0, Windows 2000

IIS 5.1, Windows XP Professional

IIS 6.0, Windows Server 2003 y Windows XP Profesional x64 Edition

IIS 7.0, Windows Vista (Solo Business y Ultimate) y Windows Server 2008

IIS 7.5, Windows 7 y Windows Server 2008 R2



## REFERENCIA BIBLIOGRÁFICA

- [1] Wikipedia ,Plies por búsqueda,[http://en.wikipedia.org/wiki/Ply\\_%28game\\_theory%29](http://en.wikipedia.org/wiki/Ply_%28game_theory%29),  
Marzo 2010
- [2] Wikipedia, alfa-beta, [http://es.wikipedia.org/wiki/Poda\\_alfa-beta](http://es.wikipedia.org/wiki/Poda_alfa-beta), Diciembre 2010
- [3] Wikipedia, Negascout. <http://es.wikipedia.org/wiki/Negascout>, Octubre 2010
- [4] Club Aranjuez de Ajedrez (CAA), Enrocando  
<http://www.ajedrezaranjuez.com/leyes.htm>, Julio 2005
- [5] .NET Framework <http://msdn.microsoft.com/eses/library/zw4w595w.aspx>
- [6] Heidi Salazar, Arquitectura de Sistema “*Modelo 3 - Tier*”,  
[http://www.mappinginteractivo.com/plantilla-ante.asp?id\\_articulo=1555](http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1555), Enero  
2009
- [7] Pokeln Documentation, Pokeln <http://pokein.com/Help/Documentation.aspx>, 2010
- [8] Luis Toapanta, Diagrama de clases, Luis Toapanta, Diciembre 2010
- [9] Menten en Blanco, Posición inicial de las piezas, <http://foro-mentesenblanco.web44.net/ajedrez.html>, Enero 2011