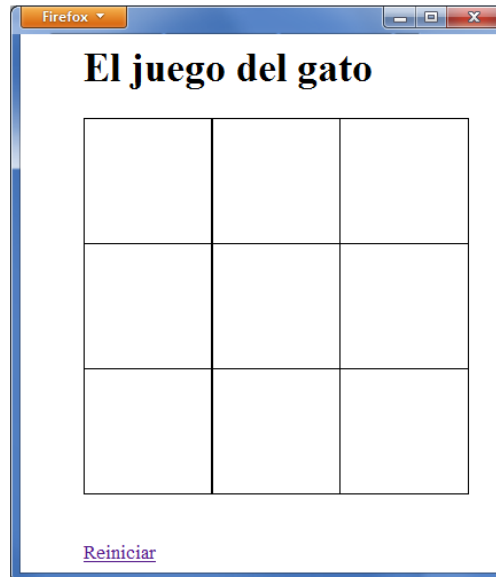


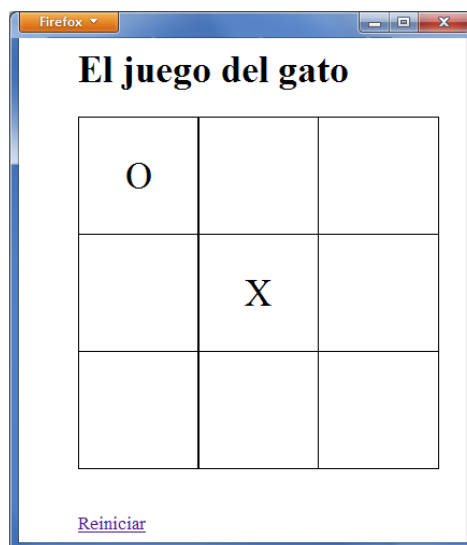
## Desarrollo de Aplicaciones Distribuidas

### Practica Ajax-XML-RMI

El objetivo de la práctica es construir una aplicación para jugar al popular juego del gato contra una computadora, utilizando Ajax, XML y RMI. La pantalla inicial muestra el tablero vacío y una liga que permite en todo momento reiniciar un juego.

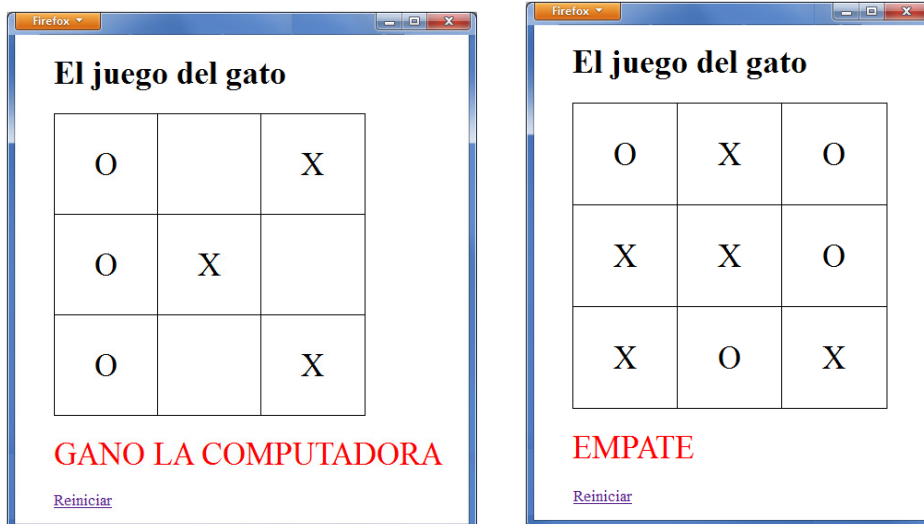


Para jugar, el usuario simplemente hace “click” en la casilla de su preferencia. El jugador siempre usará la “X” como símbolo y la computadora utilizará “O” en todos los casos el jugador humano es el que juega primero (tiene siempre el primer turno).

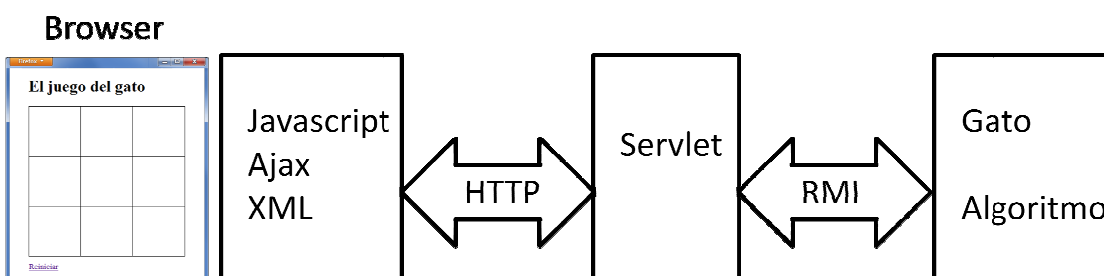


Tan pronto el jugador hace “click” en la casilla del centro, aparece su jugada (X) y la respuesta de la computadora (O).

Al terminar el juego aparece un veredicto bajo el tablero, ejemplos:



## ARQUITECTURA



## Observaciones

1. El estado del juego lo conserva el cliente. El servlet debe recibir todo el tablero cada vez que el jugador realiza un nuevo movimiento. Las casillas ya marcadas por el jugador ó por la computadora ya no pueden cambiarse. Los programas hacen referencia a las casillas del tablero numerándolas de la siguiente manera:

0	1	2
3	4	5
6	7	8

En java un tablero se representa mediante un arreglo de 9 enteros cada entero representa el estado de la casilla: 0 si esta en blanco, 1 si la marcó el jugador con una "X" y 2 si la marcó la computadora con una "O".

Ejemplo: {2,0,1,2,1,0,2,0,1} equivale a:

O		X
O	X	
O		X

2. Cada vez que el cliente hace “click” en una casilla vacía se inicia el mecanismo de **Ajax** para enviarle el estado del tablero al servlet y esperar que el servlet nos responda con un archivo XML que contiene:

- El tablero ya con la respuesta de la computadora, cada casilla tiene asociado un valor que puede ser 0, 1 ó 2 (ver el punto anterior):
- Un veredicto sobre el estado del juego: *Inconcluso, Empate, Jugador, Computadora.*

3. El servlet regresa siempre un documento XML con el siguiente formato:

```
<?xml version="1.0"?>
<juego>
  <tablero>
    <casilla>
      <posicion>0</posicion>
      <valor>2</valor>
    </casilla>
    <casilla>
      <posicion>1</posicion>
      <valor>1</valor>
    </casilla>
    <casilla>
      <posicion>2</posicion>
      <valor>2</valor>
    </casilla>
    <casilla>
      <posicion>3</posicion>
      <valor>1</valor>
    </casilla>
    <casilla>
      <posicion>4</posicion>
      <valor>1</valor>
    </casilla>
    <casilla>
      <posicion>5</posicion>
      <valor>2</valor>
    </casilla>
    <casilla>
      <posicion>6</posicion>
      <valor>1</valor>
    </casilla>
    <casilla>
      <posicion>7</posicion>
      <valor>2</valor>
    </casilla>
    <casilla>
      <posicion>8</posicion>
      <valor>1</valor>
    </casilla>
  </tablero>
  <resultado>Empate</resultado>
</juego>
```

4. El algoritmo del juego del gato se anexa en el archivo “Algoritmo.java”.

5.- Hay que convertir el programa “Algoritmo.java” en un servidor RMI con la siguiente interfaz:

```
package gato;
import java.rmi.*;

public interface Gato extends Remote {
    public int movimiento(int[] tablero)throws
    java.rmi.RemoteException;

    public String resultado( int[] tablero )throws
    java.rmi.RemoteException;
}
```

El método **movimiento()** recibe el tablero y responde con el número de casilla donde la computadora escogió jugar.

El método **resultado()** recibe el tablero y responde con un veredicto sobre el juego ya sea:

**Inconcluso** si el juego todavía no termina es decir nadie ha ganado y todavía quedan casillas vacías.

**Empate** si nadie ha ganado y ya no quedan casillas vacías.

**Jugador** si según el tablero fue el jugador el que ganó,

**Computadora** si según el tablero fue la computadora la que ganó,

6. El servlet se comunicará con el Algoritmo mediante RMI. Es decir el servlet es un cliente RMI.

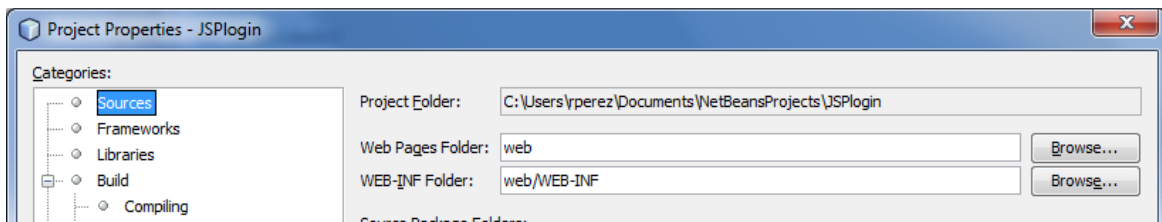
## IMPORTANTE

Es obligatorio utilizar la interfaz RMI **Gato.java** tal como se proporciona sin ninguna modificación.

Es obligatorio que el servlet responda utilizando el formato **XML** descrito en el punto 3 de las observaciones sobre la arquitectura.

## Entregable

Incluya en el zip de entrega la carpeta del proyecto generado por Netbeans. El directorio donde se encuentra el proyecto puede obtenerse ejecutando Netbeans y colocando el mouse sobre el nombre del proyecto, apoyando sobre el botón derecho y seleccionar “properties”, con esto aparecerá una ventana y en uno de los campos de la ventana podrán leer “Project Folder”:



En este caso el proyecto se llama JSPlogin y se encuentra en  
C:\Users\yperez\Documents\NetBeansProjects\JSPlogin

Incluya un script **sql** con las modificaciones que haga a la base de datos original (si utiliza Access incluya directamente el archivo).

## Importante

Antes de copiar la carpeta asegúrese de ejecutar “Clean and Build” en la herramienta Netbeans, colocando el mouse sobre el nombre del proyecto, apoyando sobre el botón derecho y seleccionar “Clean and Build”.

Preparar el reporte de acuerdo a los requisitos indicados en la página web del curso (<http://cs.mty.itesm.mx/profesores/raul.perez/DAD/practicas.pdf>). Empacar todos los archivos en un .zip nombrado según su matrícula A12345678 y subirlo a blackboard.