

“Programmers are tools for converting caffeine into code”

- Anónimo

Debido al auge de los videojuegos en red entre la población adolescente, una compañía que desarrolla software lo ha contratado para que les haga una aplicación cliente/servidor que les permita a varias personas jugar entre ellos un juego de estrategia¹ denominado *“Tado Craft”*.

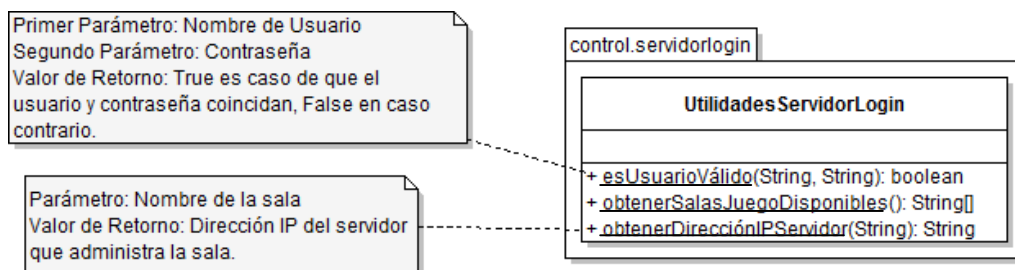
Antes de iniciar una nueva partida en *Tado Craft*, los jugadores (es decir, los clientes) deben conectarse a un servidor, al que llamaremos *ServidorLogin*, el cual tiene la labor de realizar la autenticación de los jugadores. Este servidor tiene el siguiente nombre: login.tadocraft.com, y se encuentra atendiendo solicitudes por el puerto 49585. Para autenticarse, el cliente tiene que enviar en formato texto su nombre de usuario y contraseña, separados por punto y coma.

NombreUsuario;Contraseña

En caso de que el usuario y la contraseña sean válidos, *ServidorLogin* le enviará al cliente un listado de las salas de juego disponibles [separadas cada una por el carácter *pipe* (|)]; en caso contrario, deberá enviar una cadena que contenga *“-1”*.

Las salas de juego son espacios virtuales donde un grupo de jugadores se reúnen para jugar una partida de *Tado Craft*, y para cada una de estas salas existe un servidor encargada de administrarla. El jugador deberá indicar en qué sala desea jugar, y *ServidorLogin* deberá enviarle la dirección IP del servidor que administra la sala (al cual llamaremos *ServidorPartida*).

Para facilitar su trabajo, usted dispone de la siguiente clase en *ServidorLogin*, la cual tiene métodos para saber si el usuario y contraseña de un jugador son válidos, para obtener un listado de los nombres de las salas de juego disponibles, y para obtener la dirección IP del servidor que administra una sala determinada.



Para la empresa que lo ha contratado, es muy importante que este proceso se realice exitosamente ya que de esto depende que se pueda jugar o no; por tal razón, le han pedido que sea lo más confiable posible, y esto incluye, evidentemente, a los protocolos de comunicaciones a emplear.

Una vez el proceso de login sea exitoso, se podrá dar inicio a una nueva partida. Para esto, deberá conectarse a *ServidorPartida* (del cual obtuvo la dirección IP en pasos anteriores), al puerto 49590. Lo primero que deberá hacer el cliente es enviar su nombre de usuario a *ServidorPartida* para que este sepa qué jugador se ha conectado. Una partida inicia cuando se hayan conectado 5 jugadores.

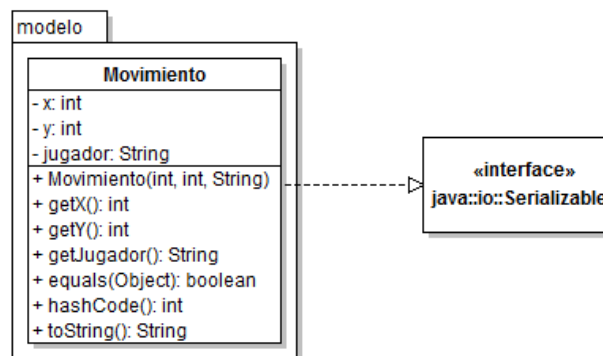
¹ Son videojuegos que requieren que el jugador ponga en práctica sus habilidades de planeamiento y pensamiento para conseguir la victoria. En la mayoría de los videojuegos de estrategia, "al jugador se le concede una vista del mundo absoluta, controlando indirectamente las unidades bajo su poder". Tomando de Wikipedia.

ServidorPartida tiene la responsabilidad de llevar todo el control del juego, y tener registro de todas las acciones que cada jugador realiza. De acuerdo a lo anterior, una de las tareas de ServidorPartida es indicar a cada uno de los jugadores en qué posición del área de juego se encuentran los demás, de forma tal que cada cliente pueda mostrar en su interfaz en qué lugar se encuentra cada jugador. Para esto, cada vez que un jugador se mueve, debe enviar las coordenadas x e y de su ubicación a ServidorPartida, para que este las reenvíe a los demás jugadores. Cuando ServidorPartida reenvía esta información, también deberá indicar el nombre de usuario del jugador que hizo el movimiento, para que de esta forma los demás jugadores puedan identificar a quién pertenece el movimiento.

El cliente deberá enviar su posición empleando el siguiente formato:

x y

Por otro lado, ServidorPartida deberá enviar a todos los jugadores un objeto de la clase modelo.Movimiento.



Una característica importante de este tipo de juegos es que generan una gran cantidad de tráfico en la red por todas las acciones que realiza un jugador, y además de eso, es muy importante que el juego sea lo más fluido posible de forma tal que se pueda ver casi en tiempo real el movimiento de cada jugador.

Su tarea consiste entonces en realizar el análisis y el código para el servidor encargado del login y el servidor encargado de la partida. NO es necesario que haga el código de las clase modelo.Movimiento y control.servidorlogin.UtilidadesServidorLogin. Además, tenga en cuenta que ambos servidores deben estar en la capacidad de atender múltiples clientes simultáneamente.

Criterios de evaluación

1. Análisis (60%)
 - a. Descripción del Protocolo. Incluye MSC y detalle del formato de los mensajes. Es necesario que **ESPECIFIQUE Y JUSTIFIQUE** los protocolos del nivel de transporte a utilizar. (20%).
 - b. Diagrama de clases de ServidorLogin. (20%).
 - c. Diagrama de clases de ServidorPartida. (20%).
2. Código (40%)
 - a. Código fuente de ServidorLogin. (20%).
 - b. Código fuente de ServidorPartida. (20%).