

Trabajo Práctico

Redes y Comunicaciones

Batalla Naval

Ing. Diego Azcurra – Ing. Damián Santos – Lic. Gustavo Fernández

**Alumnos:** Fiorentino, Kevin - 38.562.991

2016

Introducción -

Video Juego Batalla Naval, realizado con una Arquitectura Cliente – Servidor para que dos personas compitan en distintos ordenadores.

Realizado en lenguaje Java, IDE Eclipse.

Desarrollo de la Interfaz del juego utilizando el Paquete: javax.swing.\*;

Implementa un Sistema de seguridad Encriptando y Desencriptando los mensajes en Base64 para un envío seguro de los mismos.

**Comunicación:**

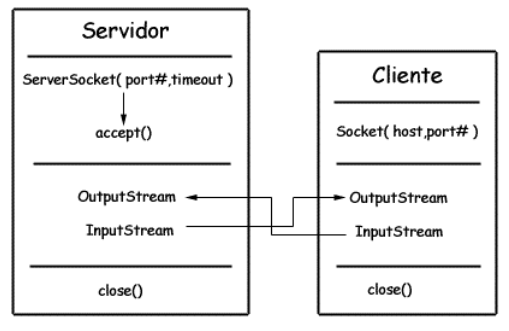
Establecida importando el Paquete **java.net** de la plataforma Java que proporciona una Clase **Socket**, la cual implementa una de las partes de la comunicación bidireccional entre un programa Java y otro programa en red.

Además java.net incluye la Clase **ServerSocket**, la cual implementa un socket el cual los Servidores pueden utilizar para escuchar y aceptar peticiones de conexión de cliente.

**Funcionamiento:** El Servidor establece un puerto y espera a que el Cliente establezca la conexión. Cuando el Cliente solicite una conexión, el servidor abrirá la conexión Socket con el método ACCEPT().

El Cliente para conectarse al Servidor deberá introducir Número de Puerto y Host (siendo “localhost” por defecto o con la IP del ordenador donde se encuentre el Servidor).

El Cliente y el Servidor se comunican con manejadores **InputStream** y **OutputStream** del paquete java.io.\*;



Desarrollo del Juego –

La aplicación consta de 4 Clases (Cliente, Servidor, Main, Hilo).

Cuando el Servidor se inicia, se queda esperando por la llegada de los dos jugadores, recibiéndolos, asignándoles una ID y un Hilo de ejecución que lo atenderá.

El Hilo es quien se ocupa de la lógica del juego del lado del Servidor, mientras que el propio Cliente se ocupa de la otra parte de la lógica del Juego. Por otro lado el Main es la Interfaz del Cliente. A partir de aquí la comunicación se realizara entre Cliente – Hilo.

A continuación se relata Detalladamente los diferentes mensajes que se intercambian entre el Servidor (Hilo) y el Cliente, y viceversa. Cada mensaje es identificado de un lado u del otro por medio de un Comando de 3 caracteres, seguido por la información necesaria para el mensaje dependiendo el Tipo. *Por Ejemplo*: Si el mensaje se trata de la posición de un barco, el Comando viene acompañado por la Fila, Columna, la Dirección, la ID del Cliente y el Tamaño del Barco.

Los mensajes no son todos del mismo tamaño, algunos Comando vienen acompañado solo por la ID del Cliente o por nada, como los mensajes que notifican el final de una partida o el reinicio de la misma.

**Intercambio de mensajes** –

***Entre Servidor 🡪 Cliente:*** Mensajes que emite y envía el Servidor hacia el Cliente.

--Confirmación del Posicionamiento de un Barco – para que el cliente “Pinte” las casillas correspondientes a las posiciones del barco en su Tablero.

*Msj*: Comando + Fila + Columna + IDCliente (*Tamaño Mensaje: 8 caracteres*)

*Comando* = “POS”

*Fila*: Representa la letra del tablero del eje vertical en forma numérica (*EJ*: G = 07)

*Columna*: Representa el número del tablero del eje horizontal (*EJ*: 04)

*IDCliente*: Id del Cliente que emitió el mensaje con las coordenadas de la posición de su barco.

--Desconexión de un Cliente – Notifica al Cliente que su rival se desconecto para que reinicie su Interfaz.

*Msj*: Comando = “OUT” (*Tamaño Mensaje: 3 caracteres*)

--Registro de un Disparo – Luego de registrar el disparo en las matrices del Servidor, registra si fue Agua o Tocado y le avisa a lo Clientes para que “Pinten” la casilla correspondiente.

*Msj*: Comando + Fila + Columna + IDCliente + Turno (*Tamaño Mensaje: 9 caracteres*)

*Comando*: “AGU” o “TOC”

*Fila/Columna*: Ídem anterior

*IDCliente*: para identificar de quien fue el disparo y revisar la matriz que le corresponda

*Turno*: Cambiar el turno después de cada disparo

--Final del Juego – Notifica a cada Cliente si Gano o Perdió la partida, luego se reinicia el juego.

*Msj*: Comando + IDGanador (*Tamaño Mensaje: 4 caracteres*)

*Comando*: “FIN”

*IDGanador*: ID del ganador

***Entre Cliente 🡪 Servidor:*** Mensajes que emite y envía el Cliente hacía el Servidor

--Efectuar un disparo – Envío de las coordenadas del disparo para que el servidores lo registre y avise si fue Agua o Tocado.

*Msj*: Comando + Fila + Columna + IDCliente (*Tamaño Mensaje: 8 caracteres*)

*Comando*: “DIS”

*Fila/Columna*: Ídem anterior

*IDCliente*: ID del Cliente que emitió el disparo.

--Notificación de Reinicio del juego – Avisa al Servidor que reinicie las matrices del juego por finalización de la partida o desconexión de un Cliente.

*Msj*: Comando = “REI” (*Tamaño Mensaje: 3 caracteres*)

--Posicionamiento de un Barco – Primera coordenada de la posición que ocupará el barco, el Servidor se encarga de completar hacia la Derecha o hacia abajo dependiendo la Dirección y el Tamaño del Barco.

*Msj*: Comando + Fila + Columna + IDCliente + Dirección + Tamaño Barco

(*Tamaño Mensaje: 10 caracteres*)

*Comando*: “POS”

*Fila/Columna*: Ídem anterior

*IDCliente*: Id del Cliente que posiciona el barco.

*Dirección*: Horizontal o Vertical.

*Tamaño Barco*: Cantidad de casillas que ocupa el barco (5, 4, 3 o 2)

Como Jugar –

El Juego inicia teniendo que posicionar los Barcos propios uno por uno, para cada uno se solicitara la primera coordenada de la posición del barco (EJ: k3) seguido de la dirección que ocupara (Vertical u Horizontal) si las posiciones están Desocupadas y no se excede del tablero (12 x 12) el barco se posicionara Satisfactoriamente en el tablero del jugador.

*Tipos de Barcos*: Portaaviones (5 casilleros)

Acorazado (4 casilleros)

Submarino (3 casilleros)

Destructor (2 casilleros)

Luego de que los barcos estén en posición, ambos jugadores comenzaran a efectuar disparos por turnos. Cada disparo propio se registra en el segundo tablero, dependiendo si se dio en el blanco o el disparo fallo, la casilla se pinta de diferentes colores.

*Colores del Tableros*: Negro (Barco Propio) – Celeste (Agua) – Rojo (Tocado)

Los jugadores deberán estar atentos de no disparar dos veces a la misma casilla ya que repetir un disparo lo hará perder su turno.

El juego finalizara cuando uno de los dos jugadores se quede sin ninguna parte de sus Barcos en buen estado, dando por Ganador al otro Jugador y reiniciándose el juego.

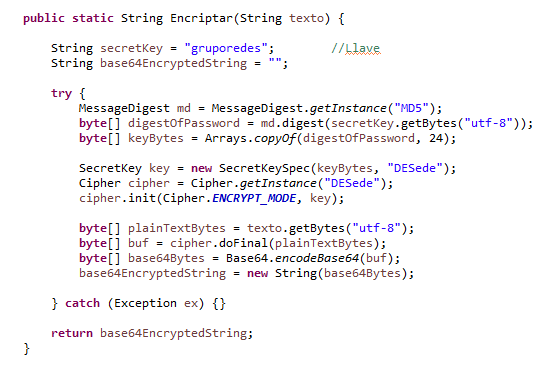
Seguridad

Para desarrollar el trabajo y aplicar una medida de seguridad decidimos optar por Apache *CommonsCodec*, un software que le permite al usuario utilizar Codificadores y Decodificadores en *Base64*.

*Librería Java*: commons-codec-1.9.jar

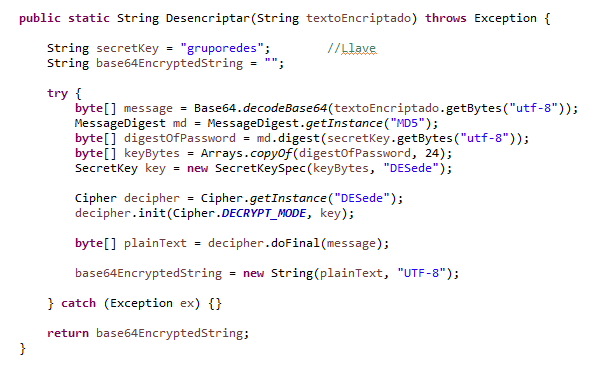
**Base 64** es un [sistema de numeración](https://es.wikipedia.org/wiki/Sistema_de_numeraci%C3%B3n) [posicional](https://es.wikipedia.org/wiki/Notaci%C3%B3n_posicional) que usa 64 como base. Es la mayor potencia de dos que puede ser representada usando únicamente los caracteres imprimibles de [ASCII](https://es.wikipedia.org/wiki/ASCII). Esto ha propiciado su uso para codificación de [correos electrónicos](https://es.wikipedia.org/wiki/Correo_electr%C3%B3nico), [PGP](https://es.wikipedia.org/wiki/PGP) y otras aplicaciones. Todas las variantes famosas que se conocen con el nombre de Base64 usan el rango de caracteres A-Z, a-z y 0-9 en este orden para los primeros 62 dígitos, pero los símbolos escogidos para los últimos dos dígitos varían considerablemente de unas a otras.

Métodos de Encriptación: Recibe un texto, utilizando la llave, codifica el mensaje y lo devuelve Encriptado.



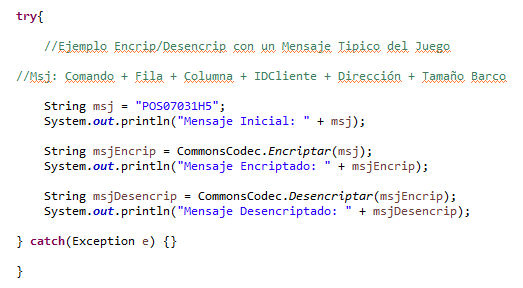
Establecemos una Clave común para encriptar y desencriptar los datos. De esta forma, el usuario no podrá hacer ninguna de las dos si no posee la clave correcta.

Método de Desencriptación: Recibe el mensaje encriptado y lo convierte a su forma original siempre y cuando la Llave sea correcta.



Devuelve la cadena de caracteres desencriptada para que el programa pueda identificar cuál fue la acción y así seguir corriendo.

Ejemplo de Encriptación y Desencriptación de un mensaje típico del Juego:



Resultado:



***Bibliografía Seguridad:***

<https://commons.apache.org/proper/commons-codec/>

https://es.wikipedia.org/wiki/Base64