

BATALLA NAVAL

Juego simple usando comunicación vía red usando Sockets



INTEGRANTES

AVALOS CHINO MARICIELO
AVILA HUATUCO RIHANA
GALVAN SALVADOR CAMILA LIZBETH
GASPAR GABRIEL JORGE VICTOR
RAMIREZ CHAVE ANGELA CAMILA
ROSALES ZUASNABAR JHENRY ROVER

18190085
18190338
18190328
18190321
18190316
18190343

5 PTOS
5 PTOS
5 PTOS
5 PTOS
0 PTOS
5 PTOS



Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América.

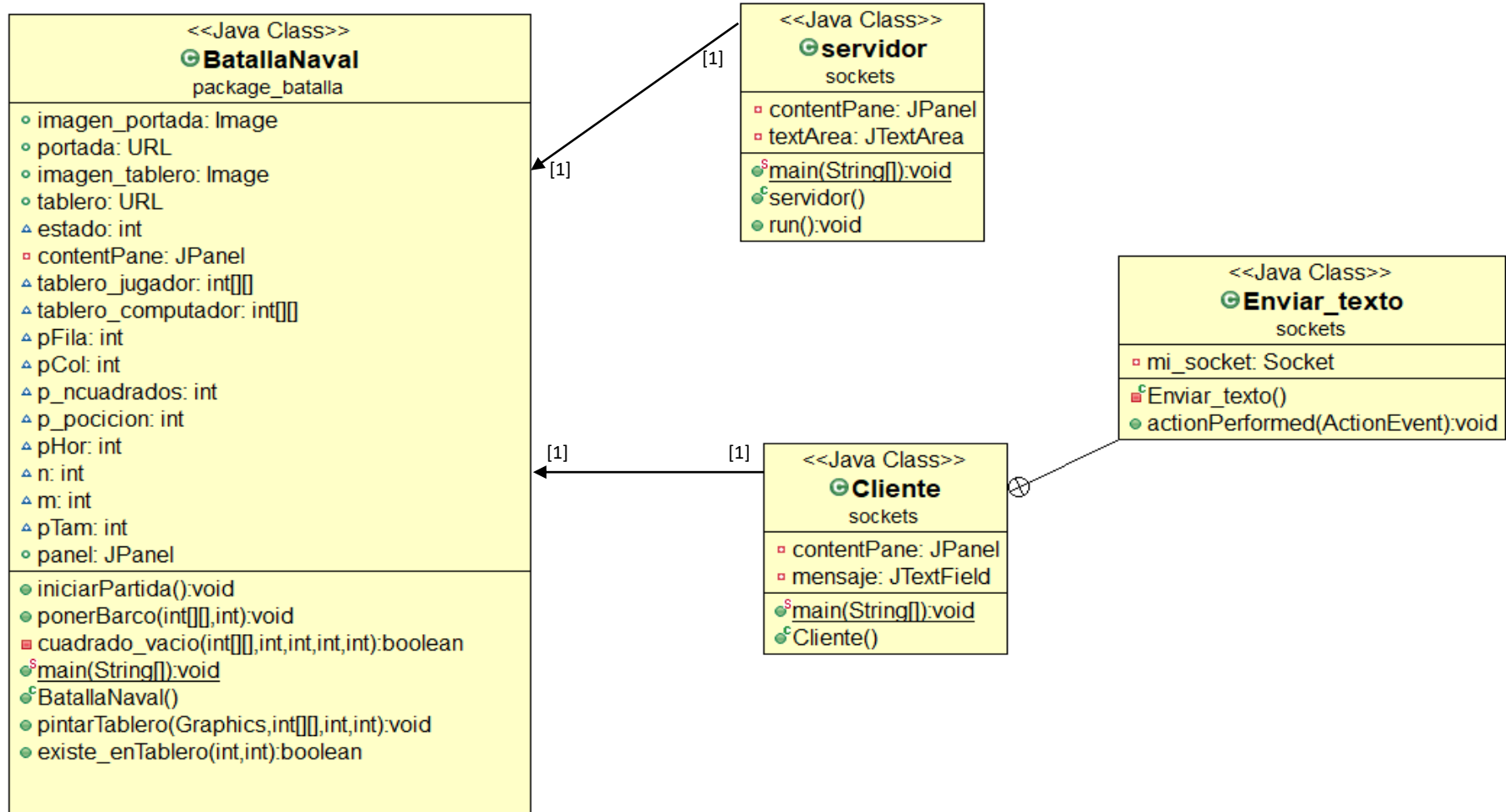
PROBLEMÁTICA

Una necesidad en los juegos y aplicativos actuales radica en que estos tengan la capacidad de permitir la comunicación entre distintos usuarios con distintas ubicaciones, es por ello, que al ver que en la programación normal de un juego del tipo Battleship (Batalla Naval) nos damos cuenta que con el uso de sockets del tipo orientado a conexión permitirá la emisión y recepción de datos entre diferentes máquinas

MOTIVACIÓN

Nos motiva principalmente la aplicación de los diversos conocimientos adquiridos como parte del desarrollo del curso, estos incluyen el manejo de métodos, diagramas, interfaces gráficas de usuarios, flujos de entrada y salida así como el manejo de errores y excepciones.

DIAGRAMA DE CLASES



CONTENIDOS APLICADOS

Para el desarrollo de nuestro proyecto hicimos uso de varios de los conocimientos aportados durante el curso así como también algunos en base a nuestra investigación.

Entre las técnicas utilizadas tenemos:

- El uso de interfaces
- Adapters
- Type casting
- Hilos
- Sobreescritura de métodos
- Una gran variedad de librerías de Java
- Sockets

```
int i,j,posicion;
do{
    i=(int)Math.random()*8;
    j=(int)Math.random()*8;
    posicion=(int)Math.random()*2;
}while(!
```

```
);
//el mouse no salga fuera del tablero
addMouseListener(
    new MouseMotionAdapter() {
        @Override
        public void mouseMoved(MouseEvent e){
            int x=e.getX();
            int y=e.getY();
            if(x>=115 && y>=250 && x<115+30*8
```

```
    y<250+30*8);
```

```
package batalla;
import java.awt.Container;
import java.awt.EventQueue;
import java.awt.Graphics;
import java.awt.Image;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JPanel;

import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.net.URL;
```

```
addMouseListener(
    new MouseAdapter(){
        public void mouseClicked(MouseEvent e){
            if (estado==0){
                estado=1;
                iniciarPartida();
                repaint();
            }
            if (estado==1){
                int pDF=0;
                int pDC=0;
                if (pHor==1){
                    pDF=1;
                }else{
                    pDC=1;
                }
                for(int
                    w=pFila;n<=pFila+pTam*pDF;n++){
                        for (int
                            k=pCol;n<=pCol+pTam*pDC;n++){
                                tablero_jugador[n]
                                    [m]=pTam;
                            }
                        }
                    }
                }
            }
        }
    );
```

```
54 private JTextArea textArea=new JTextArea();
55
56 @Override
57 public void run() {
58     try {
59         ServerSocket servidor =new ServerSocket(6666);
60         while(true){
61             Socket mi_socket=servidor.accept();
62             DataInputStream flujo_entrada=new DataInputStream(mi_socket.getInputStream());
63             String mensaje_entrada=flujo_entrada.readUTF();
64             textArea.append("\n"+mensaje_entrada);
65             mi_socket.close();
66         }
67     } catch (IOException e) {
68         // TODO Auto-generated catch block
69         e.printStackTrace();
70     }
```

DEMOSTRACIÓN EN VIVO

