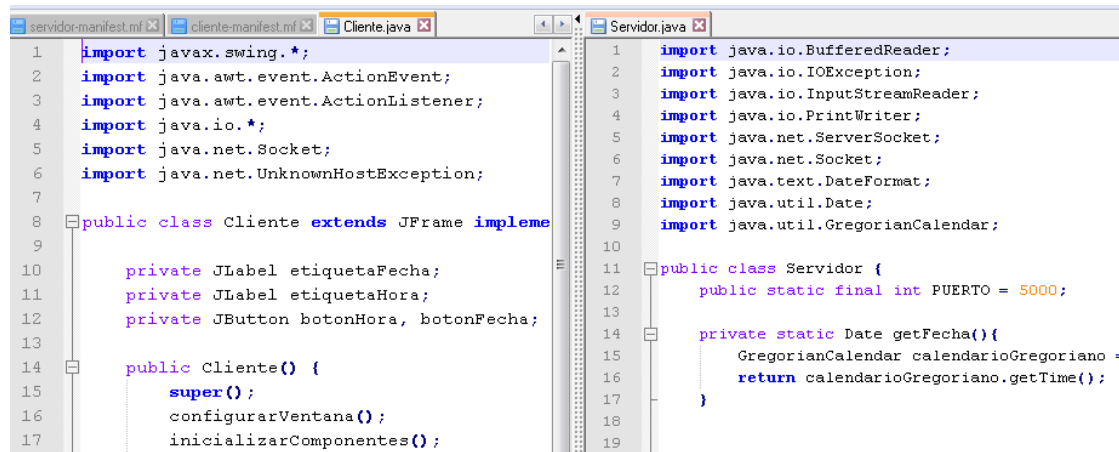


Objetivo

Desarrollar un programa utilizando arquitectura cliente-servidor, que se comunicarán a través de sockets. El cliente debe disponer de una interfaz gráfica con dos botones, uno desplegará la hora y el otro la fecha. El servidor debe proveer la hora y la fecha según corresponda

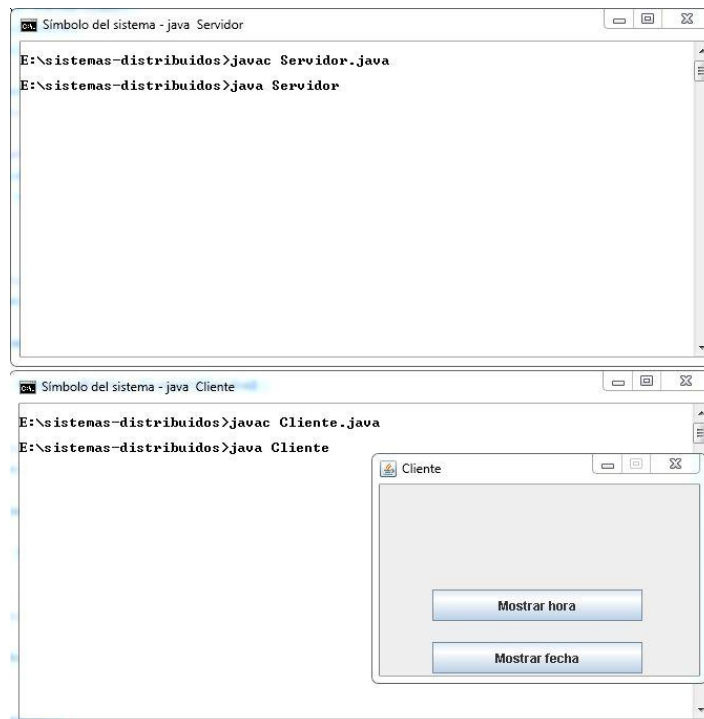
Desarrollo

1: El desarrollo se llevó acabo en Notepad++



```
servidor-manifest.mf cliente-manifest.mf Cliente.java Servidor.java
1 import javax.swing.*;
2 import java.awt.event.ActionEvent;
3 import java.awt.event.ActionListener;
4 import java.io.*;
5 import java.net.Socket;
6 import java.net.UnknownHostException;
7
8 public class Cliente extends JFrame implements
9
10     private JLabel etiquetaFecha;
11     private JLabel etiquetaHora;
12     private JButton botonHora, botonFecha;
13
14     public Cliente() {
15         super();
16         configurarVentana();
17         inicializarComponentes();
18
19
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4 import java.io.PrintWriter;
5 import java.net.ServerSocket;
6 import java.net.Socket;
7 import java.text.DateFormat;
8 import java.util.Date;
9 import java.util.GregorianCalendar;
10
11 public class Servidor {
12     public static final int PUERTO = 5000;
13
14     private static Date getFecha(){
15         GregorianCalendar calendarioGregoriano =
16             return calendarioGregoriano.getTime();
17     }
18
19
```

2: Para compilar se ejecutó el comando *javac* y se ejecutó con el comando *java*



```
Símbolo del sistema - java Servidor
E:\sistemas-distribuidos>javac Servidor.java
E:\sistemas-distribuidos>java Servidor

Símbolo del sistema - java Cliente
E:\sistemas-distribuidos>javac Cliente.java
E:\sistemas-distribuidos>java Cliente
```

The screenshot also shows the resulting GUI window titled 'Cliente'. It contains two buttons: 'Mostrar hora' and 'Mostrar fecha'.

3: Se comprueba el funcionamiento del botón *Mostrar hora*



4: Se comprueba el funcionamiento del botón *Mostrar fecha*



5: Procedemos a empaquetarlos como *jar*, añadiendo a cada uno su respectivo manifiesto y volviendo a ejecutar para validar su correcto funcionamiento

