MANUAL TÉCNICO

INDICE

Tecnologías utilizadas 3

Programas utilizados: 3

Lenguajes utilizados: 4

Diagramas 5

Diagrama de clases Servidor 5

Diagrama de clases Cliente 6

Casos de uso 7

Modo de comunicación 8

Pruebas 9

Interfaz 10

Interfaz del cliente 10

Interfaz del Servidor 11

Código 12

Javadoc 12

GitHub 12

TECNOLOGÍAS UTILIZADAS

PROGRAMAS UTILIZADOS:

- **Eclipse:** para el desarrollo de la aplicación para dispositivos móviles, empleando el SDK de android para las pruebas del programa. Se ha elegido este y no Android Studio por su larga trayectoria y extensa comunidad de usuario.

- **Netbeans:** ha sido donde se ha realizado la parte para ordenadores, se ha elegido este sistema por la gran variedad de componentes incorporados por defecto para el tratamiento de la interfaz gráfica. También se ha elegido por la posibilidad del uso de otros lenguajes como xml, php, js…

- **Visual paradigm**: utilizado para la creación de los diferentes diagramas, ya que ofrece herramientas muy potentes para la generación de los mismos.

- **VirtualBox:** para las pruebas del servidor sin necesidad de disponer de ordenadores físicos para ello, además de la posibilidad de disponer en un mismo sistemas, sistemas operativos diferentes. Concretamente se ha probado en: Windows XP SP3, Windows 7, OS X 10.7, ubuntu 14.04.

- **Wireshark:** Se ha elegido para la visualización de los paquetes que circulan por la red y para la optimización de la misma.

LENGUAJES UTILIZADOS:

- **Java:** se ha utilizado java tanto para el desarrollo de la parte cliente como la del servidor. Esto ha sido así por la ventaja de interconexión que ofrece el propio lenguaje, no generando incompatibilidades como podrían surgir en el caso de utilizar lenguajes diferentes. Siendo este un lenguaje multiplataforma podemos adaptar el programa para su funcionamiento en otros sistemas más fácilmente.

- **XML:** empleado para la generación de plantillas y lectura de ficheros de configuración. También utilizado para la creación de la interfaz gráfica tanto en el servidor como en el cliente.

- **UML:** para el diseño de los diagramas, tanto de clases como de casos de uso.

- **HTML5, JS y PHP:** estos lenguajes, en menor medida que los anteriores fueron usados para la creación del sitio web del programa, también para la generación online de plantillas para los usuarios.
  Así mismo también se ha utilizado HTML5 y JS gracias a su gran expansión en los principales dispositivos, para la creación de una aplicación cliente multiplataforma, sin necesidad de instalación del programa.

DIAGRAMAS

DIAGRAMA DE CLASES SERVIDOR

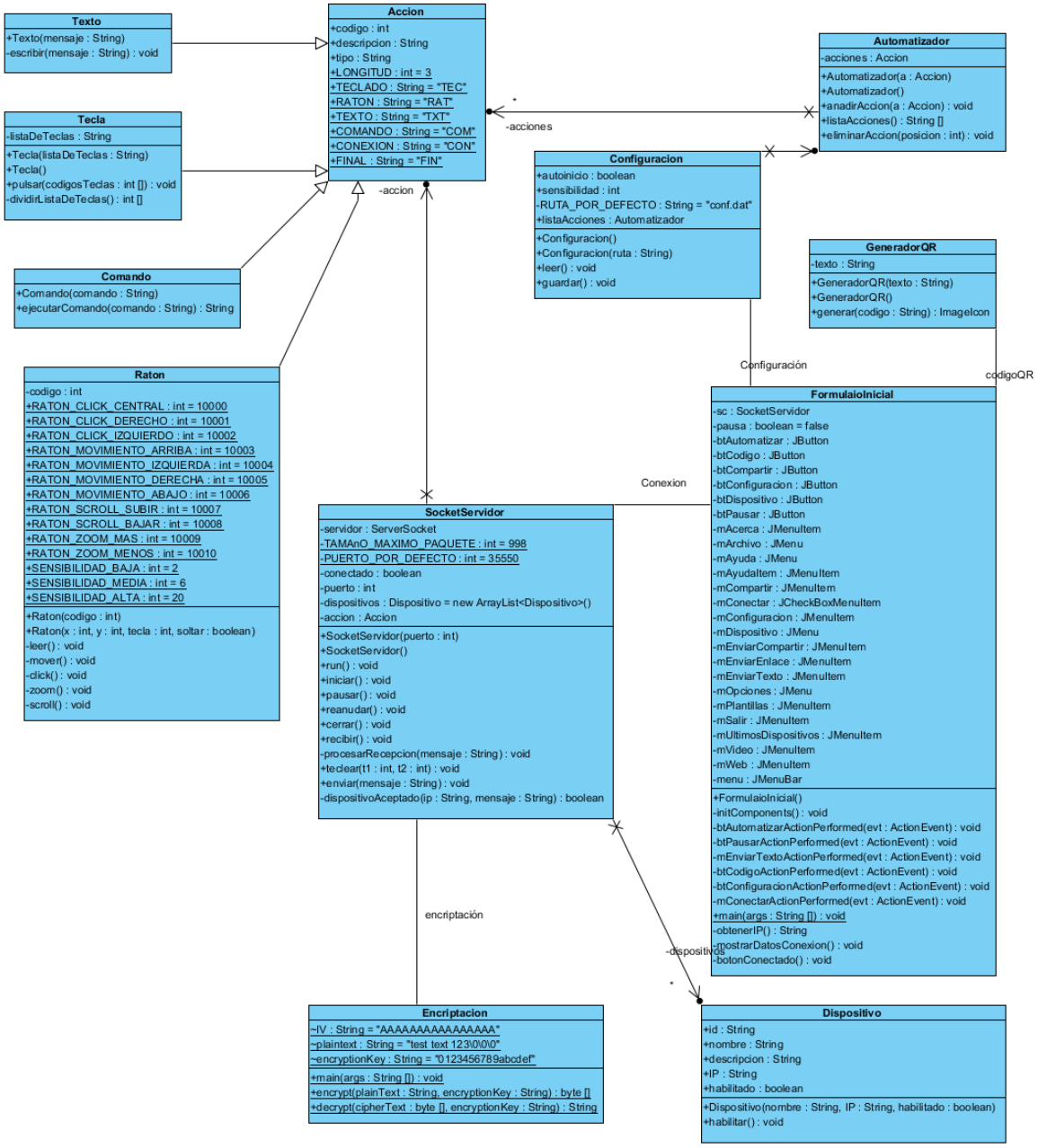
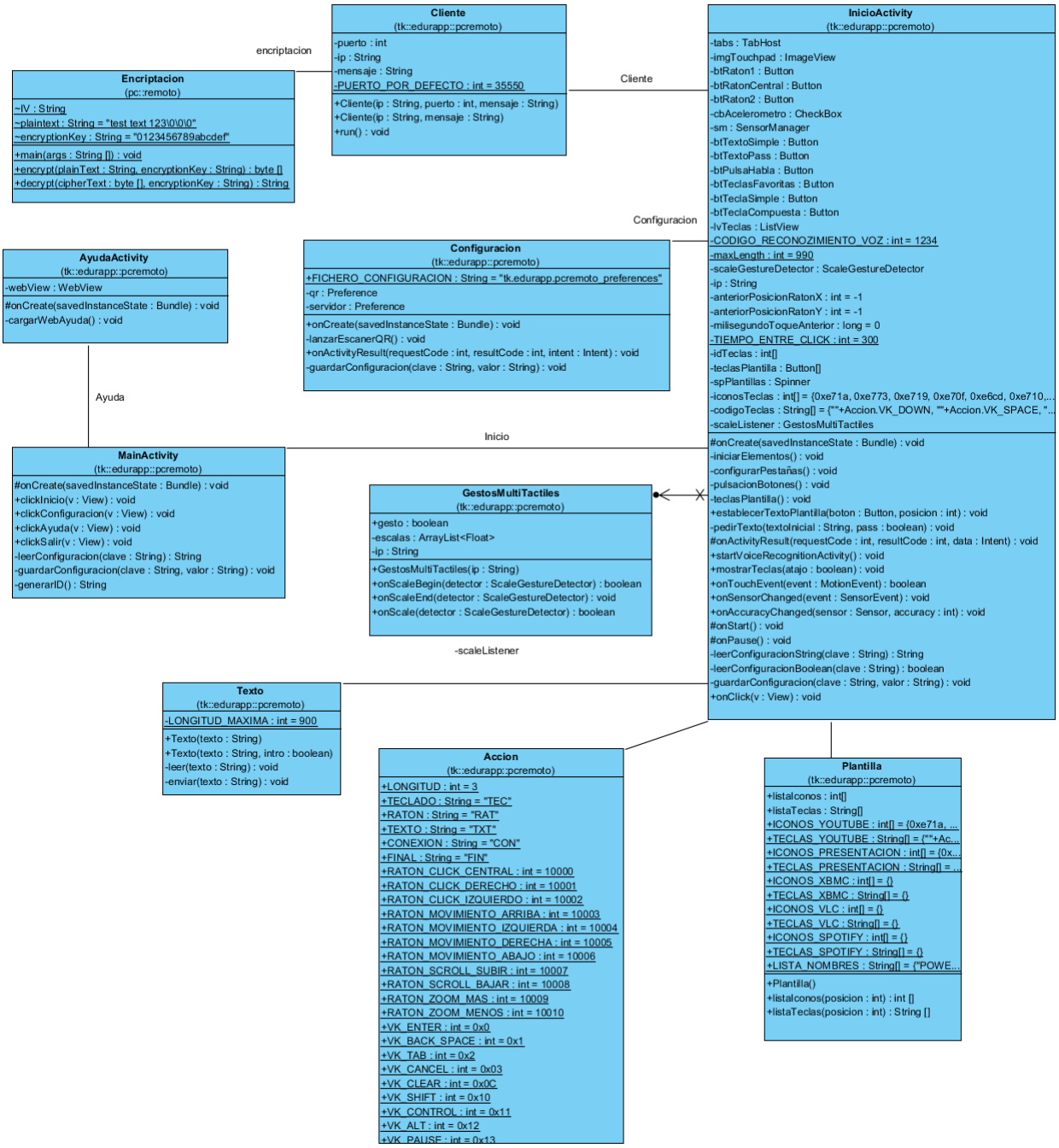
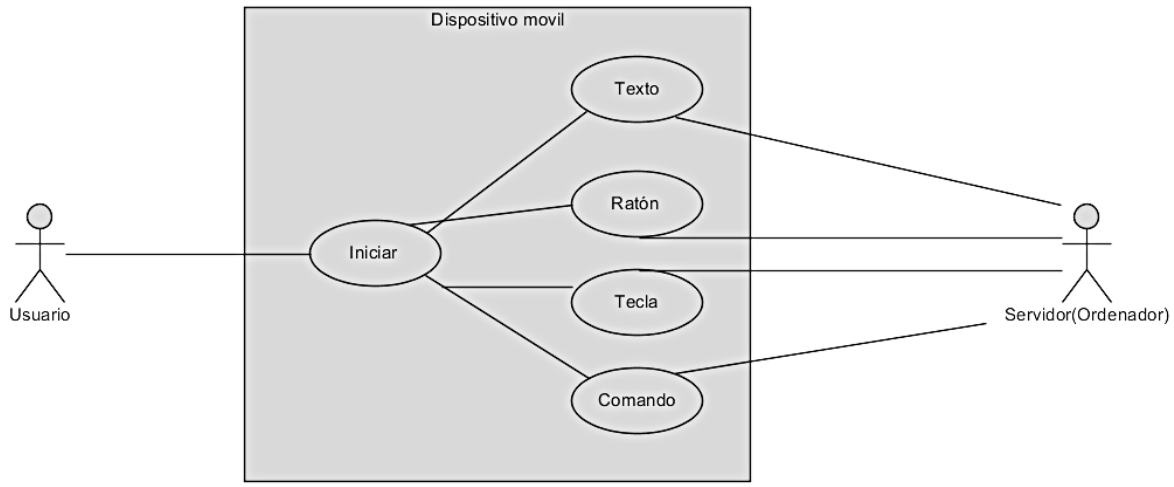


DIAGRAMA DE CLASES CLIENTE



CASOS DE USO

Diagrama de casos de uso del envió de datos al servidor desde el dispositivo móvil.



MODO DE COMUNICACIÓN

Para la comunicación de los dos roles que intervienen en la aplicación se ha optado por la comunicación por medio de ***sockets***, esta conexión se realiza por medio de una red local a la que deberán de estar conectado los dispositivos que la utilice, algo habitual en los

El protocolo utilizado para la transmisión de paquetes por la red ha sido el protocolo ***UDP*** (User Datagram Protocol), se ha utilizado este en lugar del TCP por su menor sobrecarga de la red, ya que el protocolo TCP en determinados momentos producía un uso excesivo como se puede comprobar en la comparativa entre los paquetes que circulan por la red de un modo y del otro en el envío de un mismo mensaje.

La clase empleada para la conexión de red en java ha sido ***DatagramSocket*** (java.net.DatagramSocket), ya que es la que gestiona los mensajes que se transmiten por este protocolo y nos permite enviar y recibir paquetes.

Para el manejo desde el servidor de los periféricos del ordenador, se ha empleado la clase ***Robot*** (java.awt.Robot), esta clase nos permite controlar diferentes elementos del ordenador, como su teclado, ratón, contenido en pantalla, … En la parte del cliente se ha tenido que incorporar con unas pequeñas modificaciones la clase ***KeyEvent*** (java.awt.event.KeyEvent), que es la encargada del control de los eventos de las teclas, ya que android no contiene el paquete java.awt, a nueva clase creada principalmente a partir de las constantes de KeyEvent la hemos denominado Accion.

PRUEBAS

La fase de pruebas se ha desarrollado a medida que la aplicación iba incorporando nuevas funcionalidades, comprobando su correcto funcionamiento, tanto en la parte de cliente como en la del servidor.

La parte desarrollada para dispositivos móviles ha sido probada en dispositivos reales, tanto Smartphone con las resoluciones factores de pantalla más comunes como tablets de 7“ y 10”. En ciertos casos se añadieron logs a las diferentes partes del programa susceptibles de problemas, para su posterior evaluación.

En cuanto a las pruebas realizadas sobre los equipos servidores se ha optado por pruebas físicas en los sistemas de los que se dispone, en este caso solo Windows y Ubuntu, y por pruebas en máquinas virtuales de otras versiones como es el caso de OS X.

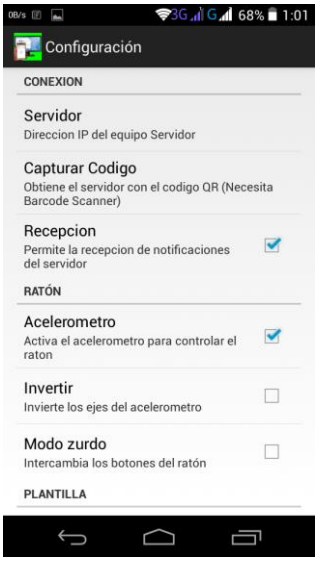
También se fue escribiendo por consola los diferentes mensajes que iba recibiendo del dispositivo para verificar su correcta recepción.

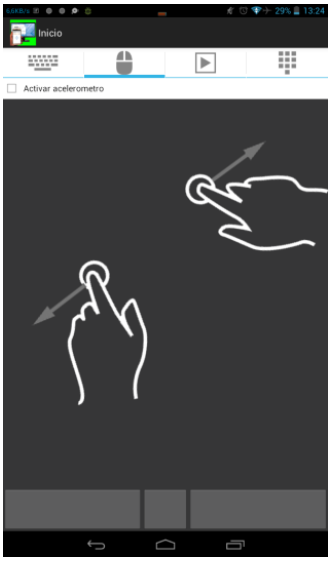
También se han realizado pruebas de consumo de recursos de red, tanto en los dispositivos móviles gracias, obteniendo en tiempo real en la barra de estado, el consumo de red actual que genera el dispositivo gracias a gravity box. En los ordenadores se utilizó el gestor de red para obtener los recursos de red obtenidos.

INTERFAZ

INTERFAZ DEL CLIENTE

En cuanto a la interfaz del cliente se ha optado por una interfaz personalizada para mantenerla en futuras versiones del programa en diferentes dispositivos. En cuanto a la pantalla de control se ha optado por un diseño dividido en pestañas, para poder alternar de una función a otra función de manera rápida.





INTERFAZ DEL SERVIDOR

Se ha diseñado un menú principal en el que destacan las 6 funciones principales, de esta forma el usuario dispondrá de las funcionalidades principales a un simple toque , aunque también se ha incorporado un menú tradicional en el que se encuentran otras funciones.

CÓDIGO

JAVADOC

Para la comprensión del código y de los diferentes métodos y objetos que forman parte del programa, se pone a disposición el documento java doc en la siguiente dirección

http://www.pc-remoto.tk/javadoc

GITHUB

El código completo del programa estará accesible a través de un repositorio público en:

https://github.com/edunaveira/pc-remoto/