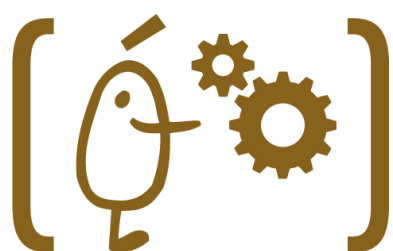
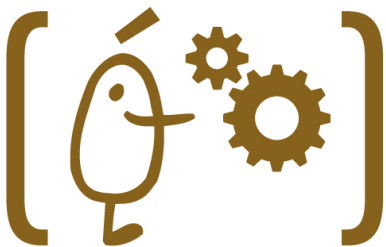


Manual de usuario



Alejandro Mengual
Carlos Peris
Raul Pina
David Sanchis

1. **Instalación**
 - 1.1. **Windows**
 - 1.2. **Android**
2. **Uso de la aplicación**
 - 2.1. **Una máquina**
 - 2.2. **Remoto**
3. **Interpretación de los resultados**
 - 3.1. **Estadísticas**
 - 3.1.1. **Errores en grupos determinados**
4. **Apéndice**
 - 4.1. **Apertura de puertos**
 - 4.2. **Interfaz**



Instalación

En la carpeta software se puede encontrar todos los archivos necesarios para adquirir nuestro proyecto.

Windows

Para ejecutar la aplicación desde Windows, se deberá tener instalado Netbeans. A nosotros no nos ha dado particularmente problemas con ninguna versión, ya que hay miembros del equipo usamos 8.2 y otros 12.1 y no hemos tenido ningún problema. Sin embargo, proponemos no hacer uso de una versión más actual porque no sabemos con certeza si puede haber algo que no funcione.

El instalador de netbeans se puede descargar mediante este enlace:

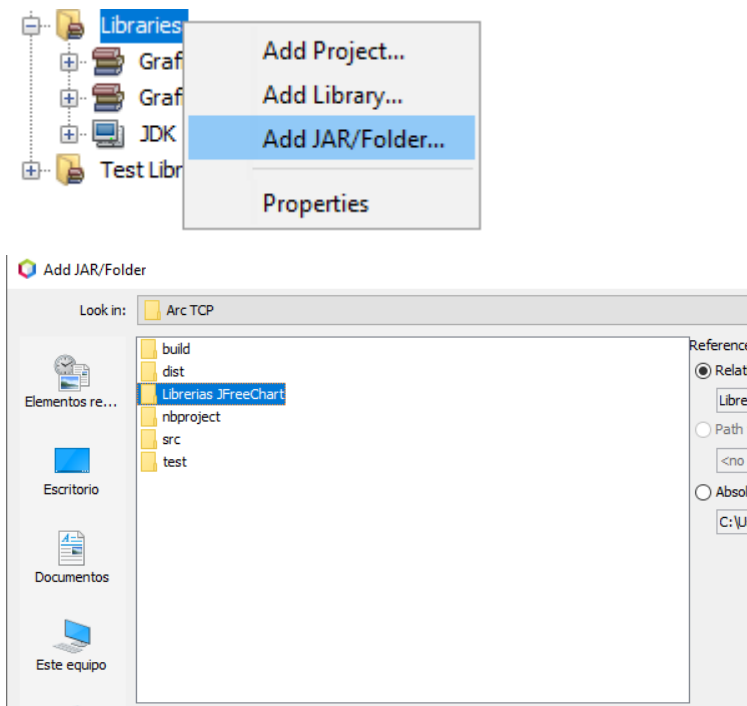
<https://netbeans.apache.org/download/index.html>

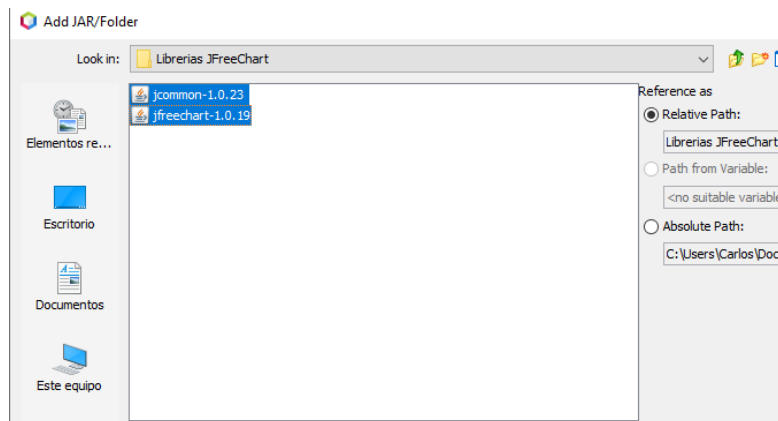
Una vez tenemos el netbeans, importamos el proyecto con el icono:



Buscamos el directorio donde lo hayamos descargado, y lo abrimos.

Las librerías que serán necesarias importar se encuentran dentro de la carpeta del proyecto. Deberán ser añadidas para no generar errores.





Android

En la carpeta del proyecto, hay un archivo llamado: ARC2021

Este archivo deberá ser transportado al móvil. Una vez dentro, desde el móvil se busca en el Gestor de Archivos y desde ahí, se clica al ejecutable, el cual nos instalará la aplicación. Ya instalada, podremos abrirla y usarla.

Uso de la aplicación

La aplicación se trata de una simulación en la que hay muchos clientes conectados y van cambiando sus coordenadas en el espacio. Cuando lo hacen, tienen que avisar a todos los vecinos (mediante el uso de un servidor), y estos responderle para que ese cliente que se haya movido sepa que el resto le han escuchado. Cada cliente se mueve asincrónicamente y de forma independiente al resto, además realizará el cambio de coordenadas un número de veces indicado.

Al finalizar todas las iteraciones, los clientes calcularán su latencia media (la cual mide el tiempo desde que envían el mensaje hasta que todos le contestan).

Los clientes se dividirán en grupos independientes y sin comunicación entre ellos, de tal manera que un cliente de un grupo solo se comunicará con los de su propio grupo.

Cuando un grupo acabe, se mostrará su latencia media. Y cuando acabe la simulación por completo, se mostrará la latencia media de la aplicación.

Una máquina

Para usar el sistema funcionando en una sola máquina se debe abrir el servidor y los clientes desde el mismo PC. Para ello, ejecuta primero el Servidor.

Esta instrucción abre un proceso que actúa como Servidor. Una vez iniciado, por consola se deberá introducir los clientes y grupos deseados:

```
Proyecto ARC 2021.
```

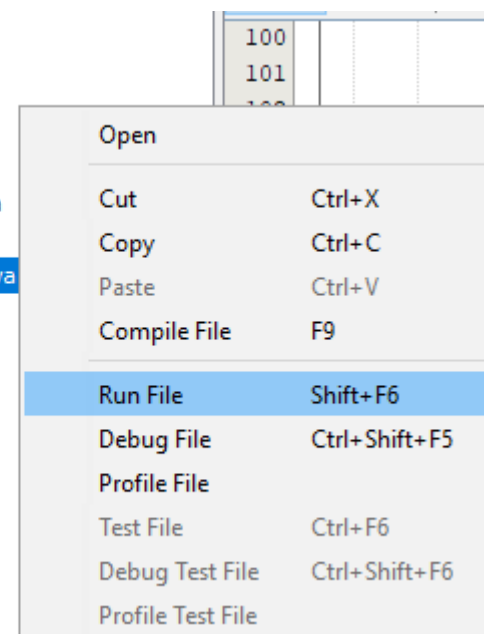
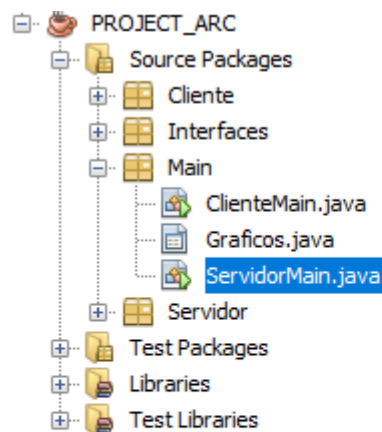
```
Prototipo 1.
```

```
Realizado por: Carlos, David, Alex, Sergio y Raúl.
```

```
Servidor iniciado
```

```
Inserte numero de Clientes: 100
```

```
Inserte numero de Grupos: |
```



Hay que tener especialmente cuidado en los siguientes puntos:

- El número de clientes debe ser divisible entre el número de grupos
- El número de clientes por grupos (la división entre clientes y grupos) no debe ser superior a 15 ni inferior a 5:

```
Servidor iniciado
```

```
Inserte numero de Clientes: 100
```

```
Inserte numero de Grupos: 50
```

```
¡Al carajo! ¡Solo puede haber entre 5 y 15 clientes por grupos!
```

```
BUILD SUCCESSFUL (total time: 2 minutes 17 seconds)
```

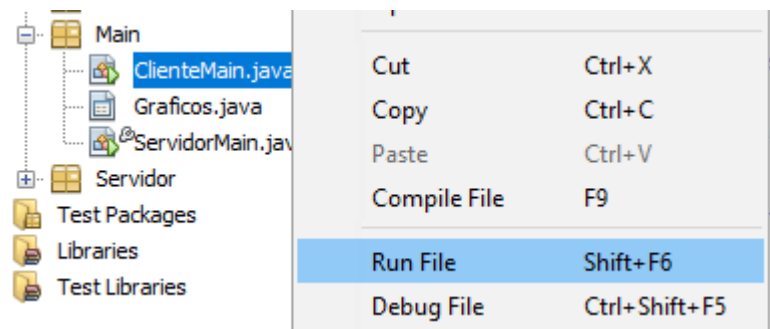
- Como estamos en local, es recomendable no iniciar la interfaz y actuar por consola. Por ello, no debemos tocar esta variable (la cual se encuentra en ServidorMain y es la primera instrucción de todas):

```
/**
 * Variable -> "control_GUI"
 * true para control por vista gráfica
 * false para control por consola normal
 */
boolean control_GUI = false;
```

Una vez iniciado el servidor, este se mantendrá a la espera de clientes, y no iniciará la simulación hasta que le lleguen todos los que se le indicó que esperase. Una vez lleguen todos, los dividirá en grupos y avisará a todos los clientes a la vez de que pueden comenzar.

Para conectar clientes, se debe ejecutar ahora el proceso que lanza a los clientes.

Una vez abierto, se escribe el número de clientes que deseamos conectar desde ese proceso (lo más sencillo es poner todos en el mismo, pero no hay problema en ejecutar 2 instancias de ClienteMain y dividir los clientes mandados entre ambos procesos). Ahora debemos indicar también el número de iteraciones a ejecutar.



```
Inserte numero de Clientes: 100
Inserte numero de Grupos: 10
Inserte numero de Iteraciones:
```

Hay que tener especialmente cuidado en que:

- Los clientes finales que mandemos entre todos los procesos que iniciemos sean coherentes con lo especificado al servidor.
- Que el número de grupos sea el mismo.
- Conectarse al servidor introduciendo el campo HOST a localhost (se encuentra al principio de ClienteMain).

```
public class ClienteMain {
    //ELEGIR IP:
    private static final String HOST = "localhost";
    //private static final String HOST = "arc.alexms.es";
}
```

Una vez conectados los clientes, la simulación comenzará y, al acabar, mostrará las latencias por grupo y la latencia global (en la consola de servidor) y las latencias de cada cliente (en la consola de cliente).

Remoto

El funcionamiento en remoto es muy similar al funcionamiento en una sola máquina. El cambio es que quien lance clientes deberá especificar en el campo HOST de ClienteMain (el mostrado anteriormente) la IP pública de quien actúe de servidor. Este último, deberá abrir los puertos a usar desde su router (ver en apéndice tanto la apertura de puertos como averiguar la IP pública).

Es importante tener en cuenta la calidad de la red, puesto que según la de quien la lance, el hardware que la lanza tendrá unas limitaciones u otras a la hora de lanzar y gestionar clientes.

Para evitar el problema de que el internet se colapse y por ello haya clientes que pierdan la conexión (generando así que la simulación no acabe nunca al no recibir de esos clientes el aviso de que han acabado), existe un mecanismo que detecta que dichos clientes han perdido la conexión y por ello se les ha ignorado para seguir con la ejecución.

Pero pese a existir este sistema, lo más recomendable es que cada persona que lanza clientes sepa más o menos sobre qué límites su dispositivo aguanta, puesto que de esa manera los datos en la simulación serán significativamente más valiosos.

Interpretación de los resultados

Una vez terminada la simulación, los datos estadísticos se guardan en ficheros. A raíz de ellos se puede sacar información sobre cómo ha ido la ejecución desde diferentes puntos de vista.

Estadísticas

La aplicación guarda datos sobre cómo fueron las simulaciones, y los guarda para mostrar en el momento deseado 3 gráficas:

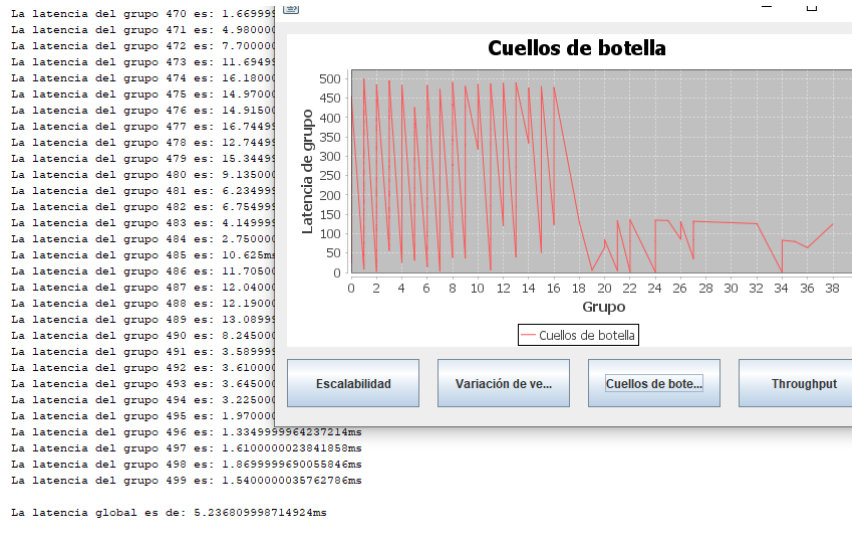
- Escalabilidad
- Aumento de vecinos
- Throughput

Aunque existe una cuarta, la cual solo nos mostrará los datos de la última simulación realizada:

- Cuellos de botella

Realmente la información de la que deseamos tener registro es de las dos primeras y la última gráfica especificadas, puesto que la detección de cuellos de botella, pese a que puedan salir más o menos en los mismos lugares, cada simulación es única y no serán demasiado relevantes los datos sobre las anteriores.

Para acceder a los datos sobre las simulaciones, al finalizar la ejecución desde el servidor, se abre una ventana automáticamente con las estadísticas actualizadas respecto a la última simulación. El gráfico mostrado aparece tras pulsar el botón: "Cuellos de botella".



Por último, pese a que los datos de los ficheros se rellenan automáticamente, estos solo se actualizarán si la simulación se hace con los parámetros preestablecidos, los cuales pueden cambiarse a mano, visitando la condición en la función lanzarGráficas (la cual se encuentra en la clase Servidor):

```
if(numClie == 1200 && (numClie/numGrup == 5 || numClie/numGrup == 8 || numClie/numGrup == 10 ||
  numClie/numGrup == 12)){
```

Si se desea resetear los datos es importante no borrar el fichero dejándolo vacío, sino que al menos dejar los datos rellenos con 0 (esto evitará problemas a la hora de guardar los datos de las simulaciones). Como es evidente, habrá que insertar tantos 0 como valores en el eje x tenga la gráfica.

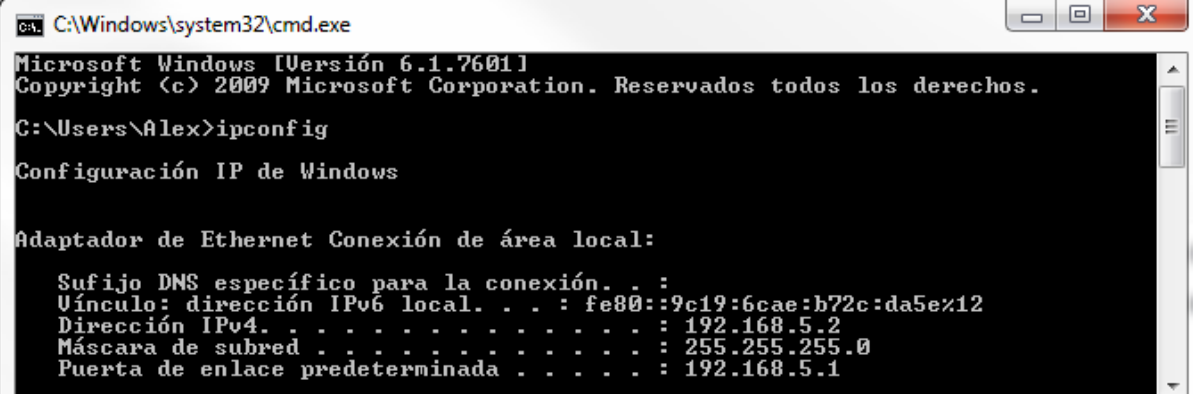
No saltará ningún error, pero no se estará guardando debidamente.

Apéndice

Apertura de puertos e IP pública

Para poder establecer la conexión a través de Internet, más concretamente la petición de conexión de los clientes al servidor, es necesario abrir puertos en nuestro router. Esto se debe a la seguridad que tienen todos nuestros routers, que de forma predeterminada bloquean todas las peticiones de todos los puertos que quieren entrar desde el exterior a para comunicarse con algún dispositivo de nuestra red. La idea es dirigir el puerto (o puertos en nuestro caso) hacia la IP local del servidor. Esto solo es necesario hacerlo en el router al que se conecte el servidor para poder recibir las peticiones de conexión de los clientes. A continuación mostramos cómo se abren los puertos en uno de nuestros routers. Cabe decir que cada fabricante usa menús e interfaces diferentes de modo que tal vez alguna opción esté en otra pestaña o menú.

En primer lugar buscaremos la dirección de la puerta de enlace con el comando "ifconfig" en Linux y con "ipconfig" en Windows. La puerta de enlace debería ser la IP local del router:



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Alex>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Conexión de área local:

    Sufijo DNS específico para la conexión. . . :
    Vínculo dirección IPv6 local. . . : fe80::9c19:6cae:b72c:da5e%12
    Dirección IPv4. . . . . : 192.168.5.2
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . : 192.168.5.1
```

Como podemos ver en la imagen, la puerta de enlace corresponde a la dirección 192.168.5.1 . Esa es la dirección que debería poner yo en el navegador para acceder a la configuración de mi router. En casos normales el router seguramente tenga la dirección 192.168.0.1 o 192.168.1.1 pero en mi caso estoy usando un router secundario y la IP que podemos ver en la imagen anterior es de este router.

Como el router de mi ISP tiene la dirección 192.168.0.1, la pondré en el navegador y accederé a la configuración. En mi caso pide una contraseña que está especificada en la información técnica del router en la parte trasera de este.



Bienvenido

Introduzca la contraseña para acceder a las opciones de configuración.



Una vez en el interior del router debemos buscar la opción de “port forwarding” o “redirección de puertos”.



En mi caso debemos acceder a la pestaña de Internet, en otros routers puede ser distinto:

Docsis 3.1

Algunos ajustes del modo avanzado están en uso

Modo Básico

Visión general

Teléfono

Internet

Wi-Fi

Configuración

Estado & Soporte

Redirección de Puertos

DMZ

Redirección de Puertos

Resetear

La redirección de puertos permite que los equipos remotos se conecten a un dispositivo específico dentro de una LAN privada

Redirección de puertos

Nombre del servicio	Dirección IP	Protocolo	Puerto LAN	Puerto público			
ARC-Project Control	192.168.0.254	Ambos	7685	7685			
ARC-Project	192.168.0.254	Ambos	10740	10740			

NAT Traversal

Es posible que no aparezcan muchas opciones que cabría esperar estuviesen aquí. Esto se debe a que estamos en modo básico. Solo tenemos que cambiar de modo “básico” a “experto” en la opción de arriba a la derecha:

Docsis 3.1

Algunos ajustes del modo avanzado están en uso

Modo Básico

Visión general

Teléfono

Internet

Wi-Fi

Configuración

Redirección de Puertos

DMZ

Redirección de Puertos

Resetear

La redirección de puertos permite que los equipos remotos se conecten a un dispositivo específico dentro de una LAN privada

Redirección de puertos

Nombre del servicio	Dirección IP	Protocolo	Puerto LAN	Puerto público			
ARC-Project Control	192.168.0.254	Ambos	7685	7685			
ARC-Project	192.168.0.254	Ambos	10740	10740			

NAT Traversal

Una vez entramos en modo experto aparecen más opciones, aunque en mi caso la opción necesaria estaba en modo básico:

Docsis 3.1
Algunos ajustes del modo avanzado están en uso
Modo Experto

Visión general
Teléfono
Internet
Wi-Fi
Configuración
Estado & Soporte

Redirección de Puertos

DMZ

Control Parental

DNS & DDNS

UPnP

Filtrado MAC

Redirección de Puertos

La redirección de puertos permite que los equipos remotos se conecten a un dispositivo específico dentro de una LAN privada

Redirección de puertos

Nombre del servicio	Dirección IP	Protocolo	Puerto LAN	Puerto público			
ARC-Project Control	192.168.0.254	Ambos	7685	7685			
ARC-Project	192.168.0.254	Ambos	10740	10740			

NAT Traversal

Resetear

Ya estamos en la ventana de “Redirección de puertos”. Aquí debemos añadir una redirección más, la de nuestro proyecto. Tanto el puerto de entrada como el de salida usaremos el de nuestro proyecto. En nuestro caso usamos 2 así que necesitamos dos redirecciones. Como IP a la que redirigir los mensajes pondremos la IP local del ordenador donde se esté ejecutando el servidor. Podemos encontrar esta IP con el comando que hemos usado anteriormente para encontrar la IP del router. En el apartado “Dirección IPv4” podemos encontrar que en mi caso estamos en la dirección 192.168.5.2, que es la dirección a la que tendré que redirigir el puerto:

En el caso de mi router deja la opción de seleccionar directamente un dispositivo y su IP se pondrá sola.

En mi caso concreto esto no funcionará ya que tendría que redireccionar el puerto a la ip del router secundario al que estoy conectado, pero en casos normales esto debería servir.

Recomendamos que el ordenador que hace de servidor use una IP fija para no tener que modificar esta configuración cada vez que la conexión con el router se corte y este mediante DHCP proporcione otra distinta. Esto también se puede hacer desde la configuración del router.

Obtener la IP pública:

Hecha la configuración en el router es hora de conectar los clientes con el servidor.

Nombre del servicio

ARC-Project

Dispositivo

TL-WR1043ND

LAN IP

192 - 168 - 5 - 2

Protocolo

Ambos

Tipo

☒ Puerto

☐ Intervalo de puertos

Puerto público

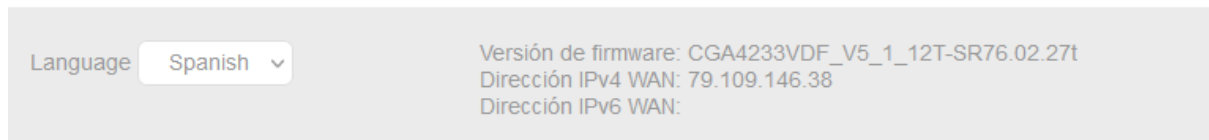
10740

Puerto local

10740

Para ello necesitamos saber la IP pública que tiene asignada el router donde se aloja el servidor.

Para ello podemos usar dos procesos. En primer lugar podemos entrar en la configuración anterior del router. En la página principal debería aparecer en algún sitio la IP pública actual del router. En nuestro caso aparece al final de la página de configuración:



Si no está en la página principal no debería estar muy escondido, pero cada router es distinto y habrá que buscar un poco.

Como siguiente opción, más rápida y sencilla, es buscar una página web que nos diga desde que IP estamos accediendo. Podemos utilizar cualquiera de las que hay. Por ejemplo "miip.es":



Mi IP, ¿Cuál es?

Tu IP es 79.109.146.38

Esta es tu dirección **IP de internet**, también conocida como IP pública.

[¿Qué es una IP?](#)

Incluso para la ocasión y no tener que estar dependiendo de servicios de terceros, podemos ver la IP pública en la propia página web de Alex "www.alexms.es". Aunque se trata de una web básica de pruebas personal podemos ver la IP desde la que estamos accediendo:

Tu dirección IP es: 79.109.146.38

Cabe mencionar que esta IP pública, a menos que se contrate un servicio de IP fija, irá cambiando, por lo que es posible que debamos repetir el proceso. En el caso de nuestro proyecto hemos usado el dominio personal de Alex. Hemos redireccionado el subdominio "arc.alexms.es" creado especialmente para la asignatura que está constantemente redirigido a la IP pública del servidor mediante un servicio DDNS.

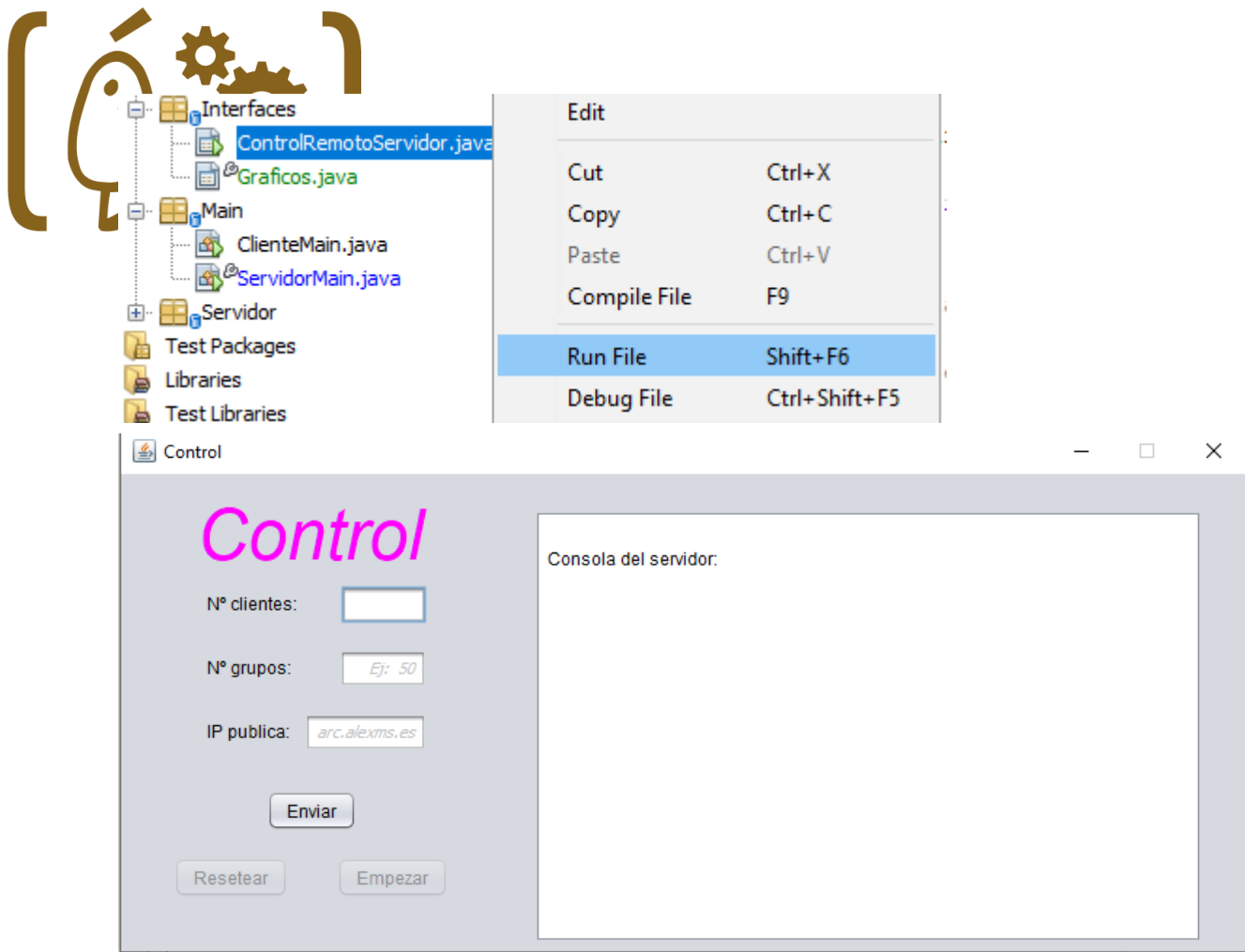
Este servicio actualizará la redirección del subdominio en caso de que cambie la IP pública.

Interfaz

Aunque es perfectamente funcional por consola, se puede hacer uso de una interfaz que actúa como control remoto y mediante la cual nos podemos conectar a un servidor remoto (en el cual se deja previamente en ejecución el proceso ServidorMain con el control GUI activado).

Desde cualquier máquina, podremos ahora mandarle los clientes que queremos que espere y los grupos, y así, si nos encontramos físicamente alejados de este, podemos usar la aplicación igualmente:

```
/**
 * Variable -> "control_GUI"
 * true para control por vista gráfica
 * false para control por consola normal
 */
boolean control_GUI = true;
```



- El botón Enviar sirve para enviar los datos una vez introducidos. Los enviará a la IP pública especificada en el campo.

- Podemos rectificar pulsando el botón Resetear, y le volvemos a enviar los datos. Es importante hacerlo antes de pulsar Empezar.
- El botón Empezar sirve para que el servidor se ponga a la escucha.