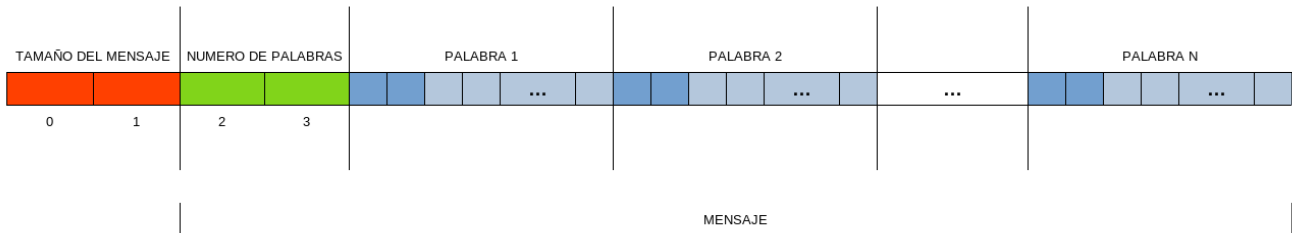
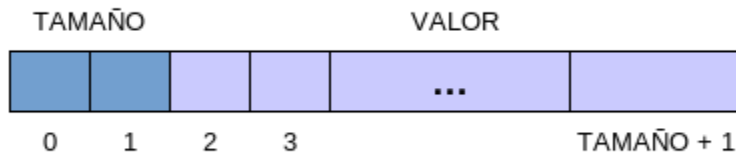


PACKAGE ClientServerImplementation

Estructura de los mensajes enviados y recibidos:



Estructura de la palabra:



Class ConnectionManager

Esta clase gestiona las conexiones, escucha por nuevas conexiones limitando a un numero máximo de clientes, además crea los respectivos gestores para la comunicación con cada cliente.

Constructores
Constructor y Descripción
ConnectionManager(String name, int port, int maxClients) Crea un gestor de conexiones asignándole un nombre, el puerto, y el máximo de clientes que permitirá al mismo tiempo.

Métodos Protegidos	
Retorno	Método y Descripción
boolean	<i>addClient(Socket client)</i> Adiciona un cliente mientras no sobrepase el limite asignado. Retorna true si se adiciono el cliente.
void	<i>deleteClient(Socket client)</i>

	Elimina este cliente.
--	-----------------------

Métodos	
Retorno	Método y Descripción
String	<i>getName()</i> Retorna el nombre del servidor.
int	<i>getPort()</i> Retorna el puerto del servidor.
boolean	<i>getShowConnections()</i> Retorna true si esta habilitado mostrar mensajes de conexión.
boolean	<i>getShowMessages()</i> Retorna true si esta habilitado mostrar mensajes enviados y recibidos.
void	<i>setShowConnections()</i> Habilita/Deshabilita mostrar mensajes de conexión.
void	<i>setShowMessages()</i> Habilita/Deshabilita mostrar mensajes enviados y recibidos.
void	<i>addService(String</i> serviceName, Service service) Registra un servicio y lo asocia a un nombre.
Service	<i>findService(String</i> serviceName) Retorna el servicio asociado al nombre dado, o null si no esta registrado.
int	<i>numberOfClients()</i> Retorna el número de clientes actualmente conectados.
void	<i>run()</i> Se ejecuta en un hilo esperando nuevas conexiones, y crea una comunicación para cada nuevo cliente.

Class CommunicationManager

Constructores
Constructor y Descripción
CommunicationManager(ConnectionManager connectionManager, Socket client)

Crea un gestor para la comunicación con un cliente.

Métodos Privados	
Retorno	Método y Descripción
void	<i>printConnection()</i> Muestra un mensaje en pantalla indicando que un cliente se conecto (si la opción esta habilitada).
void	<i>printDisconnection()</i> Muestra un mensaje en pantalla indicando que un cliente se desconecto (si la opción esta habilitada).
void	<i>printReceived(Vector<String> message)</i> Muestra en pantalla el mensaje recibido del cliente (si la opción esta habilitada).
void	<i>printSent(Vector<String> message)</i> Muestra en pantalla el mensaje enviado al cliente (si la opción esta habilitada).
Vector<String>	<i>recv()</i> Se bloquea hasta recibir un mensaje del cliente, o en caso de que la conexión falle retorna la desconexión.
String	<i>send(Vector<String> message)</i> Envía mensaje al cliente, o en caso de que la conexión falle retorna un mensaje indicando la desconexión.
Vector<String>	<i>solverCommand(Vector<String> command)</i> Retorna un mensaje de respuesta al comando en base a los servicios definidos.

Métodos	
Retorno	Método y Descripción
void	<i>run()</i> Responde a cada mensaje enviado por el cliente, por medio de una lista de servicios definidos.

Class Client

Constructores
Constructor y Descripción
Client(String serverName, String serverIp, int serverPort) Crea un cliente que se conectara al servidor con la ip y el puerto dado. Además se le adicionara un nombre.

Métodos Privados	
Retorno	Método y Descripción
String	<i>firstMessage()</i> Envía un mensaje pidiendo confirmar conexión, y espera la respuesta del servidor. Si el servidor le pide desconexión la conexión finalizo. El servidor le envía su nombre al cliente.

Métodos	
Retorno	Método y Descripción
String	<i>getServerName()</i> Retorna el nombre del servidor.
String	<i>getServerIp()</i> Retorna la ip del servidor.
int	<i>getServerPort()</i> Retorna el puerto del servidor.
int	<i>getLocalPort()</i> Retorna su propio puerto.
void	<i>setShowConnections(boolean showConnections)</i> Modifica la opción de mostrar cuando se conecta o desconecta a un servidor.
void	<i>setShowMessages(boolean showMessages)</i> Modifica la opción de mostrar los mensajes enviados y recibidos.
void	<i>addServiceConfiguration(ServiceConfiguration serviceConfiguration)</i> Adiciona una configuración de servicio.
ServiceConfiguration	<i>findServiceConfiguration(String serviceName)</i> Retorna la configuración de servicio que tenga ese nombre, si no existe retorna null.

void	<i>printConnection()</i> Muestra un mensaje en pantalla indicando que se conecto al servidor (si la opción esta habilitada).
void	<i>printDisconnection()</i> Muestra un mensaje en pantalla indicando que se desconecto del servidor (si la opción esta habilitada).
Vector<String>	<i>recv()</i> Se bloquea hasta recibir un mensaje del servidor, o en caso de que la conexión falle retorna la desconexión.
Vector<String>	<i>send(Vector<String> message)</i> Envía mensaje al servidor, o en caso de que la conexión falle retorna un mensaje indicando la desconexión.
void	<i>printReceived(Vector<String> command)</i> Muestra en pantalla el mensaje recibido del servidor (si la opción esta habilitada).
void	<i>printSent(Vector<String> command)</i> Muestra en pantalla el mensaje enviado al servidor (si la opción esta habilitada).
void	<i>Close()</i> Cierra la conexión con el servidor.
boolean	<i>isClose()</i> Retorna true si la conexión con el servidor esta cerrada.

Class ClientInformation

Constructores
Constructor y Descripción
ClientInformation(String ip, int port) Crea un objeto para guardar la ip y el puerto de un cliente.

Métodos	
Retorno	Método y Descripción
String	<i>getIp()</i> Retorna la ip del cliente.

int	<i>getPort()</i> Retorna el puerto del cliente.
-----	---

Abstract Class Service

Esta clase define como responder a un determinado mensaje, la primer palabra del mensaje permitirá identificar que tipo de servicio es, la segunda deberá contener un resumen o descripción de lo que se envió, y las siguientes palabras serán los parámetros. En este caso las palabras pueden contener cualquier tipo de carácter ASCII, incluido los espacios.

Constructores	
Constructor y Descripción	
Service(String serviceName) Crea un Service.	

Métodos	
Retorno	Método y Descripción
String	<i>getServiceName()</i> Retorna el nombre del servicio.
abstract Vector<String>	<i>solver(ClientInformation clientInformation, Vector<String> command)</i> Recibe una lista de parámetros y retorna el resultado.

Class ConfigurationService

Esta clase se crea para indicar que parte de los mensaje se van a mostrar, en el caso de mensajes muy largos es conveniente mostrar solo un resumen de lo que se envió. Se deberá definir tanto para el mensaje de petición como para el de respuesta.

Constructores	
Constructor y Descripción	
ServiceConfiguration(String serviceName, boolean printDescriptionEnabled, boolean printParametersEnabled) Crea una configuración del servicio indicando el nombre, si imprime la descripción y si imprime los parámetros.	

Metodos	
Retorno	Método y Descripción
String	<i>getServiceName()</i>

	Retorna el nombre del servicio.
String	<i>getVisibleMessage</i> (Vector<String> command) Retorna la parte del mensaje que se podrá mostrar.