**P.O.O.**

**Practica 8**

**RMI**

1.-**ChatBot**

Codificar un programa que actué como un ChatBot con al menos 10 preguntas y respuestas predefinidas. El servidor debe tener un método remoto que reciba la pregunta como parámetro y que dicho método tenga como valor de retorno la respuesta.

Ejemplos de preguntas tipo y respuestas tipo:

En que **ciudad** vives? **D.F**

Cuantos **años** tienes? **20**

En que **escuela** estudias? **ESCOM**

Cuantas **materias** llevas? **6**

2.-**Tablero de avisos electrónico en RMI**

**Almacenar los avisos en el servidor** (en un arreglo o en archivos)

**Aviso**

**Titulo**

**Contenido**

**ejemplos**

|  |  |
| --- | --- |
| **titulo** | **contenido** |
| se lavan alfombras | a domicilio precios modicos |
| se vende perro | cachorro dalmata sabe decir te amo |
| le metemos el migajon a sus bolillos | no sufra mas nosotros lo hacemos por ud. |
| disciplinamos a sus hijos | academia militar con 50 años de experiencia |

**-Codificar** un método remoto que permita obtener un aviso dado su titulo

**-Codificar** un método remoto que permita obtener los títulos de todos los avisos

**-Codificar** un método remoto que permita almacenar un aviso en el servidor

3.-**Marco digital remoto** (ver practica 4)

Modificar los archivos con extensión java en la carpeta rmima para que sea un marco digital que obtenga sus imágenes mediante llamadas a un método remoto.

4.-**Envio y Recepción de Imágenes**

Modificar los archivos con extensión java en la carpeta rmima para agregar un método remoto que permita al cliente enviar una imagen al servidor y el servidor almacene dicha imagen en un vector.

5.-**Paint RMI**

Hacer un programa de dibujo Paint que funcione en red usando RMI y que use callbacks del lado del cliente.

Notas:

1.-Compilar primero los archivos con javac

Y

2.-Compilar el archivo de la clase que implementa la interfaz remota con el compilador rmic

3.-Ejecutar el programa rmiregistry

en LINUX con rmiregistry &

en WINDOWS con start rmiregistry

4.-Ejecutar un servidor y

5.-Un cliente (o varios clientes en los problemas 3 y 4)

java -Djava.security.policy=no.policy NombreClient

donde **no.policy** es un archivo y su contenido es:

grant {

permission java.security.AllPermission;

};