

Mensajes y autómatas

Formato de los mensajes y su funcionamiento y autómatas de cada aplicación

Carlos Cañellas Tovar, Sergio Raúl Rech Lizon

20 de abril de 2020

Formato de mensajes

Mensajes binarios al directorio

Se usan OPCODEs que corresponden a un número entero.

- **OK**: Confirmación del registro de un servidor.

Código: 1

Formato: un byte que contiene el número de código.

- **NOSERVER**: Se envía en la respuesta vacía, en ausencia de servidor.

Código: 2

Formato: un byte que contiene el número de código.

- **REGISTER**: Lo usarán los servidores de chat. Sirve para registrar tal servidor en el directorio.

Código: 3

Formato: un byte que contiene el número de código + el tamaño de un entero para el protocolo + el tamaño de un entero para el puerto.

En total 9 bytes.

Se asume que el tamaño de un número entero en Java estándar son 4 bytes (32 bits).

- **GETSERVER**: Lo usarán los clientes para obtener el servidor que corresponde con su protocolo.

Código: 4

Formato: un byte que contiene el número de código + el tamaño de un entero para almacenar el protocolo.

En total 5 bytes.

- **SERVERRES**: Respuesta a un **GETSERVER**. Contiene los datos del servidor.

Código: 5

Formato: un byte que contiene el número de código + un array de 4 bytes con la IP del servidor + el tamaño de un entero para el puerto.

En total 9 bytes.

Mensajes Campo-valor

- NCNickMessage

```
1 operation:1\n
2 nick:<nick>\n // Siendo <nick> el nick elegido.
3 \n
```

- NCRoomMessage

```
1 operation:2\n
2 \n
```

- NCEnterMessage

```
1 operation:3\n
2 room:<room>\n // Siendo <room> la habitación.
3 \n
```

Se responde con la siguiente operación si se entra con éxito:

- NCInRoomMessage

```
1 operation:4\n
2 \n
```

- NCSendMessage

```
1 operation:5\n
2 message:<message>\n // Los saltos de línea in-message son %n
3 \n
```

- NCExitMessage

```
1 operation:6\n
2 \n
```

- NCInfoMessage

```
1 operation:7\n
2 \n
```

Se responde con lo siguiente:

- NCInfoReplyMessage

```
1 operation:8\n
2 topic:<topic>\n
3 users:<lista de usuarios>\n
4 \n
```

Autómatas

La interacción cliente-servidor se ha dividido en dos autómatas para mayor claridad.

Autómata del cliente con el directorio

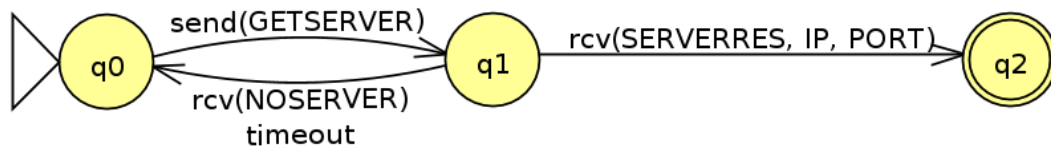


Figura 1: Autómata de cliente a directorio

Autómata del servidor con el directorio

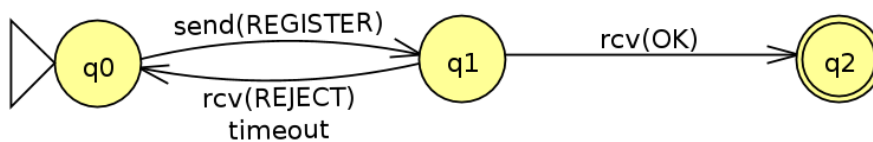


Figura 2: Autómata de servidor a directorio

Autómata del servidor para recibir peticiones del cliente y procesarlas

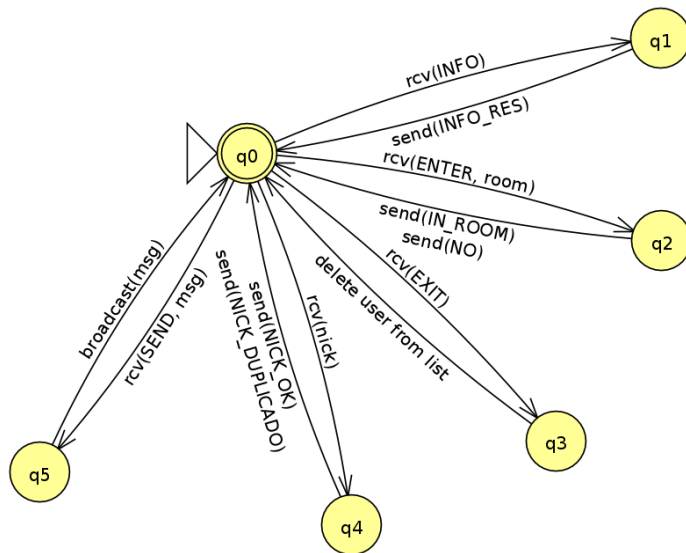


Figura 3: Autómata de servidor a cliente

Autómata del cliente para enviar peticiones al servidor y recibir respuestas

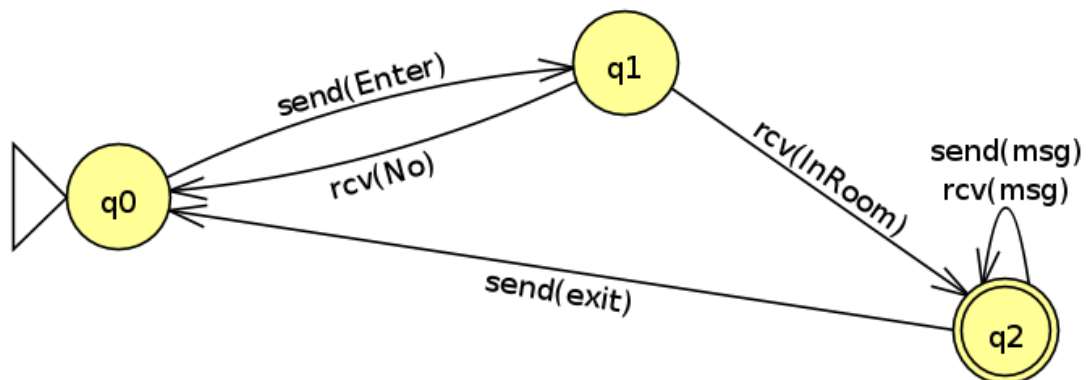


Figura 4: Autómata de cliente a servidor