



Funciones del servidor

2º Desarrollo de Aplicaciones Multiplataforma
Programación de servicios y procesos

25/02/2022

Martínez Díez, Ángel Mori

Contenido

1. El documento que estáis creando con las funciones del servidor.	3
1.1. Reserva de billetes para una compañía aérea	3
1.2. Aplicación de temperatura a tiempo real en la ubicación y hasta tres ciudades almacenadas en la aplicación.....	3
2. Se pretende hacer una aplicación cliente-servidor en la que se almacena en el servidor una base de datos meteorológica.	4

1. El documento que estáis creando con las funciones del servidor.

1.1. Reserva de billetes para una compañía aérea

El servidor debe de ser capaz de atender a una gran cantidad de usuarios (clientes) simultáneamente y de forma rápida. Debe de interactuar continuamente con todos ellos, respondiendo a todas las peticiones que puedan hacer.

Es necesario que actualice al momento los precios de los billetes y, sobre todo, la disponibilidad de estos, ya que no se puede permitir que se gestione el pago de un billete que acaba de ser comprado instantes antes por otro usuario.

La tecnología a emplear sería la de los *sockets stream* pues son más fiables y orientados a conexión. Dada la cantidad de mensajes que se deben intercambiar, es necesario para asegurarse de que lleguen todos correctamente. Los protocolos deben de ser con estado, ya que la secuencia concreta en la que se reciben los mensajes y se responden, es importante.

1.2. Aplicación de temperatura a tiempo real en la ubicación y hasta tres ciudades almacenadas en la aplicación

El servidor debe poder acceder a los datos del tiempo en tiempo real y dar esta información según las ubicaciones recibidas desde la aplicación.

Preferiblemente debería tener una caché valida durante unos minutos para no tener que consultar el tiempo de una misma ubicación cada vez que alguien no solicite. Es decir, una vez que un usuario consulte el tiempo en Guardamar del Segura, si al medio minuto otro usuario la quiere consultar, el servidor solo habrá buscado la información del tiempo la primera vez y la tendrá almacenada en la caché para el segundo cliente. De esta forma ahorra tiempo y recursos y aumenta la velocidad del servicio.

Este ejemplo se puede ver con más claridad en una ciudad grande como Madrid, donde miles de usuarios consultan el tiempo continuamente. El servidor solo tendría que devolverles la información que ya tiene en vez de hacer miles de peticiones al servidor que le provee de los datos del clima.

La tecnología a emplear sería la de los *sockets datagram*, ya que aún siendo menos fiables son más eficientes. No es problema que se pierda algún mensaje. Además, se pueden usar protocolos sin estado.

2. Se pretende hacer una aplicación cliente-servidor en la que se almacena en el servidor una base de datos meteorológica.

En ella está la información de la temperatura y pronóstico del día. EN LA APLICACIÓN, desde un dispositivo se solicita la información completa del lugar donde está ubicado (la aplicación en la solicitud manda la ubicación). Indica cómo harías la estructura de la base de datos del servidor (indica sólo los campos).

Los campos que debería tener la base de datos serían: ubicación, fecha, hora, temperatura, precipitaciones, viento, índice UV...

Especifica las funciones del servidor, escoge la tecnología de comunicaciones adecuada y diseña (de forma conceptual) el protocolo de nivel de aplicación de este servicio.

La función básica del servidor es atender a las solicitudes de los clientes de los clientes respondiendo rápidamente con los datos del tiempo de dicha ubicación. Basta únicamente con responder a la petición del cliente. No es necesario estar intercambiando mensajes entre cliente y servidor, pero sí será completamente necesario que se atienda a múltiples clientes simultáneamente.

Dependiendo de cómo se enfoque, el servidor puede sólo responder a las peticiones de los clientes cuando consulten el tiempo o hacer una especie de *broadcast* a los usuarios con la información actualizada de sus ubicaciones para mantener actualizados sus datos en segundo plano, pero vamos suponer que sólo responde a las solicitudes. Se emplearían *sockets* *datagram* y el protocolo de aplicación HTTP o más concretamente HTTPS para mayor seguridad.