

PRACTICA # 1 TOPICOS DE TELEMATICA

Jhonatan Sebastián Acevedo Castrillón

Andrew Santiago Pérez Martínez

Juan José Escudero Valencia

Universidad EAFIT

Medellín

2021

Problemática:

Decidimos seleccionar la App1 – canales que consiste en lo siguiente:

tendrá dos (2) módulos principales:

Proveedor de anuncios (AdFuente), es decir, es desde donde se generan los mensajes y un Cliente de anuncios (AdCliente), el cual recibe los mensajes enviados a un Canal por un AdFuente.

La aplicación debe gestionar:

1. Canales y Mensajes: Canales a través de los cuales fluyen los Mensajes originados en un AdFuente (puede haber varias fuentes en un mismo Canal) hacia uno o más AdCliente
2. Envíos: son los mensajes que son transmitidos por un AdFuente hacia un Canal y que le debe llegar a los AdCliente. Tenga en cuenta la situación cuando los clientes están o no en línea, que supuesto realizar al respecto.

Los canales son temáticos (deportes, tecnología, noticias, culinaria, bolsa, etc) y deben ser gestionados en el sistema (crear, modificar, borrar, etc un canal).

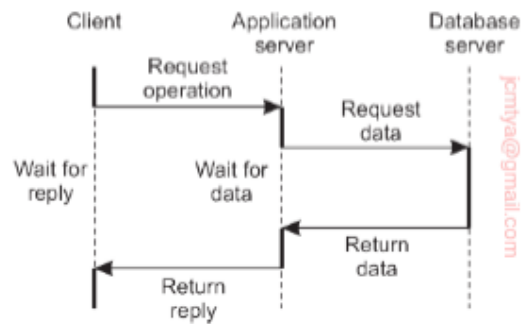
Los anuncios son recibidos por los clientes en modo PUSH y PULL. Por anuncios PUSH se entiende la característica de recibir en un cliente mensajes sin haber sido solicitada explícitamente por el cliente. Por anuncios PULL se entiende cuando el cliente explícitamente recupera mensajes de un canal específico. Debe tener en cuenta que criterio utilizaría para borrar mensajes de una cola.

Análisis:

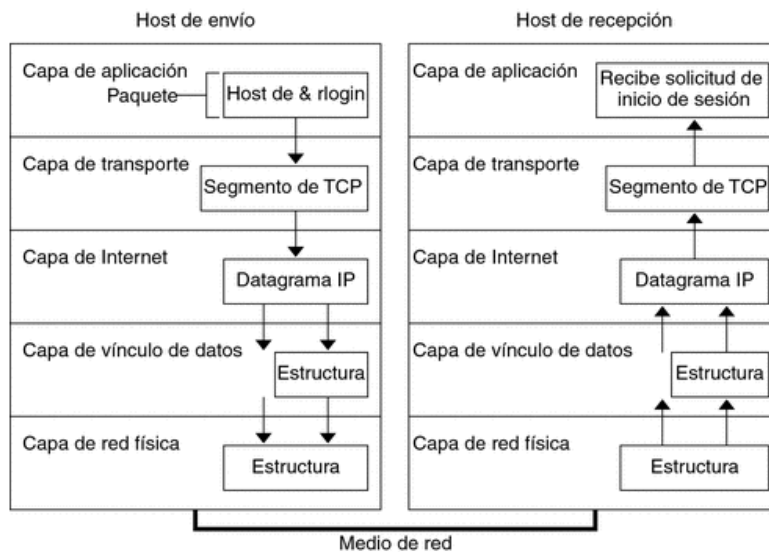
Según la problemática planteada debemos hacer una especie de aplicación donde sea posible la comunicación entre varios clientes como una funcionalidad parecida a lo que es la difusión de mensajes, adicionando que debemos poder recuperar mensajes de ciertos canales que nos sirven para tener la comunicación entre los clientes, para esto se usa el concepto de colas y por eso definimos una estructura sencilla de comunicación con la ayuda de una base de datos y lograr tener una conexión asincrónica con nuestra app.

Diseño:

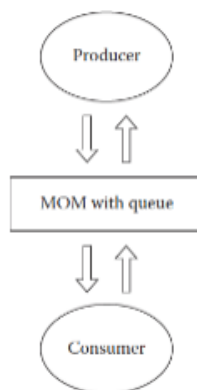
En busca de satisfacer los requerimientos dados en la problemática decidimos optar por realizar una aplicación con una arquitectura tipo N-tier, donde tenemos definidos un cliente, servidor y base de datos



Con esto logramos un desarrollo sencillo de la aplicación, adicionando que para la comunicación de nuestro servidor y cliente usamos el concepto de MOM (Message Oriented Middleware) con una conexión vía TCP por cuestiones de facilidad de implementación en el lenguaje python.



Nuestro objetivo es lograr que nuestros canales repliquen hacia todos los clientes el mensaje, eso si solo a los que son pertenecientes de dicho canal, incluyendo la cola de mensajes para aquellos clientes que quieren enterarse de lo que ha pasado antes cuando no estaban presentes.



Requerimientos:

Nuestro proyecto se realizo con el lenguaje de programación python, para poder realizar la implementación se necesitaron de las siguientes librerías sin contar con la obviedad de tener python en nuestra máquina.

- bson
- werkzeug
- flask
- "Flask-SocketIO<5.0.0"
- eventlet
- pymongo
- flask-login