

Universidad EAFIT
ST0263: Tópicos Especiales en Telemática
Reto01: Diseño e Implementación de un servicio de Anuncios Distribuido, 2012-1

Descripción:

Diseño e Implemente un sistema de anuncios (advertisement) que permita a un conjunto de clientes obtener mensajes de anuncios de productos o servicios. El sistema tiene dos 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.

El sistema debe tener un sistema de gestión que permita:

- 1) Canales y Mensajes: Canales a través de los cuales fluyen los Mensajes originados en un AdFuente (pueden haber varias fuentes en un mismo Canal) hacia uno o más AdCliente
- 2) Envíos: son los mensajes que son enviados 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 utilizará para borrar mensajes de una cola.

Las siguientes son las consideraciones de diseño:

- Definir la arquitectura más adecuada
- Utilizar un protocolo de comunicación tipo Sockets para la comunicación entre las entidades del sistema.
- Debe emular uno de los siguientes modelos/middlewares de sistemas distribuidos:
 - Mensajería (MOM).
 - Protocolo de Aplicación.
- Desde el punto de vista del sistema distribuido y teniendo en cuenta el modelo/middleware a emular seleccionado, considere:
 - Interacción sincrónica/asincrónica.
 - Interacción simétrica/asimétrica.
 - Manejo o no de sesión y estado.
 - Modelo de manejo de fallos.
 - Modelo de seguridad.
 - Niveles de transparencia.
 - Multiusuario
- Haga explícito los criterios o atributos de calidad de la arquitectura a considerar.

- Consideraciones de escalabilidad, extensibilidad y otros criterios de diseño de la arquitectura de la aplicación.

Resumen de Requerimientos:

1. Realice el análisis y diseño (arquitectura y detallado) de la aplicación distribuida Anuncios.
2. Realice la implementación en el lenguaje de programación de su preferencia. No se distraiga con la interfaz gráfica.
3. Utilice la API de programación Sockets utilizando el protocolo TCP o UDP (defina cual y porque)
4. Realice un documento que contenga los elementos de análisis, diseño e implementación de la aplicación distribuida.
5. Realizar la gestión del código fuente en un servidor SubVersión (SVN), servidor que se anunciará a los estudiantes.
6. Realizar el despliegue de la aplicación Anuncios tanto cliente como servidor en un servidor que se anunciará a los estudiantes.

Trabajo Individual o en grupos de 2 estudiantes máximo.

Fecha de entrega:

- Febrero 19 de 2012 por Entrega de trabajos de El y en los servidores de versiones y de producción respectivo.
- El formato del archivo a entregar será Reto01<<apellidos_integrantes>>.zip o .rar
- Debe contener código fuente y documentación, así como un readme.txt (para ayuda de ejecución) y un install.txt (para ayuda de instalación). Y claramente identificados los integrantes.

Criterios de evaluación:

- Informe escrito que contiene el análisis, diseño (global y detallado) además de normas del documento (portada, índice, desarrollo, referencias, etc).
 - En la documentación claramente debe especificar como se instala, que requerimientos de software tiene, como se ejecuta.
 - Desde el punto de vista de análisis y diseño debe apoyarse de los diagramas UML que considere pertinentes.
- Que el reto funcione (corra) por fuera de un editor y que corra bien, con todas las funcionalidades solicitadas.
 - Debe correr en el servidor de despliegue asignado para la materia.
 - Se probará ejecutando más de 2 clientes desde diferentes máquinas o terminales.
 - Que no tenga parámetros “quemados” en código (host, ip o puertos).
- El reto debe utilizar control de versiones en el servidor que se asignará para el curso.
- Se debe sustentar en grupo el reto. Todos los miembros del grupo deben estar plenamente enterados de todos los procesos.