

TP D: Desarrollo de una aplicación distribuida con BSD sockets

Fecha y hora límite de entrega: 24/11 23:59:59

Cácter: grupal (grupos de 2)

Objetivo.....	1
Descripción de la aplicación.....	1
Requerimientos de la aplicación.....	2
API de Sentiment Analysis.....	3
URL Base.....	3
Endpoints.....	3
Syslog.....	4
Recomendaciones generales.....	4
Entregables.....	4

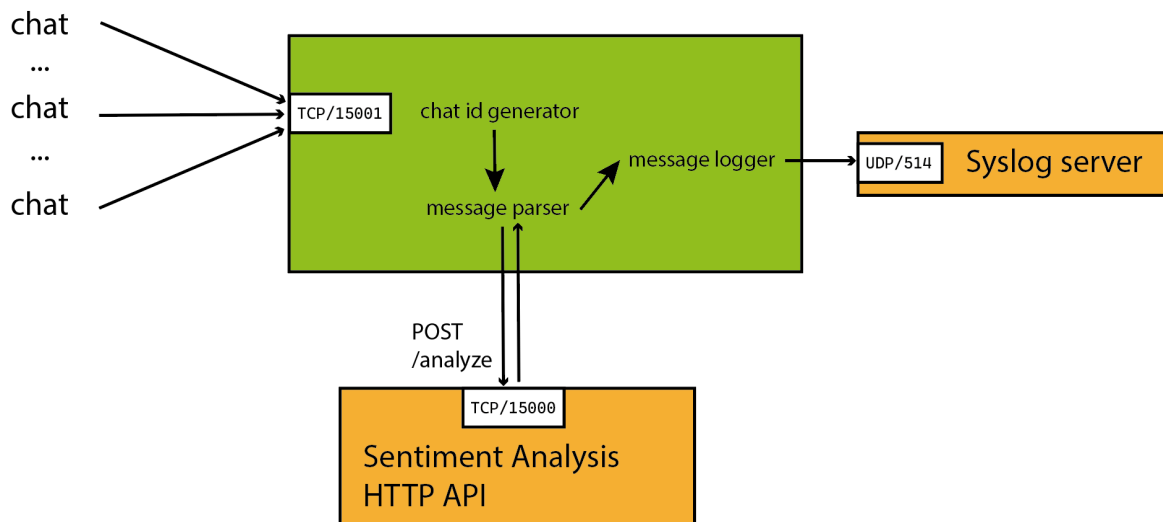
Objetivo

Que los alumnos se familiaricen con la biblioteca BSD Sockets y se interioricen con las particularidades de TCP y UDP. Que logren intercomunicar varios sistemas y sean capaces de adaptarse a protocolos ya existentes.

Descripción de la aplicación

La aplicación recibirá mensajes de chats en un formato TCP especial. Cada conexión TCP recibida pertenece a una instancia de chat entre un cliente y agente de ventas. Cada instancia de chat será asignada un ID. Se analizará cada mensaje, tanto del cliente como del agente, desde el punto de vista de un *sentiment analysis* (*positivo, neutral, negativo*). Para dicho análisis se utilizará una API provista por la cátedra. Cada mensaje intercambiado será analizado y será almacenado usando el protocolo syslog en un syslog server (por ejemplo syslog-ng). En mensaje de syslog además del mensaje, se guardará el ID del chat y el score del *sentiment analysis* y si es cliente o agente.

Podría resumirse el funcionamiento de la aplicación en el siguiente diagrama, donde lo que hay que implementar está resaltado en verde. Los elementos en naranja son estándar o provistos por la cátedra.



El formato del protocolo TCP es el siguiente

<USER><SEP><TIMESTAMP><SEP><MESSAGE_BODY><MSEP>...USER<SEP>TIMESTAMP<SEP>MESSAGE_BODY<MSEP>.

El caracter reservado <SEP> separará campos dentro de un mensaje, por ejemplo separa el usuario del timestamp del body del mensaje

El caracter reservado <MSEP> separará mensajes dentro de la misma conexión TCP.

- <SEP> es el Byte 0x02
- <MSEP> es el Byte 0x04
- <USER> es el nombre de usuario y debe ser ASCII 7 bits (máximo 40 caracteres)
- <TIMESTAMP> es la marca de tiempo en formato YYYY-MM-DD HH:mm:ss
- <MESSAGE_BODY> es el mensaje del usuario y debe ser ASCII 7 bits (máximo 200 caracteres)

Requerimientos de la aplicación

- La aplicación recibirá una conexión TCP por instancia de chat . Los múltiples mensajes tendrán el formato especificado en la sección previa.
- La aplicación deberá soportar múltiples conexiones TCP de instancias de chats en simultáneo.
- La aplicación deberá asignar un identificador único a cada instancia de chat recibida. Por ejemplo `chat-NNNN` donde `NNNN` es número único por cada instancia de chat durante la ejecución del programa.
- La aplicación deberá hacer un *sentiment analysis* de cada mensaje de cada chat.
- La aplicación deberá loguear remotamente cada mensaje analizado usando el protocolo estándar syslog según el formato simplificado que especifica más adelante

API de Sentiment Analysis

La API para determinar el sentiment tiene el siguiente contrato. Utiliza contenido de tipo JSON (<https://www.json.org/json-en.html>)

URL Base

La URL base para la API es : <http://api.udesa.matsunaga.com.ar:15000>

Autenticación

La autenticación se realiza a través de un header HTTP X-API-Token

X-API-Token: <token>

Este API token será provisto a cada grupo por los docentes

Endpoints

Endpoint: POST /analyze

Descripción:: analiza el *sentiment* de un mensaje.

Request Body:

- message(required): mensaje a analizar

Ejemplo: { "message": "I do not like this example at all, it is too boring"}

Responses:

- 200 OK: Devuelve el mensaje junto el **sentiment** (neutral, positive, negative), el **sentiment_value** (valor numérico [-1,1]) y el **subjectivity_value** (valor numérico [-1,1]).
- 400 Bad Request: el request body es inválido o le faltan campos obligatorios..

Ejemplo de uso con curl:

```
curl -XPOST http://api.udesa.matsunaga.com.ar:15000/analyze -H 'Content-type: application/json' -H "X-API-Token: token1" -d '{"message": "I do not like this example at all, it is too boring"}' -v
```

Nota1: si especificamos mal el token dará

```
401 UNAUTHORIZED
{"error": "Invalid authentication token"}
```

Nota2: si no especificamos mal el token dará

```
401 UNAUTHORIZED
{"error": "No authentication token provided"}
```

Se deberá parsear una respuesta HTTP para verificar que tenga el código de estado adecuado. Además se deberá parsear el body en JSON para poder extraer para cualquier mensaje posible el valor de sentiment y sentiment_value/

Se recomienda usar una biblioteca sencilla en C para parsear JSON como

<https://zserge.com/jsmn/>

Tip: clonar el repositorio (git clone <https://github.com/zserge/jsmn.git>) y guiarse con el ejemplo `example/simple.c` que aplica al caso de en que se tiene una estructura JSON conocida

Syslog

El formato del mensaje de syslog es

<https://datatracker.ietf.org/doc/html/rfc5424>

<https://datatracker.ietf.org/doc/html/rfc5426>

Formato simplificado para el TP

<PRIVAL> TIMESTAMP HOSTNAME CONVERSATION_ID ROLE MESSAGE WITH SENTIMENT ANALYSIS

Ejemplos:

<163> Sep 16 02:20:06 localhost chat-0001 AGENT Esto es un mensaje en syslog protocol (sentiment: positive sentiment_value: 0.8)

<163> Sep 16 03:14:16 localhost chat-0002 CLIENT Esto es otro mensaje en syslog protocol (sentiment: negative sentiment_value: -0.8)

Recomendaciones generales

- Siempre hacer error handling con un logging en pantalla o en archivo para ayudar al diagnóstico de los posibles errores.
- Testear las distintas funcionalidades / módulos de la aplicación de manera aislada previa a la integración (ejemplos de módulos: el syslog message logger, el parser de respuestas JSON de la API, etc).
- Usar el código ayuda para facilitar la delimitación de posibles PDUs y para parsear campos dentro una PDU candidata.

Entregables

Se deberá entregar el código junto con sus instrucciones de uso y compilación.

El código será probado con una serie de inputs byte stream que contemplen múltiples casos

- 1 PDU en un socket read (caso optimista)
- menos de 1 PDU en un socket read
- más de 1 PDU en un socket read (no implica cantidades enteras de PDU, por ejemplo podría ser 1 PDU y el comienzo de la próxima)

Se deberá hacer una breve presentación Powerpoint grabada en video (Youtube o similar) que será expuesta el 26/11/2024 y el 27/11/2024. Se debe mostrar la arquitectura diseñada y un diagrama de la misma. A su vez se debe hacer énfasis en la experiencia personal de

programar usando BSD Sockets (desafíos, dificultades, vivencias, expectativas, etc). La presentación deberá durar aproximadamente 8 minutos.

Cabe la posibilidad de que se haga alguna pregunta a cualquiera de los integrantes del grupo al respecto de lo expuesto o relacionado al código o a la estrategia que se utilizó para satisfacer los requerimientos de la aplicación.