



INGENIERÍA EN
COMPUTACIÓN

CÁTEDRA DE SISTEMAS OPERATIVOS II

Departamento de Computación
FCEFyN - UNC

TP1- Estaciones Hidrometeorologicas

Gustavo Gonzalez

Lugar, Fecha

Índice

- 1.Introducción
- 2.Descripción General
- 3.Requisitos Específicos
- 4.Diseño de solución
- 5.Implementación y Resultado
- 6.Conclusiones
- 7.Apendices

Introducción

Propósito del proyecto: El propósito es diseñar, implementar y testear un software tanto para cliente como servidor de un centro de estaciones meteorológicas. El servidor mantiene datos de una estación meteorológica y se los ofrece a los clientes registrados mediante una API.

Ámbito del sistema: En la Provincia de Córdoba existen aproximadamente 400 estaciones hidrometeorológicas automáticas (AWS por sus siglas en inglés), de distintas marcas, modelos y configuraciones, que pertenecen a distintas redes, instituciones y organismos, desparramadas por todo el territorio provincial. Cada AWS consta de una serie de sensores (barómetro, termómetro, sensor de radiación solar, etc.), conectado a un sistema de adquisición de datos, que toma medidas de los sensores (telemetría) un período determinado de tiempo.

Definiciones y abreviaturas:

- AWS: Estación meteorológica
- Socket: abstracción *software* que funciona como punto final de las comunicaciones entre computadoras (puerta).

Descripción del documento: El propósito de este documento es proporcionar a quien lo lea un entendimiento básico de los requisitos y alcance de este proyecto así como los pasos que se tomaron para el diseño e implementación al proyecto propuesto.

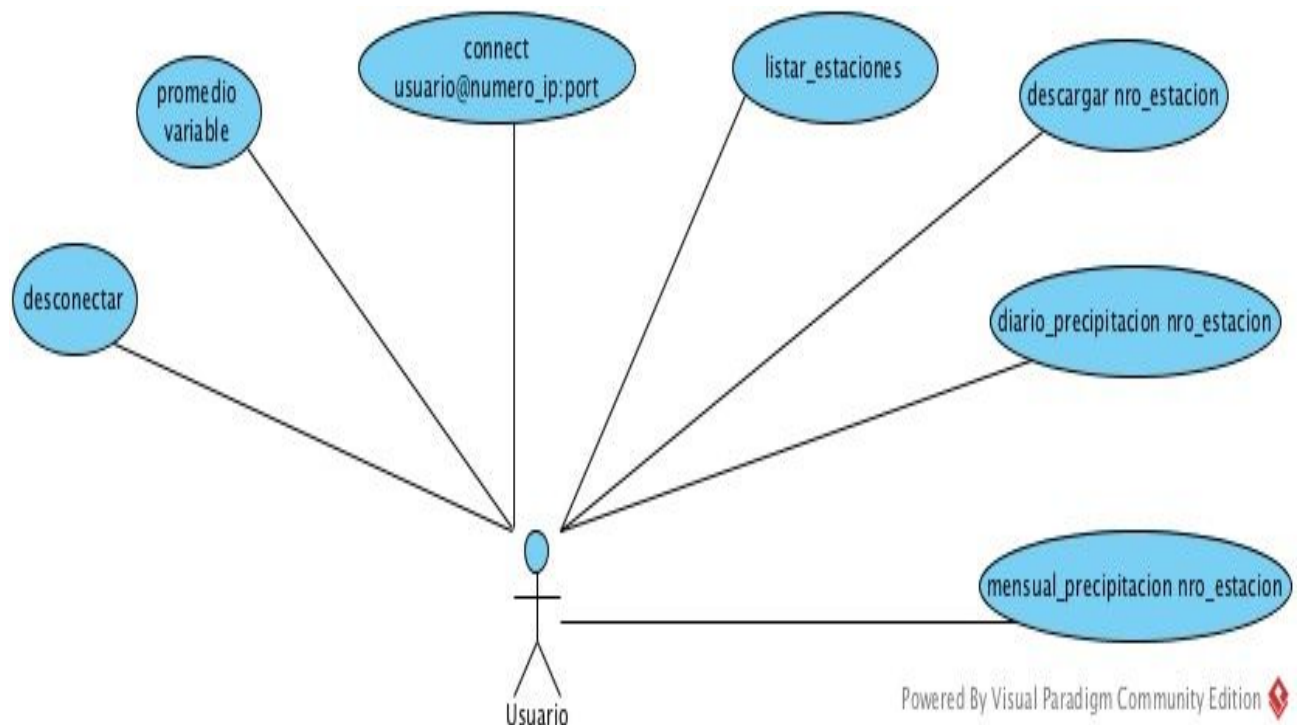
Descripción General

Funciones del Producto:

El servidor debe poder permitir al cliente:

- Autenticarse con nombre de usuario y contraseña
- Conectarse mediante un enlace TCP seguro
- Listar las estaciones disponibles
- Pedir el acumulado mensual de precipitación de una estación específica
- Pedir el acumulado diario de una estación específica
- Descargar un archivo con los datos de la estación solicitada(mediante un enlace UDP)
- Solicitar el promedio de una variable de interés de todas las estaciones
- Desconectarse del servidor

En el siguiente diagrama de caso de uso se ilustra las funcionalidades que debe poder ejecutar un cliente que se conecta al servidor AWS.



Características de los Usuarios: El usuario es cualquier persona o empresa que posea el programa cliente.exe y una computadora con alguna distribución de Linux(Las pruebas se hicieron en Ubuntu) y posea un conocimiento básico para manejarse con la consola ya que no se cuenta con una interfaz gráfica.

Restricciones: La versión actual no cuenta con todas las funcionalidades. La descarga del archivo mediante la función descargar no está disponible y no se puede solicitar el promedio de todas las variables.

Suposiciones y Dependencias:

Para el programa cliente: Únicamente debe conocerse la IP y el puerto en el que está corriendo el servidor, además de la lista de comandos que se puede ejecutar.

Para el programa servidor: Se debe contar con un archivo llamado estaciones_meteorologicas_linux.csv en el cual esten los datos de las estaciones AWS. El puerto 6020 debe estar disponible en la máquina que corre el programa servidor.

Requisitos futuros:

Como primer requisito futuro está implementar la descarga de archivos mediante un enlace UDP, lo cual era uno de los requisitos iniciales pero no se llegó a cumplir dentro de los tiempos establecidos.

El siguiente punto es corregir los bugs encontrados para la seguridad y estabilidad del programa. Dichos bugs están listados en el repositorio de Github del proyecto en el siguiente enlace:

<https://github.com/ggonzalez94/TP1-Soll/issues>

Requisitos Específicos

Requisitos de rendimiento: Ya que no es un sistema en tiempo real ni se deben tomar acciones inmediatas en base a los resultados no se tiene requisitos de rendimiento específicos.

Restricciones de diseño: El enlace entre cliente y servidor debe ser un socket con conexión(TCP), el cual debe proveer una conexión segura entre las dos partes. Para la descarga de archivos se debe implementar mediante un socket sin conexión(UDP)

Diseño de solución

Se optó por una arquitectura cliente-servidor ya que es la que mejor se adapta al problema.

Es claro que hay un servidor el cual tiene los datos de las estaciones meteorológicas en un archivo y debe proveer de manera segura, procesada y ordenada los datos a los múltiples clientes que los pueden solicitar.

El servidor debe proveer soporte para varios clientes y un sistema de autenticación básico para que solo los cliente registrados y autorizados tengan acceso a los datos.

Implementación y Resultados

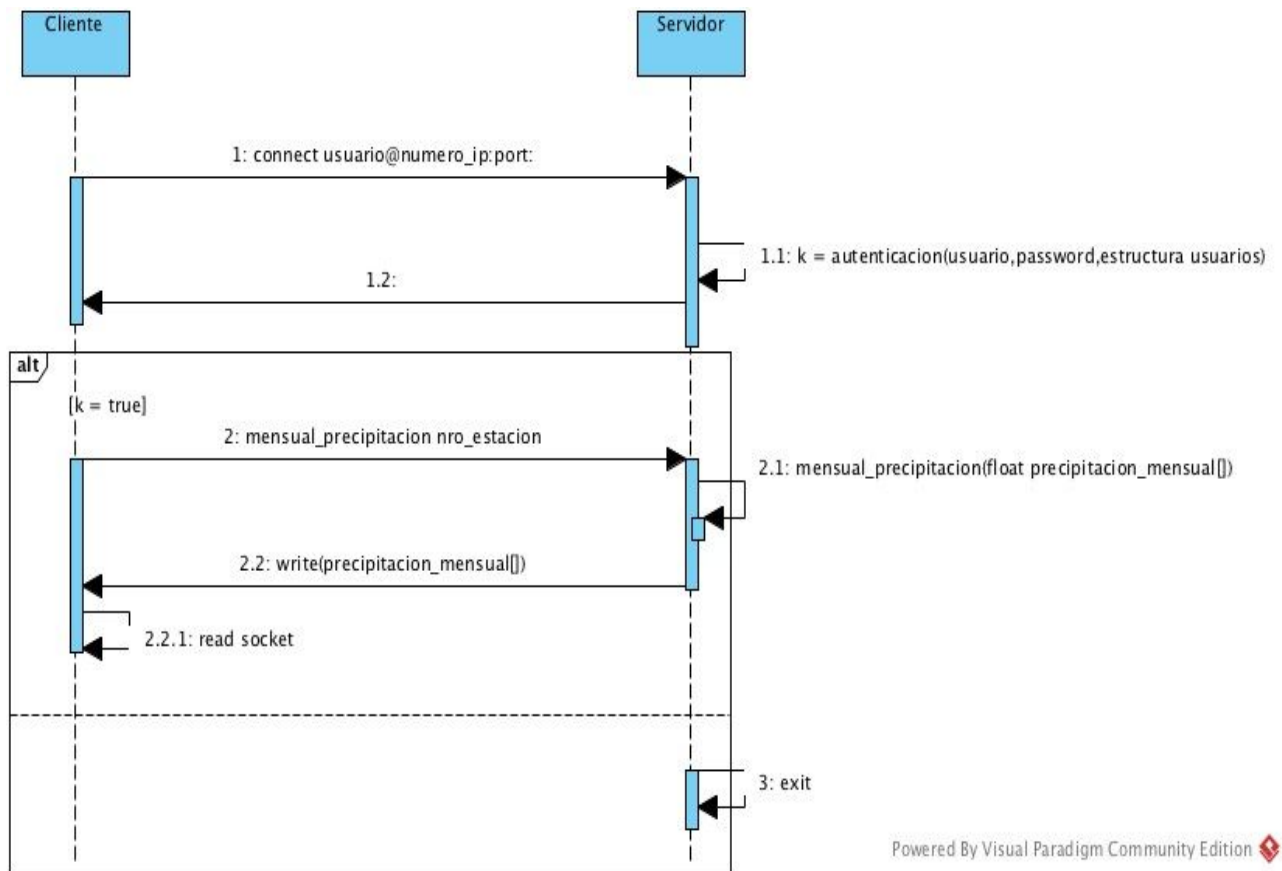
Para la implementación del programa se siguió la siguiente estructura:

El servidor al inicializarse lee el archivo ***datos_meteorologicos_linux.csv*** y almacena los valores en un arreglo de estructuras del tipo Datos(para mayor información sobre las implementaciones revisar la documentación generada por Doxygen).

Luego instancia un socket TCP y lo liga a su dirección IP y puerto 6020 y se queda en espera de clientes que se conecten a él.

Cuando recibe una conexión verifica el usuario, en caso de ser válido este empieza a solicitar información del servidor, caso contrario se termina la conexión(se mata el proceso hijo del server que maneja esta conexión).

En el diagrama de secuencia que se muestra a continuación se ejemplifica un caso de uso típico en el que un cliente se conecta al servidor y luego solicita información (para este caso se solicita el acumulado mensual de la variable precipitación de una de las estaciones).



Powered By Visual Paradigm Community Edition

Conclusiones

Este trabajo presenta un gran desafío ya que para realizarlo se requiere un manejo más profundo del lenguaje C del que estamos acostumbrados además del aprendizaje de nuevos conocimientos como los sockets.

Otro desafío fue el manejo de un proyecto medianamente grande con lo que ello conlleva (estructuración, documentación, etc).

Al realizar el trabajo me di cuenta que aun tengo problemas con el manejo de archivos y principalmente con la estructuración del proyecto (donde definir cada función, cuantos archivos crear y por que). Realizar un buen makefile que ayude a ello es un tema pendiente que aún me queda.

Apendices (Fuentes)

https://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_71/rtref/strpti.htm

<https://support.microsoft.com/en-us/help/38335/how-to-sscanf-example-using-a-comma-,as-delimiter>