



INGENIERÍA EN  
COMPUTACIÓN

# **CÁTEDRA DE SISTEMAS OPERATIVOS II**

Departamento de Computación  
*FCEFN - UNC*

*TP2- Programacion Distribuida*

Gustavo Gonzalez

## **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, testear e implementar una aplicación web que permita a un cliente conectarse al servidor instalado en alguna placa de desarrollo y realizar consultas sobre las estaciones meteorológicas, obtener información sobre el Hardware del server e instalar y remover módulos de manera remota

**Ámbito del sistema:** Nuevamente la Provincia de Córdoba le solicita a la Escuela de Ingeniería en Computación de la FCEfN-UNC, que realice el diseño, desarrollo y prueba del sistema de gestión y acceso a una base de datos hidro-meteorológica. Se solicita que, a partir de lo realizado en el Trabajo Practico No1, utilizando el mismo set de datos, y la misma placa de desarrollo, se desarrolle lo siguiente:

- 1) Elegir un servidor web liviano que se pueda instalar en la placa seleccionada.
- 2) Sobre el servidor web, debe desarrollarse una interfaz web simple, con múltiples páginas, donde cada una debe mostrar, utilizando HTML, CGI y Pearl o C donde se muestre la información listada en el punto

## **Propósito del Proyecto**

**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:**

Es una aplicación web que corre en el servidor(debe ser un Linux) y el cliente es cualquier navegador que se conecte al servidor para realizar las consultas o subir algun modulo a la placa.

El programa provee las siguientes funcionalidades:

- Consulta de la información del sistema
- Descargar Archivo estación
- obtener la precipitación diaria de una estacion
- obtener la precipitación mensual de una estacion
- obtener el promedio de una variable seleccionada
- Listar módulos instalados en el server
- Subir módulo
- Borrar modulo

**Características de los Usuarios:** El usuario es cualquier persona con un dispositivo el cual cuente con acceso a internet y un navegador web y este conectado a la misma red que el server

**Restricciones:** Al no ser público el servidor las consultas solo pueden realizarse desde la misma red.

### **Suposiciones y Dependencias:**

Se debe tener instalado el web server *lighttpd* y configurado adecuadamente (como se muestra en las instrucciones de configuración del repositorio), las páginas estáticas html y los binarios ya compilados de los programas cgi.

Las páginas estáticas se ubican en la carpeta *var/www/html* y los scripts cgi en *var/www/html/cgi-bin*. Además se debe crear una carpeta upload dentro de la carpeta html donde se suben los módulos ingresados por los usuarios antes de ser instalados en el sistema.

El programa debe tener permiso para escribir en dicha carpeta.

El archivo con la información sobre las estaciones meteorológicas se llama *datos\_meteorologicos\_linux.csv* y debe encontrarse en la carpeta

`var/www` y el programa debe tener como minimo permiso para leer dicho archivo.

### **Requisitos futuros:**

Como siguiente requisito se debe corregir y terminar de implementar la descarga del archivo con información sobre la estación deseada, además de mejoras visuales en las páginas, principalmente en la de inicio(carga del módulo) y en la que imprime los resultados de las consultas meteorológicas.

## **Requisitos Específicos**

**Requisitos de rendimiento:** El web server debe ser liviano como para correr en una placa de desarrollo(Raspberry Pi 3) de manera óptima.

**Restricciones de diseño:** Se debe utilizar CGI (Common Gateway Interface) para el contenido dinámico.

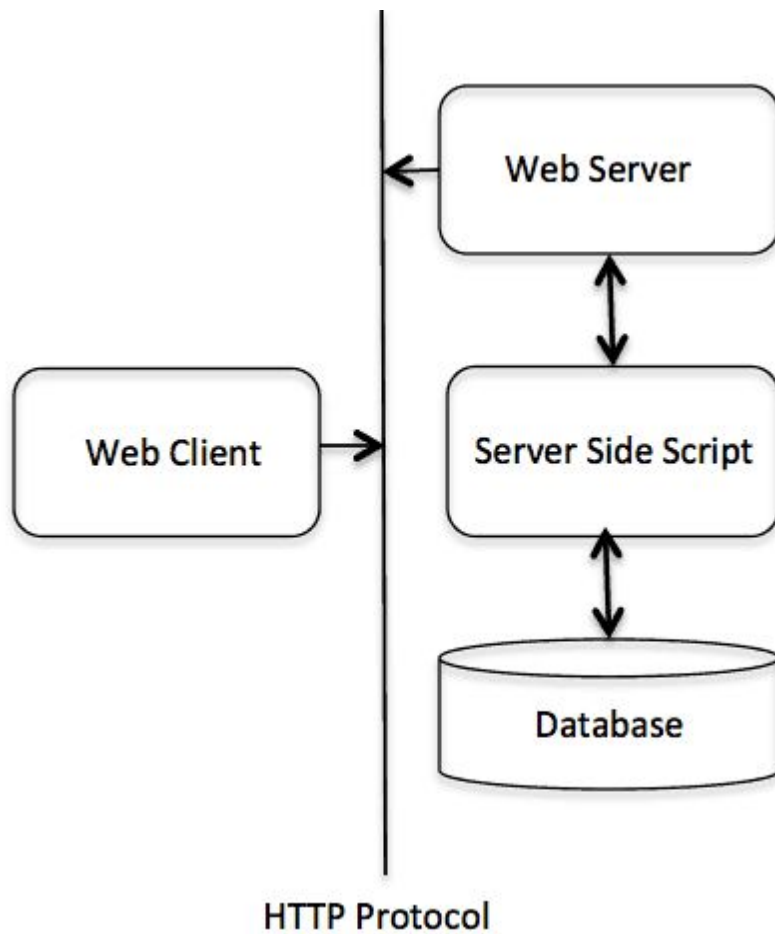
## **Diseño de solución**

El proyecto es relativamente grande por lo que se eligió una arquitectura a seguir. Esta arquitectura es cliente-servidor ya que es la que siguen la mayoría de aplicaciones web.

A esto se le debe añadir la complejidad de la utilización de CGI para ejecutar programas en el servidor.

Cuando llega una solicitud para una pagina desde el cliente, si esta es una pagina estatica se devuelve de manera convencional. La diferencia es cuando la pagina solicitada es un script cgi, en cuyo caso el servidor lo ejecuta y pasa el resultado al cliente.

CGI nos ofrece la gran ventaja de poder ejecutar programas y tener acceso al file system del server de manera segura.



## Implementación y Resultados

Para la implementación se tomó los archivos fuentes de So1(ksmap.c y ksamp.h) los cuales obtienen información del sistema(procesador,memoria,etc) y los del TP1 de esta misma materia adaptandolos para una aplicación web en lugar de utilizar sockets.

Para las páginas estáticas se utilizo la libreria bootstrap para dar una sensación de uniformidad a través de todo el sitio.

Luego se trabajo en conectar las páginas estáticas con las dinámicas mediante la utilización de CGI en el lenguaje C.

Finalmente para la carga y descarga de los módulos se prefirió el lenguaje Perl ya que es más flexible y sencillo para el manejo de archivos, además de haber más recursos en internet en lo concerniente a CGI.

## Conclusiones

Este trabajo particularmente presentó un reto elevado ya que nos introdujo a los conceptos de programación Web y CGI, este último no contando con muchos recursos ni tutoriales en internet.

Además que la estructura del proyecto crece rápidamente al mezclar archivos de anteriores trabajos y gestionarlos pasa a ser un reto.

Además de todo esto debimos enfrentarnos al manejo de módulos en Linux(si bien sencillos los requerimientos en este aspecto) no tenemos demasiada experiencia.

Como aprendizaje queda la experiencia satisfactoria de lograr unir todos los puntos mencionados anteriormente en una sola aplicación web la cual considero tiene la envergadura suficiente para ser un proyecto fuera del ambito academico.

Como reto queda el aprendizaje más profundo de alguno de estos temas especialmente en lo referente a los lenguajes Web.

## Apendices (Fuentes)

<https://www.linux.com/news/which-light-weight-open-source-web-server-right-you>

<https://www.raspberrypi.org/documentation/remote-access/web-server/nginx.md>

<http://www.penguintutor.com/linux/light-webserver>

<http://www.embedded.com/design/connectivity/4442783/10-Things-NOT-to-do-when-embedding-a-web-server>

<https://www.abeautifulsite.net/whipping-file-inputs-into-shape-with-bootstrap-3>

<https://askubuntu.com/questions/600714/creating-a-symlink-from-one-folder-to-another-with-different-names>

<https://www.sitepoint.com/uploading-files-cgi-perl-2/>