

ATI

Lenguajes de Programación

“SML Compiler”

Tarea Programada III

Luigui Madrigal

Andrés Fernández

José María Rojas

07/06/2014

**Tabla de contenidos**

• Descripción del problema……………………………………………….3

• Propósito…………………………………………………….4

• Funcionalidades…………………………………………………….5

• Tecnologías y librerías usadas………………………………………………………….6

• Manual de usuario………………………………………………………..7

• Problemas encontrados………………………………………………………..10

• Conclusión …………………………………………………….......................8

• Referencias…………………………………………………………….....9

**Descripción del problema:**

El lenguaje de programación SML se clasifica como híbrido funcional-imperativo y cuenta con dos ambientes llamados estático y dinámico. En el ambiente estático se almacenan la información de las variables del código compilado y su respectivo tipo y en el ambiente dinámico las variables y su respectivo valor.

Debido a la necesidad de obtener este tipo de información de ambos ambientes de una forma más rápida e interactiva se planteó buscar una solución por medio de una aplicación web desarrollada en cualquier lenguaje de programación (en nuestro caso python). El funcionamiento de dicha aplicación se basa en que el usuario tenga la posibilidad de elegir un archivo .sml desde su computador y lo suba como el archivo que se evaluara. Después de la carga exitosa del mismo se deberán crear dos tablas a partir del código que contiene el archivo, una con la información del ambiente estático y otra con la información del ambiente dinámico.

Es importante mencionar que la solución requerida no tomará en cuenta el manejo de funciones, solo lo hará con expresiones let, if, else, then y asociaciones de variables. Además, los tipos a tomarse en cuenta serán integers, booleanos, listas y tuplas.

**Propósito:**

**Descripción del App**

La App consiste en una página web desarrollada con python utilizando Flask como framework, dicha App permite al usuario mediante un form subir un archivo .sml al servidor, este archivo es procesado mediante nuestro programa en python que retorna el ambiente estático y el ambiente dinámico, posteriormente es mostrado en la página web.

**Requerimientos**

Para desarrollar la aplicación web necesitamos la librería Flask como framework para establecer el servidor.

**Funcionalidades**

**Subir el archivo**

Para esta funcionalidad se utiliza un form en html de tipo multipart/form-data. Este form permite seleccionar el archivo desde el equipo y se guarda en el “servidor” en una carpeta llamada uploads. De ahí se obtiene para ser procesado y luego mostrado.

**Procesar el archivo .sml**

Para procesar el archivo .sml se llama a una función nombrada leersml(nombre del archivo). Esta función se encarga de leer línea por línea el contenido del archivo. La función además se encarga de crear una lista con sablistas que contienen la información del archivo que se va procesar. Con esto se eliminan los espacios en blanco, caracteres no funcionales y listas divididas por expresiones de sml que python pueda entender.

**Mostrar el resultado**

Después de ser procesado se obtienen las dos listas que contienen los ambientes dinámico y estático que son mostrados en dos columnas cada una presenta los datos a través una etiqueta <ul> de html, es decir una “unordered list” que se llena con etiquetas <li> que representan un elemento de la lista.

**Tecnologías y librerías usadas:**

La plataforma sobre la que trabajamos fue Linux por medio de la distribución Ubuntu, por lo que todos los pasos aquí descritos de instalación y manipulación estarán especificados para este sistema operativo.

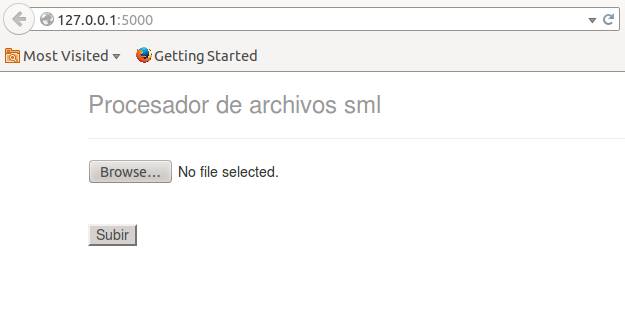
El lenguaje de programación elegido para el desarrollo del sistema fue python, el cual viene por defecto instalado en la mayoría de versiones de Ubuntu.

La creación de la aplicación móvil se realizó con Flask como framework. Se implementó la versión 0.10.1. Se decidió utilizar dicha librería por recomendación del profesor y otros consultados, además de ser la tecnología ya conocida por su implementación en tareas programadas anteriores.

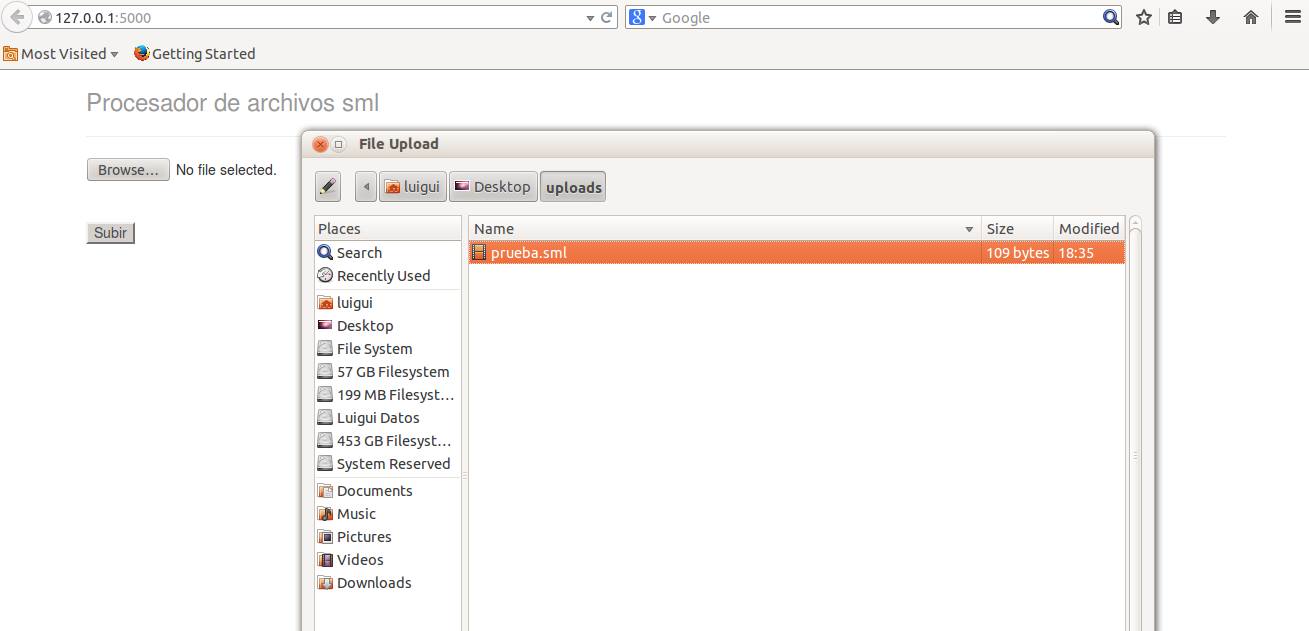
Para la inserción de templates (archivos que se utilizan para determinar la estructura y aspecto visual) utilizamos la plantilla de HTML, ya que es el formato predeterminado que se utiliza en Flask para la página de la aplicación.

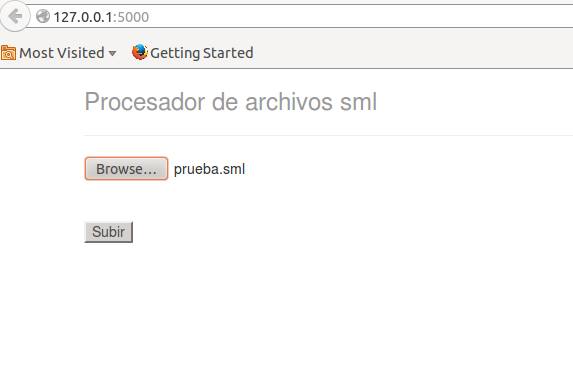
**Manual de usuario:**

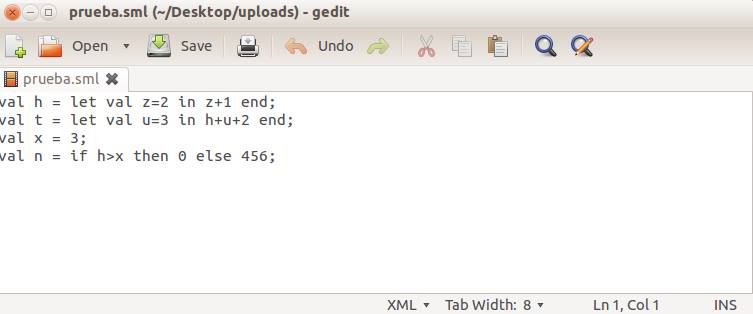
1. **Página principal del sistema:** Aquí se muestran todas las opciones del sistema incluyendo el espacio para la carga del archivo y el botón para subirlo.



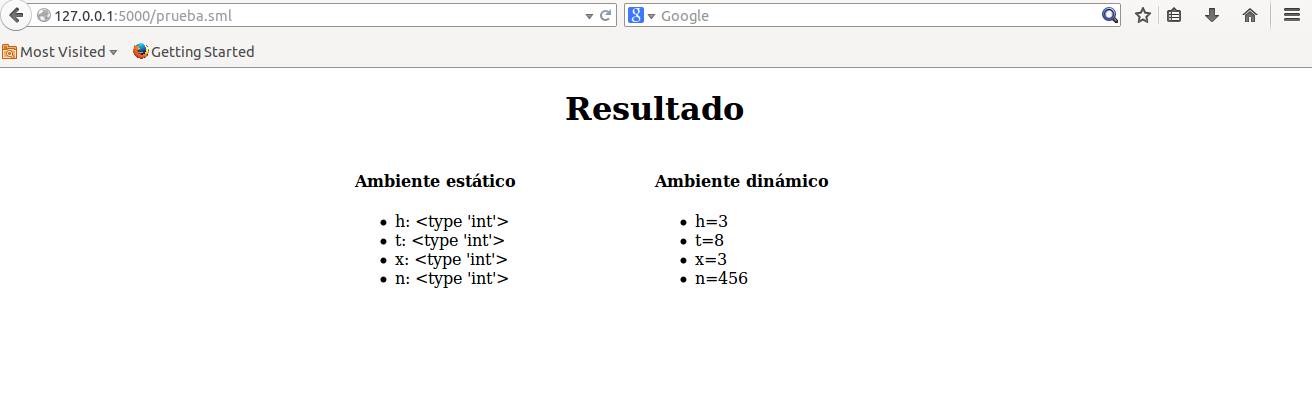
1. **Búsqueda y selección del archivo:** En esta parte de la aplicación se brinda la opción de buscar el archivo y seleccionarlo para su posterior procesamiento.



\* En este punto del manual se mostrara también el contenido del archivo prueba.sml para una mejor comprensión del resultado.



1. **Muestra del resultado:** En este punto la aplicación ya ha terminado de procesar el archivo que se subió anteriormente y se mostrará el resultado con la información del ambiente estático y dinámico.

****

**Problemas encontrados:**

Uno de los problemas encontrados o dificultades que se presentaron fue que perdimos mucho tiempo resolviendo la solución de los Val, creamos un diseño en base a la programación entera de la solución que determina el valor de las variables en SML. Después de varias horas o días de atraso no se lograba obtener los resultados deseados, investigando se dio a conocer una librería de python llamada eval que hacía muchas operaciones que teníamos programadas manualmente y no funcionaban correctamente para todos los casos. Este problema ocasionó un atraso considerable en el desarrollo y completitud de la tarea programada.

**Conclusión:**

En el desarrollo de este proyecto se requirió el análisis y entendimiento completo del proceso de compilación en el entorno sml. Este proceso nos pareció de suma importancia ya que nos invitó a conocer una parte que normalmente omitimos por ser el resultado lo único que se ve y se espera. La estructura de clasificación de sml por medio del ambiente estático y dinámico a nuestro criterio es muy interesante porque simplifica las cosas dando por un lado la clasificación de tipos (ambiente estático) y por otro la información de los valores asociados a cada variable (ambiente dinámico), proceso que tiene un impacto importante en las posteriores funcionalidades. Además, para nosotros el saber cómo piensa o procesa la información algún medio nos facilita saber la forma en que tenemos que transmitir y plantear la misma, para así lograr el mejor resultado.

El conocimiento adquirido de proyectos anteriores fue de mucha utilidad, especialmente en la parte de la aplicación web, pues ya estábamos familiarizados con el lenguaje y el framework, solamente tuvimos que aprender a implementar la utilización del form para subir archivos. Esto nos ahorró más tiempo que si hubiéramos utilizado otro lenguaje con otro framework.

Es importante mencionar que el uso de repositorios por medio de Github fue algo que nos gustó mucho por sus grandes opciones de contribución a trabajos realizados dejando un registro de todos los cambios que se hayan efectuado y las múltiples opciones de compartir proyectos con otros usuarios, además de ser un medio al cual se puede acceder desde cualquier dispositivo.

En general el proyecto fue de mucho provecho, tanto para la comprensión del paradigma funcional y el lenguaje sml como para la realización de la aplicación web.

**Referencias:**

<http://elclubdelautodidacta.es/wp/2011/09/python-capitulo-19-agregando-elementos-a-una-lista/>

<http://docs.python.org.ar/tutorial/3/datastructures.html>

http://flask.pocoo.org/docs/patterns/fileuploads/

<http://www.slideshare.net/danger-leinad/manejo-archivos-enpython-ejm-program>

http://runnable.com/UiPcaBXaxGNYAAAL/how-to-upload-a-file-to-the-server-in-flask-for-python

http://www.maestrosdelweb.com/editorial/guia-python-manejando-archivos-diccionarios-y-funciones/