# Instituto Tecnologíco de Costa Rica

# Realizado por:

# Carlos Badillla Ureña Acsa Herrera Madrigal Monica Solano Morales

# Lenguajes de programación

# TI 3404

Contenido

[Resumen Ejecutivo 2](#__RefHeading__407_1559483527)

[Propósito 2](#__RefHeading__409_1559483527)

[Descripción del app: 2](#__RefHeading__411_1559483527)

[Requerimientos: 2](#__RefHeading__413_1559483527)

[Funcionalidades 2](#__RefHeading__415_1559483527)

[aceptar\_archivo (): Esta función valida si el archivo cargado es del tipo SML 2](#__RefHeading__417_1559483527)

[lista\_imprimir (): Esta función crea una lista con los ambientes dinámicos y estáticos de forma ordenada 3](#__RefHeading__419_1559483527)

[Leer\_Archivo (): Lee el archivo y abre una página con las tablas donde se muestra el ambiente dinámico y estático 3](#__RefHeading__421_1559483527)

[Descripción de diseño de alto nivel 3](#__RefHeading__423_1559483527)

[Descripción detallada del programa 3](#__RefHeading__425_1559483527)

[Decisiones de diseño: 3](#__RefHeading__427_1559483527)

[Lenguajes de Programación usados: 4](#__RefHeading__429_1559483527)

[Tecnologías usadas: 4](#__RefHeading__431_1559483527)

[Librerías usadas: 4](#__RefHeading__433_1559483527)

[Problemas encontrados 4](#__RefHeading__435_1559483527)

[Manual de Usuario 4](#__RefHeading__437_1559483527)

[Conclusión personal 6](#__RefHeading__439_1559483527)

# Resumen Ejecutivo

El proyecto es un sistema que permite obtener la información del ambiente estático y dinámico de un programa en SML.

La idea es usar la aplicación web como una guía para conocer los datos que se almacenan en estos ambientes al subir un determinado archivo .sml.

El sistema responde a la pregunta ¿Cuáles es el ambiente dinámico y estático del archivo SML? .  
Abre la aplicación, sube tu archivo SML y podrás ver en pantalla, una tabla con el ambiente dinámico y estático del archivo que has seleccionado, puedes subir en cualquier momento diferentes archivos de tipo SML

**¿Qué es lo que vamos hacer?**Desarrollar un servicio para obtener diferente información acerca de del ambiente estático y dinámico de un programa en SML*.*

**¿Quiénes lo vamos hacer?**Carlos Badilla Ureña, Acsa Herrera Madrigal, Mónica Solano Morales. Estudiantes de Administración de Tecnologías de Información del Instituto Tecnológico de Costa Rica.

**¿Cómo lo vamos hacer?**Creando una aplicación web por la cual se puede mostrar una tabla lque contiene la información de los diferentes ambientes

**¿Por qué lo vamos hacer?**Para conocer cómo se guardan internamente la información del ambiente estático y dinámico de un programa en SML y así tener una forma más sencilla de aprender cuales son los tipos de datos que se les asigna asi como sus valores en cada etapa de la ejecución del programa

# Propósito

## Descripción del app:

Es una aplicación web para todo tipo de persona que desee conocer los diferentes ambientes de cualquier programa SML, en la cual se pueden subir archivos de cualquier ruta del sistema operativo, al finalizar el proceso de carga del archivo SML, el archivo se guardara en una carpeta temporal y el usuario revisará un mensaje informado el éxito de la operación, y así podrá ver una tabla con el tipo, nombre y valor de la variable.

La información que se mostrara en la tabla está basada en la información cargada del archivo SML seleccionado por el usuario, con el propósito de crear la tabla y conocer el ambiente dinámico y estático del código. Por cada carga del archivo que se realice el usuario se creara la tabla con la información que le corresponde y podrá ver en la pantalla los resultados obtenidos acerca de dicho archivo.

## Requerimientos:

Para poder ejecutar la aplicación correctamente se deben de tener en cuenta los siguientes requerimientos:

* Python 2.7 o posterior.
* Flask 0.10.1 última versión.
* Trabajar en Linux de 32bits.

# Funcionalidades

La aplicación cuenta con diferentes funcionalidades, las modalidades principales las de leer el archivo SML y las de validaciones, de estas se despliegan otras funciones las cuales se explicarán a continuación:

**Cargar archivos:**

### aceptar\_archivo (): Esta función valida si el archivo cargado es del tipo SML

### lista\_imprimir (): Esta función crea una lista con los ambientes dinámicos y estáticos de forma ordenada

### Leer\_Archivo (): Lee el archivo y abre una página con las tablas donde se muestra el ambiente dinámico y estático

Además esta función se encarga de invocar a la función Validar().

**Validaciones:**

**Validar():** es la función que se encarga de identificar el formato de cada una de las variables que forma parte del programa SML .

**Variable(): i**dentifica el tipo de la variable segun el valor que tenga. El tipo en ocasiones es determinado por otras funciones

**Es\_Tupla() :** ayuda a identicar el formato de valor que le ha sido ingresado, enfocado a identificar tuplas.

**ValidarLista():** identifica el tipo de la lista que le ha sido ingresado.

**ValidarTupla()**: identifica el tipo de la tupla que le ha sido ingresado.

### **Otras**:

### **Buscar():** busca el valor de una variable.

### **Hay\_letra()**: valida que haya una leytra en un lista.

### **Operaciones():** hace las operaciones que han sido ingresadas

### **Pegar()**: unifca los números que se han separa cuando el valor fue convertido en lista

### Descripción de diseño de alto nivel

La aplicación está estructurada en dos componentes Front-end y Back-end.

**Front-end**: es la estructura que se encarga de manejar la interacción con el usuario, esta será por medio de la interfaz que se ha creado en HTML que forma parte de la aplicación que ha sido creada en Python. Al usuario le permite el cargar el archivo de tipo SML y la opción de observar los datos por medio de salidas.

A través del front-end de la aplicación el usuario conocerá la información del ambiente dinámico y estático del archivo que ha cargado al haber utilizado la aplicación. Al mismo tiempo podrá subir diferentes archivos SML para conocer su s ambientes

**Back-end**: se encarga del manejo de respuesta del ambiente dinámico y estático del archivo ingresado. Como respaldo y mejor manejo, el archivo es guardado en una carpeta y a través de las funciones que se han declarado en Python que forman parte de la aplicación se pueden realizar la tabla donde se muestra el ambiente dinámico y estático.

# Descripción detallada del programa

## Decisiones de diseño: En esta sección del documento se detallaran las funciones que creemos tienen más reelevancia en el desarrollo del programa, ya que las otras en su mayoria consisten en validaciones de los datos.

aceptar\_archivo ():

**Entrada:** recibe el archivo de tipo SML

**Funcionalidad**: Revisa si el archivo que se cargó, está dentro de los tipos permitidos, SML.

**Salida:** Devuelve un True o un False

lista\_imprimir ():

**Entrada:** recibe una lista, con el ambiente dinámico y estático del archivo cargado

**Funcionalidad**: Crea una lista nueva, con el ambiente dinámico y estático acomodado para ser impreso en línea

**Salida:** Devuelve la lista ordenada de forma correcta

Leer\_Archivo ():

**Entrada:** Nada

**Funcionalidad**: Obtiene el nombre del archivo cargado, revisa si el archivo es correcto, después mueve el archivo a la carpeta temporal, abre el archivo SML y lo lee

**Salida:** Otra página de la aplicación web con una tabla que contiene el ambiente dinámico y estático.

Validar ():

**Entrada:** lista de tokens

**Funcionalidad**: agregar a la lista de variables la variable y el tipo de esta

**Salida:** la lista que contiene el resultado final, osea los ambientes.

Variable():

**Entrada:** la variable y el valor

**Funcionalidad**: evalua segun el valor de la variable el tipo que le corresponde.

**Salida:** los tipos de cada variable

Operaciones():

**Entrada:** el valor de la variable convertido en lista

**Funcionalidad**: realización de cada una de las operaciones que se encuentran en la lista.

**Salida:** el valor real de la variable.

###########################################

## Lenguajes de Programación usados:

Los lenguajes utilizados son:

**Python**: se utiliza para la definición de las funciones para validaciones y las funciones para las cargar el archivo SML, se utilizan sentencias propias de este lenguaje para el manejo lógico de los datos.

**HTML**: se utiliza para la elaboración de la página web, en esta se utiliza las funciones que fueron declaradas en Python, se utiliza para la construcción de la interfaz y para el manejo de los datos cuando se carga el archivo y cuando se debe muestra la tabla del ambiente dinámico y estático

## Tecnologías usadas:

Para el desarrollo del framework se utilizó Flask.

**Flask:** me permite crear la conexión entre Python y HTML, o sea, la utilidad que se le da a esta es para la extracción de los datos de Python a HTML, entre los framework del mercado Flask es el medio con el que puedo crear la aplicación web

## Librerías usadas:

La librería que se utilizo fue:

**Flask:** es una pequeña librería web, que se encargará de generar contenido dinámico a partir de plantillas. Este es un potente microframework web para Python, fácil de usar y con bastantes módulos para implementar numerosas funcionalidades webs.

**from** flask **import** Flask, render\_template,

request, redirect, url\_for, abort, session

# Problemas encontrados

1. Al no tener una base se debió invertir mucho tiempo a la hora de validar los posibles casos para poder realizar el ambiente dinámico y estático.
2. Ver la forma de hacer que los valores, nombres y tipos se impriman de forma correcta

# Manual de Usuario

**1.** Instale el programa Python 2.7 en su computadora. (En caso de tenerlo omita esta instrucción).

**2.** Si desea puede instalar el IDLE de Python2.7 para generar el código desde allí, esto lo podrá encontrar en el centro de software de Ubuntu, ingrese y siga los pasos de instalación.



**3.** Instale el framework Flask, para realizar la instalación debe de ingresar a la terminal e ingresar las siguientes líneas de comandos:

$ sudo apt-get install python-virtualenv

$ mkdir myproject

$ cd myproject

$ virtualenv venv

New python executable in venv/bin/python

Installing distribute............done.

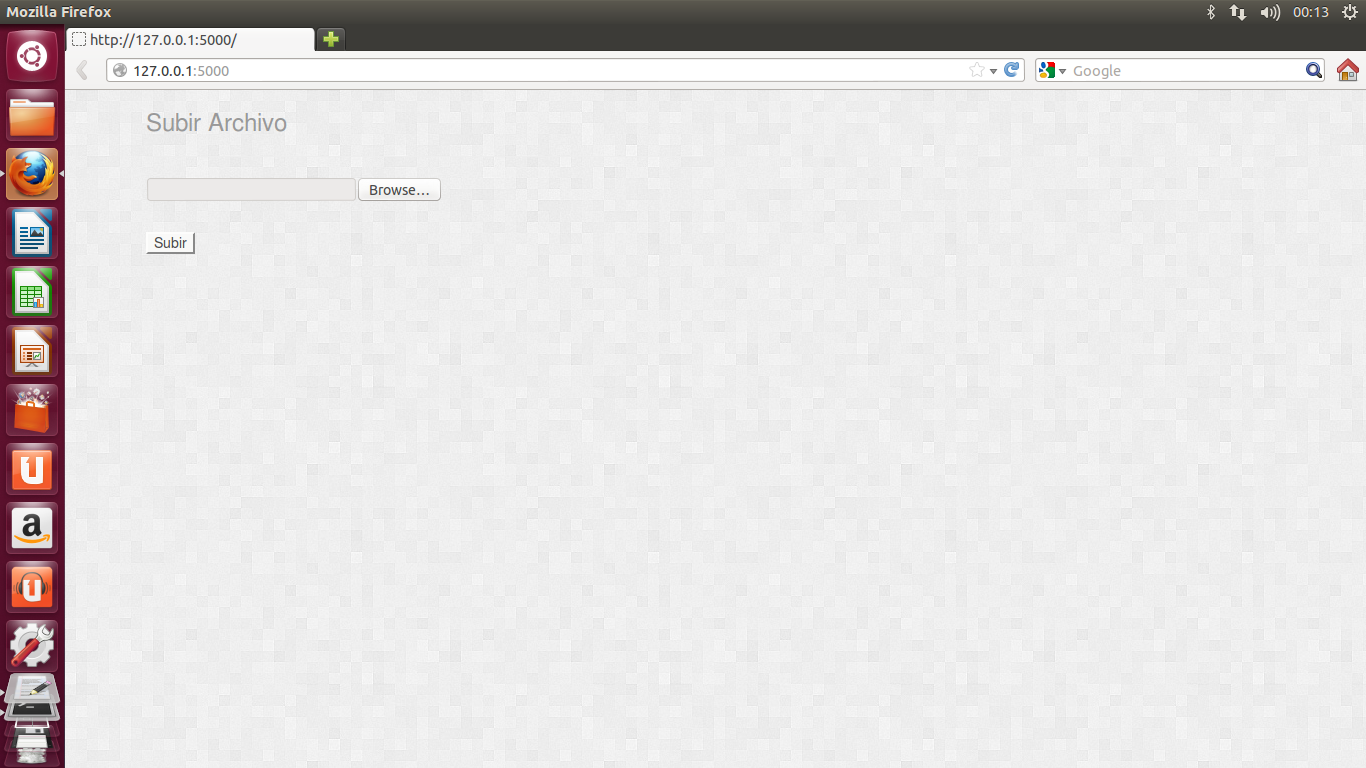
**4.** Ingrese a la terminal y luego ingrese a la ubicación donde se encuentran los archivos del programa. Ejecute el archivo con el nombre progra.py.



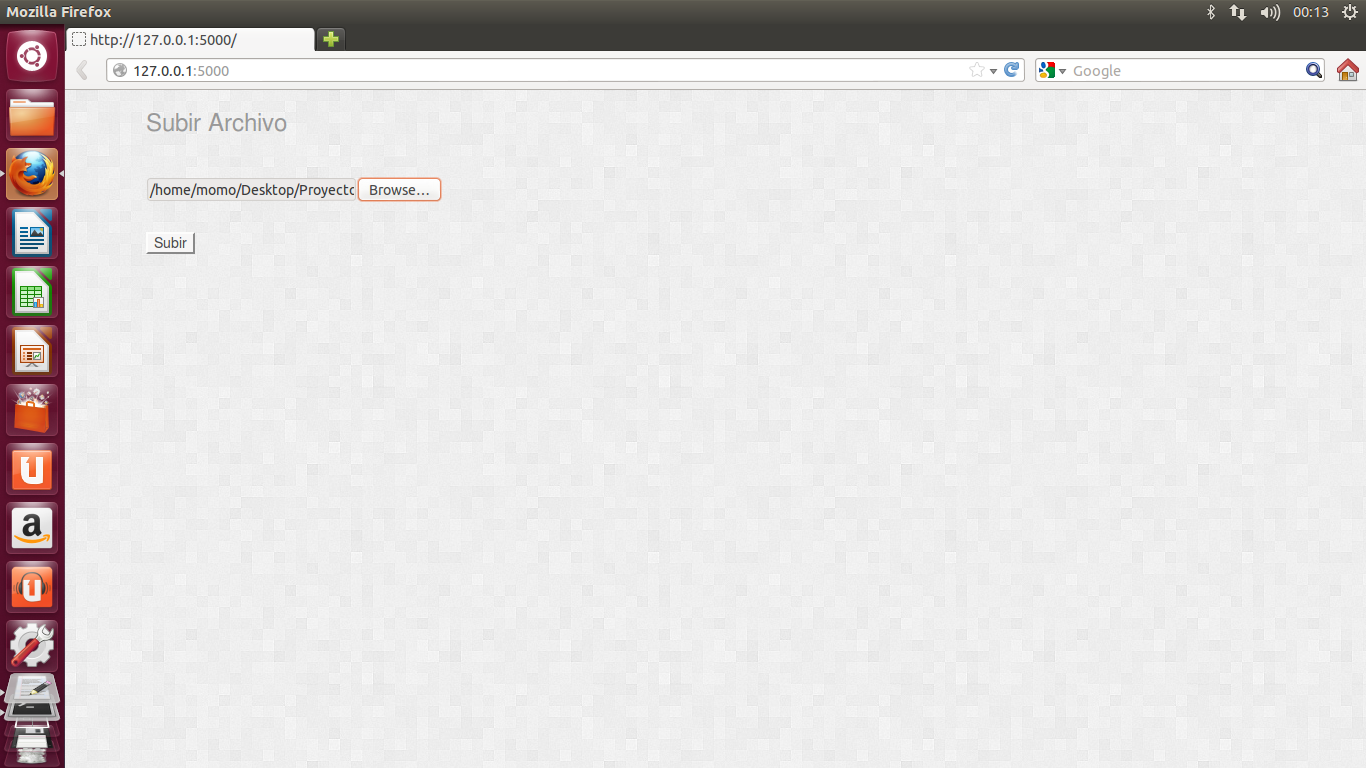
**5.** Posicione el mouse sobre la dirección http, luego presione click derecho y open link.



**7.** Se inicia la aplicación, en pantalla se mostrará la pantalla para cargar el archivo:



**8.** Se selecciona el archivo que se desea cargar y se da click a subir



# 

# Conclusión personal

Con la realización de este proyecto se pudo concluir que es de suma importancia, antes de empezar a programar, tener conocimiento sobre el campo en el que se está trabajando, investigar sobre diferentes funciones que son útiles para la realización del trabajo, y comprender su funcionamiento. Analizar y entender cuáles son las necesidades, qué es lo que se busca y el resultado al que se quiere llegar.

Para finalizar, es relevante mencionar que para facilitar la confección del programa, un método que se puede poner en práctica es la realización de un plan de trabajo que permita ir modelando un esquema o un diagrama de las ideas para la elaboración del sistema, y así poder ir implementado los algoritmos que se establecieron y afinar los detalles para una buena producción del programa deseado.