



# SML-Analyzer

*Documentación técnica y de usuario final*

Fabián Fernández  
Esteban Leandro  
Jimmy Granados

Lic. Administración de tecnologías de información  
Curso: Lenguajes de programación  
Profesor: Andrei Fuentes  
INSTITUTO TECNOLÓGICO DE COSTA RICA

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Documentación técnica . . . . .	1
1.2. Manual técnico . . . . .	1
1.3. Manual de usuario . . . . .	1
<b>2. Documentación técnica</b>	<b>2</b>
2.1. Propósito de la aplicación (descripción y requerimientos) . . . . .	2
2.2. Funcionalidades . . . . .	2
2.3. Diseño de alto nivel . . . . .	3
2.4. Descripción detallada . . . . .	4
2.4.1. Decisiones de diseño . . . . .	4
2.4.2. Lenguajes y tecnologías de programación . . . . .	6
2.4.3. Bibliotecas utilizadas . . . . .	6
<b>3. Manual técnico</b>	<b>8</b>
3.1. Instalación . . . . .	8
3.2. Problemas encontrados . . . . .	9
<b>4. Manual de usuario</b>	<b>10</b>
4.1. Ejecutando SML-analyzer . . . . .	10

# 1

## Introducción

EL PRESENTE DOCUMENTO tiene como finalidad ofrecer tanto al usuario final como al usuario técnico la información necesaria para comprender de forma rápida y ágil el funcionamiento de la aplicación SML-analyzer, correspondiente a la tercer tarea programada del curso TI-3104 Lenguajes de programación. El documento está compuesto por las siguientes partes:

### 1.1. Documentación técnica

Esta sección está dirigida al usuario que desea utilizar la aplicación como servicio de software, ofrece instrucciones de instalación, requerimientos técnicos, requerimientos de software, etc.

### 1.2. Manual técnico

El manual técnico ofrece un conjunto de instrucciones dirigido a desarrolladores de software para que puedan instalar y ejecutar SML-analyzer como servicio web.

### 1.3. Manual de usuario

En el manual de usuario se expone la forma de ejecutar la aplicación, está dirigido al usuario final”quien utiliza la aplicación sin introducirse en aspectos técnicos o de instalación.

# 2

## Documentación técnica

### 2.1. Propósito de la aplicación (descripción y requerimientos)

**S**ML-ANALYZER es una aplicación web que permite el análisis de tipos de variables y datos tanto en el ambiente estático como en el ambiente dinámico; orientado a programas escritos en SML.

Para ser ejecutada como un servicio de software se requiere de un equipo que cuente con el lenguaje de programación Ruby y un conjunto de bibliotecas propias del lenguaje que se expondrán más adelante.

Para el usuario final es necesario simplemente un explorador web, en el cual se debe introducir la dirección IP y el puerto del servidor (expuestos posteriormente); esto le permitirá acceder a la aplicación de forma simple.

### 2.2. Funcionalidades

Dentro de las funcionalidades de la aplicación se encuentran:

- **Diseño web responsive**: esto quiere decir que los elementos que componen el front-end (la vista a la cual el usuario final podrá acceder) se dispondrán de manera eficiente de acuerdo al equipo/dispositivo en el cual se accede la aplicación. De esta forma si se desea abrir SML-analyzer desde un dispositivo móvil (teléfono inteligente, tableta, etc.) se podrá tener a disposición todos los elementos de forma accesible y cómoda.
- **Carga de archivos con formato ".sml"**: mediante esta funcionalidad el usuario final podrá cargar al servidor de la aplicación archivos de dicho formato para su análisis respectivo.

- Análisis de archivos SML: la aplicación examinará por completo el archivo en formato sml que el usuario final cargue para obtener de forma automática los datos correspondientes a los ambientes dinámico y estático (información sobre los tipos de variables y sus valores).
- Vista de la información correspondiente a los ambientes: se ofrece al usuario una vista en forma de tabla con la información correspondiente a los valores de las variables y a sus tipos (entero, lista, tupla, booleano).

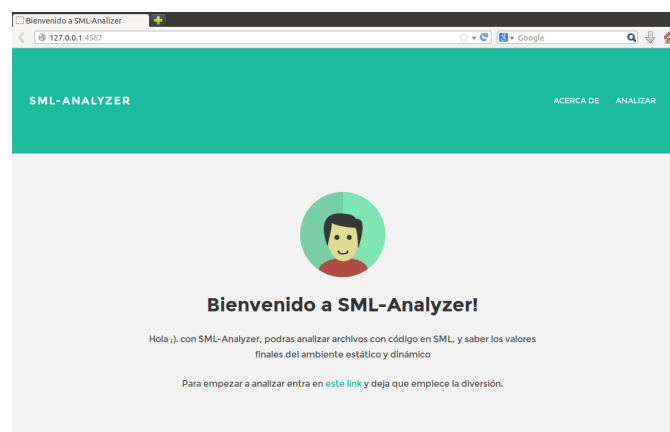
El funcionamiento de la aplicación se basa en lo expuesto en la especificación de la tarea ofrecida por el profesor, que plantea lo siguiente:

- El usuario abre la aplicación, y especifica (mediante un form) cuál es el archivo de sml que quiere subir (el archivo estará en la computadora del usuario)
- El servidor recibe el archivo y lo procesa (obteniendo la información del ambiente estático y dinámico)
- El servidor muestra al usuario una página con los resultados del análisis (puede ser en formato tabla).

## 2.3. Diseño de alto nivel

El diseño de alto nivel explica de forma explícita el comportamiento general del proyecto, ésto sin inmiscuirse en aspectos específicos del proyecto como lo es: código de la página web, código propio de la aplicación, herramientas de diseño, etc.

Se ha optado por utilizar una plantilla web utilizando la tecnología bootstrap para adecuar el diseño de la aplicación a una vista agradable y simple, similar a la correspondiente en la figura 2.1.



**Figura 2.1:** Vista principal de SML-analyzer

De igual forma se mantiene este formato para las páginas de subir archivos, acerca de y resultados del análisis.

## **2.4. Descripción detallada**

### **2.4.1. Decisiones de diseño**

Luego de realizar una proceso de evaluación con distintas tecnologías de diseño web se decidió utilizar "sinatra", una biblioteca del lenguaje de programación ruby, que ofrece un conjunto de instrucciones extremadamente simple y eficiente para producir páginas web.

Luego de tener claro el lenguaje y la biblioteca a utilizar para la aplicación se decidió separar la aplicación en distintas capas, correspondientes por carpetas y archivos autónomos donde se conserva por separado el bosquejo del front-end (la vista de la aplicación que interactúa con el usuario final), el código en sinatra donde se estructura la jerarquía de las páginas y el código ruby que se encarga de interactuar con los datos y procesarlos, para luego regresarlos al front-end de forma estructurada. De esta forma se obtiene una estructura (representada en forma de árbol) de la carpeta donde se contiene todos los archivos necesarios para la aplicación de forma similar a la vista en la figura 2.2, así se puede diferenciar claramente la división entre los archivos correspondientes al entorno web, la aplicación back-end, las imágenes a mostrar en la vista de usuario, etc.



Figura 2.2: Arbol del directorio de SML-analyzer

Para finalizar, se optó por emplear una tabla para mostrar los datos obtenidos del procesamiento del archivo sml cargado por el usuario previamente; esto con el fin de proporcionar al usuario una vista sobria y eficiente de mostrar la información obtenida por el análisis del archivo sml.

### 2.4.2. Lenguajes y tecnologías de programación

Para el desarrollo de esta aplicación, se utilizó una serie de lenguajes y tecnologías de programación, listados a continuación:

- Ruby: fue el lenguaje de programación principal, elegido principalmente por su simplicidad y su amplia documentación, disponible tanto documentación creada de por los desarrolladores oficiales del lenguaje como documentación creada por una amplia comunidad de programadores a nivel mundial.

Para información detallada del lenguaje ruby puede acceder al siguiente enlace <https://www.ruby-lang.org/en/>.

- HAML: es un lenguaje especializado para preparar páginas HTML de forma simple y con características suficientemente importantes como para ser elegido como lenguaje principal a interactuar con el ambiente web.

La siguiente corresponde a la página principal del proyecto HAML, donde se dispone de una amplia documentación y un conjunto de tutoriales realmente simples y claros: <http://haml.info/>

- Bootstrap-CSS: bootstrap es un "framework" de trabajo que permite de forma simple crear sitios web responsive", que facilita enormemente el trabajo del programador y ofrece al usuario final un diseño excepcionalmente elegante. Se escogió por proveer una amplia cantidad de plantillas con código fácilmente entendible y personalizable Documentación oficial de la tecnología bootstrap puede ser encontrada a través del enlace <http://getbootstrap.com/>.

- Javascript: es un lenguaje que se utiliza unido al entorno de usuario final para ejecutar comandos de forma similar a lenguajes de programación tradicionales, lo que permite que el código de la página web pueda ser más dinámico en cuanto a ejecución de comandos especializados se refiere (llamadas a funciones, procesamiento de tiras de caracteres, etc.) Aunque no existe una página oficial correspondiente a javascript, se puede encontrar documentación bastante extensa en wikipedia, en el enlace <http://es.wikipedia.org/wiki/JavaScript>.

### 2.4.3. Bibliotecas utilizadas

Ruby ofrece una amplia selección de bibliotecas para realizar una infinidad de operaciones, parte de eso fue tomado en cuenta para elegir ruby como lenguaje de programación principal.

Las bibliotecas utilizadas se listan a continuación:

- rubygems: esta biblioteca se comporta como un administrador de bibliotecas de ruby, esto quiere decir que permite a los programadores instalar y administrar bibliotecas de ruby de forma simple sin recurrir a comandos externos o a instaladores de bibliotecas.

La documentación oficial de rubygems se puede encontrar en el enlace <https://rubygems.org/>



- sinatra: es un "framework" para desarrollar aplicaciones web perteneciente al lenguaje ruby que ofrece una estructura simple de crear aplicaciones web y transmitir datos desde y hacia la aplicación web.  
La información oficial de la tecnología sinatra puede ser accedida por medio del siguiente enlace <http://www.sinatrarb.com/>
- haml: corresponde a la biblioteca que permite conectar con la tecnología haml explicada en el punto 2.4.2 y poder acceder a sus funcionalidades.

# 3

## Manual técnico

**E**N ESTE APARTADO se ofrece al programador un conjunto de instrucciones que le permitirán instalar y ejecutar la aplicación SML-analyzer. Se parte del hecho que el programador no tiene instalada ninguna de las herramientas listadas a continuación, en caso contrario el programador debe omitir los pasos necesarios para instalar las herramientas que ya tenga instaladas.

Las instrucciones listadas a continuación deben ser ejecutadas en sistemas operativos “UNIX-like”, esto quiere decir que sean derivados del sistema operativo UNIX (Mac OS, Linux, FreeBSD, Solaris, etc.); sin embargo, los comandos utilizados han sido probados en la distribución de linux Ubuntu versión 14, por lo que algunos de los comandos pueden no funcionar como es esperado y se requiera investigar un equivalente que corresponda al sistema operativo que utilice.

### 3.1. Instalación

1. Instalar Ruby  
*sudo apt-get install ruby*
2. Instalar RVM (ruby version manager) que permite mejorar la compatibilidad de bibliotecas entre distintas versiones de ruby.  
*\curl -sSLhttps://get.rvm.io||bash-s stable*
3. Instalar sinatra  
*gem install sinatra*
4. Instalar haml  
*gem install haml*
5. Instalar git  
*sudo apt-get install git*

6. Obtener el código fuente de SML-analyzer (esto debe ser ejecutado en la carpeta destino)  
*git clone https://github.com/jimmy-ga/SML-analyzer.git*
7. Dentro de la carpeta /sinatra iniciar el servidor de la aplicación  
*ruby server.rb*
8. Obtenga la dirección IP del servidor ejecute  
*hostname -I*
9. En el explorador web introduzca (en forma de ejemplo se utiliza la IP local) la dirección IP del servidor, seguido del puerto 4567 (seleccionado por defecto)  
*127.0.0.1:4567*

De esta forma, tiene a su disposición el código de SML-analyzer y la forma de ejecutar el servicio en el servidor. Si desea personalizar el código del SML-analyzer debería analizar a profundidad los distintos archivos dentro de la carpeta de la aplicación.

### 3.2. Problemas encontrados

Durante el desarrollo de SML-analyzer se encontró varios obstáculos, entre ellos:

- Procesamiento del archivo sml: la parte más compleja del desarrollo fue el crear un "compilador" del lenguaje sml, funcionalidad que requirió de mucha investigación y constante prueba-error hasta llegar a la versión final de las funciones que nos permitieran procesar el lenguaje sml sin problema alguno.
- Integración de ruby al entorno web: aunque el lenguaje ruby ofrece facilidades para esto, hubo que pasar por una curva de aprendizaje tanto del lenguaje como de sus bibliotecas y complementos para poder programar de manera correcta la interacción de las distintas capas de la aplicación.
- Cargar el archivo sml al servidor: al inicio la carga del archivo no se completaba con éxito, pues mostraba el archivo como si estuviera cargado pero no llegaba a la carpeta destino.

# 4

## Manual de usuario

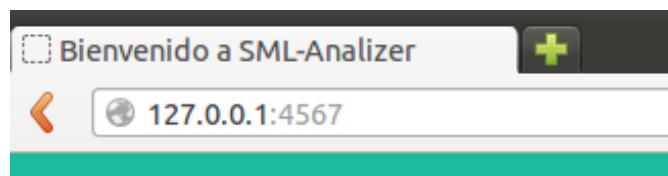
**E**L MANUAL DE USUARIO provee información de alto nivel con el fin de ofrecer al usuario final un conjunto de instrucciones simples, precisas y claras para ejecutar e interactuar con SML-analyzer.

Con el propósito de facilitar al usuario final la comprensión de este manual se integra un conjunto de imágenes ilustrativas que presenten el proceso de interacción con la aplicación.

Fuera del alcance de este documento está la explicación del lenguaje SML utilizado en el archivo con la misma extensión, debido a esto se utilizará un archivo sml de ejemplo con el fin de ilustrar el proceso de análisis.

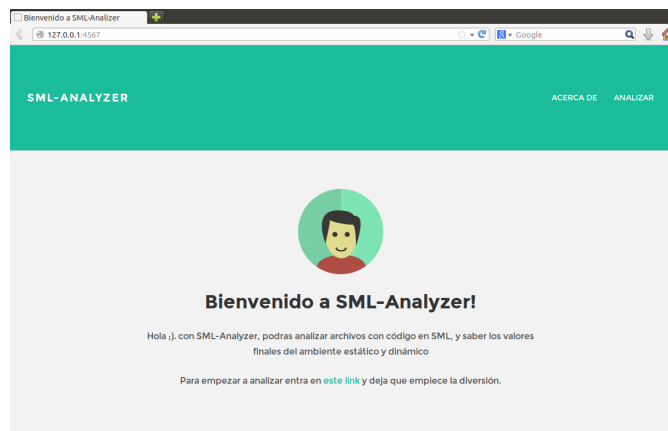
### 4.1. Ejecutando SML-analyzer

Para iniciar debe abrir la IP y el puerto del servidor en su navegador web, como se muestra en la figura 4.1



**Figura 4.1:** Introducir la dirección del servidor

Obtendrá como resultado una pantalla similar a la presentada en la figura 4.2



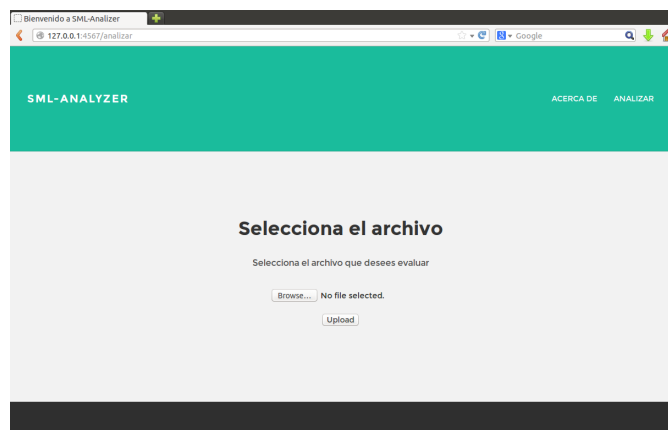
**Figura 4.2:** Página inicial de SML-analyzer

Allí dispone una serie de elementos que le permiten acceder a las distintas funcionalidades de la aplicación.



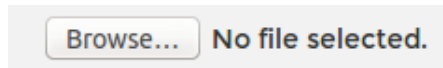
**Figura 4.3:** Botones de la página inicial

Si presiona el botón ANALIZAR podrá acceder al servicio de carga de archivos sml, similar al mostrado en la figura 4.4



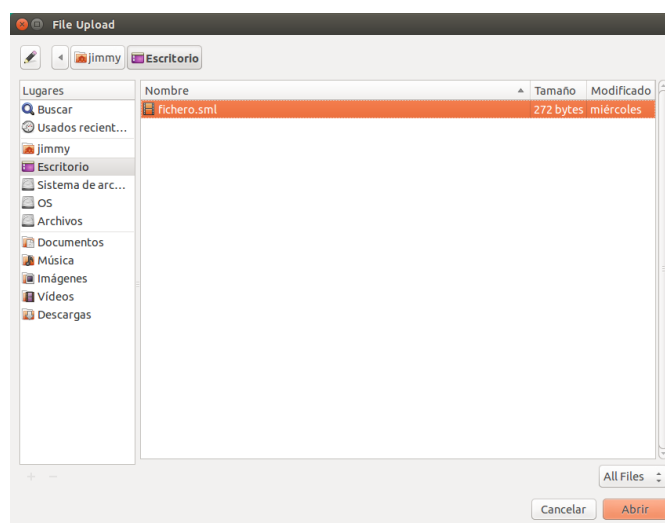
**Figura 4.4:** Página de carga de archivos sml

Para cargar un archivo sml presione el botón BROWSE ubicado en el centro de la pantalla.



**Figura 4.5:** Botón de selección de archivo

Seleccione el archivo deseado en la pantalla de exploración mostrada al presionar el botón BROWSE y luego presione el botón ABRIR



**Figura 4.6:** Explorador de archivos

Luego presione el botón UPLOAD para que el archivo sea enviado al servidor.



**Figura 4.7:** Botón de carga de archivo

Al presionarlo, la aplicación analizará y mostrará los resultados de dicho análisis de una forma similar a la expuesta en la figura 4.8



Identificador	Ambiente dinámico	Ambiente estático
X	3	int
S	3.45	float
tuplis	(5,7)	int*int
lista	[6,8,7,4,3]	int list

**Figura 4.8:** Página de resultados

Con esto concluye el proceso de carga y análisis de archivos, si desea repetir el proceso con otro archivo puede presionar el botón ANALIZAR ubicado en la parte superior derecha de la ventana, como se muestra en la figura 4.3