

Autómata Creator

Elkin Mauricio Garces Duque CC 1055831540

Manual técnico

1. Problema

Construya un programa de computador que permita el ingreso de un autómata finito. Su programa debe tener las siguientes funcionalidades:

1. Ingreso del autómata finito: como Ud. lo desee, en tabla de transiciones o en diagrama de transiciones.
2. Mostrar el autómata finito generado.
3. Determinar si el autómata entrado es determinístico o no determinístico. Si es no determinístico, convertirlo a determinístico y luego simplificarlo. Si es determinístico, simplemente proceder a simplificarlo.
4. Comprobar el autómata, es decir, que el usuario pueda entrar una hilera y que el autómata entrado acepte o rechace la hilera entrada.

Los símbolos para **fin de secuencia** y **secuencia nula** pueden ser elegidos por cada equipo.

Notas:

- Las prácticas se desarrollan en equipos de máximo dos estudiantes.
- Prácticas iguales se califican con cero.
- No olvide manual técnico y manual de usuario si su programa tiene alguna característica especial.

Propuesta

Implementar un aplicación web que le permita el usuario ingresar los parámetros para construir un autómata.

Haciendo uso de la definición de los componentes que tienen los automata el usuario deber tener a la mano los siguientes items:

1. Un conjunto finito de símbolos de entrada.
2. Un conjunto finito de estados.
3. Un estado designado como estado inicial.
4. Uno o más estados designados como estados de aceptación.
5. Un conjunto de transiciones(Tabla de transiciones).

Todos los datos mencionados anteriormente los debe tener el usuario a la mano para ingresarlos en la aplicación

A continuación se describen las principales funcionalidades del posible sistema como historias de usuario.

Historias de usuario	
H001	El usuario desea generar autómatas finitos
H002	El usuario desea validar una hilera de caracteres con el autómata que ingresó anteriormente
H003	El usuario desea saber si el autómata es determinístico o No determinístico
H004	El usuario desea convertir autómatas no determinístico en determinísticos

2. Arquitectura

Se emplea arquitectura cliente-servidor, en la parte de el servidor se procesan los datos que proporcione el cliente, es decir, la interpretación de los datos ingresados y la gestión en general de los autómatas. Por otro lado el cliente-usuario será el encargado de recibir los datos ingresados por este, tales como los estados, las símbolos, las hileras de validación, etc. El cliente será también el encargado de mostrar de una manera agradable y entendible al usuario los resultados obtenidos.

Además de la tradicional cliente servidor, se integrará una arquitectura orientada a servicios, exponiendo la aplicación como servicio, la cual permitirá exponer a muchos tipos de clientes las diferentes funcionalidades desarrolladas tales como la validación de hileras de símbolos en autómatas finitos.



Figura 1 Arquitectura a utilizar

3. Tecnologías utilizadas

Se hace necesario hacer uso de tecnologías actuales que permitan desarrollar rápido y verificación de funcionamiento de manera eficiente.



Cliente:

- HTML5:
- CSS3:

Estas dos tecnologías son el matrimonio ideal para la elaboración de la parte visual o Front, permitirá mostrar de la mejor manera al usuario.



- jQuery: es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, ayuda a gestionar el comportamiento de los elementos del DOM dados los diferentes eventos ingresados como interacciones con el usuario.



- MaterializeCss: Es un framework de estilos para personalizar algunos elementos de la aplicación.
- Jinja2: Es un motor de plantillas web para el lenguaje de programación python.

Servidor:

- Python 3: Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Toda nuestra lógica de programación desarrollada con base a los autómatas está implementada en python. Con ayuda de micro-frameworks como Flask para facilitar la comunicación.

4. Despliegue

La aplicación se ha desplegado sobre heroku para hacerla accesible en internet.

La aplicación desarrollada ha sido desplegada en la siguiente url:

<https://automatacreator.herokuapp.com/>

5. Repositorio.

A continuación ponemos a disposición el repositorio en GitHub. En el se encuentra todo el código correspondiente al programa.

<https://github.com/maurogarcesd/automatacreator>

6. Referencias

HTML. Disponible en: <https://es.wikipedia.org/wiki/HTML>

Hoja de estilos en cascada. Disponible en: https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada

jQuery. Disponible en: <https://es.wikipedia.org/wiki/JQuery>

Jinja (template engine). Disponible en: [https://en.wikipedia.org/wiki/Jinja_\(template_engine\)](https://en.wikipedia.org/wiki/Jinja_(template_engine))

Python. Disponible en: <https://desarrolloweb.com/articulos/1325.php>