

UNIVERSIDAD MARIANO GALVEZ DE GUATEMALA
FACULTAD DE INGENIERÍA EN SISTEMAS Y CIENCIAS DE LA COMPUTACIÓN



Manual Técnico Simulador de Autómatas

Autómatas y lenguajes formales

Ing. Otto Rigoberto Ortiz Pérez

Estudiante:

1890-23-2862 - Josué Fernando Hicho García

Fecha: 25/10/2025

Índice

Introducción	i
Objetivo del documento.....	ii
DOCUMENTACIÓN	1
Clase GraphvizGenerator:	1
GenerarCodigoDotDesdeMatriz(...)	1
generarSimboloPorDefecto(...)	1
obtenerColorEstado(...)	1
obtenerColorTransicion(...)	2
ordenarEstados(...)	2
Clase ControladorGraphviz:	3
detectarGraphviz()	3
verificarGraphviz(...)	3
actualizarVisualizacion(...)	3
actualizarVisualizacionDesdeMatriz(...)	3
generarImagen(...)	4
mostrarImagenEnVista(...)	4
guardarArchivoDot(...)	4
mostrarErrorGraphvizNoEncontrado()	4
mostrarMensajeError(...)	4
registrarTransicionUsada(...)	5
estaGraphvizDisponible()	5
getRutaGraphviz()	5
Clase ArchivoTexto.....	6
abrirArchivotxt(...)	6
guardarArchivo(...)	6
seleccionarArchivoAbrir()	6
extraerValorPatronEnArchivo(...)	6
extraerCadenasAnalizar(...)	6
extraerBloqueTransiciones(...)	7
mostrarError(...)	7
Clase: Controlador Documento	8

configurarListeners().....	8
abrirArchivo()	8
abrirPDFs(...)	8
abrirArchivoEjem(...).....	8
mostrarDatosTablas().....	9
probarCadena()	9
iniciarNuevaCadena(...)	9
procesarSiguienteElemento()	9
obtenerTransicion(...).....	9
validarCadenaFinal().....	10
actualizarGraphviz()	10
guardarArchivo()	10
guardarComoArchivo().....	10
Clase ModeloDocumento	11
cargarDatosDesdeArchivo(...).....	11
guardarDatosArchivo(...)	11
extrarDatosEspecificos()	11
procesarLista(...)	12
procesarTransiciones(...)	12
crearMatrizTransiciones()	12
Clase VistaPrincipal	13
mostrarEstadosAceptacionTabla(...)	13
mostrarSimboloEnTabla(...).....	13
mostrarCadenasEnTabla(...).....	13
mostrarTransicionesEnTabla(...)	13
limpiarColoresTabla().....	14
obtenerCadenaSeleccionada()	14
obtenerIndiceCadenaSeleccionada()	14
marcarCadenaValida(...)	14
mostrarEstadosTransicion(...)	15
setTextoEnCampo(...).....	15
mostrarContenido(...).....	15

mostrarEstadoInicial(...)	15
limpiarAFD()	15
resaltarEncabezadoSimbolo(...)	16
resaltarEstadoColumna(...)	16

Introducción

Este proyecto implica el desarrollo de una aplicación de escritorio que permite el análisis y la visualización de autómatas finitos a partir de la información contenida en un archivo de texto (.txt).

El sistema procesa este archivo para extraer los estados, las transiciones y los estados inicial y final, organizando estos datos en tablas estructuradas que facilitan la comprensión del autómata.

La aplicación utiliza la herramienta Graphviz para generar y mostrar una representación gráfica del autómata, permitiendo al usuario observar visualmente la estructura de los estados y las transiciones entre ellos.

La interfaz se desarrolló en Java, utilizando Swing para gestionar los componentes gráficos, lo que permite una interacción intuitiva y una experiencia visual clara.

Objetivo del documento

El presente documento tiene como objetivo describir los aspectos técnicos y estructurales del sistema “Simulador de Autómatas”, brindando la información necesaria para su comprensión, mantenimiento y futura ampliación.

Se detalla la arquitectura general, los componentes principales del software, las clases que intervienen en el procesamiento de los datos, y la forma en que se integran los módulos de lectura, procesamiento y visualización

DOCUMENTACIÓN

Clase GraphvizGenerator:

Utilidades para generar la representación DOT (Graphviz) de un autómata finito determinista a partir de una matriz de transiciones y metadatos (estados, símbolos, estado inicial/actual y estados de aceptación). También contiene funciones auxiliares para decidir colores y orden de estados.

GenerarCodigoDotDesdeMatriz(...)

Descripción: Genera el código en formato DOT para representar gráficamente un autómata finito determinista.

Parámetros: estadoInicial, estadosAceptacion, estados, simbolos, matrizTransiciones, estadoActual, transicionesUsadas.

Retorno: Cadena con el contenido DOT completo listo para procesar con Graphviz.

Función principal: Construye los nodos y transiciones del autómata, define estilos visuales y genera la estructura del grafo dirigido.

generarSimboloPorDefecto(...)

Descripción: Crea un símbolo por defecto entre dos estados usando un hash simple.

Parámetros: origen, destino.

Retorno: Cadena con el símbolo generado (normalmente "0" o "1").

Función principal: Permite representar una transición cuando no se especifica un símbolo explícito.

obtenerColorEstado(...)

Descripción: Determina el color y estilo visual de un estado según su tipo.

Parámetros: estado, estadoInicial, estadoActual, estadosAceptacion.

Retorno: Cadena con atributos DOT (por ejemplo, color y estilo).

Función principal: Resalta visualmente el estado actual y los estados de aceptación en el grafo.

obtenerColorTransicion(...)

Descripción: Asigna un color a una transición dependiendo de si fue utilizada recientemente.

Parámetros: origen, destino, simbolo, transicionesUsadas.

Retorno: Cadena con el color y grosor de línea en formato DOT.

Función principal: Destaca las transiciones activas o más recientes con un color diferente (naranja).

ordenarEstados(...)

Descripción: Ordena la lista de estados para mantener un orden visual coherente en el grafo.

Parámetros: estados, estadoInicial.

Retorno: Lista ordenada con el estado inicial primero y los demás en orden alfabético.

Función principal: Mejora la organización visual de los nodos en el diagrama generado por Graphviz.

Clase ControladorGraphviz:

Clase encargada de gestionar la visualización gráfica de autómatas utilizando Graphviz y Swing.

detectarGraphviz()

Descripción: Busca automáticamente la instalación de Graphviz en las rutas comunes del sistema.

Parámetros: Ninguno.

Retorno: String con la ruta del ejecutable Graphviz o null si no se encuentra.

Función principal: Detecta automáticamente Graphviz en diferentes sistemas operativos y ubicaciones de instalación.

verificarGraphviz(...)

Descripción: Valida que un ejecutable de Graphviz funcione correctamente.

Parámetros: ruta (ruta del ejecutable a verificar).

Retorno: boolean indicando si Graphviz funciona en la ruta especificada.

Función principal: Confirma la funcionalidad de Graphviz mediante ejecución de prueba.

actualizarVisualizacion(...)

Descripción: Genera y muestra la visualización del autómata a partir de código DOT.

Parámetros: codigoDot (código DOT que define el autómata).

Retorno: Void.

Función principal: Coordina el proceso completo de generación de imágenes con manejo robusto de errores.

actualizarVisualizacionDesdeMatriz(...)

Descripción: Convierte datos matriciales del autómata en visualización gráfica.

Parámetros: estadoInicial, estadosAceptacion, estados, simbolos, matrizTransiciones, estadoActual.

Retorno: Void.

Función principal: Transforma la representación matricial del autómata en gráfico visual.

generarImagen(...)

Descripción: Ejecuta Graphviz para convertir archivos DOT en imágenes PNG.

Parámetros: archivoDot (ruta del archivo .dot), archivoImagen (ruta de salida .png).

Retorno: Void.

Función principal: Invoca Graphviz de forma segura usando ProcessBuilder y maneja output para diagnóstico.

mostrarImagenEnVista(...)

Descripción: Carga y muestra la imagen generada en el componente JLabel de la interfaz.

Parámetros: archivoImagen (ruta de la imagen a mostrar).

Retorno: Void.

Función principal: Actualiza la interfaz con la imagen del autómata, incluyendo validación de archivos.

guardarArchivoDot(...)

Descripción: Guarda código DOT en archivo con manejo seguro de rutas y directorios.

Parámetros: contenido (código DOT), rutaCompleta (ruta absoluta del archivo destino).

Retorno: Void.

Función principal: Persiste el código DOT en archivo temporal con nombres únicos para evitar conflictos.

mostrarErrorGraphvizNoEncontrado()

Descripción: Muestra mensaje de error cuando Graphviz no está disponible en el sistema.

Parámetros: Ninguno.

Retorno: Void.

Función principal: Proporciona feedback claro al usuario sobre la necesidad de instalar Graphviz.

mostrarMensajeError(...)

Descripción: Muestra mensajes de error en el componente JLabel cuando falla la generación de imágenes.

Parámetros: mensaje (texto del error a mostrar).

Retorno: Void.

Función principal: Comunica errores al usuario directamente en la interfaz gráfica.

registrarTransicionUsada(...)

Descripción: Marca una transición como utilizada durante el análisis del autómata.

Parámetros: key (identificador único de la transición).

Retorno: Void.

Función principal: Resalta visualmente transiciones activas en la representación gráfica.

estaGraphvizDisponible()

Descripción: Verifica si Graphviz está disponible y funcional en el sistema.

Parámetros: Ninguno.

Retorno: boolean indicando disponibilidad de Graphviz.

Función principal: Permite a la aplicación detectar capacidades del sistema.

getRutaGraphviz()

Descripción: Proporciona la ruta detectada de Graphviz para fines de diagnóstico.

Parámetros: Ninguno.

Retorno: String con la ruta del ejecutable Graphviz.

Función principal: Facilita debugging y verificación de configuración.

Clase ArchivoTexto

Clase enfocada en los métodos generales del sistema relacionados con la gestión de los menús.

abrirArchivotxt(...)

Descripción: Lee y carga el contenido completo de un archivo de texto.

Parámetros: archivo (objeto File que representa el archivo a abrir).

Retorno: String con el contenido completo del archivo.

Función principal: Carga datos de autómatas desde archivos de texto.

guardarArchivo(...)

Descripción: Guarda contenido textual en un archivo especificado.

Parámetros: archivo (destino), contenido (texto a guardar).

Retorno: Void.

Función principal: Persiste automatas y configuraciones en disco.

seleccionarArchivoAbrir()

Descripción: Muestra un diálogo para que el usuario seleccione un archivo .txt.

Parámetros: Ninguno.

Retorno: File objeto representando el archivo seleccionado, o null si se cancela.

Función principal: Proporciona interfaz de usuario para selección de archivos.

extraerValorPatronEnArchivo(...)

Descripción: Busca y extrae un valor asociado a un patrón específico en el contenido.

Parámetros: contenido (texto donde buscar), patron (texto a buscar al inicio de línea).

Retorno: String con el valor encontrado después del patrón.

Función principal: Extrae configuraciones específicas como estado inicial o símbolos.

extraerCadenasAnalizar(...)

Descripción: Extrae una lista de cadenas que siguen a un patrón específico en el archivo.

Parámetros: contenido (texto completo), patron (etiqueta que precede a las cadenas).

Retorno: List<String> con las cadenas a analizar.

Función principal: Obtiene las cadenas de prueba para el análisis del autómata.

extraerBloqueTransiciones(...)

Descripción: Extrae el bloque completo de definiciones de transiciones del archivo.

Parámetros: contenido (texto completo del archivo).

Retorno: String con todas las líneas de transiciones.

Función principal: Separa la sección de transiciones para su procesamiento específico.

mostrarError(...)

Descripción: Muestra un mensaje de error en un diálogo modal.

Parámetros: mensaje (texto del error), nombre (título del diálogo).

Retorno: Void.

Función principal: Proporciona feedback al usuario sobre errores de archivo.

Clase: Controlador Documento

Clase encargada de la gestión entre la vista, modelo y clases auxiliares para el funcionamiento general del sistema.

`configurarListeners()`

Descripción: Configura todos los listeners de acción para los botones de la interfaz principal.

Parámetros: Ninguno.

Retorno: Void.

Función principal: Establece la conexión entre la interfaz de usuario y la lógica de la aplicación.

`abrirArchivo()`

Descripción: Maneja la apertura de archivos mediante diálogo de selección.

Parámetros: Ninguno.

Retorno: Void.

Función principal: Permite al usuario seleccionar y cargar un archivo de autómatas.

`abrirPDFs(...)`

Descripción: Busca y abre archivos PDF embebidos en la aplicación desde múltiples rutas posibles.

Parámetros: nombrePDF (nombre del archivo PDF a buscar y abrir).

Retorno: Void.

Función principal: Proporciona acceso a manuales y documentación del sistema mediante la creación de archivos temporales.

`abrirArchivoEjem(...)`

Descripción: Carga archivos de ejemplo embebidos en la aplicación desde recursos del JAR.

Parámetros: nombreArchivo (nombre del archivo de ejemplo a cargar).

Retorno: Void.

Función principal: Facilita el acceso a ejemplos predefinidos del sistema mediante archivos temporales.

mostrarDatosTablas()

Descripción: Procesa y muestra los datos del autómata en las tablas de la interfaz.

Parámetros: Ninguno.

Retorno: Void.

Función principal: Convierte los datos del modelo en representación visual en tablas.

probarCadena()

Descripción: Inicia el proceso de validación de una cadena seleccionada.

Parámetros: Ninguno.

Retorno: Void.

Función principal: Coordina la validación paso a paso de cadenas en el autómata.

iniciarNuevaCadena(...)

Descripción: Prepara el entorno para procesar una nueva cadena de prueba.

Parámetros: cadena (cadena a validar).

Retorno: Void.

Función principal: Inicializa variables de estado y configura el entorno para el análisis.

procesarSiguienteElemento()

Descripción: Ejecuta un paso en la validación de la cadena actual.

Parámetros: Ninguno.

Retorno: Void.

Función principal: Procesa el siguiente símbolo de la cadena y actualiza el estado del autómata.

obtenerTransicion(...)

Descripción: Consulta la matriz de transiciones para obtener el siguiente estado.

Parámetros: estadoActual, simbolo.

Retorno: String con el estado destino o null si no existe transición.

Función principal: Resuelve las transiciones del autómata basándose en el estado y símbolo actual.

validarCadenaFinal()

Descripción: Evalúa si la cadena procesada es aceptada por el autómata.

Parámetros: Ninguno.

Retorno: Void.

Función principal: Determina el resultado final de la validación y notifica al usuario.

actualizarGraphviz()

Descripción: Actualiza la visualización gráfica del autómata en tiempo real.

Parámetros: Ninguno.

Retorno: Void.

Función principal: Sincroniza la representación visual con el estado actual del autómata.

guardarArchivo()

Descripción: Guarda el contenido actual en el archivo abierto.

Parámetros: Ninguno.

Retorno: Void.

Función principal: Persiste los cambios realizados en el autómata.

guardarComoArchivo()

Descripción: Permite guardar el contenido en un nuevo archivo.

Parámetros: Ninguno.

Retorno: Void.

Función principal: Facilita la creación de nuevas instancias de autómatas.

Clase ModeloDocumento

La clase ModeloDocumento representa el núcleo de datos de la aplicación. Se encarga de:

- Cargar y parsear archivos de definición de autómatas
- Almacenar todos los componentes del autómata ($Q, \Sigma, \delta, q_0, F$)
- Estructurar los datos en formatos utilizables por la vista y controlador
- Proporcionar acceso controlado a los datos del autómata
- Gestionar la persistencia de los datos en archivos

cargarDatosDesdeArchivo(...)

Descripción: Carga y procesa un archivo de autómata, extrayendo todos sus componentes.

Parámetros: archivo (archivo .txt con la definición del autómata).

Retorno: Void.

Función principal: Inicializa el modelo completo del autómata desde un archivo de texto.

guardarDatosArchivo(...)

Descripción: Guarda el contenido del autómata en un archivo especificado.

Parámetros: archivo (destino), nuevoContenido (contenido a guardar).

Retorno: Void.

Función principal: Persiste el estado actual del autómata en el sistema de archivos.

extrarDatosEspecificos()

Descripción: Procesa el contenido del archivo para extraer todos los componentes del autómata.

Parámetros: Ninguno.

Retorno: Void.

Función principal: Coordina la extracción y estructuración de todos los elementos del autómata.

procesarLista(...)

Descripción: Convierte una cadena de elementos separados por comas en una lista.

Parámetros: texto (cadena con elementos separados por comas).

Retorno: List<String> con los elementos procesados.

Función principal: Normaliza y estructura datos de estados, símbolos y estados de aceptación.

procesarTransiciones(...)

Descripción: Convierte el bloque de texto de transiciones en una lista estructurada.

Parámetros: transicionesStr (texto con las definiciones de transiciones).

Retorno: List<String[]> con las transiciones por estado.

Función principal: Parseo y validación de las reglas de transición del autómata.

crearMatrizTransiciones()

Descripción: Construye una matriz bidimensional que representa la función de transición.

Parámetros: Ninguno.

Retorno: String[][] matriz de transiciones [estados x símbolos].

Función principal: Crea la representación matricial para acceso eficiente a transiciones.

Clase VistaPrincipal

Clase principal que contiene la interfaz del sistema, esta clase posee algunos métodos auxiliares para otras clases para mostrar los datos y actualizar la interfaz, según los eventos programados en la clase controlador.

mostrarEstadosAceptacionTabla(...)

Descripción: Muestra la lista de estados de aceptación en una tabla formateada.

Parámetros: estados (lista de estados de aceptación).

Retorno: Void.

Función principal: Visualiza el conjunto F de estados finales del autómata.

mostrarSimboloEnTabla(...)

Descripción: Muestra los símbolos del alfabeto en una tabla numerada.

Parámetros: simbolos (lista de símbolos del alfabeto).

Retorno: Void.

Función principal: Visualiza el conjunto Σ del autómata.

mostrarCadenasEnTabla(...)

Descripción: Muestra las cadenas a analizar con opciones de selección y validación.

Parámetros: cadenas (lista de cadenas de prueba).

Retorno: Void.

Función principal: Presenta las cadenas para análisis con interfaz de selección única.

mostrarTransicionesEnTabla(...)

Descripción: Construye y muestra la matriz de transiciones del autómata.

Parámetros: estados, simbolos, matrizTransiciones.

Retorno: Void.

Función principal: Visualiza la función de transición δ en formato matricial.

resaltarCelda(...)

Descripción: Resalta visualmente una transición específica durante el análisis.

Parámetros: fila, columna, estadoActual, simboloActual.

Retorno: Void.

Función principal: Destaca la transición actual y el estado durante la ejecución paso a paso.

limpiarColoresTabla()

Descripción: Restablece los colores por defecto de todas las celdas de la tabla de transiciones.

Parámetros: Ninguno.

Retorno: Void.

Función principal: Limpia el resaltado visual previo antes de una nueva ejecución.

obtenerCadenaSeleccionada()

Descripción: Recupera la cadena actualmente seleccionada en la tabla de cadenas.

Parámetros: Ninguno.

Retorno: String con la cadena seleccionada o null si no hay selección.

Función principal: Obtiene la cadena objetivo para el análisis actual.

obtenerIndiceCadenaSeleccionada()

Descripción: Obtiene el índice de la cadena seleccionada en la tabla.

Parámetros: Ninguno.

Retorno: int con el índice de la cadena seleccionada o -1 si no hay selección.

Función principal: Identifica la posición de la cadena para actualizaciones posteriores.

marcarCadenaValida(...)

Descripción: Marca el resultado de validación de una cadena en la tabla.

Parámetros: indiceCadena, esValida.

Retorno: Void.

Función principal: Actualiza la interfaz con el resultado de la validación (Sí/No).

mostrarEstadosTransicion(...)

Descripción: Actualiza los campos de texto que muestran el estado actual y siguiente.

Parámetros: estadoActual, estadoSiguiente.

Retorno: Void.

Función principal: Muestra el progreso del análisis en tiempo real.

setTextoEnCampo(...)

Descripción: Actualiza el campo que muestra el símbolo actual siendo procesado.

Parámetros: texto (símbolo actual).

Retorno: Void.

Función principal: Visualiza el símbolo en procesamiento durante el análisis.

mostrarContenido(...)

Descripción: Muestra el contenido del archivo en el área de texto principal.

Parámetros: contenido (texto del archivo cargado).

Retorno: Void.

Función principal: Presenta el contenido crudo del archivo de autómeta.

mostrarEstadoInicial(...)

Descripción: Muestra el estado inicial del autómeta en el campo correspondiente.

Parámetros: estado (estado inicial q0).

Retorno: Void.

Función principal: Visualiza el estado inicial del autómeta.

limpiarAFD()

Descripción: Limpia todos los campos y visualizaciones relacionadas con el autómeta.

Parámetros: Ninguno.

Retorno: Void.

Función principal: Restablece la interfaz a su estado inicial.

resaltarEncabezadoSimbolo(...)

Descripción: Resalta el encabezado de la columna del símbolo actual en la tabla de transiciones.

Parámetros: columnaSimbolo (índice de la columna a resaltar).

Retorno: Void.

Función principal: Destaca visualmente el símbolo actual en la matriz.

resaltarEstadoColumna(...)

Descripción: Resalta la celda del estado actual en la primera columna de la tabla.

Parámetros: filaEstado (índice de la fila del estado actual).

Retorno: Void.

Función principal: Destaca visualmente el estado actual en la matriz.