

Nombre de la asignatura: Lenguajes Y Autómatas

Nombre del proyecto: WALL•E

Título del reporte: Reporte final

Fecha de entrega: 12 de diciembre de 2016

Participantes en el proyecto:

Nayeli Mendoza Farias

Hugo Arnoldo Gutiérrez Vargas

Resumen

El proyecto “WALL•E” es un juego mediante código, se maneja el movimiento de un robot.

Se realizó el análisis del léxico y sintáctico para implantarlo en el proyecto “WALL•E” así llevar a cabo el desarrollo del léxico y sintáctico. Que es el análisis léxico y sintáctico:

Un analizador léxico lee los caracteres de la entrada y los agrupa en “objetos token”. Junto con un símbolo de terminal que se utiliza para las decisiones de análisis sintáctico, un objeto token lleva información adicional en forma de valores de atributos (Aho et al., 2008).

La segunda fase del compilador es el análisis sintáctico o parsing. El parser (analizador sintáctico) utiliza los primeros componentes de los tokens producidos por el analizador de léxico para crear una representación intermedia en forma de árbol que describa la estructura gramatical del flujo de tokens (Aho et al., 2008).

Se desarrolló el análisis léxico con los siguientes caracteres: Inicio, Avanza, GDerecha, Glzquierda, Activa, Fin, ;, (y). Los caracteres anteriores los que contienen el proyecto “WALL•E”.

La estructura del análisis sintáctico es la gramática:

INICI ::= INIC expr_list TERM;

expr_list ::= expr_list expr_part | expr_part;

expr_part ::= exprm;

exprm ::= AVAN LPAREN NUMBER:n RPAREN SEMI {:
System.out.print("Avanza: "+n); ;}

| GDE SEMI{ System.out.print("GDerecha"); ;}

| GIZ SEMI{ System.out.print("Gizquierda"); ;}

| ACT SEMI{ System.out.print("Activa"); ;};

Posteriormente se realizó el diseño de así también la programación para los botones: guardar, abrir, analizar, ejecutar y salir.

Tabla de Contenido

Resumen.....	2
Introducción.....	3
Definición del problema.....	4
Objetivos y alcances del proyecto.....	5
Diseño conceptual.....	6
Diseño de configuración.....	6
Imagen1._ análisis sintáctico.....	7
Imagen2._ análisis sintáctico2.....	7
Imagen3._ diseño inicial.....	8
Imagen4._ diseño2.....	9
Imagen5._ diseño3.....	9
Imagen4._ diseño4.....	10
Imagen5._ diseño final.....	11
Propuesta de proceso de producción.....	11
Referencias y bibliografía.....	12
Anexos.....	12

Introducción

El proyecto “WALL•E” es un juego que su funcionamiento es mediante código los caracteres léxicos: Inicio, Avanza, GDerecha, Glzquierda, Activa, Fin, ;, (y).

La estructura del análisis sintáctico es la gramática:

```
INICI ::= INIC expr_list TERM;
expr_list ::= expr_list expr_part | expr_part;
expr_part ::= exprm;
exprm ::= AVAN LPAREN NUMBER:n RPAREN SEMI {:
System.out.print("Avanza: "+n); :}
| GDE SEMI{: System.out.print("GDerecha"); :}
| GIZ SEMI{: System.out.print("Glzquierda"); :}
| ACT SEMI{: System.out.print("Activa"); :};
```

El análisis sintáctico tiene como gramática: Inicio, tal como se muestra con la I mayúscula y lo sucesivo en minúscula es la palabra que inicializa el código. Así como Fin, finaliza el código se escribe como se escribió posteriormente. En la parte central puede escribir (GDerecha;), tal como se muestra anteriormente. El (Avanza (número);), se escribe de igual forma como se muestra a diferencia de los demás tiene paréntesis y un número en el centro de ellos también finaliza con punto y coma. El (Glzquierda;) el giro a la izquierda va como se muestra anteriormente, seguido de un (ACTIVA;) tal como se muestra.

Un ejemplo claro de un código se presenta a continuación:

Inicio

Avanza (1);

GDerecha;

Avanza (1);

Glzquierda;

Activa;

Fin

El diseño cuenta con una consola de escritura de código, también la implementación de la vista de movimiento del robot mediante el código y también consta de una botonera de: guardar, abrir, analizar, ejecutar y salir.

El diseño del juego cuenta con un botón guardar el cual permite al usuario guardar su código donde lo requiere y con el nombre que considere.

El botón abrir permite al usuario abrir su código de donde lo tenga guardado.

El botón analizar permite al usuario analizar los errores tanto léxicos y sintácticos que tenga en el código.

El botón ejecutar lleva a cabo la ejecución del programa ya analizado el código sin errores y así hacer las acciones que se le codificaron.

El botón salir permite al usuario salir de juego cuando él lo desee.

Definición del problema.

La falta de comprensión del léxico y sintáctico lleva a una mal desarrollo de ello,

Mediante las fallas encontradas se modificó para mejorar.

Un mal diseño del juego muy poco atractivo visualmente, se llevó a cabo las correcciones correspondientes.

Objetivos y alcances del proyecto

- **Descripción del producto:** Es un juego maneja un robot por código.
- **Lista de requerimientos:**

Software: Léxico y sintáctico.

Scanner.java, parser.java, sym.java, CUP\$parser\$actions.class, parser.class, Scanner.class, sym.class

Java: NetBeans IDE 8.1

Sublime Text 3

Hardware: El usuario requiere una computadora.

- **Lista de especificaciones:** Escribir el código con la gramática correspondiente.

Al diseño se le agregó un apartado de ayuda con las especificaciones para que el usuario pueda verificar el Léxico y sintáctico que a continuación se muestra:

1. inicio (**Inicio**), tiene que escribirse con la I mayúscula y lo sucesivo en minúscula es la palabra que inicializa el código.
2. fin (**Fin**), tiene que escribirse con la F mayúscula y lo sucesivo en minúscula y esta es la instrucción que finaliza el código.

En la parte central pueden ir las siguientes instrucciones:

3. Giro a la derecha (**GDerecha;**), tal como se muestra anteriormente: se escribe con G mayúscula y D mayúscula lo siguiente en minúscula seguido de un punto y coma.

4. Avanza (**Avanza (número);**), se escribe de igual forma como se muestra: se escribe la con A con mayúscula lo siguiente en minúscula, a diferencia de los demás tiene paréntesis y un número en el centro de ellos también finaliza con punto y coma.
5. Giro a la Izquierda (**Gizquierda;**) se escribe como se muestra anteriormente: con G mayúscula e I mayúscula lo siguiente en minúscula seguido de un punto y coma. seguido de un (**Activa;**) tal como se muestra: con A mayúscula lo siguiente en minúscula seguido de un punto y coma.

Diseño conceptual

Se muestra el reporte de actividades y resultados del proyecto “WALL•E”, Así como las mejoras y correcciones indicadas durante el desarrollo del mismo.

- **Síntesis de información**

Inicialmente se regeneró el léxico en base a las especificaciones para generar el Scanner.java y después el sintáctico para generar (sym.java, CUP\$parser\$actions.class, parser.class, Scanner.class, sym.class) y así hacer la generación de código para los botones, así también como la generación de código del movimiento del robot.

En un inicio implementó la descripción de los tipos de errores que se generan del léxico y sintáctico.

En el movimiento análisis de cómo implementar el movimiento de robot en el parser.

- **Opciones de solución**

Implementar código para marcar los errores correspondientes.

Consultar tutoriales para implementar el movimiento del robot consultar en diversas fuentes.

- **Evaluación y selección**

Se evaluaron las diferentes correcciones y así llegar al mejor resultado para la implementación al proyecto.

Diseño de configuración

- **Opciones de solución**

Como se observa en la imagen (ver imagen1) llevó a cabo la identificación de algunos tipos de errores, para realizar un mejor análisis se implementó en sintáctico que se muestra en (ver imagen2).

```
sintáctico.cup
2  import java.io.*;
3  import java_cup.runtime.*;
4  action code {
5  }
6  parser code
7  {
8      public static void main ( String args[] ) throws Exception {
9          new parser(new Scanner(new FileReader(args[0])).parse(); }
10
11      public void syntax_error(Symbol s){
12          report_error("Error en línea: "+ s.left+"", column: "+s.right,null);
13      }
14
15      public void report_fatal_error(String message, Object info){
16          report_error("Codigo de error: " + info.toString(), null);
17          if ("#0".equals(info.toString())) {report_error("Error de sintaxis: Falta FIN", null);}
18          if ("#4".equals(info.toString())) {report_error("Error de sintaxis: Falta punto y coma o falta INICIO", null);}
19          if ("#5".equals(info.toString())) {report_error("Error de sintaxis: Falta punto y coma", null);}
20          if ("#6".equals(info.toString())) {report_error("Error de sintaxis: Falta punto y coma", null);}
21          if ("#7".equals(info.toString())) {report_error("Error de sintaxis: Falta punto y coma", null);}
22          if ("#8".equals(info.toString())) {report_error("Error de sintaxis: Falta punto y coma", null);}
23          if ("#9".equals(info.toString())) {report_error("Error de sintaxis: Estructura invalida", null);} //avanza
24          if ("#10".equals(info.toString())) {report_error("Error de sintaxis: Falta nuermero de pasos o caracter invalido", null);}
25          if ("#11".equals(info.toString())) {report_error("Error de sintaxis: Estructura invalida", null);}
26          if ("#2".equals(info.toString())) {report_error("Error de sintaxis: Estructura invalida", null);}
27      }
28  }
29  terminal SEMI, INIC, AVAN, GDE, GIZ, ACT, TERM;
30  terminal LPAREN, RPAREN;
31  terminal Integer NUMBER;
32  non terminal INICI, expr_list, expr_part, exprm;
33  precedence left AVAN, GDE;
34  precedence left GIZ, ACT;
35  /* Gramática */
36  INICI ::= INIC expr_list TERM;
37  expr_list ::= expr_list expr_part | expr_part;
38  expr_part ::= exprm;
39  exprm ::= AVAN LPAREN NUMBER:n RPAREN SEMI { : System.out.print("Avanza: "+n); : }
40  | GDE SEMI { : System.out.print("GDerecha"); : }
41  | GIZ SEMI { : System.out.print("GIZquierda"); : }
42  | ACT SEMI { : System.out.print("Activa"); : }
```

Imagen1._ análisis sintáctico

Imagen2._ análisis sintáctico2

- **Dibujos y modelos 3D**

En la imagen.3 se muestra el diseño inicial, no se tenía una perspectiva clara de lo requerido en ese momento.

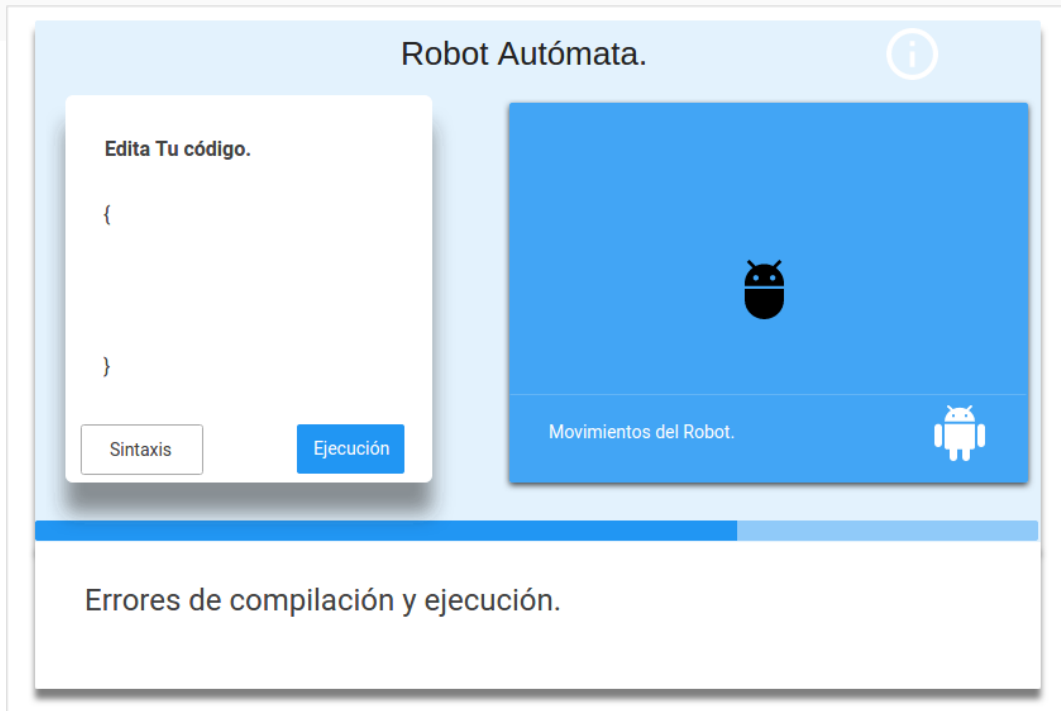


Imagen3._ diseño inicial

Teniendo una visión más amplia llevo al diseño imagen.4 con la corrección correspondiente.

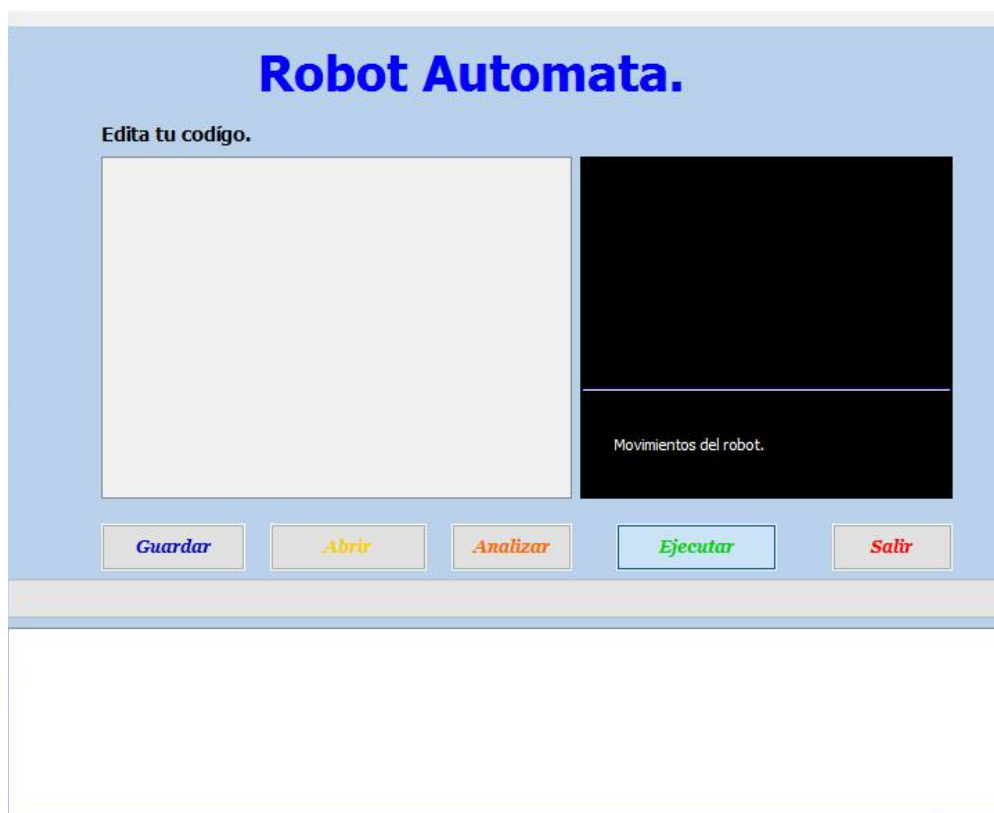


Imagen4._ diseño2

Como se observa en las imágenes 5 y 6 el diseño se hizo visualmente atractivo para el usuario.

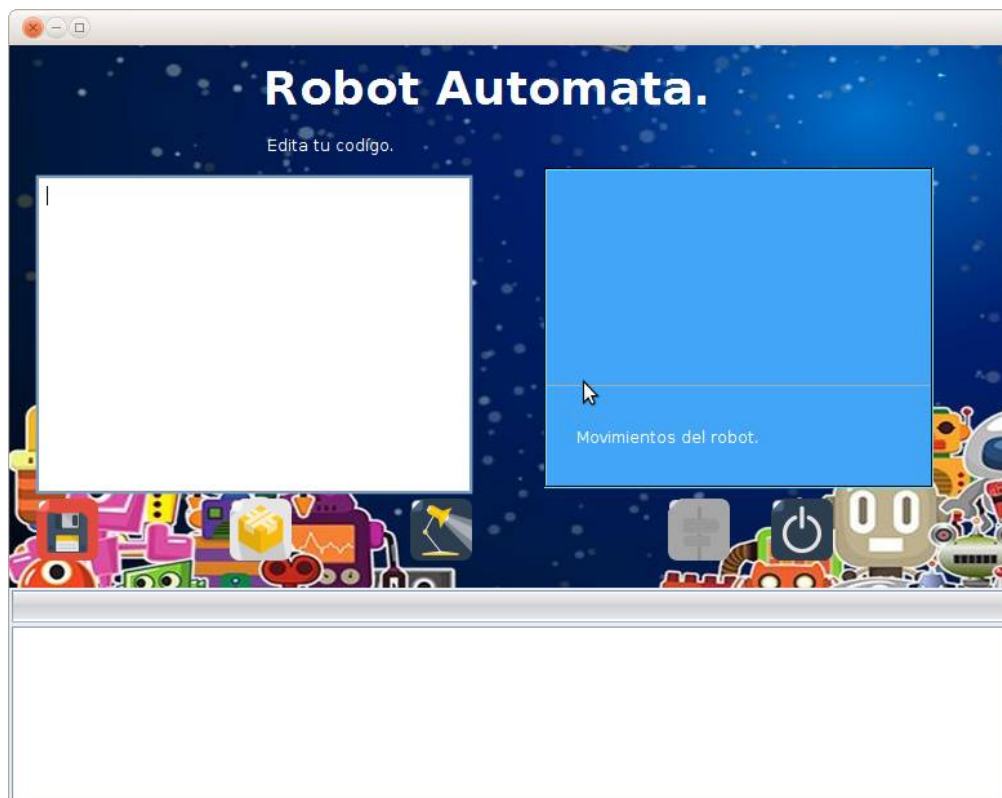


Imagen5._diseño3

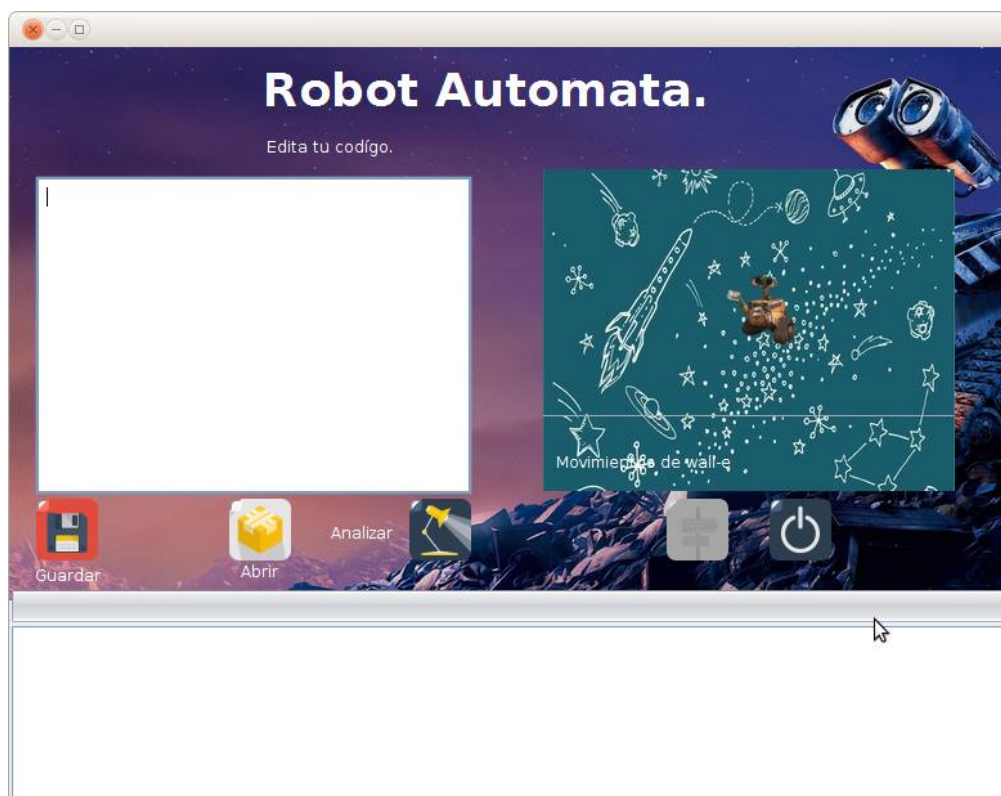


Imagen4._ diseño4

- **Evaluación y selección**

A continuación se muestra en la imagen5 el diseño final una vez implementado el movimiento del robot, llego a la conclusión que fue el mejor diseño.



Imagen5._ diseño final

Propuesta de proceso de producción

Continuación se muestra el reporte de las actividades y resultados la etapa del proyecto “WALL•E”.

• Conclusiones y comentarios finales

- Situación actual del proyecto. Comentar si los resultados a la fecha de entrega del reporte corresponden a lo planteado en la etapa del proyecto, si la calidad de los resultados es satisfactoria, si es necesario completar o profundizar en alguna actividad, etc.
- Principales interrogantes a resolver en actividades futuras. Aspectos del proyecto o del producto a los que se debe prestar particular atención o que se debe dar prioridad sobre otros.
- Lecciones aprendidas sobre el producto. Qué se aprendió sobre el producto durante las etapas del proyecto que se han desarrollado.
- Lecciones aprendidas sobre el proceso de desarrollo de productos. Qué se aprendió sobre el proceso al realizar las actividades: sugerencias de mejora, cuál actividad es la más provechosa, cuál fue de poca utilidad, etc.

Referencias y bibliografía

. Páginas Web, artículos, libros, entrevistas personales, en formato IEEE.

Bibliografía: Aho, A., Sethi, R., Ullman, J., Flores Suárez, P. and Botella i López, P. (2008). Compiladores principios, técnicas y herramientas. 1st ed. México: Luis Miguel Cruz Castillo.

Anexos

Documentos que por su extensión o contenido no son necesarios en el cuerpo del trabajo pero que constituyen documentos generados durante las actividades del proyecto reportado o contienen información relevante para el proyecto. Algunos ejemplos son: encuestas aplicadas a usuarios, hojas de especificación de productos comerciales, láminas de presentación de la etapa del proyecto, control de gastos del proyecto, etc.