



PROYECTO 1

Objetivos:

Este proyecto tiene como objetivo que el estudiante practique el análisis léxico de un lenguaje formal y que amplíe sus conocimientos con el lenguaje de programación Java, así como aplicar los conocimientos adquiridos en la clase y en el laboratorio.

Descripción:

La empresa Galaxy Games solicita sus servicios para crear un videojuego llamado ¡Repáralo! Este juego consiste en ir reparando bloques que han sido destruidos por un enemigo. La creación de los niveles se realizará por medio de comandos de programación de un lenguaje específicamente creado para este juego, dichos comandos servirán para todo el funcionamiento del juego que va desde el diseño del nivel, hasta el flujo del juego mismo.

La aplicación debe proveer la opción de cargar archivos de texto con la extensión .rep a un editor, para poder analizar el contenido del editor y cargar el tablero con los personajes.

Definición del lenguaje:

El lenguaje tiene varias etiquetas que se definen a continuación:

Etiqueta Nivel:

Esta etiqueta contendrá toda la información del nivel, puede existir más de un nivel dentro del archivo.

```
<Nivel codigo=1 nombre=Nivel1 >  
  //Bloque Edificio  
  //Bloque Personaje  
  //Bloque Enemigo  
</Nivel>
```

A continuación, se detallan los elementos de la etiqueta nivel:

- **Codigo:** El código del nivel debe ser un número entero mayor a 0. Sirve para poder definir el número de nivel, se debe de mostrar en el juego el nivel más bajo primero y así sucesivamente.
- **Nombre:** Es el nombre que define al nivel, debe ser un identificador, es decir, debe iniciar con letra, seguido de letras o números

Etiqueta Edificio:

Esta etiqueta contendrá toda la información del edificio, dentro de él pueden venir las etiquetas de Bloque y Escalera.

```
<Edificio DimensionX=5 DimensionY=8>
    //Etiquetas Bloque
    //Etiquetas Escalera
</Edificio>
```

A continuación, se detallan los elementos de la etiqueta Edificio:

- **DimensionX:** Define la dimensión en el eje x del edificio, la dimensión debe ser un número entero mayor a 0 y menor o igual a 15.
- **DimensionY:** Define la dimensión en el eje y del edificio, la dimensión debe ser un número entero mayor a 0 y menor o igual a 15.

Etiqueta Bloque:

Esta etiqueta permite definir la posición de aquellos bloques que están dañados inicialmente.

```
<Bloque PosicionX=0 PosicionY=0>
<Bloque PosicionX=1 PosicionY=3>
<Bloque PosicionX=1 PosicionY=4>
```

A continuación, se detallan los elementos de la etiqueta Bloque:

- **PosicionX:** Define la posición en el eje x del edificio donde se encuentra el bloque dañado, la posición debe ser un número entero mayor o igual a 0.
- **PosicionY:** Define la posición en el eje y del edificio donde se encuentra el bloque dañado, la posición debe ser un número entero mayor o igual a 0.

Etiqueta Escalera:

Esta etiqueta permite definir la posición de las escaleras en el juego. Solo en estas posiciones el personaje podrá cambiar de nivel dentro del edificio.

```
<Escalera PosicionX=0 PosicionY=1>
<Escalera PosicionX=4 PosicionY=2>
<Escalera PosicionX=0 PosicionY=3>
```

A continuación, se detallan los elementos de la etiqueta Escalera:

- **PosicionX:** Define la posición en el eje x del edificio donde se encuentra la escalera, la posición debe ser un número entero mayor o igual a 0.
- **PosicionY:** Define la posición en el eje y del edificio donde se encuentra la escalera, la posición debe ser un número entero mayor o igual a 0.

Etiqueta Personaje:

Esta etiqueta permite definir la imagen y las vidas que tendrá el personaje.

```
<Personaje imagen="C://Imagenes/Personaje.png">
```

A continuación, se detallan los elementos de la etiqueta Personaje:

- **Imagen:** Define una ruta donde se encontrará una imagen que representará al personaje principal, debe de estar encerrado dentro de comillas dobles.

Etiqueta Enemigo:

Esta etiqueta contendrá la imagen del enemigo y la frecuencia de los movimientos.

```
<Enemigo imagen="C://Imagenes/Enemigo.png">  
    //Etiqueta Movimiento  
</Enemigo>
```

A continuación, se detallan los elementos de la etiqueta Enemigo:

- **Imagen:** Define una ruta donde se encontrará una imagen que representará al enemigo, debe de estar encerrado dentro de comillas dobles.

Etiqueta Movimiento:

Esta etiqueta contendrá los movimientos que debe de seguir el enemigo dentro del edificio.

```
<Movimiento PosicionX=2 PosicionY=7>  
<Movimiento PosicionX=4 PosicionY=7>  
<Movimiento PosicionX=4 PosicionY=6>
```

A continuación, se detallan los elementos de la etiqueta Movimiento:

- **PosicionX:** Define la posición en el eje x al que el enemigo debe de trasladarse, la posición debe ser un número entero mayor o igual a 0.
- **PosicionY:** Define la posición en el eje y al que el enemigo debe de trasladarse, la posición debe ser un número entero mayor o igual a 0.

Archivo de entrada ejemplo:

```
<Nivel codigo=1 nombre=Nivel1 >

  <Edificio DimensionX=5 DimensionY=8>

    <Bloque PosicionX=0 PosicionY=0>
    <Bloque PosicionX=1 PosicionY=3>
    <Bloque PosicionX=1 PosicionY=4>
    <Bloque PosicionX=2 PosicionY=1>
    <Bloque PosicionX=3 PosicionY=3>
    <Bloque PosicionX=4 PosicionY=1>
    <Bloque PosicionX=5 PosicionY=2>
    <Bloque PosicionX=5 PosicionY=4>
    <Bloque PosicionX=6 PosicionY=1>
    <Bloque PosicionX=6 PosicionY=4>

    <Escalera PosicionX=0 PosicionY=1>
    <Escalera PosicionX=4 PosicionY=2>
    <Escalera PosicionX=0 PosicionY=3>
    <Escalera PosicionX=4 PosicionY=4>
    <Escalera PosicionX=0 PosicionY=5>
    <Escalera PosicionX=3 PosicionY=6>
    <Escalera PosicionX=4 PosicionY=7>

  </Edificio>

  <Personaje imagen="C://Imagenes/Personaje.png">

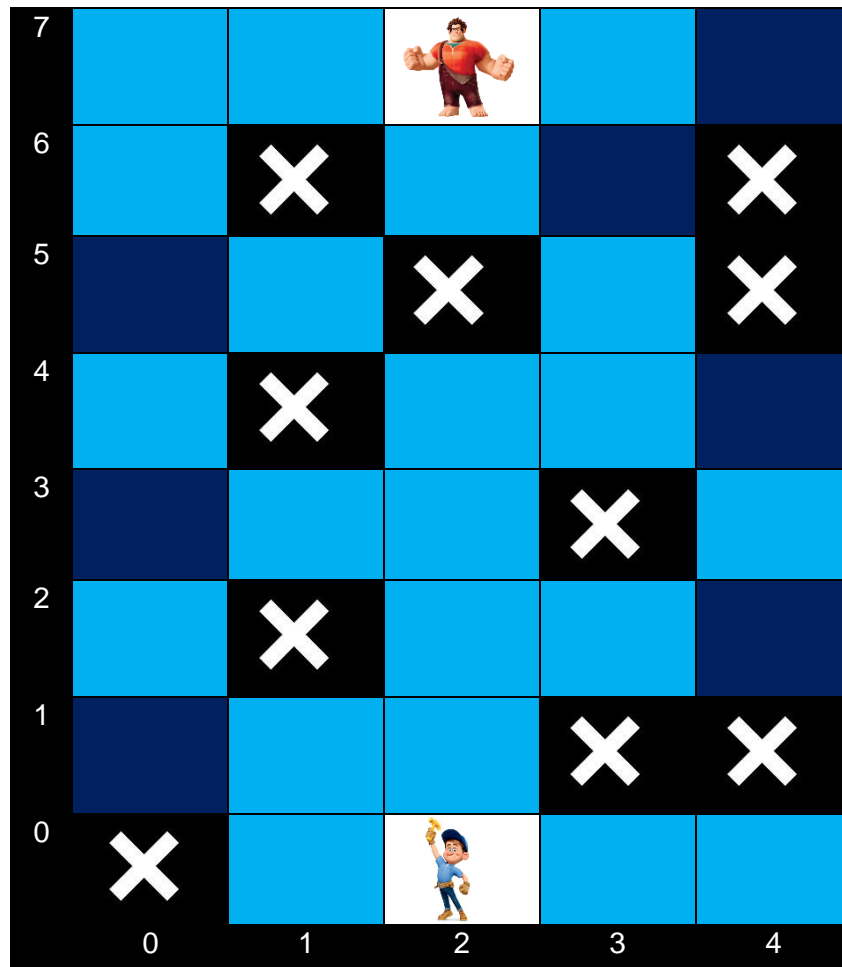
  <Enemigo imagen="C://Imagenes/Enemigo.png">

    <Movimiento PosicionX=2 PosicionY=7>
    <Movimiento PosicionX=4 PosicionY=7>
    <Movimiento PosicionX=4 PosicionY=6>
    <Movimiento PosicionX=1 PosicionY=6>
    <Movimiento PosicionX=2 PosicionY=5>

  </Enemigo>

</Nivel>
```

Al ejecutar las instrucciones anteriores en el editor, se debe de mostrar una interfaz similar a la siguiente:



En la imagen anterior podemos observar que la etiqueta Bloque está representada por los bloques negros que tienen una X, simbolizando que este bloque está dañado. Las escaleras están representadas en azul, simbolizando que es el único espacio donde el personaje principal puede subir o bajar entre los niveles del edificio.

Funcionalidad del juego:

El juego ¡Repáralo! Debe tener las siguientes funcionalidades:

- El personaje debe poder moverse entre los bloques, pueden hacerlo con el teclado físico o con botones dentro de la aplicación, queda a criterio del estudiante.
- El personaje puede subir o bajar entre los niveles del edificio solo en aquellos bloques que estén definidos dentro de la etiqueta escalera.
- Se debe poder diferenciar entre un bloque dañado, un bloque escalera y un bloque normal (con una imagen, color, etc).
- Se debe de poder reparar los bloques al pulsar un botón, cada reparación sumará 100 puntos.
- El enemigo debe de moverse según los movimientos definidos dentro del bloque movimiento, cada 3 segundos.
- Cuando se llegue al último movimiento, deberá iniciar en el primer movimiento, es decir, tendrá un comportamiento cíclico.

- Cada vez que el enemigo llegue a la posición descrita en el movimiento, este bloque pasa a ser un bloque dañado.
- Si el personaje y el enemigo se tocan en una posición, éste perderá la partida, debe mostrar un mensaje indicando que se ha perdido el juego y la puntuación total.
- El juego se gana cuando el personaje repara todos los bloques, debe mostrar un mensaje indicando que se ha ganado el juego y la puntuación total. Al momento de ganar, pasa al siguiente nivel por defecto.
- Se debe mostrar en la aplicación en todo momento: nombre, número del nivel, punteo, y el número de fila y columna del tablero.

Aplicación:

La aplicación debe analizar el texto del editor, de encontrar errores léxicos, la aplicación debe informar al usuario y mostrar un reporte de errores en formato tabla en una nueva ventana, en este caso no debe mostrar la ventana del juego. En la tabla con el reporte de errores se deben mostrar los siguientes campos:

No.	Error	Descripción	Fila	Columna
1	%	Elemento léxico desconocido	3	4

De no encontrar error alguno, se debe generar la ventana de salida correspondiente. La aplicación debe pintar dentro del editor cada uno de los lexemas, según su tipo, los colores se definen a continuación:

Tipo	Color
Palabras reservadas	Celeste
Identificadores	Amarillo
Números	Rojo
Cadenas	Verde
Menor que y mayor que	Anaranjado
Diagonal e igual	Morado
Otros	Negro

Además de colorear los elementos léxicos y generar la ventana del juego, se deberá poder acceder mediante un botón a una nueva ventana que contenga una Tabla de tokens con los siguientes campos:

No.	Token	Lexema	Tipo	Fila	Columna
1	5	<	Menor que	2	1
2	19	Bloque	Palabra reservada	2	12
3	8	=	Igual	2	11

Interfaz gráfica:

Componentes mínimos de la interfaz:

1. **Editor de texto:** Debe ser un área de texto en el cual se puedan escribir nuevas instrucciones de entrada o bien cargar texto desde archivos con extensión .rep
2. **Abrir:** La interfaz debe proveer la capacidad de abrir archivos con extensión .rep.
3. **Guardar:** La aplicación debe proveer la capacidad de guardar el texto contenido en el editor. Si no ha sido guardada nunca, debe proveer una opción para ingresar el nombre del nuevo archivo y la ubicación en la que se desea guardar.
4. **Guardar como:** Está opción permite guardar el archivo de entrada con otro nombre, se debe preguntar el nombre del nuevo archivo.
5. **Analizar:** Debe realizar el análisis léxico del lenguaje que se encuentra actualmente en el editor y generar la salida correspondiente.
6. **Tabla de Tokens:** Debe generar una ventana donde se visualice la tabla de tokens.
7. **Errores:** En caso de existir algún error deberá generar una ventana donde se visualice la tabla de errores.
8. **Botones:** Deberá tener los botones necesarios para la ejecución del juego.
9. **Acerca de:** Debe desplegar una ventana con los datos del estudiante y del curso.
10. **Salir:** Debe terminar la ejecución de la aplicación.

Entregables:

- Manual de Usuario
- Manual Técnico:
 - Debe incluirse el autómata finito determinista que se utilizó para la implementación del analizador léxico. Para la obtención del autómata se debe de utilizar expresiones regulares y el método del árbol, los cuales también deben de adjuntarse en este documento.
 - Plataforma de ejecución
 - Diagrama o diccionario de clases
 - Herramientas utilizadas
- Código fuente.
- Ejecutable del proyecto.

Documentación a entregar de forma física el día de la calificación:

- Hoja de calificación (Original y una copia)

Notas importantes:

- El proyecto se debe desarrollar de forma individual.
- Este proyecto se deberá desarrollar utilizando JAVA con Netbeans.
- El proceso de obtener tokens, se debe hacer a través de la implementación del autómata finito determinista desarrollado por el propio estudiante a través del método del árbol y expresiones regulares.
- La calificación se realizará **ÚNICAMENTE** con los archivos que el auxiliar provea en el momento de la calificación, dichos archivos no se deben modificar.
- La calificación del proyecto será personal y durará como máximo 30 minutos, en un horario que posteriormente será establecido. Se debe tomar en cuenta que durante la calificación no podrán estar terceras personas alrededor, de lo contrario no se calificará el proyecto.

- No se dará prórroga para la entrega del proyecto.
- Queda prohibido el uso de herramientas para la creación del análisis léxico.
- **Copia parcial o total del proyecto tendrá una nota de 0 puntos, y se notificará a la Escuela de Sistemas para que se apliquen las sanciones correspondientes.**
- **En el caso de no cumplir con alguna de las indicaciones antes mencionadas, NO se calificará el proyecto; por lo cual, se tendrá una nota de cero puntos.**

Fecha de entrega 19 de Junio del 2019