



## Practica No. 1

---

### Objetivos:

#### General

- El estudiante aplique los conceptos sobre la fase de análisis léxico de un compilador en una solución de software.

#### Específicos

- Que el estudiante desarrolle métodos propios de lectura de archivos, carácter por carácter, que utilizara en la posterior programación de analizadores léxicos.
- El estudiante refuerce los conocimientos del método del árbol, expresiones regulares y autómatas finitos deterministas.
- Identificar y programar el proceso de reconocimiento de lexemas mediante el uso de autómatas finitos deterministas.

### Introducción:

Tetris es un videojuego de puzzle. El nombre Tetris deriva del prefijo “tetra” (cuatro) en mención a la cantidad de cuadros que componen cada pieza del juego.

La mecánica del Tetris es muy sencilla, desde la parte superior de la pantalla comienzan a caer distintas piezas de diferentes formas, las cuales son figuras geométricas compuestas por cuatro bloques cuadrados unidos de forma ortogonal. No existen reglas en cuanto a las dimensiones del área de juego, en la versión original el área es de 20X10 aunque estas varían en cada versión del juego.

El jugador no puede parar la caída de las piezas, pero puede decidir la rotación de la pieza (0, 90, 180, o 270 grados) y en qué lugar debe caer. Cuando una línea horizontal se completa, esa línea desaparece y todas las piezas que están por encima descienden una posición, liberando espacio de juego y por tanto facilitando la tarea de situar nuevas piezas. La caída de las piezas se acelera progresivamente. El juego acaba cuando las piezas se amontonan hasta llegar a lo más alto, impidiendo la colocación de más piezas dentro del tablero de juego y finalizando el juego. Existen distintas versiones del juego entre las cuales varía tanto las dimensiones del tablero como la cantidad y formas de las piezas. La versión original cuenta con siete piezas diferentes, las cuales se detallan más adelante.

## Descripción:

La práctica consiste en el desarrollo de un juego de tetris, el cual será creado a partir de 2 archivos de entrada. El juego constara de M ( $3 \leq M \leq 10$ ) niveles, los cuales vendrán especificados dentro de uno de los archivos de entrada. Las reglas del juego serán las mismas que para un juego normal de tetris, en el cual se busca completar líneas horizontales para acumular puntos:

- En el momento que una línea horizontal es completada, esta desaparece y las piezas que se encuentran sobre esta defenderán una posición.
- Las piezas pueden ser rotadas 0, 90, 180 o 270 grados.
- Las piezas pueden colocarse en cualquier posición, a lo ancho del tablero, para obtener la mejor jugada.

## Archivos de entrada

### Definición de lenguaje

#### Comentarios

Será permitido el manejo de comentarios de una o más líneas, pueden ser de 2 formas:

- Comentario de una sola línea:  
// Este es un comentario
- Comentarios multilínea :  
<!  
Este es un comentario  
multilínea  
!>

#### Archivo 1

El primer archivo de entrada, el cual tendrá la extensión .trs, contendrá la información necesaria para la creación del tablero de juego y cada uno de los niveles. La estructura de este archivo es la siguiente:


























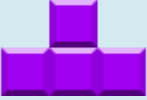


```
M
NxP Nombre
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
```

Dónde:

- M representa un numero entero positivo  $\geq 3$  y  $\leq 10$  que indica la cantidad de niveles que el archivo trae.
- NxP representa las dimensiones del nivel en cuestión, done N es el número de líneas y P el número de columnas.
- Nombre representa un identificador con el nombre del nivel.
- Luego se muestran "N" líneas con "P" caracteres cada una que representan el estado inicial del nivel, donde el numeral (#) representa una casilla libre y el asterisco (\*) representa una casilla ocupada por un bloque.

## Archivo 2

El segundo archivo tendrá la extensión .pzs, y contendrá un listado de las piezas que irán apareciendo en los diferentes niveles del juego. Las piezas que formaran parte del juego se representaran por medio de letras, las cuales se detallas a continuación:

LETRA	PIEZA QUE REPRESENTA			
	^	v	<	>
I				
J				
L				
O				
S				
Z				
T				

La estructura de este archivo es la siguiente:

L, >  
S, v  
Z, ^  
O, <  
L, <  
L, >  
T, v

En cada línea del archivo vendrá una pareja de caracteres, separados por comas, en donde:

- Primer carácter: es la letra que representa a la figura dentro del juego.
- Segundo carácter: representa la rotación con la que esta figura aparecerá.

## Forma de juego

El juego seguirá las siguientes reglas y comportamiento:

1. Las piezas aparecerán una a una conforme se fueron leyendo del archivo de entrada y en la rotación indicada.
2. Las piezas ya dentro del juego podrán ser rotadas y movidas a lo largo del tablero, para obtener la mejor jugada posible.
3. No es necesario animar la caída de las piezas, estas pueden aparecer en la parte superior del tablero y allí ser rotadas y movidas, para cuando ya se esté listo para dejarla caer indicárselo al programa y colocar la pieza en el espacio correspondiente. Se darán puntos extras a las personas que animen la caída de las piezas y que la velocidad de caída aumente en cada uno de los niveles.
4. Al formar una línea horizontal el usuario acumulara 10 pts. Si se forman dos líneas se obtendrán 15 pts, si se forman 3 o más filas se obtendrán 20 pts.
5. El primero nivel del juego se gana al acumular 100 o más puntos. Y cada nivel siguiente se ganara con un puntaje igual o superior al doble más 100 del puntaje necesario para aprobar el nivel anterior. Por ejemplo: si el nivel 1 se gana con 100 pts o más puntos, el nivel 2 se deberá ganar con  $2 * \text{nivel}1 + 100 = 300$  pts o más, y así sucesivamente.
6. El juego se pierde cuando una de las piezas sobrepasa la última fila del tablero. Cuando un usuario pierda el juego el programa se lo indicara, reiniciara el juego en el nivel actual y luego el usuario podrá seguir jugando.
7. Las piezas tendrán un comportamiento cíclico, esto significa que si se llega a la última pieza indicada en el archivo y el juego aún no ha terminado, se retomara la lista de piezas a partir de la primera que se encuentra en el archivo de entrada.
8. Al completar todos los niveles del juego el programa se lo indicara al usuario mostrando su puntuación total, y luego lo regresara al nivel 1.

## Menús de la aplicación

**Menú Archivo:** Este menú contendrá los siguientes submenús

- **Abrir archivo 1 (.trs):** Abrirá un archivo con los niveles del juego ingresando u obteniendo su ruta específica, filtrando únicamente los archivos con extensión .trs.
- **Abrir archivo 2 (.pzs):** Abrirá un archivo con las piezas que se utilizaran en el juego ingresando u obteniendo su ruta específica, filtrando únicamente los archivos con extensión .pzs.
- **Salir:** Opción con la cual se cierra la aplicación.

**Menú Juego:** Este menú contendrá las herramientas que se utilizaran para la comparación del texto presente en cada uno de los archivos abiertos.

- **Analizar Archivo 1:** analizará el archivo abierto y cargará a memoria cada uno de los niveles dentro del archivo.
- **Analizar archivo 2:** analizará el archivo abierto y cargará a memoria el listado de piezas que se utilizaran en el juego.
- **Iniciar juego:** dará inicio al juego a partir de los archivos cargados anteriormente. Si uno de los archivos no ha sido cargado el programa debe indicarlo al usuario, impidiendo iniciar un juego nuevo.

**Menú Ayuda:** Contendrá los Submenús:

- **Manual de la aplicación:** Este submenú tendrá como objeto servir al usuario como ayuda, acerca de cómo utilizar la aplicación. Dicho submenú debe de tener un enlace directo hacia el manual de usuario el cual debe abrirse en una ventana externa en formato PDF.
- **Acerca de:** Este submenú debe de mostrar la versión de la aplicación así como los datos del desarrollador
  - Nombre completo
  - Número de carné

## Documentación a Entregar:

- Manual de Usuario, Manual Técnico.
- Ejecutable.
- Código fuente de la práctica.

## Notas importantes:

- Se debe desarrollar de forma individual.
- Debe desarrollarlo utilizando JAVA.
- Queda prohibida la utilización de cualquier herramienta que no sea para la lectura y/o escritura de archivos.
- No es permitido el uso de herramientas para el análisis. **Deben de construir el analizador por medio del método del árbol.**
- Está prohibida la utilización del comando Split, de ser utilizado la practica quedara anulada.
- Copia parcial o total del proyecto tendrá una nota de 0 puntos y se notificará a la escuela para que se apliquen las sanciones correspondientes.
- Está prohibido bajar código de internet.
- No hay Prorroga.

## Modo de entrega:

1. La entrega será virtual, se debe utilizar GitHub <https://github.com/>, deben colocar el repositorio de manera privada y dar permiso al auxiliar [fernandopaz1995@gmail.com](mailto:fernandopaz1995@gmail.com) .
2. Además La entrega se realiza mediante un archivo comprimido ZIP con el nombre: [OLC2]P1\_carné.zip en el entregable en la plataforma UEDI
3. La calificación se realizará de manera virtual (ya sea en meet o zoom) con las camaras activadas, cada calificación será almacenada.

## Fecha de Entrega:

**MIÉRCOLES 10 de JUNIO de 2020**

## Ejemplo de archivo de entrada:

3

20x10 Nivel\_Alpha

[illegible]

15x10 Nivel\_Beta

[illegible]

18x10 Nivel\_teta

[illegible]