

Un punto muy importante a recalcar es que el editor de texto debe de ser capaz de analizar el lenguaje presentado y que el resultado presentado sea su tabla de tokens respectiva y su grafica que representa lo planteado en el texto, lo cual será las definiciones necesarias para crear un Diagrama de Clases basada en la notación que describe UML.

Descripción del Lenguaje:

El lenguaje soportara la definición de clases, sus atributos, métodos y asociaciones descritas de la siguiente manera:

- **Bloque Clase:**

```
[Clase]
{
    [Nombre] = <identificador>;
    <BloqueAtributos>
    <BloqueMetodos>
}
```

En donde:

- **Clase:** Sera una palabra reservada que iniciara la descripción de un objeto clase, para luego diagramarlo. Este bloque posee lo siguiente:
 - **Nombre:** Palabra reservada seguida de un “identificador”, la cual será el nombre de la clase a crear.
 - **Bloque Atributos:** Este bloque representa todos los atributos que puede contener una clase.
NOTA: Este bloque puede o no venir (si este bloque no está incluido en una clase quiere decir que la clase representa una interfaz).
 - **Bloque Métodos:** A diferencia del Bloque Atributos este bloque **siempre** debe de venir y representará todos los métodos que tendrá una clase.

Se debe de tener en cuenta que los atributos no siempre vendrán en este orden, por lo que se debe de analizar cuál será la mejor manera de guardar la información.

- **Bloque Atributos:**

```
[Atributos] {  
    <Visibilidad> <Identificador> : <Tipo> ;  
    <Visibilidad> <Identificador> : <Tipo> ;  
    <Visibilidad> <Identificador>;  
    ...  
}
```

En donde:

- **Visibilidad:** Representa la visibilidad que tiene un atributo dentro de la misma clase y existe la posibilidad que sea cualquiera de las siguientes 3:
 - Público (+)
 - Privado (-)
 - Protegido (#)
- **Identificador:** Sera el nombre que recibirá el atributo a definir.
- **Tipo:** Definirá el tipo que poseerá el atributo, el cual está definido por un <Identificador>. Este parámetro puede o no venir.

- **Bloque Métodos:**

```
[Metodos] {  
    <Visibilidad> <Identificador> : <TipoRetorno> ;  
    <Visibilidad> <Identificador> : <TipoRetorno> ;  
    <Visibilidad> <Identificador> ;  
    ...  
}
```

En donde:

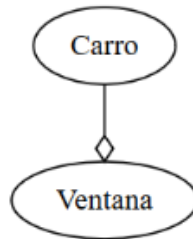
- **Visibilidad:** Representa la visibilidad que tiene un método dentro de la misma clase y existe la posibilidad que sea cualquiera de las siguientes 3:
 - Público (+)
 - Privado (-)
 - Protegido (#)
- **Identificador:** Sera el nombre que recibirá el método a definir.
- **TipoRetorno:** Definirá el tipo de retorno que poseerá el método, el cual está definido por un <Identificador>. Este parámetro puede o no venir.

- **Bloque Asociación:**

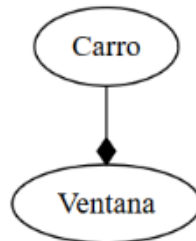
```
[Asociacion] {  
    <Identificador> : <TipoAsociación> : <Identificador>;  
    <Identificador> : <TipoAsociación> : <Identificador>;  
    ...  
}
```

En donde:

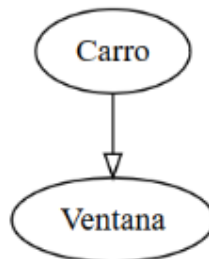
- **Identificador:** Sera el nombre de una clase que se quiere asociar.
- **Tipo Asociación:** Definirá la asociación que se desea representar entre clases y puede ser una de las siguientes opciones:
 - **Agregación**



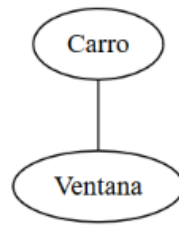
- **Composición**



- **Herencia**



- **AsociaciónSimple**



Aplicación:

Menú Editor de texto:

- **Menú Archivo:** Este menú contendrá los siguientes submenús
 - **Abrir:** Abrirá un archivo desde una ruta específica filtrando únicamente los archivos con extensión “lfp”.
 - **Guardar:** Permite guardar un archivo en una ruta específica en caso de que este no se haya guardado previamente, de lo contrario únicamente guarda los cambios que se hayan realizado si el archivo ha sido abierto de una ruta específica.
 - **Guardar Como:** Permite guardar todo lo que contenga en un momento determinado el editor de texto en un archivo nuevo, cabe mencionar que el archivo a guardar también debe de ser guardado con la extensión “lfp”.
 - **Salir:** Opción con la cual se cierra la aplicación.
- **Menú Analizar:**
 - **Analizar Léxico:** Analizara lexicamente lo que posea actualmente el editor de texto.
- **Menú Reportes:**
 - **Reporte:** Mostrar una tabla de tokens como reporte de no existir ningún error en la entrada. Estos reportes se deben de mostrar en una página html.
De existir un error debe de mostrar un reporte de errores encontrados. Estos reportes se deben de abrir automáticamente.

Ejemplo Reporte Tokens:

No	Lexema	Tipo	Columna	Línea
1	Asociación	Reservada	1	5
2	MiClase	Identificador	4	10

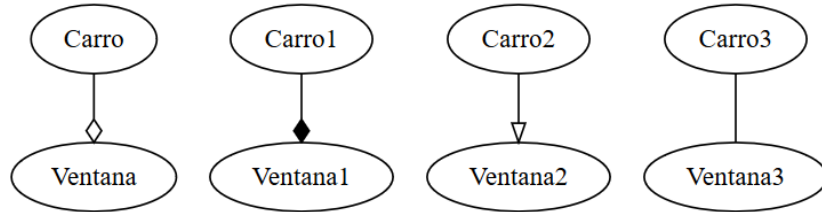
Ejemplo Reporte Error:

No	Error	Columna	Línea
1	@	13	5
2	#	4	12

- **Diagramar:** Mostrara una imagen generada mediante graphviz de no haber ningún error.
De existir un error debe de mostrar un mensaje de advertencia y no deberá de generar ninguna imagen.

Ejemplo Codigo Graphviz:

```
digraph G {
    "Carro" -> "Ventana" [arrowhead="odiamond"]
    "Carro1" -> "Ventana1" [arrowhead="diamond"]
    "Carro2" -> "Ventana2" [arrowhead="onormal"]
    "Carro3" -> "Ventana3" [arrowhead="none"]
}
```



- **Menú Ayuda:** Contendrá los Submenús:
 - **Manual de la aplicación:** Este submenú debe de tener un enlace directo hacia el manual de usuario y tecnico 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

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

Lo único que se deberá de entregar impreso es la hoja de calificación.

Notas importantes:

- Se debe desarrollar de forma individual.
- Debe desarrollarlo utilizando Visual Basic .Net 2013 superior.
- Queda prohibida la utilización de cualquier herramienta que no sea para la lectura y/o escritura de archivos.
- 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.
- No hay Prorroga.
- El autómata deberá de ser realizado e implementado **por el propio estudiante**.
- De encontrar que utilizan una herramienta para analizar texto se le colocará una nota 0.

Fecha de Entrega:

Viernes 7 de septiembre 2018