1) Write a Java program which parses the grammatical rules from a text file, then uses the parsed data to generate a random poem. Here is an example of what the output might look like:

my sun among her white meadow

moon upon my light

moon

your rain climbs

her murky bright clear willow

R/

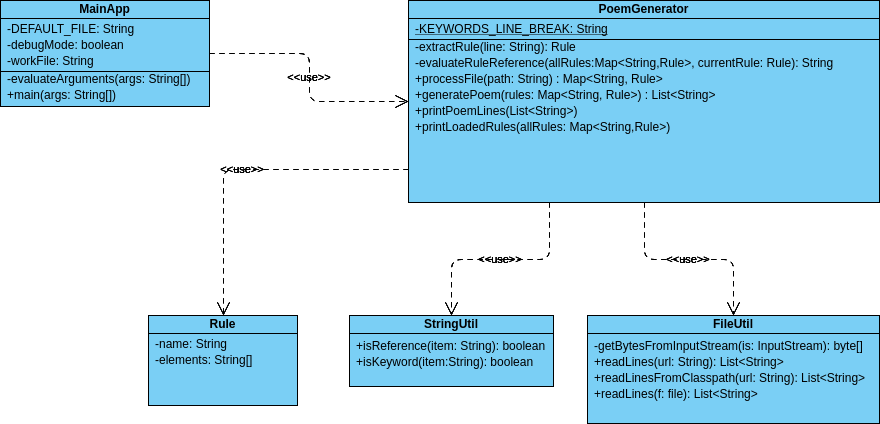
Se sube el código fuente al repositorio:

[**https://github.com/**](https://github.com/stedennis90/Poem-Generator)

En la carpeta /dist se encuentra la versión compilada del código fuente.

2) Create a class diagram which describes the design used in the implementation of 1.

R/



3) Explain two cases where you learned something through experience (e.g. solving a production issue) and could not have learned about that problem through theoretical or reading through technical documentation.

R/

1. En una estabilización de un producto puesto en ambiente productivo 3 días antes, se encontró que al iniciar sesiones los diferentes usuarios y ver su pantalla de bienvenida con un listado de documentos registrados, no mostraba los de su usuario, sino que mostraba otros o ninguno; lo anterior generaba un reproceso por parte del usuario ya que al ver documentos erroneos los borraba y volvía a registrar los suyos, pero en realidad lo que hacia era eliminar los documentos de otra persona y se generaba así una cadena de perdida de documentos.

Analizando el error se encontró que al haber concurrencia de mínimo 10 usuarios en la carga de la página principal se generaba un cruce de información con los mismos.

Se evaluó la situación y se determinó que había que controlar la concurrencia manualmente a través de la palabra reservada synchronized ya que el servidor de aplicaciones Tomcat 5.5 no estaba manteniendo la independencia de los servlets a nivel de hilos de ejecución.

Esta condición se habia detectado en un proceso de descarga simultanea de archivos, pero nunca en una carga de datos para mostrar al usuario. Como la tarea de carga tenía una duración bastante alta, se veía comprometida la sincronización de la respuesta.

2. Las configuraciones de ambientes o despliegues en diferentes servidores de aplicaciones. Este tipo de configuraciones o instalaciones poseen normalmente una guía o manual que en la mayoría de los casos resulta ser inexacto por detalles que se presentan de acuerdo con las dependencias de ciertas librerías.

4) Given the choice between writing ideal code and meeting a critical deadline, which would you choose and why?

R/

Siempre se debe velar por escribir código ideal.

Esto siempre garantiza que no habrá un reproceso de información, codificación y por lo tanto impacto a gran escala dentro de la aplicación en un mediano o largo plazo.

Adicional a lo anterior, también hay que mencionar que los desarrolladores debemos codificar para desarrolladores y no para máquinas, ya que este tipo de buenas prácitcas facilitan el proceso de entendimiento y adaptación de otro desarrollador que tenga que dar soporte a un código legado.