Práctica Análisis Estático

Metodologías de Desarrollo Seguro, Ingeniería de la Ciberseguridad

Carlos Barahona Pastor y Ángel del Castillo González, Grupo Q

Índice

Expresiones regulares

Ejercicio 1.

Enunciado

Dado un texto de una sola línea, determinar todos los años (números formados estrictamente por 4 dígitos) que aparecen. Un año es una cadena numérica de longitud 4, con cualquier valor en el rango [0000, 9999]. Se tendran que imprimir por pantalla en el lenguaje deseado usando una expresión regular todos los años que vienen en el texto en orden de aparición, en una línea cada uno.

Resolución

Ejercicio 2.

Enunciado

Dado un texto de una línea, determinar todas las matrículas que aparecen. Una matrícula es una cadena que tiene las siguientes características: - 4 dígitos seguidos de un separador (guion, espacio o nada) y 3 letras en mayusculas al final tal que [0000 - AAA, . . . , 9999 - ZZZ]. - Las matr´ıculas pueden llevar una E mayuscula delante para indicar que se trata de un vehículo especial. Ejemplos: E1337ZZZ, E-0000 PCB. - Los dígitos pueden estar separadas de las letras utilizando un guion, un espacio, o ningun separador. Se tendran que imprimir por pantalla usando el lenguaje elegido usando una expresión regular todas las matrículas que vienen en el texto en orden de aparición

Resolución

Ejercicio 3.

Enunciado

Dado un formato de fechas yyyy-mm-dd, se pide convertir a dd.mm.yyyy. Para cada match encontrado en los documentos propuestos se tendra que imprimir en el siguiente formato (los rangos de fecha puede ser erroneos, pueden existir un mes 20).

Resolución

Ejercicio 4.

Enunciado

Dado un texto, determinar cuando se ha encontrado un email de alumno de nuestra universidad “@alumnos.urjc.es” o profesor “@urjc.es”. Los emails de alumnos estan formados del siguiente patrón (puedes asumir que el input siempre estara en minúcula): - Inicial del usuario, seguido de punto. - Apellido del usuario, siempre mayor o igual a 2 caracteres. - Seguidos de un punto y la fecha de matriculacion. ´ - todos finalizan con “@alumnos.urjc.es”.

Los correos de los profesores constan de: - Nombre del profesor seguido de un punto. - Apellido del profesor. - Finalizando con “@urjc.es”.

Para cada match encontrado se tendra que imprimir en el siguiente formato. - Para el caso de prueba i.lozanoo.2015@alumnos.urjc.es reportaremos “alumno lozanoo matriculado en 2015” - Para un profesor reportaremos para el ejemplo isaac.lozano@urjc.es “profesor isaac apellido lozano”

Resolución

Ejercicio 5.

Enunciado

Dado un texto devolver las direcciones postales. Una direccion estará compuesta de una calle representada por “C/” o ”Calle” seguido de un espacio con el nombre de la calle (una sola palabra) donde la primera letra debe estar en mayuscula, opcionalmente una coma, un numero arbitrario de espacios, el número en cualquiera de los siguientes formatos (Nº7, nº7, N 7, n 7, 7, Nº 7, nº 7, n7, N7). No es valido Nº7, n º7, ni º7, la N podría estar en mayúculas o minúculas. Seguido, una coma, un número arbitrario de espaciosvy un numero de 5 dígitos correspondiente a un codigo postal. Los nombres de las calles deben poder validar calles de Madrid formadas por una sola palabra, no es necesario que reconozca “Calle Almendro Azul”, porque el nombre de la calle tiene dos palabras en vez de una. Ejemplos de casos: - “C/ Dulcinea Nº 10, 28936” - “Calle Dulcinea 10, 28106” - “Calle Dulcinea N10, 28091” Para cada calle encontrada se reportara: “CP-Calle-Numero”, por ejemplo: “28926-Dulcinea-10”.

Resolución

Ejercicio 6.

Enunciado

Dado un fichero de logs, transformar cada línea a CSV extrayendo la siguiente informacion: - Nivel de log - Hilo donde se ha producido el log (este hilo se corresponde con letras en mayúscula, minúscula y de números) - Clase responsable de emitir el log - Mensaje de log

Resolución

Integración Continua

Preguntas Puesta en marcha del proceso análisis automático

1. ¿Cuantas vulnerabilidades, bugs y code smells ha detectado SonarCloud en el proyecto entero?

Estos son los datos que ha arrojado SonarCloud:

- Vulnerabilidades: 30
- Bugs: 32
- Code smells: 565

2. Haz click sobre el proyecto para abrir la vista de detalle. Revisa las vulnerabilidades detectadas y la lista de Security Hotspots. ¿Qué diferencia crees que existe entre las vulnerabilidades y los Security Hotspots?

La diferencia es que las **vulnerabilidades** hacen referencia a código que se ha detectado que puede ser explotado, mientras que los **Security Hotspots** son fragmentos de código que deben ser revisados, ya que aunque no se haya podido determinar a priori si son una vulnerabilidad o no en el escaneo, podrían serlo por lo que deben analizarse de forma manual para determinarlo.

3. Haz cualquier commit para forzar un reanalisis (por ejemplo editando el README.md). Espera a que finalice y vuelve a Sonar. ¿Cual es el estado del proyecto (Passed/Failed)?. ¿A qué crees que se debe? ¿Crees que el numero de vulnerabilidades afecta a dicho veredicto?

Tras editar el archivo README.md, el estado del proyecto es `Failed`. Esto se debe a que para que el estado de un proyecto sea `Passed`, se tienen que cumplir una serie de condiciones de la `Quality Gate`. En este caso específico, las condiciones que no se cumplen son:

- `Reliability Rating`: este apartado hace referencia al número de bugs (errores de código que hacen que no funciones) que se han detectado en el código analizado. En el caso de esta métrica se requiere que el valor no sea menor que `A` y el valor que ha arrojado el análisis del código es de `E` (19 bugs)
- `Security Rating`: este apartado hace referencia al número de vulnerabilidades (código que puede ser explotado por adversarios) que se han detectado. Esta métrica requiere un valor que no sea menor que `A`, y en este caso el valor obtenido es `E` (27 vulnerabilidades).

Efectivamente el número de vulnerabilidades afecta a dicho veredicto, ya que el hecho de que existan vulnerabilidades hacen que no se pueda obtener una puntuación de `A` (la cual se corresponde con 0 vulnerabilidades). Con los bugs pasa exactamente lo mismo, la puntuación de `A` se obtiene con 0 bugs.

Preguntas Mitigación

1. Elige una vulnerabilidad de tipo Blocker o Critical, explica cual es la vulnerabilidad detectada, por qué ha sido detectada (y si realmente es una vulnerabilidad y no un falso positivo).

2. Propon una solucion e impleméntela en una nueva rama del repositorio. Explica los cambios que arreglan dicha vulnerabilidad.

3. Crea una Pull Request de la nueva rama a la rama principal. ¿Qué opina Sonar de los cambios realizados?

4. Junta (merge) la nueva rama con la rama principal. Una vez finalizado el analisis, ¿qué cambios se han producido en el proyecto? ¿Cuantas vulnerabilidades detecta ahora?