

Tutorial para aprender a usar la herramienta PMD

Preparación del entorno:

Estos directorios se encuentran tanto en el repositorio remoto de GitHub “TallerCalidadPMD”, como en la entrega.

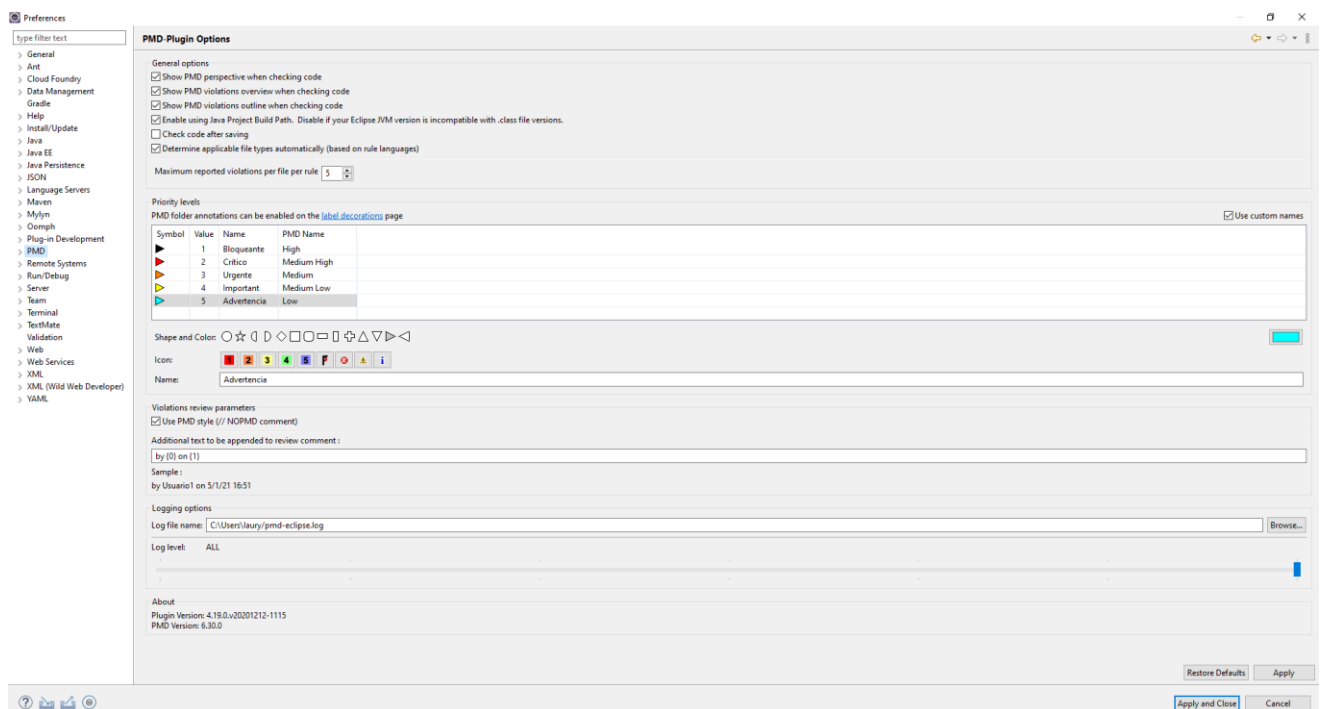
1. Obtener las herramientas y realizar los pasos contenidos en el archivo “Requisitos_para_realizar_el_taller.pdf” del directorio “Requisitos_para_realizar_el_taller”.
2. Ejecutar Eclipse e importar el archivo “CodigoParaTutorial.zip” del directorio “Otro_material”. Para esto es necesario seleccionar en la barra de herramientas de Eclipse File => Import... => General => Projects from Folder or Archive (seleccionando la opción con doble click) => Archive (en la parte superior derecha de la ventana) => Seleccionar en el directorio indicado el archivo indicado => En la ventana aparecerán solo dos elementos, deseleccionar el que no incluya en su ruta “ProgramaBasicoConErrores” => Seleccionar el botón Finish.
3. Para añadir el paquete fundamentos.jar necesario para este pequeño programa:
 - a. Seleccionar con botón derecho el proyecto, este aparecerá en la parte superior izquierda de Eclipse con el nombre “ProgramaBasicoConErrores”.
 - b. Seleccionar Build Path => Configure Build Path... => En la barra horizontal seleccionar Libraries => Seleccionar en la parte derecha de la ventana el botón Add External JARs... => buscar el archivo fundamentos.jar en el directorio “Otro_material” => Seleccionar botón inferior de la ventana “Apply and Close”.

Observación del funcionamiento de PMD:

1. Seleccionar en la barra de herramientas Window => Preferences => PMD.
 - a. En el apartado “General options” podemos seleccionar la configuración que nos resulte más conveniente, seleccionamos los checkbox de las opciones “Show PMD violations overview when checking code” y “Show PMD violations outline when checking code”, de esta manera cuando posteriormente comprobemos el código en busca de incumplimientos en las

reglas obtendremos información global de las reglas incumplidas y información más detallada de cada regla. De todas las opciones dejamos sin seleccionar “Check code after saving” ya que queremos comprobar los cambios que hagamos en el código cuando decidamos de forma manual.

- b. En el apartado “Priority levels” podemos distinguir 5 prioridades según la gravedad de la regla incumplida al comprobar el código, están ordenadas de mayor a menor prioridad, numerada de igual forma del 1 al 5. Si seleccionamos un elemento de la lista podemos cambiar el símbolo que le representa, su color y su nombre. Dado que los colores son muy parecidos vamos a cambiar los mismos, y vamos a cambiar los nombres en su traducción al castellano.
- El elemento 1 tendrá color negro y su nombre será “Bloqueante”.
 - El elemento 2 tendrá color rojo y su nombre será “Crítico”.
 - El elemento 3 tendrá color naranja y su nombre será “Urgente”.
 - El elemento 4 tendrá color amarillo y su nombre será “Importante”.
 - El elemento 5 tendrá color azul claro y su nombre será “Advertencia”.



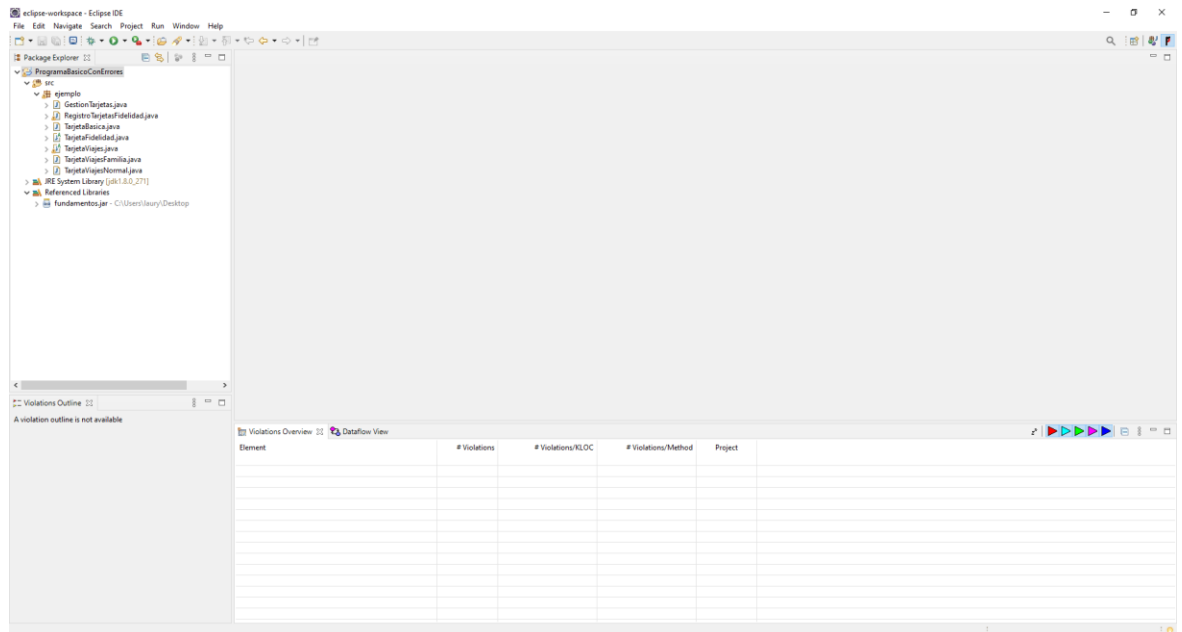
- c. Seleccionamos el botón Apply.

2. En la columna de la izquierda dentro de la venta anterior, Preferences, seleccionamos en el desplegable de PMD => CPC Preferences.
 - a. En el apartado "General Options" viene la etiqueta "Minimum Tile Size" y a su derecha un campo con un 25 por defecto. Este campo indica el número de líneas necesario para que la herramienta detecte código repetido. Vamos a indicar que sean 30 ya que no la herramienta no deja bajar el valor de 25.
 - b. Seleccionamos el botón Apply.
3. En la columna de la izquierda dentro de la venta anterior, Preferences, seleccionamos en el desplegable de PMD => Reports.
 - a. Nos encontramos un listado de formatos, seleccionaremos las casillas de html, json, xml y text. Son los formatos en los que la herramienta nos exportará los incumplimientos de las reglas.
 - b. Al seleccionar cualquiera de los elementos podremos en la parte inferior cambiar las propiedades de algunos de ellos. Ir seleccionando diferentes elementos y ver que parámetros se pueden cambiar.
 - c. Seleccionamos el botón Apply.
4. En la columna de la izquierda dentro de la venta anterior, Preferences, seleccionamos en el desplegable de PMD => Rule Configuration.
 - a. Este es el apartado más importante de la configuración de la herramienta, lo primero que haremos es seleccionar el checkbox de "Use global rule management".
 - b. Podemos ver el listado de todas las reglas que se aplican por defecto sobre el código cuando hacemos una comprobación de su calidad.
 - c. Podemos seleccionar todas o deseleccionarlas, para ello:
 - i. Pulsamos el botón justo a la izquierda del texto "Rules grouped by". Observamos que todo el listado se desmarca. En este caso si comprobásemos el código no saltaría ninguna regla, aunque deberían de saltar, pero las hemos deshabilitado.
 - ii. Si pulsamos el botón a la izquierda del anterior se marcarán todas las reglas, lo que se traduce en que se aplicarán todas en la comprobación.
 - iii. Podemos seleccionar individualmente cada regla seleccionando en sus correspondientes checkboxes.
 - d. Podemos agrupar las reglas según varios criterios, para ello seleccionaremos el desplegable que se encuentra a la derecha del texto "Rules grouped by".
 - i. El criterio de agrupación más interesante es el de "Rule set", podemos encontrar las reglas organizadas según diferentes temáticas, podemos seleccionar y deseleccionar Rule sets enteros manualmente, por si no tenemos interés en aplicar reglas de algún tipo para valorar la calidad de nuestro proyecto.

- ii. Otro criterio de agrupación interesante es el de “Priority” ya que muestra las reglas según la importancia que tienen asignadas.
- e. Podemos seleccionar cualquier regla y obtener información de ella en la parte inferior de la ventana, incluso hacer modificaciones. En la barra horizontal nos encontramos diferentes opciones:
 - i. En Summary nos encontramos información de la regla.
 - ii. En Rule podemos hacer cambios de nombre, prioridad, grupo de reglas a las que pertenece, definición de la regla y lenguaje de programación a la que va dirigida. No cambiaremos ningún valor.
 - iii. En Properties solo para algunas reglas podemos alterar su comportamiento, digamos que podemos hacer variar restricciones dentro de la regla.
- f. A la derecha del listado de reglas tenemos cuatro botones en la parte superior y uno en la inferior.
 - i. El primer botón por arriba sirve para añadir reglas propias. Esta cualidad es realmente interesante porque hace que la herramienta adquiera mucha versatilidad.
 - ii. El segundo botón sirve para eliminar una regla previamente seleccionada o un conjunto de ellas.
 - iii. El tercer botón es para importar un Rule set completo desde un archivo .xml. Es una forma rápida de añadir reglas y permite pasar reglas de una herramienta a otra.
 - iv. El cuarto botón es para exportar reglas para poderlas reutilizar en otra herramienta.
 - v. El botón inferior es una ayuda para diseñar reglas, pero no es muy intuitiva.

Uso de la herramienta PMD:

1. Añadiremos una vista que no está por defecto, es de PMD y es interesante. Seleccionamos en la barra de herramientas Window => Open Perspective => Other... => PMD (doble click), luego Window => Show View => Other... => PMD => Dataflow View (doble click). La vista aparecerá en la parte inferior, la perspectiva debe mostrar las siguientes vistas:



- a. Podemos seleccionar cualquier clase del proyecto con doble click, y seleccionar la vista Dataflow View que está en la parte inferior de Eclipse. Seleccionamos la clase `RegistroTarjetasFidelidad`.
 - i. En la vista podemos seleccionar el método del que queremos saber su complejidad ciclométrica, seleccionamos el método `buscaTarjeta()`.
 - ii. Podemos observar el número de línea a la que hace referencia cada nodo del grafo, y la propia línea de código.
 - iii. Si damos click sobre el nodo que queramos, nos lleva a la línea de código de la clase a la que corresponde. Por ejemplo, seleccionar el nodo 3 y nos llevará al bucle for del método.

2. Para analizar la calidad del código según las reglas predefinidas o creadas debemos seleccionar con click derecho la clase del proyecto que queremos analizar, se puede hacer con el proyecto entero, en este caso se pasarán las reglas por cada clase de la jerarquía de ficheros.
 - a. Seleccionamos con click derecho la clase RegistroTarjetasFidelidad, luego PMD => Check Code.
 - b. La información de pasar las reglas aparecerá en tres lugares:
 - i. La vista Violations Overview (parte inferior de Eclipse) muestra las reglas que se han incumplido y una serie de estadísticas y veces que se ha incumplido cada regla. Al hacer click derecho sobre alguna de ellas se pueden filtrar las reglas por prioridad, el resto de opciones no es tan importante. Si damos doble click sobre una de estas reglas se marcarán las ocurrencias de estas en el código, en la vista Violations Outline que está en la esquina inferior izquierda.
 - ii. En la vista Violations Outline vemos cada vez que se han incumplido las reglas para el código analizado, con el número de línea, la regla que se incumple y un mensaje de error que nos indica que ha ocurrido para incumplir esa regla. Si necesitamos más información seleccionamos una ocurrencia de incumplimiento con click derecho y seleccionamos "Show details...", aquí encontraremos un ejemplo de cómo debe realizarse correctamente la codificación. Al dar click derecho sobre una de las ocurrencias podemos marcar la ocurrencia como vista, eliminar los incumplimientos seleccionados o incluso borrar la regla.
 - iii. Podemos también ver en el código las violaciones de las reglas. Estas aparecerán indicadas en el margen izquierdo del código junto al número identificador de la línea de código, tendrá forma de triángulo, y según el color la gravedad será mayor o menor. Para saber que regla/s ha incumplido la línea marcada y como arreglar el problema hay que colocar el cursor sobre el triángulo y clickar. Quizá esta sea la forma más rápida de detectar el porqué del problema.
 - c. Seleccionamos con click derecho el proyecto entero "ProgramaBasicoConErrores", luego PMD => Check Code. Esto nos analizará todas las clases.
 - d. Damos doble click para abrir la clase TarjetaBasica.
 - e. Colocamos el cursor sobre la línea donde se inicializa la constante del descuento y observamos que se incumple la regla LongVariable, indicando que el nombre de la variable es demasiado largo.
 - f. Cambiamos el nombre de la variable mediante refactorización por DESCUENTO y guardamos.

- g. Click derecho sobre la clase TarjetaBasica, luego PMD => Check Code.
 - h. Al posar el cursor sobre el triángulo indicador de la línea ya no aparece el aviso de LongVariable ya que hemos corregido el problema.
- 3. Con el código bajo análisis se puede generar un documento con la información de todas las infracciones, para lo cual después de analizar el código del proyecto podemos hacer click derecho sobre el proyecto, luego PMD => Generate Reports.
 - a. Ahora en el directorio por defecto que cada usuario tenga seleccionado (el workspace), dentro del directorio correspondiente al proyecto “ProgramaBasicoConErrores”, aparecerá un nuevo directorio llamado “reports”, dentro se encuentra un archivo con la información de las reglas infringidas, “pmd-report.txt”
 - b. Abrir el fichero y observa su contenido.
- 4. Ahora vamos a quitar el análisis de nuestro código, para ello seleccionamos con click derecho nuestro proyecto, luego PMD => Clear Violations. Todo el análisis realizado desaparece, aunque le tenemos documentado en el fichero del directorio reports por la acción anterior.
- 5. Podemos detectar código repetido, que es una mala práctica para la calidad del producto, para ello seleccionamos click derecho sobre el proyecto y luego PMD => Find Suspect Cut and Paste...
 - a. Seleccionamos Java como lenguaje.
 - b. El número de líneas a partir del cual se detectará que es código duplicado. Indicamos que 30 ya que es un número razonable. Si se quiere se puede generar un fichero con la información en el directorio report. Decidimos que si marcando el checkbox.
 - c. Para el formato de salida seleccionamos XML.
 - d. Le damos al botón OK.
 - e. En el directorio report del proyecto se encuentra el fichero cpd-report.xml, abrirlo para observar su contenido y donde se produce la duplicación del código. Los ficheros de report se pueden leer desde eclipse dando doble click sobre ellos en la jerarquía del proyecto.

Para mayor información cabe la posibilidad de seleccionar en la barra de herramientas de Eclipse Help => Help Contents => en la ventana emergente, en el listado de la izquierda “PMD Plugin Documentation” => How to...