



PECL2

Calidad, pruebas y mantenimiento Software –
2020/2021

Adina Murg



Resumen Documentación Ejercicio 1

Objetivo	Ejecutar el código con CKJM, CKJM extendido y CK.
Código	https://github.com/whoisadn/paradigmas-de-programacion-pecl
Entorno necesario	CMD + Eclipse
Conclusiones	Podemos analizar del proyecto gracias a diversas métricas.

CKJM

Es un software que calcula las métricas Orientadas a Objeto de Chidamber y Kemerer gracias al procesado del bytecode. Por cada clase del proyecto, se calculan las 8 métricas siguientes:

WMC	Weighted methods per class	La suma de las complejidades de sus métodos.
DIT	Depth of Inheritance Tree	Niveles de herencia.
NOC	Number of Children	Número de hijos de la clase a analizar.
CBO	Coupling between object classes	Mide el acoplamiento entre las clases.
RFC	Response for a Class	Mide el número de métodos diferentes que pueden ejecutarse cuando un objeto de esa clase recibe un mensaje (cuando se invoca un método para ese objeto).
LCOM	Lack of cohesion in methods	Mide la cohesión.
Ca	Afferent couplings	Similar a CBO, mide el acoplamiento.
NPM	Number of public methods	Cuenta los métodos públicos de una clase.

Desde CMD ejecutamos el siguiente comando (java jar de CKJM y la ubicación de nuestro código con las clases a analizar):

```
C:\> Símbolo del sistema
Exception in thread "main" java.lang.NullPointerException
  at org.apache.bcel.classfile.ClassParser.parse(ClassParser.java:165)
  at gr.spinellis.ckjm.MetricsFilter.processClass(Unknown Source)
  at gr.spinellis.ckjm.MetricsFilter.main(Unknown Source)

C:\>
C:\>
C:\>java -jar \ckjm-1.9\build\ckjm-1.9.jar C:\Users\QuasarPC\Documents\PECL2020\build\classes\PECL\*.class
PECL.Cliente 3 2 0 2 15 0 2 3
PECL.Restaurante$1 2 1 0 1 4 0 1 1
PECL.Restaurante$2 2 1 0 1 4 0 1 1
PECL.Cocinero 2 2 0 2 15 0 1 2
PECL.Restaurante$3 2 1 0 1 4 0 1 1
PECL.Pausar 1 1 0 0 2 0 0 1
PECL.Restaurante$10 2 1 0 1 5 1 1 1
PECL.Restaurante$8 2 1 0 1 4 0 1 1
PECL.Restaurante$9 2 1 0 1 4 0 1 1
PECL.Restaurante$4 2 1 0 1 4 0 1 1
PECL.Restaurante$5 2 1 0 1 4 0 1 1
PECL.Restaurante$6 2 1 0 1 4 0 1 1
PECL.Restaurante$7 2 1 0 1 4 0 1 1
PECL.Log 3 1 0 0 22 1 4 3
PECL.Pedido 2 1 0 1 10 0 0 2
PECL.MostradorPedidos 4 1 0 0 17 0 4 4
PECL.Restaurante 21 6 0 16 95 208 10 2
PECL.Empleado 2 2 0 3 16 0 1 2
PECL.Main 2 1 0 1 5 1 0 2
PECL.MesaPlatos 5 1 0 0 18 0 3 5
C:\>
```

Con esto, por ejemplo podemos analizar la clase MesaPlatos:

- **WMC:** los métodos de la clase suma una complejidad de 5 (la más alta de todo el proyecto excluyendo la clase Restaurante que es el main)
- **DIT:** está a nivel 1 de herencia, lo que nos indica que esta clase tiene un parent.
- **NOC:** Esta clase no tiene hijos que hereden de ella.
- **CBO:** Tiene un índice de acoplamiento bajo, de 0.
- **RFC:** 18
- **LCOM:** cohesión con valor 0.
- **Ca:** acoplamiento 3
- **NPM:** 5 métodos públicos en la clase.

CKJM Extendido

A continuación y de manera muy similar a la anterior, ejecutamos el código con CKJM extendido, que expande el número de métricas, como la AMC (complejidad media), IC (acoplamiento de herencia), LOC (líneas de código), etc.

```
C:\>java -jar \ckjm_ext\ckjm_ext.jar C:\Users\QuasarPC\Documents\PECL2020\build\classes\PECL\*.class
PECL.Cliente 3 2 0 4 15 0 2 3 0,3333 135 1,0000 2 0,9697 0,5000 1 1 43,0000
~ public String getIdCliente(): 1
~ public void <init>(String idCliente, PECL.MostradorPedidos mpedidos, PECL.Log log): 1
~ public void run(): 1

PECL.Restaurante$1 2 1 0 1 4 0 1 1 1 0,0000 14 0,0000 1 0,0000 0,6667 0 0 5,5000
~ void <init>(PECL.Restaurante this$0): 1
~ public void actionPerformed(java.awt.event.ActionEvent evt): 1

PECL.Restaurante$2 2 1 0 1 4 0 1 1 1 0,0000 14 0,0000 1 0,0000 0,6667 0 0 5,5000
~ void <init>(PECL.Restaurante this$0): 1
~ public void actionPerformed(java.awt.event.ActionEvent evt): 1

PECL.Cocinero 2 2 0 3 15 0 1 2 2 0,0000 122 1,0000 2 0,9846 0,6000 1 1 58,0000
~ public void <init>(String idCocinero, PECL.MesaPlatos mplatos, javax.swing.JTextField text, PECL.Log log): 1
~ public void run(): 1

PECL.Restaurante$3 2 1 0 1 4 0 1 1 1 0,0000 14 0,0000 1 0,0000 0,6667 0 0 5,5000
~ void <init>(PECL.Restaurante this$0): 1
~ public void actionPerformed(java.awt.event.ActionEvent evt): 1

PECL.Pausar 1 1 0 0 2 0 0 1 2,0000 4 0,0000 0 0,0000 1,0000 0 0 3,0000
~ public void <init>(): 1

PECL.Restaurante$10 2 1 0 1 5 1 1 1 2,0000 11 0,0000 0 0,0000 1,0000 0 0 4,5000
~ void <init>(): 1
~ public void run(): 1

PECL.Restaurante$8 2 1 0 1 4 0 1 1 1 0,0000 14 0,0000 1 0,0000 0,6667 0 0 5,5000
~ void <init>(PECL.Restaurante this$0): 1
~ public void actionPerformed(java.awt.event.ActionEvent evt): 1

PECL.Restaurante$9 2 1 0 1 4 0 1 1 1 0,0000 14 0,0000 1 0,0000 0,6667 0 0 5,5000
~ void <init>(PECL.Restaurante this$0): 1
~ public void actionPerformed(java.awt.event.ActionEvent evt): 1

PECL.Restaurante$4 2 1 0 1 4 0 1 1 1 0,0000 14 0,0000 1 0,0000 0,6667 0 0 5,5000
~ void <init>(PECL.Restaurante this$0): 1
~ public void actionPerformed(java.awt.event.ActionEvent evt): 1
```

The screenshot shows a Java code editor window with the following details:

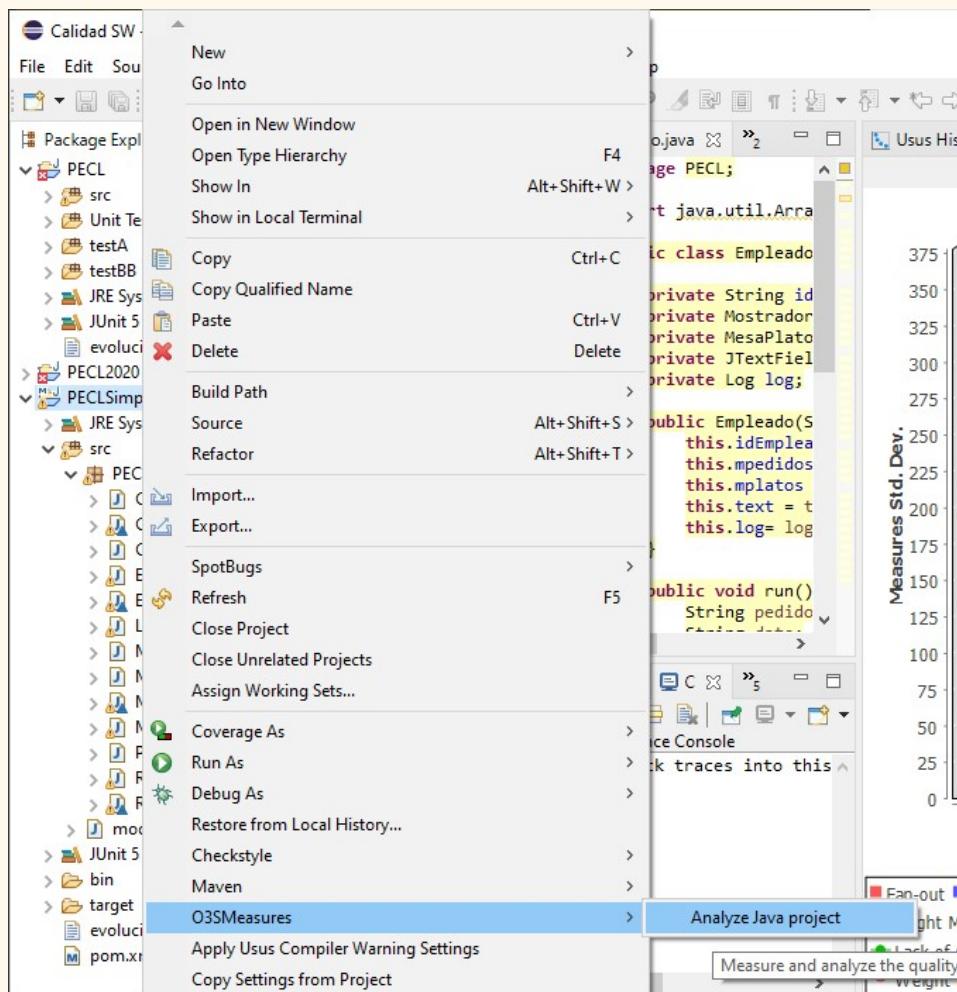
- Title Bar:** Seleccionar Símbolo del sistema
- Toolbar:** Includes standard icons for file operations.
- Code Area:**
 - PECL.Cliente:** A class with a constructor taking String, PECL.MostradorPedidos, and PECL.Log parameters, and a run() method.
 - PECL.Restaurante\$1 to \$10:** Nested inner classes for different restaurante states, each with a constructor taking PECL.Restaurante and a actionPerformed(ActionEvent) method.
 - PECL.Cocinero:** A class with a constructor taking String, PECL.MesaPlatos, and javax.swing.JTextField parameters, and a run() method.
 - PECL.Pausar:** A class with a constructor and a run() method.
 - PECL.Restaurante\$8 to \$10:** Nested inner classes for different restaurante states, each with a constructor taking PECL.Restaurante and a actionPerformed(ActionEvent) method.
 - PECL.Restaurante\$4:** A nested inner class for a specific restaurante state, with a constructor taking PECL.Restaurante and a actionPerformed(ActionEvent) method.
 - PECL.Restaurante\$2:** A nested inner class for a specific restaurante state, with a constructor taking PECL.Restaurante and a actionPerformed(ActionEvent) method.
 - PECL.Restaurante\$3:** A nested inner class for a specific restaurante state, with a constructor taking PECL.Restaurante and a actionPerformed(ActionEvent) method.
 - PECL.Empleado:** A class with a constructor taking String, PECL.MostradorPedidos, PECL.MesaPlatos, and javax.swing.JTextField parameters, and a run() method.
 - PECL.Main:** A class with a static main() method.

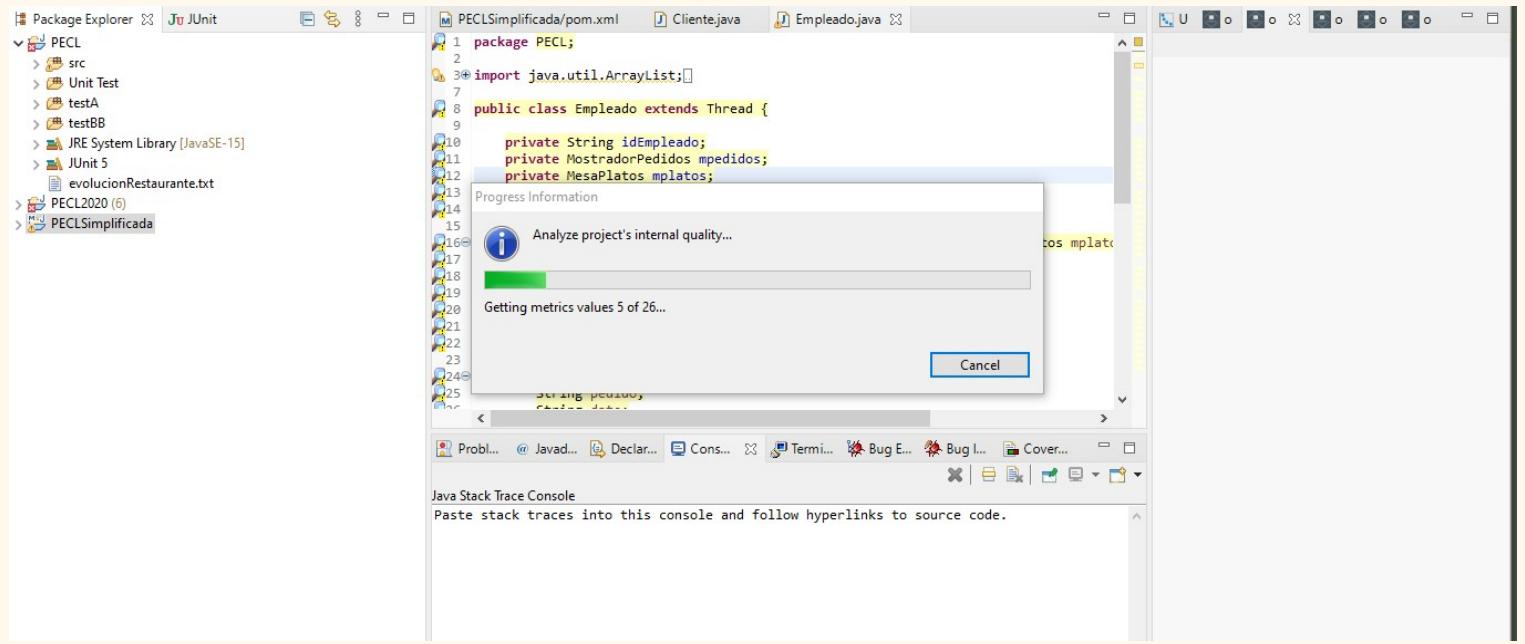
Plugin Eclipse

Como herramienta adicional, vamos a probar con un plugin disponible en la Marketplace de Eclipse, **o3measure**:

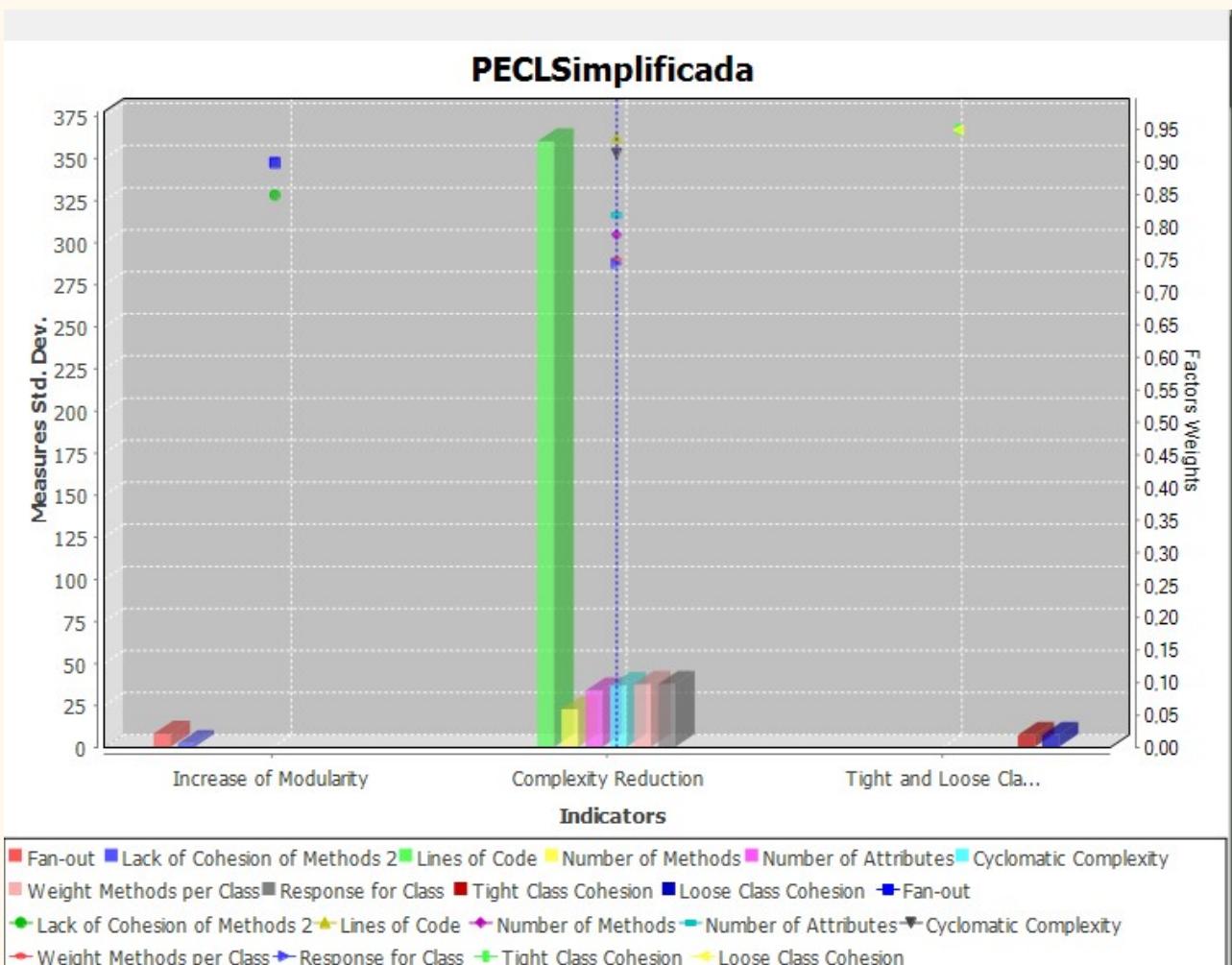
<https://marketplace.eclipse.org/content/o3smeasures#group-screenshots>

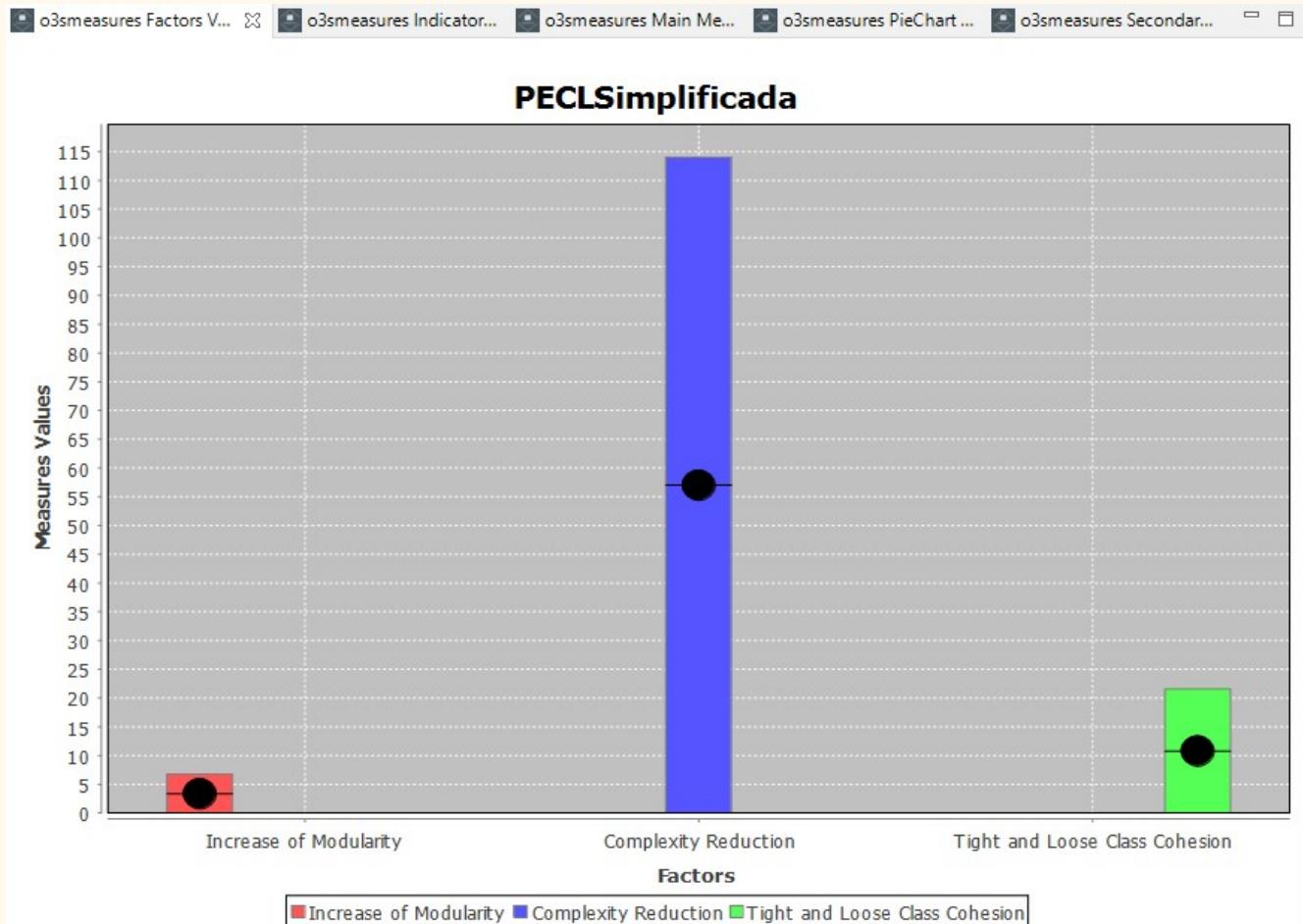
Accedemos a él haciendo click derecho sobre el proyecto a analizar > O3Measures > Analyze Java Project





Este plugin nos permite tener 5 visualizaciones para nuestro proyecto Java:

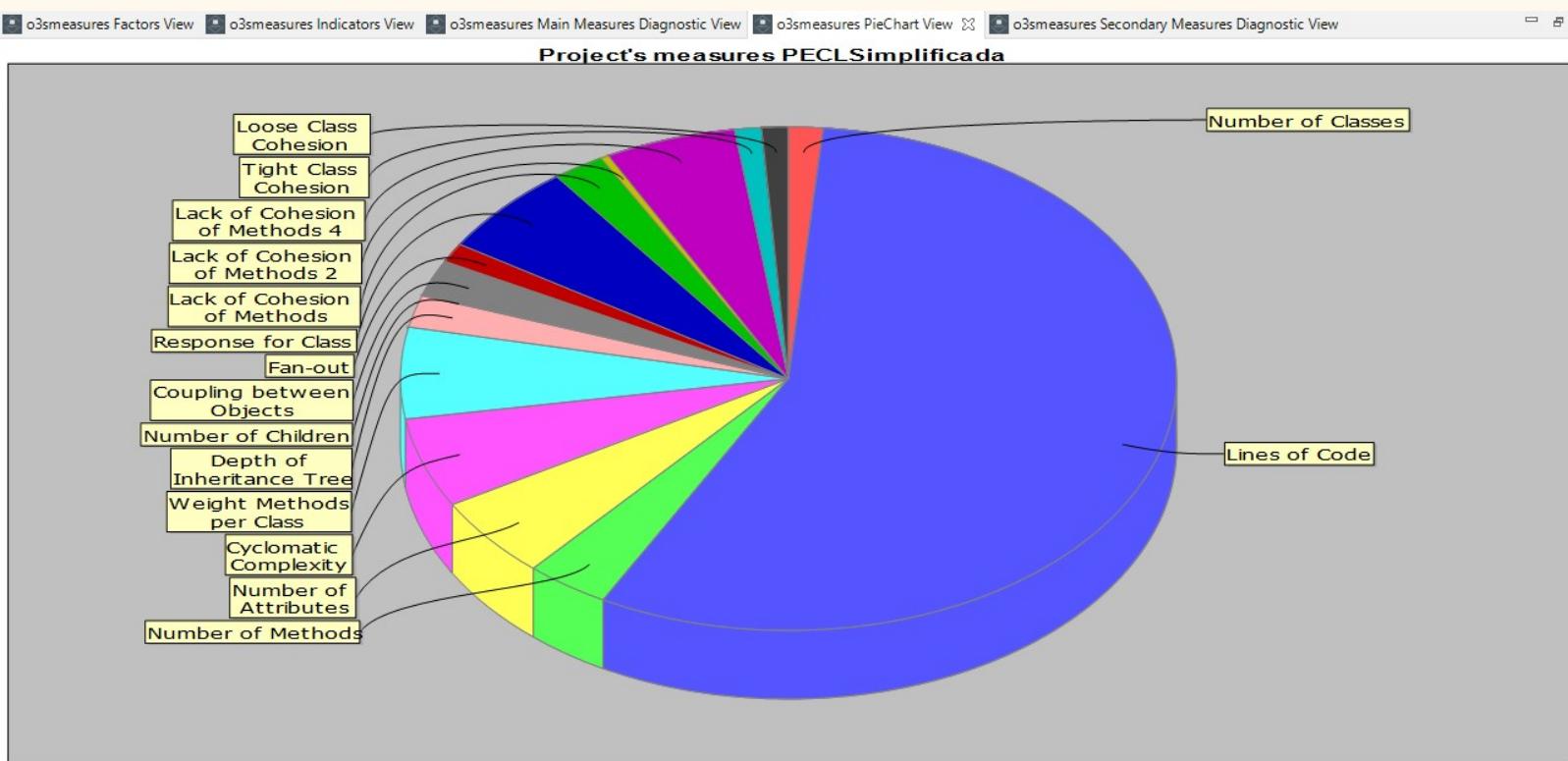




o3smeasures Factors View o3smeasures Indicators View o3smeasures Main Measures Diagnostic View o3smeasures PieChart View o3smeasures Secondary Measures Diagnostic View

Project: PECLSimplificada

Item	Value	Mean Value per...	Min Value	Max Value	Resource with ...	Description
> Number of Packages	2	0	1	1		Total number of packages in a project
> Number of Lambdas	0	0	0	0		Total number of lambda expression used in a class
> Number of Interfaces	0	0	0	0		Total number of interfaces in a project
> Number of Method References	0	0	0	0		Total number of method references used in a class
> Number of Modules	1	0	0	1	module-info.java	Total number of modules used in a project
> Efferent Coupling	45	3.214	6	7	Restaurante.java	The number of classes inside a package that depend on classes outside the package.
> Afferent Coupling	0	0	0	0		The number of classes outside a package that depend on classes inside the package.
> Abstractness	0	0	0	0		The number of abstract classes (and interfaces) divided by the total number of types in a package.
> Instability	11	0.786	1	1	Restaurante.java	$C_e / (C_a + C_e)$ - indicates the necessity of performing modifications in an entity due to updates occurred in o...
> Distance from the Main Sequence	3	0.214	0	1	module-info.java	$ A + I - 1 $, this number should be small, close to zero for good packaging design.



Project: PECL Simplificada

Item	Value	Mean Value per...	Min Value	Max Value	Resource with ...	Description
Number of Classes	14	0	1	1	module-info.java	Return the number of classes and inner classes of a class in a project.
Lines of Code	549	39.214	1	219	Restaurante.java	Number of the lines of the code in a project.
Number of Methods	34	2.429	2	12	Restaurante.java	The number of methods in a project.
Number of Attributes	51	3.643	3	24	Restaurante.java	The number of attributes in a project.
Cyclomatic Complexity	56	4	2	5	Restaurante.java	It is calculated based on the number of different possible paths through the source code.
Weight Methods per Class	57	4.071	3	5	Log.java	It is the sum of the complexities of all class methods.
Depth of Inheritance Tree	20	1.429	1	6	Restaurante.java	Provides the position of the class in the inheritance tree.
Number of Children	0	0	0	0		It is the number of direct descendants (subclasses) for each class.
Coupling between Objects	23	1.643	2	4	Cliente.java	Total of the number of classes that a class referenced plus the number of classes that referenced the ...
Fan-out	12	0.857	1	1	Pausar.java	Defined as the number of other classes referenced by a class.
(default)	0	0	0	0		
PECL	12	0.857	1	1	Pausar.java	
Pausar.java	1	0.071	1	1	Pausar.java	
Restaurante.java	1	0.071	1	1	Pausar.java	
RestauranteTest.java	1	0.071	1	1	Pausar.java	
Empleado.java	1	0.071	1	1	Pausar.java	
ClienteTest.java	1	0.071	1	1	Pausar.java	
MesaPlatos.java	1	0.071	1	1	Pausar.java	
MostradorPedidos.java	1	0.071	1	1	Pausar.java	
Cocinero.java	1	0.071	1	1	Pausar.java	
EmpleadoTest.java	1	0.071	1	1	Pausar.java	
Cliente.java	1	0.071	1	1	Pausar.java	
MesaPlatosTest.java	1	0.071	1	1	Pausar.java	
Main.java	0	0	0	1	Pausar.java	
Log.java	1	0.071	1	1	Pausar.java	
Response for Class	57	4.071	3	23	Restaurante.java	Measures the complexity of the class in terms of method calls. It is calculated by adding the number ...
Lack of Cohesion of Methods	21	1.5	2	5	MesaPlatos.java	LCOM defined by CK.
Lack of Cohesion of Methods 2	3.36	0.24	1	1	Log.java	It is the percentage of methods that do not access a specific attribute averaged over all attributes in t...
Lack of Cohesion of Methods 4	53	3.786	3	24	Restaurante.java	LCOM4 measures the number of 'connected components' in a class. A connected component is a se...
Tight Class Cohesion	10.798	0.771	0	5	Empleado.java	Measures the 'connection density', so to speak (while LCC is only affected by whether the methods a...
Loose Class Cohesion	10.798	0.771	0	5	Empleado.java	Measures the overall connectedness. It depends on the number of methods and how they group tog...

Algunas métricas que podemos observar aquí coinciden con las analizadas previamente con CKJM, como por ejemplo las líneas de código (loc), la cohesión (LCOM), el número de hijos (NOC), el DIT (nivel de herencia) y la complejidad ciclomática, además de otras métricas que no son analizadas como el CKJM, como el Fan-out.

Además, este plugin nos indica la descripción de cada métrica, el valor que le asigna por clase y la media. Todas estas métricas también las podemos ver en forma de gráfica en un *Pie chart* y en gráfica de barras, lo cual nos puede ayudar a tener una visión más clara y rápida del estado de nuestro proyecto.

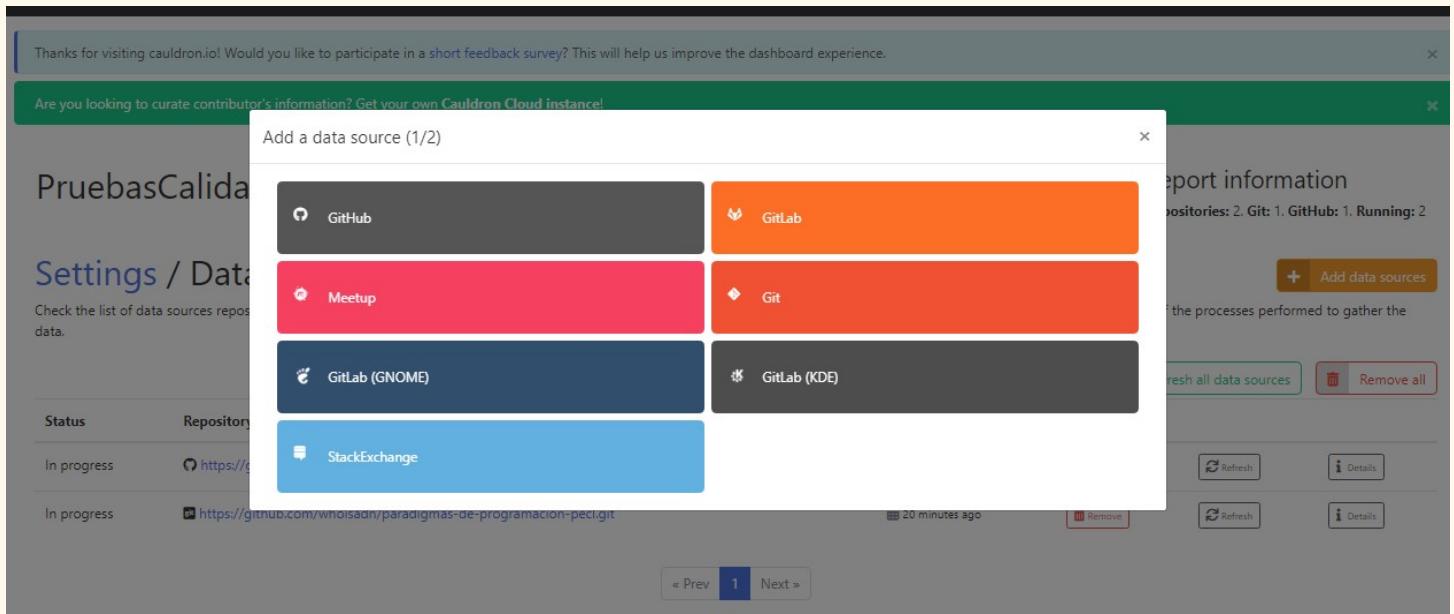
Para concluir, hay que mencionar que el plugin de Eclipse, además de ser mucho más sencillo de implementar y hacer uso (no hay que descargar un archivo, descomprimirlo, ejecutar CMD y tener nociones, aunque sean básicas de comandos para poder visualizar las métricas de las clases) visualmente gana mucho, ya que podemos ver los resultados por tablas o gráficos y no por simples salidas de terminal, lo cual se hace hasta cierto punto complejo.

Resumen Documentación Ejercicio 2

Objetivo	Analizar un proyecto con cuadros de mando.
Código	https://github.com/whoisadrn/paradigmas-de-programacion-pecl https://github.com/whoisadrn/bases-de-datos-avanzadas/blob/master/README.md
Entorno necesario	Cauldron (online)+ SonarCloud (online)
Conclusiones	Ambas herramientas comparten ciertas similaridades, ofreciéndonos de manera gráfica resultados de nuestros proyectos medidos por diversas métricas (mantenimiento, pruebas y gestión de la configuración), de manera que posteriormente analicemos y tomemos en cuenta para la toma de decisiones.

Couldron

Couldron es una herramienta gratuita y OpenSource, que nos permite ver de manera gráfica en forma de dashboard un resumen análisis de nuestros proyectos, ya sea que provengan de GitHub o GitLab.



The screenshot shows the Couldron dashboard interface. At the top, there's a green header bar with the text "Thanks for visiting cauldron.io! Would you like to participate in a short feedback survey? This will help us improve the dashboard experience." Below this, a modal window titled "Add a data source (1/2)" is open, listing various data sources: GitHub (selected), GitLab, Meetup, Git, GitLab (GNOME), GitLab (KDE), and StackExchange. The main dashboard background shows a dark theme with a list of repositories and their status (e.g., "In progress").

Su uso es muy sencillo, basta con asociar nuestro repositorio con la herramienta:

Create a new report

A report contains a set of repositories from multiple data sources. Cauldron collects and analyzes the data to display it in an elegant way.

Report name

PruebasCalidadSoftware

Add data sources

Git GitHub GitLab Meetup GNOME (GitLab) KDE (GitLab) Stack Exchange

Included

Remove all

<input checked="" type="checkbox"/> Github	whoisadn/paradigmas-de-programacion-pecl	<input checked="" type="checkbox"/> Commits
		<input checked="" type="checkbox"/> Issues/PRs

Notify me on Twitter Authenticate ?

Create report

Podemos, de golpe, mediante el link directo de nuestra cuenta, asociar todos nuestros repositorios.

Settings / Data sources

Check the list of data sources repositories being tracked to collect data for this report. The data sources section allow you to remove or refresh data from repositories, and to check logs of the processes performed to gather the data.

Add data sources

Status	Repository	Last refresh	Actions
Pending	https://github.com/whoisadn/paradigmas-de-programacion-pecl	2 minutes ago	Remove Refresh Details
Pending	https://github.com/whoisadn/paradigmas-de-programacion-pecl.git	3 minutes ago	Remove Refresh Details
Analyzed	https://gitlab.com/dte2020/m3/tg2/	1 minute ago	Remove Refresh Details
In progress	https://github.com/whoisadn/bases-de-datos-avanzadas	Not analyzed	Remove Refresh Details
Pending	https://github.com/whoisadn/bases-de-datos-avanzadas.git	Not analyzed	Remove Refresh Details
In progress	https://github.com/whoisadn/estructuras-de-datos-PECL1	Not analyzed	Remove Refresh Details
Pending	https://github.com/whoisadn/estructuras-de-datos-PECL1.git	Not analyzed	Remove Refresh Details
Pending	https://github.com/whoisadn/estructuras-de-datos-PECL2	Not analyzed	Remove Refresh Details
Pending	https://github.com/whoisadn/estructuras-de-datos-PECL2.git	Not analyzed	Remove Refresh Details

Refresh all data sources Remove all

También podemos ver de manera pública y en directo todos los repositorios que hacen uso de esta tecnología.

Cauldron higueras Reports Compare + New report whoisadn

Reports

Your reports 1 Explore reports 3686 Last created Search by name

PruebasCalidadSoftware

Data sources 1 1

	Commits	Issues	Reviews
Items	2	0	0
Authors	2	0	0

Creation date: 30th May 2021

Laminas

Data sources 124 124

	Commits	Issues	Reviews
Items	54478	1576	2056
Authors	1853	236	284

Creation date: 29th May 2021

saffron

Data sources 2 2

	Commits	Issues	Reviews
Items	79	2	7
Authors	7	2	3

Creation date: 28th May 2021

Test

Data sources 1 1

	Commits
Items	0
Authors	0

<https://cauldron.io>

GitLab2PROV

Data sources 1 1

	Commits	Issues	Reviews
Items	302	42	0
Authors	4	3	0

Blender

Data sources 7 7

	Commits
Items	154202
Authors	940

Además nos permite hacer comparaciones entre varios reports.

Cauldron higueras Reports + New report whoisadn

Thanks for visiting cauldron.io! Would you like to participate in a short feedback survey? This will help us improve the dashboard experience.

Are you looking to curate contributor's information? Get your own Cauldron Cloud Instance!

Compare reports

PruebasCalidadSoftware PruebasCalidadSoftware PruebasCalidadSoftware + Compare

Overview Commits Activity Issues Activity PRs/MRs Activity Community 2020-05-30 - 2021-05-30

Commits

Pull requests

Median review duration

Issues created

Issues closed

Median time to close

PruebasCalidadSoftware

2
0
X days
0
0
X days

Commits

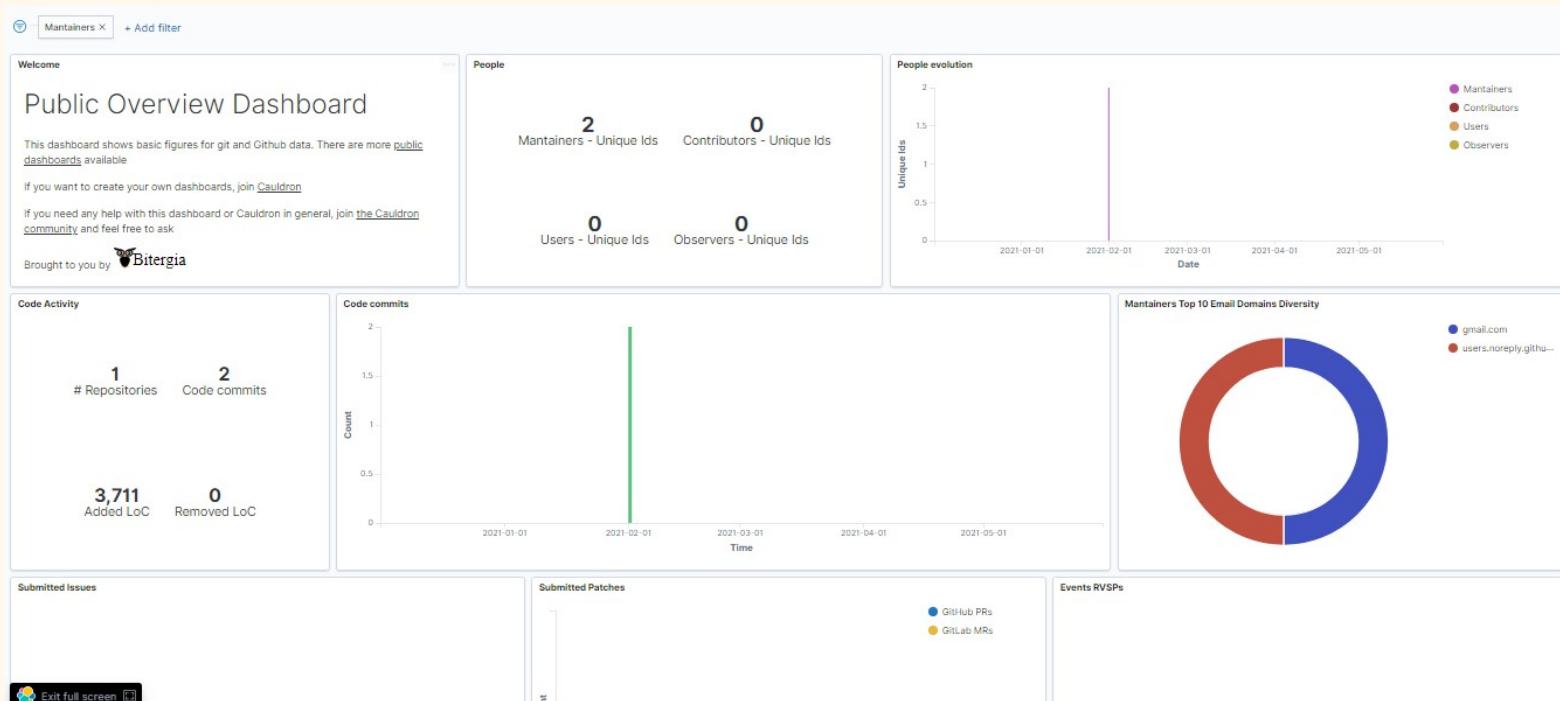
Commits

Time

Algunos datos que podemos extraer de esta herramienta son:

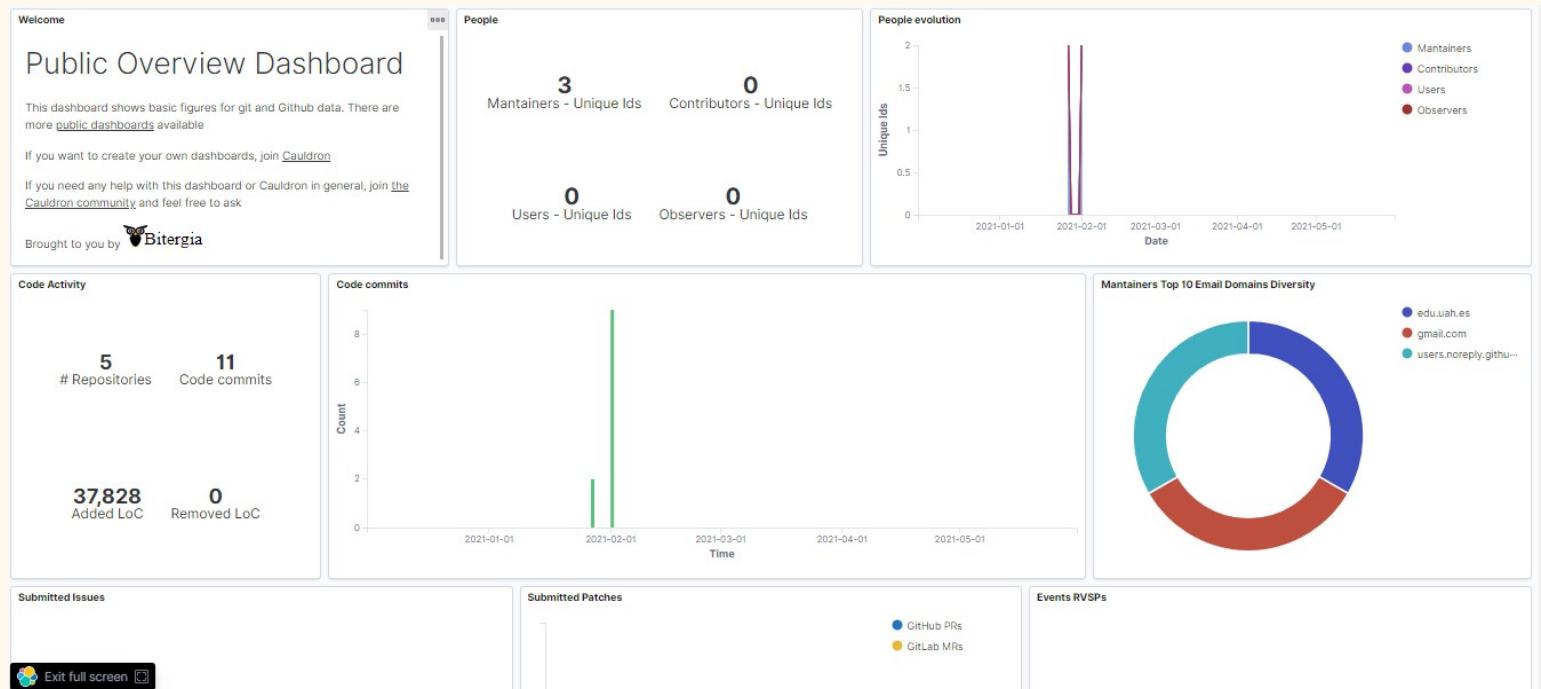
- Número de commits en el tiempo
- Número de Pull Request en el tiempo
- Número de issues creadas o cerradas
- Tiempo medio hasta que cerramos un proyecto
- Comparaciones en el tiempo (año contra año)
- Dentro del Dashboard podemos ver los usuarios que contribuyen al proyecto, el total de commits, LOC, issues, entre otros datos.

En esta captura podemos observar el dashboard del repositorio con el código:

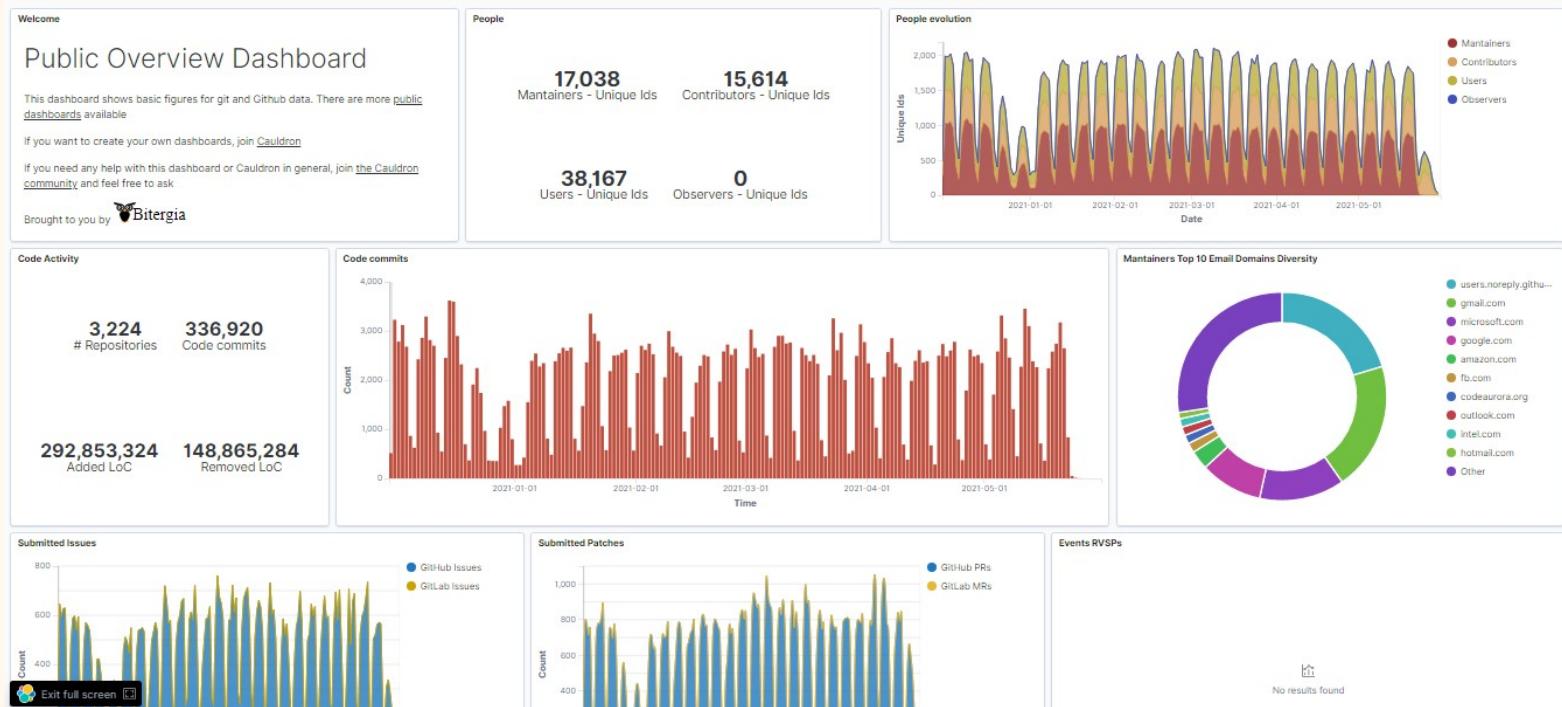


Podemos observar que, efectivamente el código está en un repositorio, y con un total de 3.711 LOC (Líneas de código).

Dashboard de todo mi repositorio:

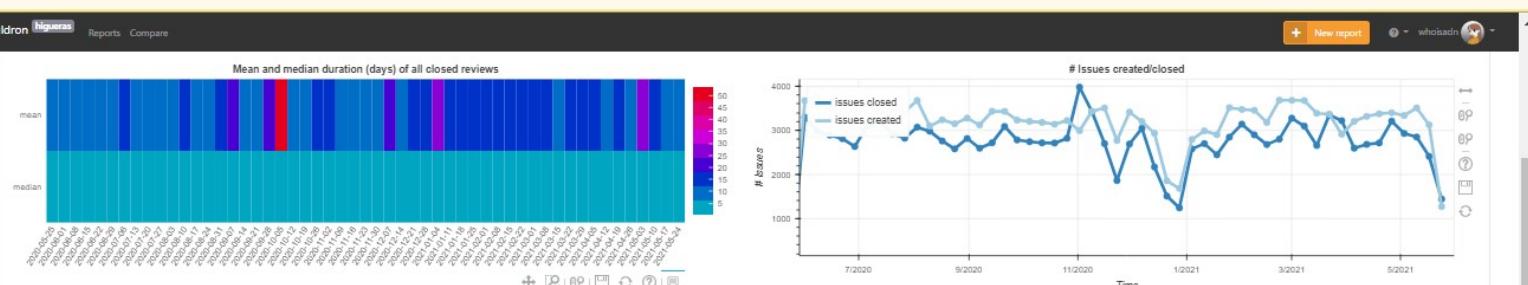


Comparativa con un Dashboard público con mucho más volumen de repositorio:



Ahora vamos a ver algunas otras comparaciones entre mi proyecto frente al anterior, para poder visualizar de mejor manera el potencial de la herramienta:

Mi proyecto desde la pestaña CHAOS vs el tomado como ejemplo:



Mi proyecto desde la pestaña Summary vs el tomado como ejemplo:

Podemos ver que está vacío, ya que no hay suficientes datos como para contabilizarlos.

	Commits	Issues	Reviews
Total	0	0	0
Total (Last year)	0	0	0
Year-over-year	0 %	0 %	0 %

	Commit authors	Issue submitters	Review submitters
Total	0	0	0
Total (Last year)	0	0	0
Year-over-year	0 %	0 %	0 %

	Commits	Issues	Reviews
Total	X	689882	686129
Total (Last year)	679369	167850	213681
Year-over-year	-2.32 %	7.89 %	26.85 %

	Commit authors	Issue submitters	Review submitters
Total	120210	222858	73637
Total (Last year)	30484	72439	30606
Year-over-year	1.39 %	12.85 %	55.35 %

Todo esto lo convierte en una muy buena herramienta para controlar versiones (ayuda con la gestión de la configuración), el mantenimiento, las pruebas, la gestión del tiempo, etc. Cumple con la función de ser sencilla de usar y mostrar datos e información relevante que posteriormente nosotros podamos usar para tomar decisiones sobre nuestro proyecto.

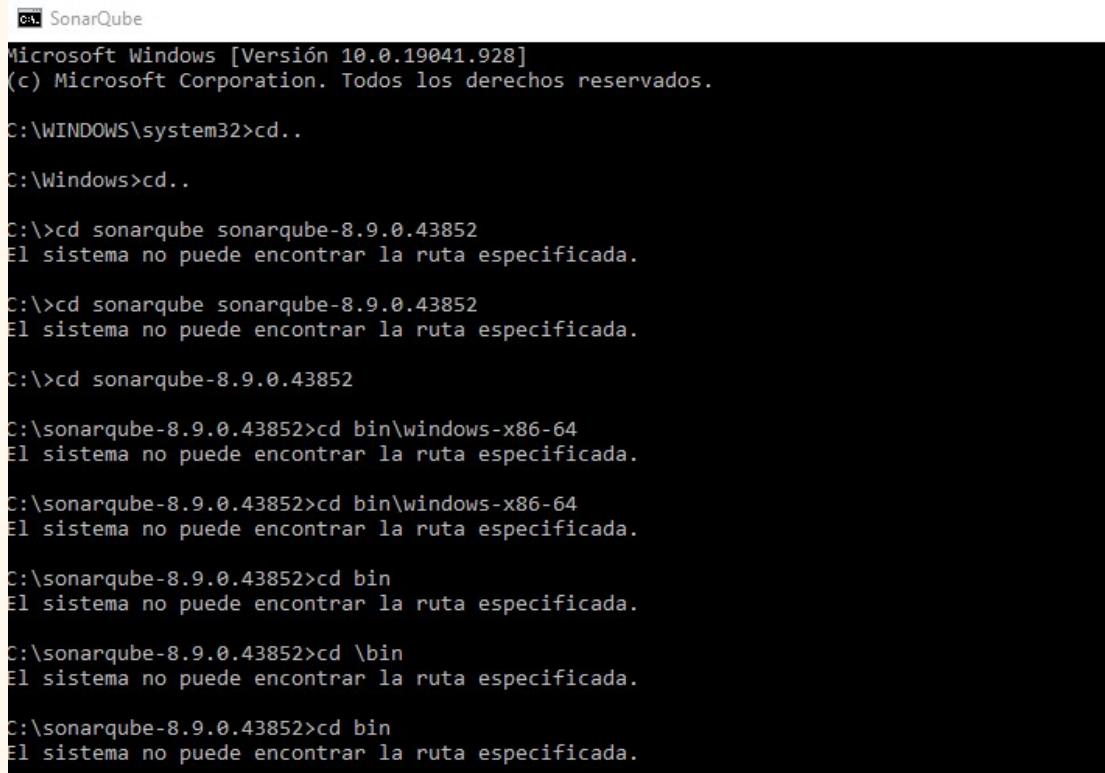
También cabe destacar que esta herramienta está muy enfocada a proyectos de grandes tamaños, con muchos contribuyentes y que se larguen en el tiempo, ya que unos proyectos más normales no son suficientes como para que el software pueda generar informes.

SonarQube

SonarQube es otra herramienta que nos permitirá visualizar en forma de dashboard nuestros proyectos. También acepta GitHub y GitLab como Coultron y adicionalmente Bitbucket y Azure DevOps.

Nos ayuda a hacer un análisis estático de código (analizar sin ejecutar el código) en más de 15 lenguajes, además de detectar bugs y vulnerabilidades y ofrecernos métricas para analizar la calidad del código con integración continua.

Descargaremos la herramienta de su web oficial (la versión gratuita community, ya que el resto son licencias de pago). Descomprimimos el archivo, lo llevamos al nuestro disco C:// para más facilidad y ejecutamos los siguientes comandos desde CMD (ya que estamos en Windows):



```
C:\ SonarQube
Microsoft Windows [Versión 10.0.19041.928]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>cd..

C:\Windows>cd..

C:\>cd sonarqube sonarqube-8.9.0.43852
El sistema no puede encontrar la ruta especificada.

C:\>cd sonarqube sonarqube-8.9.0.43852
El sistema no puede encontrar la ruta especificada.

C:\>cd sonarqube-8.9.0.43852

C:\sonarqube-8.9.0.43852>cd bin\windows-x86-64
El sistema no puede encontrar la ruta especificada.

C:\sonarqube-8.9.0.43852>cd bin\windows-x86-64
El sistema no puede encontrar la ruta especificada.

C:\sonarqube-8.9.0.43852>cd bin
El sistema no puede encontrar la ruta especificada.

C:\sonarqube-8.9.0.43852>cd \bin
El sistema no puede encontrar la ruta especificada.

C:\sonarqube-8.9.0.43852>cd bin
El sistema no puede encontrar la ruta especificada.
```

```

SonarQube
C:\sonarqube-8.9.0.43852>cd \bin
El sistema no puede encontrar la ruta especificada.

C:\sonarqube-8.9.0.43852>cd bin
El sistema no puede encontrar la ruta especificada.

C:\sonarqube-8.9.0.43852>cd sonarqube-8.9.0.43852\bin\windows-x86-64

C:\sonarqube-8.9.0.43852\sonarqube-8.9.0.43852\bin\windows-x86-64>StartSonar.bat
wrapper | --> Wrapper Started as Console
wrapper | Launching a JVM...
jvm 1 | Wrapper (Version 3.2.3) http://wrapper.tanukisoftware.org
jvm 1 | Copyright 1999-2006 Tanuki Software, Inc. All Rights Reserved.
jvm 1 |
jvm 1 |
jvm 1 | WrapperSimpleApp: Encountered an error running main: java.lang.IllegalStateException: SonarQube requires Java 11 to run
jvm 1 |     at com.google.common.base.Preconditions.checkState(Preconditions.java:508)
jvm 1 |     at org.sonar.application.App.checkJavaVersion(App.java:93)
jvm 1 |     at org.sonar.application.App.start(App.java:56)
jvm 1 |     at org.sonar.application.App.main(App.java:97)
jvm 1 |     at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
jvm 1 |     at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
jvm 1 |     at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
jvm 1 |     at java.lang.reflect.Method.invoke(Unknown Source)
jvm 1 |     at org.tanukisoftware.wrapper.WrapperSimpleApp.run(WrapperSimpleApp.java:240)
jvm 1 |     at java.lang.Thread.run(Unknown Source)
wrapper | <- Wrapper Stopped
Presione una tecla para continuar . .

```

Mientras ejecutamos los comandos llegamos a un error por lo que debemos instalar Java 11, desde los archivos de configuración y realizaremos unos cambios sobre el archivo wrapper.conf:

```

wrapper: Bloc de notas
Archivo Edición Formato Ver Ayuda
# Path to JVM executable. By default it must be available in PATH.
# Can be an absolute path, for example:
wrapper.java.command=C:\Users\QuasarPC\Downloads\openjdk-11+28_windows-x64_bin\jdk-11\bin\java.exe
#wrapper.java.command=java

#
# DO NOT EDIT THE FOLLOWING SECTIONS
#



*****#
# Wrapper Java
*****#
wrapper.java.additional.1=-Dsonarwrapped=true
wrapper.java.additional.2=-Djava.awt.headless=true
wrapper.java.mainclass=org.tanukisoftware.wrapper.WrapperSimpleApp
wrapper.java.classpath.1=../../lib/sonar-application-8.9.0.43852.jar
wrapper.java.classpath.2=../../lib/jsw/wrapper-3.2.3.jar
wrapper.java.classpath.3=../../lib/sonar-shutdowner-8.9.0.43852.jar
wrapper.java.library.path.1=./lib
wrapper.app.parameter.1=org.sonar.application.App
wrapper.java.initmemory=8
wrapper.java.maxmemory=32

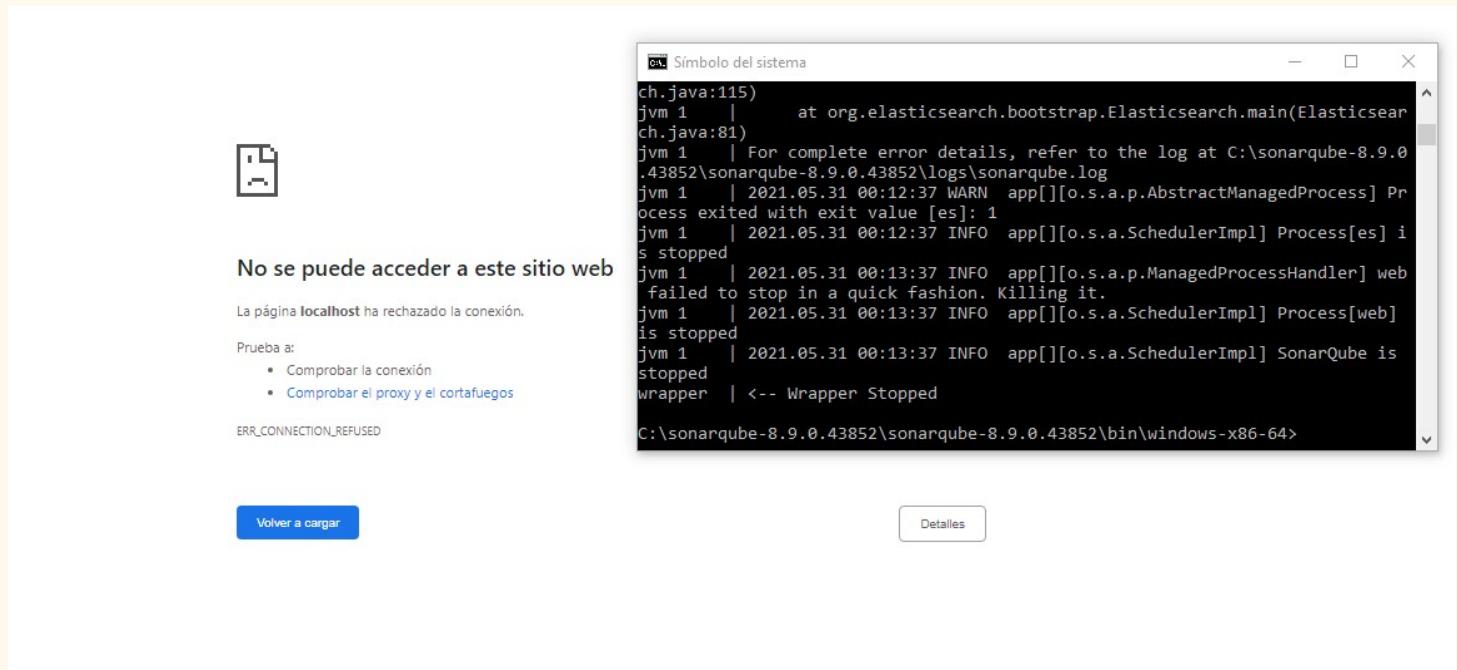
*****#
# Wrapper Logs
*****#



wrapper.console.format=PM
wrapper.console.loglevel=INFO
wrapper.logfile.format=M

```

Después volvemos a ejecutar el comando, aceptamos la alerta del Firewall de Windows y entraríamos desde un navegador buscando por localhost:9000.



Como esto generaba problemas, opté por cambiar a la opción SonarCloud, que también cuenta con una versión gratuita.

The screenshot shows the SonarCloud welcome page. At the top, there is a navigation bar with links for "sonarcloud", "My Projects", "My Issues", "Explore", and a search bar. Below the navigation, there is a large orange cloud icon. The main heading is "Welcome to SonarCloud" with the subtext "Let us help you get started in your journey to code quality". There are two main sections: "Analyze your first projects" (with a GitHub icon) and "Join an organization" (with a user icon). Both sections provide instructions for getting started with SonarCloud.



An organization is a space where a team or a whole company can collaborate across many projects.

1 Import organization details

 calidadpruebassw

2 Choose a plan

Free plan

€0

All projects you analyze will be public.
Anyone can browse source code.

Paid plan

from €10

- ✓ Unlimited private projects
- ✓ Strict control over who can view your private data
- ✓ No commitments, cancel anytime
- ✓ 14 days free trial.

[Learn more](#)

[Create Organization](#)

Descubrimos que Sonar no puede ejecutar pruebas de análisis automático sobre lenguajes como Java, C o C++. Y para nuestro código de Java necesitaríamos convertirlo a Maven (ya que necesitaríamos un archivo pom.xml y otro .yml).



Ad; /  paradigmas-de-programacion-pecl 

 master 

Configure

Issues

Security Hotspots

Measures

Code

Activity

Administration ▾



We do not recommend Automatic Analysis for this project.

100% of your code is in languages unsupported by Automatic Analysis [Hide information ▾](#)

 100% Lines of Code will be ignored (Java files)

[Learn more about Automatic Analysis](#)

 [Recheck compatibility](#)

 [Force Automatic Analysis](#)

sonarcloud  My Projects My Issues

Explore ? Search for projects and files... + 

Ad: / estructuras-de-datos-PECL1  master +

Configure Issues Security Hotspots Measures Code Activity Administration ▾

PUBLIC  

Analyze with a GitHub Action

- 1 Create a GitHub Secret
- 2 Create or update a `.github/workflows/build.yml` file

What option best describes your build?

Maven Gradle .NET Other (for JS, TS, Go, Python, PHP, ...)

Update your `pom.xml` file with the following properties:

```
<properties>
<sonar.projectKey>whoisad_estructuras-de-datos-PECL1</sonar.projectKey>
<sonar.organization>calidadpruebasw</sonar.organization>
<sonar.host.url>https://sonarcloud.io</sonar.host.url>
</properties>
```



Create or update your `.github/workflows/build.yml`  file with the following content:

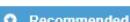
Pero sí nos ofrecen una amplia gama de herramientas de testeo de integración continua.

Choose your Analysis Method

 With GitHub Actions >

 With Travis CI >

 With CircleCI >

 Recommended

 With other CI tools >

SonarCloud integrates with your workflow no matter which CI tool you're using.

 Manually >

Use this for testing. Other modes are recommended to help you set up your CI environment.

Ya que esto suponía una dificultad elevada, decidí cambiar a otro código de Python, con el que obtuve muchos mejores resultados:

The screenshot shows the SonarCloud Quality Gates dashboard. At the top, there are tabs for Quality Gates, Members, and Administration. The main area displays a summary for the project 'bases-de-datos-avanzadas'. The status is 'Not computed'. Key metrics shown include: 0 Bugs (A), 0 Vulnerabilities (A), 100% Hotspots Reviewed (A), 0 Code Smells (A), 0.0% Duplications (green circle), and 169 Python files (169 XS). A search bar at the top right allows searching by project name or key, and it indicates there are 2 projects.

The screenshot shows the SonarCloud project details for 'bases-de-datos-avanzadas'. The project is public and was last analyzed on May 31, 2021, at 1:09 AM. It contains 169 Python files. The Reliability section shows 0 Bugs (A). The Security section shows 0 Vulnerabilities (A) and 0 Security Hotspots (A). The Maintainability section shows 0 Debt (A) and 0 Code Smells (A). The Duplications section shows 0.0% Duplications (green circle) and 0 Duplicated Blocks (A). On the right side, there are sections for Project Activity (not provided), Quality Gate (Sonar way), Quality Profiles (Sonar way), Last analysis method (Analyzed by SonarCloud), Project Key (whoisadmin_bases-de-datos-avanzadas), Organization Key (calidadpruebabassw), and links to Get project badges and Set notifications.

The screenshot shows the SonarCloud interface for the 'bases-de-datos-avanzadas' project. The top navigation bar includes 'My Projects' and 'My Issues' on the left, and 'Explore' and a search bar on the right. The main content area displays the 'Code' tab for the 'master' branch. It shows a file tree under 'PECL2-BDA_2019-20' containing several Python files: Producto.py, ticket.py, ticket_producto.py, Tienda.py, tienda_producto.py, and Trabajador-DEF.py. The table below provides a detailed breakdown of code metrics for each file:

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
PECL2-BDA_2019-20	169	0	0	0	0	—	0.0%
Producto.py	27	0	0	0	0	—	0.0%
ticket.py	31	0	0	0	0	—	0.0%
ticket_producto.py	25	0	0	0	0	—	0.0%
Tienda.py	31	0	0	0	0	—	0.0%
tienda_producto.py	25	0	0	0	0	—	0.0%
Trabajador-DEF.py	30	0	0	0	0	—	0.0%

At the bottom, it says '6 of 6 shown'.

En general, dado que el código tiene muy pocas líneas, hemos obtenido una valoración muy positiva.

Nota: podemos añadir incluso badges personalizados de nuestro código analizado.

The screenshot shows a GitHub repository page for 'bases-de-datos-avanzadas'. At the top, it indicates '1 contributor'. Below that, it shows '13 lines (10 sloc) | 1.08 KB'. On the right, there are buttons for 'Raw', 'Blame', 'Copy', and 'Edit'. The main content area features the repository name 'bases-de-datos-avanzadas' in large bold letters. Below it, a brief description states: 'Prácticas de la asignatura BDA de PostgreSQL.' and lists 'Practica 1: Arquitectura PostgreSQL y Almacenamiento físico', 'Practica 2: Carga Masiva de Datos, Procesamiento y Optimización de Consultas', 'Practica 3: Seguridad, Usuarios y Transacciones', and 'Practica 4: Replicación e Implementación de una Base de Datos Distribuida'. At the bottom, there is a SonarCloud badge with the text 'scanned on sonarcloud' and metrics: 'lines of code 169', 'reliability A', and 'security A'.

También es interesante destacar la medición de la complejidad ciclomática y el % de comentarios y líneas de código.

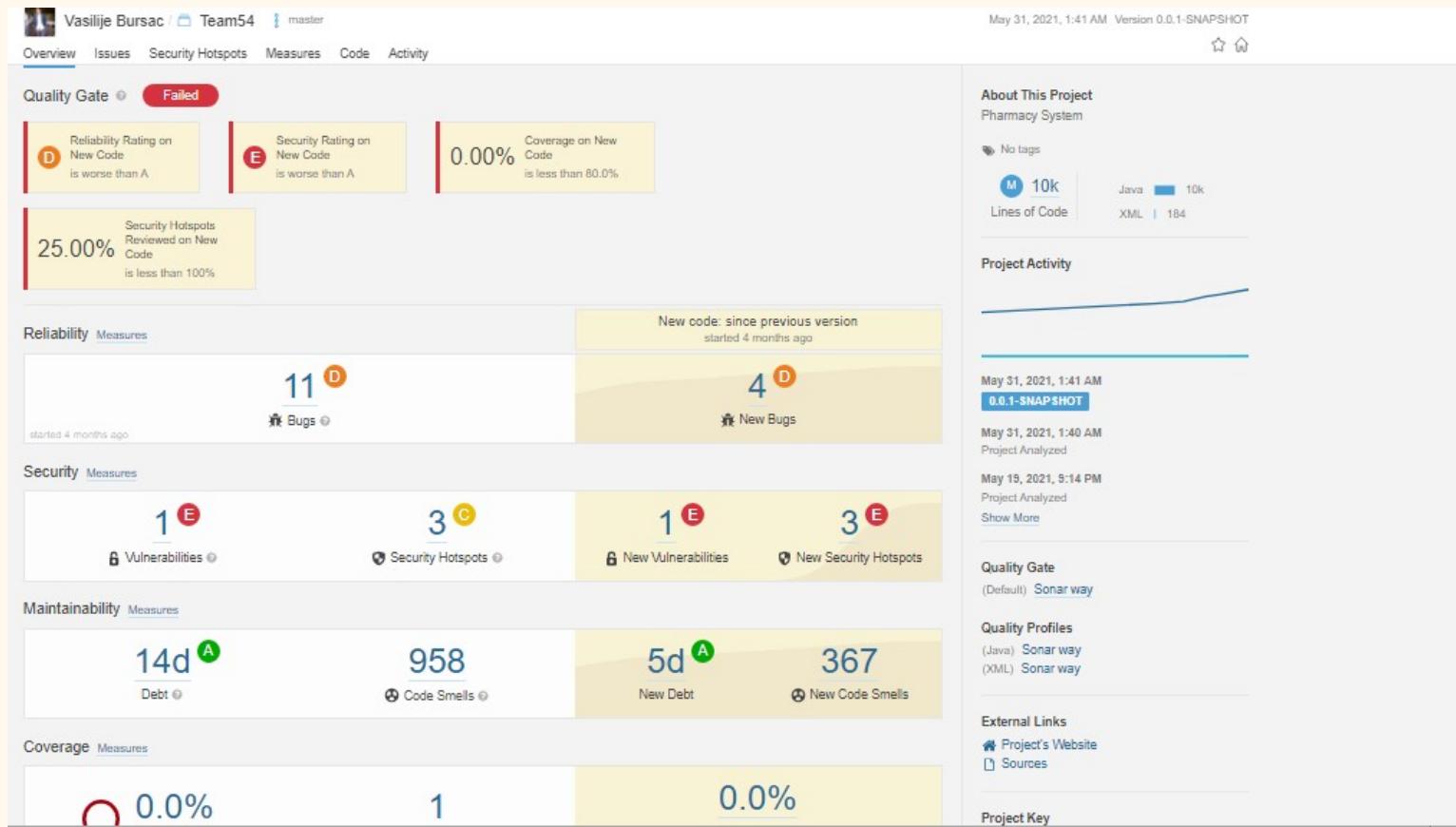
Project Metrics	
Size	
Lines of Code	169
Lines	294
Statements	168
Functions	6
Classes	0
Files	6
Comment Lines	41
Comments (%)	19.5%
Complexity	
Cyclomatic Complexity	29
Cognitive Complexity	23
Issues	

Por otro lado, también hay que destacar que al igual que con Coultron, este software nos permite ver en directo los movimientos que hay:

The screenshot shows the SonarQube interface displaying four project cards. Each card provides a summary of the project's status and analysis metrics.

- Jidim / test_3** (NEW, Passed):
 - Last analysis: May 31, 2021, 1:27 AM
 - Metrics: 0 Bugs (A), 0 Vulnerabilities (A), 0.0% Hotspots Reviewed (E), 477 Code Smells (A), 21.3% Duplications, 7.8k Python
- gaurav-dogra / excel-to-excelParsingWebApp** (NEW, Passed):
 - Last analysis: May 31, 2021, 1:27 AM
 - Metrics: 0 Bugs (A), 0 Vulnerabilities (A), 100% Hotspots Reviewed (A), 0 Code Smells (A), 0.0% Duplications, 46 XML
- Ad/ / bases-de-datos-avanzadas** (NEW, Not computed):
 - Last analysis: May 31, 2021, 1:26 AM
 - Metrics: 0 Bugs (A), 0 Vulnerabilities (A), 100% Hotspots Reviewed (A), 0 Code Smells (A), 0.0% Duplications, 169 Python
- MiTty / create-logger** (Passed):
 - Last analysis: May 31, 2021, 1:25 AM
 - Metrics: 0 Bugs (A), 0 Vulnerabilities (A), 100% Hotspots Reviewed (A), 14 Code Smells (A), 0.0% Duplications, 1.4K TypeScript, CSS

Todos estos proyectos son públicos (la versión privada es la de pago). Como ejemplo, al igual que con la herramienta anterior, podemos ver un reporte completo de un proyecto de 10k de líneas de código.



Con todos estos análisis de las 2 herramientas, podemos concluir que son herramientas muy enfocadas a trabajar con proyectos grandes, de manera que nos ayude controlando versiones, permisos, tiempos, vulnerabilidades, errores, entre muchas otras opciones.

Ambas herramientas son gratuitas y están en la nube, lo que las hace fácilmente accesibles desde cualquier dispositivo. También permiten la integración continua con las tecnologías más empleadas en el entorno de desarrollo, como son Github y GitLab y la asociación de nuestra cuenta con estas herramientas es muy sencilla.

En cuestión de minutos podemos empezar a observar nuestros resultados en formato de cuadro de mando.

Cauldron se centra más en la parte de gestión del proyecto, mientras que SonarCloud cuenta además con la opción de mostrarnos análisis de los códigos gracias a algunas métricas como el porcentaje de duplicidad del código, seguridad, fiabilidad y mantenimiento, la complejidad ciclomática, etc.

Bibliografía consultada:

<https://www.spinellis.gr/sw/ckjm/doc/indexw.html>

http://gromit.iiar.pwr.wroc.pl/p_inf/ckjm/metric.html

<https://github.com/mariazevedo88/o3smeasures-tool>