|  |  |
| --- | --- |
|  | Total de defectos detectados por revisión |
|  | 1 |
| Informe Jefe de Proyecto | 5 |
| Informe Plan de Proyecto | 2 |
| Plan de Riesgos | 1 |
| Plan de Comunicación | 1 |
| Informe Gestor de Configuración | 2 |
| Informe Analista-Arquitecto-Diseñador | 3 |

**Correcciones en el informe de Jefe de Proyecto:**

1,2 - Eliminar del documento lo referido a trabajo en equipo y preguntas de liderazgo, ya que el equipo es de una sola persona.

3 - No se corresponde los resultados del test de Belbin con la preferencia de roles.

4 - Se especificaron tareas ya cumplidas, pero no funcionalidades del software ya cumplidas.

5 – Se confunden funcionalidades del software a implementar con tareas.

**Correcciones en el informe de Planificador:**

1 - Existen tareas que no forman parte de ninguna Macrotarea

2 - Revisar la ortografía. La tarea Realizar entregables es ambigua. La tarea Programar reuniones con el tutor está mal redactada.

**Correciones de Plan de Riegos:**

1 - El plan de contingencia no se pone en el plan de proyecto, hasta tanto no ocurra el riesgo.

**Correciones de Plan de Comunicación:**

1 - Faltan datos propios de la tesis (código fuente, capítulos de tesis). Algunos de los datos deben almacenarse en el repositorio de control de versiones de su proyecto.

**Correciones de Informe Gestor de la Configuración:**

1 - No queda totalmente claro el convenio utilizado

2 – No aparecen tareas en el plan.

**Correciones de Informe Analista-Arquitecto-Diseñador:**

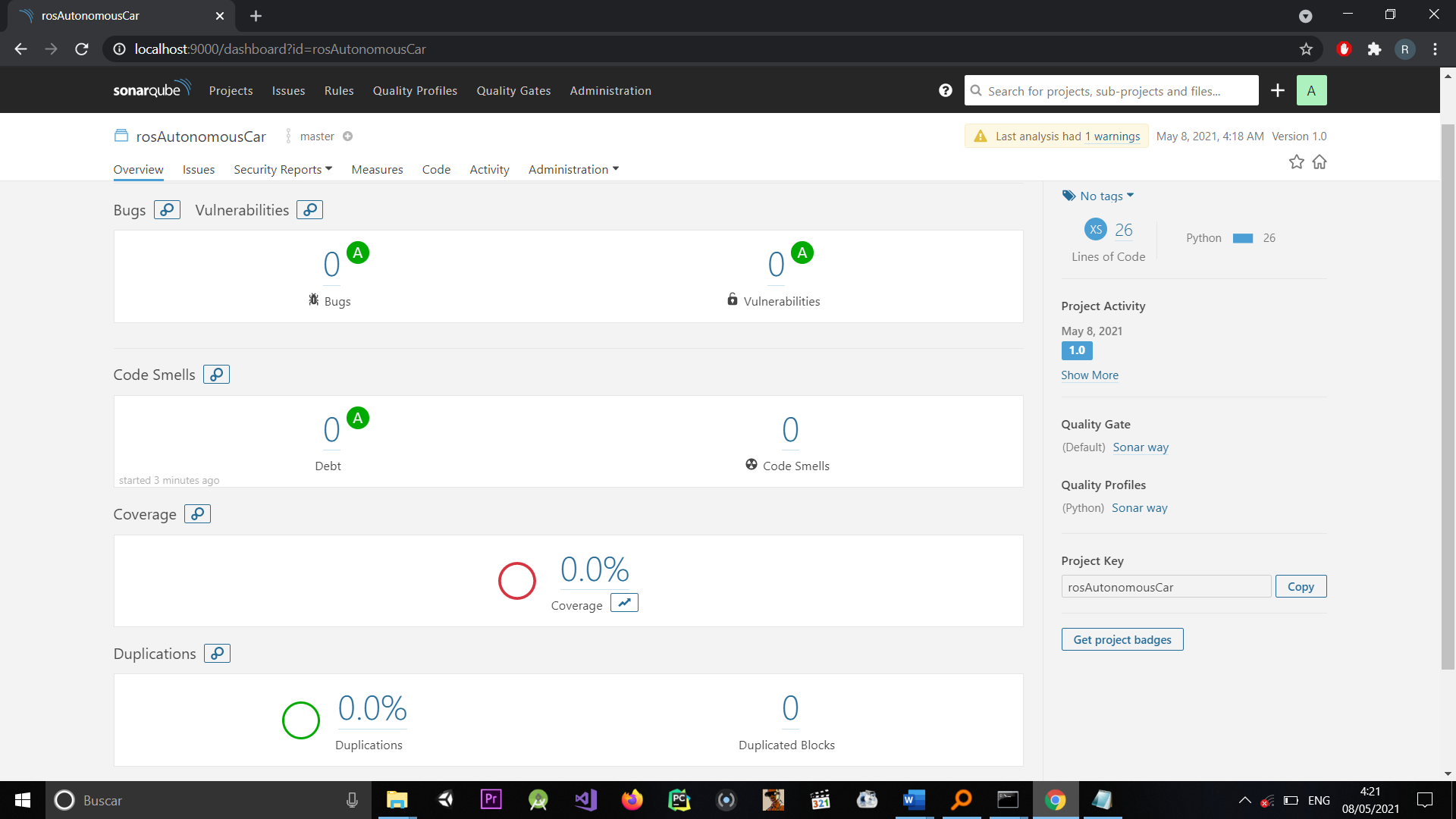
1 - Existe problema en la dirección de las relaciones *extend*

2 - Debe organizarse mejor para que se entienda, no solo poner el diagrama sino explicarlo y ponerlo en un orden lógico.

3 – Abordar otros elementos del estándar de programación.

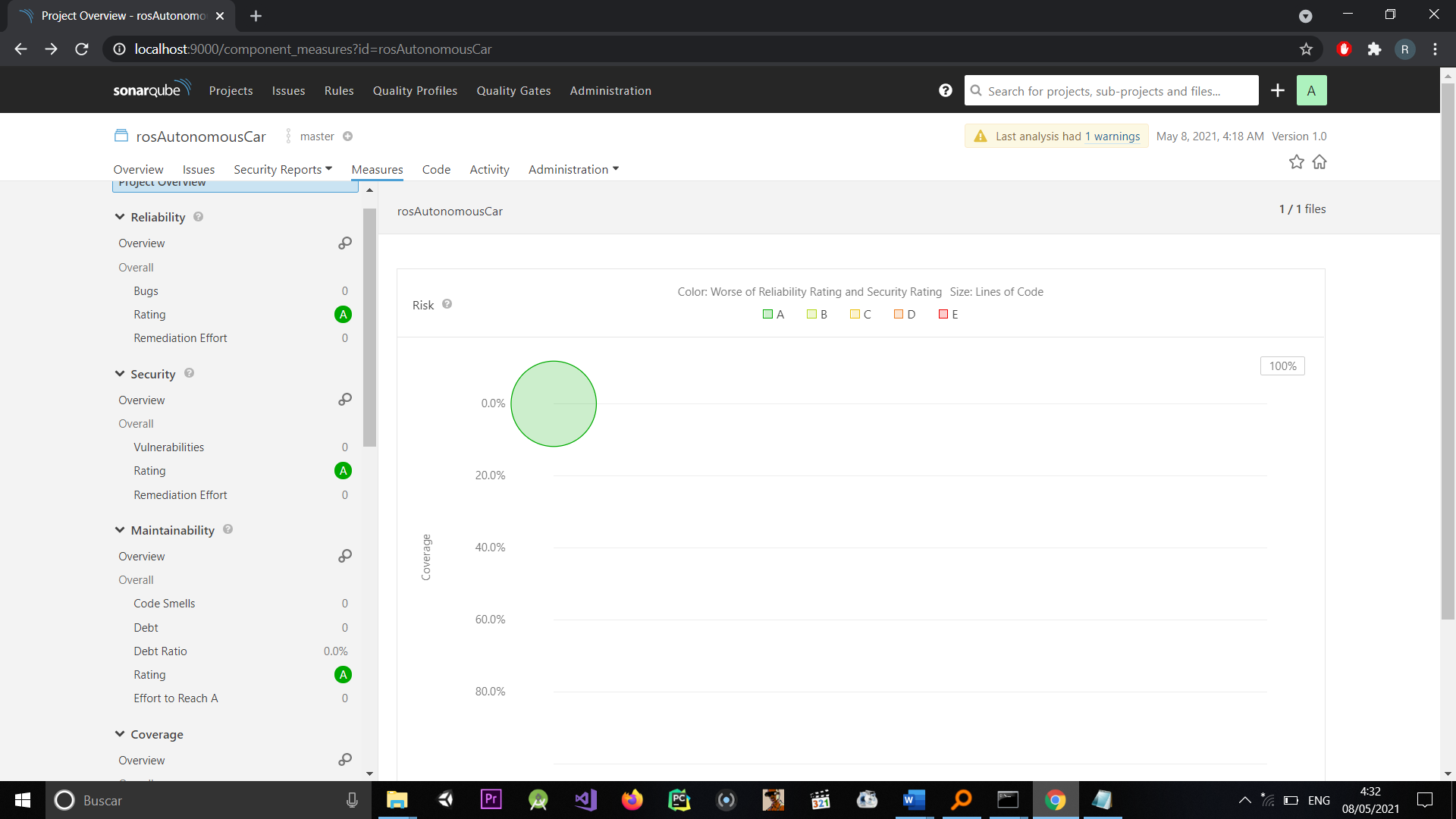
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Métrica** | **Tiempo**  **Planificado(Días)** | **Tiempo Real (Días)** | **Tiempo (%)** | **Defectos** | **Defectos (%)** |
| Jefe de proyecto | 3 | 2 | 66 | 5 | 41 |
| Planficador | 5 | 4 | 80 | 2 | 16 |
| Gestor de la configuración | 2 | 2 | 100 | 2 | 16 |
| Analista-Arquitecto- Diseñador | 4 | 2 | 50 | 3 | 25 |
| Programador | 3 | 2 | 50 | No ha sido revisado | - |
| Probador-Calidad | 2 | 1 | 50 | No ha sido revisado | - |

Análisis del proyecto en SonarQube, como se evidencia no se aprecia errores. Se analizo el archivo “patrol.py”

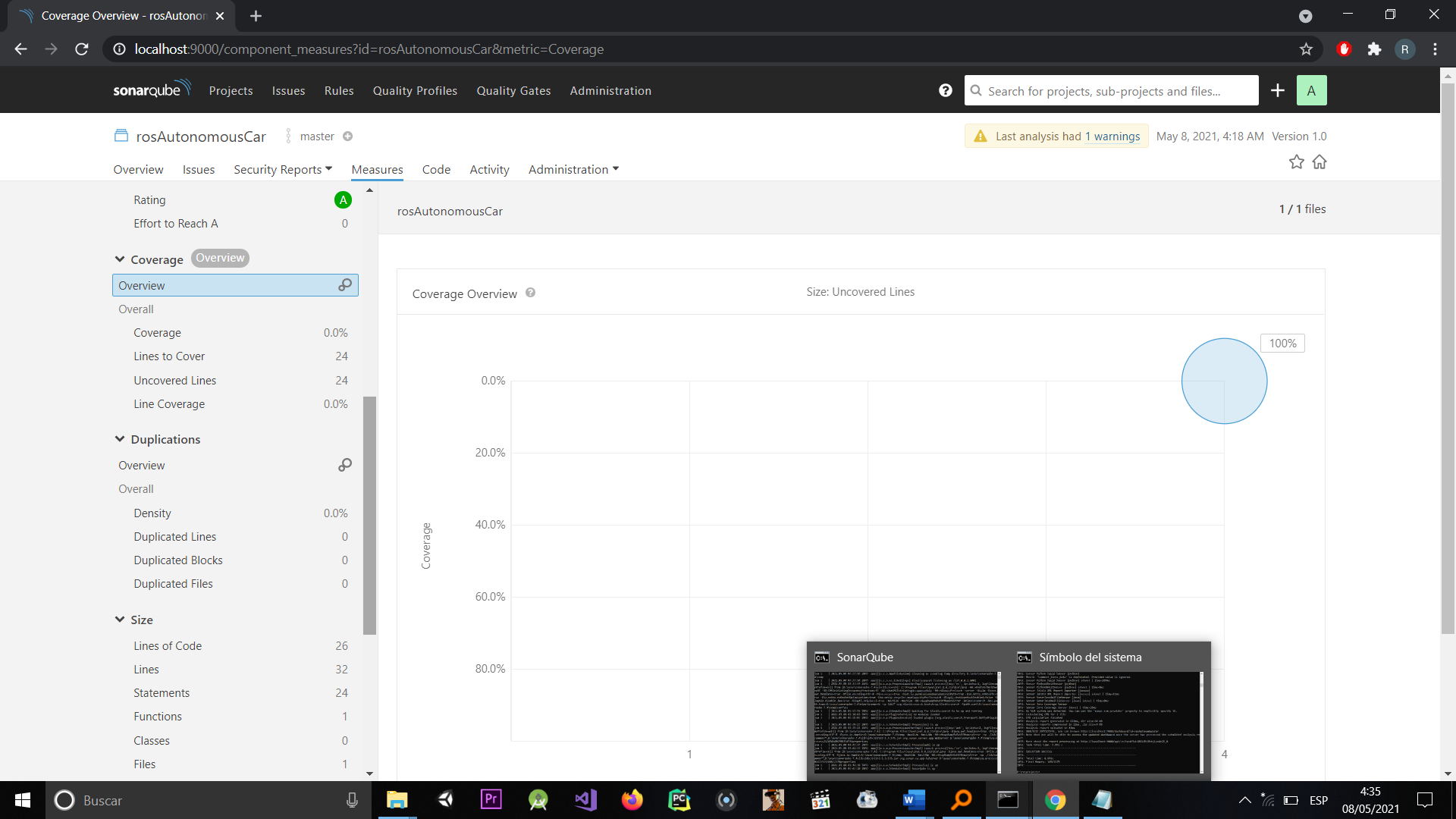


Métricas de SonarQube:

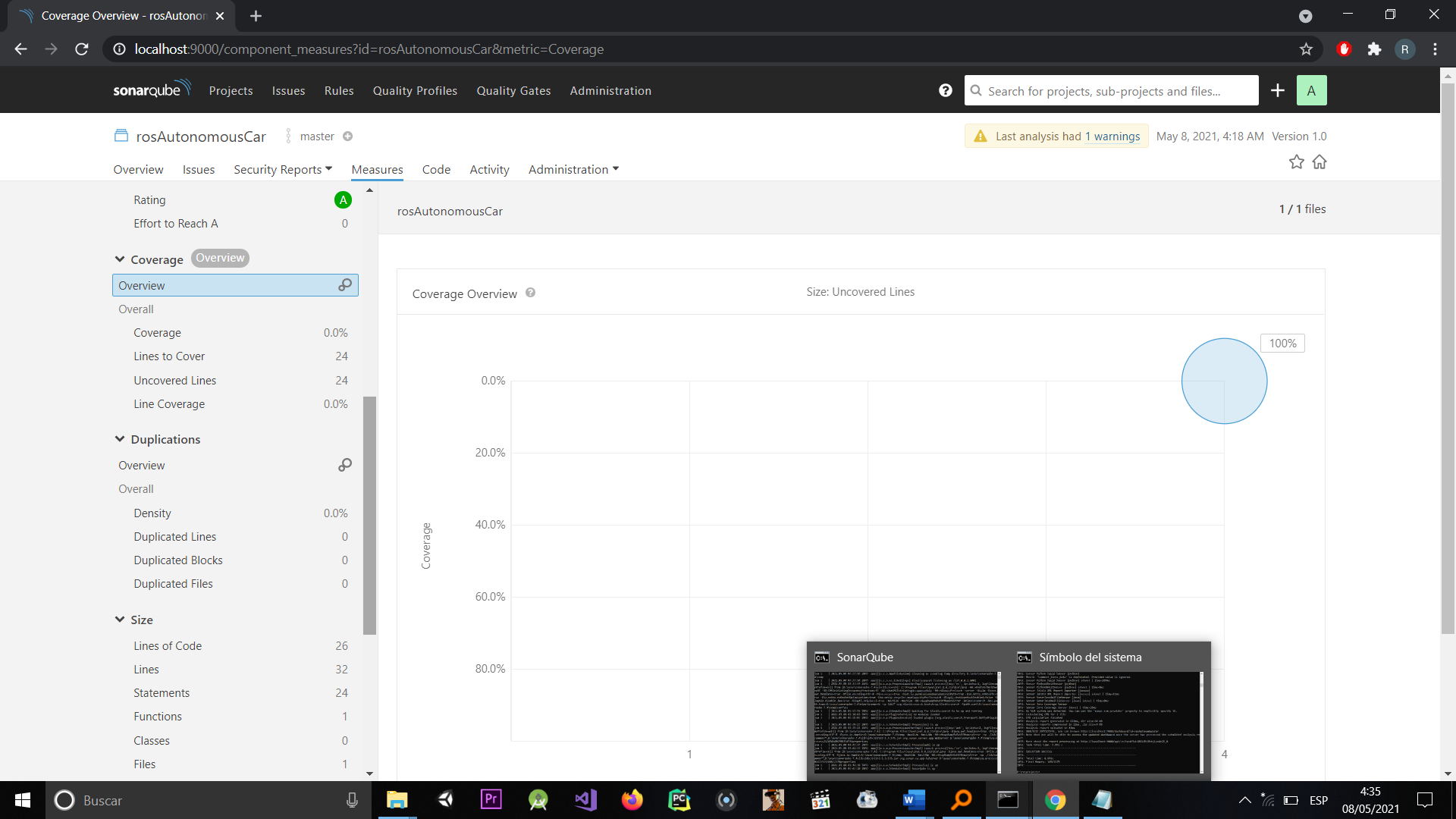
Tanto en Reliability, Security y Maintainability se obtuvo una calificación “A” lo que indica que no se encontraron bugs.



En el caso de Duplications se encuentra que no existen líneas duplicadas, por lo que se supone una buena optimización del código.



Con respecto a la métrica Coverage se encuentra que no existen líneas cubiertas.



En el caso de Complexity: se observa la complejidad ciclomatica más alta como 4, ya que al ser ROS un framework que provee un gran número de herramientas, no resulta complejo la programación de los nodos.

Para la realización de pruebas, se comprobó el correcto funcionamiento del nodo. Para esto se analizo los mensajes siendo publicados por el mismo en el topic “/move\_goal”, con el comando “rostopic echo /move\_ goal” y se comprobó que al seleccionar el punto, rViz mostrara el indicador de destino.