|  |
| --- |
| XX-PROJECTNAME-XX  Version XX-VERSION-XX  Code analysis |

|  |
| --- |
| **By: XX-AUTHOR-XX**  **XX-DATE-XX** |

# Content

[Content 1](#_Toc41730784)

[Introduction 2](#_Toc41730785)

[Synthesis 2](#_Toc41730786)

[Language 3](#_Toc41730787)

[Issues 3](#_Toc41730788)

[Charts 3](#_Toc41730789)

[Owasp metrics 4](#_Toc41730790)

[Owasp improvements 4](#_Toc41730791)

# Introduction

This document contains results of the code analysis of XX-PROJECTNAME-XX.

XX-DESCRIPTION-XX

# Synthesis

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Quality Gate | Reliability | Security | | Maintainability | | Duplication |
| XX-QUALITYGATE-XX | **XX-RELIABILITY-XX** | | **XX-SECURITY-XX** | | **XX-MAINTAINABILITY-XX** | **XX-DUPLICATION-XX %** |

The synthesis chart tries to provide a global vision and coverage of the project, and rates from 1 to 5 based in the following conversion:

**A**: 5  
**B**: 4  
**C**: 3  
**D**: 2  
**E**: 1

**QUALITY GATE:** It's there to answer ONE question: can I deliver my project to production today or not ? In order to answer this question, there are several conditions which the project is measured. To get aproved, the overall rate should be at least 4 to 5.

**RELIABILITY :** rates the codes reliability by analyzing the code’s bugs and the potential problems it could develope.

**SECURITY :** it refers to the issues that could be potential weaknesses.

**MAINTAINABILITY:** rates how the project could scale and mantain in the future. There could be issues that are not dangerous at the moment, but could potentially develope as important weaknesses.

**DUPLICATIONS :** shows the percetage of identical lines of code.

# Language

|  |
| --- |
| $VOLUME |

# Issues

|  |
| --- |
| $ISSUES\_DETAILS |

While running an analysis, SonarQube raises an issue every time a piece of code breaks a coding rule, and are initially classified according to their type and severity.

There are 4 types :

* **Bug**: A coding error that will break your code and needs to be fixed immediately.
* **Vulnerability**: A point in your code that's open to attack.
* **Code Smell**: A maintainability issue that makes your code confusing and difficult to maintain.
* **Security Hotspot** highlights a security-sensitive piece of code that needs to be checked. Upon review, you'll either find there is no threat, or you need to apply a fix to secure the code.

There are 5 severities:

* **BLOCKER**  
  Bug with a high probability to impact the behavior of the application in production. The code MUST be immediately fixed.
* **CRITICAL**  
  Either a bug with a low probability to impact the behavior of the application in production or an issue which represents a security flaw: empty catch block, SQL injection, ... The code SHOULD be immediately reviewed.
* **MAJOR**  
  Quality flaw which can highly impact the developer productivity: uncovered piece of code, duplicated blocks, unused parameters, ...
* **MINOR**  
  Quality flaw which can slightly impact the developer productivity: lines should not be too long, "switch" statements should have at least 3 cases, ...
* **INFO**  
  Neither a bug nor a quality flaw, just a finding.

# Charts

From now on, the next charts are going to be focused in security aspects. Therefore, the following chart shows the issues with an "owasp rule” attached to it alongside different metrics based on [OwaspTop10](https://owasp.org/www-project-top-ten/) page. For more specific metrics, go to [BSM](https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator) page.

# Owasp metrics

|  |
| --- |
| $IMPROVEMENTS |

Finally, it is shown the issues founded on the project, with the OwaspTop10 rule detected. It will show the file and the specific lines where the issue was founded, the owasp type, a brief explanation of the Owasp rule and its final mitigation. For further information click [here](https://owasp.org/www-project-top-ten/).

# Owasp improvements

|  |
| --- |
| $OWASP |