

# Mejoramiento de la Aplicación “La Canasta”



Integrantes:

- Corrales Solis, Moisés Alessandro - 2020067579.
- Valle Bustamante, Gustavo Alonso - 2020066916.
- Chambilla Mardinez, Renato Eduardo - 2020066918
- Viveros Blanco, Farley Rodrigo Eduardo - 2020066896

Docente: Ing. Patrick Cuadros Quiroga

Curso: Calidad y pruebas de software

Código de curso: SI-784

Facultad: Facultad de Ingeniería

Escuela: Escuela Profesional de Ingeniería de Sistemas (EPIS)

TACNA - PERÚ

2023

## Resumen:

Un proyecto de análisis y mejora de software tiene como objetivo mejorar la calidad y eficiencia del software existente mediante el análisis de su diseño, código, funcionalidad y usabilidad, se deben identificar las deficiencias del software existente. En este caso, para el análisis del código y mejora se usó SonarQube, el cual es una herramienta de código abierto utilizada para la gestión de calidad de código. Permite a los equipos de desarrollo de software analizar y mejorar la calidad del código fuente de manera continua durante todo el ciclo de vida del software.

## Abstract:

A software analysis and improvement project aims to improve the quality and efficiency of existing software by analyzing its design, code, functionality and usability, identifying the deficiencies of existing software. In this case, for code analysis and improvement we used SonarQube, which is an open source tool used for code quality management. It allows software development teams to analyze and improve the quality of the source code continuously throughout the software lifecycle.

### 1. Antecedentes o introducción:

En el presente documento se realizará el análisis del proyecto “La Canasta” por medio de sonarqube, en el cual se verán los fallos o deficiencias del código a mejorar para reducirlas lo máximo posible, viendo posibles soluciones o propuestos para ir arreglando el proyecto poco a poco.

### 2. Título:

El título del proyecto es “Análisis y mejora de la aplicación La Canasta”.

### 3. Autores

- Moisés Alessandro Corrales Solis.
- Renato Eduardo Chambilla Mardinez
- Farley Rodrigo Eduardo Viveros Blanco
- Gustavo Alonso Valle Bustamante.

#### 4. Planteamiento del problema:

##### 4.1. Problema:

El problema a resolver implica identificar los problemas existentes en el software y cómo afectan a la eficacia y eficiencia del sistema “La canasta”. Algunos ejemplos de problemas comunes incluyen los bugs, errores en algunas funcionalidades, lentitud por falta de optimización del sistema, vulnerabilidades, entre otros más. El proyecto tuvo pruebas en las cuales llega a tener algunos errores esperados e inesperados, debido a diversos factores que lo conforman.

##### 4.2. Justificación:

La importancia de tener un sistema seguro y competente, actualmente se ha vuelto un punto muy importante dentro del software, ya que de ello depende la privacidad y confidencialidad de la empresa, y si por alguna razón, este se llegara a vulnerar, la empresa podría perder datos muy importantes.

Por ende, el análisis con Sonarqube, nos ayudara a mejorar ciertos aspectos en los cuales podemos mejorar, haciendo así, que todo funcione correctamente y que cumpla con los requerimientos solicitados por el cliente, evitando obstáculos o inconvenientes en el futuro, sirviendo además como aprendizaje.

##### 4.3. Alcance:

El proyecto tendrá alcance y beneficiará a la pollería “La Canasta” por el momento, ya que el software está dirigido para esa empresa, cumpliendo con todas las funcionalidades requeridas, en la cual también beneficiará a toda la clientela de dicha empresa.

#### 5. Objetivos:

##### 5.1. General:

- Analizar e identificar los errores del sistema con SonarQube, para así, mejorar en los aspectos que estamos fallando y optimizar el sistema.

##### 5.2. Específicos:

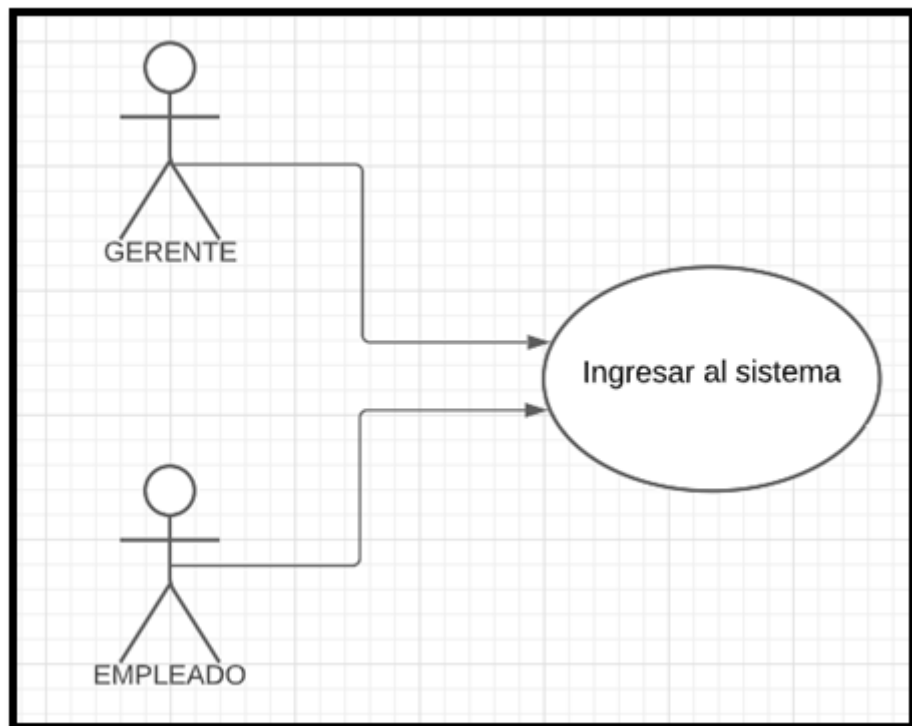
- Establecer un plan de mejora, ordenando por orden de prioridad los aspectos que debemos corregir, mejorar y optimizar.

- Realizar o implementar las mejoras del sistema.
- Realizar pruebas para verificar mejoras hechas, y finalmente, hacer la prueba de software.

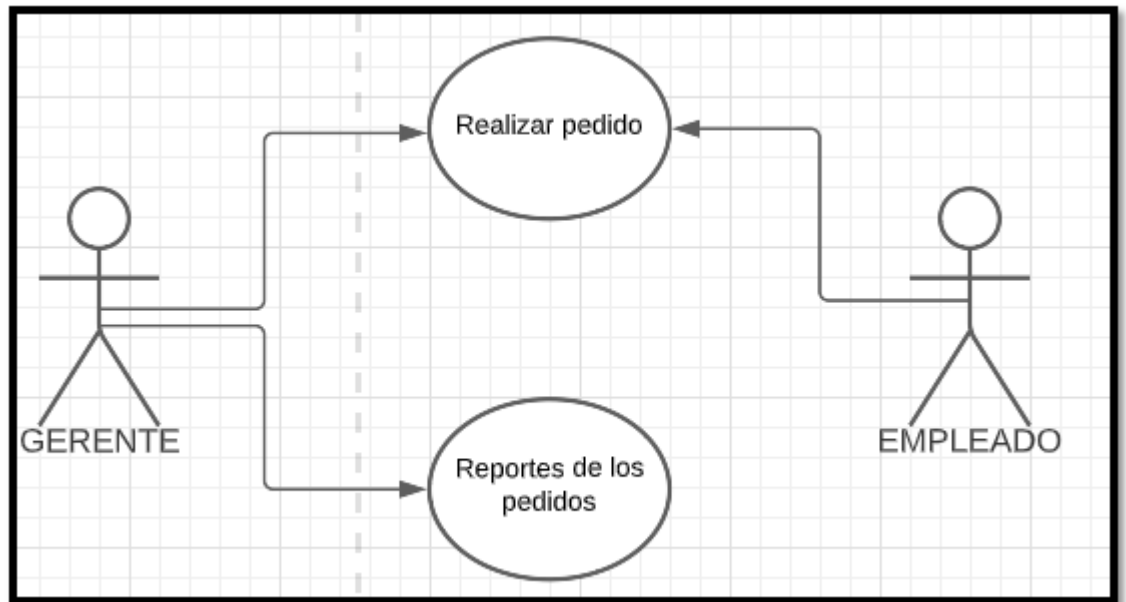
## 6. Referentes teóricos

- Diagramas de Casos de Uso:

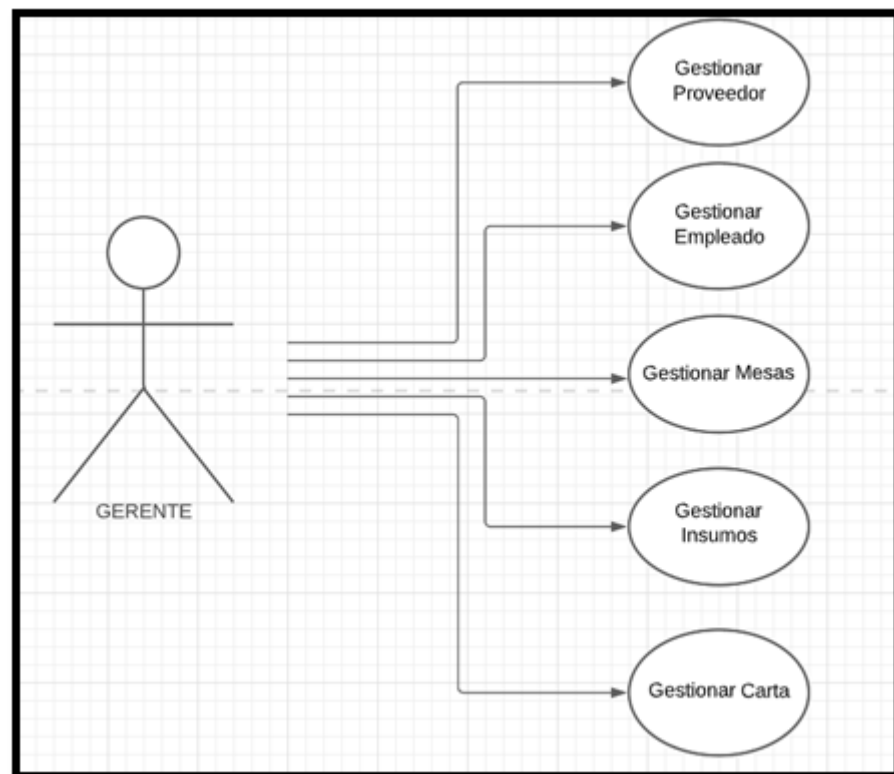
Seguridad:



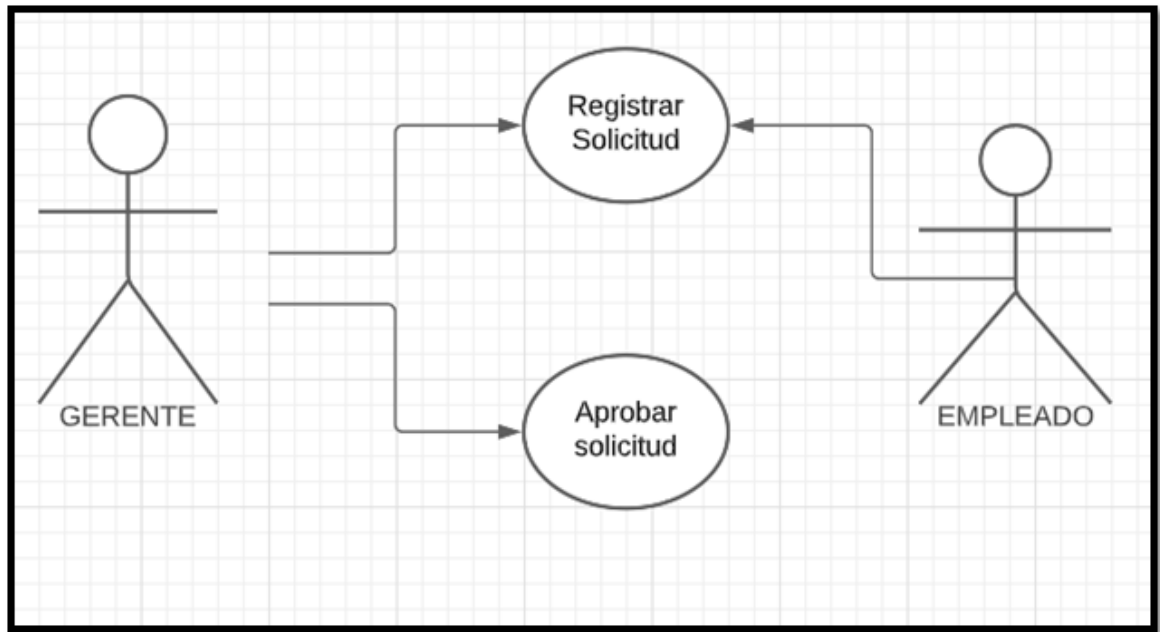
Ventas:



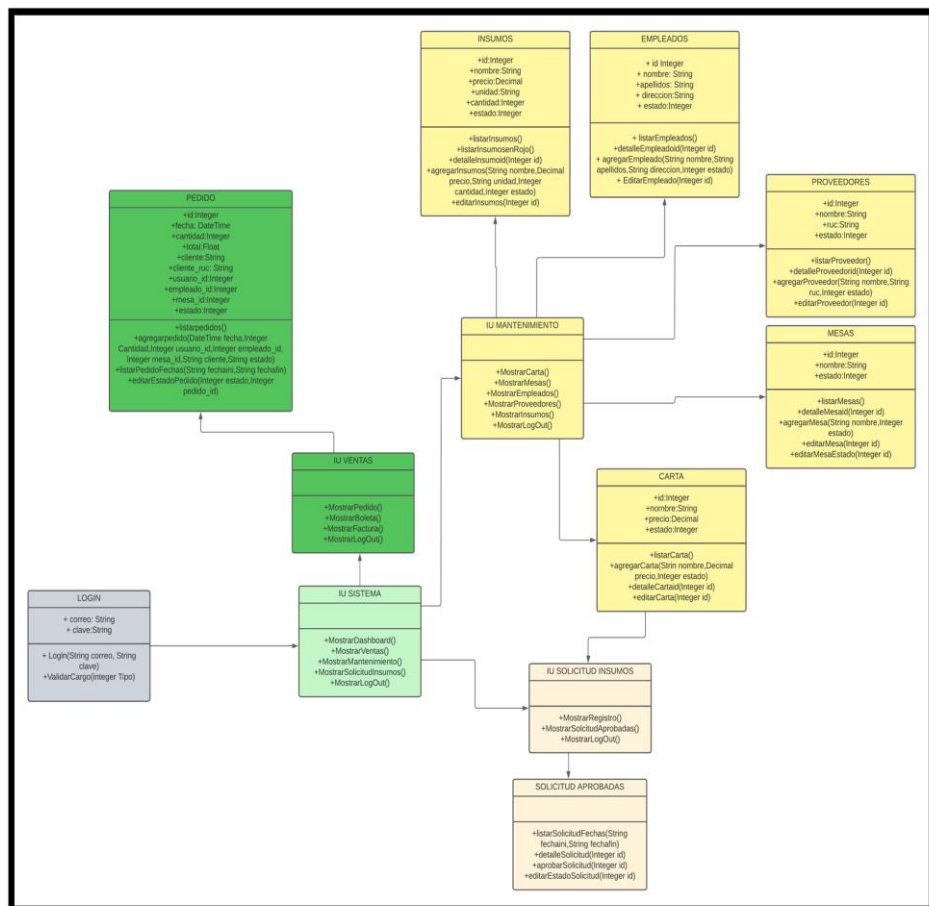
Mantenimiento:



Solicitud de insumos:



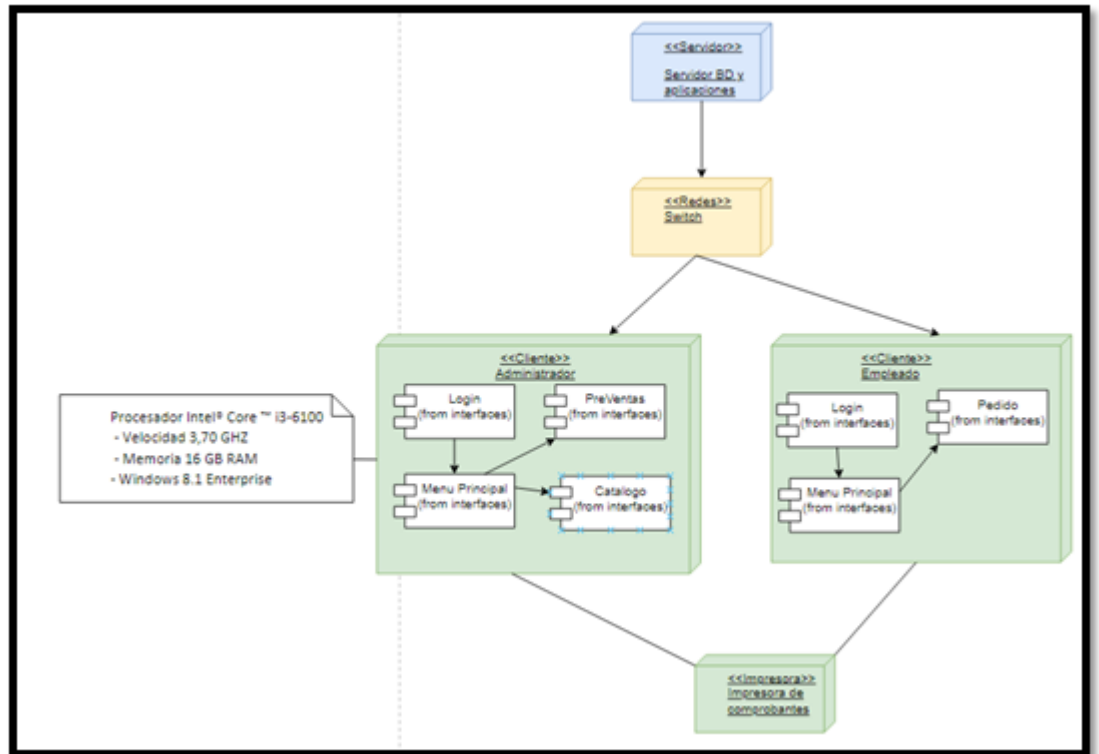
- Diagrama de Clases:



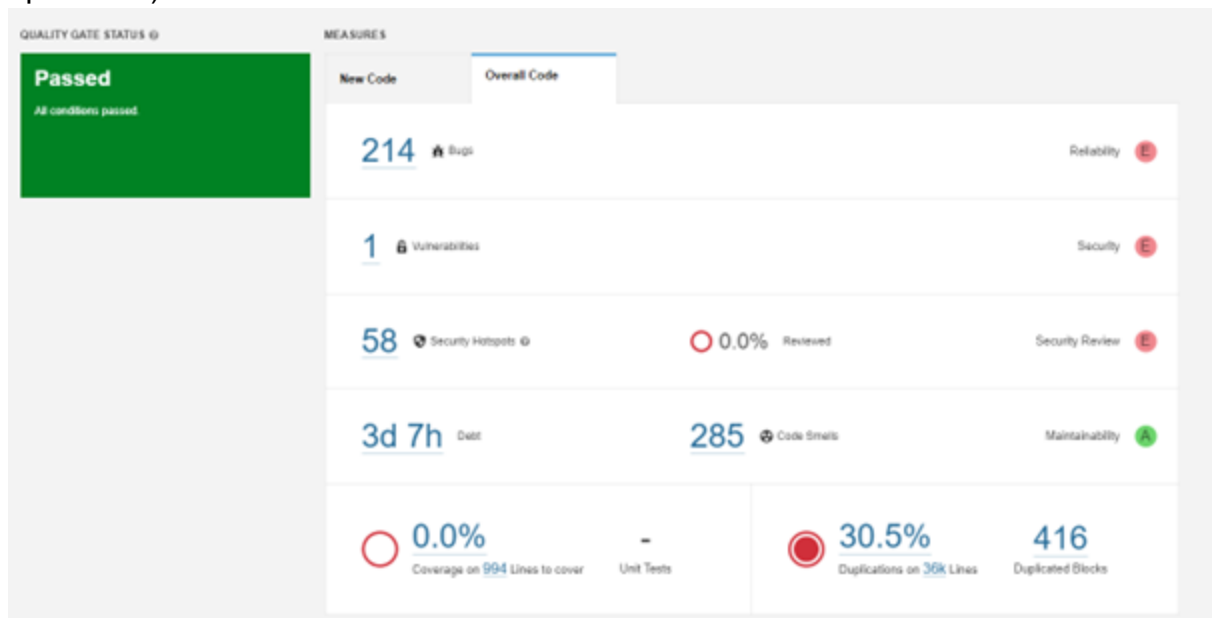
- Diagrama de Componentes:

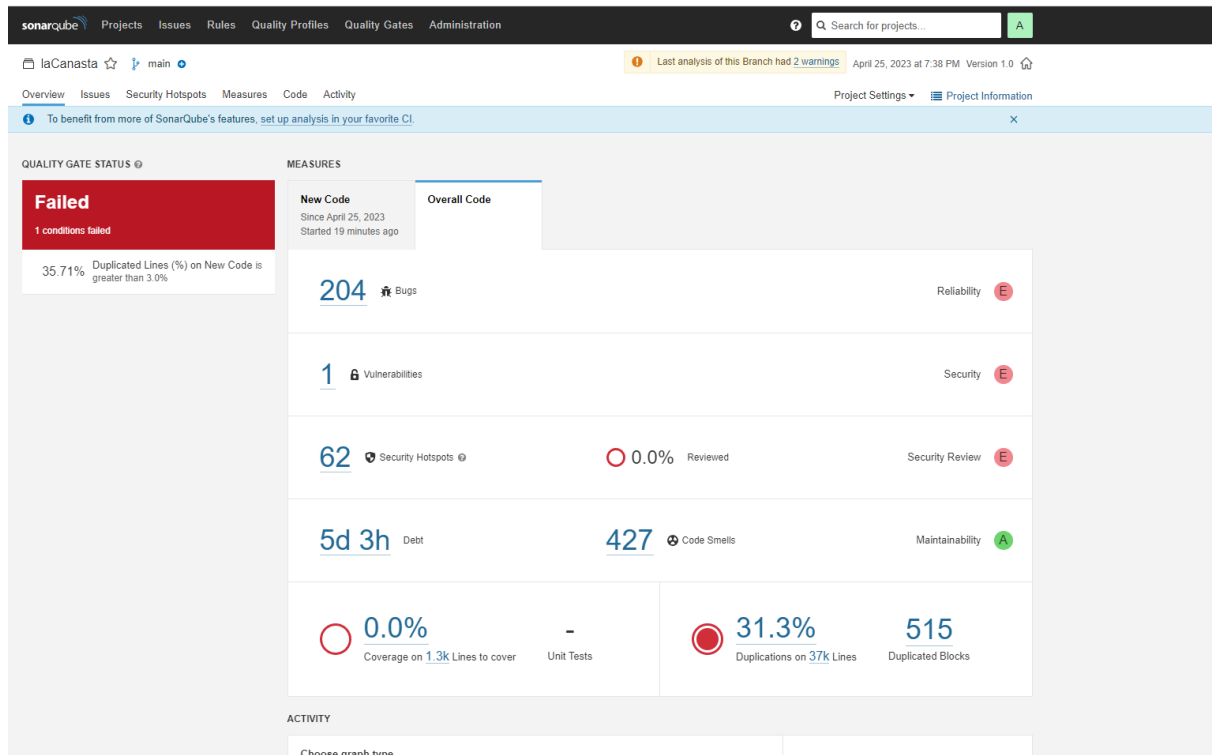
Nuestro proyecto solo conlleva un componente.

- Diagrama de Arquitectura:



- Desarrollo de la propuesta (Aquí va el análisis de su aplicación con SonarQube, para que les muestre todos los aspectos a mejorar de su aplicación)





## 7.1. Tecnología de información

**SONARQUBE:** Nos permitirá realizar un análisis estático de código. Lo que hace es identificar los puntos susceptibles de mejora, que facilitarán la obtención de métricas necesarias para la optimización del código.

**SONARSCANNER:** Nos ayudara a obtener las métricas para poder visualizar la cantidad de código que podemos mejorar ya sean, bugs, errores, vulnerabilidades, optimizaciones, etc.

**NETBEANS:** Nos ayuda en el desarrollo de cualquier tipo de software y en sus distintas fases. Sirve como medio para la escritura, compilación o, simplemente, para hallar errores en programas basados en los anteriores lenguajes.

## 7.2. Metodología, técnicas usadas

**Pruebas unitarias automatizadas:** Simplificará el trabajo dispendioso, repetitivo o complejo, haciéndolo efectivo y más productivo. De esta manera, es posible ahorrar energía, tiempo y costos asociados. A su vez, liberar de tiempo valioso a los testers para que se concentren en otras tareas.

**Análisis estático:** Una herramienta que permite a los desarrolladores identificar y corregir los fallos de seguridad en su código, haciendo así que este sea mas eficiente y eficaz, mejorando el proyecto para evitar errores, vulnerabilidades, etc.



8. Cronograma (personas, tiempo, otros recursos) Basado en las observaciones que la herramienta SonarQube les informará sobre la aplicación, a fin de reducir la deuda técnica, vulnerabilidades, fallas, etc. a 0.

Dia/ Problemas	Bugs	Hostpots
Dia 1	41 bugs	12 Hostpots
Dia 2	41 bugs	12 Hostpots
Dia 3	41 bugs	12 Hostpots
Dia 4	41 bugs	13 Hostpots
Dia 5	41 bugs	13 Hostpots
Corrección total	204 bugs corregidos	62 hostpots