**SISTEMA DE GESTIÓN DE TESIS**

CALIDAD DEL PRODUCTO SOFTWARE

**Versión <1.0>**

HISTORIAL DE REVISIONES DEL DOCUMENTO

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 07/12/2019 | 1.0 | Elaboración del documento. | Cristian Tineo Guevara |
|  |  |  |  |

TABLA DE CONTENIDO

[1. Introducción 4](#_Toc26869689)

[1.1. Objetivo 4](#_Toc26869690)

[1.2. Propósito 4](#_Toc26869691)

[2. RESULTADOS DEL ANALISIS DE CALIDAD CON SONARQUBE 4](#_Toc26869692)

[2.1. VISIÓN GENERAL DEL PRODUCTO SOFTWARE 4](#_Toc26869693)

[2.2. MEDIDAS DE FIABILIDAD 4](#_Toc26869694)

[2.3. MEDIDAS DE SEGURIDAD 5](#_Toc26869695)

[2.4. MEDIDAS DE MANTENIMIENTO 6](#_Toc26869696)

[2.5. MEDIDAS DE COBERTURA 7](#_Toc26869697)

[2.6. MEDIDAS DE DUPLICACIONES 8](#_Toc26869698)

[2.7. TAMAÑO DEL PROGRAMA 9](#_Toc26869699)

[2.8. COMPLEJIDAD CICLOMATICA 9](#_Toc26869700)

[2.9. COMPLEJIDAD COGNITIVA 10](#_Toc26869701)

[3. RESUMEN 10](#_Toc26869702)

[4. CONCLUSIONES 11](#_Toc26869703)

[5. RECOMENDACIONES 11](#_Toc26869704)

CALIDAD DEL PRODUCTO SOFTWARE

# Introducción

La calidad del producto software, es el grado en el que el producto software incorpora un conjunto de características, definidas por la organización, de tal manera que se garantiza eficiencia de uso, respecto a los requerimientos de los clientes.

## Objetivo

Este documento proporciona una visión y descripción acerca de la calidad del producto software desarrollado.

## Propósito

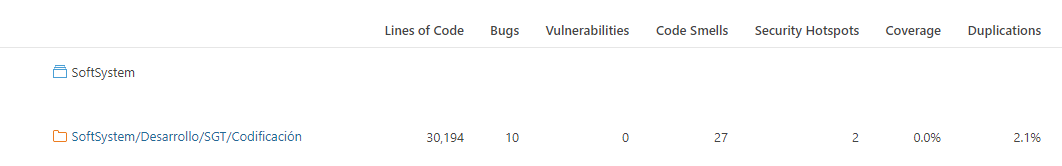
Dar a conocer al personal y al cliente, la calidad del producto que se está desarrollando para poder satisfacer las necesidades del cliente.

# RESULTADOS DEL ANALISIS DE CALIDAD CON SONARQUBE

## VISIÓN GENERAL DEL PRODUCTO SOFTWARE

El proyecto “Sistema de Gestión de Tesis” esta basado en los siguientes lenguajes:





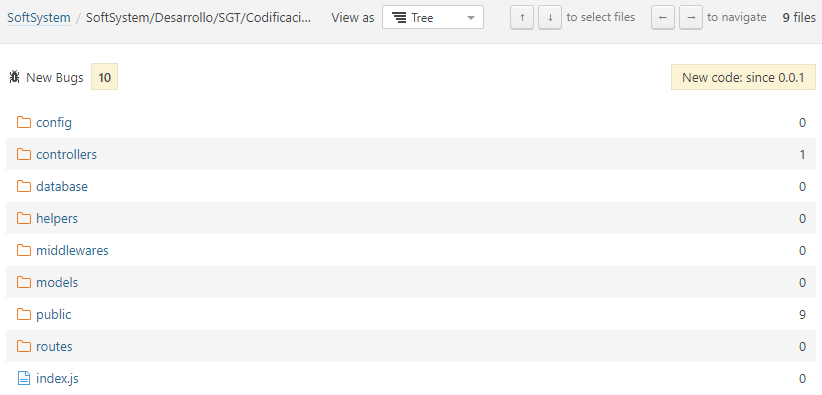
Además, la totalidad de líneas de código fuente alcanza la cifra de 30 mil, obteniendo 24 mil en CSS y 6 mil en JavaScript.

## MEDIDAS DE FIABILIDAD

El proyecto “Sistema de Gestión de Tesis” contiene 10 Bugs, por ese motivo obtuvo una calificación C, ya que esta calificación indica que por lo menos existe 1 error mayor.



Los Bugs se encuentran localizados en las siguientes rutas, las cuales la mayor cantidad de bug se encuentran en la carpeta public.

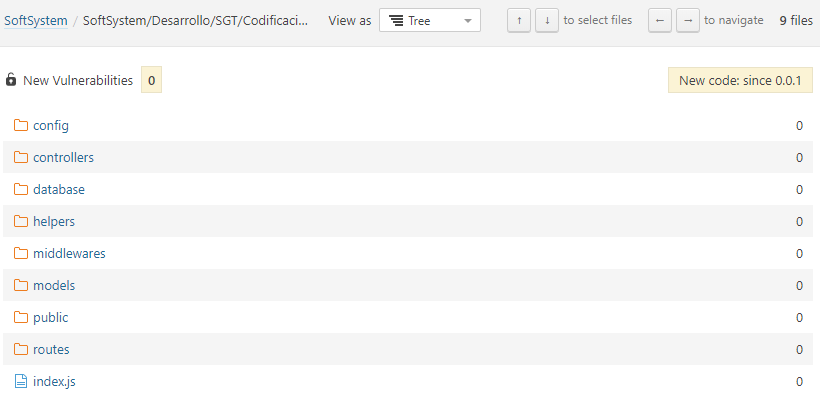


## MEDIDAS DE SEGURIDAD

El proyecto “Sistema de Gestión de Tesis” no tiene ninguna vulnerabilidad por lo que es seguro usar el sistema, además cuenta con 2 puertas de seguridad para impedir el robo de alguna información. Por ese motivo se obtuvo la calificación A que quiere decir que en este proyecto no existe vulnerabilidades.



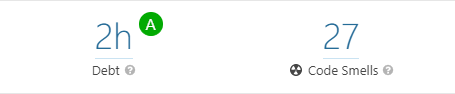
Aquí se muestra la ruta, en las cuales se aprecia que en ninguna carpeta existe alguna vulnerabilidad que pueda afectar la seguridad del sistema.



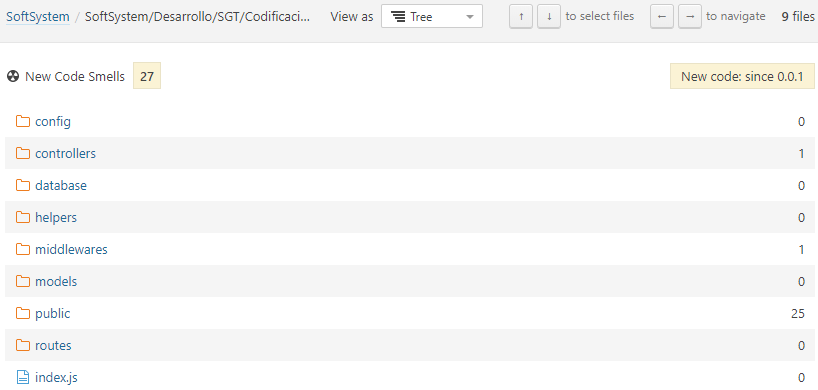
## MEDIDAS DE MANTENIMIENTO

El proyecto “Sistema de Gestion de Tesis” tiene 27 ***códigos olorosos*** los cuales nos indican que existen 27 códigos que son un tanto difíciles de mantener y por lo cual dependerá de este para poder hacer el mantenimiento al software debido a su complejidad.

En esta calificación obtuvo el puntaje de A, debido a que el costo de mantenimiento es solo de 2 horas.

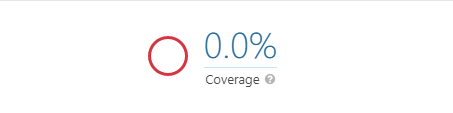


A continuación, se muestra las rutas en las cuales se encuentran estos “códigos olorosos”, donde se observa que la mayor cantidad de este tipo se encuentra en la carpeta public:



## MEDIDAS DE COBERTURA

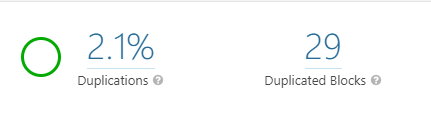
El proyecto “Sistema de Gestion de Tesis” está basado en los siguientes lenguajes:



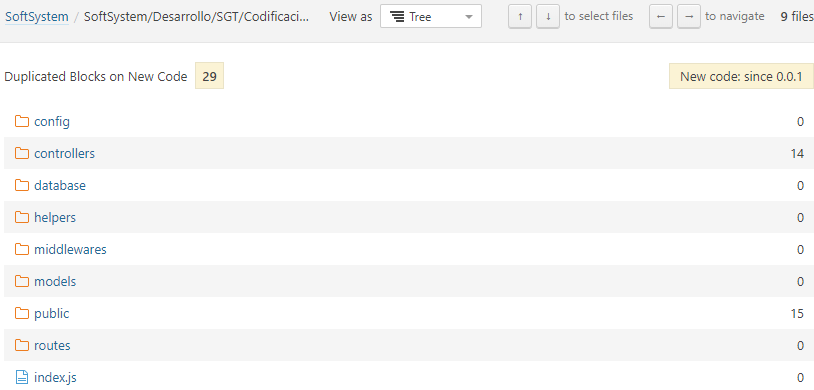
Además, la totalidad de líneas de código fuente alcanza la cifra de 30 mil, obteniendo 24 mil en CSS y 6 mil en JavaScript.

## MEDIDAS DE DUPLICACIONES

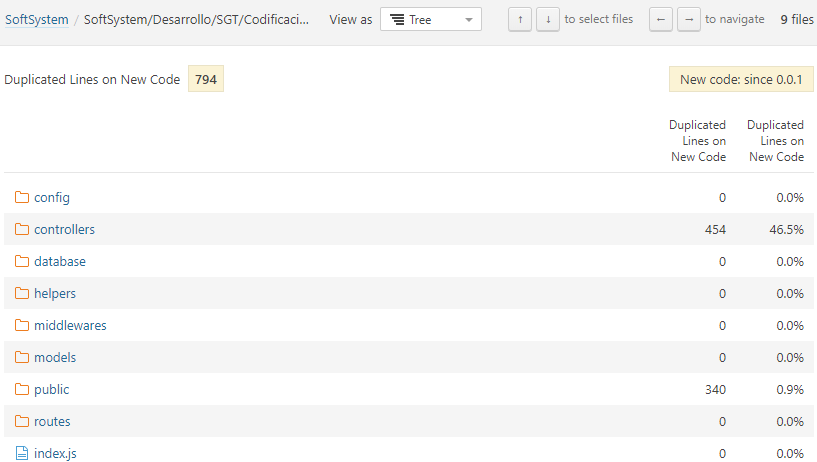
El proyecto “Sistema de Gestion de Tesis” tiene 29 bloques de códigos duplicados, lo cual en total del proyecto demostraría ser el 2.1%.



La cantidad de bloques de código duplicado se encuentra distribuido en las siguientes carpetas, en la que los mayores bloques se ubican en la carpeta public:



La cantidad de líneas de código duplicado se encuentra distribuido en las siguientes carpetas, en las cuales el mayor porcentaje la tiene la carpeta controllers debido a que el código se “recicla” para muchas funcionalidades:



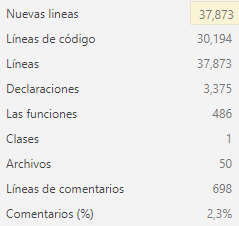
## TAMAÑO DEL PROGRAMA

El proyecto “Sistema de Gestion de Tesis” esta basado en los siguientes lenguajes.

Además, la totalidad de líneas de código fuente alcanza la cifra de 30 mil, obteniendo 24 mil en CSS y 6 mil en JavaScript.

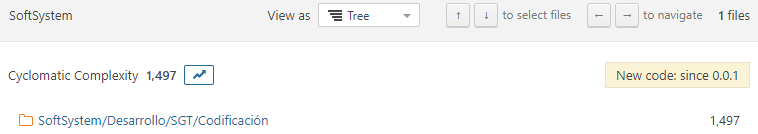


Aquí se muestra una estadística del contenido del proyecto:



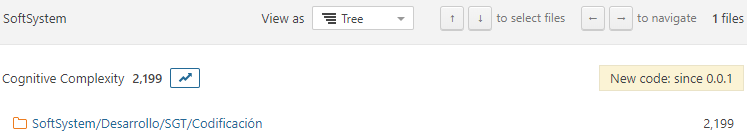
## COMPLEJIDAD CICLOMATICA

El proyecto “Sistema de Gestión de Tesis”, tiene una complejidad ciclomatica de 1497. La cual es la Complejidad Ciclomática calculada en función del número de rutas a través del código. Siempre que el flujo de control de una función se divide, el contador de complejidad se incrementa en uno. Cada función tiene una complejidad mínima de 1.



## COMPLEJIDAD COGNITIVA

El proyecto “Sistema de Gestión de Tesis”, tiene una complejidad cognitiva de 2199. La cual nos demuestra qué difícil es comprender el flujo de control del código.



# RESUMEN

|  |  |
| --- | --- |
| **MÉTRICA** | **DESCRIPCIÓN** |
| **LINEAS DE CODIGO** | **30194** |
| **CONFIABILIDAD** | **10** |
| **SEGURIDAD** | **0 Vulnerabilidades** |
| **MANTENIBILIDAD** | **27 Code Semells** |
| **DUPLICADO**  **(10% < x < 15%)** | **2.1% Duplicado** |
| **COMPLEJIDAD CICLOMATICA** | **1497** |
| **COMPLEJIDAD COGNITIVA** | **2199** |
| **COMENTARIOS**  **(10% < X < 20%)** | **2.1%** |
| **PUERTA DE CALIDAD** |  |

# CONCLUSIONES

Si bien el programa funciona correctamente, este debe estar sujeto a revisiones frecuentes para reducir la cantidad de bugs y que el producto pueda tener mejor calidad y mantenibilidad.

Además, el programa presenta alta complejidad ciclomatica y cognitiva, lo cual implicaría mayor esfuerzo y tiempo en poder comprender el programa para tener un mejor mantenimiento.

# RECOMENDACIONES

Las recomendaciones son las siguientes:

* Reducir la complejidad del código.
* Realizar más comentarios para poder hacer el código más entendible y que el código sea mas mantenible.
* Realizar las observaciones futuras para la mejorar la calidad total del sistema.
* Posterior a la mejora de calidad, realizar las pruebas de caja blanca y caja negra.