**Informe técnico sobre análisis de código estático para el software**

**Sistema Contable**

**Versión 1.0**

**Perito:**

**Nombre: Yeslie Guisel Mestizo Montaño**

**Cedula: 1032494104**

**Email: yeslie.mestizo@usantoto.edu.co**



**Contenido del informe**

[1. Propósito de la evaluación 2](#_Toc42772819)

[2. Producto a evaluar 2](#_Toc42772820)

[3. Especificación de la(s) herramientas usadas para medición 2](#_Toc42772821)

[4. Especificación de la Norma de calidad 3](#_Toc42772822)

[5. Requerimientos específicos de calidad funcionales y técnicos. 4](#_Toc42772823)

[6. Atributos de calidad. 6](#_Toc42772824)

[6.1. Asignación de puntajes para atributos de calidad. 6](#_Toc42772825)

[6.2. Atributos internos. 6](#_Toc42772826)

[6.2.1. Tamaño de sistemas y código fuente (20%) 6](#_Toc42772827)

[6.2.2. Complejidad del software (20%) 7](#_Toc42772828)

[6.2.3. Deuda técnica (20%) 8](#_Toc42772829)

[6.2.4. Seguridad (20%) 9](#_Toc42772830)

[6.2.5. 3d code metrics (20%) 10](#_3d_code_metrics)

[6.3. Atributos Externos. 12](#_Toc42772832)

[6.3.1. Usabilidad (20%) 12](#_Toc42772833)

[6.3.2. Fiabilidad (30%) 13](#_Toc42772834)

[6.3.3. Mantenibilidad (50%) 13](#_Toc42772835)

[7. Recomendaciones y conclusiones 13](#_Toc42772836)

[7.1. Seguridad 13](#_Toc42772837)

[7.2. Confiabilidad 13](#_Toc42772838)

[7.3. Usabilidad 13](#_Toc42772839)

[8. Bibliografía 14](#_Toc42772840)

[9. Firma del perito. 15](#_Toc42772841)

# Propósito de la evaluación

La evaluación del software “sistema contable” se realiza para verificar la calidad del mismo, esto, teniendo en cuenta como primera instancia la norma de calidad ISO 25000, además de revisar cómo está respondiendo el software a 13 requerimientos específicos de calidad funcionales y técnicos. Adicional a esto y como parte importante, la evaluación se basa en tres atributos de calidad, los cuales son: internos, externos y en uso; para los cuales se asignan puntajes con los que se concluye el porcentaje de calidad del software, junto con recomendaciones especiales respecto a seguridad, confiabilidad y usabilidad. Los beneficios de esta evaluación son sus resultados, con los cuales se puede mejorar el software aumento su calidad, esto, debido a que conoceremos sus atributos, como por ejemplo el código mal oliente y las posibles vulnerabilidades de seguridad.

# Producto a evaluar

El software a evaluar tiene como propósito ayudar a sus usuarios con el control de sus finanzas personales, pudiendo así, cada usuario almacenar sus frecuentes ingresos y egresos en la categoría “Recurrente”, si el caso es para conceptos que no sean frecuentes, se añaden a la categoría de “No recurrente”. Además, este software cuenta con una sección de objetivos, en la cual sus usuarios pueden manejar un tipo de ahorro o meta si se le quiere llamar, la cual estaría destinada para la adquisición de un producto o servicio y para el cual el mismo software se encarga de restar el monto deseado, marcándolo como “Egreso para objetivo”, hasta que se cumpla dicho objetivo. La versión del software es 1.0 y cuenta con tres módulos, ingresos, egresos y objetivos.

# Especificación de la(s) herramientas usadas para medición

La herramienta utilizada para la evaluación es SonarQube versión 8.3.1, la cual fue instalada en mi equipo por medio de Docker y a su vez sonar-scanner junto con el plugin 3D Code Metrics y SoftVis 3D Viewer para visualizar el software como una ciudad. SonarQube se eligió para esta práctica debido a que es una herramienta de revisión automática de código para detectar errores, vulnerabilidades y código maloliente. Esta herramienta se puede integrar a un flujo de trabajo existente para permitir la inspección continua de código en todas las ramas de un proyecto y solicitudes de extracción. [1]

La herramienta SonarQube, está compuesta por cuatro componentes:

1. Un servidor SonarQube que inicia 3 procesos principales:

* Servidor web para desarrolladores y gerentes, para explorar imágenes de calidad y configurar la instancia de SonarQube.
* Servidor de búsqueda, basado en búsqueda elástica para realizar búsquedas posteriores desde la interfaz de usuario.
* Servidor de motor de cómputo, a cargo de procesar informes de análisis de código y guardarlos en la base de datos SonarQube.

1. Una base de datos SonarQube para almacenar:

* La configuración de la instancia de SonarQube (seguridad, configuración de complementos, etc.)
* Las imágenes de calidad de proyectos, vistas, etc.

1. Múltiples complementos SonarQube instalados en el servidor, que posiblemente incluyan complementos de idioma, SCM, integración, autenticación y gobierno.
2. Uno o más Sonar Scanners que se ejecutan en sus Servidores de Integración Compuesta / Continua para analizar proyectos. [2]

Además, SonarQube es la herramienta líder para inspeccionar continuamente la calidad y seguridad del código, debido que cubre 27 lenguajes de programación, al mismo tiempo que se empareja con nuestro canal de software existente, SonarQube proporciona una guía de remediación clara para que los desarrolladores entiendan y solucionen problemas y para los equipos en general para ofrecer un software mejor y más seguro.

SonarQube admite una práctica de inspección continua. Por lo tanto, viene con todo lo necesario para respaldar la práctica, una puerta de calidad, gestión de fugas, análisis de sucursales, procesamiento de informes paralelos, funciones de gobierno, alta disponibilidad, un ciclo de retroalimentación corto y más. [3]

SonarQube proporciona una puerta GO/NO-GO para la promoción de aplicaciones. La puerta de calidad es una característica importante ya que proporciona la capacidad de saber en cada análisis si una aplicación aprueba o no los criterios de lanzamiento. En otras palabras, en cada análisis nos indica si una aplicación está lista para la producción "en términos de calidad".



**Imagen 1**. Puerta de calidad. [3]

Asimismo, la experiencia de SonarQube se puede aumentar con complementos, esta herramienta tiene disponibles más de 60 complementos comerciales y comunitarios, lo que facilita mejorar la experiencia con idiomas, métricas y páginas adicionales. Los complementos también se pueden desarrollar para satisfacer necesidades específicas dentro de una organización. [3]



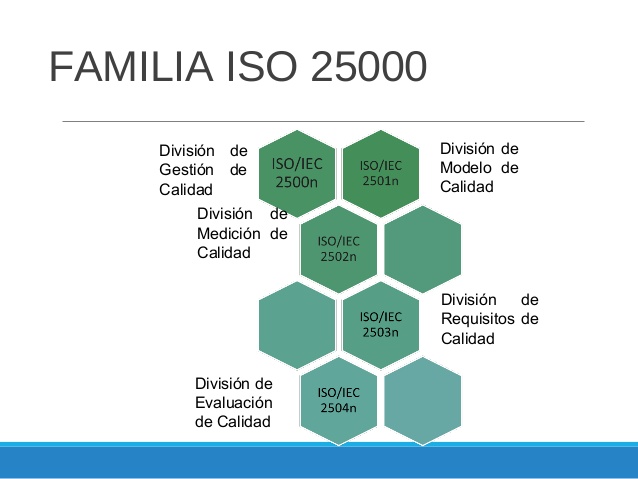
**Imagen 2**. Biblioteca de complementos. [3]

# Especificación de la Norma de calidad

La norma de calidad a usar es la ISO 25000. Esta norma es una pauta a seguir cuando se está desarrollando un software, esto por medio de requerimientos específicos y valoración de aspectos sobre calidad. Esta norma se tomó como base para la calidad debido a que es el fruto del desarrollo de normas anteriores, las cuales abarcan los modelos y procesos de calidad de un software. Estas normas son la ISO/IEC 9126 y la ISO/IEC 14598.

Dentro de sus características, como veremos en el desarrollo de este documento más adelante, se encuentran los atributos de calidad que se dividen en tres ramas: atributos internos, externos y en uso. Esto además de verificar la fiabilidad, la usabilidad, la mantenibilidad, la seguridad y la confiabilidad que son tan importantes para el éxito y calidad del software.

Esta norma ISO, como mencione anteriormente, deriva de otras dos normas, pero no sólo eso, además es una familia, ISO/IEC 25000 se compone de cinco divisiones, las cuales se pueden observar en la siguiente imagen:



**Imagen 3.** Familia ISO 25000. [4]

Como beneficios, decimos que facilita delimitar la definición del software a evaluar, así como mejora la calidad del software, ya que la evaluación se realiza en todos los aspectos de dicho producto.

# Requerimientos específicos de calidad funcionales y técnicos.

La empresa requiere que se evalué los siguientes requerimientos funcionales y técnicos específicos

|  |  |  |
| --- | --- | --- |
| Ítem | Requerimiento de calidad | Prioridad |
| 1 | Que el software pueda funcionar en sistemas operativos Android, MacOS, Windows XP, Windows 7 y Windows 10 (en 32 y 64 bits) | Alta |
| 2 | Que permita trabajar en forma rápida e intuitiva (cuente con ayudas visuales y auditivas interactivas en el software). | Media |
| 3 | Que tenga soporte multi idiomas, especialmente inglés y español | Media |
| 4 | Que permita adecuar su estilo de visualización para adecuarse a personas con limitaciones visuales (Ley 1680 de 20 de noviembre de 2013) | Alta |
| 5 | Implementación de Ley 1581 del 2012 – Protección de datos (HABEAS DATA) | Alta |
| 6 | Permita generar reportes en EXCEL Y PDF. | Baja |
| 7 | Funcionalidad/modulo para reportar errores técnicos o funcionales desde el software. | Media |
| 8 | Permitir acceso a 100 usuarios simultáneos | Alta |
| 9 | Tolerancia a fallos (caída de red, apagones eléctricos frecuentes). | Media |
| 10 | Integración con office | Baja |
| 11 | Cumplimiento del 80% con el estándar OWASP, priorizando en las vulnerabilidades de robo de información, XSS, SQL injection y ransomware. | Alta |
| 12 | Capacidad de respaldo y recuperación de información desde el software. | Media |
| 13 | El software debe demandar mínimos recursos de hardware (cpu Intel celeron, 2 gigas de Ram) | Media |

1. El software es multiplataforma por lo que funciona en sistemas operativos Android, MacOS, Windows XP, Windows 7 y Windows 10 en 32 y 64 bits. Por lo tanto, el porcentaje de cumplimiento es del 100%.
2. El software es entendible, debido a que tiene una interfaz sencilla pero funcional, cuenta con colores llamativos para los botones y las acciones lo cual facilita el trabajo rápido e intuitivo. El porcentaje de cumplimiento es del 100%.
3. El software cuenta con una configuración regional predeterminada para el idioma español, aunque también tiene una configuración regional alternativa que se utilizará cuando la actual no esté disponible, en este caso se tiene el idioma inglés. El porcentaje de cumplimiento es del 100%.
4. El software no se adecua a diferentes estilos de visualización ni tampoco a limitaciones visuales, por lo cual no cumple con la Ley 1680 del 20 de noviembre de 2013. Siendo así, el porcentaje de cumplimiento es del 0%. Como recomendación para la implementación, debe agregarse una configuración que permita que los usuarios adecuen los colores de visualización según sea pertinente para cada una de sus limitantes visuales; además, añadir un módulo lector (sintetizador de voz), el cual proporcionaría una versión hablada del software y permita que los usuarios puedan interactuar con el mismo por medio de escucha. El tiempo estimado para su puesta en marcha es de una semana y en costos de $10.000.000.
5. El software en cuanto a la Ley 1581 del 2012, por la cual se dictan disposiciones generales para la protección de datos personales, sí pide la autorización escrita del usuario para tratar sus datos personales que son guardados en la base de datos, además, el usuario o titular tiene conocimiento de los datos que son guardados, así como también puede actualizarlos en el momento que desee. Sin embargo, no informan al usuario sobre cuál es el tratamiento que utilizarán para tratar sus datos, ni tampoco la identificación del responsable del tratamiento. Por lo tanto, el porcentaje de cumplimiento es del 70%. Como recomendación para la mejora, se debe informar completamente al usuario, el cómo se van a tratar sus datos y si la empresa se hace responsable de los mismos, esto antes de que se genere la autorización. El tiempo estimado de implementación es de 5 horas y un costo de $500.000.
6. El software permite mostrar reportes que son generados por el mismo, pero no permite generarlos como documentos Excel ni PDF. Siendo así, el porcentaje de cumplimiento es del 20%. El estimado en tiempo para implementarlo es de 3 horas y en costo es de $300.000.
7. El software cuenta con opciones que configuran el comportamiento del registro de trabajos de cola fallidos para que pueda controlar qué base de datos y tablas se utilizan para almacenar los trabajos que han fallado. El problema es que esto no se encuentra como un módulo, por lo cual no reporta cuales fueros los errores técnicos ni funcionales. Así que, el porcentaje de cumplimiento es del 30%. Para agregar el módulo se necesita de 2 días y tiene un costo de $1.600.000.
8. El software permite alrededor de 120 usuarios simultáneos, por lo que el porcentaje de cumplimiento es del 100%.
9. El software tiene una alta tolerancia a fallos, puesto que tiene un controlador en la nube predeterminado vinculado a la implementación del disco en la nube. También, tiene opciones que configuran el comportamiento del registro de trabajos de cola fallidos para que pueda controlar qué base de datos y tabla se utilizan para almacenar los trabajos que han fallado. Además, utiliza un controlador de sesión nativo, donde se almacenan los archivos de sesión, por ello, no importa si sucede una caída de red o apagones eléctricos frecuentes. El porcentaje de cumplimiento es del 100%.
10. El software, como lo mencioné en el punto 6, permite mostrar reportes, pero no tiene ninguna integración con office. El porcentaje de cumplimiento es del 0%. El tiempo estimado de integración es de 1 día y un valor de $800.000.
11. Todos los componentes del software tienen su razón de ser, asimismo, el software verifica la identidad del usuario así que sólo los que están autorizados pueden iniciar sesión y sus datos son enviados en forma segura debido a que todos los datos de sesión se cifran antes de almacenarse, Laravel ejecuta automáticamente todo el cifrado, además transforma las consultas SQL a un sistema MVC lo que no permite procesar consultas SQL directamente y así proteger el software de la inyección SQL. También tiene una especificación de la cantidad de minutos para que la sesión permanezca activa antes de que caduque el cual es 120 y tiene la configuración si se desea que caduque inmediatamente al cerrar el navegador. El software no tiene puertas traseras. El porcentaje de cumplimiento es del 70%. Como recomendación para completar el 100%, deben validarse todas las entradas al software, así como agregar un generador de números aleatorios y pedir así aleatoriedad, asegurar la confidencialidad de los datos de los usuarios asegurando que no van a ser modificados. El tiempo estimado para la implementación es de 3 semanas y tendría un costo de $12.000.000.
12. La capacidad de respaldo y recuperación de información del software es buena, ya que todos los controladores de autenticación tienen un proveedor de usuario. Esto define cómo los usuarios se recuperan realmente de su base de datos u otros mecanismos de almacenamiento utilizados por esta aplicación para conservar los datos de su usuario. Por lo tanto, el porcentaje de cumplimiento es del 100%.
13. El software no necesita una gran cantidad de recursos de hardware para funcionar. El porcentaje de cumplimiento es del 100%.

# Atributos de calidad.

Los atributos de calidad que se utilizaran para la evaluación del Software Sistema Contable, se encuentran especificados en el siguiente cuadrado:

*Tabla 1 Atributos de calidad*

|  |  |
| --- | --- |
| ATRIBUTOS INTERNOS | Características del software que determinan su habilidad para satisfacer las necesidades propias e implícitas. |
| ATRIBUTOS EXTERNOS | Características del software que determinan su habilidad para satisfacer las necesidades explicitas e implícitas. |
| ATRIBUTOS EN USO | Características del software que determinan los requerimientos de los usuarios finales de manera que satisfagan sus necesidades. |

## Asignación de puntajes para atributos de calidad.

Los puntajes establecidos a los atributos de calidad seleccionados de acuerdo a las necesidades, se muestran en la siguiente tabla:

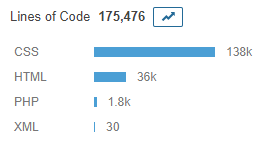
*Tabla 2, Asignación de pesos sobre la medición de atributos.*

|  |  |
| --- | --- |
| Tipo de atributo | Puntaje |
| Atributos internos | 65 |
| Atributos externos | 35 |
| Total | 100 |

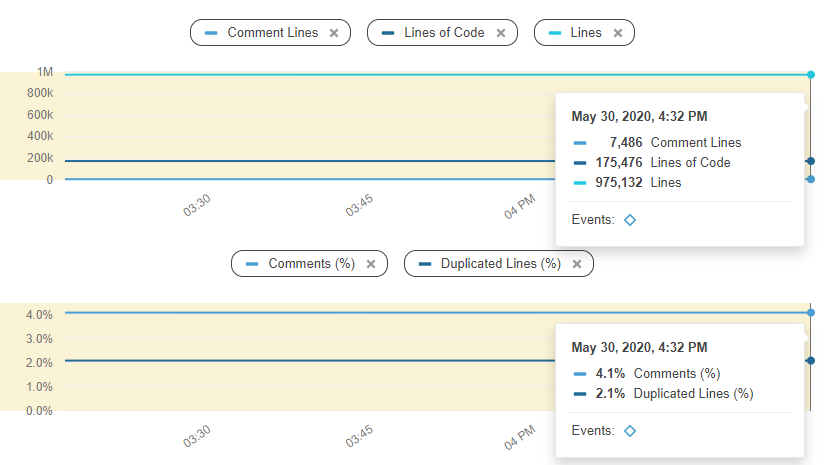
## Atributos internos.

### Tamaño de sistemas y código fuente (20%)

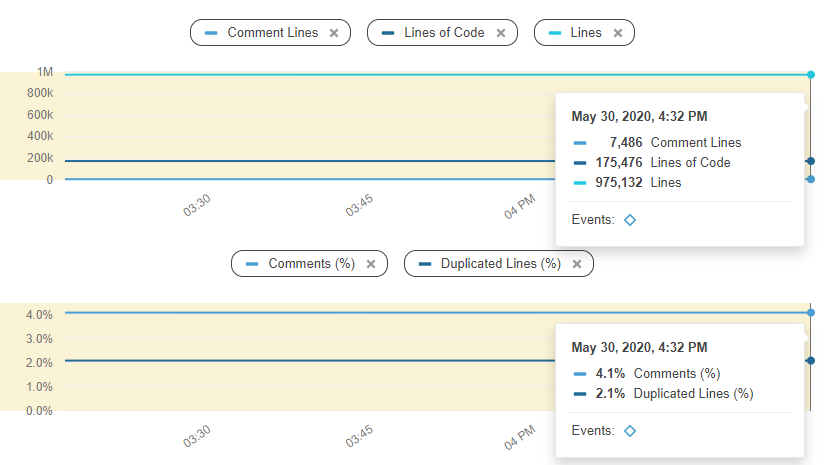
Este software cuenta con 975,132 líneas, de las cuales son: 175,476 líneas de código (LoC), 7,486 líneas de código con comentarios (CLoC); 754 líneas de código de declaraciones; 95 líneas de código de funciones; 41 líneas de código de clases; 20,658 líneas de código duplicadas; 922 bloques de código duplicados y de los 1,713 archivos con los que cuenta, 162 son duplicados.



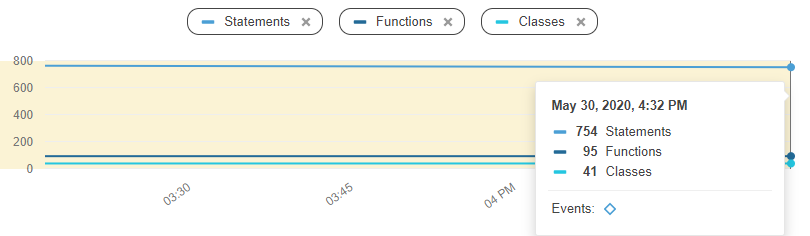
**Imagen 4.** Líneas de código y lenguajes.



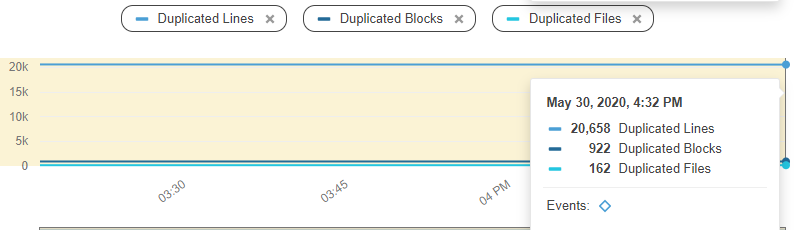
**Gráfico 1.** Líneas, líneas de código y líneas de comentarios.



**Gráfico 2.** Densidad de comentarios y densidad de código duplicado.



**Gráfico 3.** Declaraciones, funciones y clases.

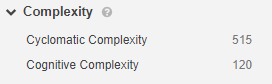


**Gráfico 4.** Líneas, bloques y archivos duplicados.

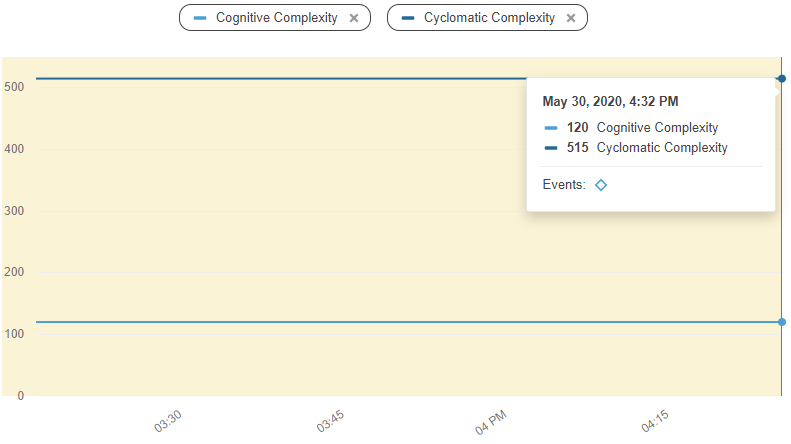
### Complejidad del software (20%)

Dentro de esta sección se evalúan los elementos correspondientes a la complejidad ciclomática y cognitiva de la siguiente manera:

* Complejidad ciclomática: Es calculada por SonarQube 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. Este cálculo varía ligeramente según el idioma porque las palabras clave y las funcionalidades lo hacen. [5]
* Complejidad cognitiva: es una nueva métrica formulada para medir con mayor precisión la comprensibilidad relativa de los métodos. Al hacerlo, aborda las deficiencias de la complejidad ciclomática en esta área. [6]



**Imagen 5.** Complejidad del software.

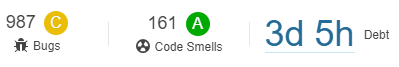


**Gráfico 5.** Complejidad ciclomática y cognitiva.

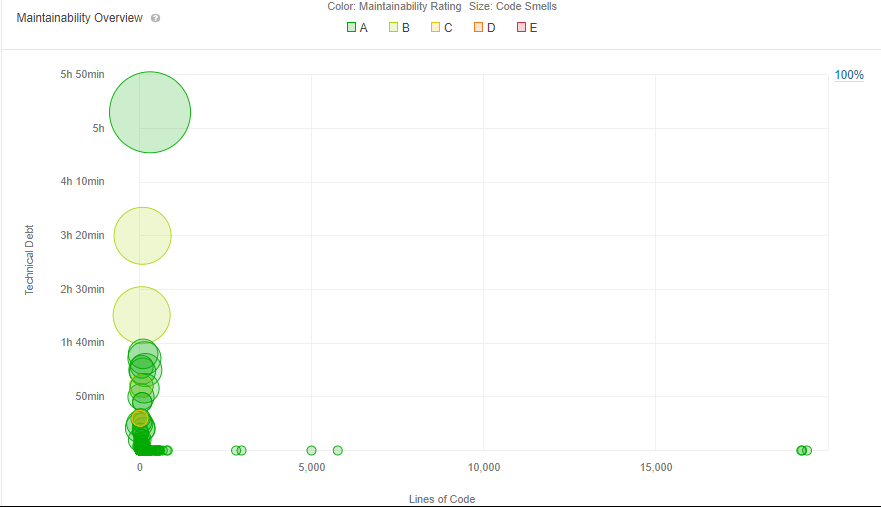
### Deuda técnica (20%)

El cálculo de la deuda técnica está basado en la metodología SQALE (Software Quality Assessment based on Lifecycle Expectations) la cual se mide en días. Esa medida en días se realiza sumando la deuda técnica asociada a cada evidencia. [7]

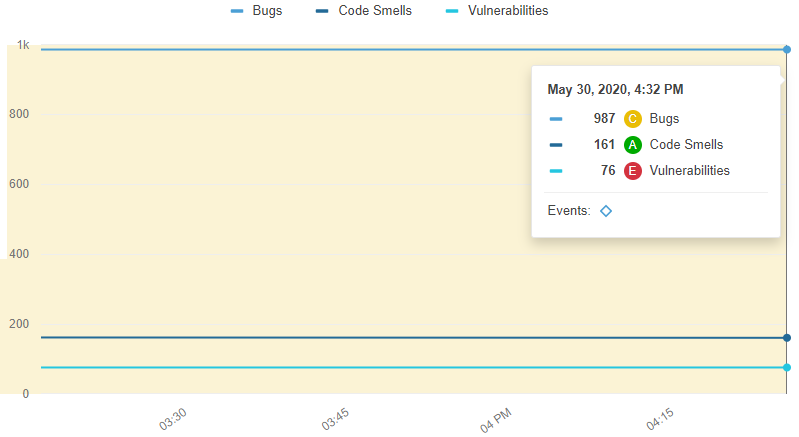
Este software tiene 161 líneas de código mal oliente (Code Smells, un problema de mantenimiento que hace que el código sea confuso y difícil de mantener), 987 bugs (un error de codificación que romperá el código y debe corregirse de inmediato) y una deuda técnica de 3 días y 5 horas. La categoría de mantenibilidad es A, debido a que el radio de deuda técnica es 0%, siendo este menor al 10% y por tanto considerado el proyecto como sano. Por otro lado, la categoría de fiabilidad es C, lo que nos indica que la calificación se encuentra entre 2.5 y 3 que es un punto medio.



**Imagen 6.** Bugs, código mal oliente y deuda técnica.



**Gráfico 6**. Mantenibilidad y deuda técnica vs líneas de código.



**Gráfico 7.** Categorías de fiabilidad y mantenibilidad.

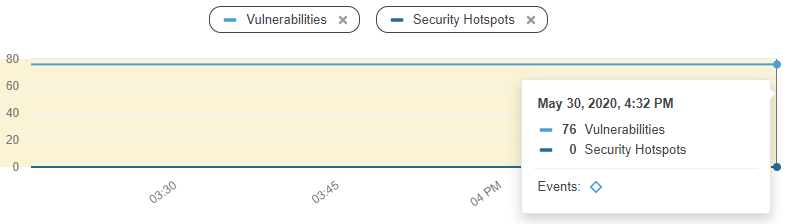
### Seguridad (20%)

En cuanto a la seguridad, se utilizó como categoría SonarSource debido a que la referencia OWASP no ofrecía ningún resultado. El total de vulnerabilidades hallados fue de 76 y muestra una categoría E, lo que nos indica que son bastantes vulnerabilidades y se deben ser solucionadas lo antes posible. Además, para esta sección tenemos en cuenta el código mal oliente y los bugs que se mencionaron en la sección de deuda técnica.

Para el software contable que se está evaluando no se encontraron puntos de acceso, solamente las antes mencionadas vulnerabilidades. Recordemos que la principal diferencia entre estos, es que el primero es un fragmento de código sensible a la seguridad, pero la seguridad general de la aplicación puede no verse afectada, depende del desarrollador revisar el código para determinar si se necesita o no una solución para asegurar el código. En cambio, las vulnerabilidades aparecen cuando se ha descubierto un problema que afecta la seguridad de la aplicación y debe corregirse de inmediato. [8]



**Imagen 7.** Categoría de seguridad.

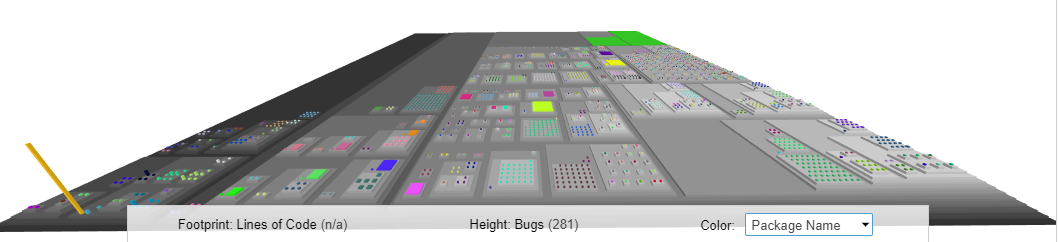


**Gráfico 8.** Vulnerabilidades y puntos de acceso.

### 3d code metrics (20%)

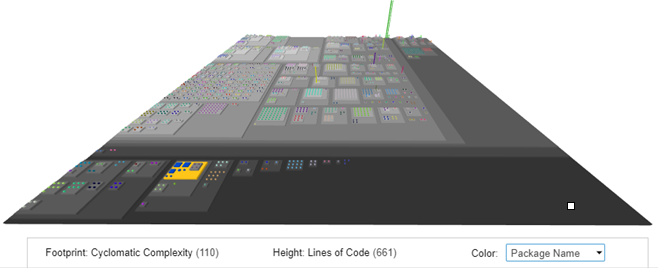
En esta sección, se instalaron dos plugins, el primero 3D Code Metrics, y el segundo SoftVis 3D Viewer, con los cuales se generó la imagen 3D del proyecto.

En el siguiente gráfico podemos observar que hay una clase que sobresale ante las demás, pero antes de eso, se observa que el ancho de los edificios son las líneas de código y su alto los bugs encontrados dentro de estas, siendo así, decimos que en el edificio más alto (resaltado con color amarillo) tiene 281 bugs, lo cual es bastante si recordamos el total de estos en el software es de 987. Este archivo es index.blade.php que se encuentra en el paquete views.



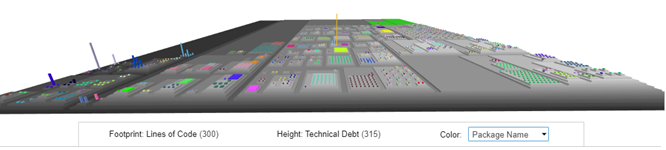
**Gráfico 9.** Líneas de código vs bugs.

Para verificar la zona con mayor complejidad ciclomática, vemos el gráfico a continuación, el cual nos muestra como ancho de los edificios dicha complejidad y sus alturas son las líneas de código del software. Lo que podemos notar es el paquete seleccionado en color amarillo, ya que este contiene los edificios más anchos de la ciudad. Este paquete en el código del software se llama Controllers y tiene una complejidad ciclomática de 110.



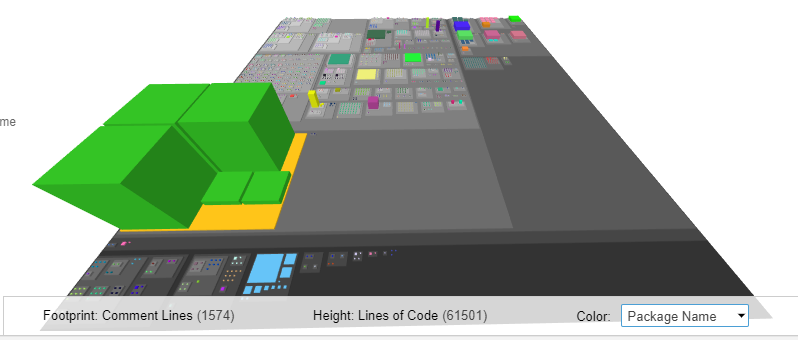
**Gráfico 10.** Complejidad ciclomática vs líneas de código.

En cuanto a la deuda técnica del software, vemos en el siguiente gráfico como ancho las líneas de código y como alto la deuda técnica y nos damos cuenta de un edificio más alto que los demás, el cual se encuentra señalado en color amarillo y cuenta con un número de 315 como deuda técnica. Esta deuda se encuentra en el archivo loadMore4.htm en la carpeta Ajax.



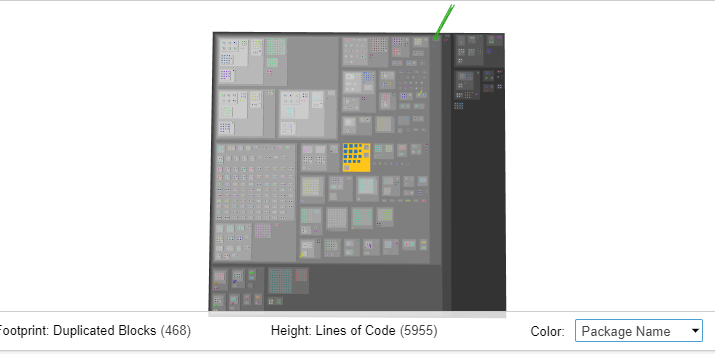
**Gráfico 11.** Líneas de código vs deuda técnica.

Revisando las líneas de comentarios vemos el gráfico siguiente en el cual tenemos como ancho de los edificios las líneas de comentarios y su altura las líneas de código. Cabe resaltar el paquete que se encuentra en amarillo, el cual contiene cinco edificios bastante anchos con respecto a los demás, la cantidad de comentarios son 1574 de 61501 líneas de código. El paquete es css dentro de global.



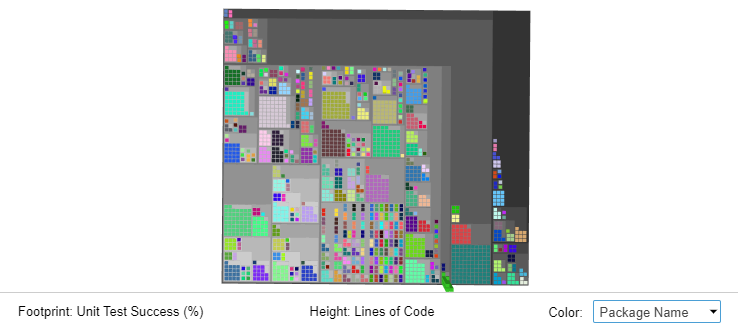
**Gráfico 12.** Líneas comentadas vs líneas de código.

En el gráfico siguiente vemos los bloques duplicados como ancho de los edificios y las líneas de código como sus altos. El paquete que se encuentra resaltado en color amarillo, es el que tiene mayor cantidad de bloques ya que estos son 468. El paquete es mapplic.



**Gráfico 13.** Bloques duplicados vs líneas de código.

Este gráfico muestra la ciudad con el ancho de los edificios como los test unitarios y el alto como las líneas de código, el color de los edificios se debe a los nombres de los paquetes.



**Gráfico 14.** Test unitarios vs líneas de código.

## Atributos Externos.

### Usabilidad (20%)

* En cuanto a la efectividad, cuenta con botones fáciles de identificar, permitiendo que el usuario pueda acceder de manera rápida y fácil al lugar que necesite; además es un software útil ya que maneja cada una de las divisiones necesarias para tener al día las cuentas del usuario, pero en la parte del manejo de errores, no apoya a los usuarios con las características necesarias.
* Respecto a la eficiencia, el software cumple con su objetivo de llevar las finanzas del usuario, es coherente en los nombres de los objetos y su funcionalidad final, además el tiempo que tarda en realizar las consultas es aceptable.
* En la parte de satisfacción, es cómodo de utilizar ya que es simple pero funcional, así que es fácil de aceptar y de utilizar por el tiempo que el usuario requiera, es recomendable también ya que es necesario para cada persona llevar sus finanzas ordenadamente.

### Fiabilidad (30%)

Para la evaluación de fiabilidad del software contable como se mostró anteriormente en la sección de deuda técnica, se tiene que el software está en la categoría C, lo que quiere decir que cuenta con bastantes bugs que pueden afectar su funcionamiento, por lo cual es necesario trabajar en arreglarlos lo antes posible para aumentar su fiabilidad.

### Mantenibilidad (50%)

El software contable tiene una gran facilidad para ser modificado y corregido, por lo que, aunque tenga código mal oliente es sencillo modificarlo y enmendarlo. El software también es adaptable, ya que es tan fácil de modificar su código, es factible actualizarlo a las normas y leyes vigentes.

# Recomendaciones y conclusiones

## Seguridad

La seguridad del software no es la mejor, pero tampoco está mal, debido a que no cuenta con puntos de acceso y cifra los datos de sus usuarios, aunque si es importante realizar las respectivas correcciones de las posibles vulnerabilidades que se encontraron al realizar la evaluación además de modificar el software para que cumpla al 100% con el estándar OWASP como se mencionó en la sección cinco sobre requerimientos específicos de calidad funcionales y técnicos. El tiempo estimado de esfuerzo para modificar el software y mejorar la seguridad es de 3 semanas con 1 hora y 16 minutos, estos últimos obtenidos en la revisión dada por la herramienta SonarQube. El valor aproximado de estos arreglos es de $12.500.000.

## Confiabilidad

La confiabilidad del software es media, debido a que tiene varias mejoras por hacerse, como en el caso de los bugs, los cuales son 987 y por ende tendría un costo en tiempo de 4 días y 2 horas para poderlos arreglar. Aunque, aun así, el software sigue funcionando y cumple con sus objetivos, sin embargo, es bueno tener en cuenta las recomendaciones realizadas por la herramienta SonarQube. Un punto positivo para confiar en el software es su alta tolerancia a fallos como caída de red y apagones eléctricos frecuentes, para lo cual cuenta con un respaldo de la información. El costo para corregir los bugs encontrados en el estudio es de $600.000.

## Usabilidad

La usabilidad del software es buena pero no lo suficiente, cuenta con una interfaz fácil de entender y manejar, para que los usuarios pueden utilizarla sin problema alguno, a excepción de los que tienen alguna discapacidad visual, para los cuales no se ha adecuado aún ninguna forma de interacción, incumpliendo así con la Ley 1680 de 20 de noviembre de 2013. Aun así, el software cumple con sus respectivas funciones demostrando una sencillez visual buena y agradable, así que la mayoría de usuarios, aún sin haberla utilizado con anterioridad, son capaces de aprenderla a manejar en un tiempo estimado de 30 minutos. Sin embargo, no todos los usuarios la van a poder manejar, y puede que a algunos les genere problemas el entender el software, y no se encuentra ningún apartado en el cual se pueda pedir ayuda o muestre las indicaciones de cómo utilizarlo y tampoco cuenta con un sintetizador de voz. Estas modificaciones podrían tener un tiempo de implementación de 8 días con un costo de $11.000.000.

**Conclusiones.**

El software obtuvo los siguientes porcentajes según el estudio realizado por la herramienta SonarQube respecto a su calidad:

*Atributos internos 62%:*

* Tamaño de sistemas y código fuente: el valor se obtuvo debido a que la densidad de comentarios está muy por debajo del mínimo establecido el cual es 20% y el 80% deberían ser líneas de código. Siendo así, 40 \* 0.2 = 8%.
* Complejidad del software: la complejidad media por clase mayor encentrada fue de 29, lo cual indica que está en el rango entre 21 y 50, para estar así en la categoría de un software complejo y de alto riesgo. 50 \* 0.2 = 10%.
* Deuda técnica: puedo decir que la deuda técnica es moderada debido a que no sobrepasa la semana, siendo así, 70 \* 0.2 = 14%.
* Seguridad: en este ámbito, encontramos múltiples vulnerabilidades las cuales son necesarias de corregir lo antes posible, 50 \* 0.2 = 10%.
* 3D code metrics = el plugin nos mostró el código de software para entenderlo con mayor facilidad, por tanto, 100 \* 0.2 = 20%.

*Atributos externos 75%:*

* Usabilidad: 60 \* 0.2 = 12%.
* Fiabilidad: 60 \* 0.3 = 18%.
* Mantenibilidad: 90 \* 0.5 = 45%

El porcentaje total de calidad del software contable es entonces:

Como vemos, el software cuenta con un porcentaje de calidad de 66.55% de 100%, lo que nos indica que no está tan mal para empezar, pero si debe ser modificado y corregido antes de ser lanzado, debido a que este porcentaje no es el mejor para ser aprobado el software.

# Bibliografía

[1] Documentación en el panel de control de SonarQube, http://localhost:9000/documentation

[2] Documentación SonarQube 8.3, https://docs.sonarqube.org/latest/architecture/architecture-integration/

[3] Documentación producto SonarQube, https://docs.sonarqube.org/latest/architecture/architecture-integration/

[4] Imagen divisiones de la norma ISO 25000, https://image.slidesharecdn.com/iso25000-161026021332/95/iso-25000-5-638.jpg?cb=1477448091

[5] Métricas de complejidad ciclomática, https://docs.sonarqube.org/latest/user-guide/metric-definitions/

[6] Métricas de complejidad cognitiva, https://www.sonarsource.com/resources/white-papers/cognitive-complexity.html

[7] Deuda técnica, https://www.sonarqubehispano.org/pages/viewpage.action?pageId=4980855

[8] Vulnerabilidad y punto de acceso, <https://docs.sonarqube.org/latest/user-guide/security-hotspots/>

# Firma del perito.

|  |  |
| --- | --- |
| **Responsable de la evaluación** | **Firma** |
| Nombre: Yeslie Guisel Mestizo Montaño  Empresa: Nortic  Cargo: Supervisor de calidad.  GitHub: https://github.com/YeslieMestizo/Quality\_software\_SonarQube1.git |  |