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

**Restaurante**

**Versión 1.0**

**Perito:**

**Nombre: Iván Felipe Pava Sánchez**

**Cedula: 1.007.475.699**

**Email: ivan.pava@usantoto.edu.co**



**Contenido del informe**

[1. Propósito de la evaluación 2](#_heading=h.gjdgxs)

[2. Producto a evaluar 2](#_heading=h.30j0zll)

[3. Especificación de la(s) herramientas usadas para medición 2](#_heading=h.1fob9te)

[4. Especificación de la Norma de calidad 2](#_heading=h.3znysh7)

[5. Requerimientos específicos de calidad funcionales y técnicos. 2](#_heading=h.2et92p0)

[6. Atributos de calidad. 3](#_heading=h.3dy6vkm)

[6.1. Asignación de puntajes para atributos de calidad. 3](#_heading=h.1t3h5sf)

[6.2. Atributos internos. 3](#_heading=h.4d34og8)

[6.2.1. Tamaño de sistemas y código fuente 3](#_heading=h.2s8eyo1)

[6.2.2. Complejidad del software 3](#_heading=h.17dp8vu)

[6.2.3. Deuda técnica 3](#_heading=h.3rdcrjn)

[6.2.4. Seguridad 3](#_heading=h.26in1rg)

[6.3. Atributos Externos. 3](#_heading=h.lnxbz9)

[6.3.1. Usabilidad 3](#_heading=h.35nkun2)

[6.3.2. Fiabilidad 4](#_heading=h.1ksv4uv)

[6.3.3. Mantenibilidad 4](#_heading=h.44sinio)

[7. Recomendaciones y conclusiones 4](#_heading=h.2jxsxqh)

[7.1. Seguridad 4](#_heading=h.z337ya)

[7.2. Confiabilidad 4](#_heading=h.3j2qqm3)

[7.3. Usabilidad 4](#_heading=h.1y810tw)

[8. Bibliografía 4](#_heading=h.4i7ojhp)

[9. Firmas de evaluadores. 4](#_heading=h.2xcytpi)

# Propósito de la evaluación

Cuando se aplica la calidad del software, el producto o servicio es eficiente en cuanto a tiempo y recursos. El sistema satisface las necesidades requeridas por el cliente o usuario final, donde se esperan los resultados, obteniendo así la satisfacción del cliente. Funciona con un rendimiento y tiempos de respuesta óptimos.

La calidad de software hace que el producto o servicio sea seguro en caso de amenazas o errores del dispositivo. En el caso de las mejoras, la norma del programa permite realizar ajustes sin ninguna dificultad gracias a la sencillez de la revisión del marco y la capacidad de modificarlo.

Dado que no exista calidad de software en la producción, nuestros bienes o servicios no se ajustan a los requisitos proporcionados por el cliente. El programa no funciona exactamente como está planeado. El programa se abandona porque es complicado de ejecutar. Su capacidad para identificar fallos y arreglar problemas es baja. Al no implementar la consistencia del programa, el marco parece ser más intuitivo, ya que es interpretado y utilizado por el consumidor. Al omitir la calidad del desarrollo de software, el producto o servicio se vuelve más vulnerable a los riesgos o a los fallos.

# Producto a evaluar

La calidad del código es un tema que surgió cuando se inventó el software. Este software está orientado a la administración de un restaurante donde se encuentran los distintos cargos, la administración de mesas y los meseros. En su versión 1.0, es la base del proyecto, utilizando CakePHP y Composer para su desarrollo y diseño. CakePHP es un sistema de desarrollo de aplicaciones web escrito en PHP, basado en los conceptos de Ruby on Rails.

# Especificación de la(s) herramientas usadas para medición

Se suele decir que la calidad del código es un atributo de calidad interno, ya que el usuario no se hace visible. Pero llega un momento en que este atributo de calidad se vuelve interno a externo, y esto ocurre cuando tener que modificar el código para hacer un cambio lleva mucho más tiempo del que debería. Para comprobar la consistencia interna de un programa, la inspección del código se realiza típicamente usando SonarQube o métodos relacionados.

En la inspección de este proyecto se utilizo dos herramientas esenciales para inspeccionar la calidad y seguridad del código. SonarQube Community Edition en su versión 8.3.1 (build 34397) y Sonar Scanner en su versión 4.3.

SonarQube es la herramienta líder para inspeccionar continuamente la calidad y seguridad del código de su base de datos y guiar a los equipos de desarrollo durante las revisiones del código. Como tantas otras herramientas similares, permite realizar automáticamente análisis de código fuente estático, buscando patrones con errores, malas prácticas o incidentes. Además, realiza un cálculo técnico de la deuda. Las siguientes son algunas de las verificaciones que hacen que dispositivos como el SonarQube: Detección de código duplicado, Falta de pruebas unitarias, falta de comentarios, Código spaghetti, complejidad ciclomática, alto acoplamiento, Tamaño de archivos de código, Tamaño de métodos, No adecuación a estándares y convenciones de código y Vulnerabilidades conocidas de seguridad.

SonarScanner se utiliza para el análisis y los resultados para después ser enviados a SonarQube. Es un escáner genérico, CLI y debe proporcionar configuraciones explícitas que enumeren las ubicaciones de sus archivos de origen.

Se utilizo plugins como SoftVis3D Viewer, con el fin de visualizar de forma gráfica, la vista de código en forma de "ciudad" da una visualización de la estructura jerárquica del proyecto. Las carpetas o productos se muestran como divisiones, construyendo archivos como. La altura y el color del edificio dependen de dos métricas arbitrarias del Sonar. Evostreet es una arquitectura segura que utiliza el modelo urbano para imaginar los sistemas informáticos emergentes.

# Especificación de la Norma de calidad

La ISO 25000 proporciona una guía para el uso de la nueva serie de normas internacionales denominadas Sistemas y Requisitos de Calidad de Software y Evaluación (SQuaRE) El objetivo de la ISO 25000 es proporcionar una visión general del contenido de SQuaRE. modelos de referencia y definiciones comunes, así como la relación entre los documentos. La ventaja general de seguir la norma ISO 25000 es la misma que la de todas las normas de este grado, garantizar que los bienes y servicios son seguros y de alta calidad.

Específicamente, la certificación ISO 25000 indica que la consistencia de las aplicaciones está asegurada, o que los servicios digitales que usted provee son de la más alta confianza y han sido monitoreados. La Organización Internacional de Normalización describe cómo demostrar que se está haciendo algo de la mejor manera posible. Las características de calidad y las mediciones asociadas pueden ser útiles no sólo para evaluar el software del producto, sino también para definir los requisitos de calidad. Esta norma está dirigida tanto a las pequeñas empresas que crean programas informáticos como a las empresas más grandes, independientemente de su volumen de producción.

# 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) | Media |
| 2 | Que permita trabajar en forma rápida e intuitiva (cuente con ayudas visuales y auditivas interactivas en el software). | Media |
| 3 | Que tenga soporte multiidiomas, especialmente inglés y español | Bajo |
| 4 | Que permita adecuar su estilo de visualización para adecuarse a personas con limitaciones visuales (Ley 1680 de 20 de noviembre de 2013) | Bajo |
| 5 | Implementación de Ley 1581 del 2012 – Protección de datos (HABEAS DATA) | Bajo |
| 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 | Baja |
| 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 |

# Atributos de calidad.

Los atributos de calidad que se utilizaran para la evaluación del Software Restaurante, de acuerdo a lo especificado 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.

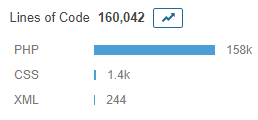
Los puntajes establecidos a los atributos de calidad seleccionados de acuerdo a las necesidades, se muestran en la siguiente tabla:

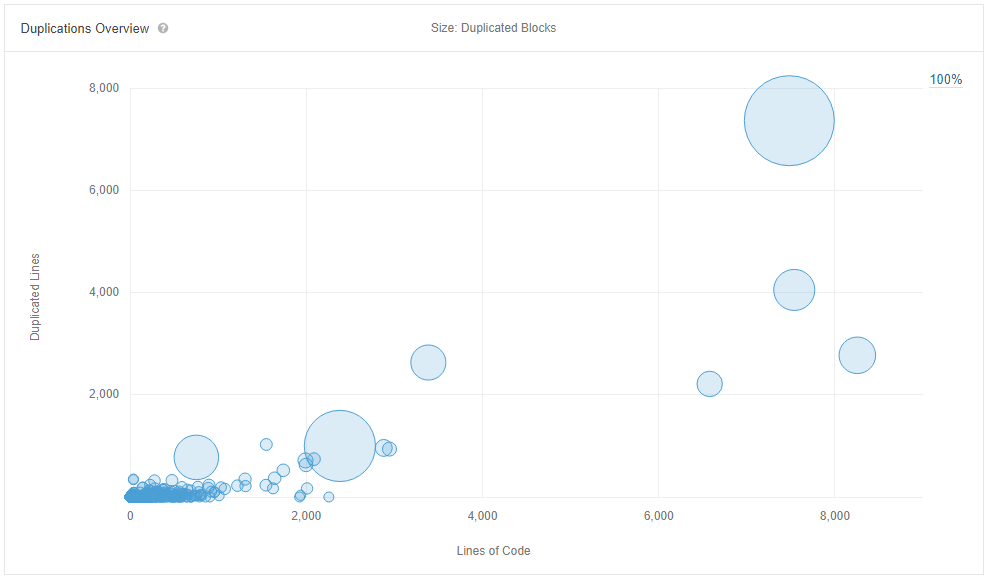
*Tabla 2, Asignación de pesos sobre la medición de atributos.*

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

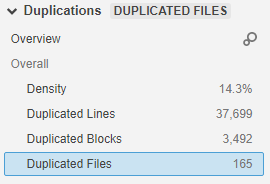
## Atributos internos.

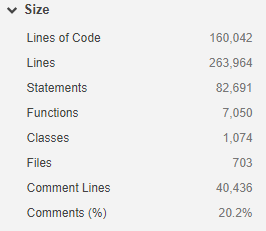
### Tamaño de sistemas y código fuente (20%)



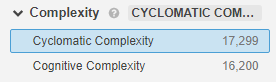


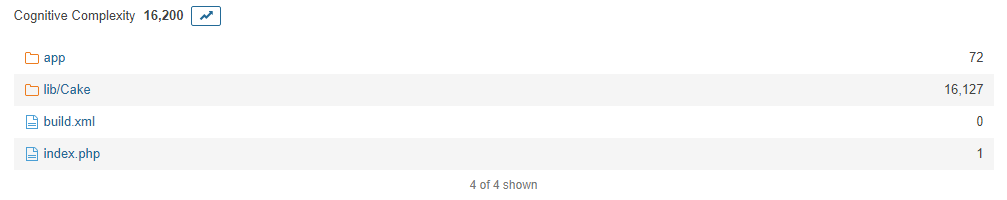
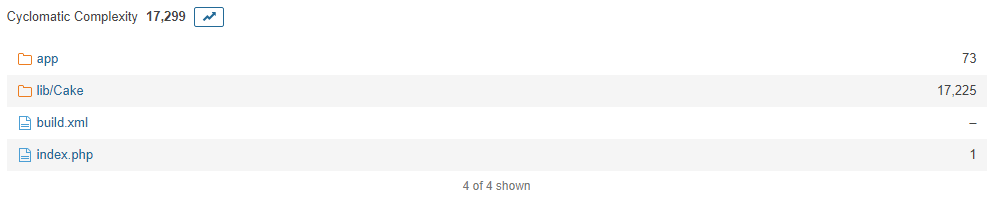
El área de los círculos representa los bloques duplicados.





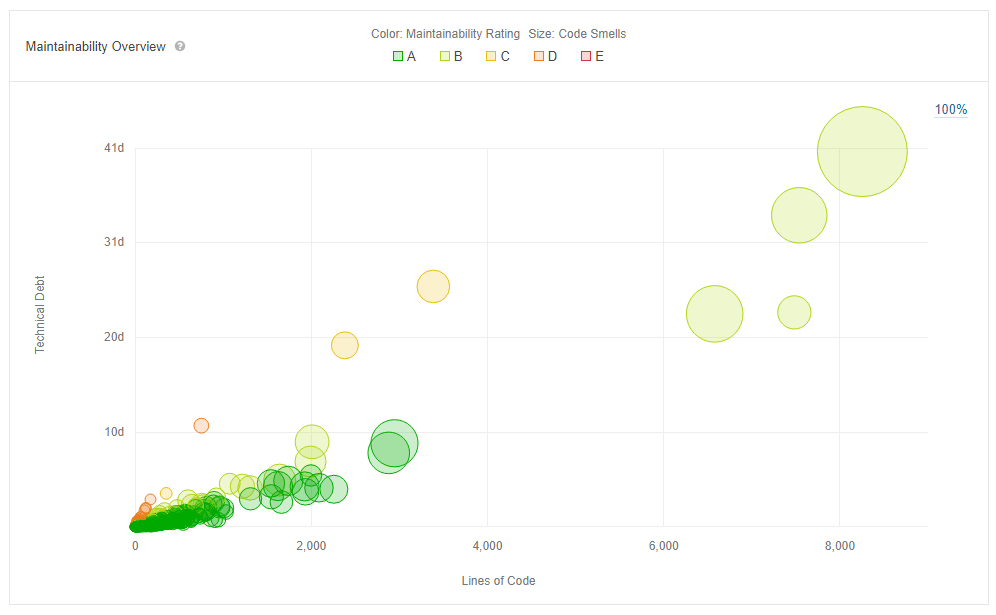
### Complejidad del software(20%)

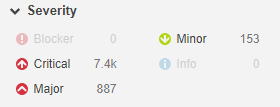




### Deuda técnica (20%)

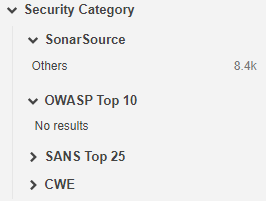
La deuda técnica es el costo y el interés que hay que pagar por hacer las cosas mal. El sobreesfuerzo de pagar para mantener un software de producto mal hecho, y lo que implica, como el costo de una mala imagen para los clientes, etc. Este proyecto tiene 468 días de deuda técnica.





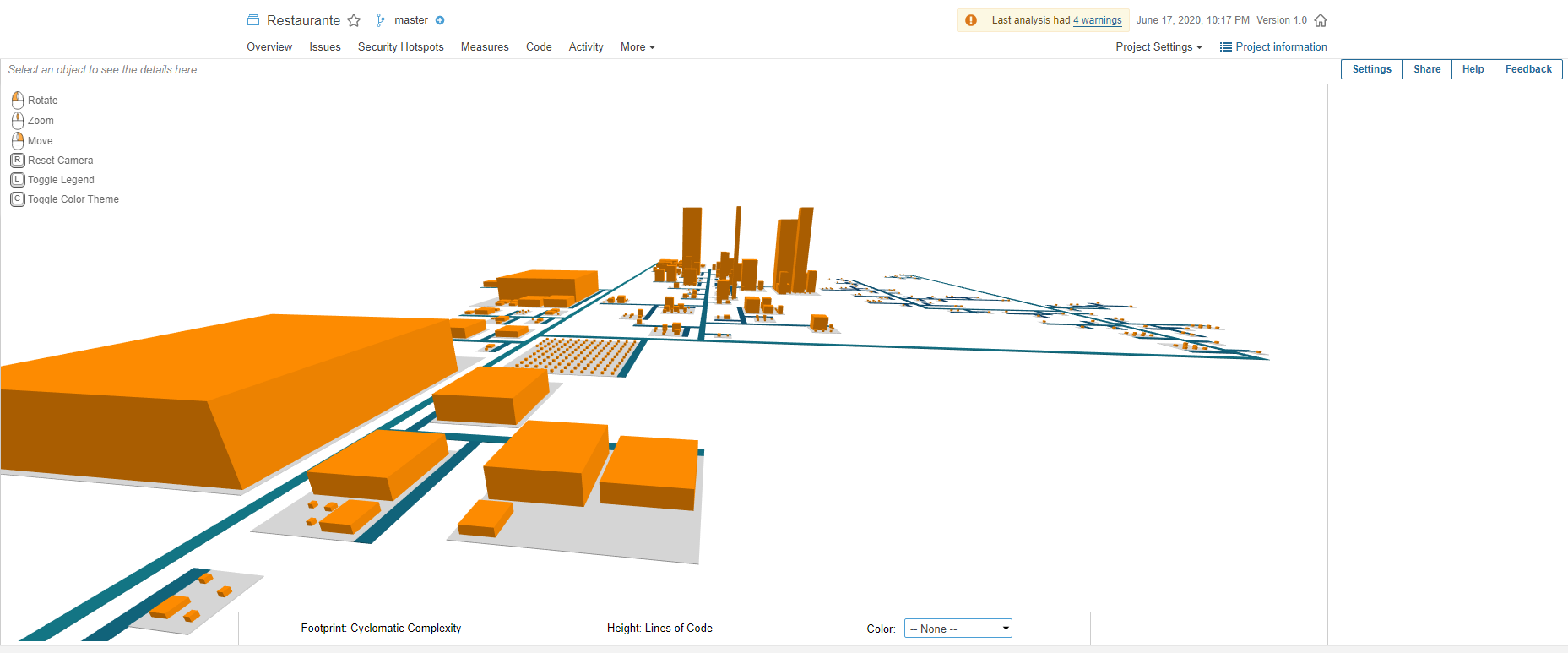
### Seguridad (20%)

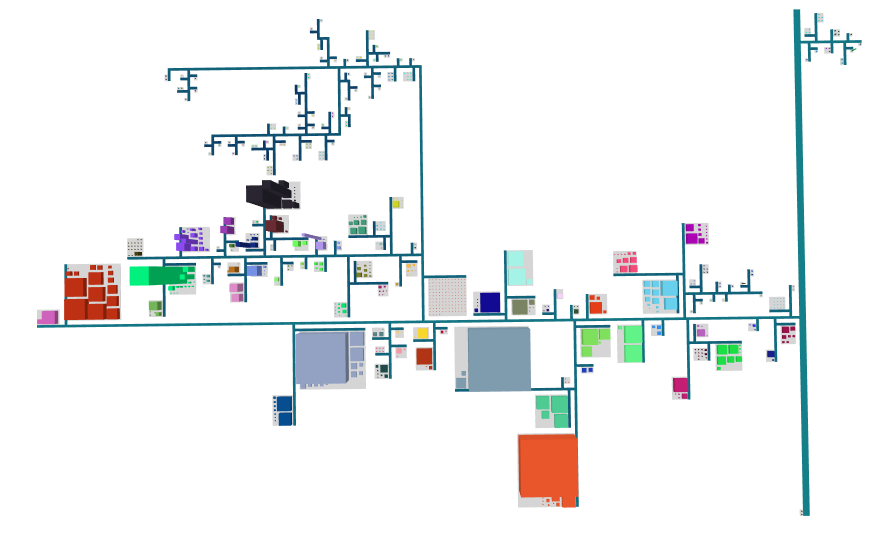
Según la organización OWASP, el proyecto no infringe ninguno de los diez principales riesgos de seguridad de las aplicaciones web.



### SoftVis 3D Viewer (20%)

Este punto de vista se centra en las dependencias dentro de la estructura. Para el marco, el simple modelo 3D está diseñado hacia abajo. Las dependencias han sido agregadas y pueden ser exploradas fácilmente sin sobrecargar la visualización.





## Atributos Externos.

### Usabilidad (20%)

Aunque aún es un sistema en desarrollo hasta el momento cumple con lo que se necesita, sus interfaces son adecuadas y sus elementos son funcionales.

### Fiabilidad (30%)

El software brinda un buen nivel de fiabilidad dado que los servidores donde se encuentra alojado son fiables y no posee errores que puedan comprometer el servicio también cuentan con sus respectivas copias de seguridad.

### Mantenibilidad (50%)

Utiliza el diseño de MVC El gran beneficio de esta estrategia de programación es que permite cambiar cada uno de ellos sin necesidad de alterar los demás, lo que permite crear sistemas flexibles y portátiles que pueden modificarse rápidamente e introducir o eliminar nuevos módulos o funcionalidades de forma agrupada, ya que cada programa informático utiliza el mismo marco para sus vistas.

# Recomendaciones y conclusiones

## Seguridad

Como no se detectaron problemas de seguridad se recomienda mantenerlo así seguir teniendo en cuenta los estándares de seguridad para el desarrollo.

## Confiabilidad

Aún faltan muchos aspectos se recomienda realizar los análisis periódicamente para ver la evolución de los proyectos

## Usabilidad

Tener en cuanta en el desarrollo un módulo de reporte de errores. Para mantener una usabilidad buena y tener contentos a los clientes

# Bibliografía

<https://stackoverflow.com/questions/51210753/difference-between-sonarqube-sonarscanner>

<https://www.federico-toledo.com/analisis-de-codigo-con-sonarqube/>

<https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/>

<https://www.excentia.es/3d-code-metrics>

# Firma del perito.

|  |  |
| --- | --- |
| **Responsable de la evaluación** | **Firma** |
| Nombre: Iván Felipe Pava Sánchez  Empresa: G-Soft  Cargo: Gerente General  GitHub: <https://github.com/iFePax> | Felipe Pava. |