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

**Skeleton PHP**

**Versión 1.0**

**Perito:**

**Nombre: JOHAN SEBASTIAN BERNAL DIAZ**

**Cedula: 1056076148**

**Email: Johan.bernal@usantoto.edu.co**



**Contenido del informe**

[1. Propósito de la evaluación 2](#_Toc43389770)

[2. Producto a evaluar 2](#_Toc43389771)

[3. Especificación de la(s) herramientas usadas para medición 2](#_Toc43389772)

[4. Especificación de la Norma de calidad 2](#_Toc43389773)

[5. Requerimientos específicos de calidad funcionales y técnicos. 3](#_Toc43389774)

[6. Atributos de calidad. 3](#_Toc43389775)

[6.1. Asignación de puntajes para atributos de calidad. 3](#_Toc43389776)

[6.2. Atributos internos. 4](#_Toc43389777)

[6.2.1. Tamaño de sistemas y código fuente (20%) 4](#_Toc43389778)

[6.2.2. Complejidad del software (20%) 5](#_Toc43389779)

[6.2.3. Deuda técnica (20%) 6](#_Toc43389780)

[6.2.4. Seguridad (20%) 8](#_Toc43389781)

[6.2.5. 3d code metrics (20%) 9](#_Toc43389782)

[6.3. Atributos Externos. 10](#_Toc43389783)

[6.3.1. Usabilidad (20%) 10](#_Toc43389784)

[6.3.2. Fiabilidad (30%) 11](#_Toc43389785)

[6.3.3. Mantenibilidad (50%) 11](#_Toc43389786)

[7. Recomendaciones y conclusiones 11](#_Toc43389787)

[8. Bibliografía 11](#_Toc43389788)

[9. Firma del perito. 12](#_Toc43389789)

# Propósito de la evaluación

El propósito de la evaluación del software Skeleton PHP, es poder determinar el cumplimiento de los diferentes atributos de calidad de este, así mismo la evaluación de sus métricas, tales como: usabilidad, confiabilidad y mantenibilidad. De esta forma, es posible establecer que tan eficiente y mantenible es el software. La realización de esta evaluación, ayudará al mejoramiento constante de la calidad del software, no sólo al producto final que finalmente usará el cliente, sino también contribuirá a una mayor facilidad y manejo para mantener el software a largo plazo por parte de los desarrolladores.

# Producto a evaluar

El software a evaluar (Skeleton PHP) es una librería para el desarrollo rápido de aplicaciones web, que implementa el patrón MVC y DAO, y usa las mejores prácticas y herramientas front-end. Skeleton PHP NO debe confundirse con un framework para el desarrollo de aplicaciones complejas.

# Especificación de la(s) herramientas usadas para medición

* Sonar Scanner 4.2.0.1873:
* Se utiliza para analizar código en la tecnología que se va utilizar, en este caso utilizaremos SonarQube 8.0.
* Se puede ejecutar de forma sencilla desde una terminal de comandos.
* Es un script que está disponible tanto para Windows, como para Mac y para Linux.
* También puede analizar el código de tu repositorio, si por ejemplo haces uso de Jenkins o Azure Devops para descargar el código y automatizar la inspección.
* Los escáneres hacen uso de las reglas de SonarQube para indicar donde están los errores
* Docker Desktop 2.3.0.2:
* Docker es un programa de código abierto que permite que una aplicación Linux y sus dependencias se empaqueten como un contenedor
* Docker proporciona una manera estándar de ejecutar su código
* Docker es un sistema operativo para contenedores. De manera similar a cómo una máquina virtual el hardware del servidor, los contenedores virtualizan el sistema operativo de un servidor.

# Especificación de la Norma de calidad

La norma ISO 25000 es la que mejor se acopla con la visión general de los contenidos de SQuaRE y de que especifican los requisitos ,además de la evaluación de calidad del software, la ISO 25000 está dividida en cinco si se puede decir subnormas que son las que nos darán los lineamientos más completos de cómo hacer la evaluación del software, la primera es la ISO 2500n: Gestión de calidad esta es la que reúne todas las normar y es está presente en la ISO 25000 y en la ISO 25001, la segunda es la que mide la calidad de nuestro software, ISO 2501n es un modelo de calidad que se compone entre otros por tener la fiabilidad, la seguridad, la mantenibilidad y la usabilidad, lo siguiente es la ISO 2502n que sirve para la medición de calidad ya sean las interna, externa las dos últimas son las más importantes ya que nos dicen los requisitos de calidad (ISO 2503n: entrada del proceso de evaluación al terminar el software) y su evaluación final del software evaluación de calidad (ISO 2504n: incluye reglas que proporcionan los requisitos, las recomendaciones y todas las guías para llevar a cabo el proceso de evaluación del producto final),es de entender que por su eficacia, su versatilidad y por su seguridad que se escogió esta norma.

# 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 multiidiomas, 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 |

A continuación, se evaluarán cada uno de los anteriores puntos específicamente para el software Skeletor PHP

* 1. El software cumple con compatibilidad en todos los sistemas operativos ya que su plataforma principal es web. se puede acceder a él mediante cualquier navegador de internet.
* 2. El software permite trabajar de manera rápida e intuitiva ya que cuenta con ayudas visuales en caso de usarse incorrectamente o de errores en el sistema, permitiendo al usuario desenvolverse de manera óptima en el uso del software.
* 3. El software no dispone de ningún metodo, archivo, o función que nos permita determinar si cuenta con soporte multi idioma. en este caso se recomienda crear en la carpeta de recursos diferentes ficheros **.*properties*** conlos nombres de las etiquetas de nuestro vocabulario y sus traducciones a mostrar en el software. se recomienda usar el formato “messages\_[lang]\_[country].properties”, donde [*lang]* es el código ISO del idioma y [*country]* es el código ISO del país. ejemplo implementación de idioma español: **messages\_es\_ES.properties**
* 4. El software no dispone de ningún método, archivo, o función que nos permita determinar si cuenta con ayudas a personas con limitaciones visuales (Ley 1680 de 20 de noviembre de 2013)
* 5. El software no hace una correcta implementación de la ley 1581 del 2012 ya que se encontró una vulnerabilidad
* 6. El software no dispone de ningún metodo, archivo, o función que nos permita determinar si cuenta generación de reportes excel y pdf. en este caso se plantea usar bibliotecas como “DynamicReports” u otras que permiten la generación de reportes pdf y excel
* 7. El software no dispone de ningún metodo, archivo, o función que nos permita determinar si cuenta con reporte de errores. se recomienda crear un método que permita el envío de reportes de errores desde la aplicación.
* 8. El software no dispone de ningún metodo, archivo, o función que nos permita determinar si permite 100 usuarios en simultáneo, se recomienda crear un pool de conexiones con los diferentes usuarios a la base de datos, permitiendo de esta manera el uso simultáneo por parte de los usuarios del software
* 9. se recomienda el uso de servicios en la nube para alojar la base de datos y conexiones de esta manera evitando pérdida de información y fallos eléctricos
* 10. El software no dispone de ningún metodo, archivo, o función que nos permita determinar si el software cuenta con integración con office. en este caso el software no requiere integración con office.
* 11. El Software cumple un 70% del estándar OWASP imcumplientdo la minimo de requerimientos de calidad se recomienda hacer las respectivas correcciones a partir del SonarQube.
* 12. Puesto que el software basa su información en la base de datos cuenta con el respaldo directo desde la la interfaz de usuario permitiendo consultar información guardada.
* 13. El software no demanda mucha capacidad ya que es un aplicativo web

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Requerimiento de calidad** | **% de implementacion** | **Costo $** | **Tiempo de implementación** |
| **1** | Que el software pueda funcionar en sistemas operativos Android, MACOs, Windows XP, Windows 7 y Windows 10 (en 32 y 64 bits) | 100% | n/a | n/a |
| **2** | Que permita trabajar en forma rápida e intuitiva (cuente con ayudas visuales y auditivas interactivas en el software). | 50% | n/a | n/a |
| **3** | Que tenga soporte multi idiomas, especialmente inglés y español | 50% | $180.000 | 1.5 días |
| **4** | Que permita adecuar su estilo de visualización para adecuarse a personas con limitaciones visuales (Ley 1680 de 20 de noviembre de 2013) | 0% | $2’400.000 | 20 días |
| **5** | Implementación de Ley 1581 del 2012 – Protección de datos (HABEAS DATA) | 10% | $1’200.00 | 4 días |
| **6** | Permita generar reportes en EXCEL Y PDF. | 0% | n/a | n/a |
| **7** | Funcionalidad/módulo para reportar errores técnicos o funcionales desde el software. | 0% | $600.000 | 5 días |
| **8** | Permitir acceso a 100 usuarios simultáneos | 0% | $4800.000 | 4 días |
| **9** | Tolerancia a fallos (caída de red, apagones eléctricos frecuentes). | 0% | n/a | n/a |
| **10** | Integración con office | 0% | n/a | n/a |
| **11** | Cumplimiento del 80% con el estándar OWASP, priorizando en las vulnerabilidades de robo de información, XSS, SQL injection y ransomware. | 60% | $360.000 | 3 días |
| **12** | Capacidad de respaldo y recuperación de información desde el software. | 0% | n/a | n/a |
|  | | **Total:** | **$5.220.000** | **34 días** |

# Atributos de calidad.

Los atributos de calidad que se utilizaran para la evaluación del Software proveedores, 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.

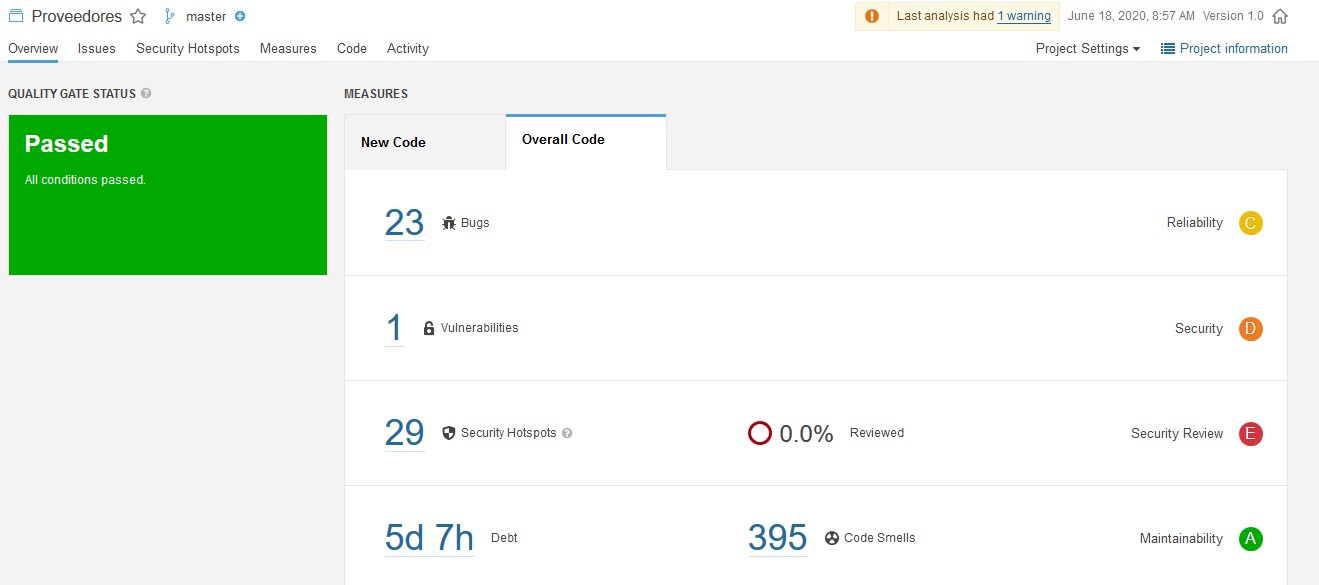
Los puntajes establecidos a los atributos de calidad seleccionados de acuerdo a las necesidades, se muestran en la siguiente tabla:

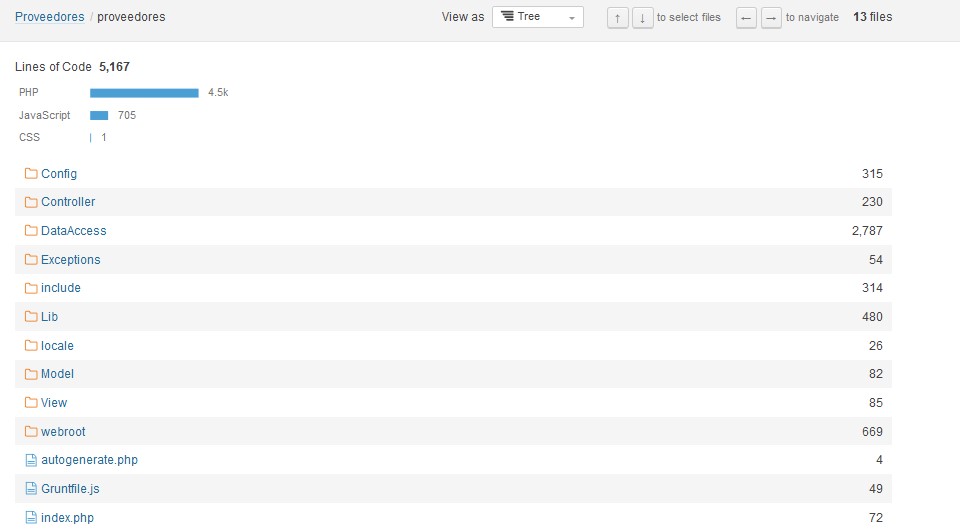
*Tabla 2, Asignación de pesos sobre la medición de atributos.*

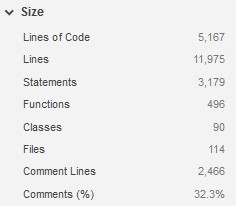
|  |  |
| --- | --- |
| Tipo de atributo | Puntaje |
| Atributos internos | 65 |
| Atributos externos | 35 |
| Total | 100 |

## Atributos internos.

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







**Densidad de comentario:**

*DoC* = *CLoC /* (*LoC* + *CLoC*)

Doc = 2466/(5,167+2466) =0.323

Doc=32,3%

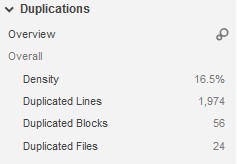
* La cantidad de comentarios en el código, es tan solo de un 32,3%, lo cual nos indica que no tiene la cantidad de comentarios que se requieren para el óptimo rendimiento del aplicativo, lo recomendado en este caso es, hacer más comentarios sobre: las clases, métodos, funciones, etc, para así, garantizar un entendimiento práctico a la hora de la mantenibilidad del código.

**Duplicidad de código:**

**-** Total líneas duplicadas: 1974

- % total de líneas duplicadas: 16,5%

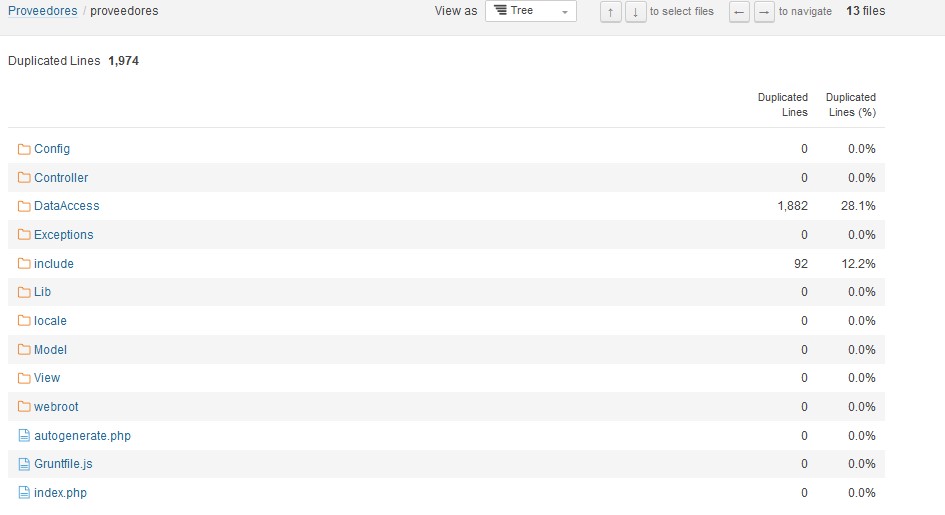
- Bloques duplicados: 56



Density of duplication =**Duplicated lines**/**Lines**\* 100

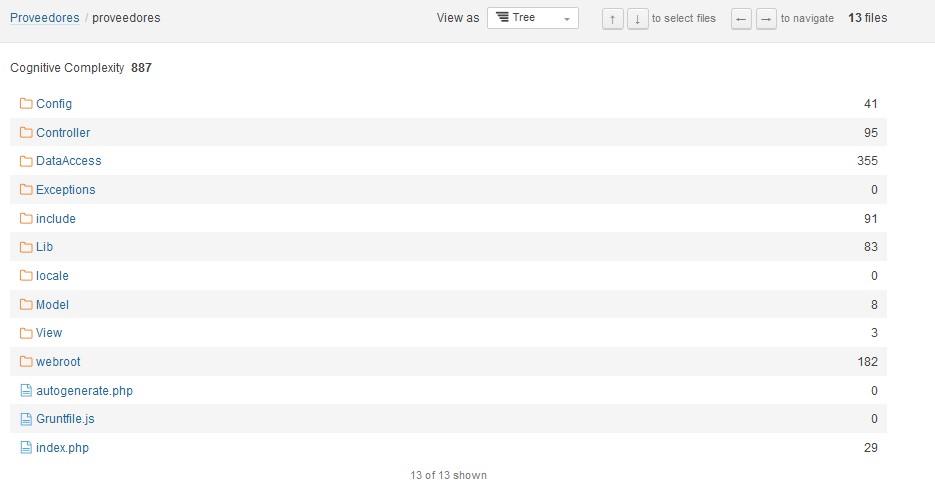
DD= 1,974/(11,975\*100)

= 11,9%



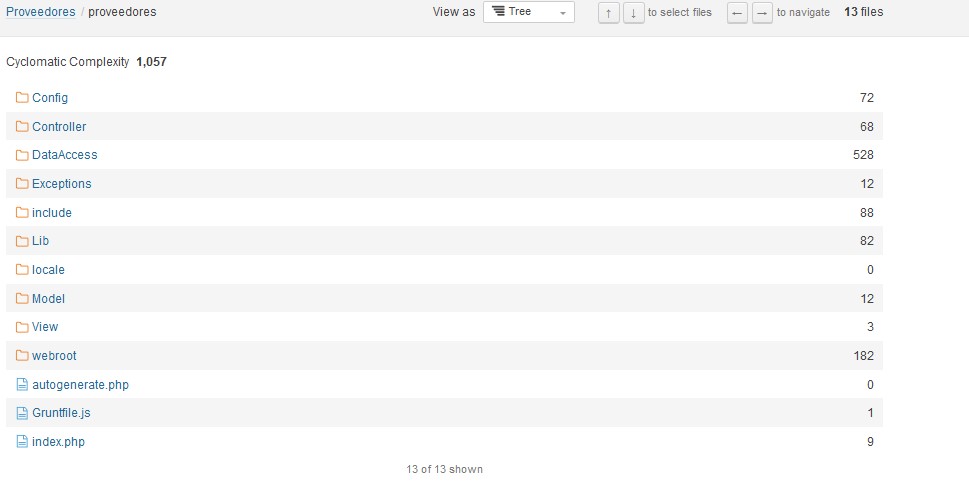
### Complejidad del software (20%)

**Complejidad Cognitiva**

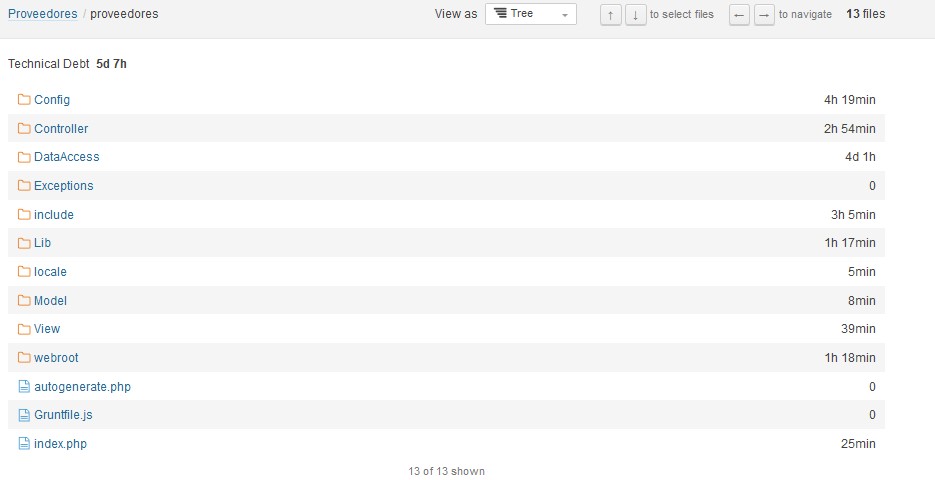


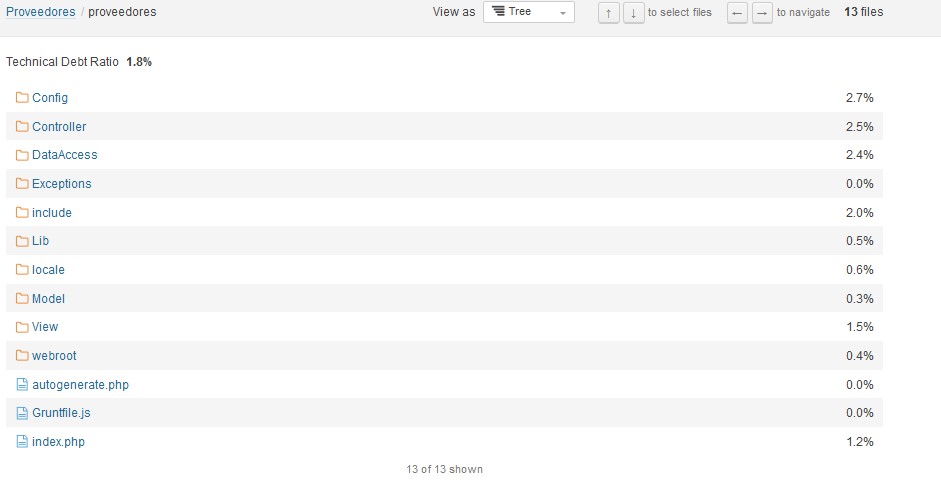
**Complejidad Ciclomatica**

Como se puede observar tiene un 1,057 de complejidad ciclomática.

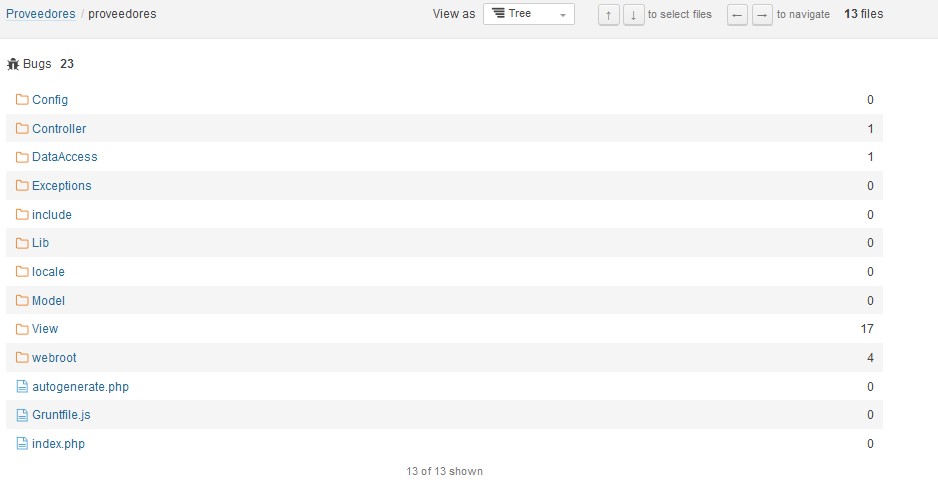
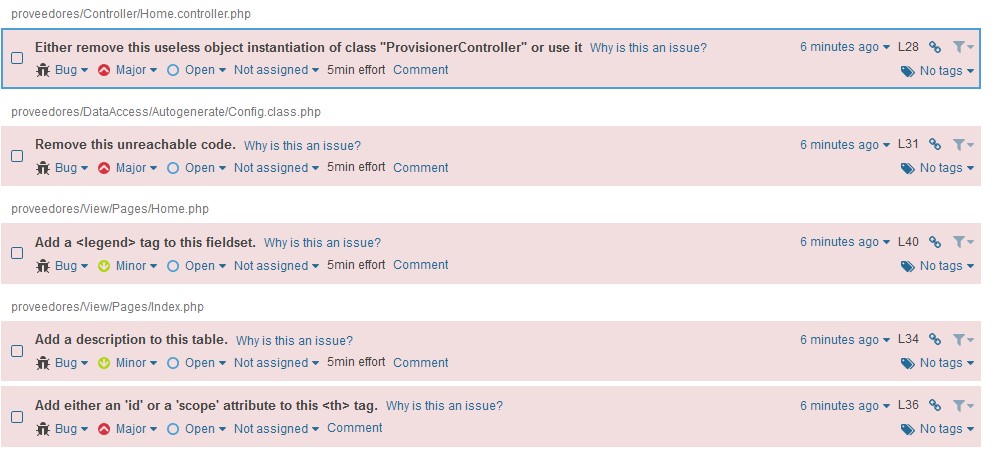
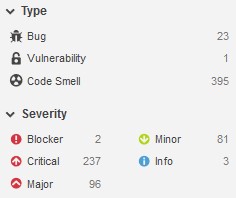
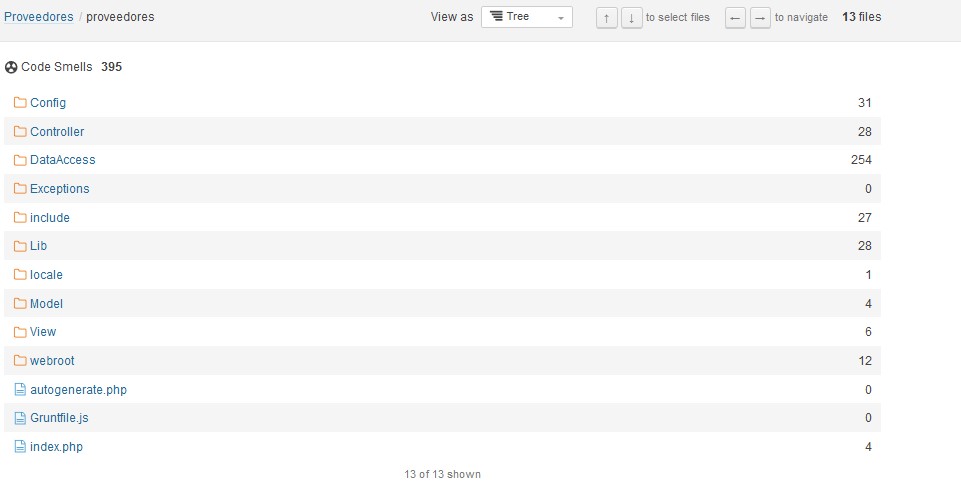


### Deuda técnica (20%)

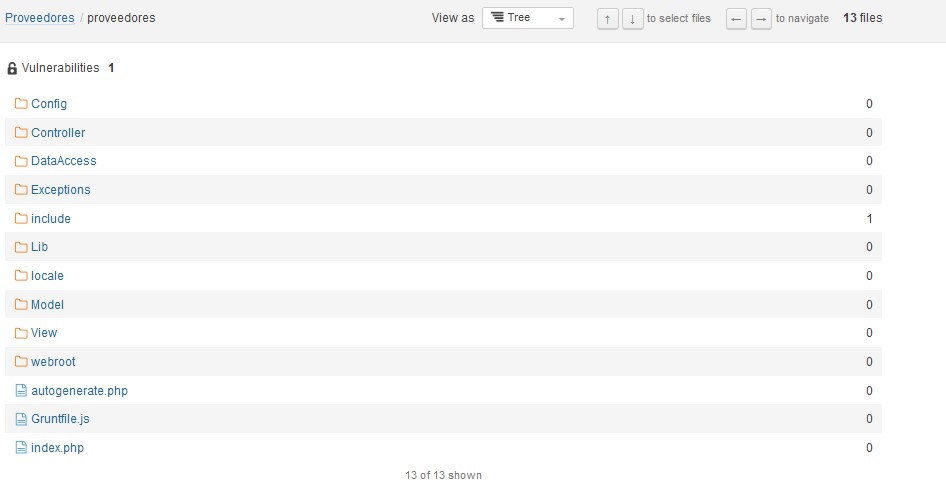




Code Smells

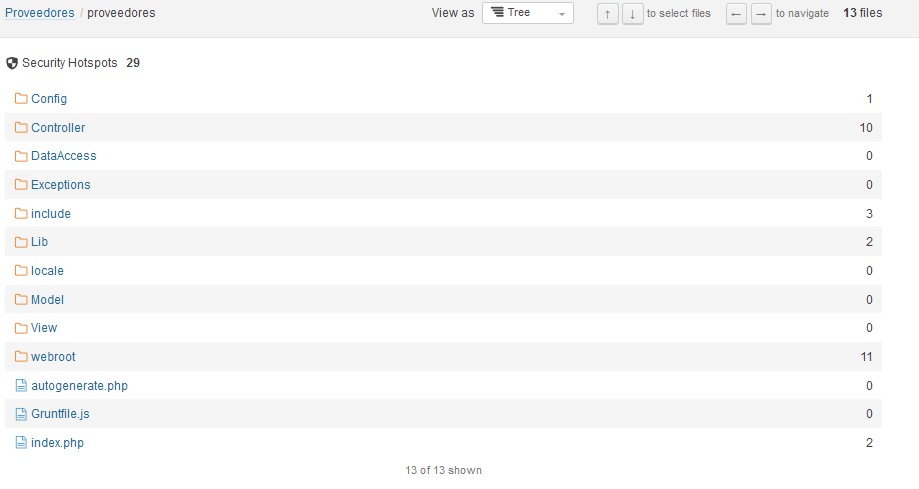


Vulnerabilidad

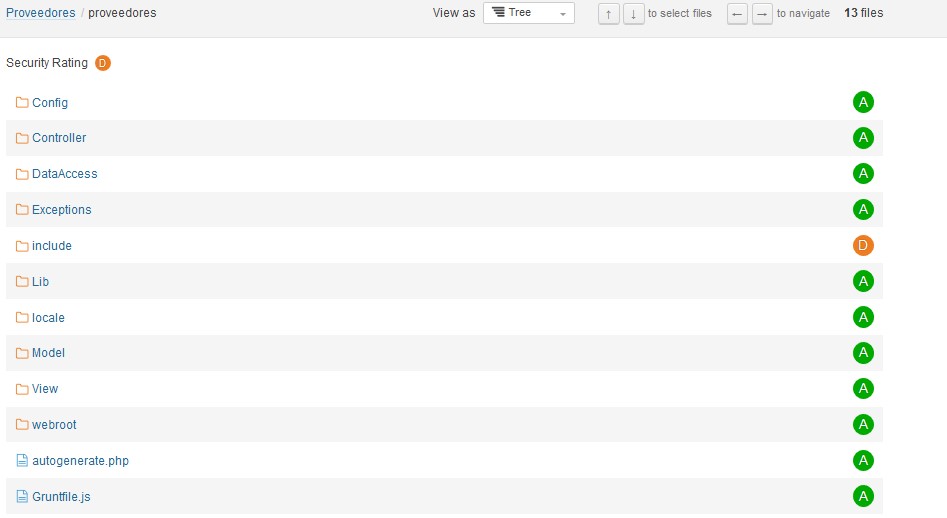


### Seguridad (20%)

Security Hotspots



Security Rating

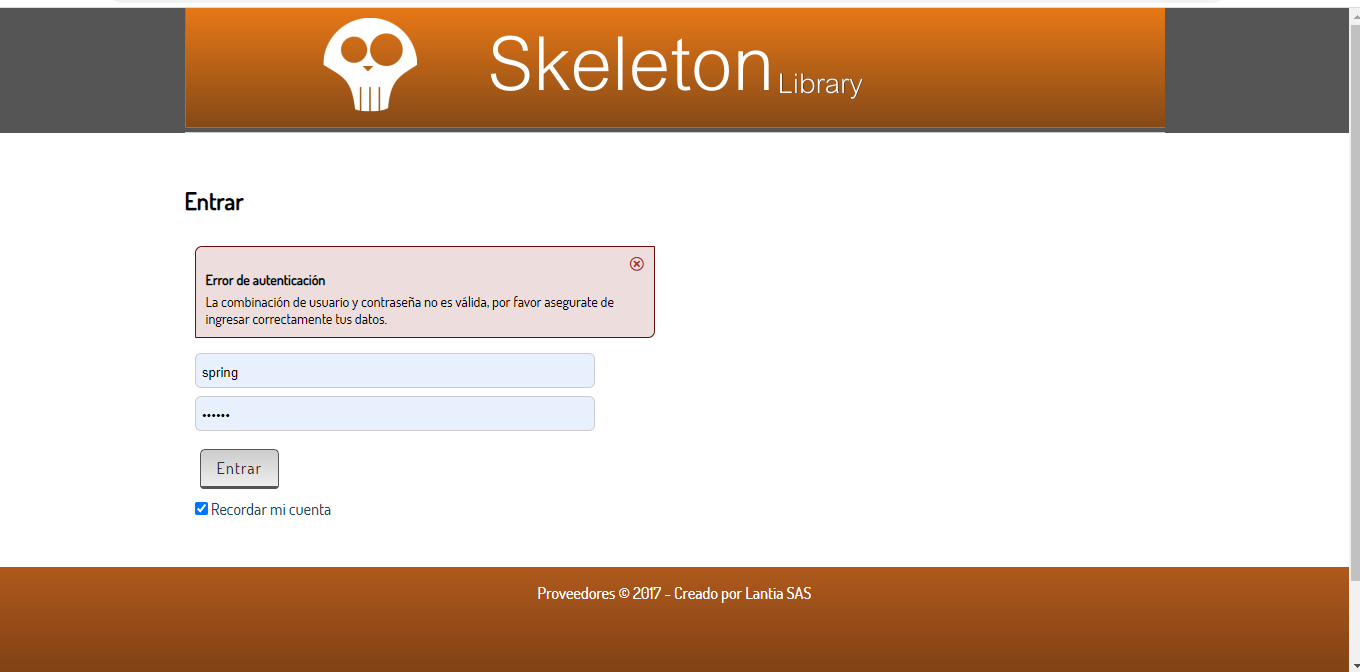


### 3d code metrics (20%)

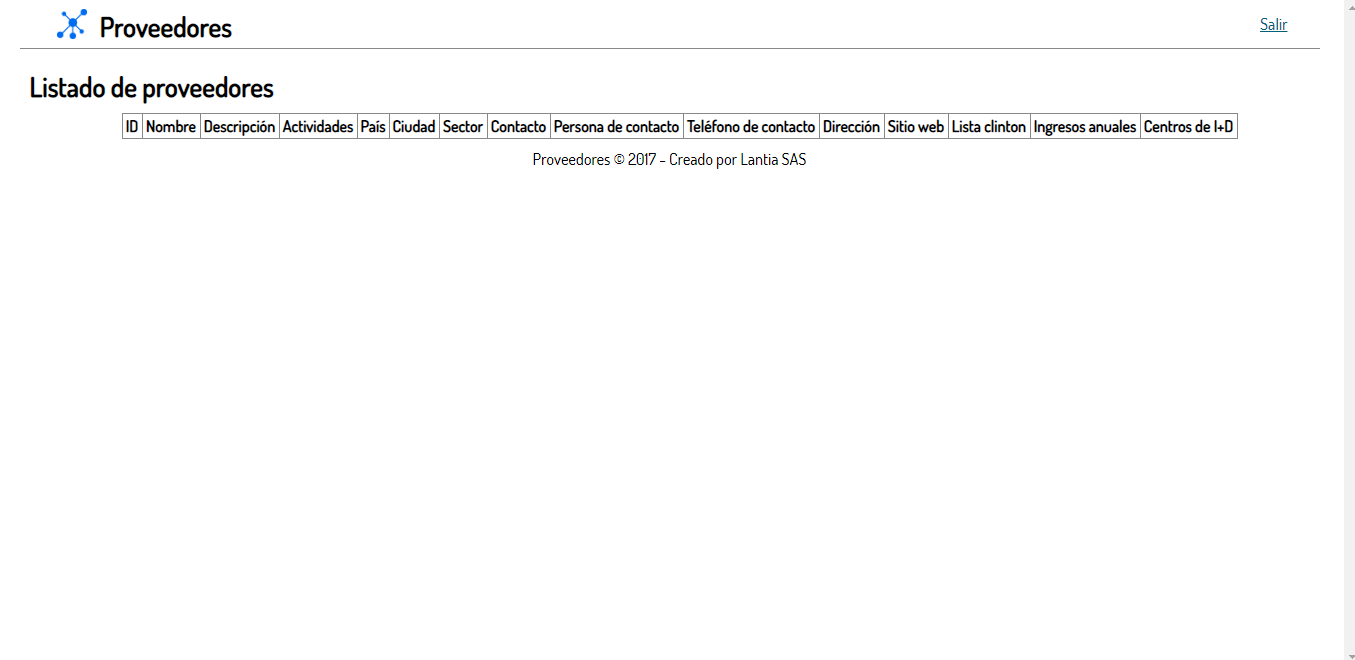
## Atributos Externos.

### Usabilidad (20%)

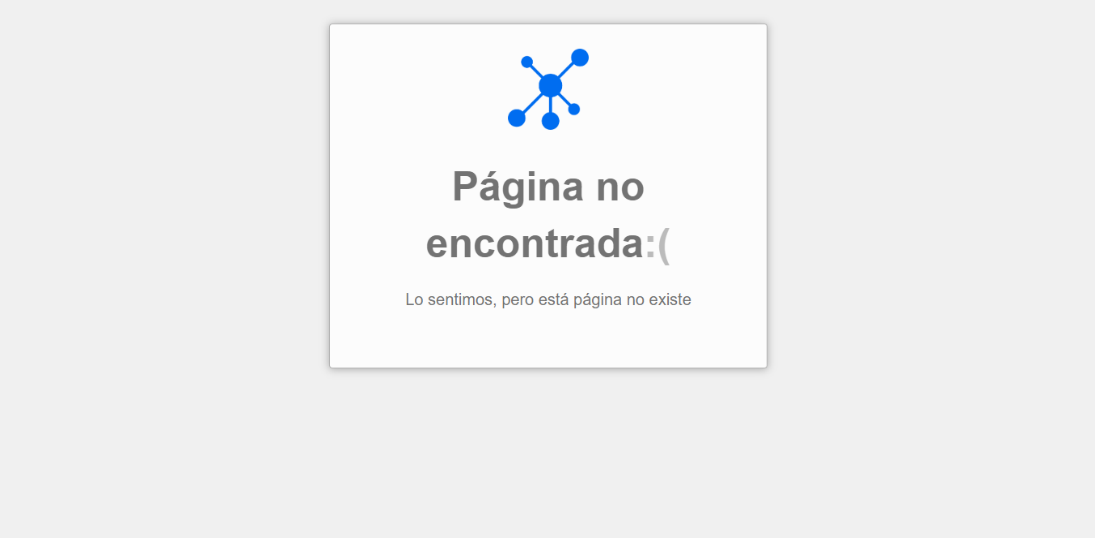
La aplicación no tiene demasiadas vistas por lo que hace difícil medir su usabilidad, solo cuenta con la página de login(img 1), página principal (img 2) y pagina de error (img 3).



Img. 1. Pantallazo de la página de login de la aplicación web



Img. 2. Pantallazo de la página de principal de la aplicación web



Img 3. Pantallazo de la página de error de la aplicación web

Evaluando los tres aspectos de la usabilidad se tiene:

* **Efectividad**: específicamente del login, funciona bien, valida los campos, usa encriptación. Los botones y demás elementos gráficos son claros y no se prestan para equivocaciones.

En este aspecto tiene una valoración del 100% debido al correcto funcionamiento.

* **Eficiencia**: la carga de la página es rápida, la navegación es básica y clara, los elementos gráficos y los botones como cerrar la sesión son fáciles de encontrar.

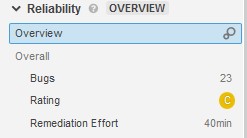
En este aspecto tiene una valoración del 100% debido a que cumple con las tareas rápidamente y es fácil de navegar.

* **Satisfacción**: no se puede hacer mucho con la aplicación web, sin embargo, posee una estructuración backend y frontend organizada para implementar nuevas funcionalidades.

En este aspecto tiene una valoración del 50% puesto que si alguien usase esta aplicación en producción no podría hacer mucho, no seria muy productivo por lo que no estaría satisfecho .

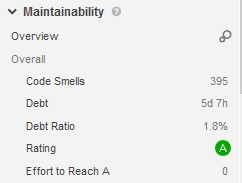
### Fiabilidad (30%)

La aplicación web cuenta con validación y encriptación de las claves usando AES CBC (algoritmo de cifrado simétrico), también maneja control de acceso a las pagina con el uso de cookies de sesión.



### Mantenibilidad (50%)

La mantenibilidad es una de las características que tiene la aplicación. El código está documentado, usa buenas prácticas de programación, usa la arquitectura de software (o patrón) MVC y DAO que lo hacen organizado y más fácil de mantener el código o solucionar un error.

****

# Recomendaciones y conclusiones

**7.1. Seguridad**

Se recomienda encriptar la información, para no exponer los datos a terceros y que esta no sea vulnerable.

Manejar estándares de la iso 25000 que aseguran que este software cumplirá con las normas y asegure la eficiencia del mismo.

**7.2. Confiabilidad**

Se recomienda arreglar los parches de seguridad y los bugs en los cuales la información de los clientes se está viendo afectada

**7.3. Usabilidad**

Se recomienda cambiar la estructura de la aplicación web para de esta manera mejorar el funcionamiento de la app

# Bibliografía

[**https://www.masreinounido.co.uk/m2m-telecom-de-eddie-velasquez/**](https://www.masreinounido.co.uk/m2m-telecom-de-eddie-velasquez/)

[**http://www.m2mtelecom.com**](http://www.m2mtelecom.com)

[**https://iso25000.com/index.php/normas-iso-25000**](https://iso25000.com/index.php/normas-iso-25000)

# Firma del perito.

|  |  |
| --- | --- |
| **Responsable de la evaluación** | **Firma** |
| Nombre:  Empresa:  Cargo:  GitHub: |  |