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

**De Telemedicina**

**Versión1.0**

**Perito:**



foto

**Nombre: Juan David Buitrago Chavez**

**Cedula: 1056804911**

**Email: juan.buitrago@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

# El propósito de la evaluación del software (Telemedicina), es determinar el cumplimiento de los diferentes atributos de calidad de este, entre ellos: deuda técnica, baches en la seguridad o código fuente, 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 tipo de software a evaluar (Telemedicina Versión 1.0), es una aplicación que tiene el propósito de suministrar soluciones tecnológicas de manera eficiente y precisa en cuanto a la prestación del servicio de ubicación de rastreo satelital para el uso de dispositivos Geepy (Dispositivos de uso cotidiano), esto, mediante el concepto y el uso de inteligencia artificial, ciberseguridad y nanotecnología. El software presenta distintos módulos para su funcionamiento, entre los cuales está: registro de usuarios, servicio de geolocalización, generación de reportes y notificaciones de usuario.

# Especificación de la(s) herramientas usadas para medición

Las herramientas que se utilizaron son:

* SonarQube es una plataforma para evaluar código fuente, además de ser un software libre, que  nos permite identificar errores en el Software antes de que llegue al usuario, de igual forma nos permite tener buenas prácticas de calidad como:
* Estándares de codificación
* Tests
* Cobertura de pruebas
* Complejidad ciclomática
* Bugs
* Comentarios
* Diseño y arquitectura Características
* Código duplicado
* Escáneres para analizar el código en función de la tecnología que vamos a usar.

Sonar Scanner:

* El cual se utiliza para analizar código en la tecnología que se va utilizar, en este caso utilizaremos SonarQube.
* 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 del 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 dónde están los errores

También se utilizaron algunos plugins como:

* Technical Debt: Este plugin es para la técnica de cada código maluco en el proyecto (lo que significa que los errores y las vulnerabilidades no contribuyen a la deuda técnica).
* Findbugs: Es un analizador de código estático de código abierto detecta posibles errores en los programas Java, se clasifican en cuatro rangos: más aterrador, aterrador, preocupante y (de preocupación
* Checkstyle: Inspeccionando el código fuente de Java y señala elementos que se desvían de un conjunto definido de reglas de codificación.

# Especificación de la Norma de calidad

# La ISO 25000 fue elegida porque otorga una visión general de los contenidos de SQuaRE(System and Software Quality Requirements and Evaluation) y de que especifican los requisitos y la evaluación de calidad del software, la ISO 25000 está compuesta por cinco áreas las cuales nos dan un parámetro más completo de cómo vamos a evaluar nuestro software la primera es la ISO 2500n: gestión de calidad, la cual es la que reúne todas las normar y es está presente en la ISO 25000 y en la ISO 25001, la siguientes son las que miden 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 que se utiliza es la ISO 2502n que es para la medición de calidad ya sean las interna, externa y en uso 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),Y es por esto que se escogió la ISO 25000 por su eficacia, su versatilidad y sobre todo su seguridad.

# 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 |

Para cada uno de los anteriores ítems deben analizar con qué porcentaje de cumplimiento está el software y en caso que no sea del 100%, dar recomendaciones para su implementación

. Determinando cuál sería el costo económico y en tiempo para su implementación.

# Atributos de calidad.

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

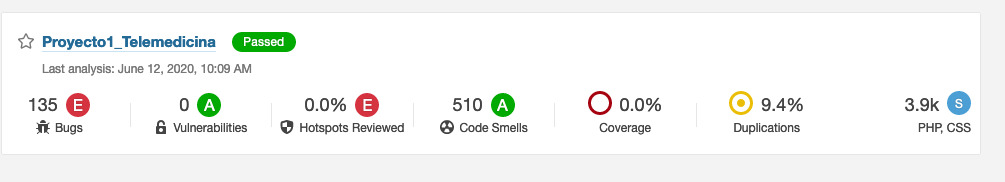
Los puntajes establecidos a los atributos de calidad seleccionados de acuerdo a las necesidades, se muestran en la siguiente tabla:

*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%)

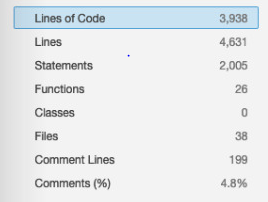


**Densidad de Comentarios:**

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

Doc = 199/(3.938+199) =0,048

Doc=4,8%



La cantidad de comentarios en el código, es tan solo de un 4,8%, lo cual nos indica que no tiene la cantidad de comentarios que se requieren, lo recomendado en este caso es hacer más comentarios sobre: todas las clases, los métodos, las funciones, entre otras, para así, garantizar la mantenibilidad del código.

**Duplicidad de código:**

**Número de líneas por método:** 0



**Número de líneas por función:** 26



**Número de líneas por procedimiento:** 2,005



### Complejidad del software(20%)

Falta aca complejidad ciclomática y cognitiva

* **Complejidad Ciclomática:**
* **Complejidad Cognitiva:**

### Deuda técnica (20%)

### Seguridad(20%)

En esta sección ustedes deben evaluar las posibles vulnerabilidades de seguridad tomando como referencia OWASP encontradas en el proyecto de software (recuerden agregar gráficos e imágenes obtenidas de sonarQube como evidencia de la evaluación).

### 3d code metrics (20%)

En esta sección ustedes deben implementar el plugin **3D CODE METRICS(**<https://www.excentia.es/3d-code-metrics>) en su SONAQUBE y generar la imagen 3D del proyecto, con esta información presentar un evaluación del código (zonas en conflicto, patrones y anomalías )

## Atributos Externos.

### Usabilidad (20%)

En esta sección ustedes deben medir (si es superior al 80% es bueno, por debajo es malo):

* la efectividad, por ejemplo: botones tengan aspecto de botones, manejo de errores, etc
* la eficiencia: sistema lógico de navegación en el software, coherencia en los nombres de los objetos y su funcionalidad, etc.
* Satisfacción: La comodidad y la aceptabilidad que se puede dar en el uso del software (¿recomendarías el software a un amigo después de usarlo durante dos horas?).

### Fiabilidad (30%)

En esta sección ustedes deben medir la prevención, tolerancia a averías/fallos (transitorios, intermitentes o permanentes) en el funcionamiento del software producto de errores en diseño, por deuda técnica, daños de hardware, comunicación o por malware/virus)

### Mantenibilidad (50%)

# Para este software la mantenibilidad no es fácil de efectuar debido a que se encuentran múltiples bugs incumpliendo con los diferentes estándares y normas establecidos Recomendaciones y conclusiones

## Seguridad

## Confiabilidad

## Usabilidad

# Bibliografía

<https://drive.google.com/drive/u/1/folders/11BD9mqLbmsqAJGK4Bp8BNMkaTcT4UN5B>

[**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: Juan David Buitrago Chavez  Empresa: ShiftLogiQ  Cargo: Gerente general  x |  |