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

**tpv.restaurant  
Versión 1.0**





**Nombre: Edgar Santiago Jimenez Ortega**

**Cedula: 1049641149**

**Email: Edgar.jimenezo@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

La presente evaluación se realizara con el objetivo de analizar la calidad del proyecto en concreto, de esta manera saber si el proyecto en cuestión puede llegar a ser calificado de calidad o no, y con base en esta evaluación realizar las respectivas verificaciones a nivel de presentación, funcionalidad y calidad del código para así estar seguros que el software cumple a cabalidad las funciones para las que fue creado, además de esto dar una guía a los desarrolladores para que reconozcan sus errores y puedan arreglarlos para no recaer en ellos, en los próximos proyectos que vayan a ser desarrollados, además gracias a este tipo de evaluaciones deberíamos ser capaces de reducir el impacto y tiempo que vaya a ser empleado en el proceso de mantención del software en un futuro, a corto, mediano o largo plazo.

# Producto a evaluar

Se procederá a realizar evaluación de calidad al software [tpv.restaurant](https://github.com/alecsvaldez/tpv.restaurant) (Administrador y Punto de venta para restaurantes, cafeterías, bares) este software cuenta con varios módulos a trabajar entre ellos encontramos, un modulo de control para las funciones de cocina, compras, ventas, inventario, caja y por su puesto el módulo de tpv (terminal punto de venta) con este software se busca manejar toda la información de venta y manejo de insumos(Inventario: ingresos y egresos de elementos del restaurante ) de un restaurante en concreto en este caso el restaurante LA TERRAZA, en cuanto a la versión del software nos encontramos con la versión 1.0 .

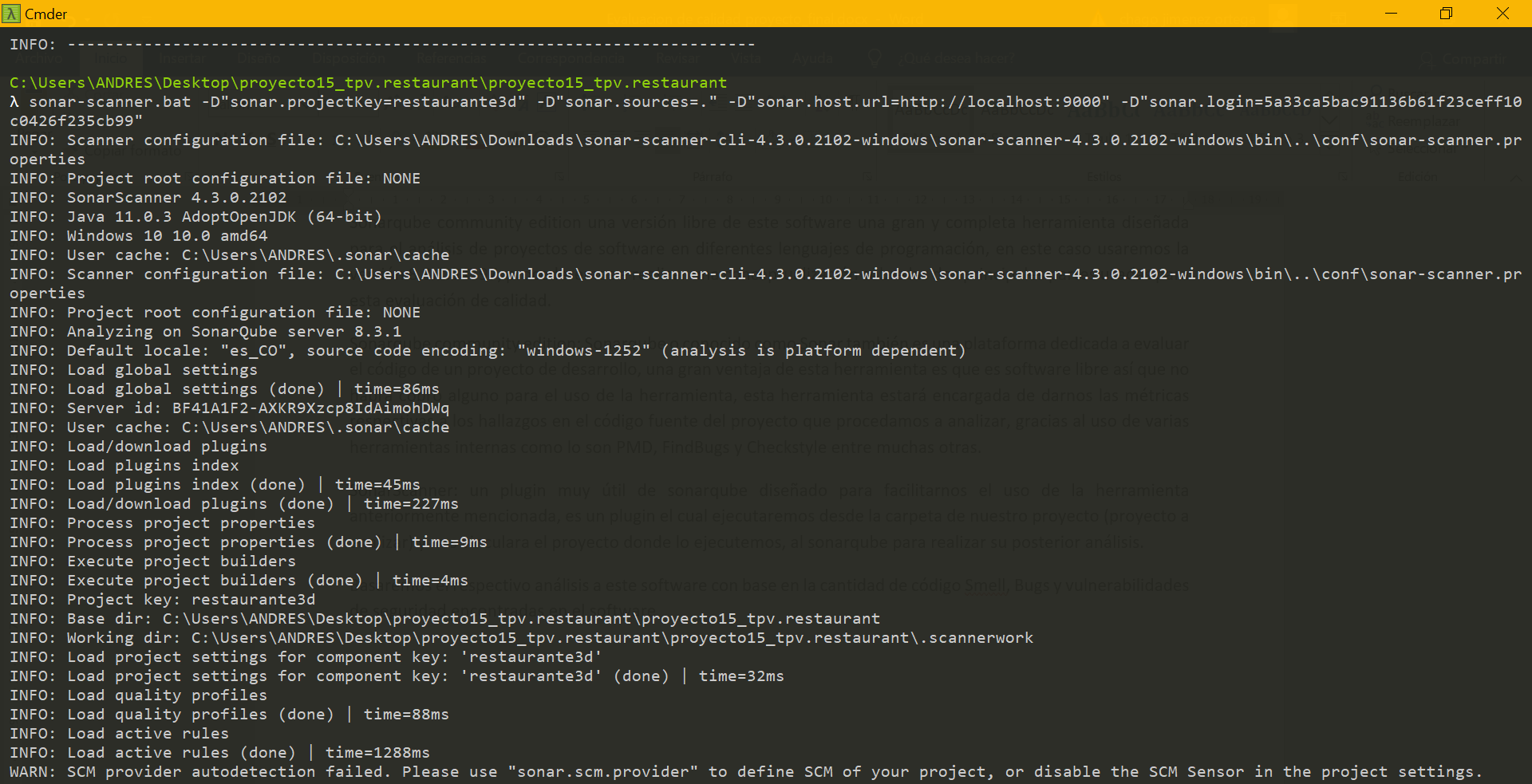


# Especificación de la(s) herramientas usadas para medición

L a herramienta que se va a utilizar para la realización de esta evaluación de calidad de software es la herramienta Sonarqube community edition una versión libre de este software una gran y completa herramienta diseñada para el análisis de proyectos de software en diferentes lenguajes de programación, en este caso usaremos la versión 8.3.1.34397, procederemos a realizar una conceptualización sobre las dos principales partes usadas para esta evaluación de calidad.

Sonarqube community edition: Sonarqube o conocido como Sonar también es una plataforma dedicada a evaluar el código de un proyecto de desarrollo, una gran ventaja de esta herramienta es que es software libre así que no habrá cobro alguno para el uso de la herramienta, esta herramienta estará encargada de darnos las métricas respectivas a los hallazgos en el código fuente del proyecto que procedamos a analizar, gracias al uso de varias herramientas internas como lo son PMD, FindBugs y Checkstyle entre muchas otras.

SonarScanner: un plugin muy útil de sonarqube diseñado para facilitarnos el uso de la herramienta anteriormente mencionada, es un plugin el cual ejecutaremos desde la carpeta de nuestro proyecto (proyecto a analizar) y nos vinculara el proyecto donde lo ejecutemos, al sonarqube para realizar su posterior análisis.



Basaremos el respectivo análisis a este software con base en la cantidad de código Smell, Bugs y vulnerabilidades de seguridad encontradas en el software



# Especificación de la Norma de calidad

Teniendo en cuenta que la ISO 25000 hace referencia a la familia de normas encargada de realizar un correcto análisis de los datos de un software para determinar si es o no de calidad, es la norma sobre la cual nos basaremos para realizar el presente análisis.

¿Por qué ISO 25000?, es sencillo, la familia de normas ISO 25000 es proporcionada por la International Standard Organization (Es una organización no gubernamental con membresía de 161 cuerpos nacionales, ubicada en Suiza. A través de sus miembros, reúne a expertos en distintas áreas para compartir sus conocimientos y desarrollar **Estándares Internacionales (conocidos como normas ISO) que apoyen la investigación y proporcionen soluciones a desafíos globales**) [1] por loque se entiende que será un estándar mundial, con lo que a software se refiere, pocas normas son aplicadas al software a nivel mundial en su mayoría, cada desarrollo se rige por su impacto en cierta posición geográfica y lo que para un país en concreto es ilegal puede que para otro sea completamente permitido, por este motivo se debe aprovechar este tipo de oportunidades bajo las cuales se puede regir de manera literal todo el mundo.

Es correcto e importante realizar cierta aclaración, esta familia de normas busca, la organización y unificación de 2 ámbitos principales: especificar los requisitos de calidad de un software y evaluar la calidad del mismo.

Por este motivo los análisis de calidad se guían por esta familia de normas que no solo dan las pautas para evaluar, sino también una guía para realizar correctamente este proceso.

# Requerimientos específicos de calidad funcionales y técnicos.

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

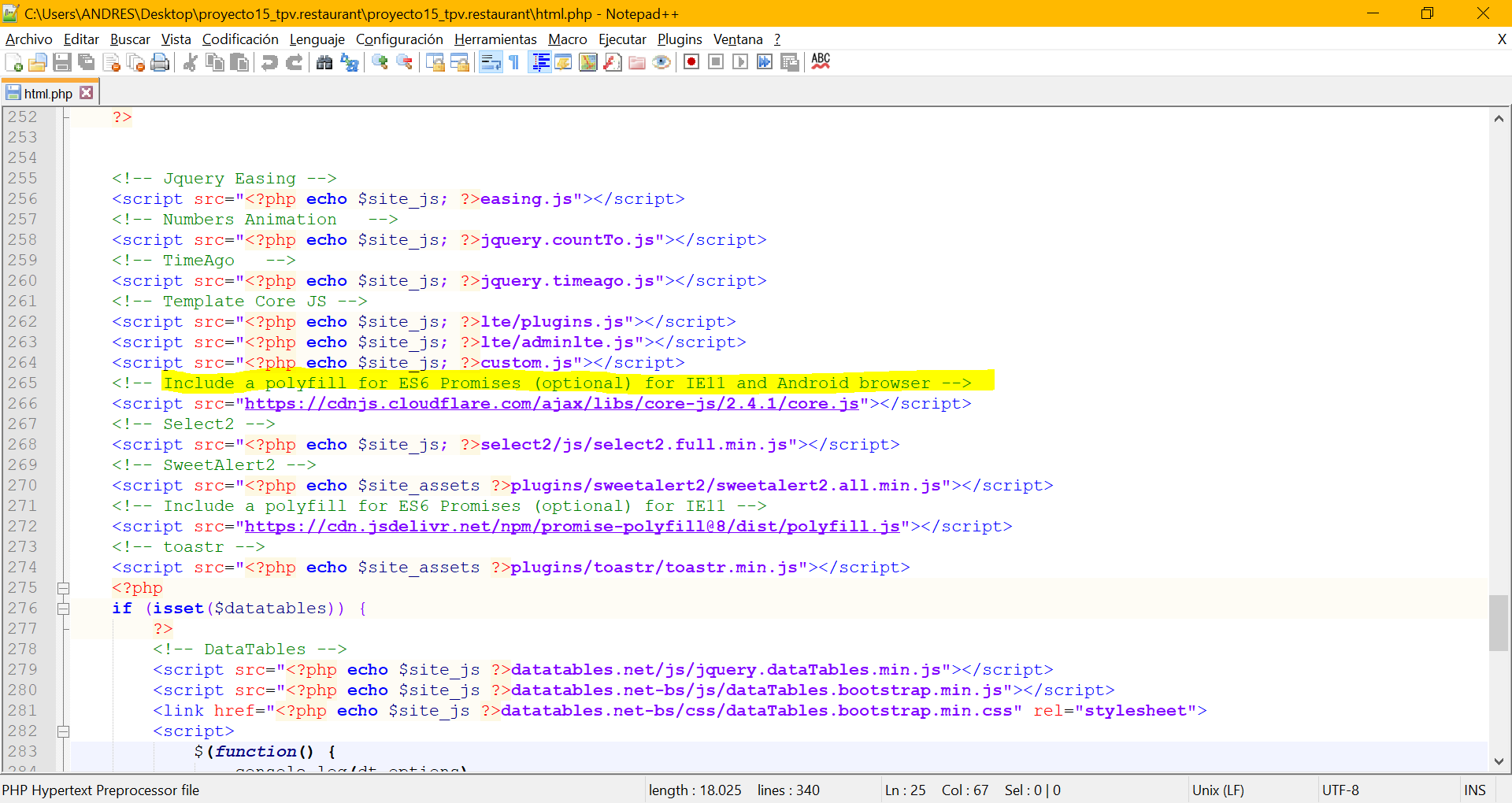
|  |  |  |
| --- | --- | --- |
| Ítem | Requerimiento de calidad | Prioridad |
| 1 | Que el software pueda funcionar en sistemas operativos Android, MacOS, Windows XP, Windows 7 y Windows 10 (en 32 y 64 bits) | Alta |
| 2 | Que permita trabajar en forma rápida e intuitiva (cuente con ayudas visuales 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 |

Desglose de análisis:

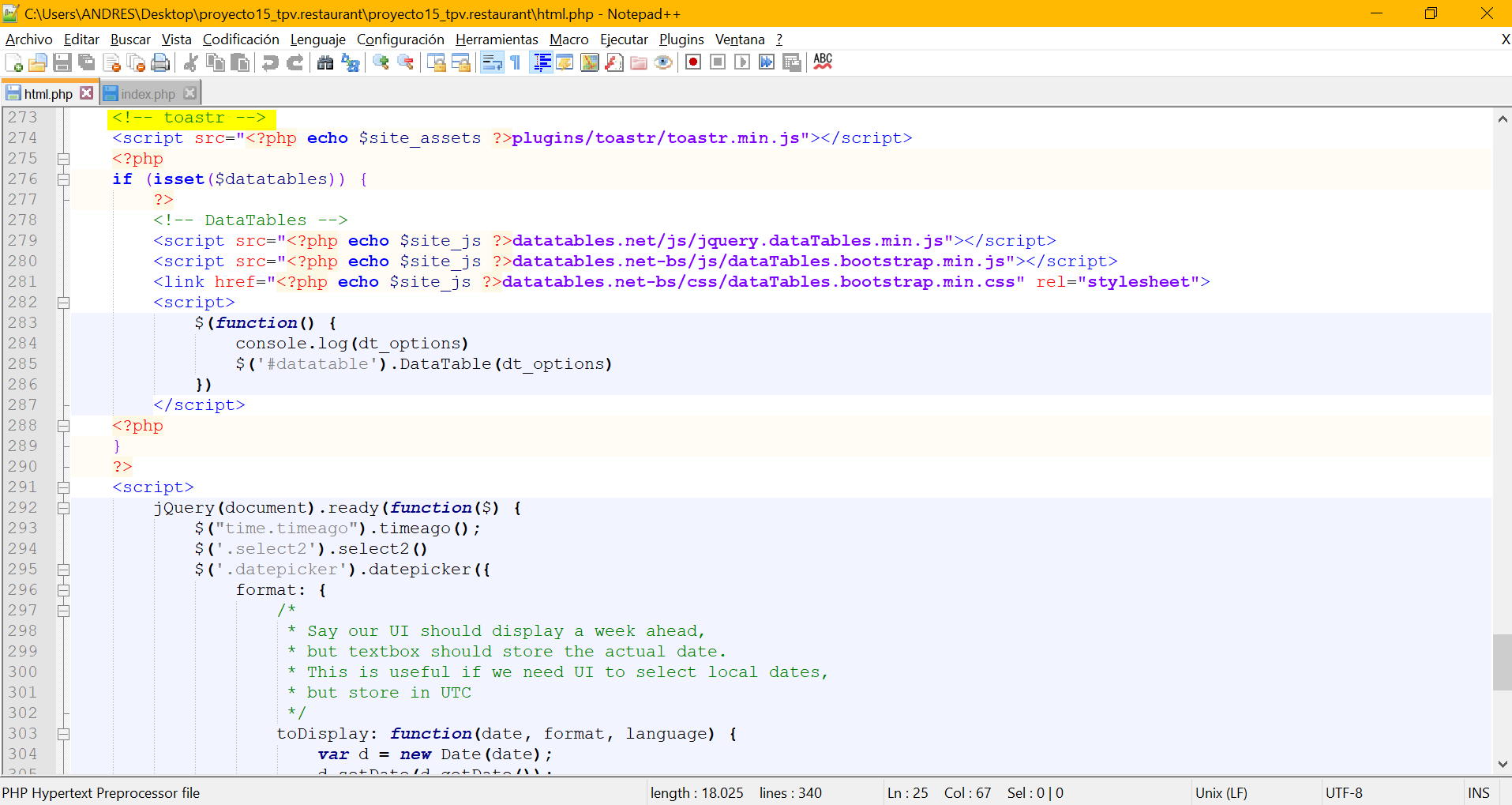
1.El software cumple con los requerimientos establecidos en el numeral uno: no solo teniendo en cuenta que este software esta manejado bajo una interfaz web y que los navegadores serán capaces de ejecutarlo sin problema sino que los mismo desarrolladores cuidaron al detalle, el añadir referenciación a librerías que entrarían a manejar la ejecución del software en si en distintos sistemas operativos o navegadores como los son Android o incluso Internet Explorer 11. Lo podemos evidenciar en el apartado de html.php

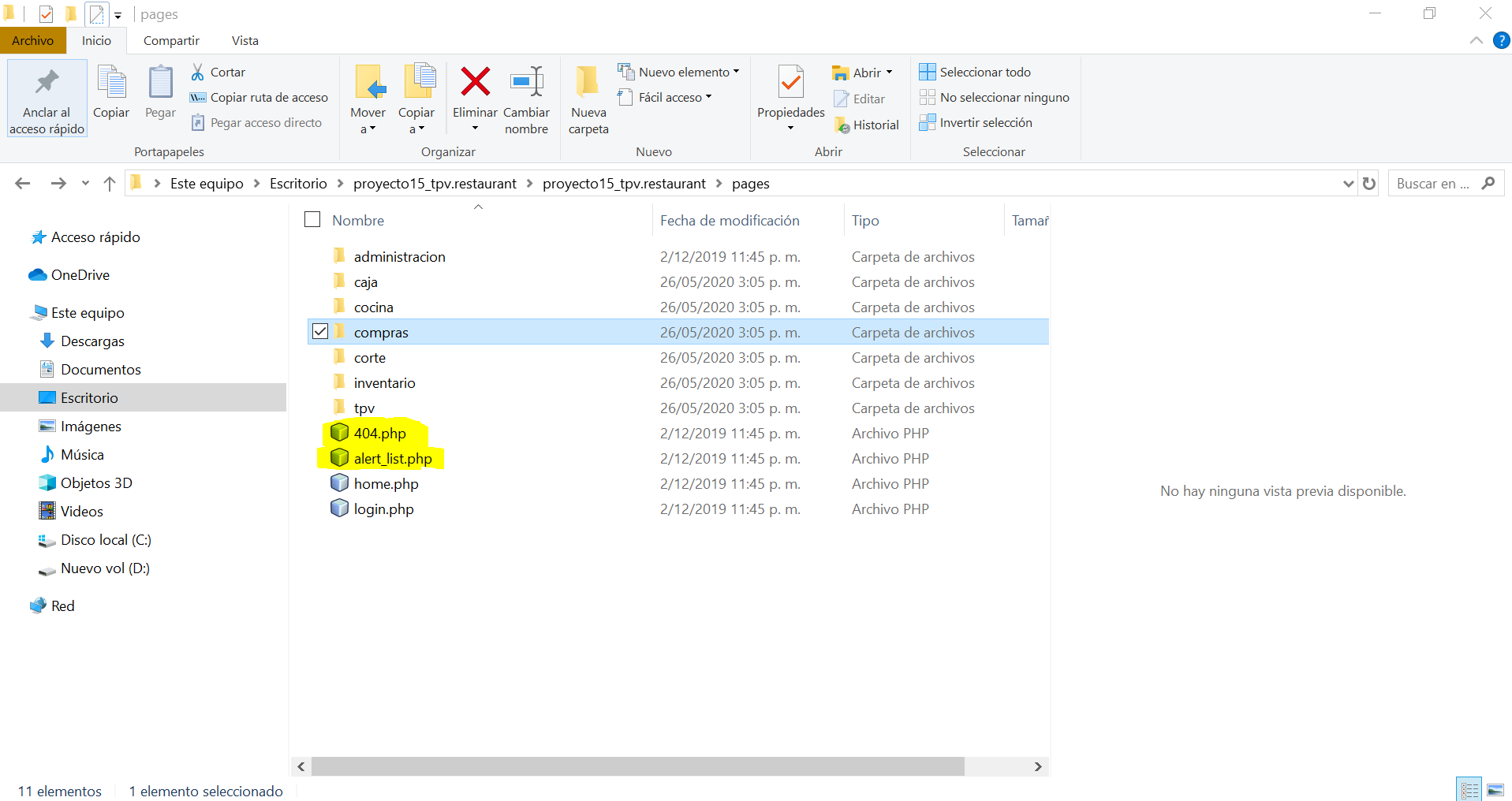
+



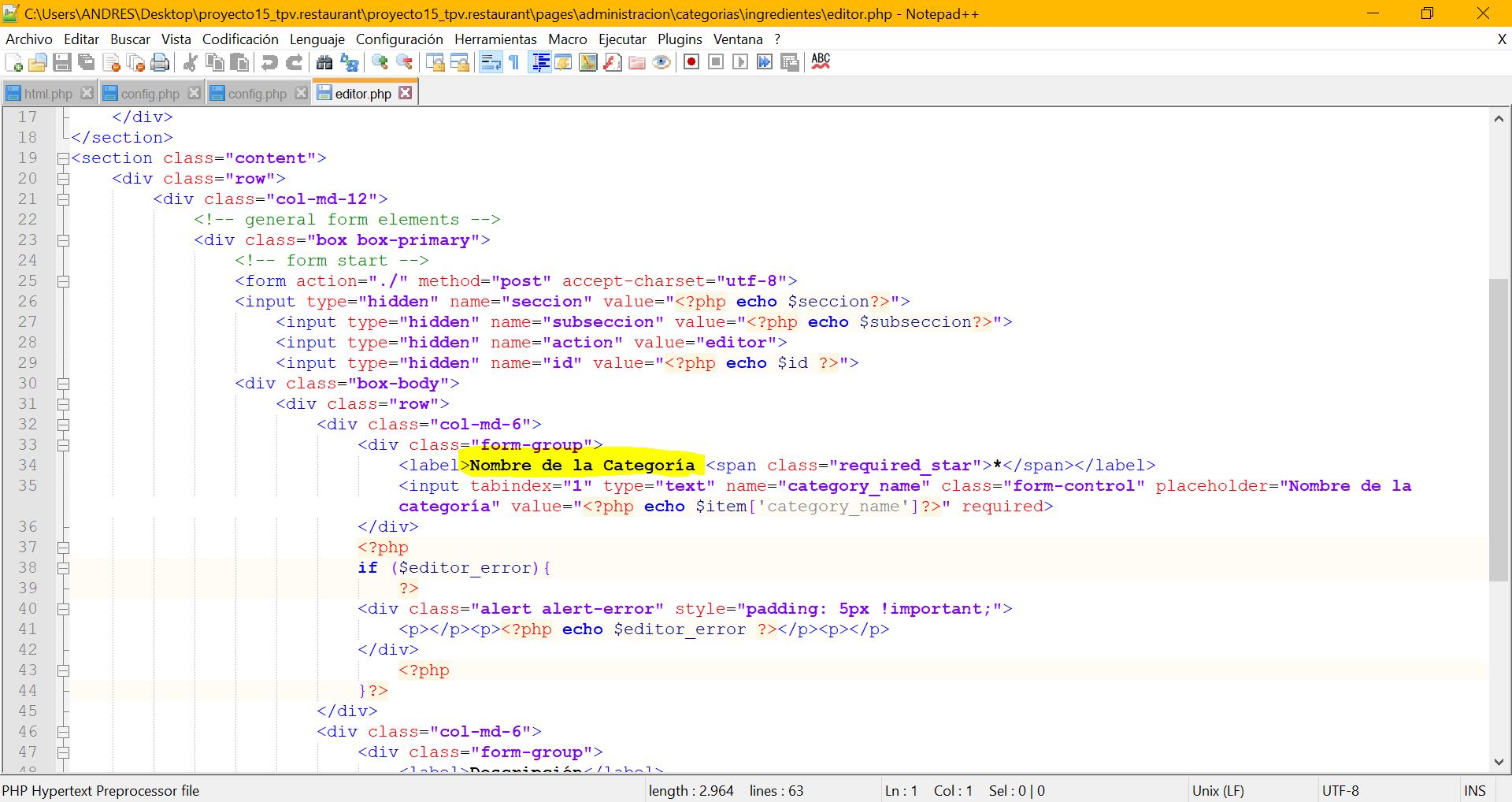


2.Se evidencia un cumplimiento con los requerimientos establecidos en el numeral 2, los desarrolladores desarrollaron el software con varias herramientas “user-friendly” como por ejemplo toastr encargado de enviar notificación graficas flotantes al usuario cuando haya algún evento importante a desarrollar en el software que se encuentra manejando, poseen un plugin de teclado para poder manejar datos de entrada escritos por medio de la misma aplicación(hemos de tener en cuenta que la mayoría de estos software serán ejecutados en pantallas táctiles para facilitar su uso a los empleados del negocio que requiera el programa).

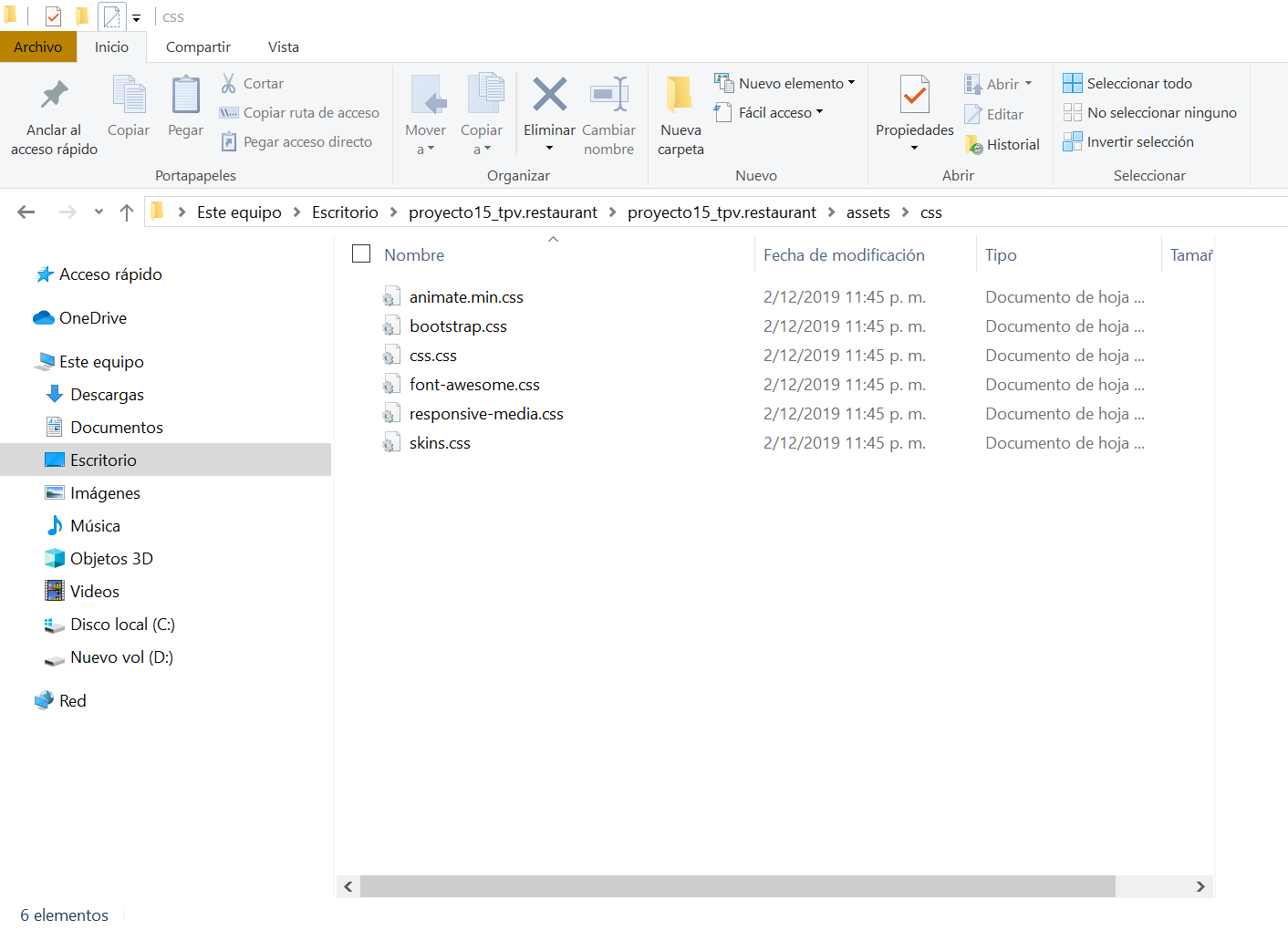


Aparte de esto realizaron un respectivo manejo de excepciones con respecto a posibles errores que el usuario fuera a encontrarse, por un fallo del software o un mal uso del mismo proporcionado por el usuario a cargo.  


3.En este caso no se evidencia un uso de un segundo idioma aparte del idioma español, se recomienda a los desarrolladores realizar una adaptación completa del software con respecto a este punto, por que, aunque el software sea un pedido muy especifico de la empresa que se encuentra comprando el producto, como buena practica de desarrollo, al día de hoy es recomendable como mínimo realizar la adaptación de software completo en al menos dos idiomas.



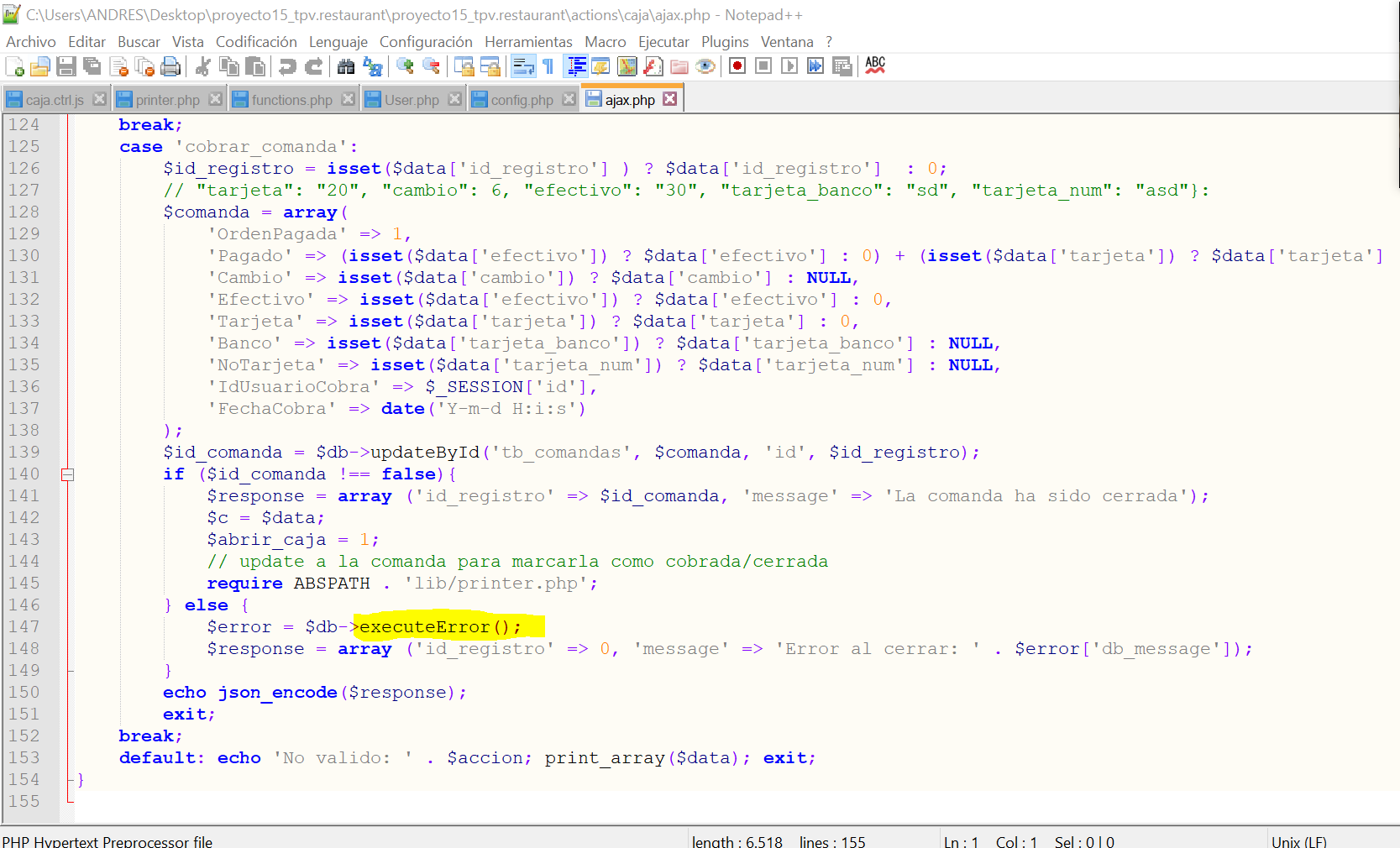
4.Vemos que se encuentran implementadas las Bootstrap.skins para poder ser implementadas desde mismo software, se puede decir que la apariencia del aplicativo es configurable con respecto a las necesidades y gustos de usuario, aun así, en la configuración inicial del software no se visualiza una opción para personas con diferentes discapacidades visuales, se recomienda a los desarrolladores realizar la inclusión de estas opciones en los respectivos menús de configuración.



5.Las bases de datos generadas en este proyecto guardan información personal de los usuarios respectivos, además de guardar la información de los empleados, información que fácilmente podría ser filtrada teniendo en cuenta que a pesar de que MySql es un motor de bases de datos bastante rápido al momento de procesar la información a su vez es uno de los menos seguros a comparación de otros motores, se recomienda a los desarrolladores realizar una migración de la información a un motor mas seguro que siga proporcionando la fiabilidad y funcionamiento de su software

6.No se encontró ningún archivo de configuración o conexión con respecto a lo que office se refiere por ende no se hace llamado en ningún momento a exportar archivos en Excel o documentos de texto Word (docx) y tampoco se evidencia una conexión o llamado a la exportación de documentos PDF, se le4 recomienda a los desarrolladores incluyan estas funciones en el software teniendo en cuenta que una de las mejores herramientas al momento de realizar cálculos matemáticos paralelos es Excel, siendo esta una herramienta fácil y rápida para utilizar.

7.Se evidencian la invocación de pop up´s con los respectivos errores que pueden llegar a presentarse durante el uso del software, para hacerle entender al usuario que es lo que esta haciendo mal, o que es lo que le esta fallando al software.

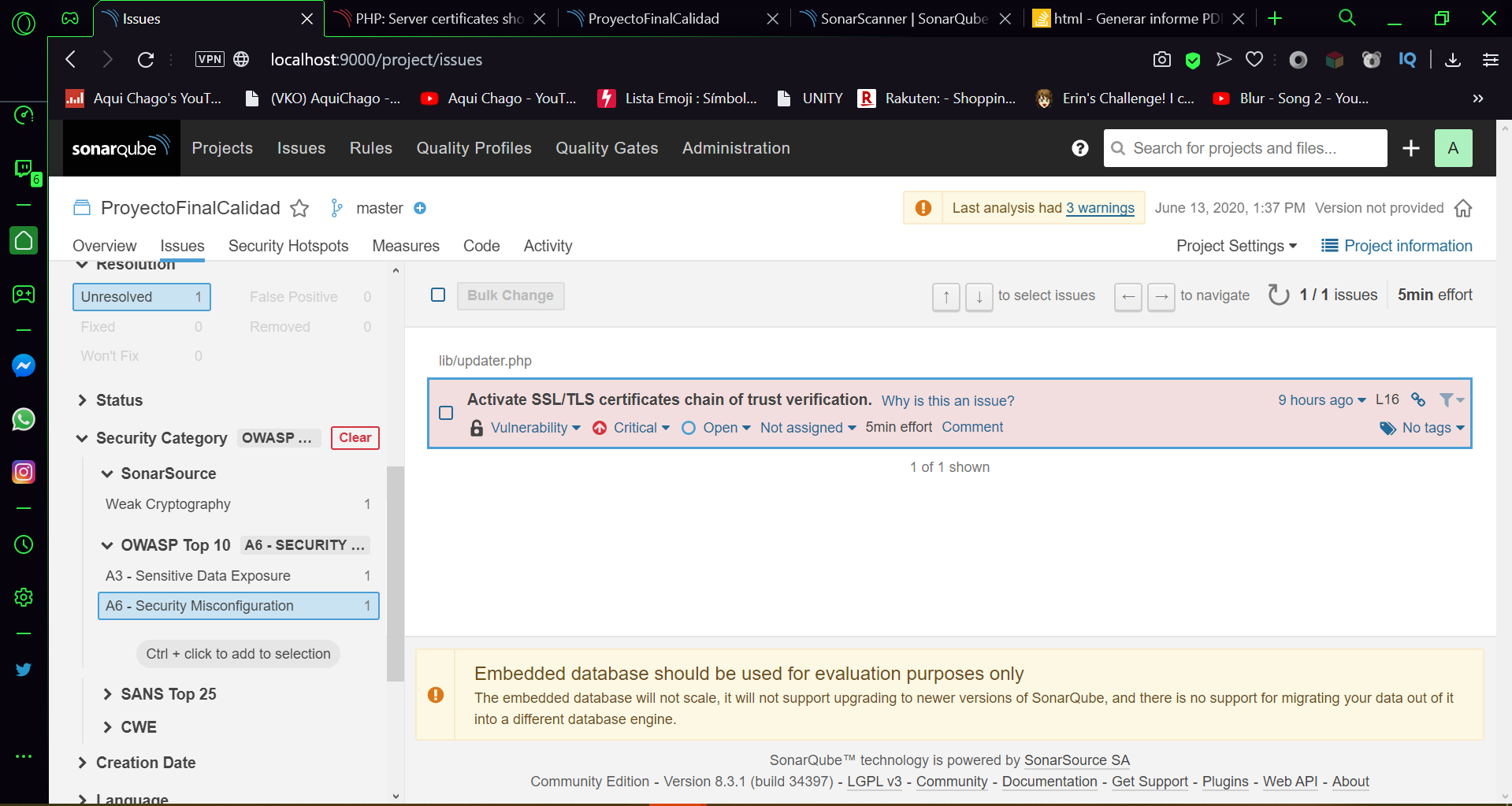


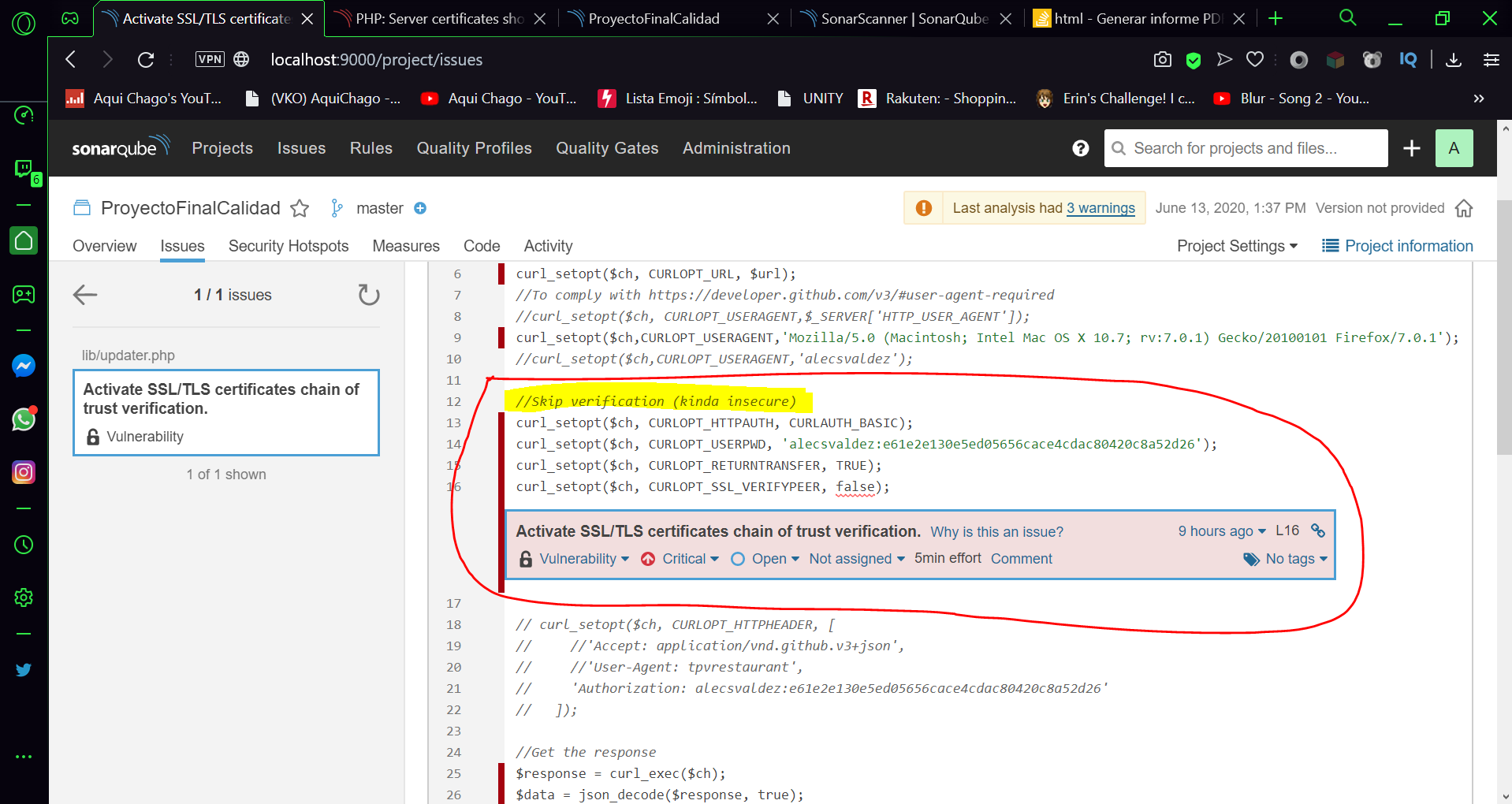
8.No se evidencia que el software pueda llegar a soportar una carga de 100 usuarios simultáneos, aunque teniendo en cuenta que es un software para un restaurante de tamaño relativamente pequeño, no se considera necesario realizar este ajuste a no ser que sea estrictamente necesario y que el software vaya a ser comercializado con una empresa de un gran tamaño.

9.No se evidencia un sistema de seguridad que proteja el sistema o la información en caso de un fallo imprevisto, se recomienda a los desarrolladores realizar un sistema de autoguardado, para que, en el caso de un apagón, o pico de energía eléctrica, la información no se pierda.

10.Referirse al numeral #6.

11.Se evidencia un cumplimiento con los protocolos OWASP a excepción de una vulnerabilidad encontrada en un código de prueba de acceso.





12.La información desde el software podrá ser guardada y manejada en la base de datos local, así de esta manera se puede realizar una copia de seguridad, como respaldo en caso de una emergencia.

13.Se encontró que el software cumple con los mínimos requisitos de rendimiento para ser ejecutado en cualquier equipo de computo hasta con los componentes mas antiguos y de menor rendimiento.

# Atributos de calidad.

Los atributos de calidad que se utilizaran para la evaluación del Software [tpv.restaurant](https://github.com/alecsvaldez/tpv.restaurant), 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:

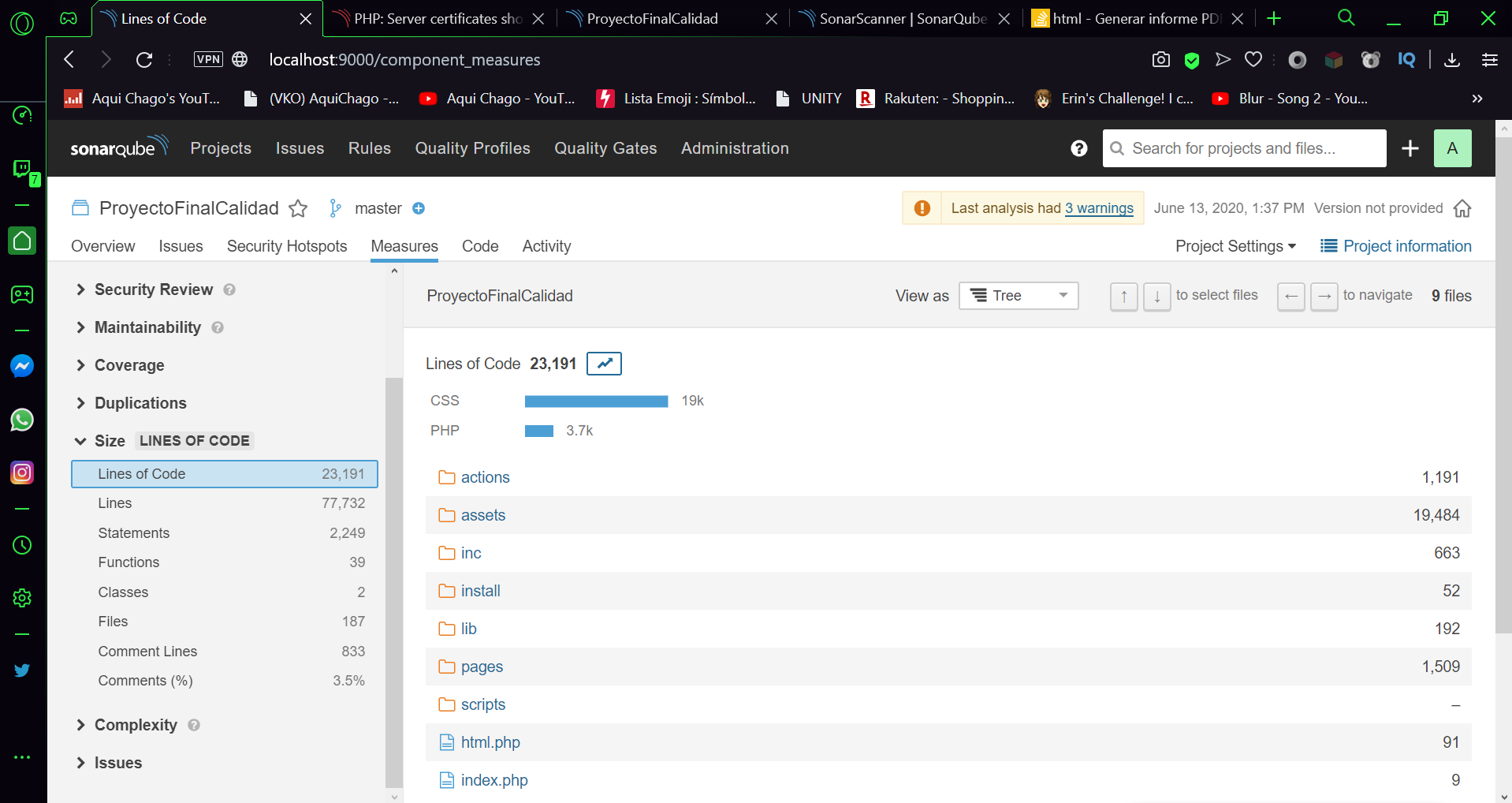
*Tabla 2, Asignación de pesos sobre la medición de atributos.*

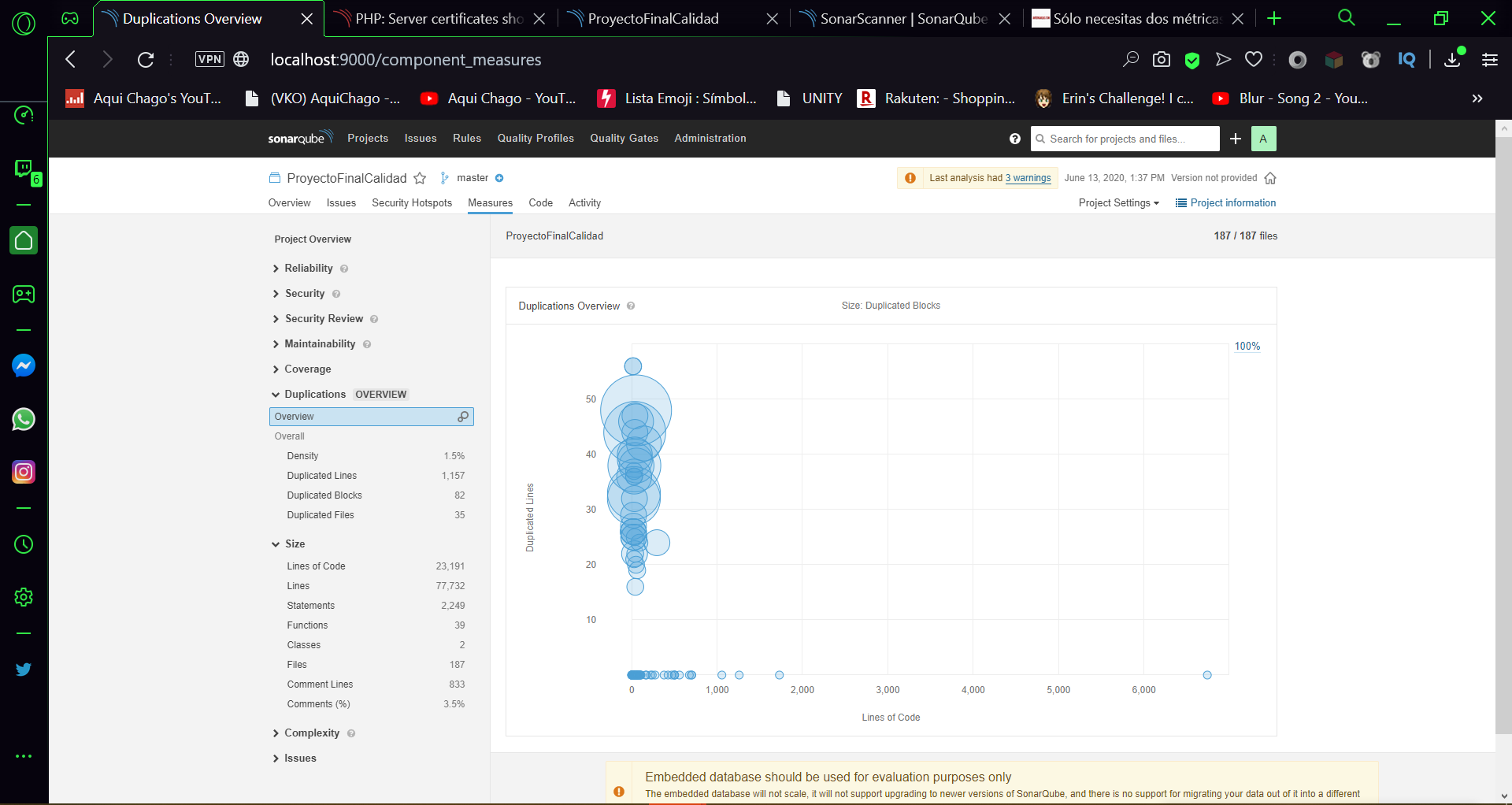
|  |  |
| --- | --- |
| Tipo de atributo | Puntaje |
| Atributos internos | 65 |
| Atributos externos | 35 |
| Total | 100 |

## Atributos internos.

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

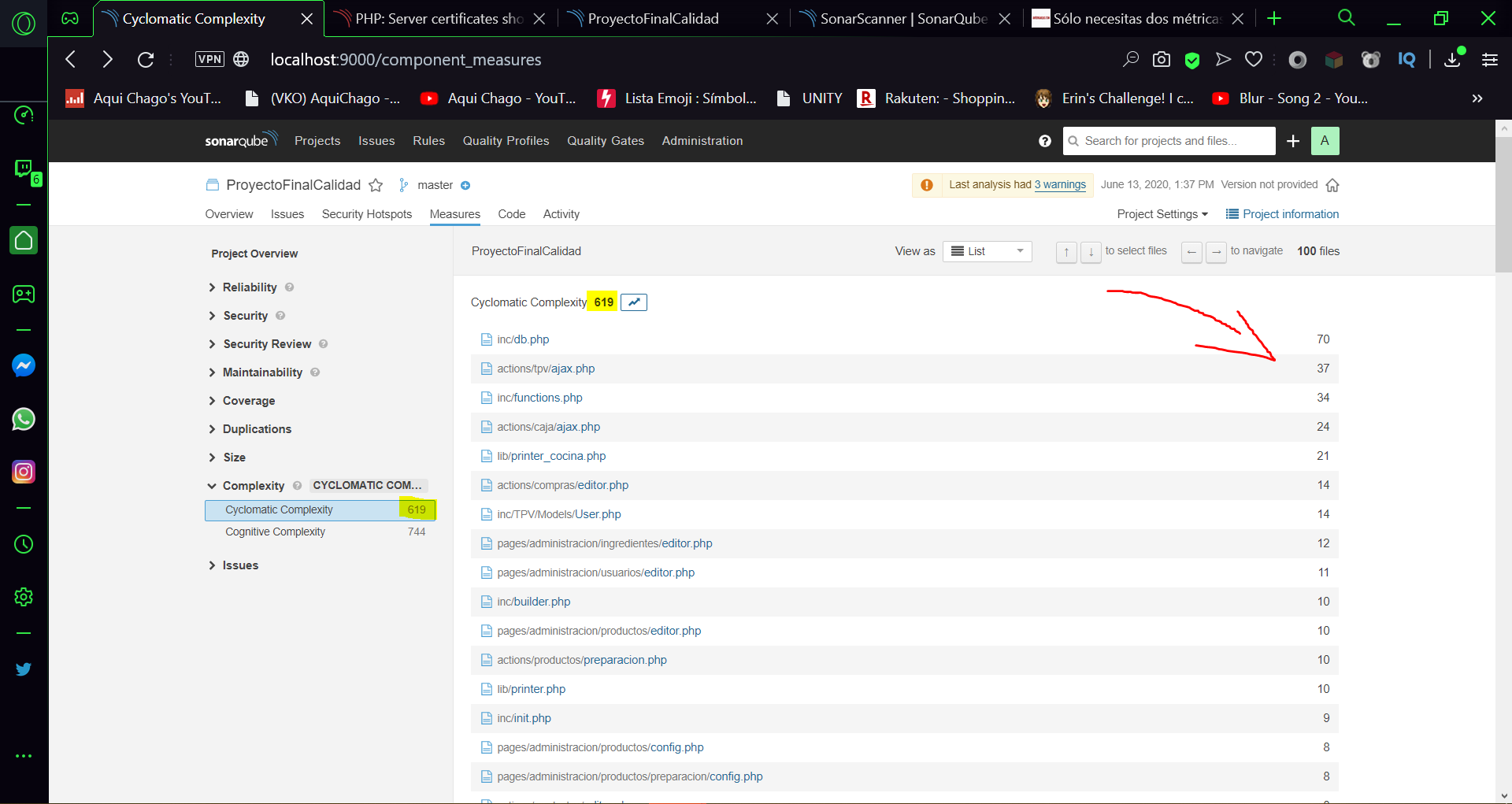
-En el software se evidencia una cantidad de líneas de código de 23191  
-En el software se evidencia una cantidad de líneas de código(Sin contar las líneas de CSS) de 3700  
-Desafortunadamente se encontró un bajo porcentaje en lo que a densidad de comentarios se refiere un 3.5% solamente (Es importante que los comentarios sean los suficientes para que otros desarrolladores tengas un pista del funcionamiento de cada módulo y así poder hacerle un mantenimiento correcto al software y a su vez no perder tiempo ni dinero en el proceso)



-En cuanto a la duplicidad del código:  


Se encontro un total de 1586 lineas repetidas, 82 bloques y 35 archivos duplicados, de los cuales aproximadamente entre 30% y 40% son eliminables

### Complejidad del software(20%)

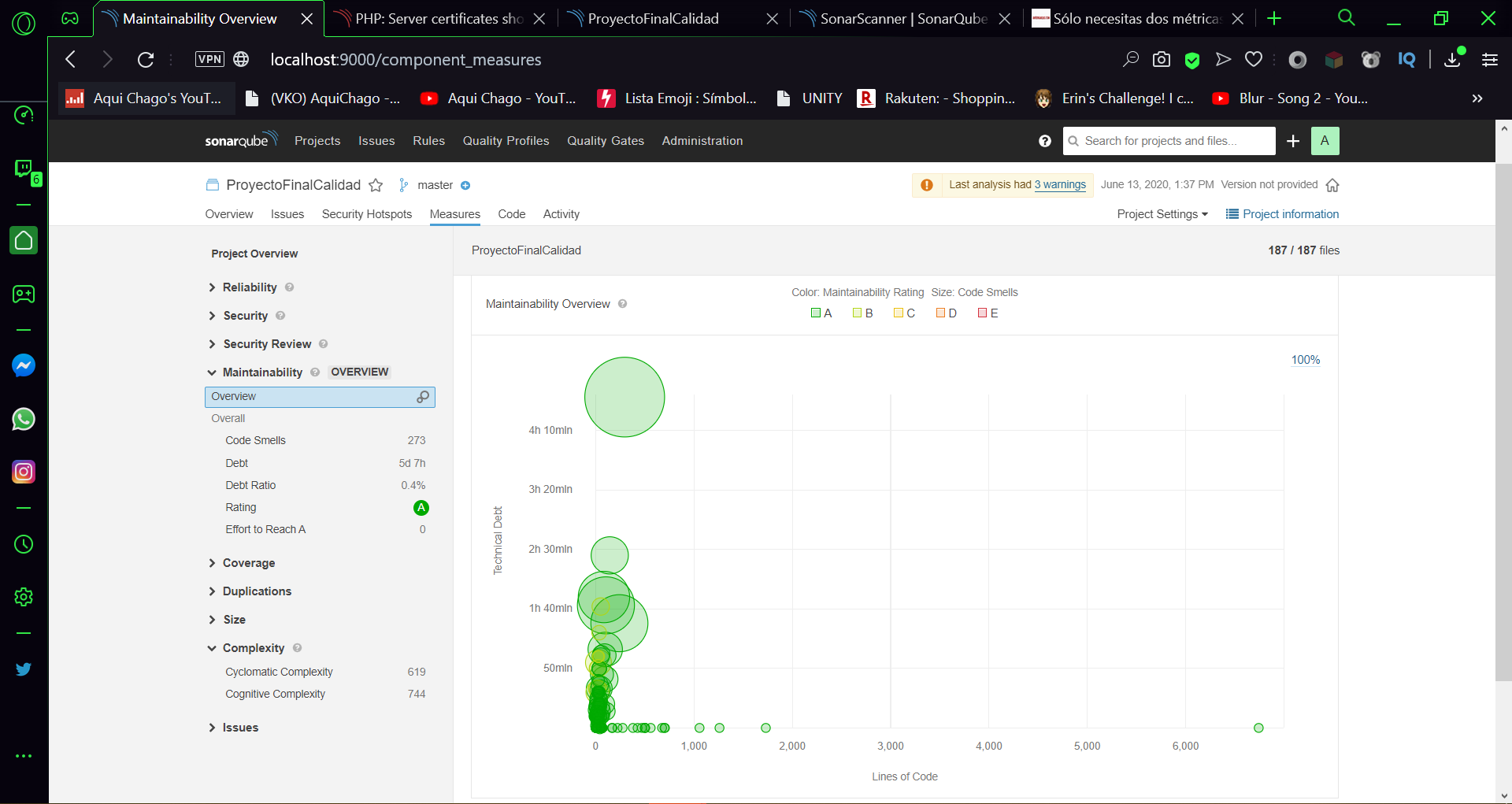


En la anterior imagen nos damos cuenta que el proyecto completo posee una calificación total en cuanto a su complejidad ciclomatica de 619, a su vez vemos una lista de los archivos que mayor complejidad poseen de mayor a menor, se recomienda a los desarrolladores realizar métodos mas simples que cumplan con las mismas funciones o utilizar una estructura diferente en los métodos comprometidos.

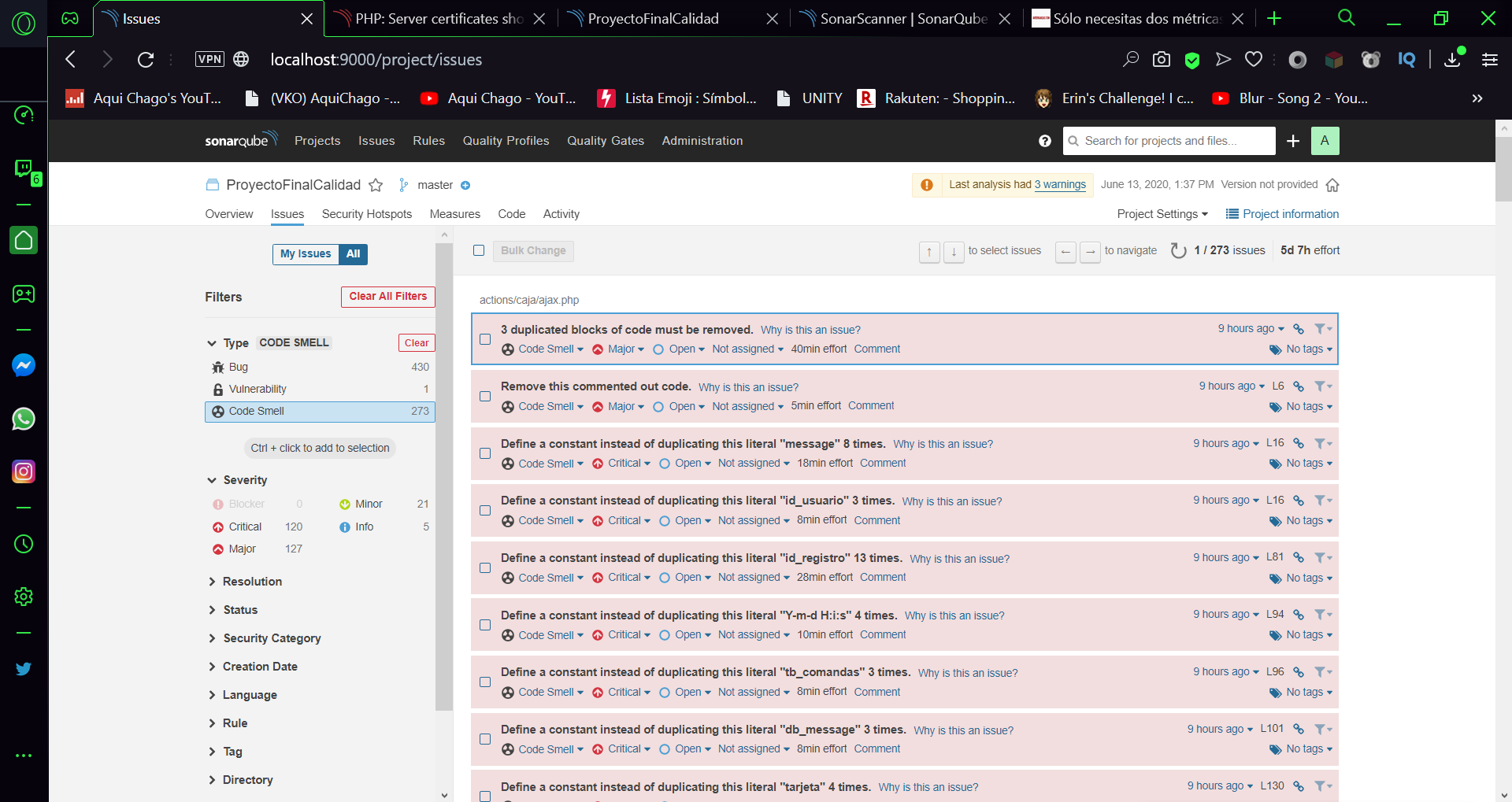


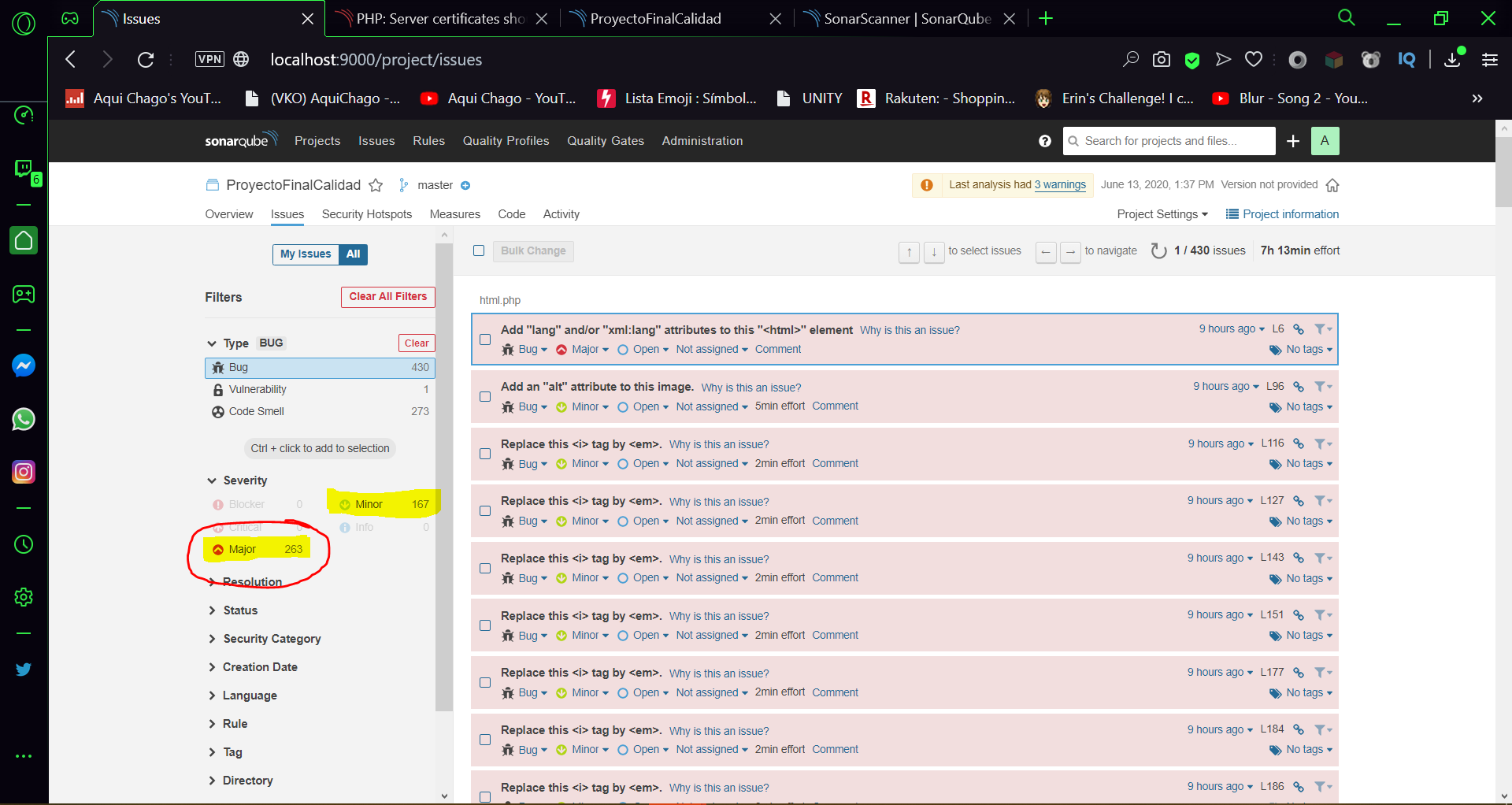
En la anterior imagen nos damos cuenta que el proyecto completo posee una calificación total en cuanto a su complejidad cognitiva de 744, a su vez vemos una lista de los archivos que mayor complejidad poseen de mayor a menor, se recomienda a los desarrolladores realizar métodos más simples que cumplan con las mismas funciones o utilizar una estructura diferente en los métodos comprometidos.

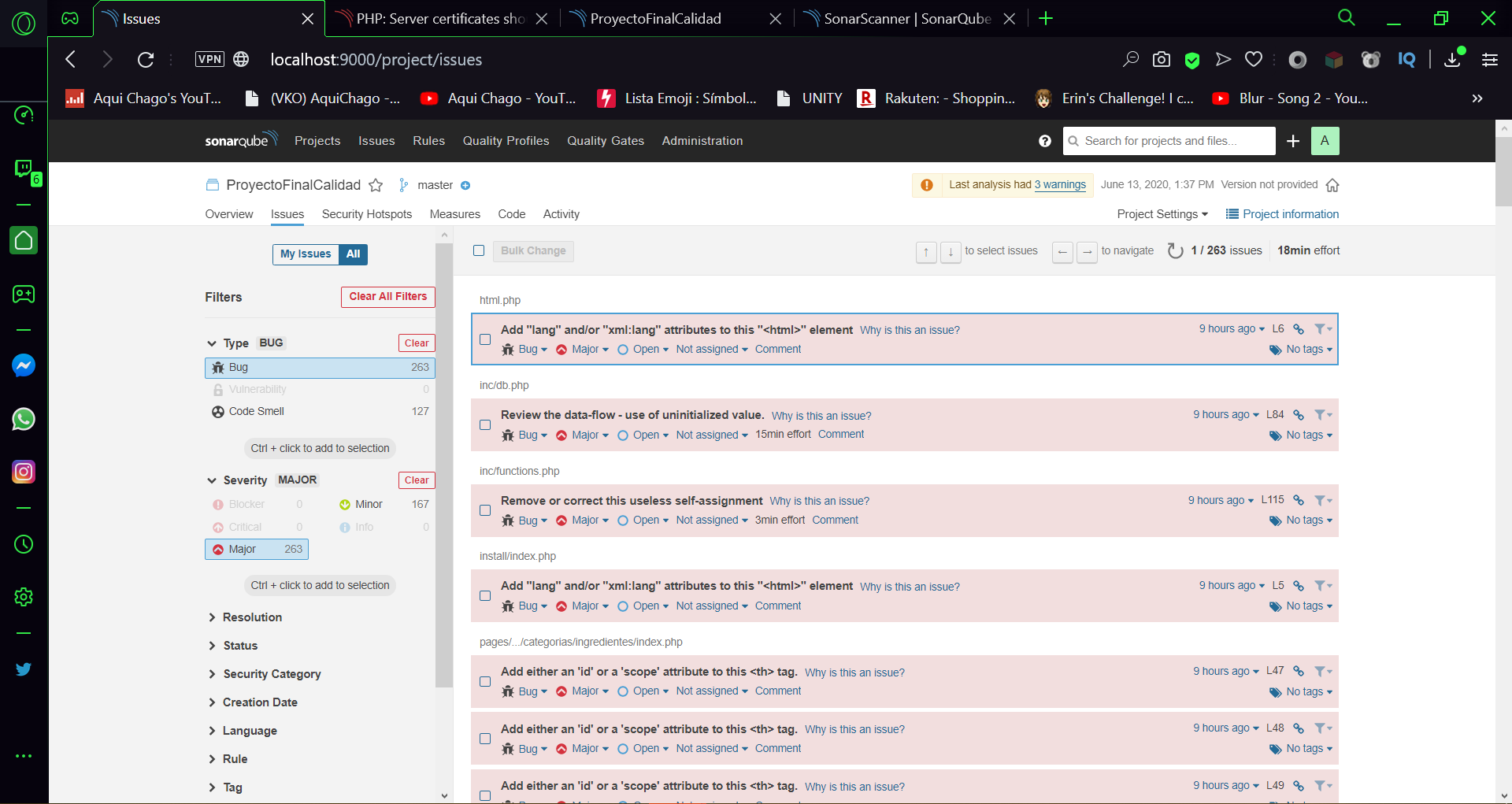
### Deuda técnica (20%)



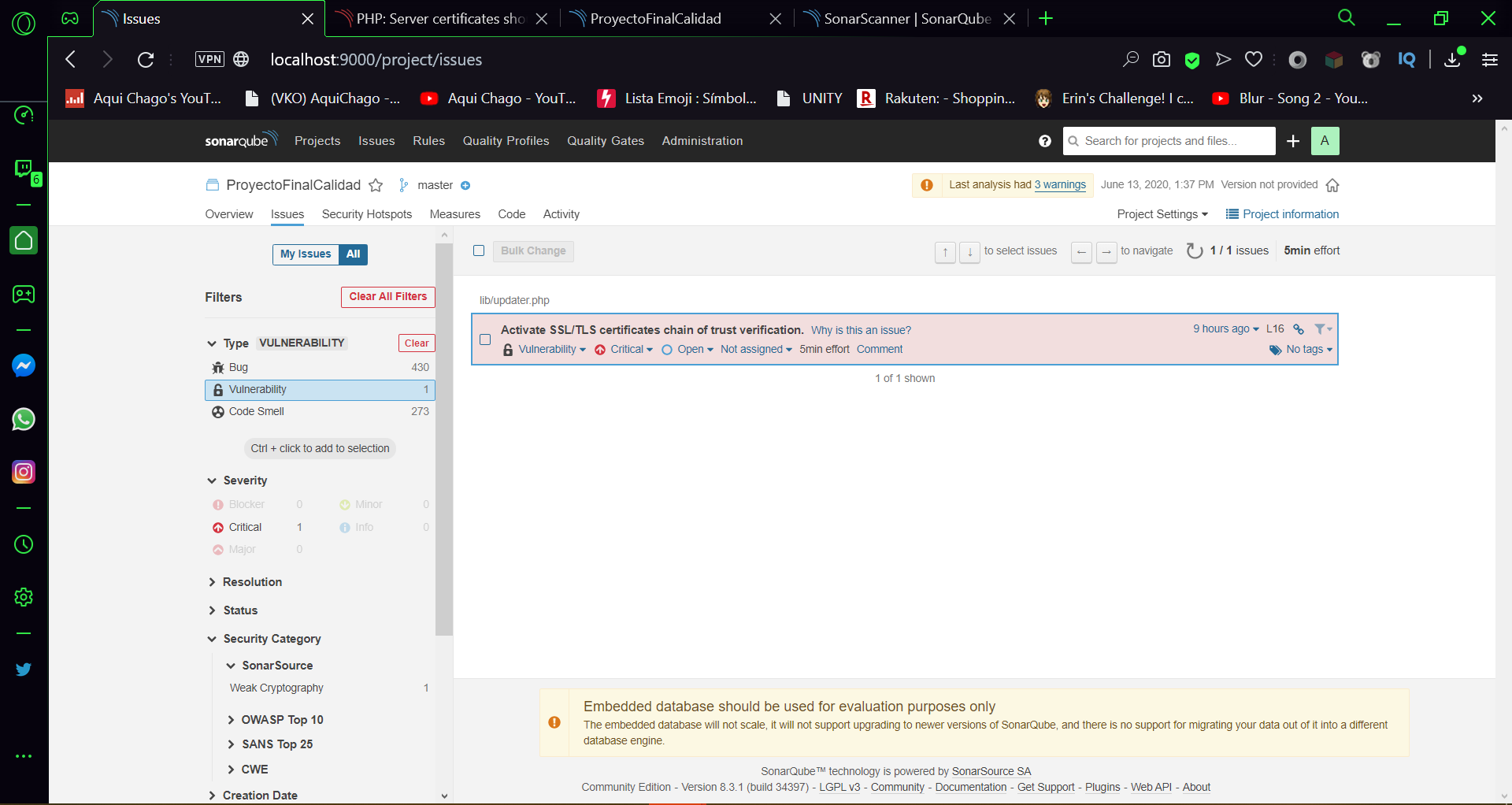
Se encontraron 273 casos de código smell , 5 dias y 7 horas de deuda técnica, para obtener un rating (A) a pesar de todo los códigos smell no afectan de manera critica en el software, razón por la cual el grado obtenido es una A.

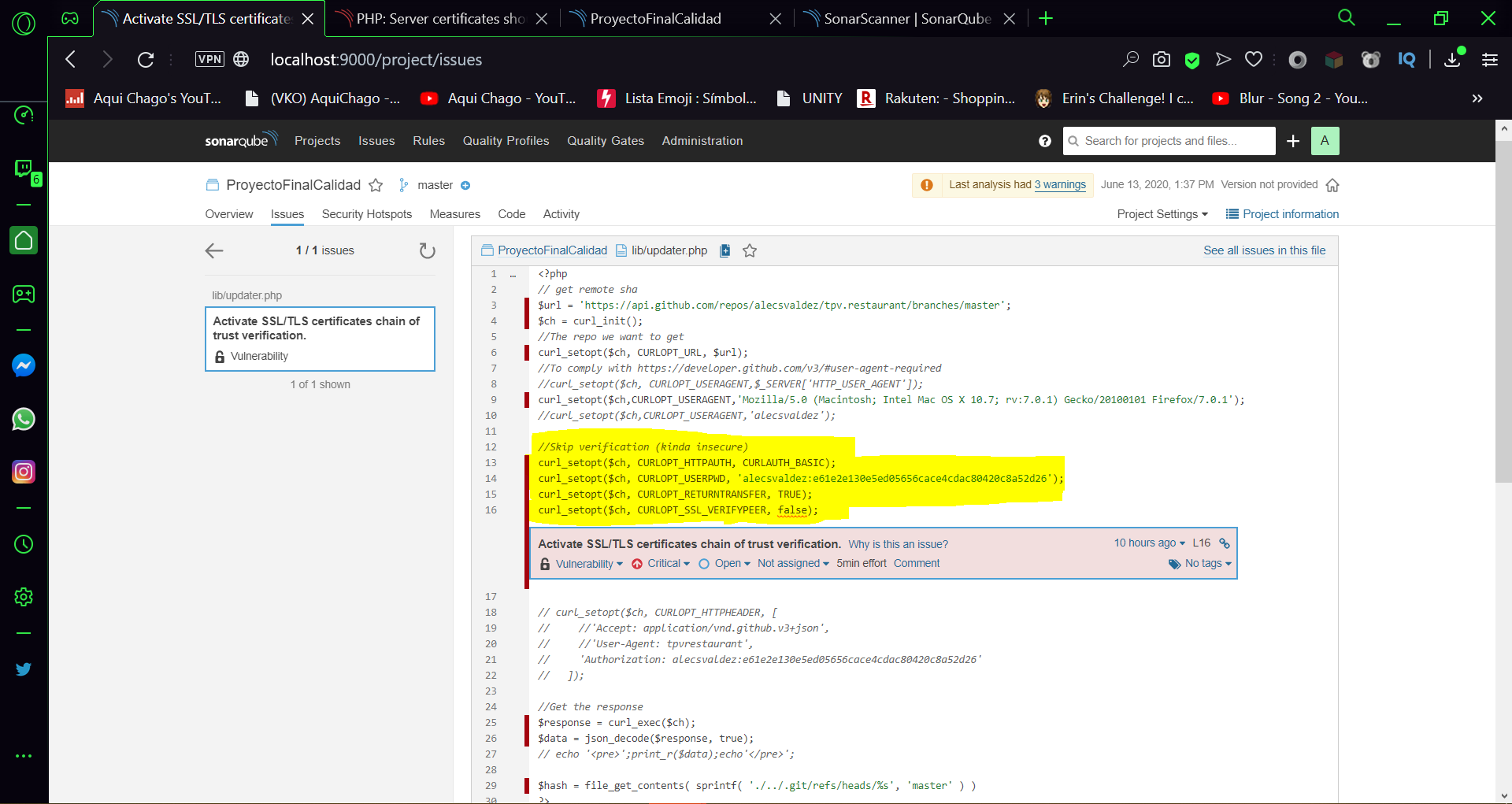


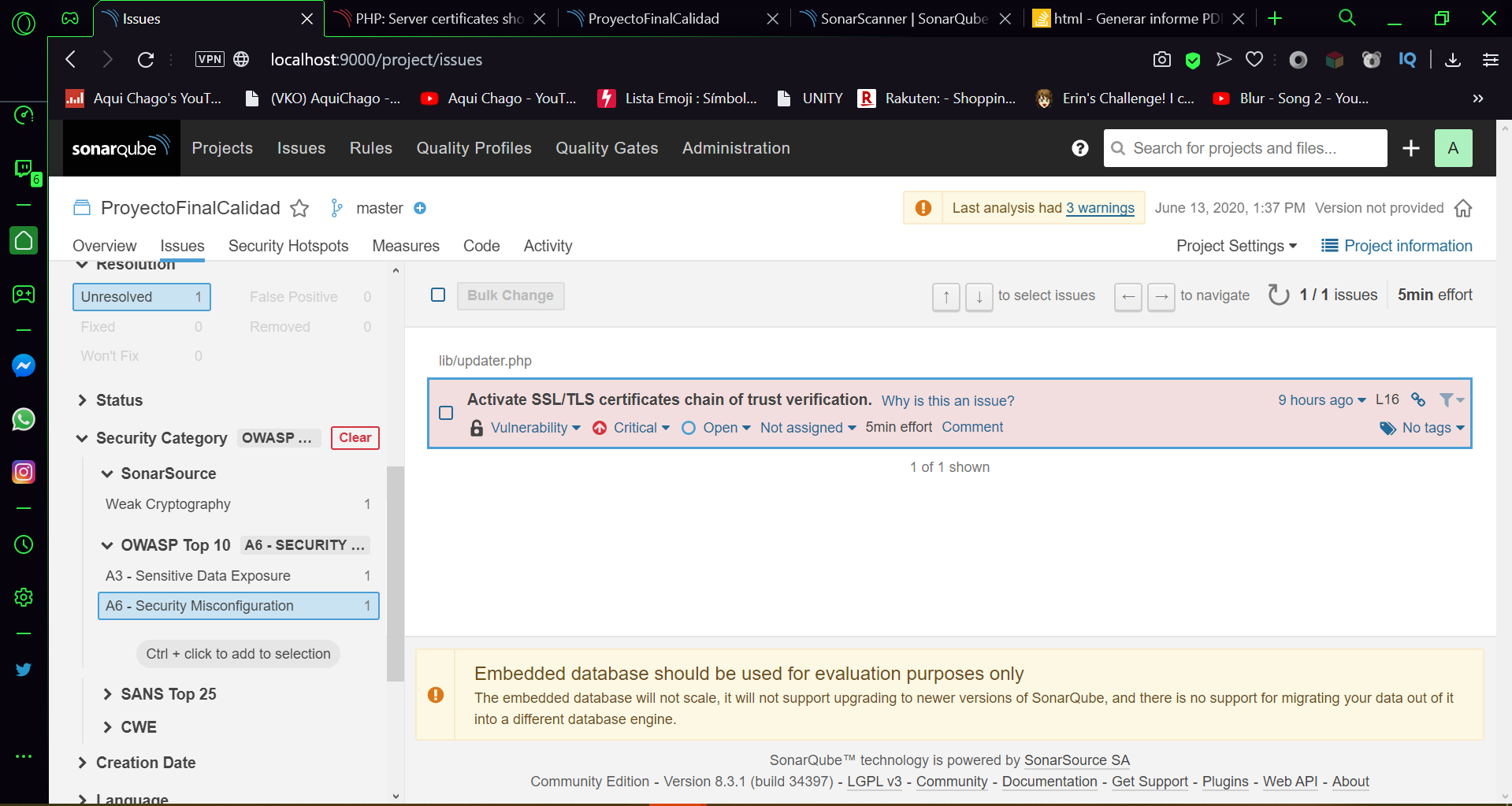




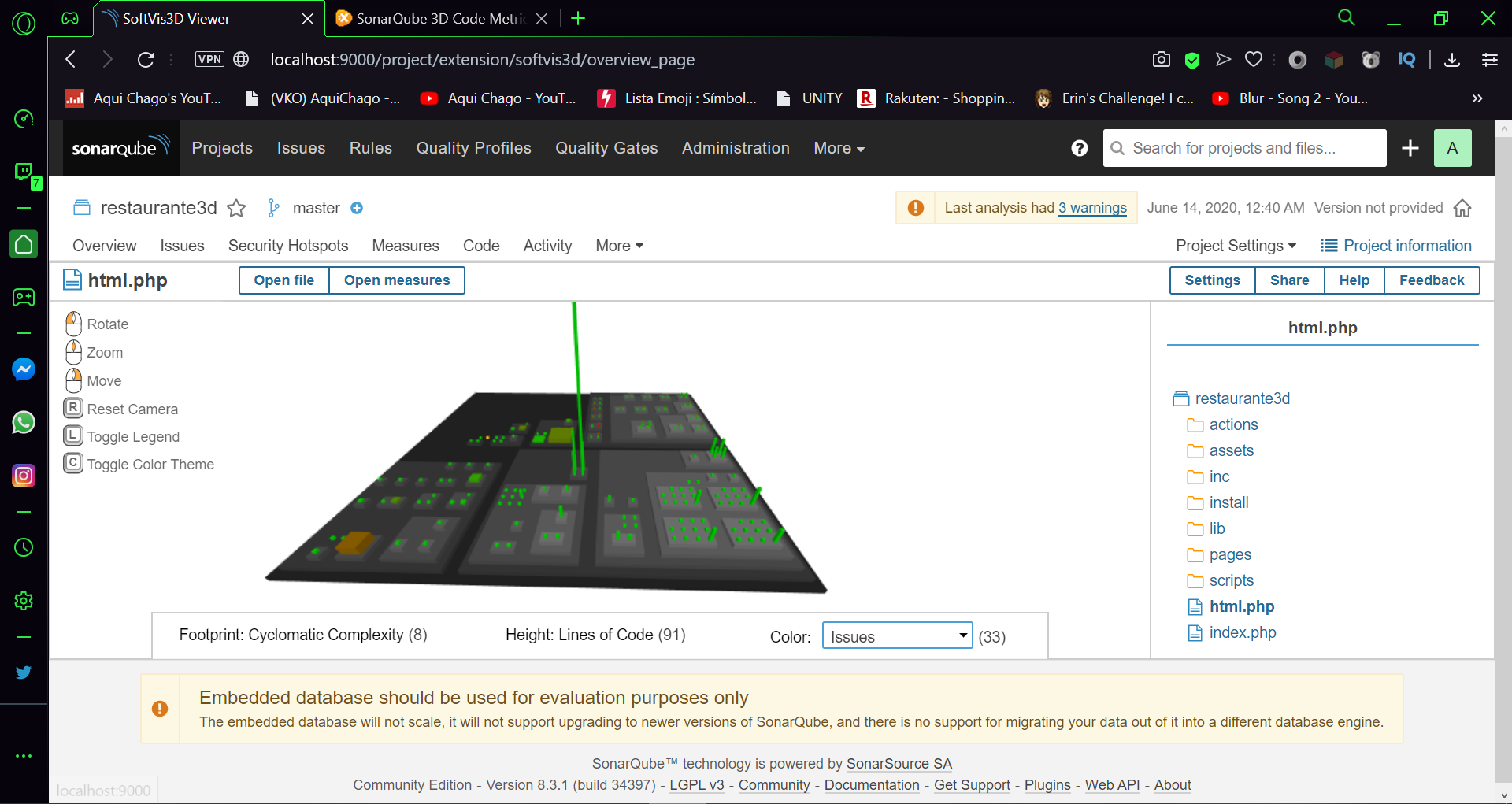
### Seguridad(20%)

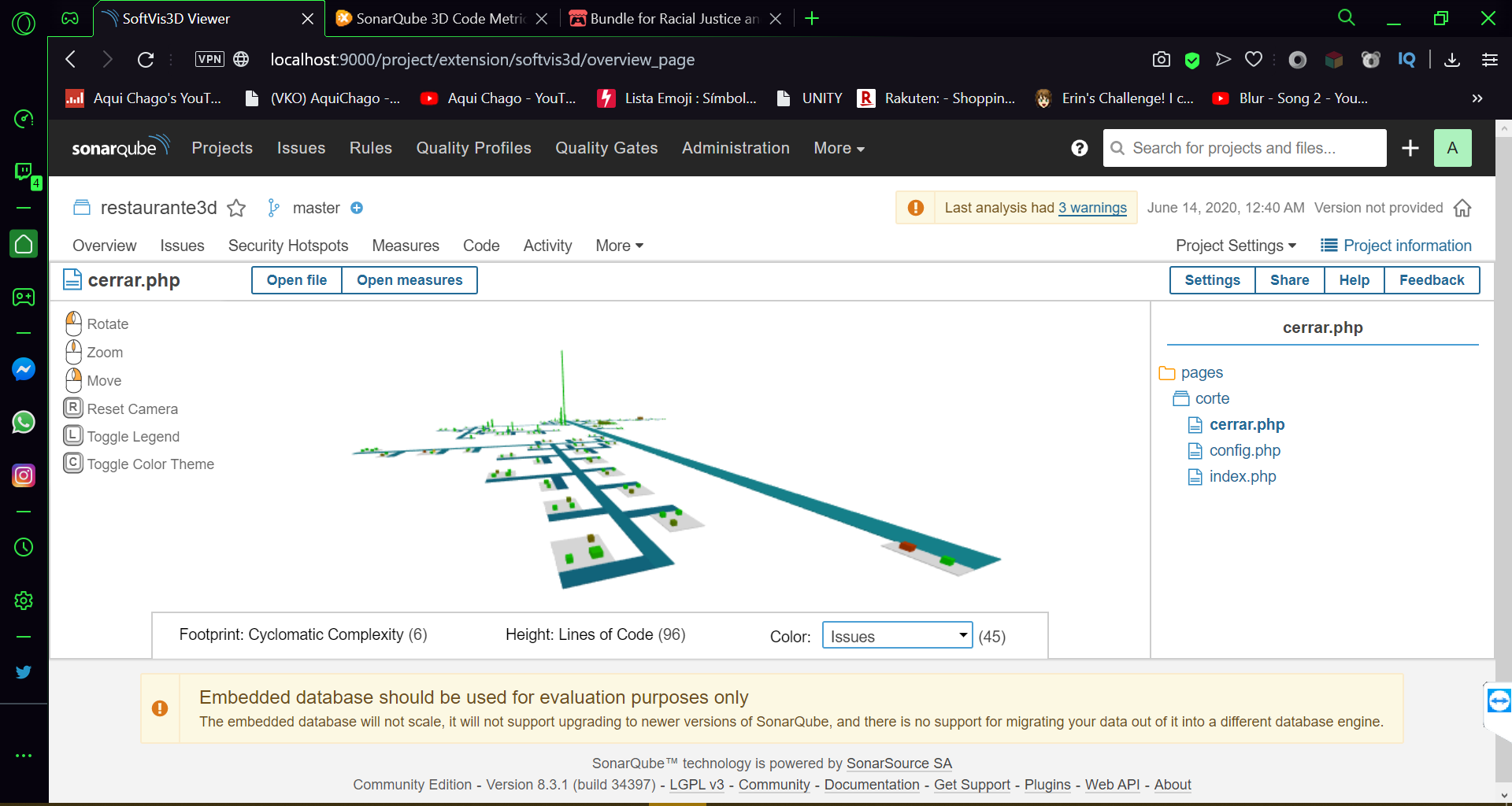
En cuanto a lo que a seguridad se refiere se encontró solo un problema, aunque el problema, o vulnerabilidad se clasifico como grave.  


Aunque analizando el código nos damos cuanta que en realidad la vulnerabilidad fue encontrada en un bloque de código donde se realiza un acceso de prueba saltándose la verificación de usuario a lo que el desarrollador avisa que es inseguro.  




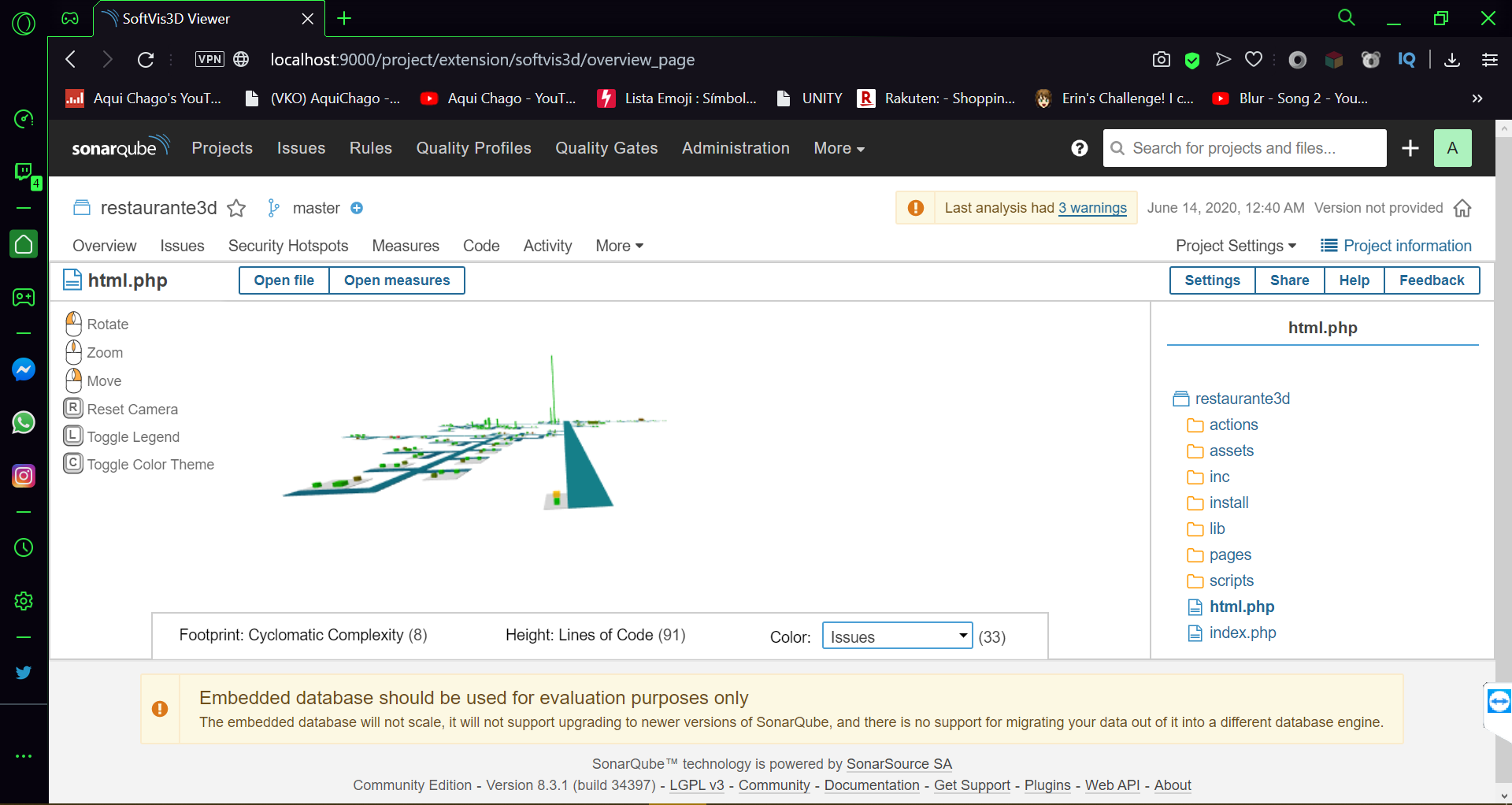
### 3d code metrics (20%)



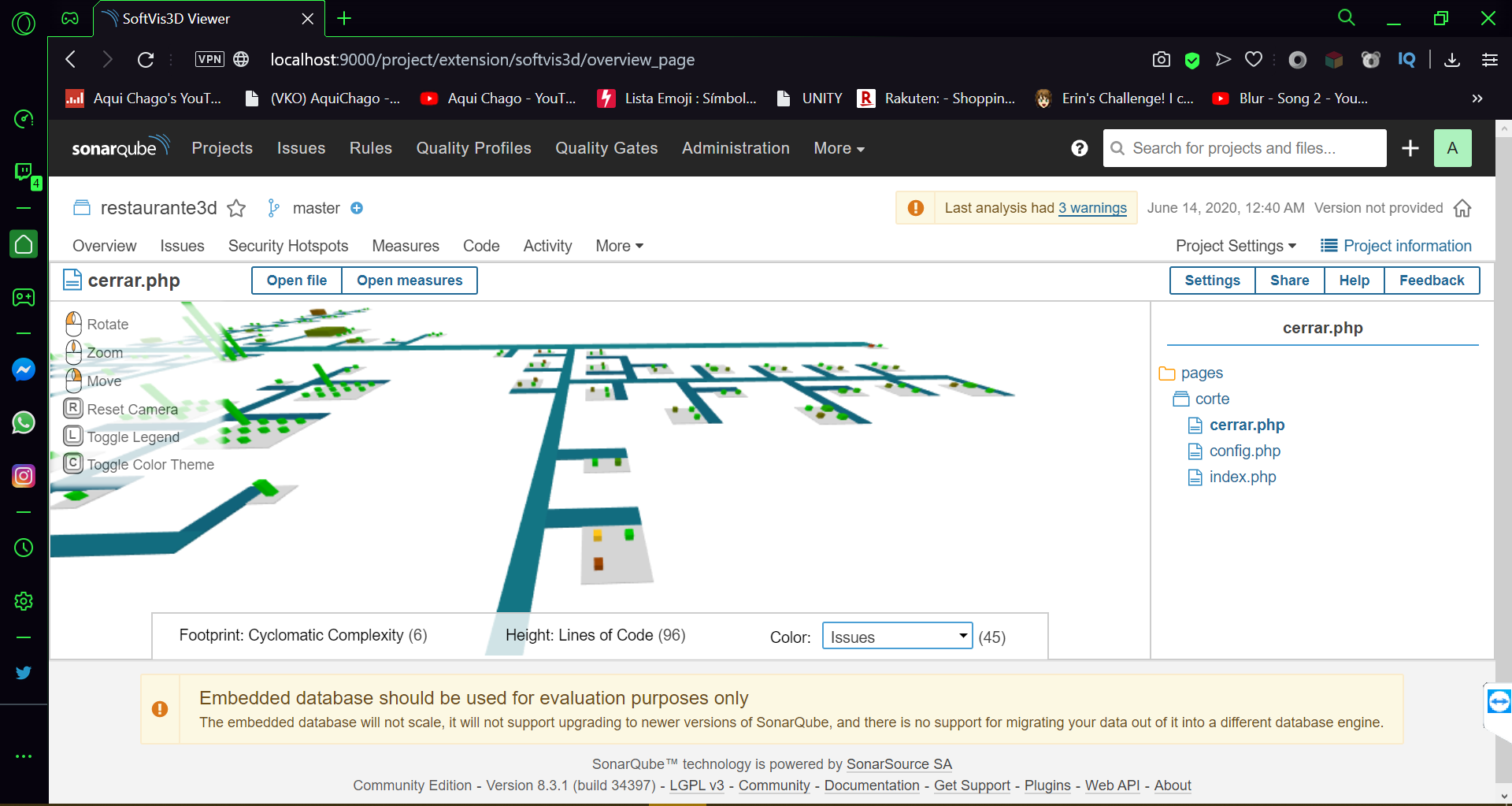


En el modelo de SoftVis3d viewer se pueden evidencias varios detalles, uno de los mas interesantes es que a nivel de complejidad de cada uno de los módulos (lo que corresponde a la altura de cada (edificio)) todos y cada uno de ellos son en relativa medida de igual altura y se encuentran de color verde lo que nos representa una buena aplicación de las practicas correctas de programación y un buen manejo de los conceptos referentes a la calidad del software, en cuanto a la imagen donde vemos algunas estructuras rojas , hacen referencia a algunos módulos donde encontramos Bugs, código Smell o la vulnerabilidad de seguridad que se trato anteriormente.

Como ejemplo hallamos los siguientes.



El HTML principal del software.

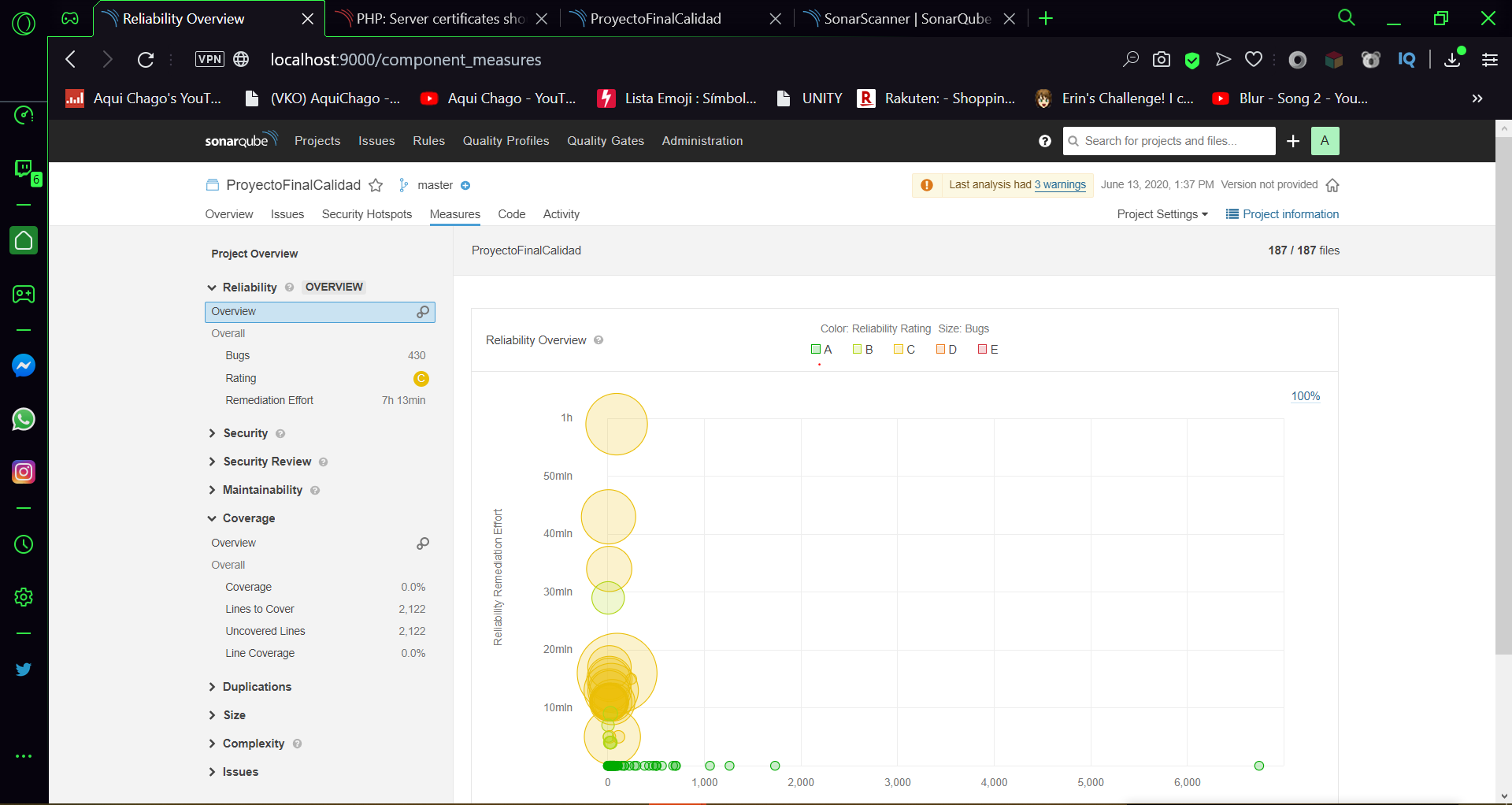


La pagina de cierre en cuanto al modulo de corte se refiere.

Y etcétera de ejemplos como patrón se encontró que en la mayoría de páginas index.php se encuentran varios inconvenientes en cuanto al código y la funcionalidad del mismo a juicio del software seleccionado para realizar esta tarea de análisis.

## Atributos Externos.

### Usabilidad (20%)



* En cuanto a la fiabilidad se obtuvo ya una calificación bastante peor Se obtuvo una (C) ya que se hallaron 430 bugs en el código, algunos leves y otros críticos que pueden llegar a afectar directamente la funcionalidad del código.
* La eficiencia, en cuanto al análisis realizado se podría tratar como eficiente cada modulo corresponde a su funcionalidad, con respecto a los nombres(son claros y concretos) y representan a cabalidad para que fueron creados.
* Satisfacción, el software no es perfecto, tiene algunos fallos y hay algunas recomendaciones a seguir para mejorarlo, pero es un software intuitivo y comprensible para el usuario que necesite utilizarlo y a nivel técnico cumple la función para la cual fue creado.

¿Se considera un software satisfactorio? A esta pregunta la respuesta es positiva

### Fiabilidad (30%)

No se evidencian un método claro para suplir fallos inesperados en el software o problemas durante el uso del mismo, se considera prudente contar con un “plan de contingencia” en caso de que se llegase a presentar algún inconveniente con el software o hardware al momento de usarlo para dar la seguridad al usuario de que no perderá su información.

### Mantenibilidad (50%)

A pesar que algunos módulos están anidados entre otros, cada uno esta bien diferenciado y se encuentra en disposición para realizarle los cambios necesarios para cumplir con normativas nuevas o funcionalidades que sea necesario añadir, en cuanto a su mantenibilidad obtiene una calificación positiva.

# Recomendaciones y conclusiones

Como anteriormente fue mencionado, este software no es perfecto y a pesar que cumple con la función para cual fue desarrollado necesita varios ajustes para adecuarse a los estándares de calidad.

Para ver las recomendaciones a seguir para mejorar la calidad del software dirigirse directamente a cada uno de los numerales anteriores.  
  
En cuanto a las conclusiones generales del proyecto de software, se considera un buen proyecto con amplio margen de proyección, pero es importante adaptarse a los lineamientos de calidad generales.

# Bibliografía

Toda la información necesaria para la realización de este análisis de calidad se obtuvo de la plataforma SonarQube y del código fuente del proyecto.

[1] (Norma ISO 25000) TOMADO DE <https://www.excentia.es/iso-25000#:~:text=La%20ISO%2025000%20proporciona%20una%20guía%20para%20el%20uso%20de,definiciones%20comunes%2C%20así%20como%20la>

# Firma del perito.

|  |  |
| --- | --- |
| **Responsable de la evaluación**  Nombre: Edgar Santiago Jimenez Ortega  Empresa: USTA Tunja  Cargo: Analista  GitHub: SantiagoJimenezOrtega | **Firma** |