



Innovación y Software

ISSN: 2708-0927

ISSN: 2708-0935

facin.innosoft@ulasalle.edu.pe

Universidad La Salle

Perú

Lima Torres, Sandra

Componente de revisión de estándar de arquitectura de datos para el gestor de bases de datos SQLite

Innovación y Software, vol. 2, núm. 1, 2021, Marzo-Agosto, pp. 20-32

Universidad La Salle

Perú

Disponible en: <https://www.redalyc.org/articulo.oa?id=673870838002>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica Redalyc

Red de Revistas Científicas de América Latina y el Caribe, España y Portugal  
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso  
abierto

Tipo de artículo: Artículo original

Temática: Tecnologías de bases de datos

Recibido: 18/10/2020 | Aceptado: 10/01/2021 | Publicado: 30/03/2021

## Componente de revisión de estándar de arquitectura de datos para el gestor de bases de datos SQLite

### *Data Architecture Standard Review Component for SQLite Database Manager*

Sandra Lima Torres <sup>1</sup>[\[0000-0003-0025-8725\]\\*](https://orcid.org/0000-0003-0025-8725)

<sup>1</sup> Departamento de Inteligencia Computacional. Facultad 1. Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km 2 1/2, Torrens, Boyeros, La Habana, Cuba. [slima@uci.cu](mailto:slima@uci.cu)

\* Autor para correspondencia: [slima@uci.cu](mailto:slima@uci.cu)

---

#### Resumen

La preocupación por la calidad de los sistemas basados en software ha aumentado a medida que estos se integran en cada aspecto de nuestras vidas. Con el incremento del número de usuarios y de los datos almacenados se hace necesario contar con bases de datos más robustas y seguras. Las pruebas de control de calidad han progresado de manera sistemática desde lo manual hasta las aplicaciones automatizadas. La Empresa de Tecnologías de la Información para la Defensa cuenta con el Centro de Calidad, Estándares y Seguridad encargado de certificar y evaluar técnicamente productos informáticos, garantizando la calidad en los mismos. Actualmente el proceso de revisión de las bases de datos en el gestor SQLite es de forma manual, dificultando dicho proceso, donde puede existir un margen de error debido a la cantidad de comprobaciones que se deben revisar. El presente trabajo tiene como objetivo el desarrollo de un componente de revisión de arquitectura de datos para el servidor de gestión SQLite. Para ello se realiza un estudio de sistemas gestores de bases de datos homólogos, las herramientas de pruebas a las bases de datos existentes y cómo está conformado un estándar de arquitectura de bases de datos. Se realizan pruebas funcionales al componente para verificar su correcto funcionamiento. Finalmente se obtiene un componente integrado al marco de trabajo Zeolides, capaz de comprobar automáticamente la arquitectura de las bases de datos del gestor SQLite en las aplicaciones informáticas desarrolladas en la empresa.

**Palabras clave:** bases de datos, calidad de software, arquitectura de datos, revisión, SQLite.

#### Abstract

*Concern for the quality of software-based systems has grown as software is integrated into every aspect of our lives. With the increase in the number of users and the data stored, more robust and secure databases are necessary. Quality control testing has progressed systematically from manual to automated applications. The Information Technology for Defense Company has the Center for Quality Standards and Security in charge of certifying and technically evaluating*

*computer products, guaranteeing their quality. Currently the database review process in the SQLite manager is manual, making this process difficult, where there may be a margin of error due to the amount of checks that must be reviewed. The present work aims to develop a data architecture review component for the SQLite management server. For this, a study of a homologous database management system is carried out, the tools for testing existing databases and how a database architecture standard is formed. Functional tests are carried out on the component to verify its correct operation. Finally, a component integrated o the Zeolides framework is obtained, capable of automatically checking the architecture of the databases of the SQLite manager in the computer applications developed in the company.*

**Keywords:** *databases, software quality, data architecture, revision, SQLite.*

---

## Introducción

La aplicación de los medios informáticos ha revolucionado la gestión de las empresas cubanas. El crecimiento y difícil manejo de grandes volúmenes de información explica las razones que progresivamente obligan a las organizaciones a desarrollar las tecnologías de Bases de Datos (BD). El valor de una información actualizada ha crecido tanto que para incrementar o mantener la productividad se deben gestionar eficientemente todos los datos. Son los Sistemas Gestores de Bases de Datos (SGBD) los que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada [1].

A lo largo del ciclo de vida del desarrollo de software se toman medidas para asegurar la calidad de los sistemas de bases de datos, uno de los instrumentos que facilita esta tarea es la adopción de estándares de diseño de BD. El uso de los mismos tiene innumerables ventajas, entre ellas está asegurar la legibilidad del modelo de datos, incluso en etapas de análisis y diseño, así como facilitar la tarea de los programadores en el desarrollo de los sistemas [2].

En Cuba existen empresas que se encargan de realizar pruebas de control de calidad. Estas han progresado de manera sistemática desde lo manual (la forma más básica) hasta las aplicaciones automatizadas para los diferentes sistemas informáticos donde se encuentran las BD. La Empresa de Tecnologías de la Información para la Defensa (XETID), posee un Centro de Calidad, Estándares y Seguridad (CCES) encargado de certificar y evaluar técnicamente productos informáticos de producción nacional o importada. Evalúa y certifica procesos de desarrollo y organizaciones para la industria de software en las Fuerzas Armadas Revolucionaria (FAR), según normas nacionales e internacionales [3]. Por la importancia que tiene la calidad para esta empresa, el CCES utiliza estándares y herramientas automatizadas para evaluar sus productos de software. Una de las pruebas de calidad que se realizan en CCES es a las BD de las aplicaciones, donde se evalúa que el estándar de arquitectura de datos que utiliza la XETID sea empleado correctamente.

Actualmente existe una herramienta automatizada que realiza este tipo de pruebas a las BD en el gestor PostgreSQL, llamada Componente de revisión de Estándar y Arquitectura de Datos (CEAD) [4].

El auge del sistema operativo Android ha causado gran impacto en el desarrollo de aplicaciones móviles a nivel nacional e internacional. Sobre todo, por la facilidad y comodidad de usar las aplicaciones desde cualquier dispositivo móvil en el momento y lugar que se necesite. Logrando una amplia aceptación en el mercado [5].

La XETID se encarga también de la realización de aplicaciones Android, tales como EnZona o Participación popular. El CCES realiza la revisión y certificación de estos productos, los cuales cuentan con BD realizadas con el gestor SQLite, ya que se ejecuta en muchas plataformas y sus BD pueden ser fácilmente portadas sin configuración o administración.

La empresa no cuenta con un sistema que compruebe que la arquitectura de datos en el gestor SQLite esté acorde a un estándar de nomenclatura dado, por lo que resulta muy engorroso el proceso de revisión de estas BD. Actualmente, en el CCES, uno de los factores que afecta es el tiempo que se dedica a las pruebas de calidad que se realizan a la arquitectura de las BD de SQLite, pues son de forma manual, donde puede existir un margen de error debido a la cantidad de comprobaciones que se deben realizar.

El presente trabajo tiene como objetivo general: Desarrollar una herramienta que contribuya a la comprobación automática de la arquitectura de las BD gestionadas con SQLite.

En la siguiente sección se exponen las bases de la investigación, se describe el diseño de la misma y un trabajo relacionado con el tema. En la sección de resultados y discusión se analizan los resultados de validar y aplicar la herramienta propuesta.

## **Materiales y métodos**

Para dar solución al problema identificado se hace necesario establecer un grupo de conceptos referentes a la estandarización de los SGBD, especialmente los desarrollados en SQLite.

### **Sistema Gestor de Bases de Datos**

El SGBD es un conjunto de programas, lenguajes y procedimientos, que permite almacenar, acceder y recuperar los datos en la BD, proporcionándole al usuario herramientas para describir y manipular dichos datos de manera eficiente y práctica. Un SGBD actúa como un intermediario entre la BD y el usuario, ya que opera como interfaz entre estas dos entidades [6].

Según la apreciación de Ramos [7] el objetivo primordial de un gestor es proporcionar eficiencia y seguridad a la hora de extraer o almacenar información en las BD. Los SGBD son aplicaciones de software diseñadas para facilitar tareas. A medida que crecen el volumen de los datos y el número de usuarios (actualmente son habituales los centenares de gigabytes y los millares de usuarios) el apoyo de los SGBD se vuelve indispensable. En la actualidad existen muchos SGBD, los más utilizados son PostgreSQL, MySQL, MongoDB, Oracle, y SQLite [8].

### **El sistema gestor de bases de datos SQLite**

González [9] define que “SQLite es un SGBD relacional, famoso por su pequeño tamaño. A diferencia de otros sistemas cliente-servidor el motor de SQLite no es un proceso independiente, lo que hace que la latencia sea menor y el acceso más eficiente. Debido a su facilidad de uso, su pequeño tamaño y su versatilidad SQLite es utilizado en una gran variedad de aplicaciones. Su uso ha sido muy popular en las aplicaciones para *Smartphone* con sistema operativo Android o iOS” [4]. Teniendo en cuenta la bibliografía consultada se puede afirmar que este gestor es mucho más rápido y ligero que MySQL y PostgreSQL. Se ejecuta en muchas plataformas, es de dominio público y por tanto sin costo. Por lo cual, según López [10] las empresas de desarrollo de software en nuestro país tienen un aumento considerable de la producción de aplicaciones para móviles con sistema operativo Android que utilizan este SGBD, como por ejemplo EnZona, Participación Popular, ONAT, Transfer Móvil.

### **Estándar de arquitectura de BD**

Una BD debe ser lo bastante sencilla como para que los interesados puedan comprenderla, además de coherente y estable, por lo que hay que cuidar su diseño y puesta en marcha [11]. Por tanto, se hace necesaria la creación o adopción por parte de la entidad de un estándar de arquitectura de datos [12]. Sánchez [13] hace referencia a que desde la aparición de los primeros SGBD se intentó llegar a un acuerdo para que hubiera una estructura común para todos ellos. Los intentos por conseguir una estandarización han estado promovidos por organismos de todo tipo. Algunos son estatales, otros privados y otros promovidos por los propios usuarios. Entre estos organismos se encuentra la *International Organization for Standardization* (ISO), la cual se encarga de la definición de estándares de gran prestigio internacional. La arquitectura de datos se diseña y se desarrolla durante la etapa de planificación de un nuevo sistema para establecer la manera en que se procesarán, almacenarán y utilizarán los datos, y cómo se podrá acceder a ellos. En ella se integran los modelos, políticas y reglas que rigen qué datos se van a recopilar; cómo van a ser almacenados, clasificados y explotados mediante la infraestructura tecnológica disponible [12]. Los estándares señalan claramente el

comportamiento esperado y deseado en las aplicaciones y son utilizados como guías para evaluar su funcionamiento y lograr el mejoramiento continuo de los servicios [14].

## **Estándares de arquitectura de BD a aplicar en las pruebas de calidad de los datos**

### **ISO/IEC 25012**

Uno de los estándares más utilizados en el mundo en relación al proceso de desarrollo de BD es la norma ISO/IEC 25012, de ella es preciso conocer que surge de la ISO/IEC 25000 conocida como SQuaRE (*System and Software Quality Requirements and Evaluation*). Es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software, se encuentra compuesta por cinco divisiones [15]:

- ISO/IEC 2500n – División de Gestión de Calidad
- ISO/IEC 2501n – División de Modelo de Calidad
- ISO/IEC 2502n – División de Medición de Calidad
- ISO/IEC 2503n – División de Requisitos de Calidad
- ISO/IEC 2504n – División de Evaluación de Calidad

Dentro de estas cinco divisiones se encuentra la ISO/IEC 2501n – División de Modelo de Calidad. Las normas de este apartado presentan modelos de calidad detallados incluyendo características para calidad interna, externa y en uso del producto software. Actualmente esta división se encuentra formada por:

ISO/IEC 25010 - Sistemas y modelos de calidad de software.

ISO/IEC 25012 - Modelo de calidad de datos: el cual define un modelo general para la calidad de los datos, aplicable a aquellos datos que se encuentran almacenados de manera estructurada y forman parte de un Sistema de Información.

### **Estándar de arquitectura de datos para la XETID**

El estándar a utilizar en la revisión automática de la arquitectura de BD en el gestor SQLite es Estándar de arquitectura de datos para la XETID en su versión 1.0. Este estándar realizado por el CCES de la empresa XETID en el año 2017, constituye una guía para el desarrollo de las BD. Desde el punto de vista arquitectónico tiene el propósito de brindar una visión de la estrategia de desarrollo del Centro. El propósito de este estándar consiste en establecer los fundamentos para la puesta en práctica del diseño e implementación de las BD.

Entre las pautas que contiene dicho estándar de arquitectura de BD están:

- Tipo de BD
- Base tecnológica para la arquitectura de datos

- SGBD
- Herramienta de diseño
- Herramientas de administración
- Seguridad de datos
- Estándar de nomenclatura
- Tipos de datos
- Políticas de indexado
- Integridad
- Normalización
- Rendimiento
- Preparación de las BD para la réplica

Estos estándares son revisados en su mayoría de forma automática. No abundan a nivel mundial sistemas elaborados que en la búsqueda de alcanzar la perfección en las realizaciones de BD realizan este tipo de pruebas basadas en los anteriores estándares de calidad. Teniendo en cuenta la necesidad de una aplicación que contribuya a la comprobación automática de la arquitectura de las BD creadas en el gestor SQLite, se realiza a continuación una propuesta de una herramienta informática que podría utilizar el CCES.

### **CEAD PostgreSQL**

Sus autores [4] aseguran que la creación del CEAD viabiliza comprobar una correcta arquitectura de datos automáticamente, además de que parte de un estándar conocido, el “Estándar de arquitectura de datos para la XETID” antes descrito. Este componente, para los probadores de la calidad del Centro, representa una poderosa herramienta que permite optimizar el tiempo de trabajo empleado en la revisión de la arquitectura de las BD. Además, realiza una búsqueda exhaustiva de no conformidades proporcionando un pequeño margen de error.

La herramienta no es adaptable a cualquier estándar, actualmente está desarrollada para la revisión de las BD en PostgreSQL. Por esta razón no es posible su utilización en la verificación de los productos con el gestor SQLite. Si bien es cierto que el código de CEAD se puede modificar incluyendo extensas líneas de código, esta modificación trae consigo una sobrecarga de información a manejar por el probador en una misma interfaz. Por tanto, afecta el tiempo de familiarización de los trabajadores con la nueva interfaz. Para evitar estos problemas el cliente desea dos componentes separados (CEAD PostgreSQL y CEAD SQLite), ambos componentes iguales en apariencia, pero cada uno con sus funcionalidades específicas. Se desea que el probador utilice el componente que necesite según el tipo de BD que esté revisando en ese momento [4].

De lo anterior se resume que la adherencia de una BD a un estándar de arquitectura de datos en particular es un área en la que se han presentado relativamente pocos trabajos. De cualquier forma, automatizar las pruebas de adherencia a un estándar es una ventaja para el desarrollo y la certificación de calidad de un producto de software. Al mismo tiempo se evidencia que el sistema analizado no fue creado para realizar comparaciones entre un estándar y la arquitectura de una BD para el gestor SQLite. Por tanto, se procede a la creación de una herramienta para lograr automatizar la revisión de la arquitectura de datos en el gestor SQLite. Sin embargo, el análisis realizado posibilita la identificación de funcionalidades y tecnologías que contribuye al desarrollo de la propuesta de solución que se propone a continuación.

## **Resultados y discusión**

### **Componente de revisión de arquitectura de datos para SQLite**

La solución se basa en una herramienta de revisión de estándar y arquitectura de datos para SQLite, la cual visualmente estará estructurada en dos áreas de trabajo fundamentales (ver Figura 1). La primera permitirá adicionar, modificar o eliminar una prueba determinada y listar todas las adicionadas anteriormente. Además, brindará la opción de darle seguimiento a una prueba determinada, así como ver los detalles de esta en específico y exportar el listado de pruebas como informe. El sistema también establecerá un lenguaje sencillo y común, que permitirá al usuario la selección de los elementos que desea revisar de la arquitectura (entre indexado, tipo de datos y nomenclatura). Para ello es necesario que el componente no permita que se realice la revisión hasta que no exista una conexión a una BD. También facilitará la obtención de incidencias y detalles de una prueba determinada. Una vez realizada la revisión, se podrá utilizar la siguiente área. La segunda área de trabajo brindará la posibilidad de visualizar los errores encontrados una vez que se haya realizado el reconocimiento de errores a la BD, mostrándose detalles de la misma. Esta permitirá generar las no conformidades encontradas automáticamente, y ser exportadas en formato Word o PDF.



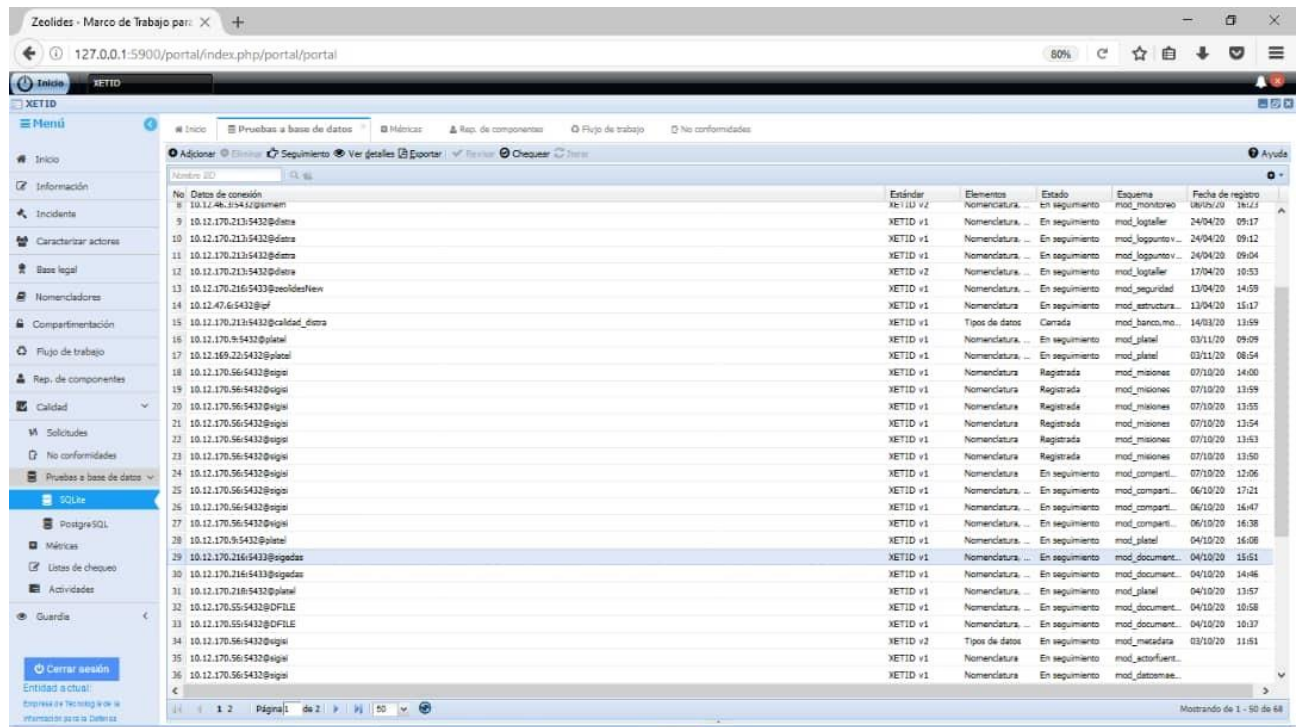


Figura 1. Interfaz de usuario seleccionando la BD a probar.

Una vez seleccionada una BD, como en la imagen anterior, se presiona el botón Revisar, se muestra una ventana emergente solicitando al usuario esperar unos minutos y luego se muestra una nueva ventana en la parte inferior que lista los errores encontrados (ver Figura 2).

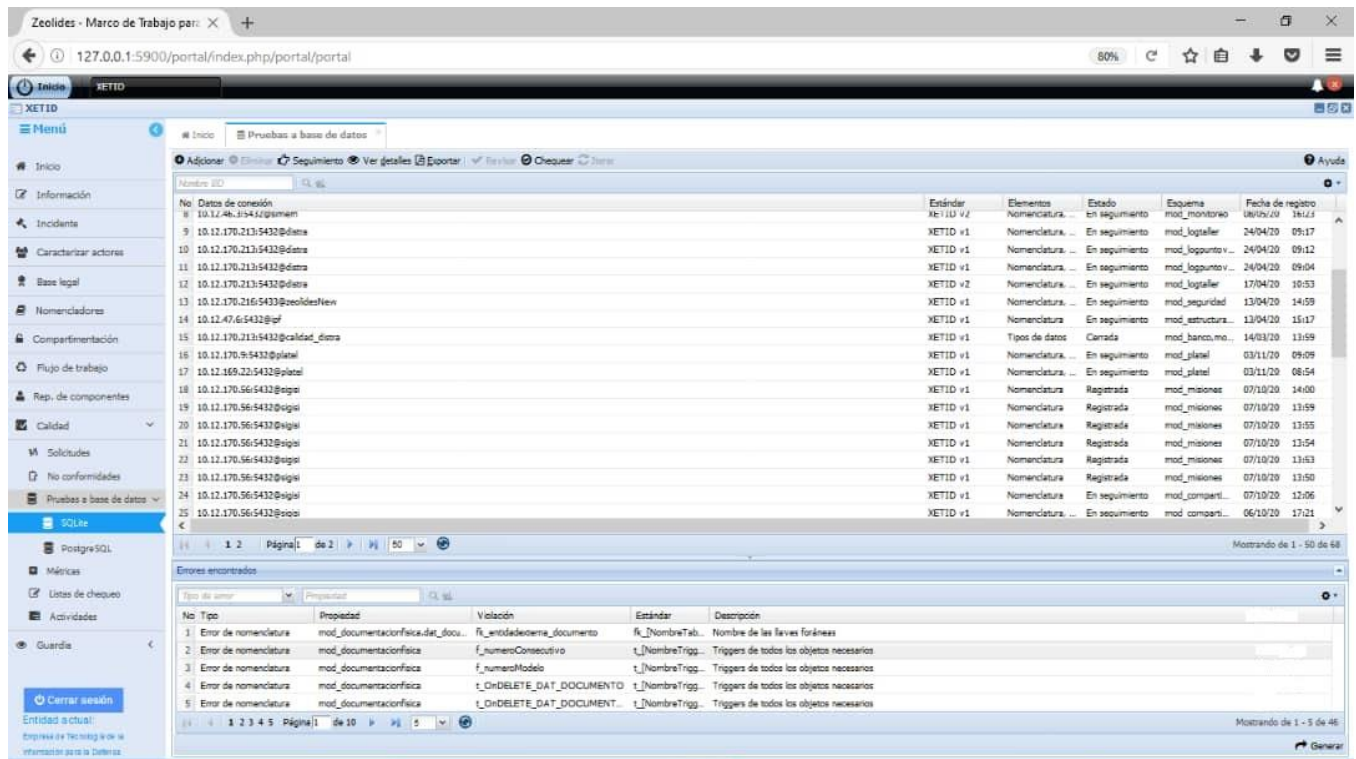


Figura 2. Interfaz de usuario listando errores encontrados.

## Pruebas realizadas al sistema

### Prueba unitaria

Se realizó la prueba unitaria haciendo uso de la herramienta RIPS; que permite realizar esta automatización del proceso de identificación de potenciales funciones vulnerables en el código fuente de aplicaciones PHP, mediante su análisis estático. En dicha prueba, como se puede observar en la Figura 3, no se encontraron vulnerabilidades.

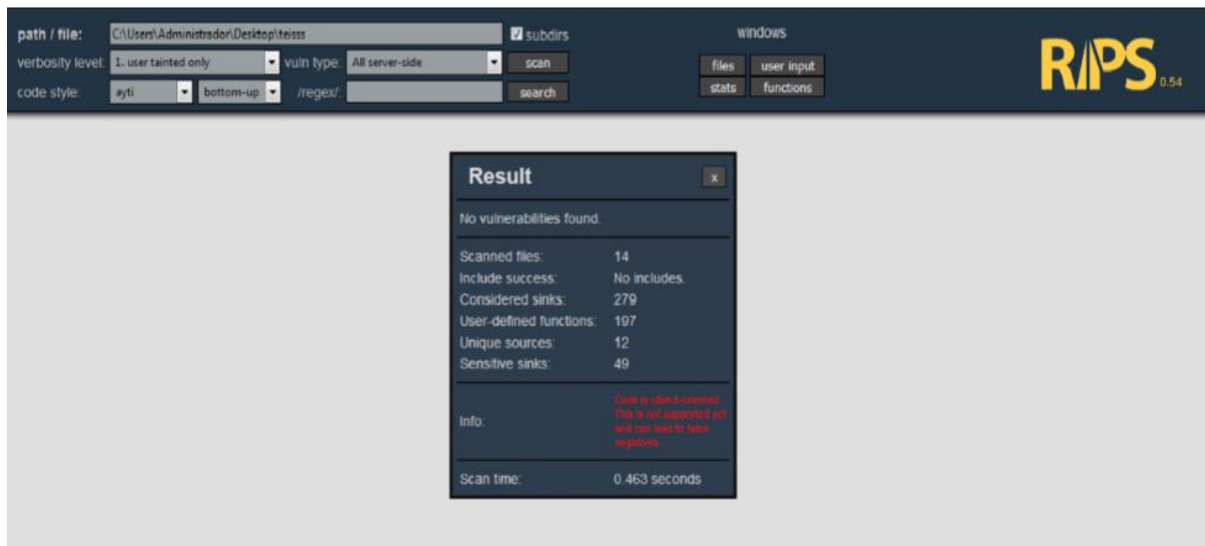


Figura 3. Resultado de la prueba unitaria.

## Prueba de seguridad

Los usuarios de negocios quieren la seguridad como un aspecto necesario de la calidad general. Una parte crítica de la creación de este compromiso es comunicarse con ellos acerca de los riesgos y beneficios, y establecer objetivos mutuamente acordados. Las actividades que se pueden realizar para hacer las pruebas de seguridad son diversas y se orientan a varios ámbitos, especialmente en lo relativo a asegurar el funcionamiento y disponibilidad de los servicios web y contenidos publicados. Al realizar la prueba de seguridad a través de la herramienta RIPS no se detectaron violaciones de seguridad (ver Figura 4).

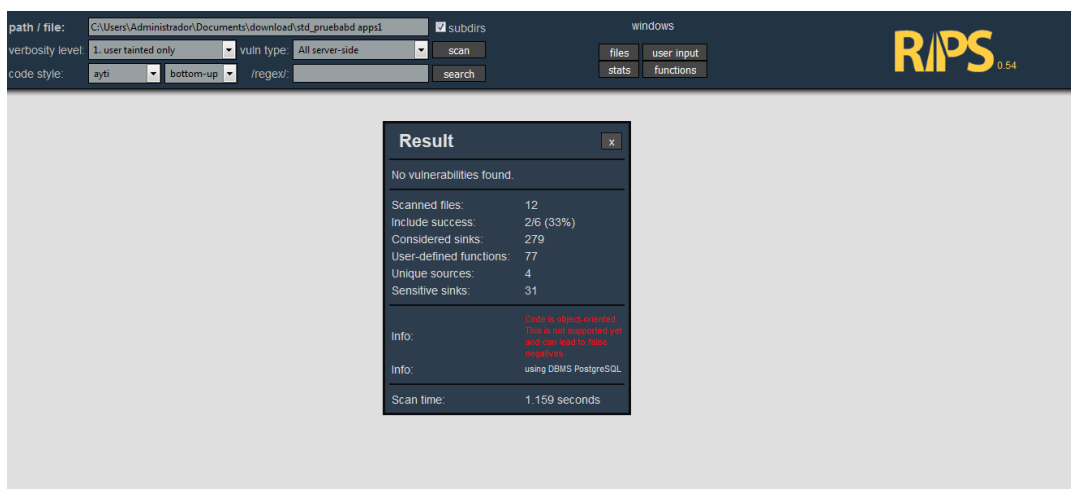


Figura 4. Resultado de la prueba de seguridad.

Se realizaron un total de 3 iteraciones de pruebas que arrojaron como resultados un total de 7 no conformidades, divididas en 2 de ortografía, 2 de redacción, 2 de funcionalidad y 1 de validación para la primera iteración, las cuales quedaron resueltas. En una segunda iteración se identifican 3 nuevas no conformidades siendo 2 de funcionalidad y 1 de validación, las cuales fueron resueltas. En una tercera iteración no se identifican nuevas inconformidades, obteniendo, de esta manera, resultados satisfactorios. La Figura 5 muestra los resultados antes descritos.

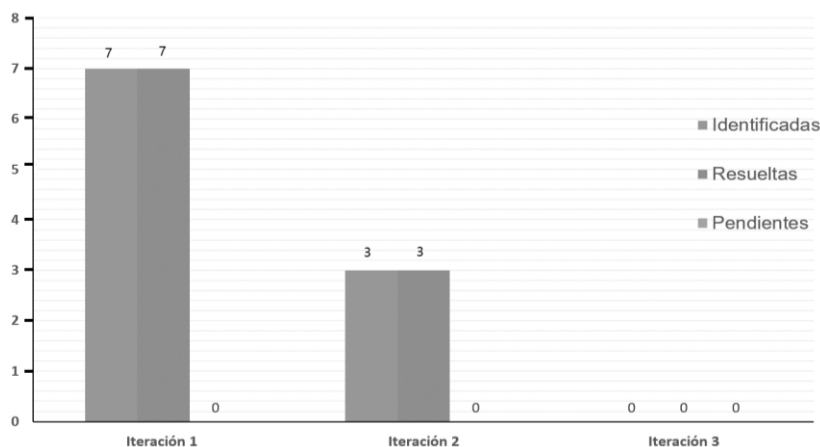


Figura 5. No conformidades por iteraciones.

Con la implementación de este sistema automatizado, el equipo de probadores de BD del CCES disminuye notablemente el tiempo que se destina a probar las aplicaciones que cuentan con SGBD SQLite. Ya que pasaron de una revisión manual, que tardaba días y/o semanas a una automatizada, que toma minutos. Dejando atrás la forma manual de comprobación se gana también en exactitud a la hora de identificar no conformidades, disminuyendo el margen de error. El CCES se beneficia con un programa de elaboración propia con el cual no es necesario pagar licencias ni actualizaciones, de fácil uso y rápido aprendizaje, con el que los probadores tienen cierto nivel de familiarización ya que cuentan con uno similar, el CEAD PostgreSQL.

Teniendo en cuenta las pruebas realizadas, así como otras consideraciones expresadas por el cliente, se puede comprobar que la solución implementada tiene un nivel satisfactorio de aceptación para el centro CCES. Además, se obtuvo un conjunto de recomendaciones y valoraciones que aportan mejoras a la propuesta de solución, en función de ampliar las funcionalidades de la misma. Finalmente se obtuvo como resultado un componente integrado al marco de trabajo

Zeolides [16], capaz de comprobar automáticamente la arquitectura de las BD del gestor SQLite en las aplicaciones informáticas desarrolladas en XETID.

## Conclusiones

Con el desarrollo del componente de revisión de estándar de arquitectura de datos para el gestor SQLite se obtienen los siguientes resultados relevantes. Con la sistematización del marco teórico de la investigación se identificaron los principales conceptos asociados al componente de revisión de arquitectura de datos del SGBD SQLite y las relaciones entre estos, lo que permitió alcanzar una mayor comprensión de la propuesta de solución. El análisis de sistemas homólogos y de los estándares de arquitectura de BD permitió identificar las tendencias en cuanto al desarrollo de herramientas revisión de estándar de arquitectura de datos e identificar las deficiencias que impiden sea utilizada. La integración de diversas áreas del conocimiento como son la ingeniería y gestión de software, BD, programación, entre otras, permitió el análisis, diseño e implementación del Componente de revisión de estándar de arquitectura de datos para el gestor SQLite. La solución fue validada a partir de la definición de una estrategia de pruebas, que permitió comprobar el correcto funcionamiento del CEAD SQLite a partir de los requerimientos definidos por el cliente. De esta manera se ha cumplido con el objetivo principal de la investigación, siendo el CEAD SQLite una herramienta que cumple con las necesidades del cliente y satisface los requerimientos del problema planteado.

## Referencias

- [1] Y. De La Paz Milán, «Desarrollo de un componente de monitoreo para el Servidor de Gestión PostgreSQL,» Universidad de las Ciencias Informáticas, La Habana, 2011.
- [2] M. Ruiz Gómez, «Creación de una base de datos estandarizada para la flota eólica de Brasil,» 2017.
- [3] XETID, «Proceso de Desarrollo de Software v1.7,» 2017.
- [4] M. Guanche Cañizares, D. Pérez Rojas y J. Reyes Pérez, «CEAD Componente de revisión de arquitectura de datos de la Empresa de Tecnología de la Información para la Defensa (XETID),» 2017.
- [5] O. Tatés Pérez, «Implementación de una aplicación móvil Android para el seguimiento de asistencia de los estudiantes de la CISICI, utilizandio Android Studio,» Ibarra, Ecuador, 2018.
- [6] D. Almonacid Inzunza, «Comparación entre gestores de bases de datos relacionales,» 2016. [En línea]. Available: <http://repositoriodigital.ucsc.cl/handle/25022009/1092>.
- [7] M. Ramos, A. Ramos y F. Montero, «Sistemas Gestores de Bases de Datos,» 2016.
- [8] R. Marín, «Los gestores de bases de datos (SGBD) más usados,» 2019.

- [9] D. González Márquez, «Creación de una práctica de bases de datos relacionales con SQLite,» 2019. [En línea]. Available: [https://repositorioinstitucional.ceu.es/bitstream/10637/10242/1/Creacion\\_DavidGonzalez\\_2019.pdf](https://repositorioinstitucional.ceu.es/bitstream/10637/10242/1/Creacion_DavidGonzalez_2019.pdf).
- [10] P. López Herrera, «Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL,» 2016.
- [11] B. Gros, A. Escofet y M. Marimón, «The design patterns as tools to guide the practice of teachers,» *Revista Latinoamericana de Tecnología Educativa*, vol. 15, n° 3, pp. 11-25, 2016.
- [12] Cognodata, «Arquitectura de datos: la base de una estrategia diferenciadora,» [En línea]. Available: <https://www.cognodata.com/blog/arquitectura-datos-estrategia-diferenciadora/>.
- [13] J. Sánchez Asenjo, «Gestión de Base de Datos. Sistemas gestores de bases de datos,» Administración de Sistemas Informáticos en Red, 2008.
- [14] D. Mera Mero y C. Villamarin Zambrano, «Categorías de beneficios de estándares y procedimientos,» 2012.
- [15] International Organization for Standardization, «NORMAS ISO 25000,» 2005.
- [16] XETID, «Tecnologías y Ciberseguridad. Marco de Trabajo para el Desarrollo de Aplicaciones Web (Zeolides),» 2019.