REVISTA CUBANA
DE TRANSFORMACIÓN DIGITAL

RNPS 2487 • ISSN 2708-3411
Vol. 4 • Nro. 2 • abril-junio 2023 • 220

ARTÍCULO DE REVISIÓN

Desafíos de las pruebas de aplicaciones iOT en Ciudades Inteligentes

Challenges of Testing iOT Applications in Smart Cities

Alejandro Miguel Güemes Esperón
aguemes@tesla.cujae.edu.cu • https://orcid.org/0000-0001-9704-9449

Martha Dunia Delgado Dapena
marta@ceis.cujae.edu.cu • https://orcid.org/0000-0002-2601-3462

UNIVERSIDAD TECNOLÓGICA DE LA HABANA "JOSÉ ANTONIO ECHEVERRÍA", CUJAE, CUBA

Francisco Maciá Pérez
pmacia@dtic.ua.es • https://orcid.org/0000-0002-2516-4728

Jose Vicente Berna Martinez
jvberna@ua.es • https://orcid.org/0000-0002-9007-6054

Iren Lorenzo Fonseca
iren.fonseca@ua.es • https://orcid.org/0000-0003-3597-4836

UNIVERSIDAD DE ALICANTE, ESPAÑA

Recibido: 2023-04-05 • Aceptado: 2023-05-12

RESUMEN

Las tecnologías de la informática y las comunicaciones constituyen elementos principales en el desarrollo de ciudades inteligentes. Permiten dotar de inteligencia a todos sus ámbitos y generar servicios y soluciones sostenibles que proporcionen una mejor calidad de vida de los ciudadanos. En la actualidad se introducen nuevos conceptos y paradigmas sociotecnológicos como Internet de las cosas (IoT). Este artículo aborda las pruebas de aplicaciones IoT en entornos inteligentes, tema de creciente interés entre investigadores y miembros de la industria del *software*, en la búsqueda de una estrategia de pruebas que permita garantizar que las soluciones obtenidas posean la calidad deseada. Se analizaron varios trabajos publicados, con el objetivo de identificar las características fundamentales de este tipo de aplicaciones y los tipos de pruebas más adecuados. Se obtuvo una estrategia de pruebas para aplicaciones IoT y la caracterización de un conjunto de herramientas que contribuyen a su automatización.

Palabras clave: pruebas de *software*, IoT, desafíos, ciudades inteligentes.

ABSTRACT

Information and communication technologies are key elements in the development of smart cities. They make it possible to provide intelligence to all areas and generate sustainable services and solutions that provide a better quality of life for citizens. Nowadays, new concepts and socio-technological paradigms such as the Internet of Things (IoT) are being introduced. This article addresses the testing of IoT applications in intelligent environments, a topic of growing interest among researchers and members of the software industry, in the search for a testing strategy to ensure that the solutions obtained have the desired quality. Several published works were analyzed, with the objective of identifying the fundamental characteristics of this type of applications and the most appropriate types of tests. A testing strategy for IoT applications and the characterization of a set of tools that contribute to their automation were obtained.

Keywords: *software testing, IoT, smart cities.*

INTRODUCCIÓN

En la concepción de Ciudad Inteligente se debe definir una arquitectura tecnológica que permita brindar nuevas soluciones y servicios a los miembros de la comunidad. En la figura 1 se muestra la arquitectura propuesta por Maciá (2017), compuesta por cuatro niveles: comunicación, monitorización, negocio o análisis de datos y aplicación (Maciá, 2017; Maciá *et al.,* 2021).

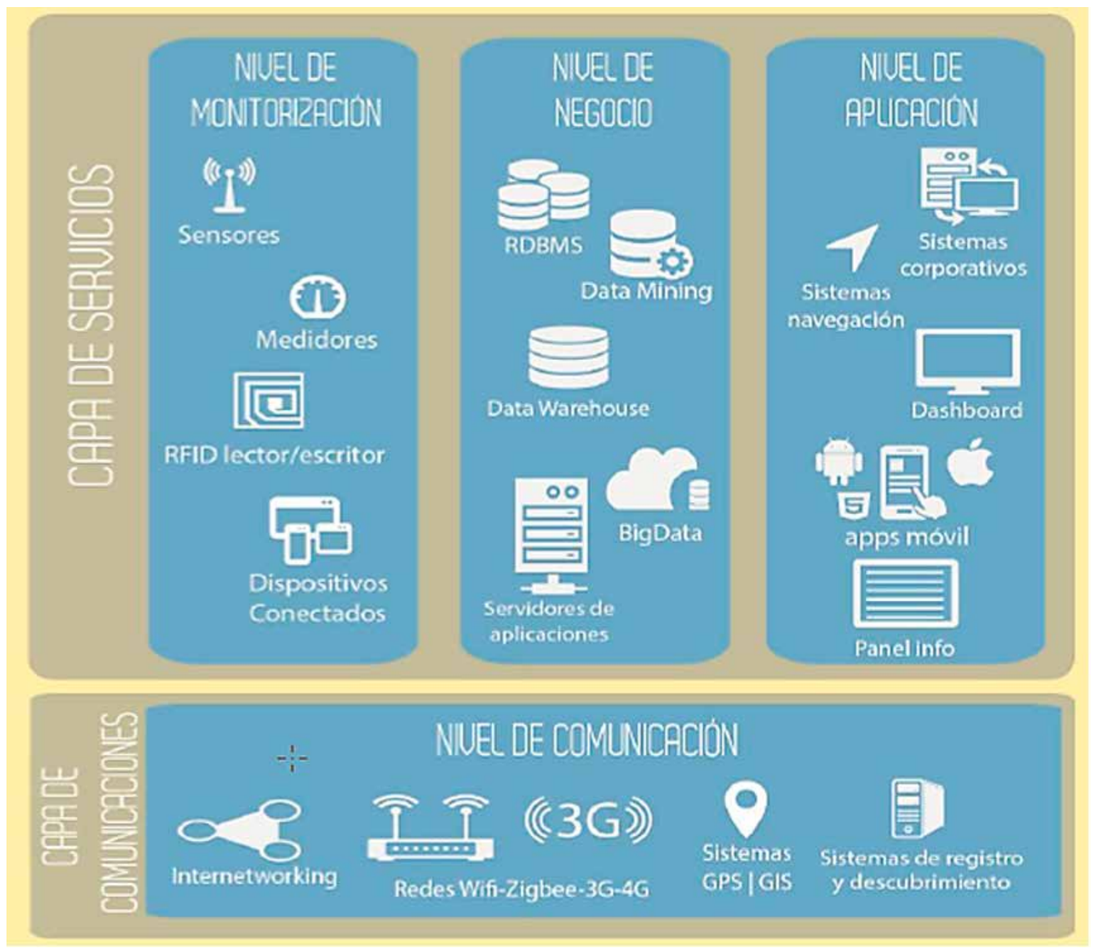

Fig. 1
Arquitectura de una Ciudad Inteligente genérica (Maciá, 2017).

Hoy no solo los ordenadores, dispositivos móviles, sensores, redes de interconexión y visores dan soporte al desarrollo digital de las ciudades inteligentes, sino que se introducen nuevos conceptos y paradigmas sociotecnológicos como Internet de las Cosas (IoT, *Internet of Things*) (Maciá, 2017; Maciá *et al.,* 2021). La calidad de las aplicaciones IoT está marcada por la prueba de diferentes escenarios de aplicación, con bancos de pruebas que se han socializado (Fink, 2020; Morrissett *et al.,* 2018; Pham *et al.,* 2020; Svítek *et al.,* 2020; Voas *et al.,* 2018).

Las aplicaciones IoT se centran en la interconexión digital de objetos cotidianos con Internet. La arquitectura IoT tiene que cumplir ciertos requerimientos para que esta tecnología sea viable (UIT, 2022). Debe permitir que la tecnología sea distribuida, donde los objetos puedan interactuar entre ellos: escalable, flexible, robusta, eficiente y segura.

En Kumar *et al.,* (2018) se definen varias capas de una arquitectura IoT: capa de aplicación, capa de soporte y gestión, capa de servicios, capa de comunicación, capa de red, capa de *hardware* y capa de entorno. Esta arquitectura describe la estructura de una solución de IoT, lo que incluye los aspectos físicos (cosas) y los aspectos virtuales (servicios y protocolos de comunicación). Adoptar una arquitectura con múltiples niveles permite concentrarse en mejorar su comprensión acerca de cómo los aspectos más importantes de la arquitectura funcionan antes de que se integre a la aplicación IoT. El enfoque modular ayuda a gestionar la complejidad de este tipo de soluciones (Santos *et al.,* 2020).

La diferencia entre probar un *software* tradicional y una aplicación IoT en un entorno inteligente, se basa en que las tradicionales toman su entrada de usuario a través de dispositivos periféricos y algunos dispositivos que se pueden tocar; pero en IoT, muchos dispositivos inteligentes están recibiendo su entrada de otros dispositivos inteligentes, como sensores que recogen las métricas y envían los valores al *software* para analizar estos datos (Fissi *et al.,* 2021; Gomez *et al.,* 2019; Quijano-Sánchez *et al.,* 2020). Por tanto, probar estas aplicaciones a gran escala suele ser difícil (Enoiu, 2020), ya que involucran grandes cantidades de datos generados y no se ha encontrado estándares o buenas prácticas en la literatura sobre estrategias de prueba que se ajusten mejor a este tipo de aplicaciones. La prueba de sistemas inteligentes requiere un marco de prueba automatizado, debido a la cantidad de dispositivos IoT y el procesamiento de datos provenientes de fuentes diversas, lo que refuerza el carácter combinatorio de estas (Ahmed *et al.,* 2019; Braem *et al.,* 2016; Popereshnyak *et al.,* 2018). En el proceso de prueba, la etapa más costosa es el diseño de los casos de prueba, lo que hace necesario automatizar su generación, utilizando criterios que permitan la reducción de la suite de pruebas y la elevación de sus niveles de efectividad (Ahmed *et al.,* 2019; Serna *et al.,* 2019; Valle-Gómez *et al.,* 2019). En las pruebas de *software* tradicional se han empleado diferentes técnicas para diseñar y reducir la suite de pruebas, que podrían ser adecuadas a las condiciones particulares de las aplicaciones IoT (Ahmed *et al.,* 2019; Krichen, 2019; Murad *et al.,* 2018; Voas *et al.,* 2018).

El objetivo de este trabajo es identificar limitaciones, desafíos y particularidades de las pruebas de este tipo de aplicaciones, a partir de un estudio inicial de la bibliografía.

METODOLOGÍA

La metodología utilizada para este trabajo consistió en una búsqueda de la literatura y la exploración del contexto. Se presenta un recorrido a través del conocimiento que existe sobre las pruebas de aplicaciones IoT en entornos inteligentes, mediante una revisión del estado del arte de los principales conceptos relacionados. La búsqueda se realizó por palabras clave en Google académico, teniendo en cuenta los trabajos publicados de 2017 a 2022. El estudio de la bibliografía se centró en dos aspectos: los desafíos de las pruebas de aplicaciones IoT y las herramientas de pruebas existentes.

RESULTADOS Y DISCUSIÓN

Desafíos y limitaciones de las pruebas de aplicaciones IoT

Durante la revisión bibliográfica inicial se pudo conocer que:

- Los aspectos de seguridad y privacidad de las soluciones de IoT se discuten ampliamente en la bibliografía consultada y se proponen muchos enfoques alternativos; sin embargo, siguen siendo, junto a las pruebas, el principal desafío de las soluciones IoT.
- El número de artículos que están dedicados a métodos de prueba especializados personalizados para las especificaciones de IoT, es relativamente bajo.
- La interoperabilidad de los dispositivos, los protocolos y un gran número de sus posibles combinaciones para probar, deberán estar respaldados por una investigación más extensa.
- La interoperabilidad de los dispositivos de IoT debe abordarse más en los trabajos científicos. Aquí, se consideran dos corrientes como perspectiva: los métodos específicos de IoT para pruebas de integración, y los métodos de cómo combinar conjuntos eficientes de variantes de dispositivos y partes de infraestructura y versiones, con respecto a la heterogeneidad de las soluciones de IoT y, a veces, incluso a la imposibilidad de actualizar a una versión más reciente del *firmware* o *software* del dispositivo.
- No se cuenta con una estrategia de prueba general para aplicaciones IoT. Para proyectos de *software*, muchas pautas sobre cómo determinar la intensidad de las pruebas parte de las particulares del sistema bajo prueba y de cómo elegir las mejores técnicas de prueba que existan. Lo mismo se deberá tener en cuenta para los proyectos de IoT, respetando todas las especificaciones de las infraestructuras de IoT.
- Otra área que vale la pena explorar es el desarrollo de técnicas de diseño de prueba específicas para la prueba de soluciones IoT, bajo una conexión de red limitada y restricciones técnicas relacionadas. A medida que la dependencia de los usuarios de Internet y los servicios de IoT crece continuamente, esta área también se vuelve más relevante.
- En el área de modelado del sistema bajo prueba se deberá desarrollar y verificar modelos adecuados para la generación semiautomatizada o automatizada de casos de prueba, en el proceso de prueba basado en modelos prácticos. A diferencia de los sistemas de *software* clásicos, estos modelos también incluirán capas físicas y de protocolo, ya que son mucho más heterogéneas en el caso de las soluciones de IoT.

- No hay un patrón definido de estrategias de prueba que se deben seguir para el desarrollo de pruebas.
- Las técnicas de verificación formal y las pruebas basadas en modelos para aplicaciones IoT y *Smart Cities*, en general sufren un problema de explosión de estado.
- Las aplicaciones IoT involucran una variedad de dispositivos finales distribuidos y multiescalables.
- La naturaleza colaborativa de los sistemas IoT conectados a través de Internet, aumenta la heterogeneidad de los datos, por lo que deben ser procesados para una correcta toma de decisiones en un entorno en tiempo real.
- Los sistemas de IoT, distribuidos de forma remota, crean bucles para las violaciones de datos, por lo tanto, abren nuevos desafíos para la seguridad y escalabilidad del sistema.

Herramientas de pruebas de software

Existen herramientas para la realización de pruebas, las cuales son, en su gran mayoría, privadas, muy costosas y se necesitan conocimientos de programación para su ejecución. En este apartado se resumen las características de algunas herramientas identificadas durante el estudio de la bibliografía, que se pueden emplear para la automatización parcial del proceso de pruebas de aplicaciones IoT.

- ***Parasoft Virtualize* (Parasoft, 2022):** brinda a los usuarios un control total sobre su entorno de prueba, al eliminar obstáculos, como dependencias no disponibles/inestables o acceso a costosos once laboratorios de prueba de terceros. Los usuarios pueden incluso usar Parasoft Virtualize para permitir que las pruebas comiencen antes de que se implemente una dependencia de servicio. Los usuarios pueden crear servicios virtuales ligeros, tanto a través del escritorio rico en funciones, como de la interfaz web intuitiva que permite incluso a los usuarios novatos crear simulaciones a partir de definiciones de servicios y tráfico registrado.
- ***SOAtest* (JMeter, 2022):** la solución de prueba de API SOAtest de Parasoft es ampliamente reconocida. Con herramientas visuales de arrastrar y soltar, los usuarios pueden crear los escenarios de prueba más complejos sin tener que escribir una sola línea de código. Con su generador de pruebas SMART API plugin para Chrome, SOAtest monitorea la actividad en su interfaz de usuario web, a partir de pruebas manuales o exploratorias, y convierte las llamadas API entre bastidores en escenarios de prueba API significativos. Para llevar más allá de la simple grabación y reproducción, SOAtest aprovecha la inteligencia artificial y el aprendizaje automático para comprender lo que hacen las llamadas a la API, y luego crea un escenario de prueba de API significativo que es reutilizable, dinámico e impactante. Los resultados de las pruebas proporcionan tareas significativas y procesables en SOAtest, sistema de informes rico y dinámico, que puede tomar la forma de un informe PDF simple a un documento HTML dinámico multinivel, el cual describe qué pruebas se ejecutaron, cuál era el estado y a qué requisitos estaban asociados, lo que permite que múltiples partes interesadas comprendan el estado de sus aplicaciones críticas.

- ***JMeter* (SmartBear, 2022):** es una herramienta de prueba de código abierto, desarrollada por Apache *Software* Foundation, una aplicación Java pura que se puede utilizar para medir el rendimiento de aplicaciones, diferentes servicios de *software* y productos tanto en recursos estáticos como dinámicos. Inicialmente, *JMeter* fue diseñado para probar aplicaciones web; pero más tarde se ha expandido para probar otras funciones como pruebas funcionales, de rendimiento, de regresión, de estrés y servidor de bases de datos probado sobre la base de varias tecnologías.
- ***SoapUI* (Acunetix, 2022):** es una herramienta desarrollada en Java, utilizada para pruebas de aplicaciones con arquitectura SOA o REST. Soporta múltiples protocolos como SOAP, REST, HTTP, JMS y JDBC. La herramienta cuenta con una versión de código abierto y otra versión paga desarrollada por la compañía SmartBear. Habilita las pruebas funcionales de API automatizadas, esenciales para los proyectos de prueba de IoT, ya que la mayor parte del intercambio de datos dentro de un sistema de IoT se realiza a través de API. Es una aplicación muy completa, con muchas funcionalidades, con lo cual puede llegar a ser un poco complicada de utilizar para la función que se requiere en cada momento.
- ***Acunetix* (Acunetix, 2022):** esta aplicación ejecuta una serie de pruebas de seguridad, totalmente configurables por el usuario, para identificar las vulnerabilidades, tanto en la programación de la página web como en la configuración del servidor, y detecta técnicas de *hacking* como pueden ser ataques de ejecución de código y de autentificación. Una vez concluido este proceso, la versión genera una serie de informes y especifica los fallos detectados en el análisis de la página web. Estos informes se mostrarán en la pantalla en forma de gráfica. Habilita el análisis de vulnerabilidades de las interfaces de usuario web de IoT y las API REST.

Estrategia de pruebas de aplicaciones IoT

A partir del estudio de la bibliografía se propone una estrategia de pruebas de aplicaciones IoT, que incluye:

- **Pruebas de integración:** este tipo de prueba garantiza una interacción estable del *software* IoT con dispositivos inteligentes. El objetivo de las pruebas de integración es garantizar que los módulos individuales funcionen como se espera después de combinarlos con otros módulos. Muchas organizaciones utilizan pruebas unitarias combinadas o pruebas de flujo de trabajo funcional de un extremo a otro, que se emplean para las pruebas de integración.
- **Pruebas de seguridad:** aseguran que el *software* IoT recopile, analice y procese datos correctamente, sin filtraciones, a través de todos los dispositivos, las redes y los sistemas inteligentes. Es una parte integral de las pruebas que garantiza que el *software* y las aplicaciones web estén libres de lagunas, vulnerabilidades, amenazas y riesgos, que puedan causar una gran pérdida a la empresa/organización, y verifique si sus datos y recursos están protegidos de posibles intrusos.

- **Pruebas de usabilidad:** ayudan a escuchar a los clientes con cuidado, identificando los puntos más vulnerables del flujo de trabajo de la aplicación. Consisten en seleccionar a un grupo de usuarios de una aplicación y solicitarles que lleven a cabo las tareas para las cuales fue diseñada, en tanto el equipo de diseño, desarrollo y otros involucrados, toman nota de la interacción, particularmente de los errores y las dificultades con las que se encuentren los usuarios.
- **Pruebas de rendimiento:** a través de escenarios de prueba de carga, estrés e Internet, se puede simular y probar productos de IoT, para aumentar la productividad, la estabilidad de la carga y hacer que el código se ejecute sin problemas, dentro de ecosistemas de IoT inusuales. Es una técnica de prueba de *software* no funcional, que determina cómo la estabilidad, la velocidad, la escalabilidad y la capacidad de respuesta de una aplicación, se mantienen bajo una determinada carga de trabajo.
- **Pruebas de confiabilidad, compatibilidad y escalabilidad:** estos tipos de pruebas ayudan a construir los entornos de IoT correctos, y a optimizar e implementar nuevas funciones. La compatibilidad se encarga de la interacción fluida y sin errores entre el *software* de IoT y diferentes dispositivos inteligentes, plataformas, capas de red y sistemas operativos. La escalabilidad es la prueba de cualquier *software* o aplicación para verificar su capacidad de soportar y escalar, de acuerdo con el número de usuarios que acceden a ella en un momento particular/específico.

CONCLUSIONES

Con el desarrollo de las aplicaciones informáticas y la aparición de entornos inteligentes, las pruebas de *software* ocupan a diferentes autores y expertos en el tema. A través del estudio de la bibliografía se pudo identificar que existen múltiples desafíos en el ámbito de las pruebas de aplicaciones IoT, relacionados fundamentalmente con la seguridad y disponibilidad, además de la generación de datos y las combinaciones de valores que faciliten el diseño de las pruebas. La estrategia de pruebas propuesta en este artículo se basó en las recomendaciones que plantean los autores de los trabajos estudiados. Está encaminada a potenciar la detección de defectos relacionados fundamentalmente con requisitos no funcionales o de calidad presentes en aplicaciones IoT. Además, se pudo conocer que existen herramientas que brindan soporte al proceso de pruebas y contribuyen a su automatización parcial o total, lo que permite agilizar el proceso de desarrollo de aplicaciones en estos entornos y garantizar una calidad adecuada. Para trabajos futuros se debe incorporar al estudio bibliográfico un conjunto mayor de propuestas vinculadas a modelos de optimización, que permitan generar conjuntos de pruebas reducidos y la generación automática de casos de pruebas aplicadas a IoT.

REFERENCIAS

Acunetix (2022). Available from: https://www.acunetix.com [accessed 6 Jul, 2022].

Ahmed, B. S., Bures, M., Frajtak, K., & Cerny, T. (2019). Aspects of quality in Internet of Things (IoT) solutions: A systematic mapping study. IEEE Access, 7, 13758-13780. doi:10.1109/access.2019.2893493.

Braem, B., Latré, S., Leroux, P., Demeester, P., Coenen, T., & Ballon, P. (2016, September). Designing a smart city playground: Real-time air quality measurements and visualization in the City of Things testbed. In 2016 IEEE International Smart Cities Conference (ISC2), pp. 1-2. IEEE. doi: 10.1109/ISC2.2016.7580871.

Enoiu, E., Tukseferi, G., & Feldt, R. (2020, December). Towards a model of testers' cognitive processes: Software testing as a problem solving approach. In 2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C), pp. 272-279. IEEE. doi: 10.1109/QRS-C51114.2020.00053.

Fink, J. (2020, September). Digital City Testbed Center: Using campuses as smart city testbeds in the binational Cascadia region. In 2020 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 362-367. IEEE. doi: 10.1109/SMARTCOMP50058.2020.00078.

Fissi, S., Romolini, A., Gori, E., & Contri, M. (2021). The path toward a sustainable green university: The case of the University of Florence. *Journal of Cleaner Production,* 279, 123655.

Gomez, A. K., & Bajaj, S. (2019, October). Challenges of testing complex Internet of Things (IoT) devices and systems. In 2019 11th international conference on knowledge and systems engineering (KSE), pp. 1-4. IEEE. doi: 10.1109 / KSE. 2019.8919324.

Harris, A., Stovall, J., & Sartipi, M. (2019, December). Mlk smart corridor: An urban testbed for smart city applications. In 2019 IEEE International Conference on Big Data (Big Data), pp. 3506-3511. IEEE. doi: 10.1109/BigData47090.2019.9006382.

JMeter. (2022). A. S. Foundation. Available from: https://www.jmeter.apache.org [accessed 6 Jul, 2022].

Krichen, M. (2019). Improving formal verification and testing techniques for Internet of Things and Smart Cities. Mobile networks and applications, pp. 1-12. doi: 10.1007/s11036-019-01369-6.

Kumar, N. M., & Mallick, P. K. (2018). The Internet of Things: Insights into the building blocks, component interactions, and architecture layers. *Procedia computer science,* (132): 109-117.

Maciá Pérez, F. (2017). Smart university: hacia una universidad más abierta. Marcombo, España. Isbn 9786076228142.

Maciá Pérez, F., Berna Martinez, J. V., & Lorenzo Fonseca, I. (2021). Modelling and implementing smart universities: An it conceptual framework. Sustainability, 13(6): 3397. https://doi.org/10.3390/su13063397.

Morrissett, A., & Abdelwahed, S. (2018, October). A physical testbed for smart city research. In 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), pp. 1-2. IEEE. doi: 10.1109/AICCSA.2018.8612899.

Murad, G., Badarneh, A., Qusef, A., & Almasalha, F. (2018, July). Software testing techniques in iot. In 2018 8th International conference on computer science and information technology (CSIT), pp. 17-21. IEEE. doi: 10.1109/CSIT.2018.8486149.

Parasoft. (2022). Available from: https://www.parasoft.com [accessed 6 Jul, 2022].

Pham, T. V., Nguyen, A. T. T., Ngo, T. D., Le, D. H., Le, K. C., Nguyen, T. H., & Le, H. Q. (2020, November). Proposed smart university model as a sustainable living lab for university digital transformation. In 2020 5th International Conference on Green Technology and Sustainable Development (GTSD), pp. 472-479. IEEE. doi: 10.1109/GTSD50082.2020.9303086.

Popereshnyak, S., Suprun, O., Suprun, O. & Wieckowski, Y. (2018). Características de prueba de aplicaciones de IoT basadas en la red de modelado, XIV-th International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), pp. 127-131, Lviv, Ucrania. doi: 10.1109 / MEMSTECH.2018.8365717.

Quijano-Sánchez, L., Cantador, I., Cortés-Cediel, M. E., & Gil, O. (2020). Recommender systems for Smart Cities. *Information systems,* (92): 101545.

Santos, M. G. D., Ameyed, D., Petrillo, F., Jaafar, F., & Cheriet, M. (2020). Internet of Things architectures: A comparative study. Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Analyze-of-IBM-IoT-Architecture_fig5_340962806 [accessed 6 Jul, 2022].

Serna, E., Martínez, R., & Tamayo, P. (2019). Una revisión a la realidad de la automatización de las pruebas del software. *Computación y Sistemas,* 23(1): 169-183.

SmartBear. (2022). Available from: https://www.SoapUI.org [accessed 6 Jul, 2022].

Svítek, M., Dostál, R., Kozhevnikov, S., & Janča, T. (2020, June). Smart City 5. 0 testbed in Prague. In 2020 Smart City Symposium Prague (SCSP), pp. 1-6. IEEE. doi: 10.1109/SCSP49987.2020.9133997.

UIT. (2022). Y.2066: Requisitos comunes de la Internet de las cosas. https://www.itu.int/rec/T-REC-Y.2066-201406-I/es (2014d) [accessed 6 Jul, 2022].

Valle-Gómez, K. J., Delgado-Pérez, P., Medina-Bulo, I., & Magallanes-Fernández, J. (2019, May). Software testing: cost reduction in Industry 4.0. In 2019 IEEE/ACM 14th International Workshop on Automation of Software Test (AST), pp. 69-70. IEEE. doi: 10.1109/AST.2019.00018.

Voas, J., Kuhn, R., & Laplante, P. (2018, March). Testing IoT Systems. In 2018 IEEE Symposium on Service-Oriented System Engineering (SOSE), pp. 48-52. IEEE Computer Society. doi: 10.1109/SOSE.2018.00015.