F-IoT: Una herramienta para la configuración de sistemas de IoT

Sebastian U. Flores[1], Mario M. Berón[1]

[1] Área de Programación y Metodologías de Desarrollo de Software,
Departamento de Informática, Facultad de Ciencias Físico Matemáticas y Naturales,
Universidad Nacional de San Luis (UNSL), San Luis, Argentina
s.flores@outlook.com.ar, mmberon@gmail.com

Abstract. En las últimas décadas, los ámbitos de la vida de las personas y de la industria conectados a Internet se vieron incrementados, llegando a abarcar a una amplia gama de dispositivos bajo el concepto de Internet de las Cosas (IoT por sus siglas en inglés). Los progresos en el IoT favorecieron la transición hacia la cuarta revolución industrial, y la pandemia del Covid-19 no hizo más que acelerarla. Sin embargo, los beneficios ofrecidos por la revolución del IoT pueden verse seriamente afectados por las dificultades que se presentan en el desarrollo de sistemas que conecten a Internet, de forma escalable, estable y segura, a grandes cantidades de dispositivos, mayoritariamente limitados en sus capacidades de conectividad, memoria o procesamiento. En este artículo, se presenta una herramienta desarrollada para abordar algunos de los desafíos latentes asociados al IoT, y así facilitar su adopción en países en vías de desarrollo.

Keywords: IoT, Internet, Dispositivo, Nube, Escalabilidad, Seguridad, Privacidad, Confiabilidad, Arquitectura.

1 Introducción

En los últimos siglos, los avances en las ciencias dieron lugar a que se crearan nuevos ámbitos de trabajo y novedosas tecnologías. Así es que los progresos en las telecomunicaciones y la informática en general permitieron alcanzar uno de los mayores logros de la humanidad: La creación de Internet, que permitió conectar a diferentes partes del mundo bajo una misma red.

Internet abrió un nuevo universo de posibilidades para que las personas pudieran compartir toda clase de contenidos entre ellas. Esto trajo nuevos beneficios para la humanidad, tales como la creación de nuevos puestos de trabajo en los que se involucre a personas de diferentes partes del mundo; el fortalecimiento de la participación ciudadana en decisiones de carácter político y la distribución de contenido multimedia en masa.

En las últimas décadas, el grado de adopción a Internet alcanzó tal magnitud, que comenzaron a desarrollarse sistemas para el control a distancia de diferentes eventos

que ocurren en ámbitos industriales, sociales o de la naturaleza. Una de las primeras pruebas se realizó públicamente en el año 1984, cuando se presentó la primera máquina de bebida Cola que reportaba por Internet la temperatura y disponibilidad de bebidas en su interior [1].

A partir de esta presentación, se continuaron realizando avances en electrónica e informática, de tal forma que, en 1999, Ashton [2] introdujo el término Internet of Things (i.e. IoT, en español, Internet de las Cosas), al presentar la vinculación de dispositivos de la vida cotidiana a Internet, a través de un pequeño y económico chip llamado RFID.

Con el paso del tiempo, el término IoT continuó utilizándose para clasificar de forma global a diferentes avances tecnológicos. Sin embargo, no existe en la actualidad una definición completa que contemple todas las facetas del IoT y no restrinja el alcance de su horizonte, para permitir que continúe expandiéndose día a día, a medida que avanzan las tecnologías y las formas de ver el mundo.

1.1 Contexto del IoT

De acuerdo con Oracle [3], el IoT describe la red de objetos físicos ("cosas") que llevan incorporados sensores, software y otras tecnologías con el fin de conectarse e intercambiar datos con otros dispositivos y sistemas a través de Internet.

Pueden identificarse cuatro ámbitos de aplicación de tecnologías IoT [4]:

- **Hogares conectados.** Los dispositivos de IoT pertenecientes a este ámbito incluyen una amplia gama de electrodomésticos creados con un enfoque de diseño orientado al usuario [5, 6], que permite analizar el comportamiento de este último y adaptarse a sus necesidades, para mejorar la experiencia de uso y la utilidad. Este ámbito también incluye dispositivos para el incremento de la seguridad hogareña, como sensores de gas/humo, cerraduras inteligentes, alarmas, entre otros.
- **Vehículos conectados.** Este ámbito todavía se encuentra en etapas tempranas, e incluye, entre otros beneficios, la posibilidad de formar una red de vehículos conectados a Internet que puedan comunicarse entre sí para realizar optimizaciones en el tráfico e incrementar la seguridad de los pasajeros.
- **Ciudades inteligentes.** Este ámbito incluye la posibilidad de instalar sensores en diferentes partes de una ciudad, tanto en ámbitos públicos como privados, para obtener información que ayude a mejorar los servicios públicos, la calidad del aire, disminuir el ruido ambiental, optimizar el mantenimiento de los sistemas de iluminación y carreteras, detectar fugas en cañerías de agua potable, entre otros.
- **La industria.** Este ámbito es conocido como Internet Industrial de las Cosas (IIoT), y refiere al uso de dispositivos para optimizar e incrementar la seguridad en fábricas, servicios de logística, venta minorista, sistemas de producción de energía, entre otros.

En los últimos años, se produjeron grandes avances tecnológicos que propiciaron el surgimiento y masificación del IoT. Algunos de estos avances se presentan a continuación:

- Disminución de los costos en sensores de buena calidad, y en chips de alta potencia.
- Grandes progresos en los medios de comunicación, que incrementaron su ancho de banda y disminuyeron, tanto su latencia como sus costos, a nivel mundial.
- Expansión y perfeccionamiento de las plataformas en la nube para el almacenamiento y procesamiento de información.
- Avances en las tecnologías de análisis de datos y aprendizaje automático.
- Incremento de la accesibilidad a la tecnología, a partir de la expansión de inteligencias artificiales conversacionales y dispositivos móviles.

1.2 Componentes de un Sistema de IoT

Cada sistema de IoT, dependiendo del ámbito y de las necesidades de negocio, estará conformado por diferentes componentes, entre los que se encuentran los siguientes [4]:

- **Dispositivos de IoT.** Son módulos tecnológicos conformados por sensores, actuadores, sistemas de conectividad y microcontroladores, cuyo fin es la recopilación de datos del entorno, de los usuarios o de los patrones de uso; la comunicación de los datos recopilados a través de Internet; y la ejecución de acciones para transformar el ambiente físico donde se encuentran, por medio de sus actuadores.
- **Servidores de almacenamiento y cómputo.** Se encargan de integrar los datos publicados por los dispositivos IoT, almacenarlos, analizarlos por medio de aprendizaje automático y otras tecnologías, y tomar decisiones sobre los mismos. Estas decisiones incluyen informar a administradores de los sistemas de IoT, enviar órdenes automáticas a los dispositivos o disparar procesos de trabajo en la nube, entre otras. En su mayoría, se encuentran desplegados en la nube, por lo que cuentan con capacidades de almacenamiento y cómputo prácticamente sin límites.
- **Puertas de enlace.** Son módulos tecnológicos que actúan de intermediarios entre dispositivos IoT y servidores. Estos no solo funcionan como puntos de conexión, sino que llevan a cabo un subconjunto de las funciones de cómputo y almacenamiento realizadas por los servidores, buscando incrementar la seguridad y optimizar los tiempos de respuesta al reducir el número de llamadas a servidores.
- **Aplicaciones de usuario.** Son aplicaciones de software que poseen interfaces visuales por medio de las cuales los diferentes usuarios de la solución IoT pueden visualizar los datos obtenidos por los dispositivos, controlar el estado de conexión, modificar configuraciones de la solución IoT y enviar órdenes a los dispositivos.
- **Computación de borde.** Es un conjunto de tecnologías que permite dotar a los dispositivos y a las puertas de enlace de la capacidad de realizar un análisis y procesamiento sobre los datos obtenidos, para tomar decisiones más rápidamente e incrementar la seguridad, evitando una dependencia total de los servidores.

En base a los párrafos anteriores, puede notarse que el espectro del IoT es muy amplio y que, si bien existen grandes avances, debe tenerse en cuenta una amplia variedad de factores al momento de diseñar e implementar un sistema de IoT [7, 8, 9]:

- **Seguridad.** Este es un factor muy importante que incluye múltiples facetas como la protección de datos en reposo y en tránsito; la seguridad integrada en los dispositivos y puertas de enlace; y las protecciones complementarias añadidas en los procesos de autenticación usados por las aplicaciones de usuario y servidores [10].
- **Privacidad.** Dependiendo del entorno en donde se desplieguen los sistemas de IoT, diferentes personas podrían verse involucradas, de forma voluntaria o no, incluso pudiendo desconocer la clase de información que se recopila, la forma en que se manipula y los fines de tal recopilación. Es por ello que, sumado a las protecciones de seguridad establecidas, debe minimizarse (y de ser posible, evitarse) el almacenamiento de Información de Identificación Personal (PII) [11, 12]. Por otra parte, debe informarse con claridad a todas las personas que puedan interactuar con el sistema de IoT, sobre el propósito de sus interacciones, además de solicitar su consentimiento, tal y como se ha establecido en diferentes legislaciones alrededor del mundo.
- **Restricciones legales.** Como se mencionó anteriormente, existen diferentes legislaciones alrededor del mundo, asociadas a la privacidad, la localidad, la pertenencia y los medios habilitados para la extracción, almacenamiento y procesamiento de los datos [13, 14, 15].
- **Confiabilidad.** Las soluciones IoT a menudo se implementan a gran escala y, pueden funcionar en redes poco seguras o de mala calidad, con una alta exposición a condiciones ambientales adversas, y con dispositivos restringidos en sus capacidades de procesamiento, almacenamiento y protección frente a fallas externas. Para disminuir los riesgos asociados a los factores mencionados, es esencial diseñar la arquitectura de la solución IoT poniendo foco en la resiliencia, la alta disponibilidad y el uso de estándares de la industria para todos los componentes de la solución. Debe asegurarse que, a pesar de los problemas que se puedan presentar, la solución funcionará de forma correcta en los momentos críticos.
- **Escalabilidad.** La arquitectura de una solución IoT debe planificarse teniendo en cuenta que, con el paso del tiempo, el mismo puede escalar de forma horizontal (si se agregan más dispositivos de un mismo tipo) o vertical (si se incluyen dispositivos diferentes). También puede escalar el número de usuarios y sistemas externos que interactúan con la misma, y que pueden afectar a su rendimiento general.
- **Flexibilidad.** Las soluciones de IoT deben ser capaces de integrar componentes de diferentes proveedores, y de reemplazarlos por otros en caso de ser necesario.
- **Eficiencia energética.** Los dispositivos podrían ser desplegados en entornos donde sea difícil o costoso acceder al suministro eléctrico, por lo que deberían tener un bajo consumo energético. Esto trae otras implicancias como la selección de hardware de menor potencia, el uso de protocolos de comunicación más ligeros (y quizás, menos seguros) o la necesidad de utilizar energías renovables.
- **Conectividad.** Es esencial analizar los requisitos de conectividad de una solución IoT, desde sus etapas preliminares. Dependiendo del nivel de disponibilidad requerida, podría verse afectada la arquitectura de la solución y los estándares utilizados en sus diferentes componentes. Esto finalmente tendrá un fuerte impacto en los costos de implementación y mantenimiento de la solución, y se verá notablemente influenciado por las características del entorno en el que se despliega.

2 F-IoT: una herramienta para la creación y monitoreo de Sistemas de IoT

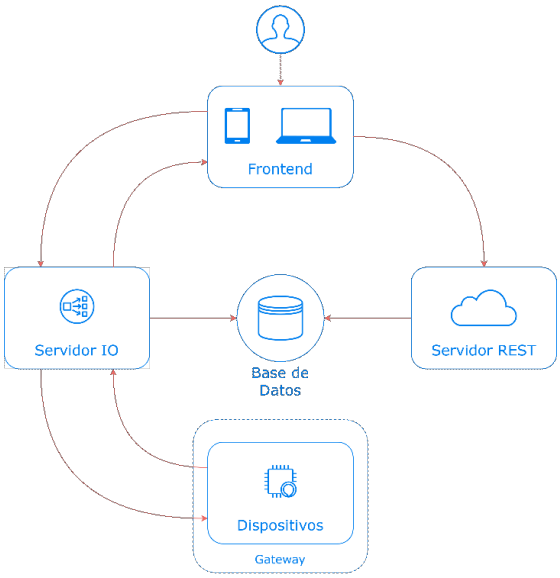

Ilustración 1: Arquitectura de F-IoT.

F-IoT es una plataforma desarrollada utilizando tecnologías modernas que poseen un gran soporte de la comunidad y de sus creadores. El mismo ofrece a los usuarios la posibilidad de diseñar uno o más sistemas de IoT, indicando los dispositivos que hay en los mismos, los componentes físicos que conforman a cada clase de dispositivo, la distribución física de los dispositivos y la forma en que estos se comunican entre sí. Cada dispositivo registrado dentro de F-IoT es un gemelo virtual de un dispositivo físico real, por lo que no puede ser modificado luego de su creación. Únicamente puede ser añadido o removido de un sistema de IoT existente. Una vez que se conecta un dispositivo físico con su gemelo virtual, F-IoT brinda una gran flexibilidad para recibir la información publicada por sus sensores, enviar órdenes a sus actuadores, monitorear su estado físico en general e interconectarlo con otros dispositivos registrados.

En la Ilustración 1 se expone la arquitectura de F-IoT. Como puede apreciarse, está compuesta por cuatro componentes principales, cada uno de los cuales será definido en detalle en las secciones subsiguientes.

2.1 Repositorio de datos

Para el almacenamiento de los datos utilizados en cada componente de F-IoT, se decidió utilizar el gestor de bases de datos relacionales Microsoft SQL Server. Esta

decisión se basó esencialmente en que SQL Server ha demostrado poseer un rendimiento excepcional, además de ser reconocido por su seguridad y cumplimiento normativo, y de ofrecer una integración directa con Microsoft Azure [16].

Habiendo elegido el gestor de base de datos, se procedió a diseñar el modelo de datos que sería utilizado en todas las operaciones de F-IoT, como puede verse en la Ilustración 2.

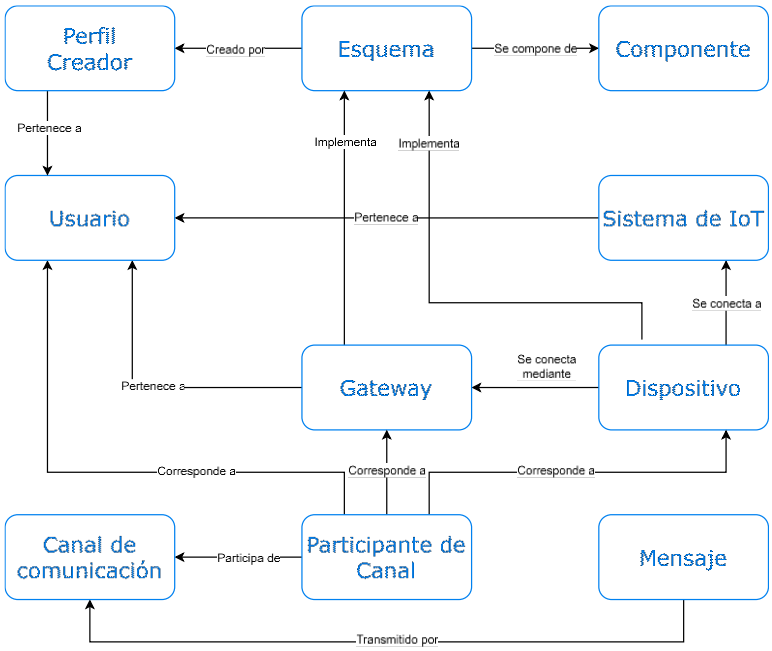

Ilustración 2. Modelo de datos de F-IoT.

2.2 Servidor REST

El servidor REST es un sistema que ofrece servicios web siguiendo los principios de la arquitectura REST, un estilo de diseño que se basa en estándares web para facilitar la comunicación entre sistemas distribuidos. La arquitectura REST se compone de tres elementos principales: controladores, servicios y modelos. En F-IoT, el servidor REST expone una interfaz de APIs REST para que usuarios y dispositivos puedan acceder a la información y a las funcionalidades de negocio de F-IoT.

El servidor REST está desarrollado íntegramente utilizando el framework open source **.NET**, en su versión 8 [17]. Este permite construir aplicaciones multiplataforma con una gran versatilidad, incluyendo todos los paquetes necesarios para la implementación de los patrones de diseño del estado del arte y las últimas funcionalidades de seguridad y eficiencia. Cuenta con soporte directo de Microsoft y de la comunidad de desarrolladores del mundo.

Por otra parte, .NET ofrece la posibilidad de instalar el ORM **Entity Framework Core**, que presenta beneficios como la integración nativa al framework .NET, la creación de consultas a la base de datos por medio de **Language Integrated Queries (LINQ)** [18], la compatibilidad con múltiples motores de base de datos y un robusto manejo de sesiones en la base de datos.

Para brindar seguridad en las operaciones, todo cliente del servidor REST debe autenticarse mediante un token de seguridad. Cada token de seguridad es creado a través del estándar abierto **JSON Web Token (JWT)** [19, 20], uno de los más usados en la actualidad. Entre los beneficios principales de usar JWT se encuentra la posibilidad de transmitir un token de seguridad en un objeto JSON cuyo tamaño es relativamente pequeño cuando se lo compara con otros estándares basados en XML como **Security Assertion Markup Language Tokens (SAML)** [21], razón que lo hace ideal para el uso en ambientes Web, y más aún en sistemas de IoT donde las capacidades de los dispositivos suelen ser limitadas. Por otro lado, JWT permite firmar los tokens utilizando una clave secreta (a través del algoritmo HMAC [22]) o un par de claves pública/privada, utilizando RSA o ECDSA [23].

2.3 Servidor IO

El servidor IO establece el soporte necesario para que las comunicaciones en tiempo real funcionen, a través de canales de comunicación. Un canal de comunicación es un espacio en el que pueden intercambiar mensajes dos o más clientes del servidor IO entre sí, utilizando el patrón publicar-suscribir [24]. Cada vez que un cliente se conecta a un canal de comunicación, puede seleccionar entre tres roles: publicador, suscriptor o ambos en simultáneo. Un publicador, como su nombre lo indica, tiene la potestad de emitir (publicar) mensajes dentro del canal, cada vez que lo desee. Cada uno de los suscriptores recibe los mensajes publicados por los distintos clientes que tengan el rol de publicador (en otras palabras, se suscriben a los mensajes publicados en el canal).

El patrón publicar-suscribir es de gran utilidad, ya que garantiza escalabilidad, multidireccionalidad en las comunicaciones y un bajo acoplamiento entre las partes que se comunican. Existe una amplia oferta de servicios y librerías de software destinados a la administración de canales de comunicación que implementen este patrón. Algunas de las opciones más usadas son Azure Service Bus, Socket.IO, RabbitMQ y SignalR.

En F-IoT se decidió utilizar la librería SignalR [25]. Esta se proporciona de forma nativa en el framework .NET y ofrece una solución simple y eficiente para la creación de servicios que administren canales de comunicación personalizados. Cada servicio puede desplegarse en servidores on-premise o en la nube, por medio de servicios administrados como Azure SignalR Service [26].

En cuanto a la seguridad, se utiliza el mismo modelo que en el servidor REST. De esta forma, cada uno de los clientes de un servidor de SignalR deberá autenticarse previamente en el servidor REST para recibir un token JWT [19, 20]. Luego, podrá usar este para autenticarse en el servidor IO y conectarse a los canales de comunicación a los que se encuentre autorizado.

En F-IoT diferentes tipos de cliente se conectan al servidor IO, cada uno con fines distintos. Estos son los dispositivos, los agentes gateway y los usuarios:

- Los **dispositivos** se conectan a canales de comunicación para transmitir el estado interno de sus componentes, y el estado del ambiente en el que se encuentran (medido a través de sus sensores). Por otra parte, los dispositivos se suscriben a canales de comunicación para recibir información u órdenes transmitidas por otros clientes del servidor IO. Estas órdenes pueden incluir tanto la asignación de configuraciones internas como el accionamiento de los actuadores de los dispositivos.
- Los **gateways**, por su parte, se conectan a canales de comunicación en representación de aquellos dispositivos que no pueden realizarlo. Es así como cada gateway crea una conexión al servidor separada por cada dispositivo que representa. Dado que los gateways son dispositivos con mayores capacidades, es importante que transmitan su estado interno, para que puedan detectarse fallas potenciales de forma anticipada. También, los gateways pueden recibir órdenes e información externa (incluyendo actualizaciones de software). Es por ello que los gateways se conectan a canales de comunicación específicos para transmitir y recibir información personalizada, independiente de los dispositivos a los que representan.
- Finalmente, los **usuarios** se encargan de crear los canales de comunicación en los que se conectan dispositivos y gateways. Luego, los utilizan para visualizar la información transmitida y para enviar órdenes a dispositivos y gateways. Para administrar los canales de comunicación, los usuarios utilizan el Frontend de F-IoT, que será presentado en la siguiente sección.

2.4 Frontend

Para poder administrar F-IoT en su totalidad, se desarrolló una interfaz de usuario Web, a partir de React [27], una librería de JavaScript de código abierto creada por Facebook y ampliamente utilizada por la comunidad global de desarrolladores Web.

La interfaz de usuario proporciona las siguientes funcionalidades:

- Registro e inicio de sesión para usuarios.
- Administración de sistemas de IoT, dispositivos y gateways dentro de una cuenta de usuario.
- Visualización de mensajes transmitidos en canales de comunicación, y envío de mensajes personalizados con el fin de dar órdenes a dispositivos o gateways.
- Administración de perfiles de usuario creador, registro de nuevos esquemas de dispositivo, creación de nuevos dispositivos y creación de nuevas versiones de esquemas existentes.

Para desempeñar las funcionalidades mencionadas, el frontend se conecta de forma automática tanto al servidor IO como al servidor REST.

2.5 Dispositivos y gateway

El último eslabón de F-IoT, quizás uno de los más importantes, es aquel conformado por los dispositivos. Como se especificó previamente, un dispositivo o cosa es un objeto

físico integrado con electrónica, software, sensores y conectividad, que puede intercambiar información con el fabricante, operador y otros dispositivos.

Para poder conectarse a F-IoT de forma directa, todo dispositivo debe poder establecer comunicaciones con el servidor IO, de forma segura y eficiente, sin que sus características de hardware o software lo limiten. En caso de contar con alguna limitación, se recomienda utilizar un gateway como intermediario. Este último además podrá ser utilizado como dispositivo de comunicación central para múltiples dispositivos, de modo que se incrementa la eficiencia energética, se simplifica la configuración de la red local y se fortalece la seguridad en las comunicaciones.

Con el objetivo de simplificar la conexión a F-IoT, se desarrolló una aplicación gateway en .NET, diseñada para poder ser ejecutada en cualquier computadora con bajos recursos, como podría ser una Raspberry PI o una mini-PC, para convertirla en un gateway compatible con F-IoT.

3 Conclusión y trabajo futuro

Se desarrolló F-IoT, un panel de control virtual que permite crear, monitorear y comunicarse con dispositivos pertenecientes a uno o más sistemas de IoT. Cada dispositivo registrado en F-IoT puede ser desarrollado utilizando cualquier combinación de hardware y software, siempre y cuando pueda comunicarse con el servidor IO, o bien, se disponga de un gateway que lo haga.

El desarrollo de F-IoT implicó la creación de cuatro sistemas de software, que interactúan entre sí para cumplir con los objetivos de la herramienta. Cada uno de los sistemas creados utiliza herramientas gratuitas, algunas de código abierto, con el fin de que pueda ser utilizado por usuarios de la región, donde los recursos económicos para proyectos tecnológicos no siempre están al alcance de la mano. A partir de F-IoT, puede planificarse la creación de cualquier sistema de IoT, teniendo resueltos los desafíos de la selección e implementación de protocolos de comunicación y seguridad.

Como trabajo a futuro, se realizarán mejoras en la seguridad de cada parte de F-IoT, se implementarán cambios en el modelo de datos y se desarrollará una nueva interfaz de usuario. También se trabajará en la integración con Matter [28], el estándar para la comunicación entre dispositivos de IoT de diferentes fabricantes que fue publicado en 2022, bajo el respaldo de la Alianza de Estándares para la Conectividad, que integra importantes compañías y organizaciones, como Amazon, Microsoft, LG, Philips y Signify, entre otras.

Finalmente, se añadirán nuevas funcionalidades como la posibilidad de compartir sistemas de IoT entre múltiples usuarios, y de configurar sistemas que funcionen usando redes inalámbricas no convencionales.

4 Referencias

1. IBM: The Little-known story of the first IoT device (2018).
2. K. Ashton: That ‘internet of things’ thing. RFID journal, vol. 22, n° 7, pp. 97-114 (2009).
3. Oracle: ¿Qué es el IoT?, https://aws.amazon.com/es/what-is/iot/.

4. Amazon: ¿Qué es el IoT?, https://aws.amazon.com/es/what-is/iot/.
5. M. Mezzenzana: Internet-of-Things as an enabling factor for user-centered service engineering (2019), https://www.researchgate.net/publication/360256759_Internet-of-Things_as_an_enabling_factor_for_user-centered_service_engineering/.
6. ThingsCon: User Centered IoT-Design (2017), https://medium.com/the-state-of-responsible-internet-of-things-iot/andreakrajewski-aff52af1e065/.
7. Microsoft: IoT Overview, https://learn.microsoft.com/es-es/azure/architecture/framework/iot/iot-overview/.
8. Microsoft: IoT Reliability, https://learn.microsoft.com/es-es/azure/architecture/framework/iot/iot-reliability/.
9. S. N. C. Z. S. e. a. Moore: IoT reliability: a review leading to 5 key research directions (2020), https://doi.org/10.1007/s42486-020-00037-z/.
10. Entrust: Qué es la Autenticación Multifactor (MFA)?, https://www.entrust.com/es/resources/faq/what-is-multi-factor-authentication-mfa/.
11. Investopedia: What is Personally Identifiable Information (PII)? Types and Examples, https://www.investopedia.com/terms/p/personally-identifiable-information-pii.asp/.
12. V. J. Guareteguá: Información de identificación Personal (PII) / Personally Identifiable Information (PII) (2020), https://www.linkedin.com/pulse/informaci%C3%B3n-de-identificaci%C3%B3n-personal-pii-personally-javier/?originalSubdomain=es/.
13. European Union: General Data Protection Regulation (GDPR), https://gdpr-info.eu/.
14. Gobierno Nacional de la República Argentina: Ley 25.326 de la Protección de los Datos Personales (2000), http://servicios.infoleg.gob.ar/infolegInternet/anexos/60000-64999/64790/texact.htm/.
15. State of California Department of Justice: California Consumer Privacy Act (2023), https://www.oag.ca.gov/privacy/ccpa/.
16. Microsoft: SQL Server (2022), https://www.microsoft.com/es-ar/sql-server/sql-server-2022/.
17. Microsoft: .NET is open source, https://dotnet.microsoft.com/platform/open-source/.
18. Microsoft: LINQ, https://learn.microsoft.com/es-es/dotnet/csharp/linq/.
19. Auth0: Introduction to JSON Web Tokens, https://jwt.io/introduction/.
20. IETF: JSON Web Token (JWT) (2015).
21. Navarra Tecnología del Software S.L.: ¿Qué es SAML? (2020), https://www.nts-solutions.com/blog/saml-que-es.html/.
22. Wikipedia: HMAC, https://es.wikipedia.org/wiki/HMAC/.
23. SSL.com: Comparing ECDSA vs RSA (2018), https://www.ssl.com/article/comparing-ecdsa-vs-rsa/#introduction.
24. Wikipedia: Publish–subscribe pattern (2020).
25. Microsoft: Introducción a SignalR, https://learn.microsoft.com/es-es/aspnet/signalr/overview/getting-started/introduction-to-signalr/.
26. Microsoft: Azure SignalR Service, https://learn.microsoft.com/es-es/azure/azure-signalr/signalr-overview/.
27. React: React, https://react.dev/.
28. The Connectivity Standards Alliance: Smart Home Innovation Set To Accelerate With Matter (2022), https://csa-iot.org/newsroom/smart-home-innovation-set-to-accelerate-with-matter/.