

Tecnologías Emergentes

Tema 07. Internet de las cosas (IoT) Parte II

Índice

Esquema

Ideas clave

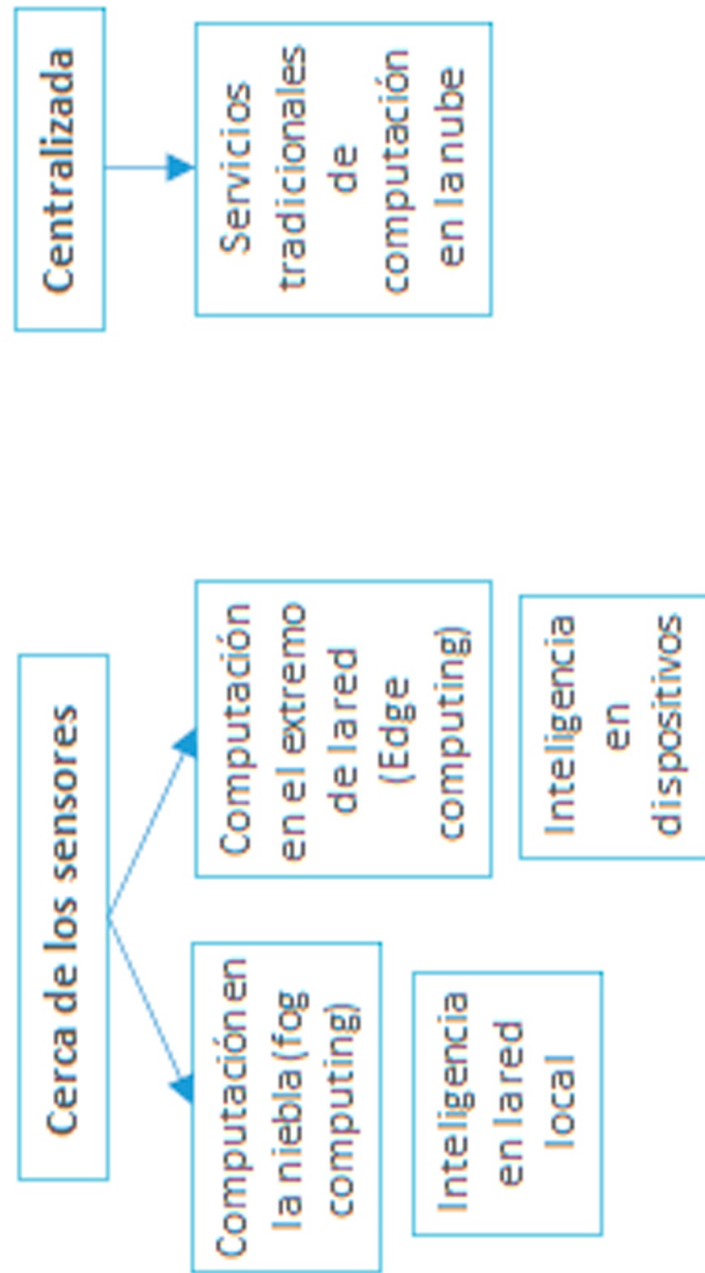
- 7.1. ¿Cómo estudiar este tema?
- 7.2. Más allá de la computación en la nube
- 7.3. Ciudades inteligentes (smart cities)
- 7.4. Gemelos digitales
- 7.5. Proyectos IoT

A fondo

- Comunicaciones en ciudades inteligentes
- Edge Computing: qué es y por qué hay gente que piensa que es el futuro
- Las ciudades inteligentes son más eficientes
- 2030: smart city life 360 view
- WIP: why limiting work in progress makes sense (Kanban)
- PMBOK® Guide Processes Flow
- Plan Nacional de Ciudades Inteligentes

Test

Modelos de computación para IoT



7.1. ¿Cómo estudiar este tema?

Para estudiar este tema, lee las ideas clave disponibles a continuación. Los **objetivos** a conseguir son:

- ▶ **Modelos de computación específicos para IoT:** cuando existen millones de dispositivos sensorizados que transmiten información, el modelo tradicional de computación en la nube puede producir tiempos de latencia elevados, lo que hace más recomendable procesar los datos cerca de la fuente, y para ello se estudiará la computación en el extremo de la red (*Edge Computing*) y la computación en la niebla (*Fog Computing*).
- ▶ **Las ciudades inteligentes o *smart cities*:** como aglutinador de muy diversas tecnologías, relacionadas con el IoT y la sensorización de elementos urbanos, gestionables de forma centralizada.
- ▶ Los **gemelos digitales:** como nexo de unión entre el mundo físico y el virtual en procesos de fabricación de productos, procesos y servicios, que se realimentan con datos reales del funcionamiento físico de los mismos.
- ▶ La **gestión de proyectos de IoT**, como caso específico dentro del ámbito de la gestión de proyectos de Ingeniería del Software.

7.2. Más allá de la computación en la nube

Son ya sobradamente conocidas las ventajas de la tecnología *cloud computing* (computación en la nube), en términos de flexibilidad, escalabilidad, seguridad, acceso simultáneo, etc.

Sin embargo, este modelo centralizado —en los servidores que conforman la nube— suele tener el hándicap de tiempos de latencia no precisamente cortos. En términos generales la tecnología IoT, normalmente asociada a modelos *cloud*, está generando enormes volúmenes de datos, y cuando llegan a la nube para ser sometidos a análisis, normalmente es ya muy tarde para actuar.

Es por ello que están apareciendo arquitecturas complementarias a la «tradicional» computación en la nube, sobre todo orientadas a reducir los actualmente elevados tiempos de latencia.

La arquitectura *Edge Computing*, o «computación al extremo (de la red)» se basa en traer el procesamiento cerca de la fuente de datos, por lo que no es necesario enviarlos a una nube u otros servidores centralizados para procesarlos. Esto permite mejorar enormemente la velocidad y rendimiento el transporte de datos.

Esta arquitectura es muy adecuada para diversas situaciones, como por ejemplo dispositivos IoT con conectividad pobre, y que por tanto no tienen facilidad para el acceso a una nube central.

Es importante no centrar la atención en la ubicación física de los dispositivos. La división entre lo que está en el núcleo o «*core*» y lo que se encuentra en el borde de la red tiene que ver más bien con dónde los ubicaríamos en un diagrama de red.

Desde luego que esto no es obvio en ocasiones. Por ejemplo, los ordenadores PC de sobremesa normalmente son dispositivos del núcleo, pero si el negocio de la

empresa se basa en la nube, probablemente sean dispositivos en el extremo de la red.

Lo cierto es que reducir a solamente dos categorías los dispositivos ayuda enormemente a identificar elementos de estrés o congestión y a planificar mejoras.

La computación en la niebla («*Fog Computing*») es una estructura de computación descentralizada, que en muchos sentidos está emergiendo como la nueva generación de la computación en la nube. En la computación en la niebla, tanto el almacenamiento como el procesamiento de los datos se distribuyen de la forma más lógica y eficiente entre la fuente de datos y la nube.

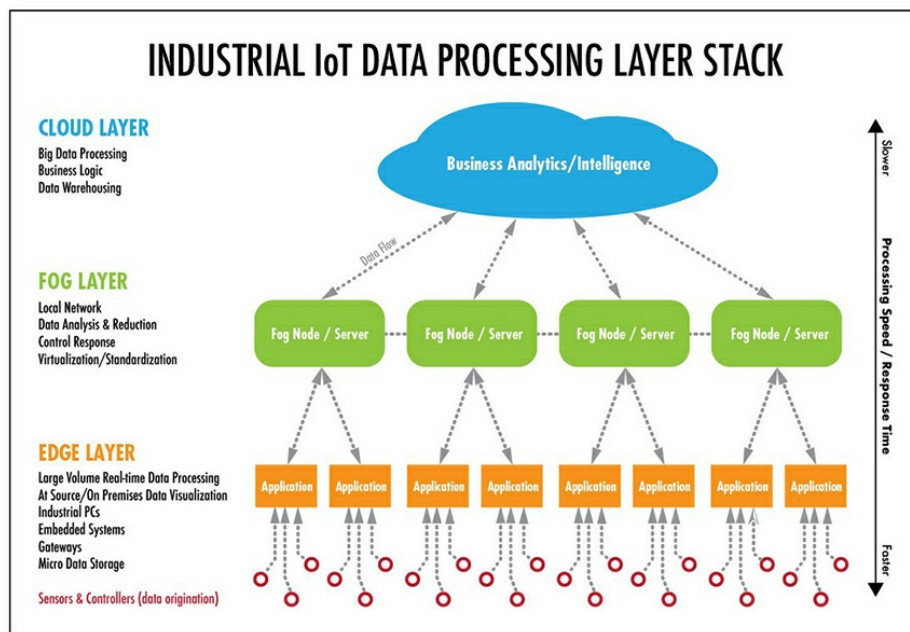


Figura 1. Los distintos tipos de computación en un modelo por capas para IoT Industrial. Fuente: winsystems.com

Tanto la computación en la niebla como la computación en el borde tratan de hacer frente al enorme incremento de ancho de banda requerido por los dispositivos finales. Ambos intentan de alguna forma «acercar» los centros de datos de la nube para minimizar latencias, costes e incrementar la calidad de operaciones.

La principal diferencia técnica entre ambas está en el lugar en el que finalmente se hace el procesamiento. La computación en la niebla desplaza la inteligencia de validación de datos a la red local, mientras que la computación en el extremo lo hace en los propios dispositivos.

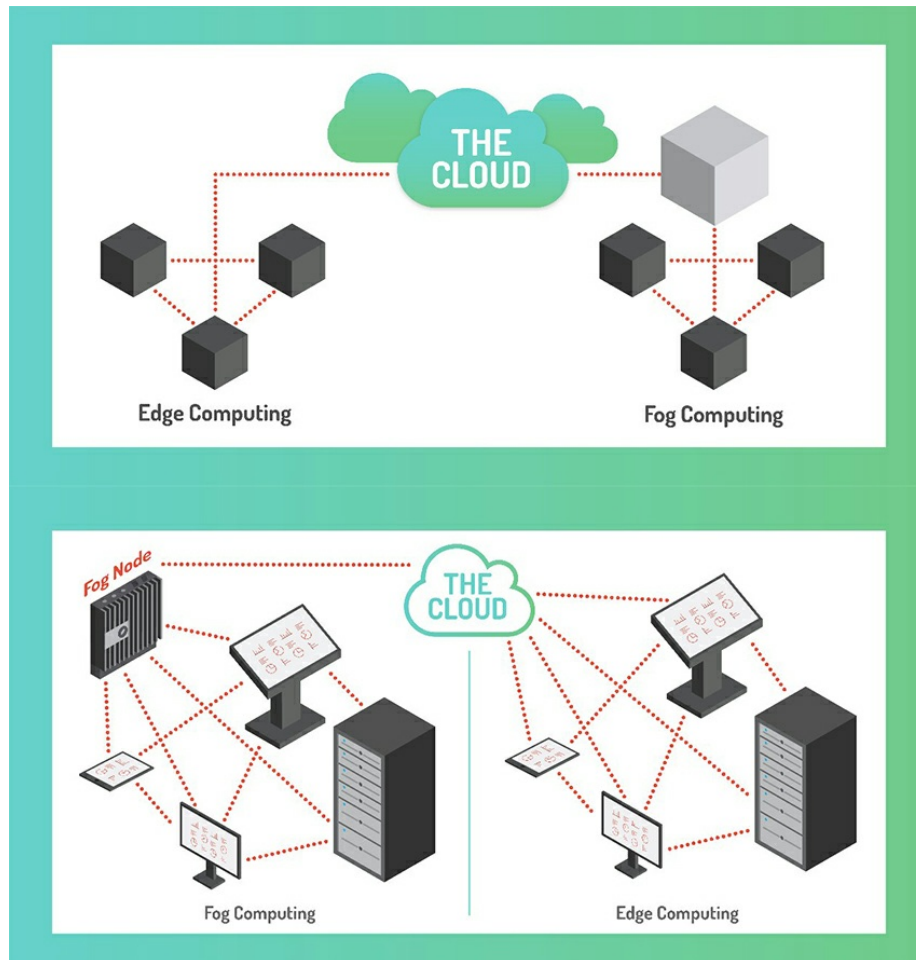


Figura 2. Tomadas aisladamente, se aprecian las diferencias entre ambas arquitecturas. Fuente: forestgiant.com

En el ámbito de servicios móviles, actualmente se está trabajando en arquitecturas de computación en el extremo intentando sacar el máximo partido de la tecnología 5G. Puesto que esta introduce tiempos de latencia muy reducidos, de 1ms, tiene sentido colocar nodos de procesamiento y almacenamiento conectados en la red de acceso radio (cerca de los elementos de gestión de antenas).

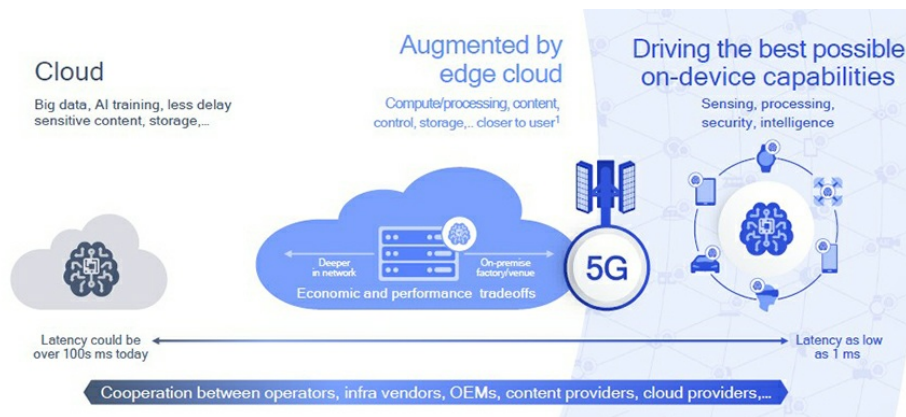


Figura 3. Comunicación entre dispositivos finales y la red 5G para el acceso a servidores de procesamiento en el extremo de la red. Fuente: Qualcomm.com

En la figura 4 se puede apreciar que toda la potencia de procesamiento se coloca en una infraestructura de tipo «nube en el extremo», lo que permite posicionar como primerísimos actores y con posición privilegiada a los distintos operadores móviles con red de acceso radio propia (no operadores móviles virtuales) para ofrecer estos servicios a terceros.

Si además los operadores, entre su cartera de servicios, disponen de comunicaciones LPWAN para la conexión con dispositivos IoT, se cierra el círculo para ofrecer infraestructuras completas de gestión de grandes volúmenes de dispositivos IoT.

7.3. Ciudades inteligentes (smart cities)

El concepto de *smartcity* o ciudad inteligente ejerce una gran atracción, por no decir fascinación, en muchos ámbitos, fundamentalmente porque en él convergen otras grandes tecnologías del momento: IoT, *e-Mobility*, *Data Science*, etc.

En realidad, se trata de un término bastante glamuroso pero que responde a una cuestión bastante sencilla: se ocupa de las distintas técnicas que pueden usar las ciudades para realizar su transformación digital, con el fin de llegar a ser más acogedoras, más eficientes, más respetuosas con el medio ambiente, más centradas en el ciudadano, y como consecuencia de todo ello, más competitivas.

Aunque pareciera que el centro de todo es la tecnología, en realidad una de las partes más importantes de una *smart city* es la colaboración de una ciudadanía comprometida. Es gracias a la tecnología por lo que los ciudadanos pueden ofrecer una retroalimentación continua a los regidores de la ciudad.

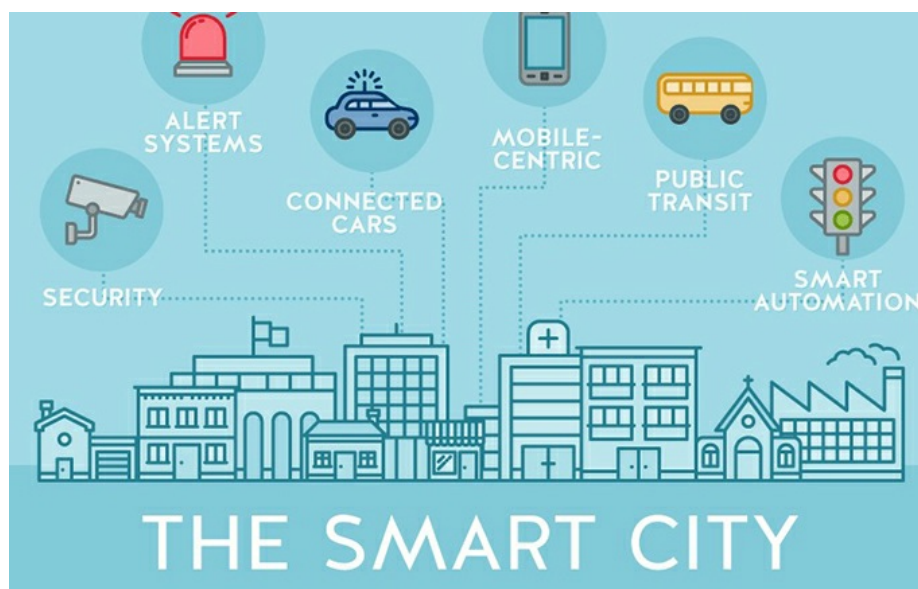


Figura 4. Componentes de una ciudad inteligente. Fuente: Philips.com

En efecto, los ciudadanos, haciendo uso de la tecnología (por ejemplo, mediante una simple app en su dispositivo móvil) pueden informar sobre socavones en la calzada, desperfectos en mobiliario urbano, semáforos estropeados e incluso residuos y basuras que no se han recogido. Es esta colaboración ciudadana lo que ha dado lugar a que algunas fuentes prefieran hablar de «*crowdsourced city*» (o ciudad de colaboración abierta distribuida), más que de *smart city*.

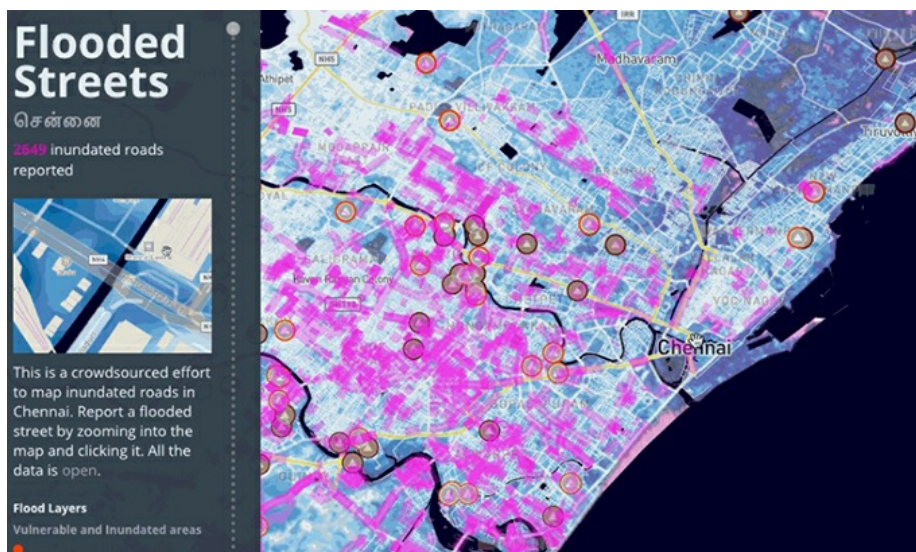


Figura 5. Ejemplo de colaboración abierta distribuida ciudadana para informar sobre inundaciones en tiempo real. Fuente: mapbox.com

Un asunto relevante, que se está aún debatiendo en muchas de las implementaciones piloto de *smart cities* es hasta qué punto se deben extraer y procesar datos. Las cuestiones de la privacidad y soberanía de los datos se deben abordar cuidadosamente.

En efecto, puede haber pérdida de control para los ciudadanos de una Smart City si la tecnología hace posible que distintas cámaras y dispositivos sensores puedan seguir sus pasos vayan donde vayan, y si los algoritmos influyen su vida diaria.

Lo cierto es que una *smart city* puede, como casi todas las tecnologías, gestionarse para bien o de otras formas, haciendo de ella un aparato de control. La

monitorización de los transportes públicos, del uso de los aparcamientos con los parquímetros que obligan a introducir matrículas, los sistemas de telemetría para lectura de consumo instantáneo de suministros en viviendas...todo ello se puede usar potencialmente para propósitos de vigilancia masiva de la ciudadanía.

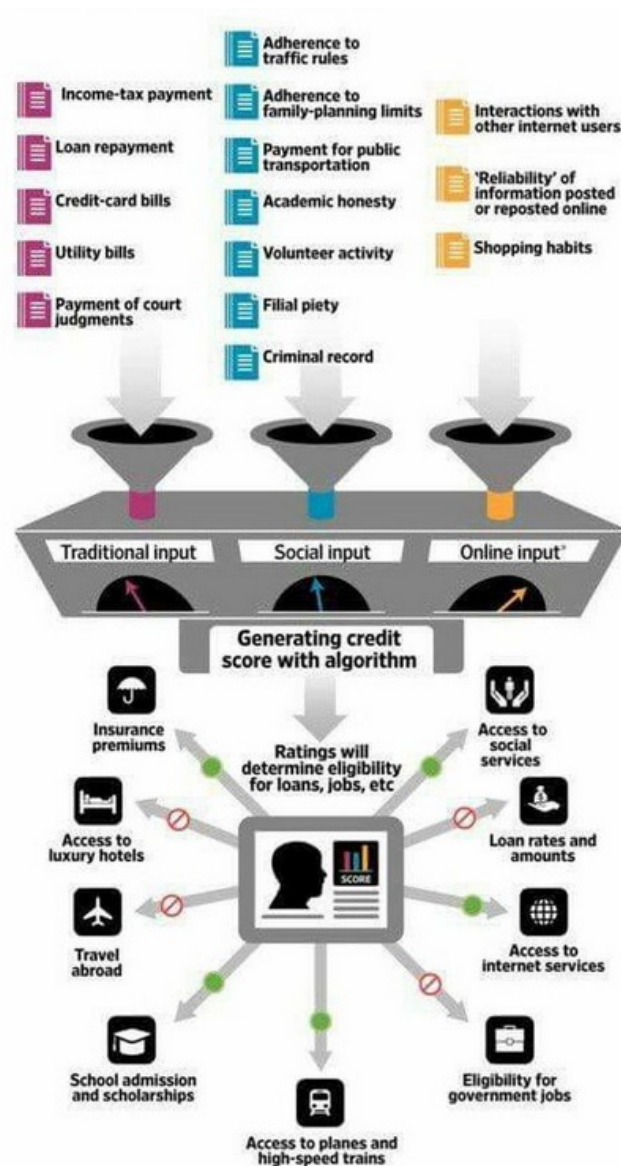


Figura 6. Funcionamiento del «Crédito Social» para ciudadanos de China. Fuente: The Wall Street Journal.

7.4. Gemelos digitales

Un gemelo digital («*digital twin*») es una réplica digital de un producto, proceso o sistema físico, gracias a información que se obtiene a partir de sensores, y que se puede utilizar para muchos propósitos. La representación digital de un objeto proporciona información precisa sobre sus elementos como la dinámica de cómo funciona a lo largo de su ciclo de vida.

Un gemelo digital pretende ser una réplica actualizada y exacta de todos los elementos de un objeto físico de los que se dispone de información obtenida mediante sensores.

Esta simbiosis **del mundo físico con el virtual** hace posible tanto la aplicación de analítica de datos como de monitorización con el objeto de:

- ▶ Anticiparse a posibles problemas o anomalías operativas antes de que lleguen a producirse.
- ▶ Realizar pruebas de posibles modificaciones para ofrecer productos nuevos.
- ▶ Simular el funcionamiento tras ajustes personalizados a petición de los clientes.

Los gemelos digitales aglutinan diversas tecnologías, como la **inteligencia artificial**, el **aprendizaje automático** («*machine learning*»), **analítica predictiva** y **telemetría de sensores** para crear tanto clones digitales con información del rendimiento actual e histórico de máquinas físicas, como modelos de simulación digital ideales que van evolucionando con los datos de sus homólogos del mundo real.

Lo cierto es que, en las implantaciones realizadas hasta la fecha del concepto de gemelo digital, no solo en grandes plantas de fabricación industrial, sino también en empresas de tamaños más modestos, se llega a **reducir el tiempo de puesta en mercado** («*time to market*») casi a la mitad, y se han registrado **incrementos de**

productividad de casi un 20 %, y eso se traduce en un incremento claro de la competitividad de las empresas.

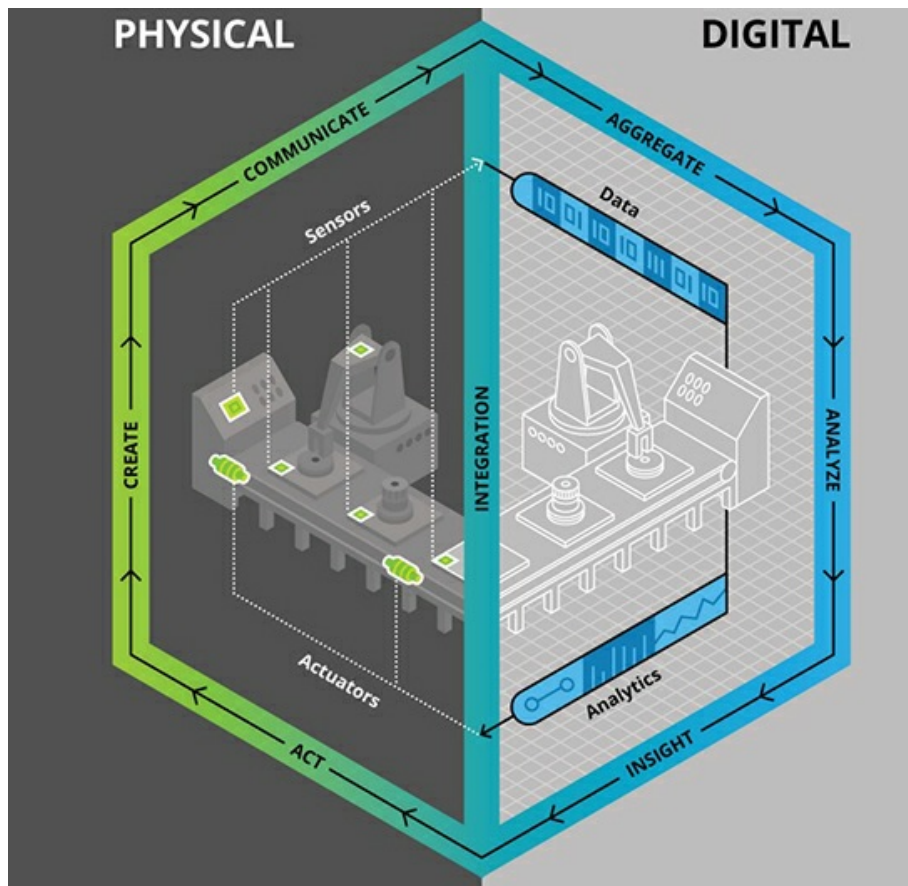


Figura 7. Modelo de gemelo digital en proceso de fabricación. Fuente: Deloitte.com

Es importante hacer hincapié en una cuestión esencial: **para poder generar un gemelo digital es necesario que existan datos reales** procedentes del homólogo físico, si no, obviamente no se puede construir el gemelo digital.

La mecánica es la siguiente: los sensores del producto, proceso o servicio obtienen datos en tiempo real sobre el estado actual del mismo. Toda esta información se envía a un sistema que puede ser local o basado en la nube, donde se procesa adecuadamente, comparando con información anterior y con modelos de funcionamiento ideal. Finalmente, cuando concluye el análisis realizado en el

sistema, se pasa a generar el gemelo digital del producto o servicio.

Es en este momento cuando se pueden probar distintos cambios o ajustes, para ver los efectos, antes de pasar a la fabricación. **únicamente cuando en el gemelo digital se pueden validar de forma virtual todas las modificaciones realizadas, se pasa a la producción física.**

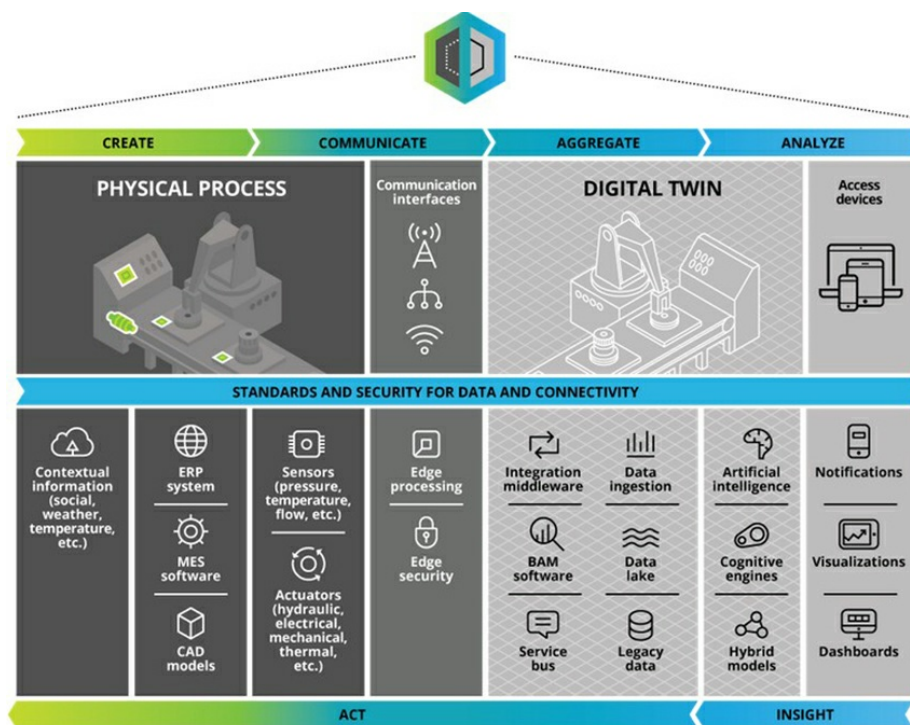


Figura 8. Arquitectura conceptual de Gemelo Digital. Fuente: Deloitte.com

Los gemelos digitales se usan hoy día en muy diversas industrias, como en fabricación, energía, transporte y construcción. Los tipos de productos que normalmente están involucrados en gemelos digitales incluyen motores de aviación, locomotoras de gran potencia para trenes de alta velocidad, plataformas petrolíferas, turbinas eólicas...etc.

7.5. Proyectos IoT

Un proyecto es un **esfuerzo temporal** emprendido para crear un **único producto o servicio**. En función de proyectos se puede trabajar en el mundo editorial, construcción, investigación, etc. Se debe distinguir claramente los proyectos de las operaciones. Hacer una nueva carretera es un proyecto. Mantener una carretera en las condiciones adecuadas para que sea transitable forma parte de las operaciones de mantenimiento. Las **operaciones** finalizan cuando finaliza la vida del producto o servicio en cuestión. El proyecto finaliza cuando se ha creado el producto o servicio y, normalmente, entregado al cliente.

La **gestión de proyectos** es la aplicación de conocimientos, habilidades, herramientas y técnicas para proyectar actividades destinadas a satisfacer las necesidades y expectativas de los interesados (*stakeholders*) de un proyecto. Entre estos interesados podemos encontrar además de los propios clientes o usuarios a otros actores como directivos, servicios jurídicos de la compañía, departamento de seguridad, departamento de marketing, entidades reguladoras, legisladores, gobiernos, grupos de presión, etc.

¿Qué tipo de producto o servicio se espera de un proyecto de IoT? Quizá estemos pensando en un sistema centralizado de gestión de flotas, la sensorización de una planta agrícola con un cuadro de mando que muestre el estado de distintos indicadores en tiempo real, etc. Todos esos productos tienen algo en común: software. El software es el producto más habitual que un proyecto de estas características genera. Por supuesto, en el caso de IoT el software viene acompañado de hardware con sensores o actuadores que ejecutan las órdenes emitidas por el software.

El ciclo de vida del software

Se considera ciclo de vida de un producto software a la **evolución del producto desde su concepción hasta su retirada**. Las actividades del ciclo de vida deben ser separadas en **fases**.

Siendo más explícito, el **ciclo de vida** del software es el marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo la explotación y el mantenimiento de un producto de software abarcando la vida del sistema desde la definición de requisitos hasta la finalización de su uso.

Los modelos de ciclo de vida han evolucionado desde 1968. Esa evolución se ha dirigido a mitigar los riesgos y a conseguir una mejor adaptación del producto a lo que realmente se espera de él. En los inicios, el **ciclo de vida en cascada** era la principal opción.

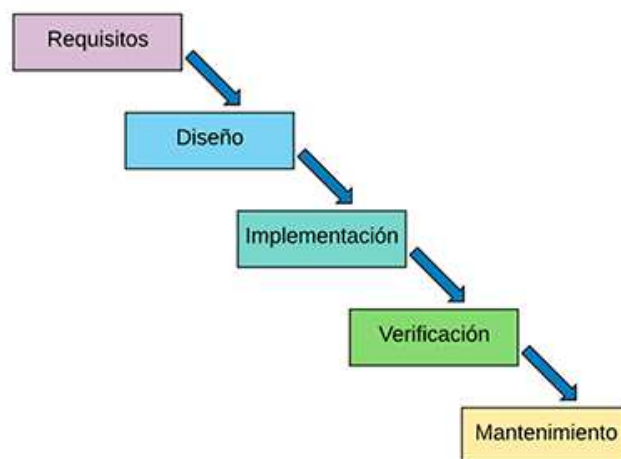


Figura 9. Ciclo de vida en cascada. Fuente: <https://openclassrooms.com/courses/gestiona-tu-proyecto-de-desarrollo/en-que-consiste-el-modelo-en-cascada>

Este modelo presentaba serios problemas ya que la fase de verificación tenía lugar al final del proceso. En caso de detectar errores en la implementación, en el diseño o

en la propia toma de requisitos (que determina los objetivos a conseguir) el camino a desandar implica unos costes más que considerables en tiempo y dinero.

El **modelo del ciclo de vida en espiral** desarrollado por Boehm en 1986 introdujo elementos importantes al incorporar una completa gestión del riesgo y control de la calidad. El desarrollo iterativo permite avanzar gradualmente hacia el producto final mitigando los riesgos existentes.

Este modelo da participación continua al cliente evaluando el producto de forma constante, de esta forma, los errores se pueden detectar de forma anticipada. Las distintas iteraciones permiten generar diferentes prototipos del producto final. Este modelo exige equipos muy preparados y un seguimiento continuo y detallado.

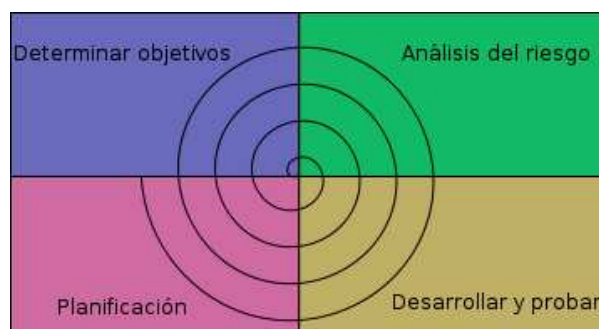


Figura 10. Ciclo de vida en espiral. Fuente: [https://upload.wikimedia.org/wikipedia/commons/thumb/3/39/ModeloEspiral.svg/359px-ModeloEspiral.svg.png?](https://upload.wikimedia.org/wikipedia/commons/thumb/3/39/ModeloEspiral.svg/359px-ModeloEspiral.svg.png?x-ModeloEspiral.svg.png)

El **modelo del ciclo de vida iterativo e incremental** es uno de los más usados hoy en día. Con este modelo se persigue que el cliente reciba de forma continua algo de valor. Según se avanza en el número de iteraciones el producto final incorpora más funcionalidades y se convierte en un producto más elaborado y complejo. Este modelo permite acometer el producto en base a pequeños entregables que hacen más realizable la tarea.

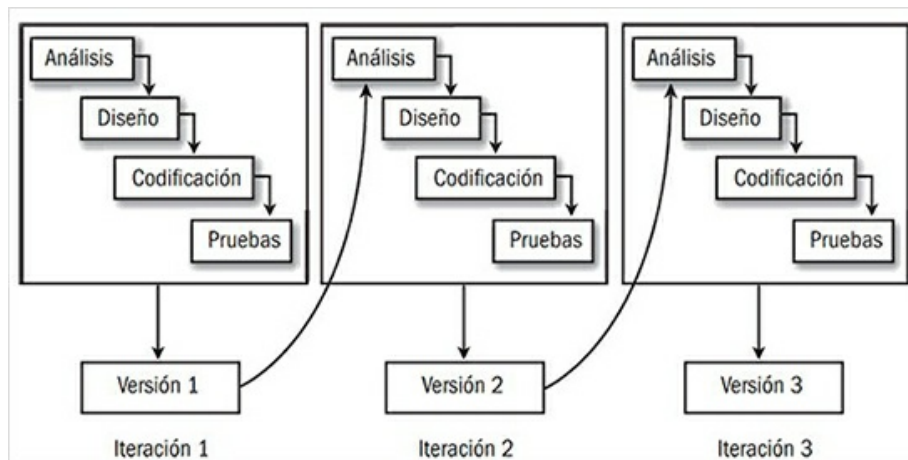


Figura 11. Ciclo de vida iterativo e incremental. Fuente: http://3.bp.blogspot.com/-6HiwEBD3k1A/T5YoY36dtZI/AAAAAAAAAEk/EQ_Y5eVPo7Y/s1600/iterativo.png

Antes de iniciar el proyecto no solo debe estar claro el ciclo de vida escogido sino también la metodología seleccionada.

Metodologías de desarrollo de software

Una metodología es un conjunto de procedimientos que permiten producir y mantener un producto software. La metodología define cuáles son las fases del ciclo de vida del software de las que se va a ocupar, definiendo para cada fase las actividades, tareas, los objetivos correspondientes y los participantes involucrados en cada tarea.

Para llevar a cabo las tareas, las metodologías utilizan técnicas que describen cómo llevarlas a cabo. Las técnicas suelen ser gráficas con apoyos textuales formales, lo que mejora una descripción sencilla, detallada y comprensible para el usuario. Las reglas de notación y sintaxis de cada técnica se dan con la guía de la metodología. Existen muchas y diversas metodologías, **MÉTRICA 3**, *Structured Systems Analysis and Design Methodology (SSADM)*, *Rational Unified Process (RUP)*, *Extreme Programming (XP)*, *PRjects IN Controlled Environments 2 (PRINCE2)*, *International Project Management Association (IPMA)*, la norma **ISO 21500:2012**, etc.

Por ejemplo, un paso crucial en cualquier proyecto es la **definición de requisitos**. Es esta una tarea nada trivial donde es esencial detectar objetivos y las suposiciones de partida de forma precisa y rigurosa. Es muy habitual en este contexto la imagen siguiente explicando el tipo de confusiones a las que puede dar lugar no manejar un lenguaje y alcance compartido del proyecto.

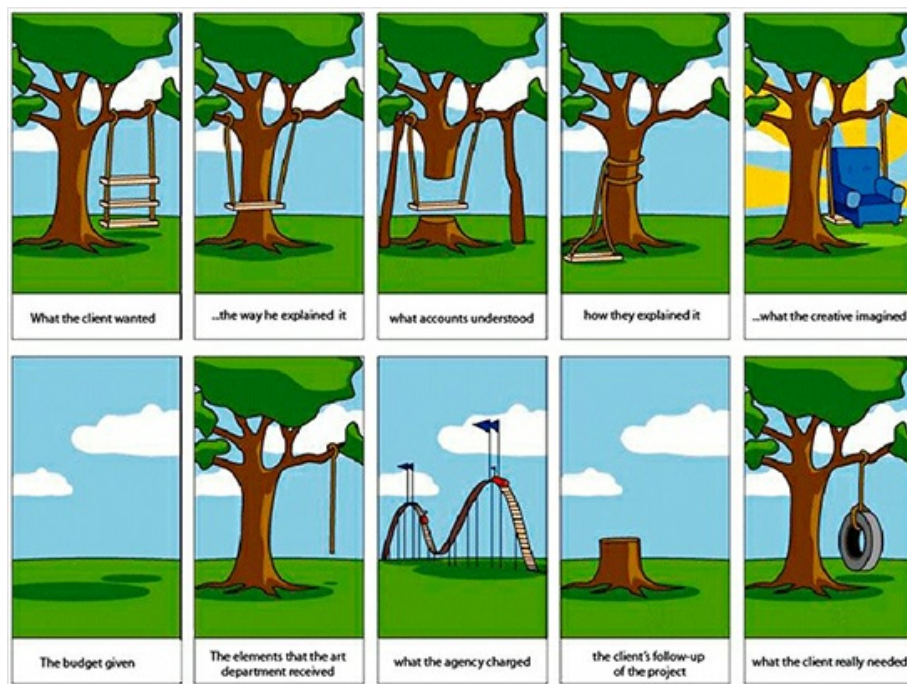


Figura 12. Requisitos de un proyecto de software, confusiones. Fuente:

<https://tavoalencia.files.wordpress.com/2010/08/creative-process.png>

Sin duda alguna, una de las instituciones que más ha aportado a la estandarización de la práctica de dirección de proyectos es el **Project Management Institute (PMI)**. Se trata de una la asociación profesional sin fines de lucro a nivel mundial que tiene como misión promover el desarrollo del conocimiento y competencias básicas para el ejercicio profesional de la dirección de proyectos. Ofrece varias certificaciones siendo la certificación **Project Management Professional (PMP)** una de las más solicitadas. Los conocimientos que requiere esta certificación están recogidos en el **Project Management Body of Knowledge (PMBOK)**.

El PMBOK no se trata una metodología en sí sino de una guía de estándares internacionales para que los usuarios puedan adaptar a cada caso y contexto particular los procesos, reconocidos como **buenas prácticas** por el PMI. El PMBOK es el marco de referencia formal para desarrollar proyectos y alcanzar los objetivos. En septiembre de 2017 se publicó la sexta versión de este, actualmente vigente.

En el apartado A fondo podrás encontrar un enlace para una mayor explicación gráfica de los distintos procesos.

Gestión ágil de proyectos

Las metodologías ágiles permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno. Se persigue de esta forma **reducir costes e incrementar la productividad**, al avisar rápidamente de los errores y problemas encontrados. Entre las metodologías ágiles más aplicadas hoy en día tenemos **Scrum**, **Kanban**, **Scrumban** y **Extreme Programming (XP)**. Se comentará muy brevemente *Scrum* y *Kanban* por ser metodologías de uso habitual en los proyectos que nos ocupan.

Las metodologías ágiles suelen ser útiles en proyectos donde los requisitos no están claros o son muy cambiantes. En este tipo de proyectos, el prototipado y la investigación continua marcan el camino a seguir.

Scrum fue desarrollada por Ikujiro Nonaka e Hirotaka Takeuchi a principios de los 80, analizando cómo desarrollaban los nuevos productos destacadas empresas de manufactura tecnológica como Fuji-Xerox, Canon, Honda, etc. El término *Scrum* proviene de la figura del rugby del mismo nombre y se escogió por sus implicaciones a la hora de referirse al trabajo en equipo.



Figura 13. Scrum de rugby. Fuente:

https://upload.wikimedia.org/wikipedia/commons/thumb/5/5b/2014_Women%27s_Six_Nations_Champions_hip_-_France_Italy_%2851%29.jpg/300px-2014_Women%27s_Six_Nations_Championship_-_France_Italy_%2851%29.jpg?

Scrum se caracteriza por la planificación en base a iteraciones (*sprints*), con una duración de entre 2 y 4 semanas. Cada *sprint* genera un entregable para el cliente. Las necesidades o requisitos del cliente se anotan en una «pila» de producto de forma priorizada. Esta lista será creada y gestionada por el cliente con la ayuda del equipo.

La lista de los trabajos que debe realizar el equipo durante el *sprint* se añade a la pila del *sprint* y se selecciona una persona responsable de su ejecución. Así el proyecto se descompone en unidades más pequeñas y gestionables.

En ocasiones, la pila del *sprint* acumula un número excesivo de tareas para que el equipo pueda abarcarlas con eficiencia. Kanban es otra metodología que trata de solucionar esos excesos.

Kanban nace en Japón alrededor del año 1950 como una propuesta para manejar de forma eficiente el flujo de materiales en una línea de producción. Algunos de sus componentes principales son:

- La visualización del flujo de trabajo: dividiendo el trabajo en bloques y escribiendo cada elemento en una tarjeta para después ponerlo en la tabla o pizarra creada para ello.

- ▶ La limitación del trabajo en curso (Work In Progress – WIP) asignando límites concretos del número de elementos que pueden estar en progreso en cada estado del flujo trabajo.
- ▶ La medición del denominado tiempo de ciclo que es el tiempo medio para completar un elemento y también ayuda a optimizar el proceso para conseguir que este tiempo sea lo más pequeño y predecible posible.

El siguiente puede ser un ejemplo de tablero Kanban:




















Backlog (4)	Análisis (3)		Construcción (3)		Revisión (4)		Terminado
	En curso	Hecho	En curso	Hecho	En curso	Hecho	
							
							
							
							

Figura 14. Tablero Kanban. Fuente: <http://www.scrummanager.net/blog/2013/09/kanban-las-4-cosas/>

El control visual del proceso y la limitación del número de trabajos en curso contribuye a un entendimiento más completo del estado del proyecto y permite que las tareas avancen de forma eficiente.

Entornos de pruebas y entornos de producción

Para que un grupo de trabajo de un proyecto pueda realizar su trabajo necesita de una infraestructura informática con determinadas capacidades, tanto de procesamiento como de almacenamiento de información.

En el ámbito de proyectos reales, es habitual contar -como mínimo- con **dos infraestructuras diferenciadas**, un entorno de análisis más

adecuado para **desarrollo y pruebas, y un entorno de producción**
donde reside el producto final para su interacción con los usuarios.

Pongamos por caso el siguiente escenario, un equipo de proyecto desarrolla una aplicación para la gestión de plazas libres de aparcamiento en la calle, usando sensores robustecidos en la parte de la calzada reservada a vehículos aparcados, y con transmisión mediante tecnologías LPWAN.

Cuando el equipo de desarrollo trabaja, lo hace en el entorno de desarrollo. Una vez finalizado el trabajo, en ese mismo entorno se hacen las pruebas pertinentes, seleccionando para ello por ejemplo cinco calles con tráfico significativo y con un elevado número de plazas de aparcamiento.

Una vez realizadas las pruebas y estando todo perfectamente depurado, el software pasaría al entorno de producción, que puede contemplar amplias redes de sensores de varias ciudades diferentes.

Evidentemente, cada proyecto de IoT tiene su tipología e idiosincrasia particular, pero es importante señalar que se debe ser muy exhaustivo examinando el comportamiento del sistema en producción, máxime considerando que hay usuarios reales con interacciones reales, que podemos examinar y que con bastante seguridad permitirán aflorar situaciones —por ejemplo, de seguridad— que no estaban inicialmente contempladas.

Estimación del esfuerzo y recursos necesarios para el diseño y desarrollo en cada una de las fases del ciclo de vida de un proyecto de IoT

Uno de los mayores retos para el director de proyectos en cualquier ámbito es estimar las tres principales líneas base del proyecto:

- **Alcance:** detallando las funcionalidades a alcanzar y los servicios a ofrecer.

- ▶ **Tiempo:** estimar el marco temporal que guiará el proyecto.
- ▶ **Coste:** estimar los costes que ocasionará el proyecto. Implica estimar recursos teniendo en cuenta todos los objetivos del trabajo.

Dado que esta tarea no es en absoluto trivial, las metodologías ágiles se están imponiendo como un estándar frecuente dado que permiten la descomposición del proyecto en bloques realistas, iterativos e incrementales. No obstante, procede sacar a relucir ciertos costes ocultos que suelen ser frecuentes en este tipo de proyectos.



Figura 15. Enfoque tradicional versus enfoque ágil. Fuente:

<http://www.pmoinformatica.com/2017/12/pmbok-6-metodologias-agiles.html>

Los factores reflejados a continuación incrementan los costes económicos del proyecto y reducen el grado de innovación en el mismo:

- ▶ **Complejidad estructural:** a medida que la solución propuesta se vuelve más compleja la velocidad de desarrollo disminuye, cada nuevo componente requiere el análisis de dependencias adicionales, la detección de errores se convierte en un laberinto, se requiere personal más cualificado para el desarrollo, etc.
- ▶ **Complejidad del código:** no solo la proliferación de múltiples módulos interrelacionados entre sí incrementa la complejidad de la solución. Según crecen las líneas de código aumenta el esfuerzo necesario para validar y gestionar toda la

codificación. El uso de librerías externas y código desarrollado por terceros puede insertar importantes capas de ineficiencias.

- **Cambios en el entorno externo:** la experiencia demuestra que el mundo externo es inestable por naturaleza, presentando cambios continuos de forma notable. En este sentido, es un **gran error entregar un modelo al cliente o usuario final y olvidarnos de él**. Es preciso revisar de forma más o menos constante según el caso, los resultados que genera y comprobar que sigue alineado con los objetivos iniciales. Estas acciones generan un coste que debe ser considerado de antemano.

Precisamente debido a la dificultad para precisar de forma exacta y rigurosa los costes económicos y temporales del proyecto, un enfoque frecuentemente adaptado consiste en llegar a la mejor solución posible prefijando de antemano un plazo de tiempo y presupuesto inicial. Acometiendo entregas iterativas y una gestión de proyectos ágil se intenta llevar a la mejor versión posible cumpliendo con las restricciones iniciales. Llegado el momento y en función de los resultados obtenidos se determina si procede la asignación de una nueva partida presupuestaria y nuevo límite temporal para la siguiente versión.

Gestión de equipos de trabajo

Los expertos en el área de IoT no abundan en el mercado. El perfil más común acumula en su currículum una carrera técnica o científica, experiencia en desarrollo, y posiblemente un máster especializado en gestión de dispositivos remotos y comunicaciones.

Un factor intrínseco a estos profesionales es la **renovación continua de conocimientos y capacidades**. En un entorno tan innovador y cambiante es pura ilusión pensar que un conjunto de conocimientos estáticos adquiridos en un momento concreto servirá para toda la vida.

Evidentemente, existe un bloque común de conocimientos del que partirá el resto de

las competencias. Sin embargo, la formación continua es una realidad ineludible en este tipo de perfiles.

Gestionar estos equipos suele implicar habilidades para liderar un equipo de alto rendimiento. Este tipo de equipos suele demandar autonomía, *feedback* preciso y explicación clara y rigurosa de lo que se espera de ellos. Las metodologías ágiles facilitan estos elementos.

Detallar de forma rigurosa técnicas efectivas de gestión de equipos escapa del ámbito de la asignatura por demandar competencias no contempladas en el programa. Sin embargo, sí que es necesario conocer las características de algunos perfiles que suelen estar involucrados en este tipo de proyectos:

- ▶ **Analista de negocio:** perfil con formación variada y conocimiento concreto del negocio. No suele ser un experto técnico, pero sí comprende las necesidades de los clientes y usuarios y conoce las tendencias del mercado. Pero los perfiles más científicos suele ser un elemento clave en el que apoyarse para entender las características que debe tener el producto final en un entorno empresarial.
- ▶ **Jefe de proyecto:** persona con formación específica en el área y en metodologías ágiles de gestión de proyectos que se responsabiliza de la gestión de equipo, gestiona las expectativas de los interesados y controla la calidad del producto final.
- ▶ **Arquitecto de sistemas:** perfil responsable de proporcionar una infraestructura, normalmente basada en entornos *cloud*, que almacena toda la información y proporciona capacidad de cómputo de forma escalable y tolerante a fallos. Es el encargado de diseñar los recursos técnicos adecuados para el desarrollo del proyecto.
- ▶ **Ingeniero de desarrollo:** sobre la infraestructura diseñada principalmente por el arquitecto de sistemas, desarrolla software eficiente de diversa complejidad que procesa datos cumpliendo con los objetivos del proyecto.

- ▶ **Experto en IoT:** profesional con altas capacidades analíticas y conocimiento experto en gestión de dispositivos IoT (que integran sensores y/o actuadores) y comunicaciones con la infraestructura. Suele diseñar la arquitectura global software/hardware que debe tener la aplicación.

La lista anterior está centrada en perfiles más relacionados con el aspecto más puramente técnico y volcado en el tratamiento de la información. Por supuesto, no se trata de una lista cerrada ya que de forma frecuente perfiles más centrados en cuestiones de diseño y experiencia de usuario (*UX, User Experience*), por ejemplo, están teniendo un papel muy relevante en este tipo de proyectos. Estos expertos tienen la misión de generar una experiencia de usuario gratificante y ejemplar, factor que suele ser una parte muy importante del éxito de un proyecto.

Comunicaciones en ciudades inteligentes

En esta lección magistral se mostrarán los tipos de comunicaciones precisos para la gestión de las distintas redes de elementos que conforman una ciudad inteligente.

Edge Computing: qué es y por qué hay gente que piensa que es el futuro

Pastor, J. (18 de diciembre de 2018). Edge Computing: qué es y por qué hay gente que piensa que es el futuro. Xataka [Mensaje en un blog]. <https://www.xataka.com/internet-of-things/edge-computing-que-es-y-por-que-hay-gente-que-piensa-que-es-el-futuro>

Interesante artículo en el que se ilustra la importancia del *Edge Computing*, y se compara con *Fog Computing*, expresando finalmente la relación que ambos conceptos guardan con el *Cloud Computing*.

Las ciudades inteligentes son más eficientes

Benito, A. (1 de febrero de 2018). Las ciudades inteligentes son más eficientes. *Geographica*. [Mensaje en un blog]. <https://geographica.gs/es/blog/las-ciudades-inteligentes-mas-eficientes/>

Artículo en el que se detallan los incrementos de eficiencia en distintos aspectos que se logran cuando las ciudades son inteligentes. Es interesante la referencia incluida a la página de la Red de Ciudades Inteligentes de España.

2030: smart city life 360 view

Signify. (16 de noviembre de 2016). *2030: smart city life 360 view* [Vídeo]. YouTube. <https://youtu.be/A1fcDP8L4gs>

Interesante vídeo de Philips Lighting en el que se muestra un anticipo de cómo evolucionarán las ciudades inteligentes en 2030, desde el punto de vista únicamente de tecnologías. Tendrán gran importancia las luminarias conectadas, para proporcionar muy diferentes servicios.

WIP: why limiting work in progress makes sense (Kanban)

David Lowe. (8 de octubre de 2013). *WIP: why limiting work in progress makes sense (Kanban)* [Vídeo]. YouTube. <https://youtu.be/W92wG-HW8gg>

Explicación del concepto *Work in progress* (WIP) de la metodología Kanban y sus implicaciones para la gestión del proyecto.

PMBOK® Guide Processes Flow

PMBOK® Guide Processes Flow. (S. f.). Ricardo Vargas. <https://ricardo-vargas.com/pmbok6-processes-flow/>

A través de este enlace tendrás acceso a una guía de conocimientos de gestión de proyectos (Guía PMBOK®): la sexta edición es la publicación más importante del Instituto de gestión de proyectos y es un recurso fundamental para la gestión eficaz de proyectos en cualquier industria. Se ha actualizado para reflejar las últimas buenas prácticas en gestión de proyectos.

Plan Nacional de Ciudades Inteligentes

Plan Nacional de Ciudades Inteligentes. (S. f.). red.es.
<https://red.es/es/iniciativas/proyectos/plan-nacional-de-ciudades-inteligentes>

En esta Web se recoge la información más relevante de la estrategia nacional para el desarrollo de Ciudades Inteligentes, como parte de la agenda digital para España.

Es interesante poder ver la información sobre los distintos aspectos que se contemplan: territorios inteligentes, transformación digital, profesionales digitales, etc, que recogen las mejores prácticas observadas en otras partes del mundo.

1. La computación en la niebla:
 - A. Contempla los problemas de utilizar computadores en entornos de niebla casi perenne, como la campiña inglesa.
 - B. Es conceptualmente idéntica a la computación en el extremo de la red, son sinónimos.
 - C. Desplaza el procesamiento de los datos de los sensores a la red local.
 - D. Ninguna de las anteriores es cierta.

2. En una Ciudad Inteligente o Smart City:
 - A. Se utilizan tecnologías muy diversas para mejorar la calidad de vida de los ciudadanos
 - B. Los distintos sistemas de sensorización pueden utilizarse para comprometer la privacidad
 - C. Se consigue una mayor eficiencia en distintos aspectos, como la gestión energética o la gestión de residuos urbanos
 - D. Todas las anteriores son ciertas.

3. Los Gemelos Digitales:
 - A. Son criptomonedas, como el Bitcoin.
 - B. Son gemelos para puños de camisa que están conectados a Internet y permiten recibir notificaciones de aplicaciones de mensajería, vibrando cuando ello sucede.
 - C. Son actores creados digitalmente para poder terminar películas cuando el protagonista ha fallecido o ha sido despedido en mitad del rodaje.
 - D. Ninguno de los anteriores.

4.Cuál de las siguientes frases es cierta en relación a un proyecto (señala todas las que procedan):

- A. Es un esfuerzo temporal.
- B. Incluye la fase de mantenimiento del producto.
- C. Pretende crear un único producto o servicio.
- D. No debe tener en cuenta a los interesados del proyecto.

5. El ciclo de vida del software:

- A. Abarca la evolución desde el primer prototipo del producto hasta su retirada.
- B. Abarca la evolución desde la concepción del producto hasta su retirada.
- C. Abarca la evolución desde la concepción del producto hasta su puesta en producción.
- D. Abarca la evolución desde que se consiguen los fondos necesarios hasta que no se dispone de presupuesto para continuar con el servicio.

6. El ciclo de vida en cascada:

- A. Permite gestionar de forma eficaz todo tipo de proyectos.
- B. No incluye la fase de mantenimiento.
- C. El diseño se hace de forma iterativa.
- D. No permite una detección temprana de errores y fallos en la toma de requisitos.

7. El ciclo de vida iterativo e incremental:
- A. Permite entregar «algo» de valor al cliente o usuario final de forma iterativa.
 - B. No permite entregar «algo» de valor al cliente o usuario final de forma iterativa.
 - C. Incluye una única fase de análisis y en la primera iteración.
 - D. Incluye una única fase de diseño y en la primera iteración.
8. Una metodología:
- A. Se limita a especificar cómo se realizará el diseño del producto.
 - B. No debe contemplar elementos de gestión del equipo.
 - C. Define un conjunto de procedimientos que permiten producir y mantener un producto software.
 - D. Estándar es la PMP.
9. La gestión ágil de proyectos:
- A. Permite ir más rápido ya que no obliga a documentar.
 - B. Es especialmente útil donde los requisitos son cambiantes y no están definidos con precisión.
 - C. Se basa en poner en una pizarra las tareas pendientes.
 - D. Sirve para todo tipo de proyectos en todos los ámbitos.
10. La limitación del trabajo en curso o Work In Progress (WIP):
- A. Es un concepto difundido por el PMI.
 - B. Aplica a todo tipo de proyectos y metodologías.
 - C. Es un concepto abstracto y poco práctico.
 - D. En la metodología Kanban limita el número de elementos que pueden estar en progreso en cada estado del flujo de trabajo.