

TEMA 1. INTRODUCCION

1.1 DEFINICIONES

Ubicua: dicho de una persona que todo lo quiere presenciar y vive en continuo movimiento. La definición tecnológica sería la siguiente; integración de la informática en el entorno de la persona, puede ser en cualquier dispositivo y soporta tecnologías como Internet, middleware, SO, código móvil, sensores, microprocesadores...

Se llama también Pervasive computing, Calm technology, Things That Think, Everyware, Inteligencia ambiental.

1.2 HISTORIA

Primera computadora electrónica ENIAC entre (1947-1955).

Concepto de computación ubicua se atribuye a Mark Weiser (1988-1994). Consideró tres revoluciones en computación:

- 1950 ordenadores centrales, usados por muchas personas.
- 1975 ordenadores pc, usados por una persona.
- 2000 computación ubicua, muchos ordenadores utilizados por una persona.

También planteó diferentes categorías; **tabs** (escasos centímetros), **pads** (tamaño de una mano), **boards** (llegan a medir metros) y se han añadido nuevos **dust** (dispositivos miniaturizados sin salida visual), **skin** (pueden emitir luz y dispositivos de fabricación orgánica), **clay** (conjunto de distintos MEMS que se combinan para crear formas 3D).

Inteligencia ambiental: unión entre tecnología y bienestar que supone mejoría sustancial en la vida del ciudadano construyendo una relación más segura, sostenible, natural y productiva entre el usuario y la tecnología.

1.3 ACTUALIDAD

La computación ubicua tiene trabajo por delante; problemas de costes y beneficios y problemas tecnológicos (consumo, fiabilidad...).

Persasive Computing: más conocido como IOT, son dispositivos informáticos no invasivos, los cuales no se vean y que abarquen entornos como el hogar.

Mobile Computing: es la interacción humano computadora, en donde se espera que una computadora sea transportada (móvil).

Wireless: es capaz de enviar información de un punto a otro sin la necesidad de hardware que conecte dos puntos (bt, wifi, nfc, gps...).

Sensores: encargados de interaccionar con nuestro entorno, variando ante magnitudes físicas o químicas y transformándolas a variables eléctricas.

La computación ubicua abrirá un nuevo abanico de posibilidades y se predice que para 2020 habrá 25millones de "cosas".

1.4 APLICACIONES EN EL MUNDO REAL

1.4.1 Internet of Things (IOT)

Es un sistema de dispositivos de computación interrelacionados que tienen id únicos y capaces de transmitir datos a través de una red sin necesidad de que un humano interactúe. Su finalidad es la conexión de dispositivos digitales entre sí para facilitar la automatización y empleo de estos u otros dispositivos. La conexión puede ser por Internet, bluetooth o infrarrojos.

Smart Buildings: sus instalaciones y sistemas permiten una administración y control automatizados para aumentar la eficiencia energética, seguridad y accesibilidad. Usan la domótica, son flexibles a nuevas tecnologías, respetuosos con el medio ambiente, buscan el confort de sus habitantes y reducen el consumo energético.

Smart Cities: sirven para obtener una gestión eficiente de todas las áreas de la ciudad, satisfaciendo las necesidades de esta y sus habitantes. Algunas características son; medio ambiente inteligente (gestión de recursos y energía), gobierno inteligente, movilidad inteligente (incentivar el uso del transporte público), economía inteligente...

Sistemas de transporte inteligentes: conjunto de aplicaciones informáticas y sistemas tecnológicos creados para mejorar la seguridad y eficiencia de los transportes. Algunos de los objetivos son; mejorar la seguridad, reducir el consumo de combustible, aumentar la capacidad de vías y sistema de transporte...

Industria 4.0: se denomina así a una fábrica inteligente, en donde cada máquina tiene conocimiento de que máquinas están modificando otras piezas pudiéndose adaptar a la secuencia de producción.

Logística Inteligente: conjunto de aplicaciones informáticas y sistemas tecnológicos creados con el objetivo de facilitar todas las tareas relacionadas con logística. Algunas ventajas serían buscar una ruta más óptima, mejorar los procesos de entrega, control remoto, visualización del estado de todos los productos, análisis exhaustivo del proceso de fabricación de un producto...

E-Health: tecnologías orientadas al ámbito de la salud. Algunos componentes son telemedicina, telesalud, m-health (uso de smartphones), telemonitorización de pacientes crónicos...

Agricultura Inteligente: tecnologías para la mejora del desarrollo agrícola y rural. Algunos de las tecnologías utilizadas son; sensores para medir diferentes variables, análisis de los datos, satélites y drones para recopilar información, tecnologías de telecomunicaciones...

TEMA 2. SISTEMAS INTELIGENTES

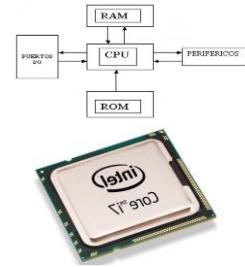
Hay diferentes elementos para crear sistemas inteligentes, a continuación, veremos algunos de ellos.

2.1 TIPOS

2.1.1 Microcontroladores y microprocesadores

Microcontroladores: circuito integrado que tiene CPU, RAM, ROM, puertos de entrada y salida y periféricos. Es una microcomputadora completa dentro de un circuito integrado. Ejemplo: Arduino.

Microprocesadores (CPU): tiene solo una unidad de procesamiento central, es el cerebro de un ordenador y hace operaciones a gran velocidad.



Las diferencias fundamentales son que el microcontrolador no es solo una CPU si no que además está conectado con periféricos a parte de tener RAM y ROM integrados. Los microprocesadores son más rápidos (como el mío jj 4,9 GHz en OC) mientras que los microcontroladores solo llegan a 200MHz. Son más costosos los microprocesadores.

2.1.2 Arduino

Es OpenSource, con un IDE para crear aplicaciones y cuenta con una gran comunidad que apoya el desarrollo. Se basa en un microcontrolador ATMEtal en donde se pueden grabar instrucciones teniendo las placas Arduino interfaz de entrada (para periféricos) e interfaz de salida (para enviar información procesada a otros periféricos).

2.1.3 Sensores

Pueden detectar magnitudes físicas y químicas para transformarlas a variables eléctricas. Estos pueden ser analógicos o digitales. Algunas características son rango de medida, precisión, offset, sensibilidad, resolución, rapidez en respuesta...

Algunos ejemplos son LDR (luz), Temperatura (LM35), Temperatura y humedad (DHT11).

2.1.4 Actuadores

Dispositivos capaces de transformar la energía (hidráulica, neumática, eléctrica) en la activación de un proceso para generar un efecto sobre un elemento externo.

Algunos ejemplos Led, Buzzer o bocina, Bomba de agua...

2.1.5 Periféricos

Dispositivo auxiliar que se conecta la CPU y que normalmente se necesitan drivers para su correcto funcionamiento. Son los que se comunican con el exterior.

Algunos ejemplos son Pantallas LCD, Teclados, Memorias externas, Micrófonos, Impresoras...

2.1.6 Shields

Son placas de circuitos modulares que se montan unas encima de otras para poder dar funcionalidades extra a las placas Arduino. Se puede añadir más de una shield y se alimentan mediante los pines GND y 5V



2.1.7 NodeMCU

Son placas o kits que llevan incorporados chips llamados SoC (System on a Chip) que dentro de ellos tienen microcontroladores o MCU.

Hay varias versiones; V0.9 (módulo ESP-12 y memoria flash de 4MB), V1.0 (módulo ESP-12E y pines extra) y los nuevos que son módulos que son ESP8266 (wifi con TCP/IP) y el ESP32 (wifi y bluetooth integrados).

2.1.8 Raspberry Pi

Es un ordenador de placa simple SBC de bajo coste, es software (no hardware como Arduino) OpenSource con un SO basado en Debian llamado Raspbian.

2.2 SISTEMAS INTELIGENTES

Un sistema inteligente es aquel capaz de resolver problemas complejos de una forma automática, dando soporte a las decisiones de un experto. Tienen características asimilables a la inteligencia humana o animal.

Algunos hitos y avances en los sistemas inteligentes

- 1950 – Alan Turing publica “Computing Machinery and Intelligence”
- 1956 – Conferencia en la Universidad de Dartmouth
- 1961 – Marvin Minsky publica “Pasos hacia la inteligencia artificial”
- 1966 – El programa interactivo conversacional ELIZA
- 1972 – Hubert Dreyfus publica “Lo que no pueden hacer los ordenadores”
- 1979 – Un ordenador vence al backgammon
- 1981 – Japón comienza el proyecto “Quinta Generación”
- 1987 - Martin Fischer y Oscar Firschein describen los atributos de un agente inteligente
- 1997 – “Deep Blue” vence a Kasparov, campeón mundial de ajedrez
- 2005 – Un ordenador al volante
- 2011 – Watson gana Jeopardy!
- 2014 - Un ordenador supera con éxito el Test de Turing
- 2016 – Microsoft lanza Tay
- 2016 – Alphago vence al Go
- 2017 – Libratus vence al póker

TEMA 3. ASISTENTES VIRTUALES

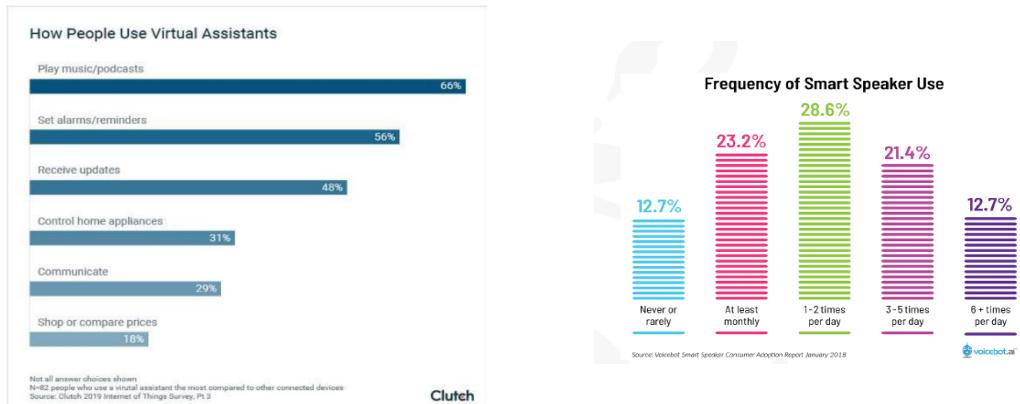
Un asistente virtual es un agente de software que ayuda a usuarios de sistemas computacionales, automatizando y realizando tareas con la mínima interacción hombre-máquina.

Los asistentes virtuales se pueden encontrar en diversos dispositivos como son, ejemplo, Amazon Echo (Alexa), Google Home, Whatsapp, Siri, Cortana, Bixby...

3.1 HISTORIA DE LOS ASISTENTES VIRTUALES

- 1961 – IBM Shoebox, primera herramienta con reconocimiento digital de voz. Reconoce 16 palabras y dígitos de 0 a 9
- 1970 – Harpy dominó 1 millón de palabras
- 1980 – Primer sistema capaz de reconocer cadenas de palabras
- 1990 – IBM, Microsoft, Philips y Lernout & Hauspie incorporan reconocimiento de voz a sus computadoras personales
- 1994 – IBM Simon, primer teléfono inteligente
- 1996 – Microsoft incorpora un asistente a Office llamado Clippy
- 2011 – Siri, primer asistente virtual instalado en un teléfono
- 2014 – Amazon lanza el altavoz Amazon echo
- 2016 – Google comercializa Google Home

3.2 USO DE LOS ASISTENTES VIRTUALES



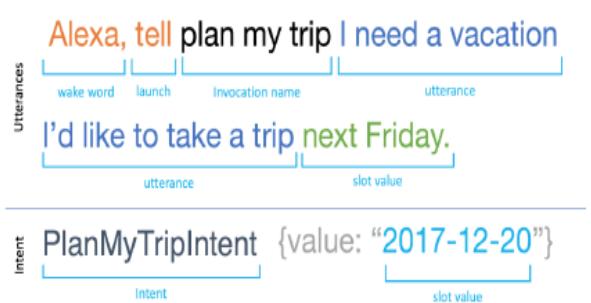
3.3 RECONOCIMIENTO POR VOZ

Hay 3 elementos esenciales:

Intents: Solicitud o acción principal asociada con el comando del usuario.

Slots: argumentos que para un intent proporcionan más información, estos pueden tener de tipo obligatorio u opcional.

Utterances: las palabras que los usuarios dicen para transmitir lo que quieren que el asistente haga.



3.4 TIPOS DE ASISTENTES VIRTUALES

3.4.1 Cortana

Este es el asistente virtual creado por Microsoft como una pieza para su cambio de imagen. Comenzó su desarrollo en 2009 pero no fue presentado hasta 2013. Este asistente está disponible para Windows 10, Windows Phone, Android y Xbox One.

Creación de skills: para crear una skill en Cortana se utiliza Microsoft Azure. Para crear la skill el procedimiento será el siguiente; pulsaremos en

“Crear recurso” -> “AI+Machine Learning” -> “Web App Bot”

Una vez que hemos creado el bot podremos probarlo online, revisar su diseño descargar o cargar el código, publicarlo o conectarlo a canales.

3.4.2 Alexa

Fue desarrollado por Amazon en 2014, su nombre fue elegido debido a la X es una consonante complicada lo cual facilitaría reconocer el nombre. Se creó para ser incorporado a altavoces inteligentes “Echo”.

Creación de skills: para esto se debe utilizar la Amazon Developer Console. El procedimiento será el siguiente;

“AlexaSkillsKit” -> ponemos el nombre -> “Next” -> “Intents” -> “Create custom intent”

Para terminar la configuración de la Skill se debe usar Amazon AWS, además hay que buscar el servicio “Lambda” y crear una función (el servicio “Lambda” se encarga de recibir las solicitudes de Alexa y responderlas).

3.4.3 Google Assistant

Fue desarrollado y presentado por Google en 2016 para los dispositivos Pixel y Pixel XL. Se encuentra disponible para Android y en altavoces inteligentes y Google anuncio que podría interactuar con la cámara para poder reconocer objetos.

Creación de skills: para esto necesitamos una cuenta de Gmail para poder acceder a la herramienta Dialogflow y crear un nuevo agente;

“Create agent” -> ponemos nombre -> “Create”

Luego crearemos los intent:

“Intent” -> “Create Intent” -> personalizamos -> “Save”

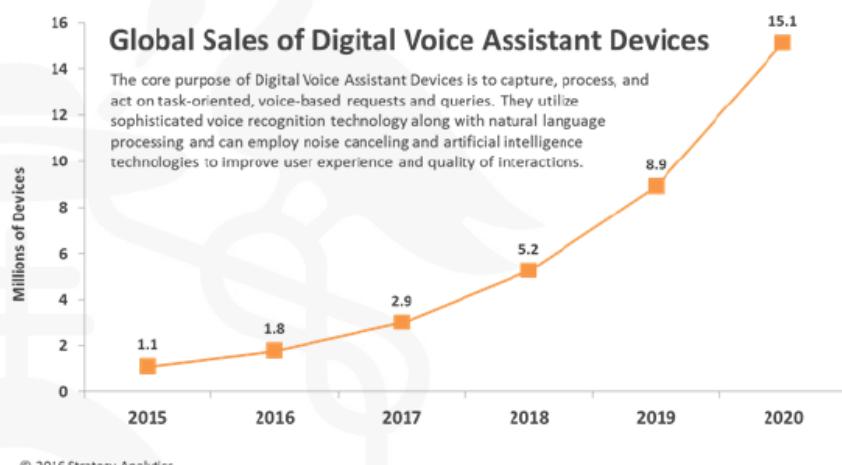
3.4.4 Siri

Es considerada la primera aplicación de asistente personal virtual, fue desarrollado por Sri venture group en 2008. En 2010 se lanzó una app para iPhone y dos meses después Apple compró Siri. En 2011 fue incorporada al iPhone 4s y en 2018 fue lanzado su altavoz inteligente, el Home pod.

3.4.5 Otros asistentes

- Bixby: creado en 2017 por Samsung y disponible en algunos de sus dispositivos
- Braina: asistente virtual para ordenador que permite traducir a más de 100 idiomas.
- VoiceMate: creado en 2017 por LG está disponible en algunos de sus dispositivos aunque puede instalarse en otros que no tienen que ser necesariamente de LG.
- J.A.R.V.I.S: asistente virtual para el control de electrodomésticos inteligentes, se puede descargar en dispositivos móviles.
- Aura: desarrollado por Telefónica en 2017, está orientado a la aplicación Movistar+ y programado sobre los servicios Azure Machine Learning.
- Dueros: desarrollado por Baidu's, está disponible para dispositivos Android y tiene una gran popularidad en China.

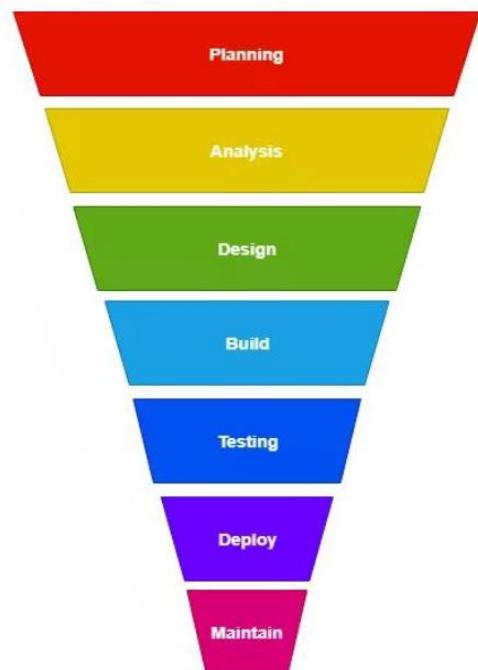
3.5 EVOLUCION



TEMA 4. METODOLOGIAS DE DESARROLLO

El ciclo de vida de desarrollo de sistemas (SDLC) es el proceso de creación o modificación de los sistemas, modelos y metodologías que la gente usa para desarrollar estos sistemas de software. Este concepto en ingeniería del software sostiene muchos tipos de metodologías de desarrollo de software

La metodología de desarrollo en ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. Es una filosofía de desarrollo de programas de computación con el enfoque del proceso de desarrollo de software. Además, contiene herramientas, modelos y métodos para asistir al proceso de desarrollo de software



4.1 TIPOS DE METODOLOGIAS

Metodologías Tradicionales	Metodologías Agiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo	Basadas en heurísticas provenientes de prácticas de producción de código
Cierta resistencia a los cambios	Preparados para cambios durante el proyecto
Proceso mucho más controlado, con numerosas políticas/normas	Proceso menos controlado, con pocos principios
El cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente es parte del equipo de desarrollo
Grupos grandes y posiblemente distribuidos	Grupos pequeños y trabajando en el mismo sitio
La arquitectura del software es esencial y se expresa mediante modelos	Menos énfasis en la arquitectura del software

4.1.1 Metodologías tradicionales

-**Modelo en cascada:** modelo con un enfoque secuencial propuesto en 1970, las fases están ordenadas de forma que el principio de cada etapa debe esperar a la finalización de la inmediatamente anterior y cualquier fallo en fases anteriores será arreglado en la fase actual.

- Ventajas: requisitos muy definidos, gente con poca experiencia, comenzar el desarrollo con rapidez, precisión de calendarios y presupuestos, proyectos grandes.
- Inconvenientes: poca flexibilidad, realmente pocos proyectos siguen un flujo secuencial, reunir todos los requisitos desde el inicio, las revisiones de proyectos de gran complejidad son difíciles y usuarios participación limitada.

-Modelo incremental: modelo iterativo y creciente que surge como solución a los problemas del modelo en cascada, se basa en mini cascadas que se hacen en cada iteración así de este modo se pasa en cada iteración por todas sus fases.

- Ventajas: generación de software rápido y en etapas tempranas, modelo flexible a los cambios, facilidad para probar y depurar en iteraciones pequeñas, fácil gestión de riesgos y cada iteración es un hito.
- Inconvenientes: cada fase en una iteración es rígida y no se superpone con otras y posibles problemas de arquitectura en el caso de no definir correctamente los requisitos.

-Modelo en V: proceso que representa la secuencia de pasos en el desarrollo del ciclo de vida de un proyecto. La V significa “Verificación y Vendetta”. Muy similar al modelo de cascada clásico ya que es muy rígido y contiene una gran cantidad de iteraciones.

- Ventajas: especifica bien los roles de las pruebas a realizar, implica chequeos en cada etapa, sencillo y de fácil aprendizaje e involucra al usuario en las pruebas.
- Inconvenientes: no permite retomar etapas inmediatamente anteriores, si un proceso es mal diseñado debe ser revisado, ocasionando sobrecostes, las pruebas pueden ser caras y a veces no efectivas.

-Modelo basado en prototipado: modelo con enfoque iterativo, basado en realizar pequeños prototipos finales de la aplicación de forma que sus funcionalidades se construyen encima de la versión anterior, hasta llegar al producto definitivo y su entrega al cliente.

- Ventajas: ofrece visibilidad del producto desde el inicio, es flexible con respecto a los cambios, permite la retroalimentación del cliente, el prototipo es un documento vivo del producto final, el cliente reacciona mucho mejor ante el prototipo, reduce el riesgo de producir algo que no cumpla con las necesidades del cliente.
- Inconvenientes: es un desarrollo lento, necesidad de realizar una fuerte inversión en un producto que se descartará, puede aumentar el coste del desarrollo.

-Modelo en espiral: permite un enfoque evolutivo, las actividades se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades. Este modelo toma en consideración la evaluación de riesgos, se usa la creación de prototipos como mecanismo de reducción de riesgos.

- Ventajas: se adapta a lo largo de la vida del software, mejor reacción ante riesgos en cada nivel, permite aplicar prototipos, permite incluir otros métodos de desarrollo en las iteraciones, se tienen puntos de control en cada iteración.
- Inconvenientes: dificultad para convencer a clientes sobre un enfoque evolutivo, requiere mucha administración, dificultad para definir los objetivos, metas que indiquen que podemos avanzar al siguiente ciclo, complejo para sistemas pequeños y pueden caer en un desarrollo de nunca acabar.

4.1.2 Metodologías ágiles

Aquellas que permiten adaptar la forma del 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.

Manifiesto ágil

- La prioridad es satisfacer al cliente mediante la entrega temprana y continua de software.
- Los cambios son bienvenidos
- Entregas frecuentes de software que funcione, en periodos breves.
- El cliente y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto.
- Construcción de proyectos en torno a individuos motivados.
- Comunicación cara a cara.
- El software que funciona es la principal medida del progreso.
- Promueven un desarrollo sostenido, donde se mantenga un ritmo constante.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos que se auto-organizan.
- El equipo debe reflexionar sobre cómo ser más efectivo y ajustar su conducta.

4.2 METODOLOGIA SCRUM

Metodología basada en la experiencia y la toma de decisiones. Se centra en la división del trabajo en distintos bloques que pueden ser abordados en periodos cortos de tiempo (1-4 semanas) denominados Sprint. Consta de cuatro elementos fundamentales:

- Reunión de planificación del Sprint (Sprint Plan)
- Scrum diario (Daily Scrum)
- Revisión del Sprint (Sprint Review)
- Retrospectiva del Sprint (Sprint Retrospective)



El equipo de desarrollo se compone de; dueño del proyecto, scrum master (ayuda al equipo y cliente), equipo de desarrollo (5 a 9 personas)

-Ventajas: gestión de las expectativas del cliente, resultados anticipados, flexibilidad y adaptación a los contextos, cada persona sabe que tiene que hacer, equipo más motivado, programación organizada.

-Inconvenientes: difícil implantación en equipos grandes, definición de las tareas y sus plazos, exceso de reuniones con pocos avances, problemas si el desarrollo está restringido por un plazo o precio de entrega, requiere un equipo con experiencia.

4.3 METODOLOGIA XP (EXTREME PROGRAMMING)

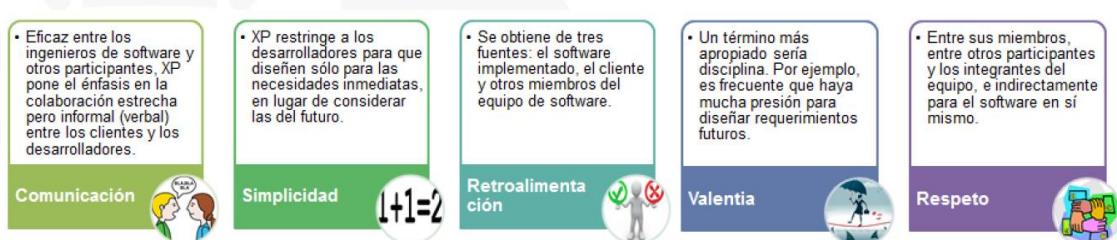
Proporciona un desarrollo iterativo e incremental. Esta metodología se basa en reglas y principios que se han utilizado a lo largo de la historia del desarrollo software. Aquí se pone el énfasis en la retroalimentación continua entre clientes y el equipo de desarrollo, es idónea para proyectos con requisitos imprecisos y muy cambiantes.

Los roles de trabajo son los siguientes; programador, cliente, encargado de pruebas, encargado de seguimiento, entrenador, consultor, jefe de proyecto.

-Ventajas: programación organizada, mayor eficiencia en la planificación y pruebas, menor tasa de errores, propicia la satisfacción del programador, fomenta la comunicación entre clientes y desarrolladores, fácil adaptación ante cambios o problemas, realización de pruebas continuas durante el proyecto, el cliente tiene el control sobre las prioridades.

-Inconvenientes: es recomendable emplearlo solo en proyectos a corto plazo, altas comisiones en caso de fallar, requiere de un rígido ajuste al inicio del proyecto, demasiado costoso e innecesario.

4.3.1 Valores establecidos como fundamento para el trabajo



4.4 METOLOGIA KANBAN

Es una metodología complementaria a SCRUM, se visualiza el flujo de trabajo y este se tiene que dividir en tareas e incluirlas en el tablero. Dicho tablero está organizado en tres secciones: tareas que hay que hacer, tareas en curso y tareas terminadas.

Los principios de Kanban son poder visualizar el flujo de trabajo, limitar el trabajo en progreso, respetar los roles y realizar un seguimiento, monitoreo y análisis constantes.

-Ventajas: Estimula el rendimiento, mayor organización y colaboración, comodidad para distribuir el trabajo, fácil de aplicar, gestión de tareas muy visual, promueve el trabajo en equipo, control del flujo de trabajo, reducción del tiempo perdido, facilidad para añadir mejoras

-Inconvenientes: no es una técnica específica del desarrollo software, dificultad para anticiparse a los problemas, difícil implantación en proyectos grandes, falta de reglas.

4.5 METODOLOGIA SCRUMBAN

Metodología derivada de los métodos de desarrollo Scrum y Kanban. El flujo de trabajo es el establecido por Kanban, pero con elementos de Scrum como son reuniones diarias y análisis retrospectivos para incorporar mejoras. Se pueden tomar dos caminos:

○ Kanbanizar Scrum

- Visualizar el trabajo previsto en el Sprint
- Gestionar el flujo de forma activa por medio de la autoorganización
- Mejorar en base a la experiencia y a la retrospectiva
- Añadir límites WIP

○ Scrumizar Kanban

- Introducir roles de Scrum
- Introducir los eventos

-Ventajas: permite conocer el estado del proceso de ejecución del proyecto, introduce soluciones ante errores, permite un mayor análisis de las tareas realizadas, mejora la interacción de los miembros del grupo, aumenta la productividad de proyectos complejos, favorece la adaptabilidad de las herramientas al proyecto.

-Inconvenientes: difícil implantación en equipos grandes, requiere de una exhaustiva definición de las tareas y sus plazos, exceso de reuniones con pocos avances, problemas si el desarrollo por un plazo o precio de entrega requiere de un equipo con experiencia, inflexible a cambios.

4.6 METODOLOGIA LEAN

Creación de equipos altamente preparados y que puede llevar tareas en poco tiempo. El equipo es lo más importante. Cuando el equipo más haya aprendido y más unidos se encuentren el tiempo cada vez será menor. Busca un equipo sólido con programadores que analicen la situación, tomen decisiones y que el cliente esté lo más satisfecho posible con el menor número de recursos. Es perfecta para proyectos a medio plazo.

○ Se puede aplicar siguiendo 7 principios:

1. **Eliminar el desperdicio.** Eliminar lo que no aporte valor (Muda). Código basura, mala toma de requisitos, documentación excesiva...
2. **Ampliar aprendizaje.** Todos los miembros del equipo deben tener mentalidad de aprendizaje continuo
3. **Tomar decisiones tardías.** No tomar la toma de requisitos inicial al pie de la letra, ir construyéndola con los requisitos del cliente
4. **Entregar rápido.** Cada entrega incluirá funcionalidades requeridas por los usuarios lo antes posible
5. **Potenciar el equipo.** Permitir que los desarrolladores participen en la toma de decisiones
6. **Crear integridad.** Contar con un buen sistema de integración continua que incluya pruebas automatizadas, de usabilidad, etc
7. **Visualizar todo el conjunto.** *"Pensar en grande, actuar en pequeño, equivocarse rápido y aprender con rapidez"*

-Ventajas: la optimización de procesos permite reducir costes, el emprendedor asume menos riesgos, reducción del plazo de ejecución y las actividades sin valor, se fomenta el trabajo en equipo.

-Inconvenientes: dependencia de la cohesión del equipo, fuerte inversión inicial, necesidad de un equipo con altas cualidades, toma de decisiones tardía puede acarrear problemas

4.7 ELECCION DE UNA METODOLOGIA

A la hora de elegir una metodología se debería de tener en cuenta la capacidad de cada metodología para afrontar proyectos:

-Requerimientos claros y concretos o cambiantes.

-Tiempo disponible amplio o reducido

-Funcionalidades sencillas o complejas

También hay que evaluar el nivel de conocimiento de la tecnología a utilizar, determinar la complejidad del sistema a crear, el nivel de claridad y concreción de los requerimientos, establecer la fiabilidad esperada del sistema, determinar los plazos y establecer los límites para cada fase del proyecto

TEMA 5. MODELO DE NEGOCIO

5.1 DEFINICION DE MODELO DE NEGOCIO

Un modelo de negocio describe la lógica de cómo una organización **crea, entrega y capta** un valor.

Representación abstracta de una organización, ya sea de manera textual o gráfica, de todos los conceptos relacionados, acuerdos financieros, y el porfolio central de productos o servicios que la organización ofrece y ofrecerá con base en las acciones necesarias para alcanzar las metas y objetivos estratégicos.

5.2 OBJETIVOS

El principal objetivo de una empresa es **ganar dinero**. Para que esto sea posible son necesarios 3 factores: **crear, distribuir y retener valor**

5.3 MODELO CANVAS

Plantilla de gestión estratégica para desarrollar nuevos modelos de negocio o documentar y mejorar los ya existentes.



5.3.1 Propuesta de valor (qué ofrecemos)

Describe el conjunto de productos y servicios que una empresa ofrece a sus clientes: **beneficios que aportamos y factores que nos diferencian de la competencia**.

Para una buena propuesta tenemos que saber:

- El valor que le entregamos al cliente.
- El problema de nuestro cliente que solucionamos.
- Las necesidades que estamos satisfaciendo.

Algunos ejemplos de propuestas de valor son la **novedad, el diseño, la marca, el precio, mejoras en producto, mejoras en la prestación del servicio, propuestas de valor diferentes para cada segmento**, etc.

Otra característica para una buena propuesta sería encontrar un problema que valga solucionar. Puedes ocurrir dos posibilidades:

- **Detectar un problema** propio que nos proporcione beneficios.
- **Estudiar el mercado** y detectar problemas no solucionados.

Para estos problemas, valdría la pena solucionarlos. **Solución aceptable, mejor solución o solución más barata.**

Para terminar, es evidente la propuesta de los beneficios, ya que tenemos que explicar cómo vamos a solucionar el problema de los futuros cliente:

- **Qué beneficios** les aportamos
- **Qué nos diferencia** de la oferta existente

5.3.2 Segmentos de Cliente (a quién va dirigido)

Define los diferentes grupos de personas u organizaciones a los que **ofrecer** nuestro producto o servicio. Los clientes son el corazón de nuestro modelo.

Podemos agrupar a los clientes por diferentes segmentos como edad, gustos, comportamiento... Debemos saber **para quién** estamos vendiendo el producto y **quiénes** son los clientes más importantes.

Ejemplos de segmentos de cliente:

- **Mercado de masas:** abarca un mercado lo más amplio posible sin distinguir entre segmentos.
- **Nichos de mercado:** segmentos de mercado con una alta especialización que atiende a unos requisitos muy específicos
- **Segmentados:** segmentos de mercado con necesidades poco diferenciadas.
- **Diversificados:** segmentos muy diferentes entre sí, con necesidades únicas.
- **Multi-segmentados:** segmentos que dependen de varios segmentos a la vez.

5.3.3 Canales (cómo te conocen)

Describe el medio por el que se comunica y alcanza a sus segmentos de mercado para entregar una propuesta de valor.

Los canales centran su actividad en 5 fases:

- **Notoriedad.** Cómo hacer que nos conozcan.
- **Evaluación.** Cómo ayudamos al cliente a evaluar la propuesta de valor.
- **Compra.** Cómo permitimos que compren nuestro servicio/producto.
- **Entrega.** Cómo entregamos la propuesta de valor.
- **Postventa.** Cómo proporcionamos servicios de postventa al cliente.

Tipos de Canal		Fases del Canal				
Propio	Directo	Fuerza de ventas	1. Percepción. ¿Cómo elevamos la percepción acerca de los productos y servicios?	2. Evaluación. ¿Cómo ayudamos a nuestros clientes a evaluar la proposición de valor de nuestra organización?	3. Compra. ¿Cómo permitimos que nuestros clientes compren de manera específica nuestros productos y servicios?	4. Entrega. ¿Cómo entregamos la proposición de valor a los clientes?
		Ventas por Internet				
	Indirecto	Tiendas propias				
		Tiendas asociadas				
		Mayorista				

5.3.4 Relaciones con clientes (cómo interactúas)

Describe los tipos de **relaciones** que una empresa establece con un segmento específico del mercado.

Para saber las claves de las relaciones con los clientes es necesario saber **qué tipo** de relación queremos establecer, **cuánto nos costará** y **cómo de integrado** está con el resto de nuestro modelo de negocio.

Existen tres tipos de relación con el cliente: **directa, indirecta y automatizada**.

Ejemplos de relación:

- Asistencia personal
- Asistencia personal dedicada
- Self-service
- Servicios automatizados
- Comunidades
- Co-creación
- Redes sociales

5.3.5 Fuentes de ingresos (dinero, satisfacción)

Representa el dinero que la empresa genera de cada segmento de mercado

Para encontrar las fuentes de ingresos:

- Cuál es la propuesta de valor por la que están **dispuestos** a pagar nuestros clientes.
- Cuál es la propuesta por la que pagan **actualmente**
- Cómo **están pagando**
- Cómo **preferirían pagar**
- Cuál es el **porcentaje de ingresos** de cada línea de ingreso respecto a los ingresos totales

Tipos de fuentes de ingresos: **puntuales y recurrentes**

Fuentes de ingresos:

- **Venta de activos.** Transferencia de los derechos de propiedad
- **Cuota por alquiler.** Derecho temporal de uso de un producto o servicio
- **Cuota por uso.** Uso de un servicio en particular
- **Concesión de licencias.** Se concede el derecho de propiedad intelectual
- **Cuota por publicidad.** Se cobra para dar visibilidad a un producto/servicio/marca
- **Cuota por suscripción.** Se concede acceso continuo a un servicio

5.3.6 Recursos clave (quién eres y qué tienes)

Describe los más **importantes activos** requeridos para hacer funcionar el modelo de negocio.
Necesitamos saber los recursos clave de:

- **La propuesta de valor**
- **Los canales de distribución**
- **La relación con el cliente**
- **Las fuentes de ingresos**

Tipos de recursos Clave:

- **Físicos.** Instalaciones de fabricación, edificios, vehículos, puntos de venta
- **Intelectuales.** Marcas, patentes
- **Humanos.** Las personas que hacen posible el proyecto
- **Económicos.** Efectivo, líneas de crédito

5.3.7 Actividades clave (Qué haces)

Describe las **actividades** más **importantes** que una empresa debe hacer para que un modelo de negocio funcione. Las actividades claves deben englobar desde la creación del producto hasta su entrega al cliente

Necesitamos saber las actividades clave de:

- **La propuesta de valor**
- **Los canales de distribución**
- **La relación con el cliente**
- **Las fuentes de ingresos**

Tipos de actividades clave:

- **Producción.** Relacionadas con el diseño, la fabricación y la entrega.
- **Resolución** de problemas. Implican la búsqueda de soluciones nuevas para los clientes.
- **Plataforma/Redes.** Define actividades para mantener una buena relación con los clientes, proveedores y grupos de interés.

5.3.8 Socios clave (quién te ayuda)

Describe la **red de suplidores y socios** que hacen que un modelo de negocios funcione

Para encontrar a nuestros socios clave necesitamos saber:

- **Quiénes son nuestros socios clave**
- **Quiénes son nuestros proveedores clave**
- **Cuáles son los recursos clave que estamos adquiriendo de los socios**
- **Cuáles son las actividades clave que hacen los socios**

5.3.9 Estructura de costes (tiempo, energía, estrés)

Describe los **costes necesarios** para operar un modelo de negocio.

Para encontrar la estructura de costes necesitamos saber cuáles son:

- **Los costes más importantes de nuestro modelo de negocio.**
- **Los recursos clave más caros.**
- **Las actividades clave más caras.**

Fases para determinar la estructura de costes:

- 1) **Clasificación** de los elementos integrantes del coste
- 2) **Periodificación** de los gastos del proceso de producción
- 3) **Localización**, atribución o repartos de los costes (alquileres, salarios, etc)
- 4) **Imputación** de los costes a los elementos productos elaborados



TEMA 6. CLOUD E INTERNET DE LAS COSAS

6.1 CLOUD COMPUTING

La computación en la nube es un paradigma que permite ofrecer servicios de computación **a través de una red**, que usualmente es Internet.

Características:

- Agilidad
- Costo
- Escalabilidad
- Independencia
- Rendimiento
- Seguridad
- Mantenimiento

Ventajas:

- Integración de servicios
- Rápida implementación
- Servicios ubicuos
- Green computing

Desventajas:

- Seguridad
- Madurez
- Escalabilidad
- Centralización
- Disponibilidad

Servicios en la nube



6.1.1 IAAS

La infraestructura como servicio (IaaS) es la más simple de las tres categorías para definir, ya que es ampliamente el mismo independientemente del proveedor que elija. En pocas palabras, IaaS es un tercero que proporciona una **infraestructura de TI** altamente automatizada y escalable (almacenamiento, alojamiento, computación, redes) y solo cargos por lo que usa.

Por lo tanto, en lugar de poseer activos como licencias de software o servidores, las empresas pueden alquilar recursos de forma flexible de acuerdo con sus necesidades.

Entre las ventajas de la IaaS, destaca su alta escalabilidad. Estas infraestructuras permiten ampliar su capacidad en función de las necesidades concretas de tu negocio. Además, ya no son necesarias grandes inversiones en hardware ni afrontar sus costes de amortización. Los ejemplos más utilizados de plataformas IaaS son Amazon Web Services (AWS) y Microsoft® Azure®.

6.1.2 PAAS

La plataforma como servicio (PaaS) es posiblemente el más difícil de los tres modelos de nubes para definir limpiamente. La idea es proporcionar todos los conceptos básicos de IaaS, así como las **herramientas y capacidades necesarias para desarrollar e implementar aplicaciones de forma segura**. Eso podría ser middleware, administración de bases de datos, análisis o un sistema operativo.

Una plataforma como servicio debería proporcionarle a un desarrollador todo lo que necesita para construir e implementar una aplicación sin tener que hacer ninguna provisión de la infraestructura subyacente.

Entre las ventajas del PaaS, destacan: su flexibilidad, pues el cliente tiene el control pleno sobre las herramientas instaladas en su plataforma; la adaptabilidad, ya que puedes cambiar las características en función de tus necesidades; la movilidad, puesto que tienes acceso a la plataforma desde cualquier dispositivo sólo con conectarte a Internet; y la velocidad en el desarrollo, debido a que con esta infraestructura el entorno estará preparado para tal efecto – muchos incluyen CMS. Algun ejemplo podría ser AppEngine o Force.com.

6.1.3 SAAS

El software como servicio (SaaS) es el lugar donde una parte del software está alojada por un tercero y **se puede acceder a través de la web**, normalmente solo iniciando sesión, y generalmente se cobra por suscripción o por usuario. . Esto difiere del antiguo modelo de compra e instalación de software en una máquina o servidor de forma manual.

SaaS es mucho más relevante para aplicaciones muy específicas, como el correo electrónico o el software de gestión de relaciones con el cliente (CRM). Cualquiera que haya usado una aplicación de Google como Gmail o Google Docs, o almacenamiento de archivos en la nube como Dropbox, habrá usado una pieza de SaaS.

Una de sus grandes ventajas son el acceso multidispositivo – ya que sólo necesitas disponer de conexión a Internet – y el ahorro en costes. Con soluciones SaaS tienes a tu disposición soluciones profesionales de gran calidad, evitando que queden obsoletas, ya que alquilas su uso junto con otros profesionales y empresas. Grandes proveedores de esas tecnologías son los que se encargan de la creación de la aplicación, su almacenamiento, administración, seguridad y mantenimiento. Youtube, Gmail, Netflix, Dropbox...

6.2 INTERNET DE LAS COSAS

El IOT es un concepto que se refiere a una **interconexión digital** de objetos cotidianos con internet. Alternativamente, el internet de las cosas es la conexión de internet con más cosas u objetos que con personas.

Redes de comunicación M2M (machine to machine):

- Xbee
- Bluetooth
- Wifi
- Ethernet
- Mobile

6.3 NUBE E IOT

Internet de las cosas nos permiten automatizar procesos en base a una recogida de datos en tiempo real y acumular esta información para posibles tomas de decisión posteriores, o como base para el aprendizaje (inteligencia artificial) como apoyo a la toma de decisiones.

Aplicaciones de gestión, monitorización, Dashboard.

6.4 EJEMPLO DE PROYECTO

- Objetivo: controlar temperatura confort de una casa
- Soluciones:
 - Poner un termostato inteligente
 - Cambiar todas las llaves de los radiadores por llaves inteligentes y el termostato actual por uno inteligente
 - Implantar un sistema domótico personalizado
- Pros/Contras:
 - Opción A: único punto de control de temperatura en la casa. Dependemos de un proveedor externo. Coste aprox 300€ + coste mensual de controlador avanzado (10-25€).
 - Opción B: varios puntos de control de la temperatura. Compra de un termostato inteligente. Dependemos de un proveedor externo. Coste 1000€ + coste mensual.
 - Opción C: sistema personalizado. Necesitamos conocimientos de programación y de dispositivos. Coste 200€.

Nuestra solución

- 3 placas ESP8266/ESP32
- 3 Sensores DHT22
- 1 relé para el termostato
- Raspberry Pi 3 B+ y raspbian
- MQTT, Python, nginx, mongodb y javascript

TEMA 7. INDUSTRIA 4.0

La Industria sufrió 4 revoluciones industriales:

- Primera Revolución Industrial (1760 – 1840)
 - Fuentes de energía: Vapor
 - Sistema propulsor: Máquina de vapor
 - Tipo de mercado: Nacional, Internacional
 - Industrias básicas: Textil, Siderurgia, Metalurgia
 - Producción: Equipamientos mecánicos
- Segunda Revolución Industrial (1840 – 1920)
 - Fuentes de energía: Electricidad, Petróleo
 - Sistema propulsor: Motor explosión, Motor Eléctrico
 - Tipo de mercado: Imperialista, Colonial
 - Industrias básicas: Química, Electrónica, Automovilista
 - Producción: En serie
- Tercera Revolución Industrial
 - Fuentes de energía: Electricidad, Petróleo, Gas, Solar
 - Sistema propulsor: Motor, Ordenador, Señales digitalizadas
 - Tipo de mercado: Globalizado, Zonas de libre comercio
 - Industrias básicas: Informática, Electrónica, Robótica, Medios de comunicación
 - Producción: Automatizada
- Cuarta Revolución Industrial
 - Fuentes de energía: Electricidad, Petróleo, Gas, Renovables
 - Sistema propulsor: Motor, Ordenador, Sensores, Sistemas y modelos de visualización avanzado
 - Tipo de mercado: Globalizado, Compartido, Virtual (Internet)
 - Industrias básicas: Informática, Electrónica, Robótica, Redes sociales, Móviles, Personal, Internet de las cosas, Big data, Nanotecnología, Biotecnología
 - Producción: Industry 4.0

Lo real y lo digital están conectados. Todo está conectado en una Cadena de producción – Pieza – Máquina.

Cada máquina está informada cuando la pieza esté modificándose en otra máquina. Las máquinas se adaptarán automáticamente a la secuencia de la producción para ajustar la unidad de producción en serie a fabricar.

7.1 NUEVO PARADIGMA DE DISEÑO

Los programas de diseño 3D han cambiado la producción y diseño de los productos. A partir de un diseño 3D es posible diseñar otros sistemas donde irá incorporada esa pieza. También se utiliza para diseñar la pieza y comprobarla como si fuera la misma y ya después decidir si fabricarla o no.

7.2 GESTION DEL CICLO DE VIDA DE UN PRODUCTO

- Diseño asistido por ordenador (CAD)
- Ingeniería asistida por ordenador (CAE) y simulación 3D
- Análisis de elementos finitos (FEA o FEM)
- Realidad Virtual (RV), la realidad aumentada (RA)
- Manufacturing Execution System (MES)
- Fabricación asistida por ordenador (CAM)
- Sistemas de planificación de recursos empresariales (ERP)
- Gestión de operaciones de producción (MOM)
- Gestión de datos del producto (PDM).

7.3 FABRICACION INTEGRADA POR ORDENADOR

Tiene diferentes niveles de automatización:

- Nivel 0: Comunicación a pie de máquina, buses y sensores que intercambian información con la planta.
- Nivel 1: Control básico, control de partes de forma autónoma
- Nivel 2: Sistema de supervisión y adquisición de datos por ordenador
- Nivel 3: MES: Manufacturing Execution System
- Nivel 4: ERP: Enterprise Resources Planning

7.4 FABRICACION DIGITAL

Permite:

- El desarrollo de un modelo 3D de la planta que sirve para diseñar y planificar como van a ir las maquinas y utillajes para la fabricación
- El desarrollo simultáneo de todos los procesos relacionados con ellos.
- Reducir el ciclo de puesta en marcha de la fábrica. Elaborar un gemelo digital de la planta y así ver todas las gestiones necesarias de la planta.

7.5 SISTEMAS CIBERFISICOS

Tiene dos partes diferenciables, la virtual y la real de forma integrada permitiendo el intercambio de datos. Sistemas que ofrecen una combinación estrecha y coordinada entre los elementos computacionales y físicos de un sistema. Permite la reducción de tiempos de desarrollo de la planta permitiendo adelantar en otras fases.

7.6 REALIDAD VIRTUAL Y AUMENTADA

Aplicaciones:

- Optimización de diseños
- Mantenimiento y control de planta
- Training de operaciones y formación de operarios
- Asistencia y resolución de incidencias

Ventajas:

- Mejora de los procesos
- Reducción de tiempos de espera
- Aumento de la seguridad
- Ahorro de costes

7.7 COMUNICACIÓN M2M

La interconexión de maquinaria va a permitir el funcionamiento coordinado de diferentes medios de producción de una empresa.

Se considera cualquier tecnología que permita a los dispositivos que se encuentren en una misma red, intercambiar información y realizar acciones de forma totalmente autónoma.

7.8 EL INTERNET DE LAS COSAS (IoT)

IoT está utilizando esa comunicación para proporcionar una solución.

Algunas aplicaciones de impacto según el foro mundial de IoT:

- Monitorizar la calidad del aire
- Detectar la presencia de Ozono
- Localización de piezas dentro de la planta

7.9 CLOUD COMPUTING

Modelo tecnológico que permite acceder a un conjunto de recursos informáticos de forma ubicua, personalizada y bajo demanda a través de internet.

Ventajas:

- Deslocalización de los datos
- Permite la integración de datos provenientes de sistemas integrados de fabricación
- Mayor capacidad de almacenamiento
- Mayor capacidad de procesamiento
- Permite la compartición de la información
- Mayor integración de bases de datos

7.10 ¿POR QUÉ SE USAN LOS DATOS?

Obtener, analizar y explotar los datos es una necesidad de la empresa:

- Para conocer el comportamiento y el rendimiento de las fábricas es necesario medir
- Lo que no se puede medir no se puede mejorar
- Los datos deben ser transformados en información de valor añadido
- La información es fundamental para la toma de decisiones que mejoren la competitividad

TEMA 8. IOT – ASISTENTES VIRTUALES

Los asistentes virtuales más populares son:

Aura (Telefónica), Siri (Apple), Google Assistant (Google), Alexa (Amazon), Cortana (Microsoft), Bixby (Samsung), Xiao AI (Xiaomi)

Algunos conceptos que hay que tener en cuenta son:

- NLP: Natural Language Processing
- NLU: Natural Language Understanding
- NLG: Natural Language Generation
- Intents / Slots / Utterances: Acciones del habla

Phishing en Google Assistant:

1. Se crea una Action maliciosa en Google Assistant
2. Se crean 2 intents, uno con un silencio y otro escuchando un token
3. Al invocar el segundo intent, se fuerza al usuario a decir su contraseña como el token

Estafa del CEO:

1. A través de Generative Adversarial Networks (GANs) se genera imagen en movimiento
2. Mediante Neural Text-To-Speech (TTS) basado en Deep Neural Networks generamos la voz artificial
3. Con retroques digitales se genera y afina el video.

8.1 CONCEPTOS DE IOT

HARDWARE: El corazón de IoT son miles de millones de dispositivos interconectados con sensores y actuadores conectados que detectan y controlan el mundo físico

PROGRAMACION INCRUSTADA (EMBEDDED PROGRAMMING): Los dispositivos IoT son dispositivos integrados y se pueden crear prototipos utilizando plataformas de microcontroladores comercializadas, como Arduino, con placas de circuito impreso (PCB) personalizadas desarrolladas en una etapa posterior

SEGURIDAD: La seguridad es una de las preocupaciones más críticas en IoT, estrechamente relacionada con la ética, la privacidad y la responsabilidad de los datos. Debe estar integrado en cada paso del diseño del sistema

REDES E INTEGRACION DE LA NUBE: El diseño y la administración de la red son esenciales dentro de IoT, debido al gran volumen de dispositivos conectados y al impacto que las decisiones de diseño de la red pueden tener a escala

ANALISIS DE DATOS Y PREDICCION: Los desarrolladores necesitarán ingerir, almacenar y consultar de forma segura y confiable las grandes cantidades de datos heterogéneos que se originan en estos dispositivos

APRENDIZAJE AUTOMATICO Y IA: Para ser verdaderamente inteligente, el análisis de big data debe aplicar técnicas de computación cognitiva extraídas de la minería de datos, el modelado, la estadística, el aprendizaje automático y la IA

8.2 ETAPAS EN IOT

1. ESCANEOS: Se detectan dispositivos vulnerables por crawlers
2. REPORTE: Mediante documentación se recopilan contraseñas por defecto, malas configuraciones...
3. INTELIGENCIA: Se recopila información de los objetivos a atacar
4. ATAQUE: Ataque masivo con los dispositivos infectados
5. COLAPSO: Organismos e infraestructuras críticas colapsan

8.3 SEGURIDAD CONVENCIONAL VS SEGURIDAD IoT

- **CONVENCIONAL:**
 - SEGURIDAD: Protección del dispositivo y de la información que almacena
 - PRIVACIDAD: Manipulación y compartición de datos con terceros
 - FIABILIDAD: Funcionamiento correcto bajo condiciones concretas
- **IoT:**
 - SEGURIDAD: Protección de la infraestructura
 - PRIVACIDAD: Protección de los datos
 - FIABILIDAD: Protección de la información

TEMA 9. SMART CITIES Y FIRWARE

9.1 DEFINICIONES SMART CITIES (SM):

- Integración de sistemas físicos, digitales y humanos en un entorno construido para ofrecer un futuro sostenible, próspero e inclusivo para sus ciudadanos (British Standards Institute 2014)
- Ciudades que adoptan soluciones escalables aprovechando las tecnologías TIC para aumentar la eficiencia, reducir costes y mejorar la calidad de vida (Falconer and Mitchell, 2012)

Las SM son aquellas que utilizan el potencial tecnológico y la innovación y otros recursos para hacer de ellos un uso más eficaz, promover el desarrollo sostenible, y mejorar la vida de sus ciudadanos.

9.2 URBANICACION:

En 2017 la población urbana superó a la población rural en todo el mundo

Pronóstico para 2050:

- 70% de la población mundial vivirá en ciudades
- Existirán ciudades con más de 10 millones de habitantes
- En 2025 sólo en China habrá 221 ciudades que superarán el millón de habitantes

Las ciudades tienen un gran impacto económico y social, son grandes centros consumidores de recursos, generando el 80% de gases de efecto invernadero y consumiendo el 75% de recursos y energía mundiales.

9.3 CARACTERISTICAS DE LAS SM

Las características principales que brindan las Smart Cities son mejores en; medio ambiente, vida, movilidad, gobierno, economía y ciudadanos.

9.4 BENEFICIOS DE LAS SM:

- Favorece la innovación: ayuda a innovar, desarrollar nuevos negocios e ideas y favorecer el desarrollo social.
- Ayuda en la toma de decisiones: facilita identificar las necesidades reales de la ciudad y el planteamiento de nuevos servicios para ayudar
- Reducción del gasto público
- Incremento de la eficiencia (gestión más eficiente de los recursos) y la calidad de servicios
- Ofrece información en tiempo real: más y mejor info en tiempo real y al mismo tiempo mejora la transparencia de la administración

9.5 BARRERAS DE LAS INFRAESTRUCTURAS INTELIGENTES:

- Gobierno inteligente: necesidad de invertir a largo plazo
- Calidad y gestión de los datos: la calidad de los datos debe conocerse antes de utilizarlos para optimizar un sistema
- Privacidad
- Inversión: los sistemas tradicionales para el retorno de inversión no tienen en cuenta la complejidad de un sistema de sistemas
- Vida: la vida útil del sistema debe coincidir con la vida útil de la infraestructura

9.6 SMART LONDON:

- Plan de tecnología inteligente de Londres: Creciente presión sobre la salud, el transporte, otros servicios públicos, la energía y la gestión de la contaminación
- Contribución de las empresas tecnológicas
- El mejor sistema de transporte: sistemas de movilidad de vanguardia, estacionamiento inteligente y gran esfuerzo en la gestión de problemas de congestión crónica
- London Datastore: conecta a todos los ciudadanos con todos los eventos que ocurren en la ciudad, abierto para que todos accedan a la información y estadísticas.
- La última construcción de transporte inteligente: ofrecer un viaje seguro y eficiente

9.7 SMART BARCELONA

Conectividad wifi en bus, metro, mercados; apps para ayudar a ciudadanos; eventos tecnológicos de gran innovación en IoT con sensores de aparcamiento, alumbrado público, gestión de recogida de residuos...

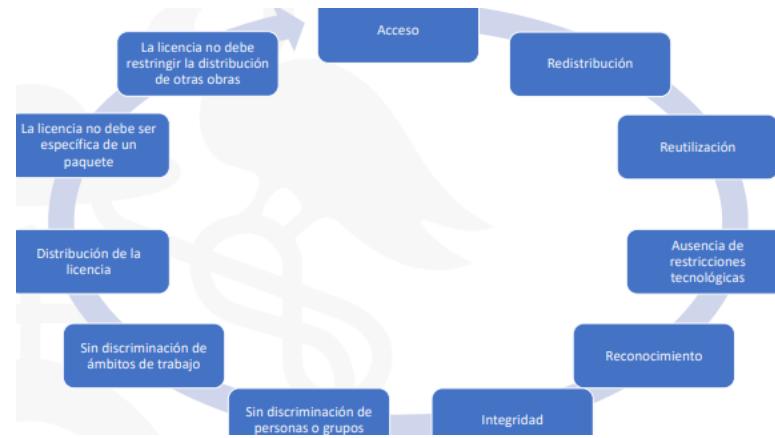
Ranking	City	Total Score	Vision	Leadership	Budget	Financial Incentives	Support Programmes	Talent-Readiness	People Centricity	Innovation Ecosystem	Smart Policies	Track Record	Ranking	City	Performance	CIMI
1	London	33,5	3,1	4	3	4	3	3,1	3	4,1	3,1	3,1	1	London - United Kingdom	H	100,00
2	Singapore	32,3	3	4	3	4,1	3	3,1	2	3,1	4	3	2	New York - USA	H	94,63
3	Seoul	31,4	3,1	3	3	2,2	3	3	4,1	3	3	4	3	Amsterdam - Netherlands	RH	86,70
4	New York	31,3	3	3	3	3,1	3	3,1	3	4	2	4,1	4	Paris - France	RH	86,23
5	Helsinki	31,2	3	2	4	3,1	3	4	3	3,1	2	4	5	Reykjavik - Iceland	RH	85,35
6	Montreal	30,1	3,1	3	3	4	3	2	3	3	3	3	6	Tokyo - Japan	RH	84,11
7	Boston	29,6	3	3	3	2,1	3	3,1	3,1	3,1	3,1	3,1	7	Singapore - Singapore	RH	82,73
8	Melbourne	29,5	3	3	3	2,1	3,1	3,1	4	3,2	2	3	8	Copenhagen - Denmark	RH	81,80
9	Barcelona	29,4	3	3	3	2,1	2	3,1	3	3,1	3,1	4	9	Berlin - Germany	RH	80,88
10	Shanghai	29,2	3	3	4	3,1	3	2	2	3	2,1	4	10	Vienna - Austria	RH	78,85
													11	Hong Kong - China	RH	78,76
													12	Seoul - South Korea	RH	78,13
													13	Stockholm - Sweden	RH	77,89
													14	Oslo - Norway	RH	77,45
													15	Zurich - Switzerland	RH	76,66
													16	Los Angeles - USA	RH	76,04
													17	Chicago - USA	RH	75,55
													18	Toronto - Canada	RH	75,30
													19	Sydney - Australia	RH	75,26
													20	Melbourne - Australia	RH	75,08
													21	San Francisco - USA	RH	75,07
													22	Helsinki - Finland	RH	74,08
													23	Washington - USA	RH	73,14
													24	Madrid - Spain	RH	73,02
													25	Boston - USA	RH	72,91
													26	Wellington - New Zealand	RH	72,82
													27	Munich - Germany	RH	72,71
													28	Barcelona - Spain	RH	72,25

Ranking SM

IESE Cities in Motion

9.8 OPEN DATA:

Persigue que tipos determinados de datos estén disponibles de forma libre para todo el mundo, sin restricciones de derechos de autor, de patentes o de otros mecanismos de control. Tiene una ética parecida al SW libre, código abierto...



- Beneficios económicos Open Data: creación de servicios y aplicaciones basados en contenidos digitales. La reutilización de la información pública en España generó un volumen de negocio anual de entre 330 y 550 millones de euros
- Beneficios para el ciudadano: acercamiento a gobierno abierto, pueden generar apps y servicios que mejoren la vida de los ciudadanos.
- Beneficios para las administraciones públicas: intercambio de datos entre administraciones, aumento de colaboración ciudadana para mejorar el servicio público.

9.9 FIWARE: Tecnología + ecosistema

Fomenta la innovación y desarrollo de soluciones digitales. El principal objetivo es la creación de un ecosistema capaz de unir a los proveedores de tecnología, los emprendedores, desarrolladores y dueños de los datos.

9.9.1 Etapas del proyecto FIWARE

-Etapa 1:

- Duración: abril 2011 – mayo 2014 (3 años)
- Inversión: 90 millones de €
- Objetivo: Definir la base tecnológica y crear 8 proyectos de casos de uso
- Tareas realizadas:
 - Sentar las bases de la tecnología
 - Definición de escenarios de “casos de uso” en diferentes sectores
 - Hacer un inventario de las infraestructuras públicas disponibles a través de la creación de capacidades

-Etapa 2:

- Duración: abril 2013 – marzo 2015 (2 años)
- Inversión: 80 millones de €
- Objetivo: Desarrollo de la plataforma central a través del proyecto XIFI y la implantación de nodos FIWARE
- Tareas realizadas:
 - Desarrollo de pilotos y plataformas de casos de uso a gran escala
 - Creación de infraestructuras para operar una red europea de nodos FIWARE, conocidos hoy como laboratorio FIWARE

-Etapa 3:

- Duración: septiembre 2014 – septiembre 2016
- Inversión: 100 millones de €
- Objetivo: Mejora de una infraestructura estable para los ensayos a gran escala y creación de un ecosistema sostenible para la innovación de las PYME
- Tareas realizadas:
 - Expansión de los casos de uso de las aplicaciones
 - Ampliación de la base tecnológica
 - Lanzamiento del Programa Acelerador FIWARE con más de 1.000 empresarios, nuevas empresas o PYME que se benefician de las tecnologías y servicios FIWARE de forma gratuita

9.9.2 Ventajas de usar FIWARE

Posibilidad de acceder a una biblioteca de recursos y herramientas disponibles de forma libre y gratuita, soluciones TIC en sectores con alta demanda, acceso a información abierta y gratuita, posibilidad de formar parte de una comunidad de desarrolladores, posibilidad de acceder a financiación europea, desarrollo de apps que impulsan soluciones innovadoras.

9.10 COMUNIDAD FIWARE

“Construir un ecosistema sostenible abierto a todo el público, libre de reglas y estándares de plataformas de software de aplicación, que facilitará el desarrollo de nuevas aplicaciones inteligentes en múltiples sectores”.

Miembros individuales: Cualquier persona que quiera participar por su cuenta contribuyendo a las actividades de la comunidad. Los requisitos son contribuir de forma activa a dichas actividades FIWARE, ser constructivo y seguir las normas, acceso gratuito.

Miembros del núcleo: empresas que hacen una apuesta estratégica importante a FIWARE. Los requisitos son tener una estrategia corporativa alineada con la misión FIWARE, proporcionar una inversión sustancial a las actividades y pagar una cuota. Telefónica, Orange, NEC...

Miembros regulares: empresas que proporcionan una gran inversión, pero a nivel inferior. Los requisitos son tener una estrategia corporativa alineada con la misión FIWARE, proporcionar recursos para las actividades en curso y pagar una cuenta de miembro regular.

Algunos ejemplos de uso de FIWARE son SmartSantander, detección de multitudes en Madrid o SmartValencia

9.11 ESTRUCTURA FIWARE

FIWARE Lab: entorno no comercial donde tiene lugar la innovación y experimentación. Los recursos ofrecidos por FIWARE Lab son una infraestructura de cómputo, componentes SW de la plataforma FIWARE, datos abiertos por ciudades y organizaciones.

FIWARE Accerate: su objetivo principal es promover las tecnologías FIWARE entre las PYME y start-ups (la UE aportó 80 millones para apoyar a PYME y empresarios que quieran innovar en FIWARE). Promueve un mercado global donde industrias, ciudades y empresarios pueden participar, atracción de nuevas empresas mediante la interoperabilidad.

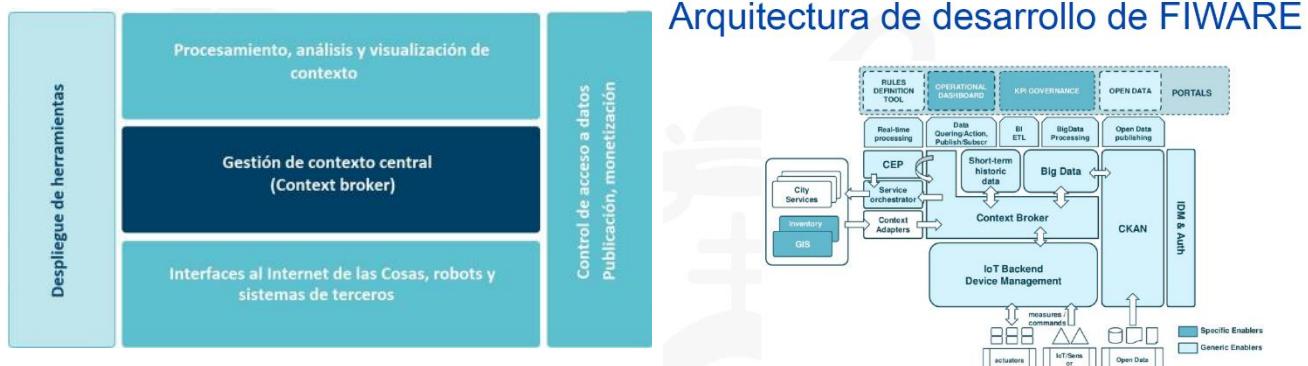
FIWARE Mundus: nace como proyecto europeo, pero con diseño global, ofrece cobertura en otras regiones, con ayuda de participantes y la coordinación de gobiernos de América latina, Asia y África

FIWARE iHubs: enfocado al ecosistema de negocio, educación e innovación. Cada uno de los iHubs está formado por 1 o más centros que llevan a cabo actividades como formación, instrucción, educación y ayudas a PYME, StartUps y empresas. Las actividades se centran en la promoción de FIWARE, así como la creación de conexiones entre universidades, empresas, instituciones, etc.

9.12 PLATAFORMA FIWARE

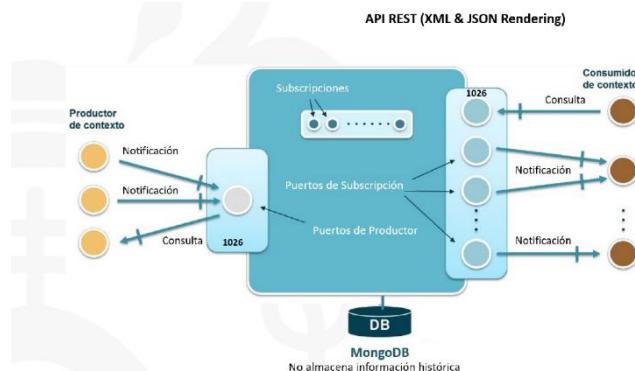
Compuesta por componentes de código abierto que se pueden ensamblar a otros para acelerar el desarrollo. Dichas componentes proporcionan APIs con especificaciones públicas.

El principal y único componente obligatorio es Context Broker, este permite administrar la info, consultarla y actualizarla. Está rodeado por otros componentes que suministran datos de otras fuentes y ofrece una serie de herramientas que facilitan la implementación y diversos ajustes.



9.12.1 Orion Context Broker(OCB)

Servidor que implementa una API basada en NGSI (Next Generation Service Interface) con el que podemos registrar apps de proveedores, actualizar info, ser notificado de cambios, consultar info de contexto. Se utiliza la base de datos MongoDB que no almacena info histórica.



9.12.2 Modelo NGSI (Next Generation Service Interface)

Modelo de datos para la info de contexto, interfaz que sirve para intercambio de información, disponibilidad de contexto de cómo obtener la información.

9.13 JSON

9.13.1 Representación de una entidad en JSON

–“id”. Identificador de la entidad

–“type”. Tipo de entidad

```
{  
    "id": "entityID",  
    "type": "entityType",  
    "attr_1": <val_1>,  
    "attr_1": <val_1>  
}
```

9.13.2 Representación de un atributo

–Nombre del atributo

–“value”. Valor del atributo. Puede ser cualquier tipo JSON

–“type”. Tipo de atributo

–“metadata”. Contiene una propiedad por elementos de metadatos definidos

```
{  
    "value": <...>,  
    "type": <...>,  
    "metadata": <...>  
}
```

9.13.3 Modelo de alertas

```
{  
    "id": "urn:ngsi-ld:Alert:Alert:1",  
    "type": "Alert",  
    "createdAt": "2019-06-06T12:06:06",  
    "modifiedAt": "2019-06-07T12:07:06",  
    "category": {  
        "type": "Property",  
        "value": "traffic"  
    },  
    "subCategory": {  
        "type": "Property",  
        "value": "trafficJam"  
    },  
    "validTo": {  
        "type": "Property",  
        "value": {  
            "@type": "DateTime",  
            "@value": "2017-01-02T09:25:55.00Z"  
        }  
    },  
    "description": {  
        "type": "Property",  
        "value": "The road is completely blocked for 3kms"  
    },  
    "location": {  
        "type": "GeoProperty",  
        "value": {  
            "type": "Point",  
            "coordinates": [-3.71224722222222, 40.42385277777775]  
        }  
    },  
}
```

```
"dateIssued": {  
    "type": "Property",  
    "value": {  
        "@type": "DateTime",  
        "@value": "2017-01-02T09:25:55.00Z"  
    }  
},  
"alertSource": {  
    "type": "Property",  
    "value": "https://account.lab.fiware.org/users/8"  
},  
"validFrom": {  
    "type": "Property",  
    "value": {  
        "@type": "DateTime",  
        "@value": "2017-01-02T09:25:55.00Z"  
    }  
},  
"severity": {  
    "type": "Property",  
    "value": "high"  
},  
"@context": [  
    "https://schema.lab.fiware.org/ld/context",  
    "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"  
]
```

Modelo de Parques y Jardines

```
{  
  "id": "urn:ngsi-ld:Garden:Santander-Garden-Piquio",  
  "type": "Garden",  
  "category": {  
    "type": "Property",  
    "value": ["public"]  
  },  
  "style": {  
    "type": "Property",  
    "value": "french"  
  },  
  "description": {  
    "type": "Property",  
    "value": "Jardines de Piquio. Zona El Sardinero"  
  },  
  "dateLastWatering": {  
    "type": "Property",  
    "value": {  
      "@type": "DateTime",  
      "@value": "2017-03-31T08:00:00Z"  
    }  
  },  
  "location": {  
    "type": "GeoProperty",  
    "value": {  
      "type": "Point",  
      "coordinates": [-3.7836974, 43.4741891]  
    }  
  },  
  "refRecord": {  
    "type": "Relationship",  
    "object": ["urn:ngsi-ld:Record:Santander-Garden-Piquio-Record-1"]  
  },  
  "areaServed": {  
    "type": "Property",  
    "value": "El Sardinero"  
  },  
  "address": {  
    "type": "Property",  
    "value": {  
      "addressLocality": "Santander",  
      "postalCode": "39005",  
      "streetAddress": "Avenida Casta\u00f1eda",  
      "type": "PostalAddress"  
    }  
  },  
  "openingHours": {  
    "type": "Property",  
    "value": "Mo-Su"  
  },  
  "name": {  
    "type": "Property",  
    "value": "Jardines de Piquio"  
  },  
  "@context": [  
    "https://schema.lab.fiware.org/ld/context",  
    "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"  
  ]  
}
```

Datos del ambiente

```
{  
  "id": "waterqualityobserved:Seville:D1",  
  "type": "WaterQualityObserved",  
  "dateObserved": {  
    "type": "DateTime",  
    "value": "2017-01-31T06:45:00Z"  
  },  
  "temperature": {  
    "value": 24.4  
  },  
  "NO3": {  
    "value": 0.01  
  },  
  "location": {  
    "type": "geo:json",  
    "value": {  
      "type": "Point",  
      "coordinates": [-5.993307, 37.362882]  
    }  
  },  
  "pH": {  
    "value": 7.4  
  },  
  "measurand": {  
    "value": ["NO3, 0.01, M1, Concentration of Nitrates"]  
  },  
  "conductivity": {  
    "value": 0.005  
  }  
}
```

9.13.4 Operaciones con la API RESTful NGSI v2

Comprobar la versión: GET <URL>:1026/versión

-Operaciones con **Entidades**:

- Buscar todas las entidades: GET /v2/entities
- Crear una nueva entidad: POST /v2/entities
- Buscar/comprobar el valor de una entidad: GET /v2/entities/{entityID}
- Actualizar el valor de una entidad: [PUT|PATCH|POST] /v2/entities/{entityID}
- Eliminar una entidad: DELETE /v2/entities/{entityID}

-Operaciones con **atributos**:

- Buscar un atributo de una entidad: GET /v2/entities/{entityID}/attrs/{attrName}
- Actualizar el valor de un atributo: POST /v2/entities /{entityID}/attrs/{attrName}
- Eliminar un atributo: DELETE /v2/entities /{entityID}/attrs/{attrName}
- Obtener el valor de un atributo: GET /v2/entities /{entityID}/attrs/{attrName}/value
- Actualizar el valor de un atributo: PUT /v2/entities /{entityID}/attrs/{attrName}/value

-Operaciones sobre **Suscripciones**:

- Obtener suscripciones: GET /v2/subscriptions
- Crear nuevas suscripciones: POST /v2/subscriptions

-Operaciones sobre una **Subscripción**:

- Leer: GET /v2/subscriptions/{subscriptionId}
- Modificar: PATCH /v2/subscriptions/{subscriptionId}
- Borrar: DELETE /v2/subscriptions/{subscriptionId}

-Filtrar entidades según un patrón:

- GET <URL>:1026/v2/entities?q={attribute}>{value}
- GET <URL>:1026/v2/entities?q=temperature>22

-Obtener representación sólo con los valores de los atributos

- GET <URL>:1026/v2/entities/{id}?options=keyValues

GET <URL>:1026/v2/entities/Room1?op