



# Universidad de Alcalá

Escuela Politécnica Superior

Universidad de Alcalá

## **Computación ubicua**

### ***Macroresumen***

Viernes 10:00 – 12:00

Grado en Ingeniería Informática – Curso 2019/2020

Marcos Barranquero Fernández – 51129104N

## CONTENIDO

TEMA 1 – Introducción .....	4
Historia de la computación ubicua .....	4
Situación actual .....	4
El negocio de la computación ubicua .....	5
Aplicaciones .....	5
IOT .....	5
Smart buildings .....	5
Smart cities .....	5
Transportes inteligentes .....	5
Industria 4.0 .....	6
Logística inteligente .....	6
E-Health .....	6
Agricultura inteligente .....	6
TEMA 2 – Introducción .....	7
Ejemplo – arduino .....	7
Ejemplo – NodeMCU .....	7
Ejemplo – Raspberry pi .....	8
Sistemas inteligentes .....	8
TEMA 3 – Introducción .....	9
Reconocimiento de voz .....	9
Cortana – Microsoft .....	9
Alexa – Amazon .....	9
Google assistant – Google .....	10
Siri – Apple .....	10
Otros .....	10
T4 – INTRODUCCIÓN .....	11
Metodologías tradicionales .....	11
Metodologías ágiles .....	13
Metodología Scrum .....	13
Metodología XP .....	14
Metodología Kanban .....	14
Metodología scrumban .....	14
Metodología lean .....	14
Estadísticas varias .....	15
Elección de metodología .....	15

T5 – Modelos de negocio .....	16
Modelo canvas.....	16
Propuesta de valor .....	16
Segmentos del cliente .....	16
Canales de comunicación .....	16
Relaciones con los clientes.....	16
Fuente de ingresos .....	16
Recursos clave .....	16
Actividades clave .....	17
Socios clave .....	17
Estructura de costes .....	17
T6 – Introducción .....	18
IOT – Internet of things .....	18
Nube e IOT .....	18
T7 – Introducción .....	19
t8 - Introducción .....	20
T9 – Introducción .....	20
Contexto .....	20
Ciudades inteligentes.....	21
Open data.....	21
Fiware.....	22
Arduino .....	22

# T1 – COMPUTACIÓN UBICUA

## TEMA 1 – INTRODUCCIÓN

La **computación ubicua** es una tendencia tecnológica que persigue **integrar la informática con el entorno de la persona**, con cualquier dispositivo **embebido**, en cualquier lugar y formato, mediante tecnologías que lo soporten como internet, móviles, sensores, microprocesadores, etc.

Tenemos ejemplos como los móviles, las pulseras inteligentes, etc.

## HISTORIA DE LA COMPUTACIÓN UBICUA

El primero en utilizar el término fue **Mark Weiser** durante los 90, y vaticinó 3 revoluciones:

1. Ordenadores centrales – un ordenador para muchas personas.
2. Ordenadores personales – un ordenador para una persona.
3. Computación ubicua – varios ordenadores para una persona.

Sin embargo, se debe añadir la computación móvil.

También consideraba 4 principios informáticos:

- El propósito de un dispositivo informático es ayudar en una tarea
- El mejor dispositivo es aquel que pasa desapercibido
- Los dispositivos deben entender tu inconsciente
- La tecnología debe aportar calma y bienestar

Este señor planteó 3 categorías, y se han añadido tres más:

- Tabs: dispositivos de centímetros pequeños que lleva el usuario
- Pads: dispositivos del tamaño una mano
- Boards: dispositivos de metros
- Dust: dispositivos miniatura que no se ven
- Skin: dispositivos diminutos que se engloban en tejidos de la ropa
- Clay: dispositivos MIMO que se combinan para crear formas 3D.

La **inteligencia ambiental** es la unión entre tecnología y bienestar que, mediante la computación ubicua, crea una relación sostenible entre el usuario y la tecnología, mejorando la vida del ciudadano.

## SITUACIÓN ACTUAL

Un **sistema embebido**, a diferencia de un ordenador de propósito general, es un sistema de computación diseñado para realizar una o algunas funciones dedicadas específicas.

La **movilidad** de un sistema es la capacidad de poder desplazarse y adaptarse a varios entornos.

A día de hoy, podemos clasificar los dispositivos ubicuos según la siguiente tabla:

<b>Mobility</b>	High	Mobile Computing	Ubiquitous Computing
	Low	Traditional Computing	Pervasive Computing
		Low	High

**Embeddedness**

Algunos conceptos más incluyen:

- **Pervasive computing (IOT):** dispositivos informáticos no invasivos que buscan ser invisibles y automatizar entornos e interacciones.
- **Mobile computing:** comunicación, hardware y software móvil, computadora transportada.
- **Tecnología Wireless:** comunicación sin una conexión física, mediante protocolos sin cable.
- **Sensores:** encargados de interactuar con el entorno y transformar magnitudes físicas o químicas en eléctricas y extraer información.

## EL NEGOCIO DE LA COMPUTACIÓN UBICUA

Se espera que la computación ubicua crezca exponencialmente en el futuro y la adición de tecnologías abra nuevas posibilidades a las empresas.

## APLICACIONES

### IOT

Son sistemas de dispositivos de computación relacionados. Usan una conexión y envían datos a través de una red. Facilitan la automatización y uso de estos dispositivos a otros a los que controlen.

### SMART BUILDINGS

Son edificios cuyos sistemas permiten la administración y control de sistemas integrados. Buscan aumentar la eficiencia energética, seguridad y accesibilidad, el bienestar y poder adaptarse a las nuevas tecnologías.

### SMART CITIES

Pues igual que los edificios pero con ciudades. Vaya locura, ¿eh? Busca mejorar el transporte, la economía, la ciudadanía, todo lo que se te ocurra.

### TRANSPORTES INTELIGENTES

Buscan mejorar la eficiencia, capacidad, movilidad, seguridad, etc...

---

## INDUSTRIA 4.0

Fábricas inteligentes conectadas al IOT. Integran las piezas con las máquinas de la cadena de producción. Cada máquina se comunica con las otras para informar y verificar estados, y se adaptan a la secuencia de producción.

---

## LOGÍSTICA INTELIGENTE

Aplicaciones y sistemas tecnológicos que buscan facilitar el control, gestión y seguimiento de la producción. Búsqueda de ruta más óptima, análisis del proceso en busca de ser más eficiente, toma de decisiones en tiempo real, información sobre el estado, etc.

---

## E-HEALTH

Uso de tecnologías en apoyo a campos relacionados con la salud, atención médica, educación, etc. Servicios médicos a distancia, historial clínico, prescripción online...

---

## AGRICULTURA INTELIGENTE

Aplica las tecnologías para el desarrollo agrícola y rural: sensores de control, gps, robótica y automatización, análisis de datos, etc.

# T2 – SISTEMAS INTELIGENTES

## TEMA 2 – INTRODUCCIÓN

Un **microcontrolador o MCU** es un circuito integrado con **CPU, RAM y ROM, puertos de entrada y salida y periféricos**. Viene a ser una microcomputadora encapsulada en un circuito integrado.

El microcontrolador se suele utilizar en sistemas **específicos**, es más **sencillo y barato**, emplea un **bus interno** y trabaja a **bajas frecuencias**.

Un **microprocesador** es un chip integrado con una **CPU**. Sin ram, ni rom, ni periféricos...

El microprocesador se utiliza en sistemas de **propósito general**, es más **complejo y caro**, utiliza un **bus externo** y trabaja a **altas frecuencias**.

### EJEMPLO – ARDUINO

Una placa **Arduino** está basada en un microcontrolador en el que se pueden grabar instrucciones. Posee interfaz de entrada y salida para periféricos.

Se le pueden conectar **sensores**, que pueden ser **analógicos o digitales**, y que tienen características como el rango de medida, la precisión, la sensibilidad, la rapidez de respuesta, la tasa de actualización... El Arduino se puede programar para leer estos sensores.

Se puede interactuar con los sensores de forma **asíncrona o síncrona**.

Un **actuador** es un dispositivo que transforma energía en la activación de un proceso con la finalidad de generar un efecto sobre el elemento externo. Vienen a ser salidas (por ejemplo, un led) o dispositivos finales (por ejemplo, una bomba de agua).

Un **periférico** es un dispositivo exterior e independiente conectado a la CPU.

Un **shield** es una placa de circuitos modular que se montan unas sobre otras para añadir funcionalidad extra a un Arduino.

### EJEMPLO – NODEMCU

Existe un firmware y una placa de desarrollo llamados NodeMCU.

La placa lleva un chip con un microcontrolador programable dentro.

Existen varios modelos con más o menos pines, Wifi, Bluetooth, etc. Programable con el IDE de Arduino, MicroPython, etc.

## EJEMPLO – RASPBERRY PI

Es un ordenador de bajo coste dedicado a la enseñanza, con software de código abierto.

La raspberry pi es más potente, más flexible por sw, se conecta fácil a internet.

El Arduino es más sencillo, barato, ampliable, pero solo se programa en C/C++.

## SISTEMAS INTELIGENTES

Un sistema inteligente es aquel que **resuelve problemas complejos y multidisciplinares** de forma automática dando soporte a las decisiones del supervisor.

El sistema está definido por su objetivo, su inteligencia o capacidad de lograr sus objetivos, sus reglas y su capacidad de aprendizaje.



# T3 – ASISTENTES VIRTUALES

## TEMA 3 – INTRODUCCIÓN

Un **asistente virtual** es un agente software que ayuda al usuario automatizando y realizando tareas con la mínima interacción hombre-máquina.

Desde los años 60 se lleva inventando con reconocimiento de voz y comprensión de texto, móviles, asistentes como el Clippy, Apple inventó a Siri para contar chistes, etc.

Los asistentes virtuales permiten interactuar por **texto, voz e imágenes**.

Sus principales usos son interactuar con multimedia, establecer alarmas o recordatorios, recibir actualizaciones, control de otros sistemas, comparar precios, etc.

La media de uso está en 1 o 2 veces al día.

## RECONOCIMIENTO DE VOZ

---

### CORTANA – MICROSOFT

Asistente de voz que permite comandos como abrir aplicaciones, gestionar calendario, buscar en internet, etc. por ordenador.

Utiliza big data y procesamiento Machine Learning sobre esos datos para calibrar sus acciones.

En azure se pueden crear Skills, comandos creados por el usuario para ejecutarse.

---

### ALEXA – AMAZON

Su nombre está bien pensado porque pocas palabras tienen x y así es más fácil reconocer el nombre cuando la llamas. Qué hackers, ¿eh?

Permite reproducir música, crear alarmas, etc... Pensado pa usarse en un altavoz de amazon.

Arquitectura similar a Cortana y se pueden crear Skills con AWS.

El algoritmo de Alexa de reconocimiento de voz distingue entre tres tipos de componentes:

- Intents: solicitud o acción principal
- Slots: argumentos del intent para dar más info.
- Utterances: palabras que transmiten lo que el usuario quiere que el asistente haga.

---

## GOOGLE ASSISTANT – GOOGLE

Lo mismo pero en los móviles Android.

---

## SIRI – APPLE

Lo mismo pero en iOS. Fue la primera en desarrollarse.

---

## OTROS

- Samsung creó Bixby pero no termina de despegar.
- Está por ahí Brina que es pa traducir entre idiomas.
- También Voice mate que es de L.G. y si no te suena pues imagínate lo malo que debe ser.
- Y uno que se llama JARVIS pero en lugar de controlar la armadura de Iron Man te controla la lavadora desde el móvil.
- Aura que es de telefónica y deben usarlo para no aguantar clientes y despedir personal.
- Dueros que es chino y se usa en china.

Y según un estudio que me acabo de inventar esto es cierto:

### Query Results

	Answered Correctly	Understood Query
Google Assistant	87.9%	100%
Siri	74.6%	99.6%
Alexa	72.5%	99.0%
Cortana	63.4%	99.4%

# T4 – METODOLOGÍA DE DESARROLLO

## T4 – INTRODUCCIÓN

**Fábula del columpio:** de la especificación al producto final, el proyecto cambia mucho.

El ciclo de vida del desarrollo de sistemas es **el proceso de creación o modificación de los sistemas, modelos y metodologías** que se usan para desarrollar estos sistemas.

La **metodología** es un marco de trabajo compuesto por **herramientas, modelos y métodos** usado para **estructurar, planificar y controlar** el proceso de desarrollo.

Las **metodologías tradicionales** tienen su eje central en la arquitectura del software, se guía por **estándares** y realiza **estudios y especificaciones controladas**.

Las **metodologías ágiles** tienen su eje central en la **producción de código**, basadas en la heurística de esta producción, **sujeta a cambios**, y proceso menos estudiado y especificado.

## METODOLOGÍAS TRADICIONALES

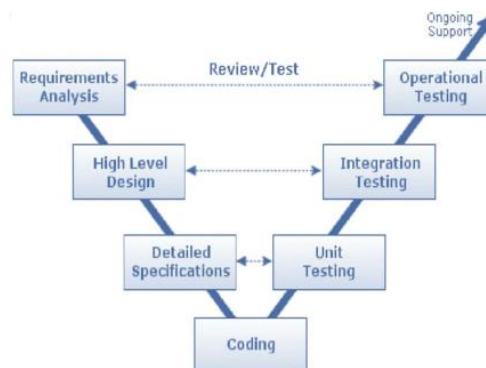
- **Modelo en cascada:** enfoque por fases ejecutadas secuencialmente.
  - Útil para proyectos con requisitos definidos, estructurado, comienza el desarrollo con rapidez y usado para proyectos grandes.
  - Por contras, es poco flexible, pocos proyectos siguen un proceso secuencial, la revisión del proyecto es complicada y necesitas reunir los requisitos desde el inicio.



- **Modelo incremental:** modelo en cascada iterativo hasta que se llega al producto final.
  - Genera software rápido en etapas tempranas, es flexible, facilidad de testing.
  - Cada fase es rígida, puede haber problemas de arquitectura.



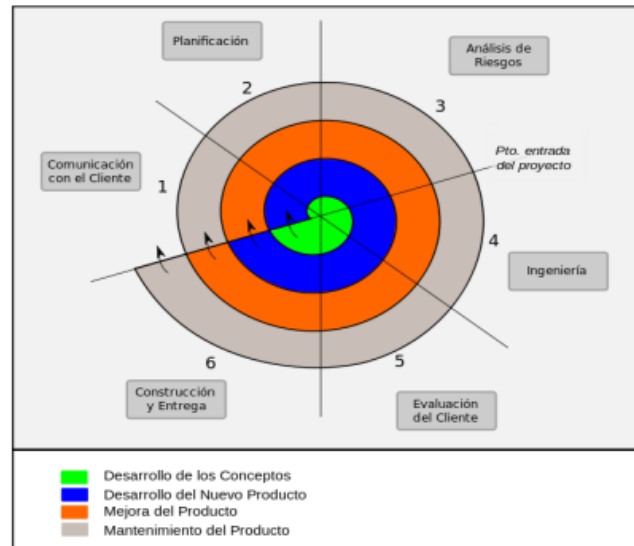
- **Modelo en V:** similar al cascada pero van de artistas. El lado izquierdo es la descomposición de necesidades, y el derecho verificación y testing.
  - Ventajas: chequeos de cada etapa, involucra al usuario, define roles de las pruebas, fácil aprendizaje.
  - Desventajas: no permite retomar etapas, sobrecostos si hay algo mal diseñado, las pruebas pueden ser caras y no efectivas.



- **Modelo en prototipo:** realizar prototipos de forma que la funcionalidad se construye encima de la versión anterior, hasta llegar al modelo definitivo.
  - Ventajas: visibilidad del producto, flexible a cambios, permite retroalimentación del cliente, reduce riesgo.
  - Desventajas: desarrollo lento, fuerte inversión, puede aumentar coste de desarrollo.



- **Modelo en espiral:** enfoque evolutivo en donde cada iteración representa una fase y está compuesto por un conjunto de actividades que se construyen una sobre la otra.
  - **Ventajas:** reacción a riesgos, permite prototipos, permite incluir métodos de desarrollo, permite puntos de control.
  - **Desventajas:** dificultad para convencer al cliente, requiere administración, dificultad para definir objetivos, complejo.



## METODOLOGÍAS ÁGILES

Son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en amoldar el proyecto a las circunstancias específicas del entorno.

Algunos de sus principios más importantes son:

- Satisfacer al cliente
- Los cambios son bienvenidos
- Entregas frecuentes
- El cliente y los desarrolladores deben trabajar juntos
- Requiere individuos motivados que trabajen cara a cara
- Busca la simplicidad
- Los equipos se auto-organizan y buscan ser lo más efectivos posibles

## METODOLOGÍA SCRUM

Basada en experiencia y toma de decisiones. Busca dividir el trabajo en bloques de pocas semanas llamados Sprints. Cada bloque tiene 4 eventos:

- Reunión
- Scrum diario
- Revisión del Sprint
- Retrospectiva del sprint

El equipo está conformado por el cliente dueño del producto, el scrum master que comunica cliente con desarrollo, y el equipo de desarrollo que crea y entrega el producto.

- Ventajas: gestión de expectativas de cliente, flexibilidad, gestión de riesgos, cada persona sabe qué hacer, programación organizada.
- Inconvenientes: difícil para equipos pequeños, requiere una definición de tareas y plazos exhaustiva, y un equipo con experiencia.

---

## METODOLOGÍA XP

Desarrollo iterativo e incremental, ideal para proyectos con requisitos imprecisos y cambiantes. Valora la comunicación, simplicidad, retroalimentación, valentía.

- Ventajas: programación organizada, eficiencia, menor tasa de errores, adaptación a cambios, pruebas continuas, cliente encargado de priorizar.
- Desventajas: solo para proyectos a corto plazo, altas comisiones, costoso e innecesario.

---

## METODOLOGÍA KANBAN

Complementaria a Scrum. Distingue entre *to do*, *doing*, *done*. Permite visualizar el flujo de trabajo, limitar el trabajo en proceso y llevar un seguimiento.

- Ventajas: organización, comodidad, fácil de aplicar, muy visual, control de flujo de trabajo, facilidad de añadir mejoras.
- Desventajas: no es una técnica específica del desarrollo software, dificultad para anticipar problemas, difícil implantación, falta de reglas.

---

## METODOLOGÍA SCRUMBAN

Flujo de trabajo Kanban con elementos Scrum.

- Ventajas: permite conocer el estado del proyecto, introduce soluciones, mejora integración, aumenta productividad.
- Inconvenientes: difícil para equipos grandes, exceso de reuniones, equipo con experiencia, inflexible a cambios.

---

## METODOLOGÍA LEAN

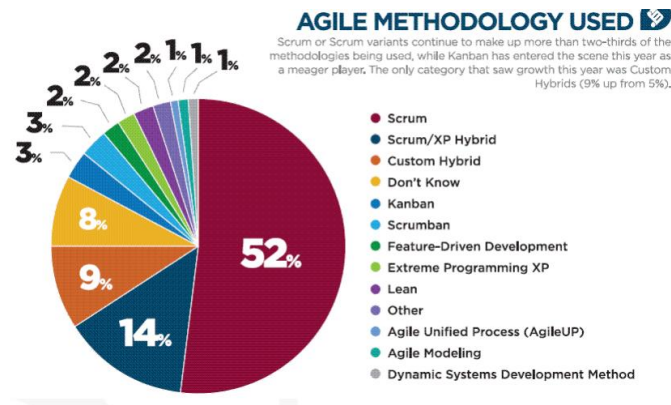
Crea equipos altamente preparados que hagan las tareas en poco tiempo. A más unidos, mejor. Ventaja del equipo sólido con programadores capaces de analizar la situación y que se ayuden entre sí.

Buscan, entre otras cosas, ampliar el aprendizaje, eliminar el desperdicio, entregar rápido, visualizar el conjunto, crear integridad y no seguir los requisitos al pie de la letra.

- Ventajas: reducir costes, menos riesgos, reducción del plazo, trabajo en equipo.
- Desventajas: dependencias de la cohesión del equipo, fuerte inversión inicial, toma de decisiones tardía puede acarrear problemas.

## ESTADÍSTICAS VARIAS

Uso:



Mayores ventajas:

- Acelerar la entrega del software
- Mejorar habilidad de gestionar prioridades
- Mejorar productividad
- Mejorar la integración negocio – IT.
- Mejorar calidad del software.

Mayores ventajas	Mayores barreras
<ul style="list-style-type: none"> <li>• Acelerar la entrega del software</li> <li>• Mejorar habilidad de gestionar prioridades</li> <li>• Mejorar productividad</li> <li>• Mejorar la integración negocio – IT.</li> <li>• Mejorar calidad del software.</li> </ul>	<ul style="list-style-type: none"> <li>• Cambiar cultura de organización</li> <li>• Resistencia al cambio</li> <li>• Complejidad del proyecto</li> <li>• Personal con habilidades adecuadas</li> </ul>

## ELECCIÓN DE METODOLOGÍA

A considerar:

- Requerimientos y su claridad
- Tiempo disponible
- Complejidad de la funcionalidad
- Nivel de conocimiento de la tecnología empleada
- Plazos
- Límites de fases del proyecto

# TEMA 5 – MODELOS DE NEGOCIO

## T5 – MODELOS DE NEGOCIO

Un modelo de negocio **describe la lógica** de cómo una organización crea, entrega y capta un valor. Es una **representación abstracta** de una organización y todo lo que conlleva.

Una empresa debe crear valor, distribuir valor y retener valor.

### MODELO CANVAS

Es un modelo de negocio que estudia:

---

#### PROPUESTA DE VALOR

Productos y servicios que una empresa ofrece a sus clientes. Contempla qué aporta y qué la diferencia de la competencia.

Busca un **problema** al que aportar una **solución** y venderla generando un **beneficio**.

---

#### SEGMENTOS DEL CLIENTE

Define los **grupos de personas u organizaciones** a los que ofrecer el servicio. Se agrupa a los clientes en segmentos y se estudia como enfocar el producto a cada segmento.

---

#### CANALES DE COMUNICACIÓN

Define por qué medio se **comunica y alcanza a los segmentos de mercado**.

Para estudiar el canal, se puede valorar los conceptos de **percepción, evaluación, compra, entrega y servicio postventa**.

---

#### RELACIONES CON LOS CLIENTES

Define el **tipo de relación** que la empresa establece con el cliente. Puede ser directa, indirecta y automatizada. También se debe estudiar coste, integración, etc.

---

#### FUENTE DE INGRESOS

Define el **dinero que la empresa genera** de cada segmento de mercado. Pueden ser ingresos puntuales o ingresos recurrentes.

---

#### RECURSOS CLAVE

Define los **activos más importantes** requeridos para el negocio. Pueden ser **físicos, intelectuales, humanos o económicos**.



---

## ACTIVIDADES CLAVE

Define las **actividades más importantes** que una empresa realiza para que funcione su modelo de negocio. Pueden ser la **producción** de un producto, la **resolución de problemas** o la proporción de **plataformas y redes** que comunican clientes, proveedores y grupos de interés.

---

## SOCIOS CLAVE

Define la **red de suplidores y socios** que hacen funcionar el modelo de negocio.

---

## ESTRUCTURA DE COSTES

Definen los **costes necesarios** para operar un modelo de negocio. Se debe:

1. Clasificar los elementos integrantes del coste
2. Periodificar los gastos del proceso de producción
3. Localización y reparto de costes (alquiler, salarios, etc)
4. Imputación de costes.

# TEMA 6 – CLOUD E IOT

## T6 – INTRODUCCIÓN

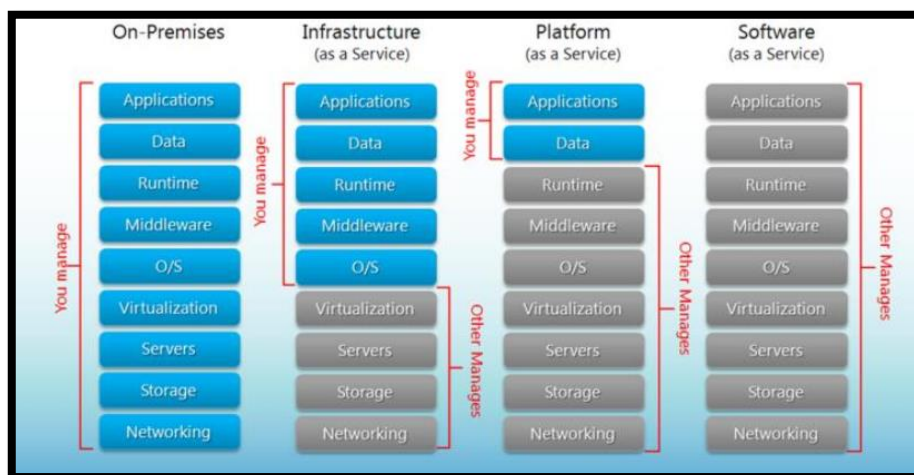
La **computación en la nube** es un paradigma que permite ofrecer servicios, recursos y software a través de la red bajo demanda.

Es **ágil, escalable, seguro y proporciona rendimiento**. Tiene por **ventajas** la **integración de servicios, rápida implementación y actualización automática**, aunque también tiene por **desventajas** la **centralización, madurez, seguridad y disponibilidad**.

Los modelos típicos de servicio son:

- Software como servicio: los consumidores acceden a aplicaciones software. Se alquila software en lugar de comprarlo.
- Plataforma como servicio: ofrece una plataforma a desarrolladores para crear aplicaciones y servicios.
- Infraestructura como servicio: proporciona un recurso en la nube, y el cliente debe hacer un self-service.

El siguiente diagrama muestra el nivel de responsabilidad del cliente y el proveedor:



## IOT – INTERNET OF THINGS

Es la interconexión digital de objetos cotidianos con internet.

Por ejemplo móviles, smartwatches, coches, cámaras de seguridad, farolas, etc.

Para comunicarse, emplean redes de comunicación como Xbee, Bluetooth, Wifi, Ethernet o móvil.

## NUBE E IOT

Se puede combinar ambos conceptos. Consiste en usar los datos recogidos por los dispositivos terminales y procesarlos en la nube para obtener conclusiones o resultados. Usado para aplicaciones de gestión, monitorización, dashboard, etc.

# TEMA 7 – INDUSTRIA 4.0

## T7 – INTRODUCCIÓN

La industria ha tenido varias revoluciones a lo largo de la historia. Actualmente nos encontramos en la cuarta revolución industrial, donde se busca integrar lo digital con los elementos de producción.

Las **fábricas** tienen una integración entre pieza, máquina y cadena de producción, donde cada máquina conoce el estado de las demás y se adapta a la secuencia de producción.

Se tiene un **nuevo paradigma de diseño**, donde el diseño asistido por ordenador es fundamental para diseño 3d, modelización, análisis, simulación, etc.

La **gestión del ciclo de vida** del producto está intervenido en todas sus fases por la computación, facilitando dicho proceso.

Se tienen **diferentes niveles de automatización** en la fabricación digital. Esta **fabricación digital permite**:

- Modelar la **planta** para diseñar la integración de la máquina, simulación, etc.
- Desarrollo simultáneo de todos los procesos relacionados a un producto.
- Reducir ciclo de puesta en marcha.
- Integración de datos del producto y sistemas de gestión de la empresa.

Se cuentan **sistemas cyberfísicos** donde la combinación de elementos computacionales y físicos permite reducción de tiempos de desarrollo, acortar procesos en la realidad, etc.

La **realidad virtual y aumentada** permite optimizar diseños, asistencia y resolución de incidencias, aumento de seguridad, reducción de tiempos de espera, asistencia a la formación...

La comunicación **M2M** es cualquier tecnología que permita a los dispositivos que se encuentren en la misma red intercambiar información de forma autónoma. Permite coordinar diferentes medios de producción.

El IOT y Cloud Computing también proporcionan soluciones en el ámbito industrial.

La **ciberseguridad industrial** son las prácticas, procesos y tecnologías diseñadas para gestionar el riesgo derivado del uso de sistemas de información relacionadas con la producción.

La obtención y análisis de datos permite conocer el comportamiento y rendimiento de las fábricas, permitiendo transformar los datos en valor añadido.

La I4.0 tiene por ventajas la información total de procesos para la mejora continua, el intercambio de datos en tiempo real, la optimización de recursos y la gestión eficaz de energía y materias primas.

# TEMA 8 – ASISTENTES VIRTUALES

## T8 - INTRODUCCIÓN

Los de telefónica está muy locos.

El proyecto **LEIA** es una inteligencia artificial que trata de entender el español.

El **phishing de Google Assistant** es un troleíto. Consiste en poner en un intent un audio diciendo que digas tu contraseña pa instalar una actualización y otro intent escuchando tu pass.

El chema tiene un proceso para detectar vulnerabilidades en IOT de 5 etapas:

1. Se detectan dispositivos vulnerables por Crawlers
2. Se recopilan contraseñas por defecto, malas configuraciones...
3. Se recopilan información de los objetivos a atacar.
4. Se hace un ataque masivo con los dispositivos infectados.
5. Se colapsan organismos o infraestructuras.

Problemitas IOTícos:

- Seguridad de infraestructura
- Privacidad y protección de datos
- Fiabilidad y protección de información

La herramienta HomePwn sirve pa hacer pentesting en IoT. Con ella se puede hacer, entre otras cosas, una captura de los datos enviados por el Airdrop de Apple.

También podemos escanear el wifi y hacernos usuarios de Air Profiling o una cosa así.

# TEMA 9 – SMART CITIES

## T9 – INTRODUCCIÓN

Una Smart city es aquella ciudad que aprovechan las TIC para aumentar la eficiencia, reducir costes y mejorar la calidad de vida de forma escalable y sostenible.

### CONTEXTO

Se estima que cada vez más gente va a vivir en la ciudad. Las ciudades tienen gran impacto en el desarrollo económico y social, y son grandes centros de producción y consumo de recursos.

Las Smart cities **protegen el medio ambiente**, buscan mejorar la **calidad de vida**, la **movilidad**, implicar al ciudadano en materia de **gobierno**, incrementar cooperación entre empresas y estado para mejorar la **economía**, y entender al **ciudadano** recogiendo y estudiando sus datos.

Favorece la **innovación**, ayuda en toma de decisiones, reduce el gasto público, mejora eficiencia, ofrece información en tiempo real...

Se debe cuidar la privacidad, estudiar la calidad de los datos antes de procesarlos, reducir la vulnerabilidad, reducir inversiones, estudiar la vida útil de las infraestructuras, etc.

---

## CIUDADES INTELIGENTES

- Londres: servicios públicos como transporte, fomenta empresas tecnológicas, London Datastore que permite ver eventos y consultar estadísticas, etc.
- Barcelona: wifi en autobuses, apps para ayudar a ciudadanos, eventos tecnológicos importantes, sensores iot.
- Alcalá de Henares: espacios integrados inteligentes para gente que no sabe abrir Google.

## OPEN DATA

Consiste en que los datos estén disponibles de forma libre sin restricciones. Trae beneficios **económicos** como crear servicios y apps basados en contenidos digitales, beneficios para el **ciudadano** en forma de servicios y aplicaciones que mejoren su calidad de vida, y para la **administración pública** ya que facilitan intercambio de datos y colaboración entre administraciones y ciudadanos.

## FIWARE

Es un ecosistema de innovación y emprendimiento abierto que fomenta el desarrollo de soluciones digitales. El objetivo es crear un ecosistema capaz de unir a los proveedores de tecnología, emprendedores, desarrolladores y dueños de los datos.

Es un proyecto de varias etapas:

1. Definir base tecnológica, casos de uso en sectores e inventario de infraestructuras públicas.
2. Desarrollo de plataforma central a través de proyecto XIFI e implantación de nodos FIWARE.
3. Mejora de infraestructura estable para ensayos a gran escala. Ampliación de base tecnológica y casos de uso.

Ventajas:

- Biblioteca de recursos y herramientas disponibles gratis.
- Soluciones TIC en sectores con alta demanda.
- Acceso a información abierta y gratuita.
- Financiación europea y soluciones innovadoras.

Ejemplos de uso: detección de multitudes en Madrid.

Revisar diapos para ver ejemplos de proyectos.

## ARDUINO

El proyecto Arduino es una plataforma de creación de electrónica basada en hardware y software libre, flexible y fácil de utilizar. La placa empleada en los proyectos es un microcontrolador en el que se pueden grabar instrucciones y posee una interfaz de entrada para escribir código desde el ordenador, o conectar otros periféricos.

El código Arduino tiene siempre la misma estructura:

```
// Puertos de pines a leer
const int LED = 5;

void setup()
{
    // Configuración de sensores
}
void loop()
{
    // Método "main" que ejecuta el loop de la aplicación
}
```

Podemos tener métodos:

```
// Declaración de métodos como en C...
int suma(int a, int b)
{
    return a + b;
}
```

Tenemos lectura y escritura analógica:

```
const int SENSOR_ANALOGICO = 7;

void setup()
{
    // PinMode establece el modo del pin a input
    pinMode(SENSOR_ANALOGICO, INPUT);
}

// Método "main" que ejecuta el loop de la aplicación
void loop()
{
    // analogRead lee la resistencia.  $V = I * R$ 
    int voltaje = (analogRead(SENSOR_ANALOGICO) * 5000);

    // cambio a escritura
    pinMode(SENSOR_ANALOGICO, OUTPUT);

    // ESCRITURA ANALÓGICA
    analogWrite(LED, 255); // output entre 0 y 255
}
```

Y lectura y escritura digital:

```
const int SENSOR_DIGITAL = 3;

void setup()
{
    // LECTURA DIGITAL
    pinMode(SENSOR_DIGITAL, INPUT); // pongo modo lectura
}

// Método "main" que ejecuta el loop de la aplicación
void loop()
{
    // leo
    int lecturaDigital = digitalRead(SENSOR_DIGITAL);
    // cambio a escritura
    pinMode(SENSOR_DIGITAL, OUTPUT);

    // ESCRITURA DIGITAL
    digitalWrite(LED, LOW);
}
```

Y utilizar Serial para comunicar Arduino y ordenador:

```
// la comunicación serie permite comunicación entre arduino y ordenador
const int PIN = 14;

void setup()
{
    Serial.begin(9600); // tasa de muestreo e inicio de comunicación puerto se
    rial
    pinMode(14, OUTPUT);
}

void loop()
{
    String entrada;
    // lectura serial
    if(Serial.available() > 0) // n° de bytes disponibles para ser leídos
    {
        entrada = Serial.readString(); // leo como string
    }

    // salida por pantalla serial
    Serial.print("hola");
    Serial.println("adios");
}
```