

INTERACCIÓN GESTUAL CON UNA VIVIENDA INTELIGENTE

LIBRERÍA GESTUAL ESPACIAL

**Master Universitario en Ingeniería del Software,
Métodos Formales y Sistemas de Información**

Universidad Politécnica de Valencia

Autor: Francisco Miguel Pardo Veses

Tutor: Joan Josep Fons i Cors

Valencia, 15 de Septiembre de 2014

ÍNDICE DE ILUSTRACIONES

Ilustración 1-1 Ventas totales de dispositivos 2014.....	10
Ilustración 1-2 Mundo conectado.....	11
Ilustración 1-3 Internet de las cosas	12
Ilustración 1-4 Interfaz de una aplicación actual para Smart home	15
Ilustración 1-5 Smart home.....	16
Ilustración 2-1 Dispositivos de una Smart Home	20
Ilustración 2-2 Arquitectura domótica centralizada	21
Ilustración 2-3 Arquitectura domótica distribuida.....	21
Ilustración 2-4 Arquitectura domótica descentralizada.....	22
Ilustración 2-5 Smart City	26
Ilustración 2-6 Puntos clave Smart City.....	27
Ilustración 2-7 Arquitectura tecnología Smart City.....	28
Ilustración 2-8 Casos de Uso Smart City.....	29
Ilustración 2-9 Protocolos de comunicaciones Smart City.....	30
Ilustración 2-10 Smart Building	34
Ilustración 2-11 Arquitectura Smart Building.....	35
Ilustración 2-12 Smart Building Ecológico.....	36
Ilustración 2-13 Termostato Nest	42
Ilustración 2-14 Homekit Apple.....	44
Ilustración 2-15 Kinect.....	45
Ilustración 2-16 Magus Free 3D	46
Ilustración 2-17 Eclipse SDK	47
Ilustración 2-18 Java Platform, Standard Edition.....	50
Ilustración 2-19 Ventas dispositivos Android Fuente: análisis de TomiAhonen Consulting	52
Ilustración 2-20 Arquitectura Android	53
Ilustración 2-21 Arquitectura XML	56
Ilustración 2-22 Arquitectura Servicios Web REST.....	58
Ilustración 3-1 Gesto Arriba	65
Ilustración 3-2 Gesto Abajo.....	65
Ilustración 3-3 Gesto Derecha.....	66
Ilustración 3-4 Gesto Izquierda	66
Ilustración 3-5 Gesto Shake.....	67
Ilustración 3-6 Gesto Boca Abajo	67
Ilustración 3-7 Gesto Boca Abajo Luminosidad.....	68
Ilustración 3-8 Gesto touch.....	69
Ilustración 3-9 Gesto abrir puerta x3	70
Ilustración 3-10 Gesto cerrar puerta x3	70
Ilustración 3-11 Gesto Puente.....	71
Ilustración 3-12 Gesto Saludo	72
Ilustración 3-13 Gesto Pincel.....	72
Ilustración 3-14 Android Control Software R2	73
Ilustración 3-15 BeHome 247.....	74
Ilustración 3-16 Loxone	75
Ilustración 3-17 Red Eyes	76
Ilustración 3-18 Magus Free 3D gestos, Launcher	77

Ilustración 3-19 Logbar.....	78
Ilustración 3-20 Especificaciones Logbar	78
Ilustración 3-21 Funciones Google Glass	79
Ilustración 3-22 VTouch	80
Ilustración 3-23 Leap Motion	81
Ilustración 3-24 Magic Control.....	82
Ilustración 3-25 Smart TV LG.....	82
Ilustración 4-1 Smart Home conectada.....	84
Ilustración 4-2 Arquitectura domótica centralizada	84
Ilustración 4-3 Interacción con una Smart Home	85
Ilustración 4-4 Sensor Acelerometro	86
Ilustración 4-5 Eje de coordenadas	87
Ilustración 4-6 Sensor de proximidad	88
Ilustración 5-1 Listado de gestos espaciales	94
Ilustración 5-2 Declaración de la clase ShakeEventListener	94
Ilustración 5-3 Método onSensorChanged de la interface SensorEventListener	95
Ilustración 5-4 Implementación del método onSensorChanged de la clase Shake	96
Ilustración 5-5 Eje de coordenadas dispositivo móvil.....	96
Ilustración 5-6 Implementación del método onSensorChanged de la clase gesto Boca Abajo ..	97
Ilustración 5-7 Diagrama de clases de la librería de gestos	98
Ilustración 5-8 Interface propia de cada gesto	99
Ilustración 5-9 Interface propia del gesto arriba	99
Ilustración 5-10 Objeto OnUpListener	99
Ilustración 5-11 Implementación del método onSensorChanged	100
Ilustración 5-12 Método que recibe los datos del gesto.....	100
Ilustración 5-13 Implementación de los métodos GET y PUT de la clase OperationRest.....	101
Ilustración 5-14 Diagrama de clases de la aplicación de ejemplo.....	102
Ilustración 5-15 Control de iluminación.....	103
Ilustración 5-16 Control de las persianas.....	104
Ilustración 5-17 Control de la calefacción.....	104
Ilustración 5-18 Control de la cafetera.....	105
Ilustración 6-1 Aplicación para iOS.....	107
Ilustración 6-2 Dispositivo Windows Phone.....	108
Ilustración 7-1 Internet of Things.....	111

ÍNDICE DEL CONTENIDO

1. INTRODUCCIÓN.....	9
Motivación	11
Planteamiento del problema	15
Objetivos del trabajo.....	15
Motivación de las asignaturas del master.....	17
2. ESTADO DEL ARTE	18
Ámbito de aplicación.....	19
Domótica	19
Smart City.....	25
Smart Building	33
Desarrollo móvil	37
Interacción espacial.....	45
Tecnologías utilizadas	47
Eclipse.....	47
Java	49
Android.....	51
XML.....	56
REST.....	57
3. PATRONES DE INTERACCIÓN GESTUAL.....	59
Definición gestual abstracta simple	65
Gesto arriba.....	65
Gesto abajo	65
Gesto derecha	66
Gesto izquierda	66
Gesto shake (agitar dispositivo).....	67
Gesto boca Abajo	67
Gesto boca abajo luminosidad	68
Gesto touch	69
Gesto abrir puerta(X3)	70
Gesto cerrar puerta(X3)	70
Definición gestual abstracta compleja.....	71
Gesto puente.....	71
Gesto saludo.....	72

Gesto pincel.....	72
Alternativas existentes en el mercado.....	73
Aplicaciones.....	73
Empresas	78
4. ARQUITECTURA DE UNA PROPUESTA DE SOLUCIÓN.....	83
Requisitos.....	84
Arquitectura general	84
Sensores utilizados.....	86
Implementación de los patrones de interacción	89
Gesto arriba.....	89
Gesto abajo	89
Gesto derecha	89
Gesto izquierda	90
Gesto shake (agitar dispositivo).....	90
Gesto boca abajo.....	90
Gesto boca abajo luminosidad.....	90
Gesto touch	91
Gesto abrir puerta(X3)	91
Gesto cerrar puerta(X3)	91
Gesto puente.....	91
Gesto saludo.....	92
Gesto pincel.....	92
5. APLICACIÓN AL ÁMBITO DE LAS VIVIENDAS INTELIGENTES	93
Implementación	94
Funcionalidades de la aplicación.....	103
Control de iluminación.....	103
Control de las persianas	104
Control de la calefacción	104
Control de encendido y apagado de la cafetera	105
6. TRABAJOS FUTUROS Y MEJORAS	106
Librería para iOS.....	107
Librería para Windows Phone	108
Librería mas flexible	109
7. CONCLUSIONES	110

8. BIBLIOGRAFÍA..... 112

1. INTRODUCCIÓN

Debido al gran auge de la utilización de dispositivos inteligentes también llamados Smartphone sobre todo desde que Apple lanzara el iPhone 3G en 2007, el mercado de la telefonía dio un giro radical para abandonar paulatinamente los teléfonos convencionales para dejar paso a una nueva generación de dispositivos que son los Smartphone. Desde ese momento las grandes compañías de tecnología vieron una gran oportunidad en el mercado con estos dispositivos. Posteriormente Google lanzo su primer dispositivo Android llamado HTC Dream que fue presentando en el año 2008. Desde entonces ambas empresas fueron comiendo terreno al que en aquella época era el líder del mercado (BlackBerry). Actualmente, Android lidera este mercado por delante de las otras opciones.

Posición	Marca	Unidades	Cuota 2014 Q1	Cuota 2013 Q4	Sistemas operativos actuales (futuros)
1 (1)	Samsung	86,8 M	30,7%	29,3%	Android, Windows (Tizen)
2 (2)	Apple	43,7 M	15,5%	17,8%	iOS
3 (3)	Huawei	18,2 M	6,4%	6,1%	Android (Tizen)
4 (4)	Lenovo	14,1 M	5,0%	4,7%	Android (Tizen)
5 (5)	LG	12,3 M	4,4%	4,6%	Android
6 (6)	ZTE	11,4 M	4,0%	3,8%	Android, Windows (Firefox)
7 (8)	Coolpad / Yulong	11,1 M	3,9%	3,7%	Android
8 (-)	Xiaomi	11,0 M	3,9%	2,6%	Android
9 (7)	Sony	9,6 M	3,4%	3,7%	Android
10 (9)	Nokia (Microsoft)	7,1 M	2,5%	2,9%	Windows, Android
	Otros	53,8 M			
	Total	282,6 M			

Ilustración 1-1 Ventas totales de dispositivos 2014 Fuente: análisis de TomiAhonen Consulting (3 junio 2014), a partir de datos de los fabricantes y del sector.

La Smart home lleva años operando en miles de viviendas y con distintos niveles de complejidad con tres objetivos básicos: la seguridad (sistemas que simulan automáticamente que la casa está habitada), la habitabilidad (controlar en la distancia la temperatura o incluso programar el horno o la lavadora) y el ahorro en el consumo energético (aspersores, calefacción, aire acondicionado). Una casa domótica supone una inversión alta, pero a largo plazo es rentable. Convertir una casa normal en un hogar inteligente, con las tecnologías más avanzadas en materia de control energético, seguridad y comunicaciones, cuesta entre 1.500 y 3.000 euros, lo que equivale a entre el 1,5 y el 2% del coste de las instalaciones de una vivienda. Y la domótica, esos sistemas capaces de automatizar todos los servicios de una vivienda, va de la mano de las nuevas tecnologías y de los nuevos dispositivos de comunicación. Partiendo de esta premisa, los Smartphone sirven de gran utilidad en este campo, ya que permiten controlar toda la vivienda domótica sin tener que estar físicamente en la vivienda. Y debido también a la posibilidad que brindan los sistemas operativos de los móviles a los desarrolladores para crear aplicaciones propias, abre un gran abanico de posibilidades en este campo, ya que no existen dos aplicaciones iguales y cada una de ellas ofrece funcionalidades que la otra no aporta.

Motivación

La domótica busca el poder tener el control de todos los dispositivos que componen una vivienda. Es una tendencia de aumentar el equipamiento electrónico de los hogares. La domótica es una disciplina que ha crecido de forma lenta desde los años 90 que en la actualidad ha cobrado un nuevo impulso. Hasta esa fecha los dispositivos habían permanecido aislados realizando labores específicas que tenían asignadas. Con esta disciplina se impone la necesidad de conectar estos dispositivos de la vivienda entre si y otorgarse salida a internet para obtener servicios más avanzados.

La evolución histórica de la domótica proviene desde principios del siglo XX:

- 1935: USA. Monitorización temperaturas y mecanismos (motores, compresores, etc.)
- 1952 (Casa Blanca): se monitorizaron varios cientos de termostatos, válvulas y relés para supervisión de las instalaciones de luz y aire acondicionado
- 1965: introducción de computadores digitales. Primero supervisión y luego control digital directo
- 1970: minicomputador. Primeros sistemas de control distribuido (sólo grandes edificios)
- 1973: microprocesador y crisis energética. Ahorro energético: climatización, iluminación, ascensores.
- 1980: Disminución de precios en Electrónica. Productos comerciales (Johnson, Honeywell, Satchell, JEL, Sauter, Danfoss, Centra, Landis & Gyr, etc.).
- 1985: generalización uso ordenadores personales. Precio. Sistemas pequeños para residencias.
- 1990:
 - Auge comunicaciones. Redes de ordenadores Buses de campo
 - Distintos puestos de tratamiento (incluso en lugares distintos) pero con intercambios de datos en tiempo real
 - Sensores y elementos finales inteligentes

La motivación para realizar este proyecto es sobre todo el de obtener un mundo conectado, que está relacionado con el objetivo principal del trabajo, que pretende controlar todos los dispositivos conectados en este caso de una Smart Home, pero que se podría extender para controlar mediante gestos espaciales cualquier dispositivo de un Smart Building o una Smart City, en donde todos los elementos cotidianos están interconectados.



Ilustración 1-2 Mundo conectado

El internet de las cosas se encuentra todavía en una edad muy temprana de desarrollo, es por ello que no existe una estandarización de este término, pero una aproximación de este término se encuentra en un artículo de 2009 en el diario RFID que fue realizado por Kevin Ashton:

“Los ordenadores actuales—y, por tanto, Internet—son prácticamente dependientes de los seres humanos para recabar información. Una mayoría de los casi 50 peta bytes (un peta byte son 1,024 terabytes) de datos disponibles en Internet fueron inicialmente creados por humanos—a base de teclear, presionar un botón, tomar una imagen digital o escanear un código de barras. Los diagramas convencionales de Internet... dejan fuera a los routers más importantes de todos, las personas. El problema es que las personas tienen tiempo, atención y precisión limitados —lo que significa que no son muy buenos a la hora de conseguir información sobre cosas en el mundo real. Y eso es un gran obstáculo. Somos cuerpos físicos, al igual que el medio que nos rodea... No podemos comer bits, ni quemarlos para resguardarnos del frío, ni meterlos en tanques de gas. Las ideas y la información son importantes, pero las cosas cotidianas tienen mucho más valor. Aunque, la tecnología de la información actual es tan dependiente de los datos escritos por personas que nuestros ordenadores saben más sobre ideas que sobre cosas. Si tuviéramos ordenadores que supieran todo lo que tuvieran que saber sobre las “cosas”, mediante el uso de datos que ellos mismos pudieran recoger sin nuestra ayuda, nosotros podríamos monitorizar, contar y localizar todo a nuestro alrededor, de esta manera se reducirían increíblemente gastos, pérdidas y costes. Sabríamos cuando reemplazar, reparar o recuperar lo que fuera, así como conocer si su funcionamiento estuviera siendo correcto. El Internet de las Cosas tiene el potencial para cambiar el mundo tal y como hizo la revolución digital hace unas décadas. Tal vez incluso hasta más.”

Este proyecto según Cisco, que está involucrada en el proyecto del internet de las cosas, costaría 19 billones de dólares

Algunas empresas han tratado de encontrar soluciones posibles y eficaces al problema planteado:

- **AT&T:** “Vida digital” es la solución más conocida. En su página web cuenta con todo tipo de medidas domóticas que se pueden controlar a través de una aplicación del teléfono móvil.
- **Muzzley:** Utiliza una sola aplicación con la que poder acceder a cientos de dispositivos gracias a que los fabricantes están comenzando a unirse a su proyecto de APIs con el fin de proporcionar una única solución para controlar los dispositivos personales.
- **My shortcut:** Es una propuesta que incluye un conjunto de dispositivos que permiten al usuario establecer una interacción con la aplicación, al estilo Siri. Mediante el uso de comandos de voz, se le ofrece la posibilidad al usuario de utilizar las herramientas más comunes del Internet de las Cosas.
- **Realtek:** “IoT my things” es también una aplicación que pretende controlar un sistema cerrado de dispositivos de Realtek tales como sensores.

Otro de los objetivos principales de este proyecto, es el de desechar la utilización de sistemas informáticos para el control de una Smart Home, Smart Building o Smart City. Es el

llegar a la conclusión que es viable la gestión de un sistema inteligente sin recurrir a la interacción física con un dispositivo electrónico. Es decir sin tener que utilizar pantallas, teclados o ratones, simplemente con gestos espaciales aunque sea con un dispositivo inteligente pero puede extrapolarse a una interacción sin este dispositivo. Para este proyecto se ha utilizado un Smartphone pero debido a que se utiliza un sistema abierto como Android, puede usarse en cualquier otro dispositivo que contenga este sistema operativo, como Tablets o Smartwatches.

Planteamiento del problema

El problema planteado es el que actualmente las aplicaciones de domótica, se sirve de un enfoque muy clásico en cuanto a interacción, ya que abarrotan las pantallas con numerosos botones que da como resultado una interfaz muy sobrecargada y puede cansar al usuario y como consecuencia que deje de utilizar la aplicación. Por ello cuando se planteó el problema en cuestión se optó por la sencillez y minimizar en la medida de lo posible todos aquellos elementos prescindibles que podían ser suplantados por gestos naturales o uso de sensores. Como ejemplo más claro, tenemos el de encender una bombilla o apagarla, en nuestro caso práctico se ha optado por eliminar todos aquellos botones innecesarios y sustituirlos por un ligero movimiento para apagar o encender. O por otro caso el que según la luz ambiental de la habitación que la aplicación decida por el usuario la graduación óptima de luz, siempre claro dejando la opción al usuario de graduarla manualmente.



Ilustración 1-4 Interfaz de una aplicación actual para Smart home

Objetivos del trabajo

Los objetivos que abarca este trabajo es el de poder crear una aplicación totalmente funcional que aproveche todo el potencial que proporcionan los dispositivos móviles en cuanto a la utilización de los sensores que disponen y la utilización de gestos naturales de movimiento para su uso. Para ello se han realizado casos prácticos donde se utilizan dichas funcionalidades para comprobar su usabilidad y correcto funcionamiento en futuras aplicaciones. Principalmente los objetivos del trabajo era explorar un su totalidad aquellas situaciones donde el uso de dichas funcionalidades fueran necesarias y útiles.

Solución propuesta

La aplicación propuesta para el desarrollo de esta tesina, abarca el área de poder controlar totalmente o casi totalmente una Smart home desde tu Smartphone o Tablet basada en el S.O de Google (Android). Esta aplicación utiliza los sensores del dispositivo y gestos en el propio dispositivo para controlar los distintos dispositivos que compone la vivienda inteligente. La aplicación se conectara a un servidor localizado dentro de la vivienda que tendrá todos los dispositivos registrados y desde donde se enviara la información al dispositivo dependiendo de la consulta que se realice en el mismo. La aplicación permitirá al usuario también poder monitorizar la casa desde cualquier sitio sin tener que estar el usuario dentro de la vivienda. Cada vez que un usuario instalara un nuevo dispositivo la aplicación se descargaría la nueva configuración para poder controlarlo. Debido a que las aplicaciones de referencia sobrecargan demasiado las pantallas y por consiguiente el usuario puede sentirse abrumado entre tanta interfaz, se optó por hacer uso de los gestos naturales para interactuar con los objetos. De esta forma el usuario podría gestionar en tu totalidad todos los objetos que compondrían sus vivienda con un conjunto de gestos predefinidos que realizaran una o otra acción dependiendo de la función asociada a dicho gesto. Lo que se plantea en este trabajo es saber si puede o no ser viable controlar una Smart Home utilizando simplemente gestos espaciales con el dispositivo sin tener que recurrir a interfaces sobrecargadas que obligan al usuario a tener que estar pendiente de la pantalla del dispositivo continuamente para realizar las acciones. El usuario cómodamente, elegiría el objeto que desea gestionar y mediante gestos espaciales podría realizar todas las funciones permitidas en él.

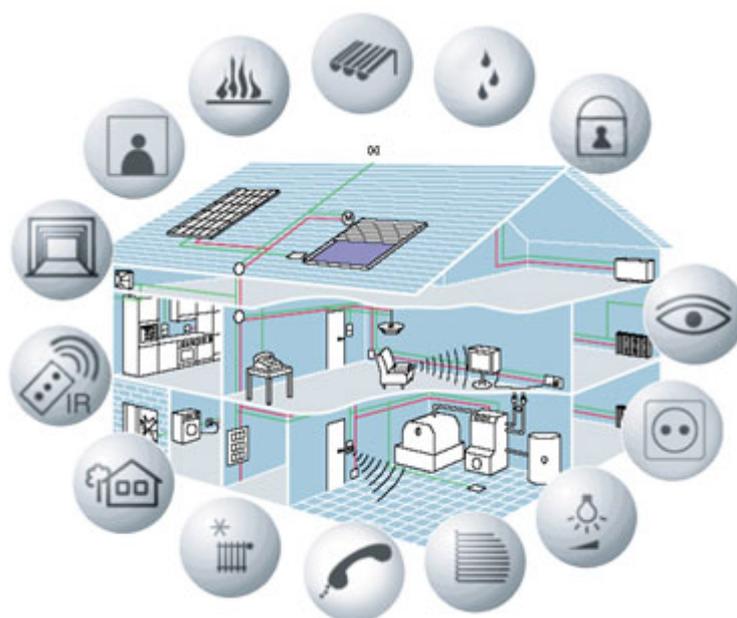


Ilustración 1-5 Smart home

Motivación de las asignaturas del master

Durante la realización del master se han cursado muchas asignaturas, pero las más relacionadas al trabajo que se ha realizado son las siguientes:

- MODELADO, DISEÑO E IMPLEMENTACIÓN DE SERVICIOS WEB.
- TECNOLOGIA SOFTWARE PARA AMBIENTES WEB
- DESARROLLO DE SISTEMAS UBICUOS E INTELIGENCIA AMBIENTAL
- TECNICAS HC(HUMAN-COMPUTER INTERACTION) EN EL ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN
- DESARROLLO DE APLICACIONES JAVA

Todas estas asignaturas comentadas, son las que más relación tienen, ya que el trabajo se fundamenta en el desarrollo de una librería de gestos con las que el usuario puede interactuar de forma natural con su entorno y la comunicación con su entorno se realiza mediante el consumo de unos servicios web que son los intermediarios entre nosotros y en este caso una Smart home.

2. ESTADO DEL ARTE

Ámbito de aplicación

Domótica

La Domótica es la automatización y control centralizado y/o remoto de aparatos y sistemas eléctricos y electrotécnicos en la vivienda. Los objetivos principales de la domótica es aumentar el confort, ahorrar energía y mejorar la seguridad.

El concepto domótica se refiere a la automatización y control (encendido / apagado, apertura / cierre y regulación) de aparatos y sistemas de instalaciones eléctricas y electrotécnicos (iluminación, climatización, persianas y toldos, puertas y ventanas motorizados, el riego, etc.) de forma centralizada y/o remota. El objetivo del uso de la domótica es el aumento del el confort, el ahorro energético y la mejora de la seguridad personal y patrimonial en la vivienda.

Los servicios que ofrece la Smart home se pueden agrupar según cinco aspectos o ámbitos principales:

- **Ahorro energético:** El ahorro energético no es algo tangible, sino un concepto al que se puede llegar de muchas maneras. En muchos casos no es necesario sustituir los aparatos o sistemas del hogar por otros que consuman menos sino una gestión eficiente de los mismos.
- **Confort:** El confort conlleva todas las actuaciones que se puedan llevar a cabo que mejoren el confort en una vivienda. Dichas actuaciones pueden ser de carácter tanto pasivo, como activo o mixtas.
- **Seguridad:** Consiste en una red de seguridad encargada de proteger tanto los bienes patrimoniales como la seguridad personal.
- **Comunicaciones:** Son los sistemas o infraestructuras de comunicaciones que posee el hogar.
- **Accesibilidad:** Se incluyen las aplicaciones o instalaciones de control remoto del entorno que favorecen la autonomía personal de personas con limitaciones funcionales, o discapacidad. El concepto "diseño" para todos es un movimiento que pretende crear la sensibilidad necesaria para que al diseñar un producto o servicio se tengan en cuenta las necesidades de todos los posibles usuarios, incluyendo las personas con diferentes capacidades o discapacidades, es decir, favorecer un diseño accesible para la diversidad humana. La inclusión social y la igualdad son términos o conceptos más generalistas y filosóficos. La domótica aplicada a favorecer la accesibilidad es un reto ético y creativo pero sobre todo es la aplicación de la tecnología en el campo más necesario, para suplir limitaciones funcionales de las personas. El objetivo no es que las personas con discapacidad puedan acceder a estas tecnologías, porque las tecnologías en si no son un objetivo, sino un medio. El objetivo de estas tecnologías es favorecer la autonomía personal. Los destinatarios de estas tecnologías son todas las personas, ya que por enfermedad o envejecimiento.

Además, la domótica facilita la creación de escenarios que se complementan con los avances en la Sociedad de la Información:

- **Comunicaciones:** Transmisión de voz y datos, incluyendo textos, imágenes, sonidos (multimedia) con redes locales (LAN) compartiendo acceso a Internet, recursos e intercambio entre todos los dispositivos, acceso a nuevos servicios de telefonía sobre IP, televisión digital, televisión por cable, diagnóstico remoto, videoconferencias, etc.
- **Mantenimiento:** Con capacidad de incorporar el telemantenimiento de los equipos.
- **Ocio y tiempo libre:** Descansar y divertirse con radio, televisión, multi-room, cine en casa, videojuegos, captura, tratamiento y distribución de imágenes fijas (foto) y dinámicas (vídeo) y de sonido (música) dentro y fuera de la casa, a través de Internet, etc.
- **Salud:** Actuar en la sanidad mediante asistencia sanitaria, consultoría sobre alimentación y dieta, telecontrol y alarmas de salud, medicina monitorizada, cuidado médico, etc.
- **Compra:** Comprar y vender desde la casa, etc. Finanzas: Gestión del dinero y las cuentas bancarias mediante la tele banca, consultoría financiera....
- **Aprendizaje:** Aprender y reciclarse mediante la tele-enseñanza, cursos a distancia...
- **Actividad profesional:** Trabajar total o parcialmente desde el hogar, posibilidad viable para ciertas profesiones (teletrabajo) , etc.
- **Ciudadanía:** Gestiones múltiples con la Administración del Estado, la Comunidad Autónoma y el Municipio, voto electrónico, etc.
- **Acceso a información:** Museos, bibliotecas, libros, periódicos, información meteorológica, etc.



Ilustración 2-1 Dispositivos de una Smart Home

La Arquitectura de los sistemas de domótica hace referencia a la estructura de su red. La clasificación se realiza en base de donde reside la “inteligencia” del sistema domótico. Las principales arquitecturas son:

- **Arquitectura centralizada:** En un sistema de domótica de arquitectura centralizada, un controlador centralizado, envía la información a los actuadores e interfaces según el programa, la configuración y la información que recibe de los sensores, sistemas interconectados y usuarios.

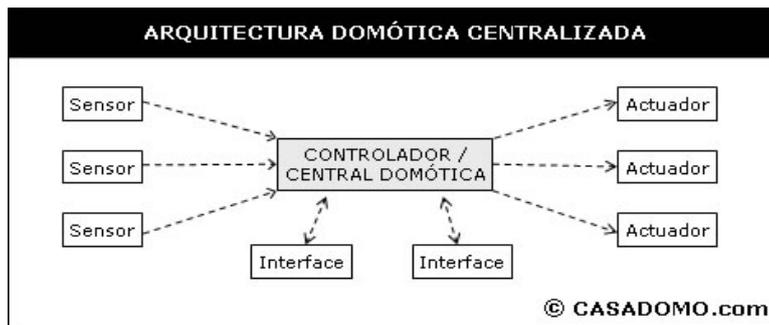


Ilustración 2-2 Arquitectura domótica centralizada

- **Arquitectura distribuida:** En un sistema de domótica de arquitectura distribuida, cada sensor y actuador es también un controlador capaz de actuar y enviar información al sistema según el programa, la configuración, la información que capta por si mismo y la que recibe de los otros dispositivos del sistema.

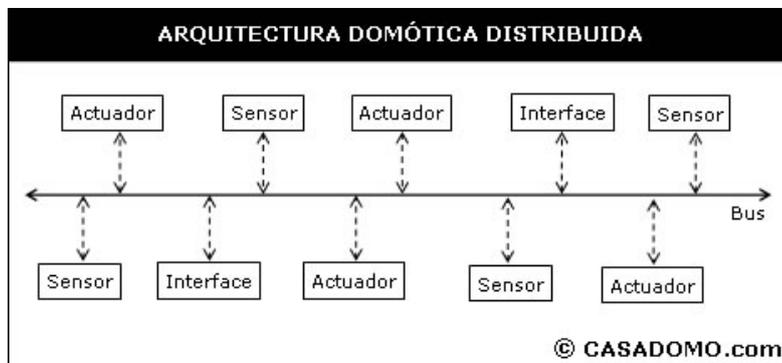


Ilustración 2-3 Arquitectura domótica distribuida

- **Arquitectura descentralizada:** En un sistema de domótica de Arquitectura Descentralizada, hay varios controladores, interconectados por un bus, que envía información entre ellos y a los actuadores e interfaces conectados a los controladores, según el programa, la configuración y la información que recibe de los sensores, sistemas interconectados y usuarios.

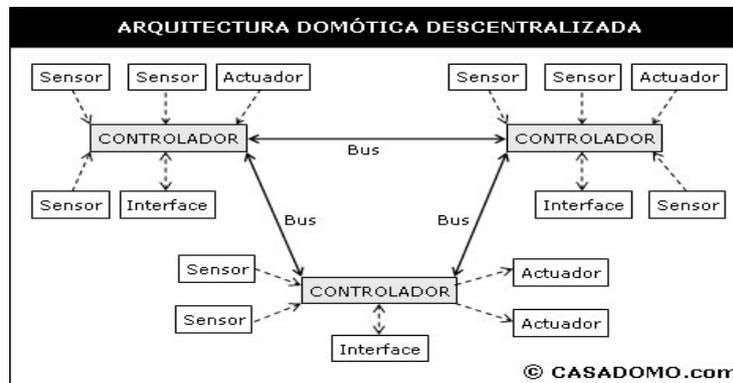


Ilustración 2-4 Arquitectura domótica descentralizada

En lo referente a los sistemas de comunicación que se utiliza en la vivienda domótica, pueden clasificarse en dos categorías:

- **Cableados:** Sistemas en los que todos los sensores, actuadores y controladores están conectados entre sí con cables. Se pueden catalogar los sistemas cableados en función de si el cableado que se utiliza es exclusivo o compartido con otro sistema:
 - **Cableado exclusivo:** en los hogares normalmente no existe ningún otro cableado que no sea el de la tensión eléctrica a 220V o el telefónico, excepto en obras nuevas que pueden disponer de cableado para transmisión de música. Cuando hablo de cableado exclusivo me refiero a tener un sistema de cableado solo para el sistema domótico, sin compartirlo con ningún otro sistema. Esto implica cablear toda la vivienda para añadir un nuevo bus de comunicaciones que permita el envío de datos con los dispositivos domóticos. Esto resulta práctico cuando se aplican a viviendas de obra nueva mientras se construyen, ya que no supone un gran coste adicional añadir tres cableados en vez de dos.
 - **Cableado compartido:** Cuando no hay posibilidad de cablear la vivienda de nuevo se puede optar por esta solución, que consiste en utilizar un cableado ya existente y compartirlo. Concretamente se suele utilizar la línea eléctrica, la tecnología se llama *Power Line Carrier (PLC)*, funciona mediante la modulación de una onda portadora cuya frecuencia oscila entre los 20 y 200 kHz inyectada directamente en el cableado eléctrico.

- **Inalámbricos:** Cuando no es posible cablear la vivienda y tampoco utilizar la tecnología PLC se puede optar por utilizar tecnologías inalámbricas como WiFi, Bluetooth, infrarrojos o radiofrecuencia. Estas tecnologías permiten que el dispositivo domótico no necesite estar en un lugar fijo, ya que puede comunicarse con el sistema desde cualquier lugar dentro del alcance del receptor.

Los protocolos de comunicación son los procedimientos utilizados por los sistemas de domótica para la comunicación entre todos los dispositivos con la capacidad de “controlador”. Existen una gran variedad de protocolos, algunos específicamente desarrollados para la domótica y otros protocolos con su origen en otros sectores, pero adaptados para los sistemas de domótica. Los protocolos pueden ser del tipo estándar abierto (uso libre para todos), estándar bajo licencia (abierto para todos bajo licencia) o propietario (uso exclusivo del fabricante o los fabricantes propietarios).

- **X10:** Es una tecnología de control desarrollado por la compañía americana *Echelon Corp.* Puede utilizar una gran variedad de medios de transmisión. Aire, par trenzado coaxial, fibra o red eléctrica. Necesita la instalación de una serie d nodos a lo largo de la red, estos gestionan los diferentes sensores y actuadores. La configuración de estos nodos se tiene que realizar utilizando la herramienta que ofrece la propia empresa, *Lonmaker*. Se trata de una tecnología muy robusta y fiable, especialmente indicada para la automatización industrial, que es el ámbito de donde proviene. Aunque su uso se ha extendido a la domótica.. Puede funcionar correctamente para la mayoría de los usuarios domésticos. Es de código abierto y el más difundido. Poco fiable frente a ruidos eléctricos.
- **KNX/EIB:** Bus de Instalación Europeo con más de 20 años y más de 100 fabricantes de productos compatibles entre sí.
- **ZigBee:** Se trata de un protocolo de comunicaciones inalámbrico, similar a Bluetooth y basado en el estándar para redes inalámbricas de área personal *WPAN IEE_802.15.4*. Surgió de una alianza sin ánimo de lucro entre más de 200 empresas, con el objetivo de conseguir el desarrollo de una tecnología inalámbrica de bajo coste. Se pensó especialmente en la utilización en domótica debido a su bajo consumo, su sistema de comunicación vía radio y su fácil integración, ya que se pueden integrar con poquísima electrónica. Se le pueden conectar un máximo de 64000 nodos, consume 30ma transmitiendo, 3ma en reposo y transmite a una velocidad de 250kbps. Estas características hacen que este protocolo sea ideal para domótica, ya que ésta necesita enviar pequeños paquetes de datos. El hecho que ZigBee pueda situarse en modo

“reposo” resulta muy útil ya que puede pasar mucho rato entre cada acción del usuario.

- **OSGi:** Open Services Gateway Initiative. Especificaciones abiertas de software que permita diseñar plataformas compatibles que puedan proporcionar múltiples servicios. Ha sido pensada para su compatibilidad con Jini o UPnP.
- **LonWorks:** Plataforma estandarizada para el control de edificios, viviendas, industria y transporte.
- **Universal Plug and Play (UPnP):** Arquitectura software abierta y distribuida que permite el intercambio de información y datos a los dispositivos conectados a una red.
- **Modbus:** Protocolo abierto que permite la comunicación a través de RS485 (Modbus RTU) o a través de Ethernet (Modbus TCP). Es el protocolo libre que lleva más años en el mercado y que dispone de un mayor número de fabricantes de dispositivos, lejos de desactualizarse, los fabricantes siguen lanzando al mercado dispositivos con este protocolo continuamente.

Smart City

Una Smart City se define como aquella ciudad que usa las tecnologías de la información y las comunicaciones para hacer que tanto su infraestructura crítica, como sus componentes y servicios públicos ofrecidos sean más interactivos, eficientes y los ciudadanos puedan ser más conscientes de ellos. Es una ciudad comprometida con su entorno, tanto desde el punto de vista medioambiental como en lo relativo a los elementos culturales e históricos.

El concepto de *Smart City* camina de la mano con el Internet de las Cosas, ya que ambos tienen en las comunicaciones M2M su fundamento y adelantan, con sus aplicaciones y usos, la que está llamada a ser la Internet del futuro. Una Internet que no solo consistirá en la conexión de cada vez más personas, sino en el planteamiento de un mundo digital en el que, idealmente, todo podrá estar conectado. Desde dispositivos, hasta objetos del mundo físico que habitualmente no disponían de esta conectividad; es el caso de los elementos urbanos, de los edificios, los coches, los electrodomésticos, los contadores, etc. y en general todo aquello que haya que gestionar o controlar.

En este contexto, una *Smart City* es un sistema complejo, un ecosistema en el que intervienen múltiples agentes, en el que coexisten muchos procesos íntimamente ligados. Además, la *Smart City* se convierte en una plataforma digital que permite maximizar la economía, la sociedad, el entorno y el bienestar de las ciudades, y facilita el cambio hacia un comportamiento más sostenible entre todos los agentes: usuarios, empresas y Administración.

Por otro lado, permite habilitar nuevos modelos de negocio, constituyendo una excelente plataforma para la innovación en su entorno.

De manera descriptiva, una *Smart City* es un espacio urbano con infraestructuras, redes y plataformas inteligentes, con millones de sensores y actuadores, dentro de los que hay que incluir también a las propias personas y a sus teléfonos móviles.

Una *red de sensores*, refiere a una tecnología específica que ayuda a la creación de **ciudades inteligentes**. Su objetivo es el de crear una red de nodos con captadores inteligentes, con los que se puedan medir parámetros diversos que ayuden a una mejor gestión del territorio. Todos los datos recogidos son transmitidos en tiempo real, e inmediatamente disponibles tanto para autoridades como para ciudadanos de a pie.

Es así por ejemplo, que los ciudadanos podrán supervisar el nivel de polución del aire presente en una determinada calle o en un determinado lugar, o incluso recibir un alerta cuando por ejemplo el nivel de radiaciones sobrepase un determinado límite. Este tipo de sistemas también ofrecerán a las autoridades la posibilidad de optimizar la irrigación de plazas y parques, o la iluminación de ciertas zonas de la ciudad. Además, con este tipo de tecnología, las fugas de agua potable o filtraciones de aguas servidas podrán ser más fácilmente detectadas, así como también se podrán elaborar planos con el estado de la polución sonora sobre el territorio. Igualmente, los contenedores de basura también podrán ser más

inteligentes, al tener captore que desencadenen una alarma cuando se encuentren cercanos al límite de su capacidad.

Asimismo, el tráfico en las calles y en las autorutas también podrá ser monitorizado, a efectos por ejemplo de modificar en forma dinámica, la iluminación de las zonas concernidas. Además, tráfico local en búsqueda de estacionamiento podrá ser reducido, gracias a sistemas que detecten los lugares libres más próximos, y gracias también a la aplicación dinámica de peaje de estacionamiento que se ajustaría por demanda;⁴⁶ de esta forma, los automovilistas serían informados en tiempo real, y rápidamente podrían estacionar, economizando así tiempo, carburante, y ocupación de vías de circulación. Las consecuencias del uso de sistemas como los que acaban de señalarse, además de los aspectos económicos, tendrán la ventaja de reducir la polución y los embotellamientos, lo que traerá aparejado un mejoramiento de la calidad de vida.

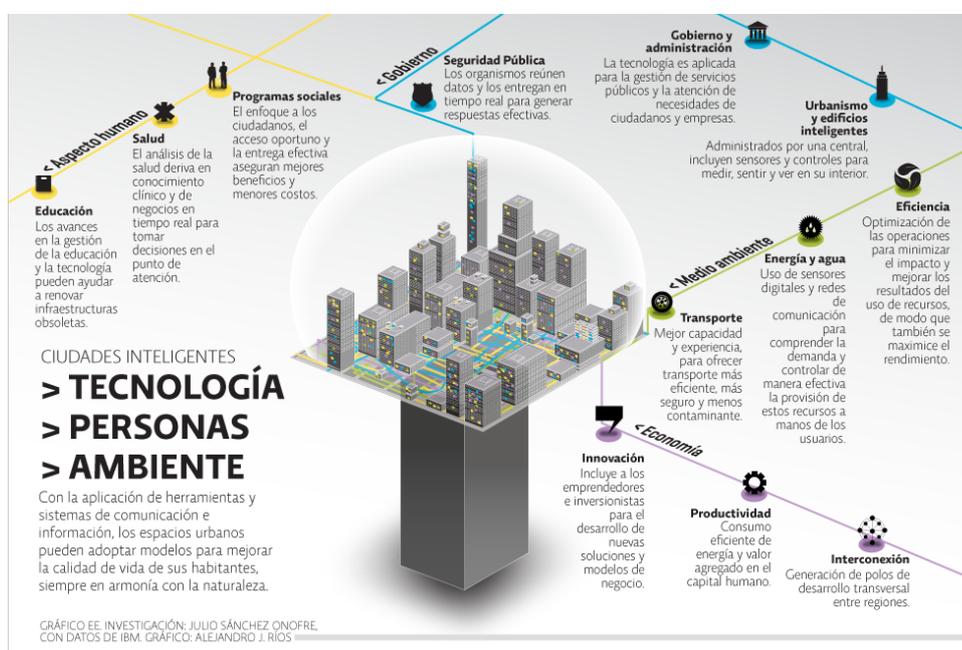


Ilustración 2-5 Smart City

El concepto de Smart City se articula en base a cuatro ideas esenciales

- Cuestiones ambientales y restricciones energéticas
- Comunicación fluida de los actores entre sí: colectividades, ciudadanos, empresas e instituciones
- Uso compartido de bienes y servicios, con una activa participación de los usuarios en la concepción de productos, servicios, y modalidades operativas, y renunciando en algunos casos a la propiedad y uso individual;
- Integración de las nuevas tecnologías de la información y de la comunicación, la robótica y los sistemas inteligentes de transporte, que potencian el funcionamiento en red; la modificación de la matriz energética a favor de las energías renovables, y el cambio de comportamiento y usos por parte de los ciudadanos

Pero que haría falta para definir una Smart City? Una de las clasificaciones más extendidas se basa en estos seis puntos:

- Smart Economy: una economía inteligente,
- Smart Mobility: una movilidad inteligente,
- Smart Environment: un medioambiente inteligente,
- Smart People: una sociedad comprometida y participativa,
- Smart Living: una constante mejora de la calidad de vida,
- Smart Governance: una gestión de los servicios inteligente.

A su vez, estos seis puntos se basan, respectivamente, en las teorías de la competitividad regional, el transporte, los recursos naturales, el capital humano y social, la calidad de vida y la participación de los ciudadanos en el gobierno de las ciudades. Y, englobando a varios de ellos, las TIC tienen mucho que aportar. Por tanto, la cuestión es si lo que realmente estamos buscando son ciudades más inteligentes o bien, al contrario, ciudades más competitivas. Veremos que, en realidad, no son excluyentes.



Ilustración 2-6 Puntos clave Smart City

Una ciudad puede ser definida como “inteligente” cuando las inversiones en capital humano y social, ya sean tradicionales como infraestructuras o modernas como las tecnologías de la información, buscan un desarrollo económico sostenible y una mejor calidad de vida, a través de una gestión racional de los recursos naturales y de la participación de todos los entes que interactúan en una población.

Un desarrollo de una Smart City generará grandes oportunidades de mercado para el sector TIC, en particular en el área de las tecnologías M2M. Los estándares están llamados a desempeñar un papel esencial en la evolución de ese mercado, ya que cualquier retraso en el desarrollo de los mismos impactará necesariamente en el ritmo del crecimiento de las Smart Cities.

Las principales organizaciones de estandarización están trabajando en el campo de los estándares M2M. En la arquitectura tecnológica de la Smart City, las tecnologías M2M juegan un papel central, al constituir el núcleo de la miríada de sensores y actuadores que permiten recoger toda la información de la ciudad y llevar a la práctica las decisiones adoptadas por las plataformas de gestión de los servicios. Estos elementos necesitan comunicarse entre sí y con los gateways M2M, para posteriormente acceder a las redes de comunicaciones de la ciudad, sean estas fijas o móviles, y a través de éstas a las plataformas de gestión de servicios donde, una vez agregada y procesada la información, se adoptan las decisiones de gestión de los servicios prestados por la Smart City

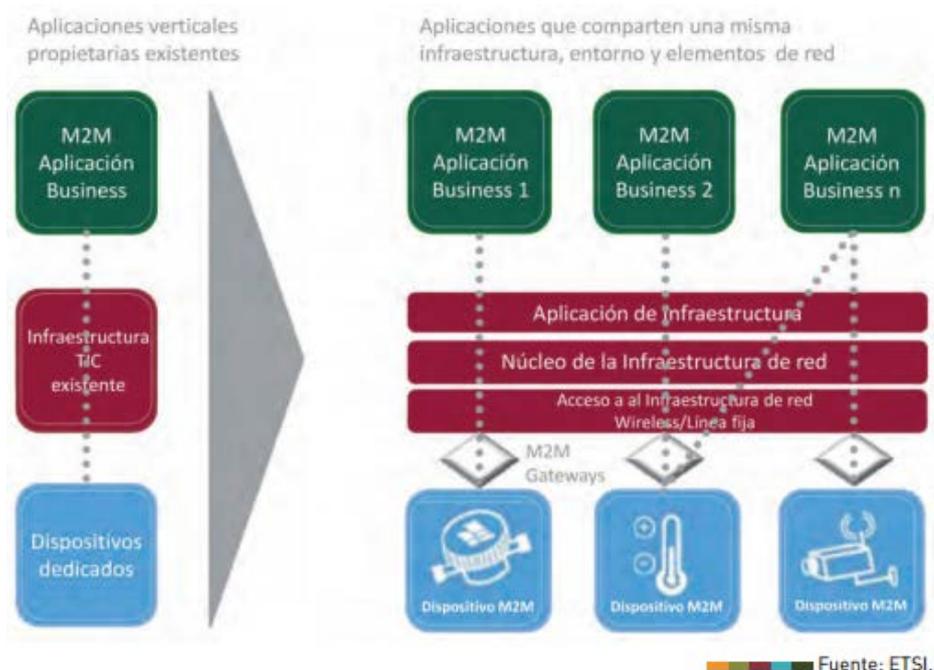


Ilustración 2-7 Arquitectura tecnología Smart City

Como se muestra en la ilustración anterior, se distingue entre el dominio de los dispositivos M2M, que permite la comunicación entre estos, y el dominio de la red, a través de la cual toda la información fluye hacia el dominio de las aplicaciones de negocio. El principal campo de estandarización es el del dominio M2M, en el que a día de hoy se dan multitud de soluciones, muchas de ellas propietarias. La iniciativa OneM2M58, recientemente puesta en marcha59, agrupa a más de 240 instituciones y empresas, incluyendo los principales organismos de estandarización como el propio ETSI e importantes empresas españolas como Telefónica. Esta iniciativa ha nacido con el objetivo de eliminar una de las principales barreras que ha venido ralentizando el desarrollo de la Internet de las Cosas: la falta de estándares.

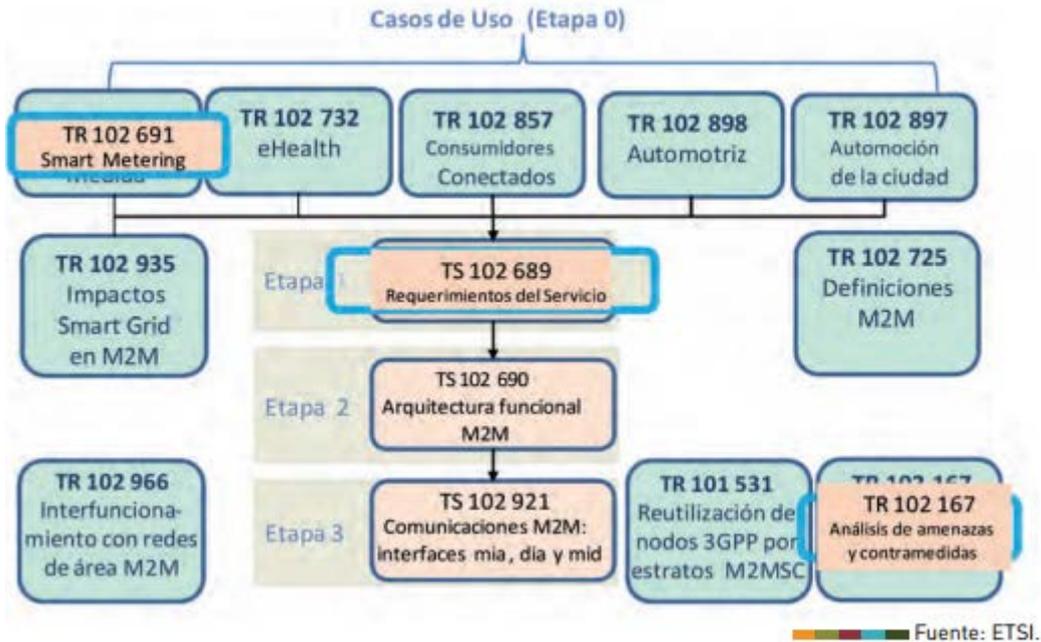


Ilustración 2-8 Casos de Uso Smart City

Los tres grupos de especificaciones citados en el diagrama, los cuales fueron publicados en febrero de 2012, cubren la arquitectura de dispositivos M2M, gateways e interfaces, aplicaciones, tecnologías de acceso y también la capa de servicios:

- ETSI TS 102 68961: requisitos generales, funcionales, de gestión y de seguridad para M2M.
- ETSI TS 102 69062: arquitectura funcional M2M.
- ETSI TS 102 92163: interfaces de comunicaciones M2M.

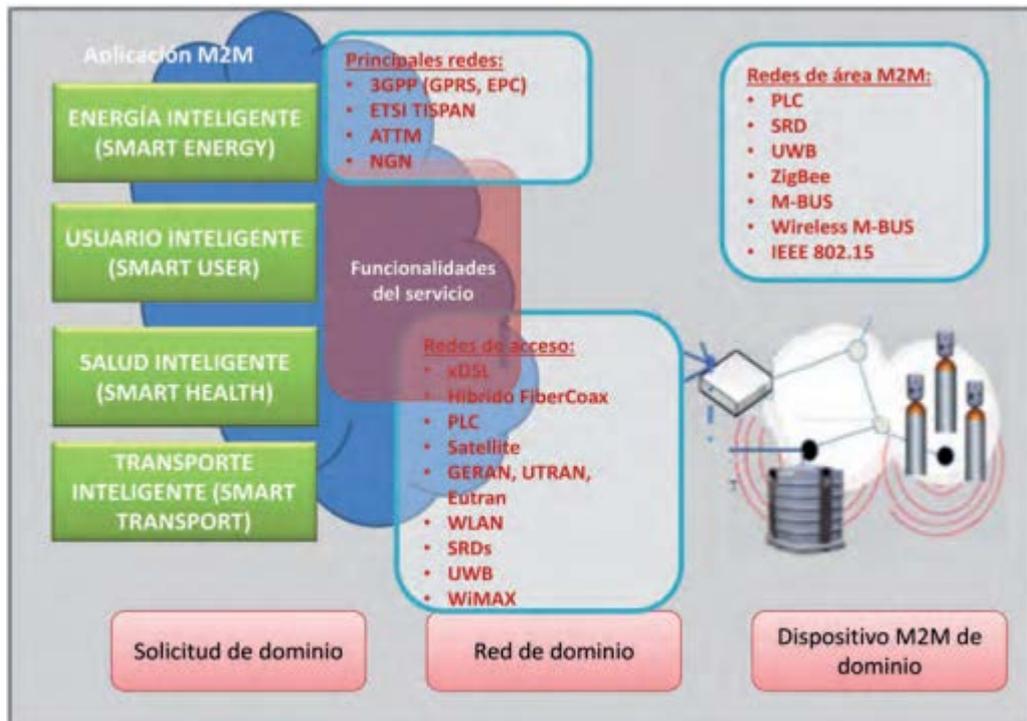
Otras organizaciones de estandarización, como el 3GPP64, también están trabajando en el área de M2M. El 3GPP, dedicado a los estándares para comunicaciones móviles de banda ancha, ha publicado los siguientes documentos aplicables a las tecnologías M2M:

- TR 22.86865: Study on Facilitating Machine to Machine Communication in 3GPP Systems.
- TR 33.81266: Study on Security Aspects of Remote Provisioning and Change of Subscription for M2M equipment.
- TS 22.36867: Service Requirements for Machine-Type Communications (MTC)

El IETF68, por su parte, también está trabajando en estándares para las tecnologías M2M:

- 6LoWPAN69 (IPv6 over Low power WPAN).
- ROLL70 (Routing Over Low power and Lossy networks).
- CORE71 (Constrained RESTful Environments).

Además de los estándares específicos para M2M, los estándares de comunicaciones son clave para facilitar la conectividad entre todos los dispositivos y actuadores de la Smart City. Como puede verse en la siguiente figura, la arquitectura M2M se apoya en los protocolos de comunicaciones, tanto en el dominio M2M como en el dominio de red:



Fuente: ETSI.

Ilustración 2-9 Protocolos de comunicaciones Smart City

Los estándares de comunicaciones permiten la comunicación de los dispositivos M2M entre sí y con el dominio de aplicaciones, en el que se sitúan las plataformas de gestión de los servicios. Entre los principales estándares de comunicaciones susceptibles de ser utilizados por los dispositivos M2M, cabe citar los siguientes:

- IEEE 802.15.1 (Bluetooth/WiBree)
- IEEE 802.15.4 (ZigBee y 6LowPan)
- IEEE 802.11 (WiFi)74
- GSM (Global System for Mobile Communications)
- GPRS (General Packet Radio System)
- EDGE (Enhanced Data Rates for GSM Evolution).
- UMTS (Universal Mobile Telecommunication System, 3GPP-Release 4)
- WiMAX, IEEE 802.1679
- HSPA (High Speed Packet Access, 3GPP Release 5 y 6) y HSPA+ (EDGE 3GPP Release 7)
- LTE (Long Term Evolution 3GPP Release 8 y 9)
- LTE-A (LTE Advanced, 3GPP Release 10)
- WiMAX II, IEEE 802.16j/m8

Algunos ejemplos de Smart City, que muchos aún están en fase de proyectos pilotos, son los siguientes:

- **Masdar** – Eco ciudad "inteligente" en construcción en el desierto de Abou Dabi.
- **Lyon** – El gran Lyon, en Francia, refuerza su política de desarrollo económico destinado a empresas de todos los tamaños, elaborando una "smart city" estratégica.
- **Campus de la USTL (Université de Lille/Villeneuve d'Ascq)**, en 2013, en asociación con una "Red Urbana Inteligente" (en francés: "Réseaux Urbains Intelligent").
- **Proyectos "Smart City" en Angoulême, Besançon, Vigo, y Carros proyecto IssyGrid (en Issy-les-Moulineaux)** – Lo que principalmente se destaca en estos proyectos es un reforzamiento de la comunicación, de la coordinación, y de la integración.
- **Amsterdam Smart City** – Entre otras cosas, una propuesta de la Wageningen_University_and_Research_Centre Wageningen University and Research Centre (WUR) y el MIT, ha sido retenida para el proyecto 'Amsterdam Metropolitan Solutions'.
- **Cairo Smart Village** – Proyecto de ciudad inteligente en Egipto.
- **Dubái Smart City y Dubái Internet City** – Parque tecnológico creado por el Gobierno de Dubái como una zona franca y una base estratégica para compañías que apunten a mercados emergentes locales.
- **Ciudad de Edimburgo en el plan de valoración y acción para el e-gobierno – Implementación y modernización en conjunción con BT Global Services.62 63**
- **Kochi Smart City** – Parque tecnológico, zona económica especial en construcción en Kochi, Kerala (India).
- **Malta Smart City** – SmartCity de negocios.
- **SmartSantander, hoy día probablemente la mayor red de sensores del mundo** – Por el momento, 1100 captosres sin hilos de la sociedad Libelium65 han sido instalados, 400 de ellos para medir los lugares de aparcamiento, y 700 para controlar parámetros ambientales tales como el ruido, el monóxido de carbono, la temperatura, o la luz solar.
- **Yokohama Smart City Project (YSCP)** – En una de las ciudades más grandes en Japón, Yokohama, se proyecta construir la siguiente infraestructura de generación energética, para entre otras cosas maximizar la reducción de emisiones de CO2, y estar en la vanguardia en cuanto al sistema social y en cuanto a la protección de la naturaleza.

Las ventajas de una Smart City son las siguientes:

- Debe de existir y coexistir una simbiosis en cuatro puntos de referencia para proporcionar una base sólida; Aspecto Humano, Gobierno, Medio Ambiente y Economía que implicará un compromiso de los diferentes agentes involucrados en un proceso de mejora constante con el fin de mejorar la calidad de vida, tanto del entorno como de sus habitantes.
- Está íntimamente relacionado con la evolución hacia la llamada Internet del Futuro, particularmente en lo relacionado con la Internet de las Cosas (Internet of Things)
- Incremento de un nuevo modelo y oportunidad de negocio. Nuevos servicios que respondan mejor a las necesidades específicas de los ciudadanos.
- Gestión automática y eficiente de las infraestructuras urbanas. Mejora ahorro energético, mejoras de eficiencia energética ...etc. Mejora en la gestión de la movilidad y el aparcamiento urbano, para mejorar el tráfico y disminuir los tiempos de búsqueda de lugar de aparcamiento, disminución de las colas y los tiempos de espera en las oficinas municipales y los centros de salud, etc.
- Mejora del urbanismo y entorno. Más y mejora de las zonas verdes, zonas periféricas...etc.
- Reducción de los gastos que puede producir un inmueble, electricidad, comunidad...etc. Reducción y optimización de tiempos de cara al consumidor.

Pero claro, las Smart City también tienen sus desventajas, ya que todo giro de rumbo de un modelo de ciudad tradicional a una ciudad inteligente, y estas son algunas de ellas:

- La financiación por parte de la Administración, dado que se requiere una inversión importante en tecnología.
- Dada la implantación de un alto grado de tecnología se depende de compañías que ofrezcan estos servicios. Tanto a nivel público como particular.
- Reducción de la intimidad. "Para ser más eficientes, se debe de observar que hábitos dispone el consumidor en todos sus aspectos y niveles"
- Los inmuebles encarecen. Son más complejos de ejecutar y construir.
- Mayores brechas tecnológicas entre ciudades y realidades. No todas las ciudades pueden asumir tal coste.
- Por la complejidad que absorben las ciudades Smart Cities, producen al mismo tiempo un aumento considerable de residuos.

Smart Building

Los Smart Buildings o Edificios Inteligentes son aquellos cuyas instalaciones y sistemas (de climatización, iluminación, electricidad, seguridad, telecomunicaciones, multimedia, informáticas, control de acceso, etc.) permiten una gestión y control integrada y automatizada, con el fin de aumentar la eficiencia energética, la seguridad, la usabilidad y la accesibilidad.

El concepto de Smart Buildings es aplicable para todas las tipologías de edificios, tanto para su rehabilitación como para la nueva construcción.

En el nivel más fundamental, los edificios inteligentes ofrecen servicios de construcción útiles que hacen ocupantes productiva (por ejemplo, iluminación, confort térmico, calidad del aire, la seguridad física, saneamiento, y muchos más), al menor costo y el impacto ambiental sobre el ciclo de vida del edificio. Alcanzar esta visión requiere la adición de la inteligencia desde el comienzo de la fase de diseño hasta el final de la vida útil del edificio. Edificios inteligentes utilizan tecnología de la información durante la operación para conectar una variedad de subsistemas, que típicamente operan de forma independiente, de modo que estos sistemas pueden compartir la información para optimizar el rendimiento total del edificio. Edificios inteligentes miran más allá de los equipos de construcción dentro de sus cuatro paredes. Están conectados y responden a la red eléctrica inteligente, y que interactúan con los operadores de edificios y ocupantes para darles poder con nuevos niveles de visibilidad e información procesable.

Un edificio inteligente debe reunir las siguientes características:

- Ser eficiente en el consumo: un Smart Building tiene sistemas de ahorro de energía y agua. Controlando y regulando el caudal, y teniendo la información sobre el consumo de energía.
- Integración en sus sistemas de control: El sistema de control tiene que estar totalmente integrado en el edificio y formar parte de él, centralmente automatizado para optimizar su operación y administración en forma electrónica.
- Ser seguros: altamente seguros, con los sistemas de seguridad más innovadores.
- Ser flexibles: edificio altamente adaptable para implantar los continuos cambios tecnológicos.
- Ser ergonómico: los Smart Buildings han de ser confortables para sus habitantes, ya que uno de sus objetivos es de hacer la vida más fácil a sus ocupantes.

Los objetivos de un Smart Building son los siguientes:

Objetivos arquitectónicos

- Satisfacer las necesidades presentes y futuras de los ocupantes, propietarios y operadores del edificio.
- La flexibilidad, tanto en la estructura como en los sistemas y servicios.
- La funcionalidad del edificio.
- Mayor confort para el usuario.

- El incremento de la seguridad.

Objetivos tecnológicos

- La disponibilidad de medios técnicos avanzados de telecomunicaciones.
- La automatización de las instalaciones.
- La integración de servicios.

Objetivos ambientales

- La creación de un edificio saludable.
- El ahorro energético.
- Integración del edificio con el medio donde está.

Objetivos económicos

- La reducción de los altos costos de operación y mantenimiento.
- Beneficios económicos para el cliente.
- Incremento de la vida útil del edificio.
- La relación costo-beneficio.

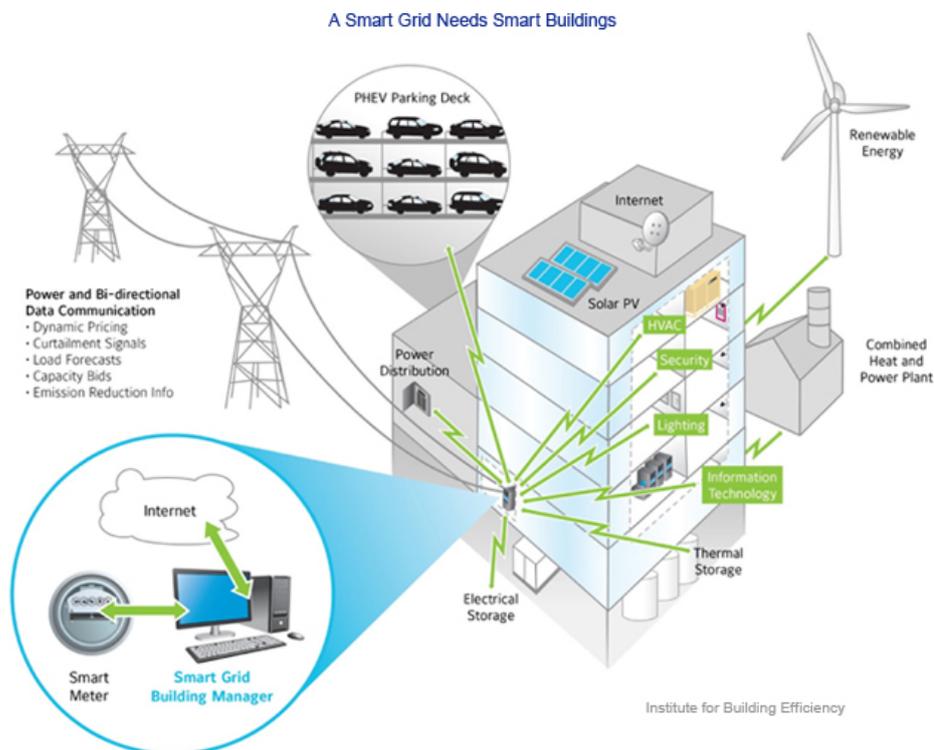


Ilustración 2-10 Smart Building

El control y gestión de las instalaciones y sistemas del edificio se llevan a cabo a través de la inmótica o automatización de edificios.

La inmótica ofrece la posibilidad de monitorización del funcionamiento general del edificio. Del mismo modo, permite un mayor control de accesos y el seguimiento continuo de quien haya ingresado al edificio, con el objetivo de reducir el consumo de energía, aumentar el confort y la seguridad de los mismos.

Entenderemos que un edificio es "inteligente" si incorpora sistemas de información en todo el edificio, ofreciendo servicios avanzados de la actividad y de las telecomunicaciones. Con control automatizado, monitorización, gestión y mantenimiento de los distintos subsistemas o servicios del edificio, de forma óptima e integrada, local y remotamente. Diseñados con suficiente flexibilidad como para que sea sencilla y económicamente rentable la implantación de futuros sistemas.

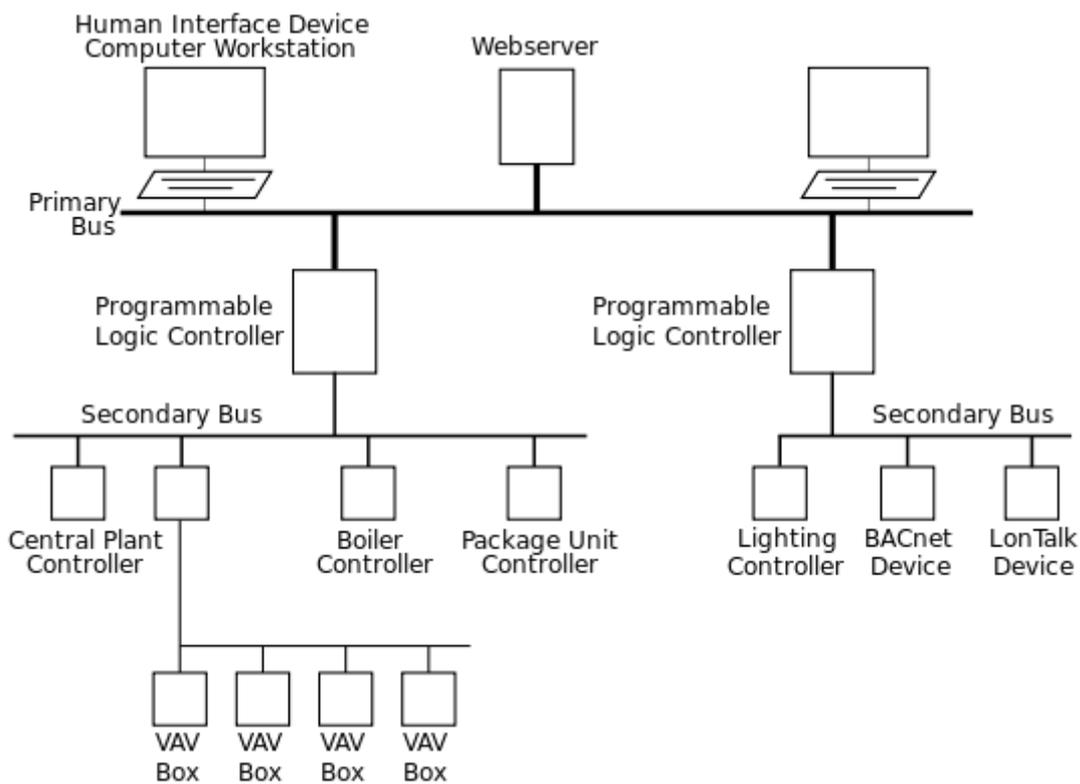


Ilustración 2-11 Arquitectura Smart Building

Bajo este nuevo concepto se define la automatización integral de inmuebles con alta tecnología. La centralización de los datos del edificio o complejo, posibilita supervisar y controlar confortablemente desde una PC, los estados de funcionamiento o alarmas de los sistemas que componen la instalación, así como los principales parámetros de medida. La Inmótica integra la domótica interna dentro de una estructura en red.

La centralización de los datos del edificio o complejo posibilita supervisar y controlar confortablemente los estados de funcionamiento o alarmas de los sistemas que componen la instalación, así como los principales parámetros de medida.

La Inmótica integra la domótica interna dentro de una estructura en red.

Se entiende por domótica interna al conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación, cableadas o inalámbricas.

En un edificio inteligente también cabe pensar en los materiales con los cuales esta construidos. Éstos deben de ser materiales que respeten el medio ambiente y reciclable, como por ejemplo el aluminio, que es un material reciclable.

Existen los llamados edificios verdes, los cuales no solo incluyen los sistemas automatizados integrados, sino que también están diseñados para reducir el impacto negativo en la salud humana y en el entorno donde están construidos.

La construcción verde crea estructuras y utiliza procesos que son medioambientalmente responsables y eficientes en el uso de recursos a través del ciclo de vida de las edificaciones. El ahorro en los costes de mantenimiento y gestión del inmueble debe justificar el coste de la inversión inicial.

Las posibles características que hacen que un edificio sea inteligente y además respetuoso con el medio ambiente son la instalación de:

- Sistemas de recogida de aguas pluviales para uso sanitario.
- Programas de recuperación de residuos y depuración de vertidos.
- Sistemas para el ahorro de recursos.
- Empleo de materiales saludables para el medioambiente.
- Instalación de jardines tanto en las fachadas como en los terrados.



Ilustración 2-12 Smart Building Ecológico

Desarrollo móvil

Un teléfono inteligente (Smartphone en inglés) es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de almacenar datos y realizar actividades semejantes a una mini computadora y conectividad que un teléfono móvil convencional. El término «inteligente» hace referencia a la capacidad de usarse como un ordenador de bolsillo, llegando incluso a remplazar a un ordenador personal en algunos casos.

Generalmente los teléfonos con pantallas táctiles son los llamados "teléfonos inteligentes", pero el completo soporte al correo electrónico parece ser una característica indispensable encontrada en todos los modelos existentes y anunciados desde 2007. Casi todos los teléfonos inteligentes también permiten al usuario instalar programas adicionales, normalmente inclusive desde terceros —hecho que dota a estos teléfonos de muchísimas aplicaciones en diferentes terrenos—, pero algunos vendedores gustan de tildar a sus teléfonos como inteligentes aun cuando no tienen esa característica.

Entre otras características comunes está la función multitarea, el acceso a Internet vía WiFi o red 3G, función multimedia (cámara y reproductor de videos/mp3), a los programas de agenda, administración de contactos, acelerómetros, GPS y algunos programas de navegación así como ocasionalmente la habilidad de leer documentos de negocios en variedad de formatos.

Estos dispositivos, debido al reinado casi por totalidad en el mundo de la telefonía móvil de los Smartphone, resulta una gran herramienta para el control de una vivienda inteligente ya que permite interactuar con la vivienda de muy diversas formas, ya sea mediante el control por voz o mediante el uso de los sensores para controlar y monitorizar la casa. Además con el uso del 3G o el 4G (Tecnología de comunicación que está tomando importancia), permite al usuario ver en directo el interior de la casa o los exteriores (si la vivienda dispone de cámaras IP). Y la interacción con los distintos dispositivos de la casa mediante gestos naturales conocidos para todos los usuarios.

Entornos de desarrollo (Frameworks)

Existen multitud de entornos de desarrollo para dispositivos móviles, algunos son exclusivos para ciertos dispositivos y otros no. Los que son exclusivos para ciertos dispositivos normalmente se utilizan para desarrollar lo que se llaman aplicaciones nativas y los que no son exclusivos se utilizan para crear aplicaciones web.

Vamos a exponer los diferentes tipos de entorno de desarrollo para los principales sistemas operativos que predominan actualmente en el ecosistema del desarrollo móvil, que actualmente son Android e iOS y que son utilizados en la actualidad para el desarrollo de aplicaciones enfocadas al control de una vivienda inteligente.

Eclipse

El entorno de desarrollo Eclipse es un conjunto de herramientas de programación de código abierta multiplataforma.

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

La base para Eclipse es la Plataforma de cliente enriquecido (RCP). Los siguientes componentes constituyen la plataforma de cliente enriquecido:

- Plataforma principal - inicio de Eclipse, ejecución de plugins
- OSGi - una plataforma para bundling estándar.
- El Standard Widget Toolkit (SWT) - Un widget toolkit portable.
- JFace - manejo de archivos, manejo de texto, editores de texto
- El Workbench de Eclipse - vistas, editores, perspectivas, asistentes

Para este proyecto hemos utilizado este entorno de desarrollo para su realización, ya que es un entorno el cual me parece más sencillo de utilizar que NetBeans y es sobre el que se basa todo el desarrollo.

NetBeans

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE2 es un producto libre y gratuito sin restricciones de uso.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

El NetBeans IDE es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

NetBeans IDE 6.5, la cual fue publicada el 19 de noviembre de 2008, extiende las características existentes del Java EE (incluyendo Soporte a Persistencia, EJB 3 y JAX-WS). Adicionalmente, el NetBeans Enterprise Pack soporta el desarrollo de Aplicaciones empresariales con Java EE 5, incluyendo herramientas de desarrollo visuales de SOA, herramientas de esquemas XML, orientación a web servicios (for BPEL), y modelado UML. El NetBeans C/C++ Pack soporta proyectos de C/C++, mientras el PHP Pack, soporta PHP 5.

Modularidad. Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente.

Desde julio de 2006, NetBeans IDE es licenciado bajo la Common Development and Distribution License (CDDL), una licencia basada en la Mozilla Public License (MPL). En octubre de 2007, Sun anunció que NetBeans desde entonces se ofrecerá bajo licenciamiento dual de Licencia CDDL y la GPL versión 2.

PhoneGap

PhoneGap es un framework para el desarrollo de aplicaciones móviles producido por Nitobi, y comprado posteriormente por Adobe Systems.^{3 4} Principalmente, PhoneGap permite a los programadores desarrollar aplicaciones para dispositivos móviles utilizando herramientas genéricas tales como JavaScript, HTML5 y CSS3. Las aplicaciones resultantes son híbridas, es decir que no son realmente aplicaciones nativas al dispositivo (ya que el renderizado se realiza mediante vistas web y no con interfaces gráficas específicas de cada sistema), pero no se tratan tampoco de aplicaciones web (teniendo en cuenta que son aplicaciones que son empaquetadas para poder ser desplegadas en el dispositivo incluso trabajando con el API del sistema nativo).

En la tercera versión de PhoneGap se incorpora el uso de una interfaz de comandos a través de consola, una nueva arquitectura de complementos descentralizados y la posibilidad de utilizar un código web unificado para crear múltiples proyectos.

PhoneGap maneja APIs que permiten tener acceso a elementos como el acelerómetro, la cámara, los contactos en el dispositivo, la red, el almacenamiento, las notificaciones, etc. Estas API se conectan al sistema operativo usando el código nativo del sistema huésped a través de una Interfaz de funciones foráneas en Javascript.

PhoneGap permite el desarrollo ya sea ejecutando las aplicaciones en nuestro navegador web, sin tener que utilizar un simulador dedicado a esta tarea, y brinda la posibilidad de soportar funciones sobre frameworks como Sencha Touch o JQuery Mobile.

PhoneGap es una distribución de Apache Cordova.⁵ La aplicación se llamó en un principio "PhoneGap", y posteriormente "Apache Callback". Ambos sistemas tienen funciones casi idénticas, la diferencia principal entre Apache Cordova y Phonegap es que el segundo tiene acceso a servicios de compilación en la nube proporcionados por Adobe Creative Cloud.

Apache Cordova es un software de código abierto y tanto este como PhoneGap pueden ser utilizados libremente en cualquier aplicación sin necesidad de atribución o licencias de ningún tipo.

Este framework permite a los desarrolladores web enfocarse en el desarrollo para los teléfonos inteligentes teniendo como base un código genérico con herramientas tales como JavaScript, HTML, CSS, y creando una interfaz de funciones foráneas para embeber una vista Web en el dispositivo móvil.

Xcode

Xcode es el entorno de desarrollo integrado (IDE, en sus siglas en inglés) de Apple Inc. y se suministra gratuitamente junto con Mac OS X. Xcode trabaja conjuntamente con Interface Builder, una herencia de NeXT, una herramienta gráfica para la creación de interfaces de usuario.

Xcode incluye la colección de compiladores del proyecto GNU (GCC), y puede compilar código C, C++, Objective-C, Objective-C++, Java y AppleScript mediante una amplia gama de modelos de programación, incluyendo, pero no limitado a Cocoa, Carbón y Java. Otras compañías han añadido soporte para GNU Pascal,¹ Free Pascal,² Ada y Perl.³

Entre las características más apreciadas de Xcode está la tecnología para distribuir el proceso de construcción a partir de código fuente entre varios ordenadores, utilizando Bonjour.

APIS Domótica

A continuación, enumeraremos algunos Apis relacionadas con la domótica en general y Apis centradas en dispositivos concretos dentro de una Smart Home. En los últimos años, el interés por las Smart Home ha recibido un gran auge entre el sector empresarial el cual ve un gran potencial gracias al uso de dispositivos inteligentes como Smartphones, tabletas entre otros dispositivos.

Nest

Nest Labs es una compañía fundada por Tony Fadell y Matt Rogers, el primero ex-ingeniero de Apple al que se le conoce como uno de los padres del iPod, no en vano fue vicepresidente senior de la división iPod; el segundo, otro ex-ingeniero jefe de los departamentos iPod y iPhone en Apple. Y desde el momento en que se funda la compañía hasta ahora han sido sólo dos productos los lanzados, siendo el termostato Nest el principal. Aunque Nest Protect, detector de humo, también tiene su interés.

Nest empezó su andadura en el 2011 vendiendo un termostato inteligente; este dispositivo se encarga de regular la temperatura de las habitaciones dependiendo de si estamos en casa o no. Para ello aprende de nuestras costumbres, de cuándo hemos salido al trabajo y de cuándo vamos a llegar, además de ser programable vía remota por Wifi. El hecho de que aprenda por si solo le ayuda a ahorrar en la factura de la luz, ya que así la temperatura de nuestra casa siempre será la correcta sin derrochar electricidad. Desde entonces, se ha convertido en una de las compañías más importantes en el sector de las casas inteligentes, expandiéndose con otros productos como un detector de humo inteligente.



Ilustración 2-13 Termostato Nest

El termostato ofrecía una solución inteligente a un “problema” que según Tony Fadell era inaceptable. Ese problema no era ni más ni menos que el control responsable y eficiente del consumo energético de los sistemas de refrigeración y calefacción de los hogares. Un consumo que, según ellos, supone el 10% de toda la energía que se consume en Estados Unidos.

Opinaban que no era lógico como aquellos primeros termostatos de 1970, con su funcionamiento complejo para el usuario, no hubiesen evolucionado ofreciendo diseños más sencillos y un funcionamiento mejorado gracias a los avances tecnológicos. Por tanto se lanzaron al desarrollo de un producto que permitiese a ese 89% de propietarios, que rara vez hicieron uso del suyo, ahorrasen energía y por implicación directa unos dólares en la factura.

El termostato Nest es mucho más que un simple termostato con forma circular que nos trae recuerdos de diferentes dispositivos, es un producto que mediante una combinación de sensores, algoritmos de aprendizaje y la computación en la nube es capaz de aprender de nuestro comportamiento y preferencias. De modo que, poco a poco dispone de más información que le permiten funcionar de forma autónoma, sin que apenas debamos interactuar con él para que la temperatura de nuestro hogar sea óptima.

Este año 2014, la empresa Nest ha liberado su API (<http://api-portal.anypoint.mulesoft.com/nest/api/nest-thermostat-api>) para que podamos realizar desarrollos utilizando su producto y potenciar así la aparición de aplicaciones de la comunidad Android que potencie su desarrollo y mejoras ya que con este movimiento se obtendría mucho feedback por parte de la comunidad que ayudara a mejorar el producto.

Esta compañía en este año 2014, ha sido comprada por Google en un claro ejemplo en el que Google quiere entrar en la campo de las Smart Home. En la llamada era del “Internet de las cosas”, es importante estar conectado con absolutamente todo lo que nos rodea; eso supone controlar una gran cantidad de información. La compra de Nest por Google supone un nuevo apoyo para algo que ya conocimos en el 2011 durante una de las Google I/O, Android at Home, del cual hablaremos más adelante en este documento. Otra de las novedades relacionadas con la compra de Nest por Google es la integración de las funcionalidades de Nest en Google Now.

Homekit

Es la apuesta por parte de Apple de llegar al mercado de los hogares inteligentes a través de una API específica para que los fabricantes de dispositivos puedan integrarlo: desde termostatos, hasta cerraduras pasando por control mediante Siri, todos los dispositivos podrán integrar este nuevo sistema. Con el uso de este, se pondrán a realizar diferentes acciones de forma sencilla. Por ejemplo, si se desea aumentar la temperatura de la casa con decir “subir la calefacción” al iPhone esto se realizará de forma automática.

Uno de los factores esenciales es el de la seguridad, algo lógico por otra parte ya que opciones como abrir las puertas de casa o el garaje serán de las accesibles con HomeKit. Aquí Apple ha asegurado que la red que utilizará esta nueva plataforma será totalmente segura en lo referente a la conexión (opciones como la encriptación a buen seguro que están contempladas). Por lo tanto, parece que la compañía está teniendo en cuenta todas las opciones.

Otro de los pilares en los que se debe sustentar es que el API que utilice HomeKit sea accesible para las compañías que trabajan en este sector. Aquí Apple ha asegurado que está trabajando con empresas “líderes” del sector (como por ejemplo Philips o Chamberlain). El caso es que habrá que conocer con más detalle esto, ya que otros intentos similares no fructificaron por no contemplar debidamente esto. Un ejemplo es Google Home, que fue anunciado hace más de tres años y poco o nada se ha sabido de él.



Ilustración 2-14 Homekit Apple

Android@Home

Durante la conferencia Google I/O del 2011, Google anuncio el proyecto Android Home, donde hablaba de un sistema de domótica donde todas las piezas de este sistema giraban alrededor de un dispositivo Android. La idea era que desde un teléfono o tableta se pudiese controlar aspectos tales como abrir la puerta de tu casa, encender o apagar la luz, controlar la temperatura o abrir tu coche.

El proyecto por aquel entonces se trataba de un concepto demasiado ambicioso y con el paso del tiempo se quedó en el olvido.

Interacción espacial

La interacción mediante el uso de estas formas es muy diversa, la más conocida es la de poder controlar la intensidad de la luz de una habitación simplemente deslizando el dedo de arriba hacia abajo para disminuir la intensidad o al revés para aumentarla. También usando los sensores podemos adecuar esta intensidad a las condiciones lumínicas ideales automáticamente. O mediante un ligero movimiento poder encender o apagar la luz de dichas habitaciones. Otra forma de interacción es la de poder cambiar entre dispositivo simplemente apuntando hacia donde se encuentra dicho dispositivo mediante el uso de la brújula del dispositivo o el de con un simple gesto hacia el borde de la pantalla el cambiar de dispositivo dentro de dicha habitación.

La interacción mediante el uso de gestos espaciales, es una nueva forma de interacción que está tomando fuerza en los últimos años. El ejemplo más conocido es el de Kinect, que mediante el uso de una cámara detecta al usuario y sus gestos realizados y responde realizando las acciones que el usuario realiza. En este ejemplo lo que podría asociarse en común al trabajo realizado, sería en cuanto a centro multimedia, el usuario cómodamente desde su sofá, puede gestionar sus contenidos multimedia con gestos espaciales. Otros ejemplos serían también, el Move de PlayStation o la Wii de Nintendo, que se valen de acelerómetros y giroscopios para sensar los movimientos del usuario y realizar acciones.



Ilustración 2-15 Kinect

Otro ejemplo más acorde con el planteamiento de este trabajo sería la aplicación “Magus Free 3D”. Esta aplicación permite asociar diversos gestos, ya sean simples o complejos (Encadenando varios gestos) a diferentes acciones del dispositivo como por ejemplo silenciar el dispositivo, Encender o apagar el wifi, abrir la galería de imágenes. La aplicación funciona en segundo plano y permite al usuario realizar los gestos incluso con el teléfono bloqueado.

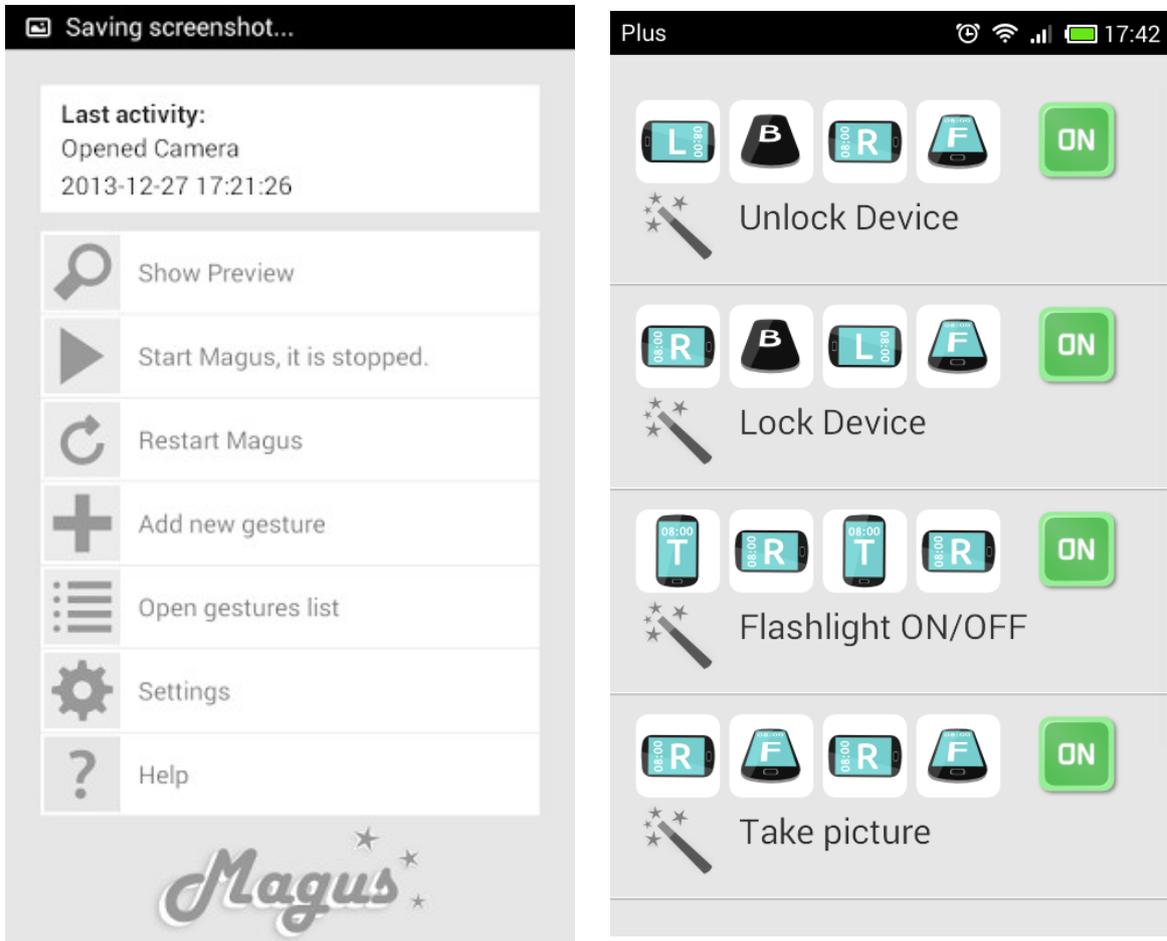


Ilustración 2-16 Magus Free 3D

Tecnologías utilizadas

Eclipse

El entorno de desarrollo Eclipse es un conjunto de herramientas de programación de código abierta multiplataforma.

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

La base para Eclipse es la Plataforma de cliente enriquecido (RCP). Los siguientes componentes constituyen la plataforma de cliente enriquecido:

- Plataforma principal - inicio de Eclipse, ejecución de plugins
- OSGi - una plataforma para bundling estándar.
- El Standard Widget Toolkit (SWT) - Un widget toolkit portable.
- JFace - manejo de archivos, manejo de texto, editores de texto
- El Workbench de Eclipse - vistas, editores, perspectivas, asistentes

Para este proyecto hemos utilizado este entorno de desarrollo para su realización, ya que es un entorno el cual me parece más sencillo de utilizar que NetBeans y es sobre el que se basa todo el desarrollo.

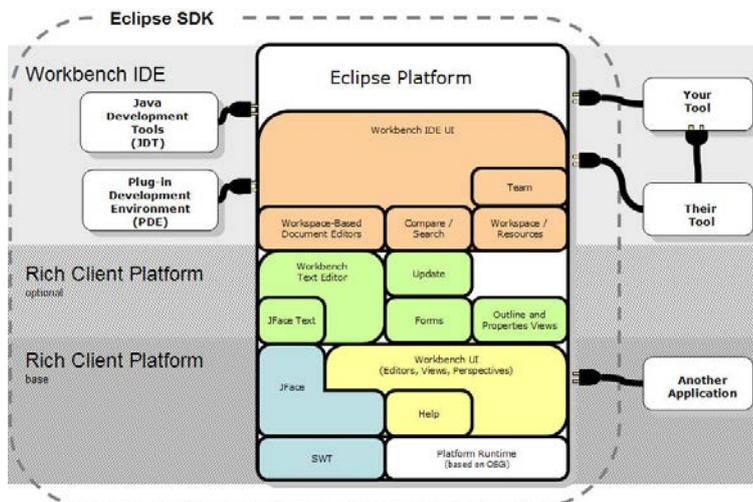


Ilustración 2-17 Eclipse SDK

Eclipse utiliza un número amplio de lenguajes de programación, pero el más utilizado es Java.

La definición que da el proyecto Eclipse acerca de su software es: "una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular".

En cuanto a las aplicaciones clientes, Eclipse provee al programador con frameworks muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones web, etc. Por ejemplo, GEF (Graphic Editing Framework - Framework para la edición gráfica) es un plugin de Eclipse para el desarrollo de editores visuales que pueden ir desde procesadores de texto WYSIWYG hasta editores de diagramas UML, interfaces gráficas para el usuario (GUI), etc. Dado que los editores realizados con GEF "viven" dentro de Eclipse, además de poder ser usados conjuntamente con otros plugins, hacen uso de su interfaz gráfica personalizable y profesional.

El SDK de Eclipse incluye las herramientas de desarrollo de Java, ofreciendo un IDE con un compilador de Java interno y un modelo completo de los archivos fuente de Java. Esto permite técnicas avanzadas de refactorización y análisis de código. Mediante diversos plugins estas herramientas están también disponibles para otros lenguajes como C/C++ (Eclipse CDT) y en la medida de lo posible para lenguajes de script no tipados como PHP o Javascript. El IDE también hace uso de un espacio de trabajo, en este caso un grupo de metadato en un espacio para archivos plano, permitiendo modificaciones externas a los archivos en tanto se refresque el espacio de trabajo correspondiente.

Un punto muy importante a notar son los diversos lenguajes de programación utilizados en el desarrollo del proyecto. De acuerdo al análisis realizado usando SLOCCount, el lenguaje más utilizado es Java.

Lenguaje	Líneas de código	%
Java	1.911.693	92,66%
ANSI C	133.263	6,46%
C++	10.082	0,49%
JSP	3.613	0,18%
sh	2.066	0,10%
Perl	1.468	0,07%
Php	896	0,04%

Java

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en el 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva mucho de C y C++, pero tiene menos facilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente. Java es un lenguaje de programación de general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir del 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de servidor de web, con unos 10 millones de usuarios reportados.

Uno de los principales problemas de Java son las fugas de memoria, que se evita mediante el uso del garbage collector (recolector de basura). Cuando no quedan referencias a un objeto, el recolector de basura de Java borra el objeto, liberando así la memoria que ocupaba previniendo posibles fugas. En definitiva, el recolector de basura de Java permite una fácil creación y eliminación de objetos y mayor seguridad.

La compañía Sun desarrolló la implementación de referencia original para los compiladores de Java, máquinas virtuales, y librerías de clases en 1991 y las publicó por primera vez en el 1995. A partir de mayo del 2007, en cumplimiento con las especificaciones del Proceso de la Comunidad Java, Sun volvió a licenciar la mayoría de sus tecnologías de Java bajo la Licencia. Otros también han desarrollado implementaciones alternas a estas tecnologías de Sun, tales como el Compilador de Java de GNU y el GNU Classpath.

El lenguaje Java se creó con cinco objetivos principales:

1. Debería usar el paradigma de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir por defecto soporte para trabajo en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

Para conseguir la ejecución de código remoto y el soporte de red, los programadores de Java a veces recurren a extensiones como CORBA (Common Object Request Broker Architecture), Internet Communications Engine u OSGi respectivamente.

La primera característica que se comenta como objetivo principal es la orientación a objetos. Una primera idea es diseñar el software de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se

combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el “comportamiento” (el código) y el “estado” (datos). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallido.

La segunda característica, la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Este es el significado de ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, “write once, run anywhere”.

Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como “bytecode” (específicamente Java bytecode) —instrucciones máquina simplificadas específicas de la plataforma Java. Esta pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino. El bytecode es ejecutado entonces en la máquina virtual (JVM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código.

Uno de los principales problemas de Java son las fugas de memoria, que se evita mediante el uso del garbage collector (recolector de basura). Cuando no quedan referencias a un objeto, el recolector de basura de Java borra el objeto, liberando así la memoria que ocupaba previniendo posibles fugas. En definitiva, el recolector de basura de Java permite una fácil creación y eliminación de objetos y mayor seguridad.

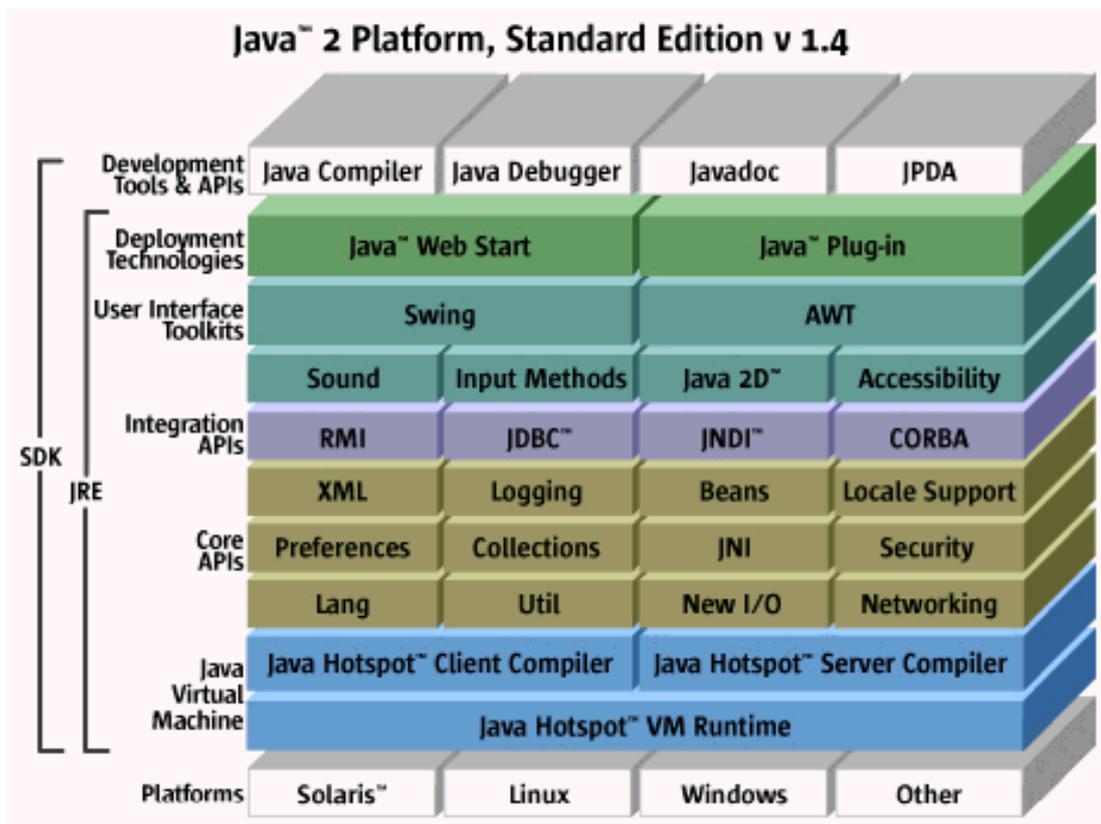


Ilustración 2-18 Java Platform, Standard Edition

Android

Android es un sistema operativo basado en Linux para dispositivos móviles, tales como teléfonos inteligentes o tabletas. Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio. El anuncio del sistema Android se realizó el 5 de noviembre de 2007 junto con la creación de la Open Handset Alliance, un consorcio de 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles. Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto. A la fecha, se han sobrepasado las 700.000 aplicaciones (de las cuales, dos tercios son gratuitas) disponibles para la tienda de aplicaciones oficial de Android: Google Play, sin tener en cuenta aplicaciones de otras tiendas no oficiales para Android como la tienda de aplicaciones Samsung Apps de Samsung. Google Play es la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software externamente. Los programas están escritos en el lenguaje de programación Java. No obstante, no es un sistema operativo libre de malware, aunque la mayoría de ello es descargado de sitios de terceros.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una Interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic. El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.

Android desde sus comienzos ha sufrido muchas actualizaciones de software que han ido puliendo el sistema operativo y mejorándolo frente a su gran rival iOS (Apple). Vamos a enumerar las distintas versiones de Android:

- Apple Pie (v 1.0)
- Banana Bread (v 1.1)
- Cupcake (v 1.5)
- Donut (v 1.6)
- Éclair (v 2.0/2.1)
- Froyo (v 2.2)
- Gingerbread (v.2.3)
- Honeycomb (v 3.0/3.1/3.2)
- Ice Cream Sandwich (v 4.0)
- Jelly Bean (v 4.1/4.2)
- KitKat (4.4.X)
- Key Lime Pie (v 5.0)

Según estudios de mercado en el 2013 Android domina el mercado de los teléfonos inteligentes, por delante iOS e Windows Phone que son sus rivales más directos en ese sector. En el primer trimestre de 2013 el total de dispositivos móviles (portátiles, móviles y tabletas) fueron de unos 308.7 millones de unidades vendidas. Desglosando un poco esos números, se puede comprobar que el 59.5% de ese total correspondió a los dispositivos con sistema operativo Android, frente al 19.3% correspondiente a Apple y 18.1% de Microsoft.

Posición	Plataforma	Unidades	Cuota 2014 Q1	Cuota 2013 Q4	Fabricantes principales
1	Android	1.209 M	69%	66%	Samsung, Huawei, Sony, Lenovo, ZTE, LG, Coolpac, HTC, Xiaomi, SonyEricsson
2	iOS	353 M	21%	21%	Apple
3	Symbian	58 M	4%	5%	Nokia, Sharp, Panasonic, Fujitsu
4	BlackBerry	52 M	3%	4%	BlackBerry
	Windows Phone	50 M	3%	3%	Nokia, Samsung, HTC, Huawei
	Otros	13 M	1%	1%	
	Total	1.748 M	smartphones en uso a 31 de marzo de 2014		

Ilustración 2-19 Ventas dispositivos Android Fuente: análisis de TomiAhonen Consulting

La arquitectura Android se compone de los distintos apartados

- **Aplicaciones:** Las aplicaciones. Todas las aplicaciones están escritas en lenguaje de programación Java.
- **Marco de trabajo de aplicaciones:** La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.
- **Bibliotecas:** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras
- **Runtime de Android:** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente.
- **Núcleo Linux:** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

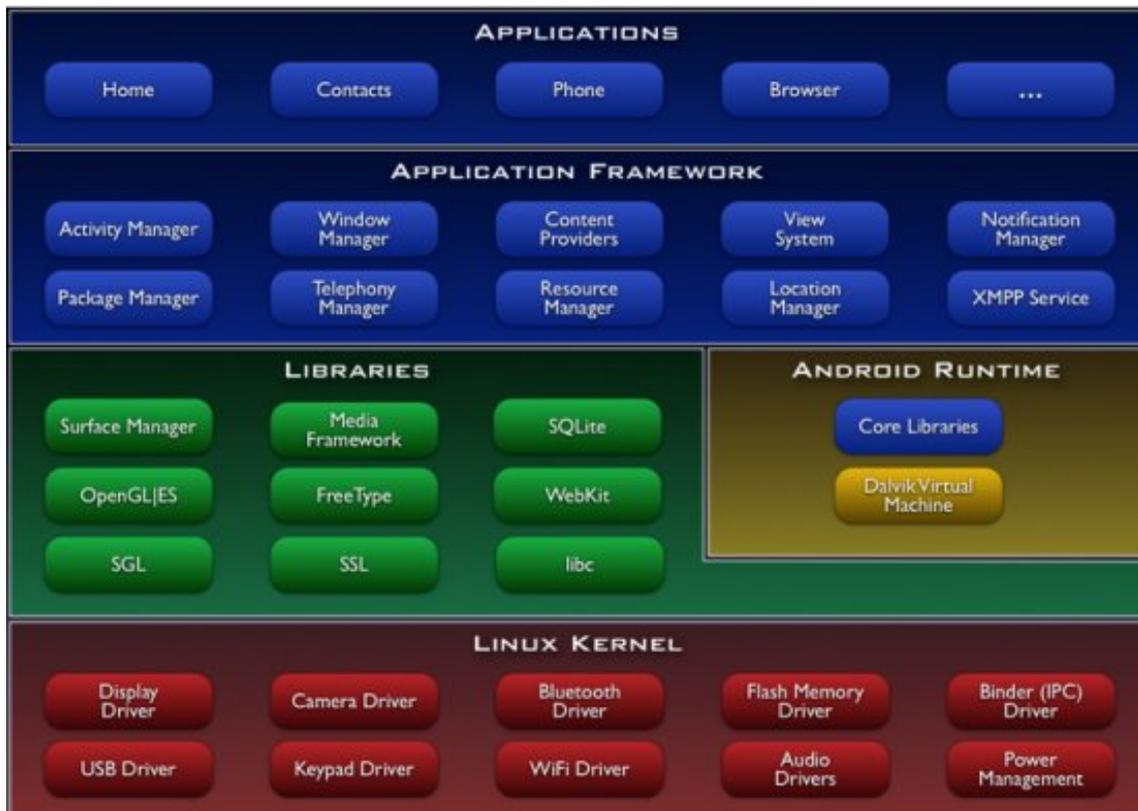


Ilustración 2-20 Arquitectura Android

Android, al contrario que otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se desarrolla de forma abierta y se puede acceder tanto al código fuente como a la lista de incidencias donde se pueden ver problemas aún no resueltos y reportar problemas nuevos.

El que se tenga acceso al código fuente no significa que se pueda tener siempre la última versión de Android en un determinado móvil, ya que el código para soportar el hardware (controladores) de cada fabricante normalmente no es público, así que faltaría un trozo básico del firmware para poder hacerlo funcionar en dicho terminal, y porque las nuevas versiones de Android suelen requerir más recursos, por lo que los modelos más antiguos quedan descartados por razones de memoria (RAM), velocidad de procesador, etc.

Para el desarrollo de este proyecto se eligió Android, ya que es un sistema abierto que permite gran personalización y que además permite tocar absolutamente todo el código del sistema operativo que por otra parte las otras opciones disponibles en el mercado no permite. Otra de las razones de su elección es por la gran cantidad de diversidad de dispositivos existentes en el mercado con lo que permite al usuario elegir un dispositivo que se adecue a sus posibilidades tanto económicas como otras diversas.

Aplicación nativa vs Web App

El mundo de las aplicaciones se debate siempre entre dos mundos, elegir aplicación nativa o aplicación web. Primero explicaremos que características contiene cada una de las opciones.

Web App: Una web app es una aplicación web optimizada mediante HTML5, CSS3 y JQUERY MOBILE para la correcta visualización en los Smartphone. Aunque no es una app propiamente dicha, pues no se instala en el dispositivo, consigue tener un look&feel bastante nativo. Uno de las principales ventajas de las web app con respecto a una app nativa es que por definición son originalmente multiplataforma gracias a ejecutarse en cualquier navegador que soporte HTML5. Con tan sólo unos retoques para cada plataforma, cualquier usuario disfrutará de tu webapp en su dispositivo. Por el contrario, en el caso de app nativas, habrá que hacer un desarrollo distinto para cada sistema operativo, lo que supondrá un incremento del coste de desarrollo de la aplicación. La principal ventaja de las webapps es su versatilidad. En principio, una web app se podrá visualizar en cualquier dispositivo móvil, ya sea Android, iOS, Blackberry o Windows Phone, basta con que dicho terminal tenga un navegador móvil. Si bien es cierto que para la correcta visualización en cada plataforma la webapp deberá incorporar algún que otro fragmento de código especial para cada una.

Principales ventajas:

- La ventaja de desarrollar aplicaciones web es que son multiplataforma. Pese a la posible necesidad de tener que hacer determinados ajustes, una aplicación en HTML5 funciona generalmente en todos los dispositivos móviles, siempre y cuando sean compatibles con el engine del Safari Webkit.
- Programar en lenguajes como HTML5 es mucho más sencillo y se necesitan menos requisitos que programar en algún lenguaje nativo. Por citar un ejemplo, para desarrollar aplicaciones para iPhone, tal y como ya comentamos en este post, hace falta un Mac. Una vez que tienes un Mac, si tienes la última versión de Mac Os, podrás descargarte el último Xcode, en cuyo caso contrario, tendrás que pagar. En consecuencia, está mucho más extendida.
- Al ser independientes de la plataforma, no se someten a ningún control, ni en el inicio ni en las posteriores actualizaciones, como sucede por ejemplo con las aplicaciones en el AppStore. Por tanto, no hay revisión de las versiones que se publiquen, y las actualizaciones llegan de manera automática a todos los usuarios y plataformas a la vez.
- Pese a que las distancias se van reduciendo, el rendimiento en las aplicaciones Web sigue siendo mucho menor, por lo que es son mucho más lentas en comparación con las apps nativas.
- La interfaz de usuario. En una aplicación web es difícil conseguir la sensación de la aplicación parezca integrada en el propio teléfono, algo que sí ocurre con las nativas.

App nativa: Una app nativa es una aplicación software desarrollada para smartphones diseñadas para explotar al máximo las características del dispositivo móvil. Cada día se

publican apps más y más potentes gracias a las últimas tecnologías de vanguardia como la geolocalización, realidad aumentada, tecnología NFC, etc.

Una de las principales ventajas de tener una aplicación nativa sobre otras posibilidades como las webapps son los market places. Cada plataforma cuenta con un servicio de recopilación y distribución de apps tanto de pago como gratuitas. Gracias a su programación en lenguaje nativo del terminal, todos los recursos de éste estarán accesibles para poder sacar el máximo partido a tu app.

Principales ventajas:

- El programar aplicaciones nativas podremos explotar al máximo cada una de las prestaciones integradas en los dispositivos, tales como acelerómetro, calendario, agenda de contactos, audio y vídeo, 3D, y así un largo etcétera.
- Las tiendas de aplicaciones son otra de las ventajas. El hecho de disponer de un lugar donde encontrar todas las aplicaciones disponibles supone una mayor difusión y propagación de la aplicación, permitiendo que mucha más gente acceda a ella.
- Notificaciones instantáneas. En aplicaciones que requieran actualizarse constantemente y en el momento exacto, las notificaciones push son completamente necesarias.
- Las compras dentro de la app se realizan a través de la cuenta de usuario del dispositivo, por lo que el proceso es muy sencillo para cualquier usuario y no requiere facilitar los datos de la tarjeta de crédito.
- Una aplicación nativa puede funcionar aunque no esté conectada a la red. De esta manera, el usuario podrá acceder a los contenidos de la aplicación en cualquier momento. Aunque en HTML5 ya hay mecanismos para el funcionamiento off-line, todavía está poco desarrollado y no se pueden garantizar buenos resultados.
- Las aplicaciones nativas proporcionan una mejor experiencia de usuario, puesto que la interfaz no tiene que cargarse junto con el resto de datos, lo que se traduce en interfaces más intuitivas y un mejor rendimiento en general.
- Necesitaremos programar una aplicación para cada plataforma, si queremos que esté disponible para todo el mercado.
- La implementación es bastante más costosa (véase programar para iOS).
- Poder lanzar una aplicación y que esté disponible en la tienda de aplicaciones supone cumplir con todos los requisitos impuestos por las respectivas compañías, destacando Apple y sus normas a la hora de lanzar una aplicación, lo cual limita al programador de hacer cuanto le plazca.

En este proyecto hemos decidido optar por una aplicación nativa ya que se puede aprovechar todo el potencial del dispositivo que utilizando una aplicación web no tendríamos a nuestra disposición. Otra de las razones por la que utilizar una aplicación nativa es la de no depender de una conexión permanente a Internet que si es bien cierto para nuestra aplicación es necesario una conexión permanente, para futuras ampliaciones de esta aplicación podría no necesitarse conexión permanente a Internet para algunas de esas nuevas características.

XML

XML, es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información. (Bases de datos Silberschatz).

XML no ha nacido sólo para su aplicación para Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Ventajas del XML:

- Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan *bugs* y se acelera el desarrollo de aplicaciones.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones. Podemos comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos, es decir, podríamos tener una aplicación en Linux con una base de datos Postgres y comunicarla con otra aplicación en Windows y Base de Datos MS-SQL Server.
- Transformamos datos en información, pues se le añade un significado concreto y los asociamos a un contexto, con lo cual tenemos flexibilidad para estructurar documentos.

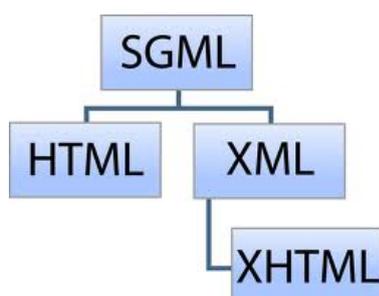


Ilustración 2-21 Arquitectura XML

REST

La Transferencia de Estado Representacional (Representational State Transfer) o REST es una técnica de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web. El término se originó en el año 2000, en una tesis doctoral sobre la web escrita por Roy Fielding, uno de los principales autores de la especificación del protocolo HTTP y ha pasado a ser ampliamente utilizado por la comunidad de desarrollo.

Si bien el término REST se refería originalmente a un conjunto de principios de arquitectura —descritos más abajo—, en la actualidad se usa en el sentido más amplio para describir cualquier interfaz web simple que utiliza XML y HTTP, sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes como el protocolo de servicios web SOAP. Es posible diseñar sistemas de servicios web de acuerdo con el estilo arquitectural REST de Fielding y también es posible diseñar interfaces XMLHTTP de acuerdo con el estilo de llamada a procedimiento remoto pero sin usar SOAP. Estos dos usos diferentes del término REST causan cierta confusión en las discusiones técnicas, aunque RPC no es un ejemplo de REST.

REST afirma que la web ha disfrutado de escalabilidad como resultado de una serie de diseños fundamentales clave:

- Un protocolo cliente/servidor sin estado: cada mensaje HTTP contiene toda la información necesaria para comprender la petición. Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes. Sin embargo, en la práctica, muchas aplicaciones basadas en HTTP utilizan cookies y otros mecanismos para mantener el estado de la sesión (algunas de estas prácticas, como la reescritura de URLs, no son permitidas por REST).
- Un conjunto de operaciones bien definidas que se aplican a todos los recursos de información: HTTP en sí define un conjunto pequeño de operaciones, las más importantes son POST, GET, PUT y DELETE. Con frecuencia estas operaciones se equiparan a las operaciones CRUD que se requieren para la persistencia de datos, aunque POST no encaja exactamente en este esquema.
- Una sintaxis universal para identificar los recursos. En un sistema REST, cada recurso es direccionable únicamente a través de su URI.
- El uso de hipermedios, tanto para la información de la aplicación como para las transiciones de estado de la aplicación.
- REST son típicamente HTML o XML. Como resultado de esto, es posible navegar de un recurso REST a muchos otros, simplemente siguiendo enlaces sin requerir el uso de registros u otra infraestructura adicional.

Un concepto importante en REST es la existencia de recursos (elementos de información), que pueden ser accedidos utilizando un identificador global (un Identificador Uniforme de Recurso). Para manipular estos recursos, los componentes de la red (clientes y servidores) se comunican a través de una interfaz estándar (HTTP) e intercambian representaciones de estos recursos (los ficheros que se descargan y se envían) - es cuestión de debate, no obstante, si la

distinción entre recursos y sus representaciones es demasiado platónica para su uso práctico en la red, aunque es popular en la comunidad RDF.

La petición puede ser transmitida por cualquier número de conectores (por ejemplo clientes, servidores, cachés, túneles, etc.) pero cada uno lo hace sin "ver más allá" de su propia petición (lo que se conoce stateless, otra restricción de REST, que es un principio común con muchas otras partes de la arquitectura de redes y de la información) Así, una aplicación puede interactuar con un recurso conociendo el identificador del recurso y la acción requerida, no necesitando conocer si existen cachés, proxys, cortafuegos, túneles o cualquier otra cosa entre ella y el servidor que guarda la información. La aplicación, sin embargo, debe comprender el formato de la información devuelta (la representación), que es por lo general un documento HTML o XML, aunque también puede ser una imagen o cualquier otro contenido.

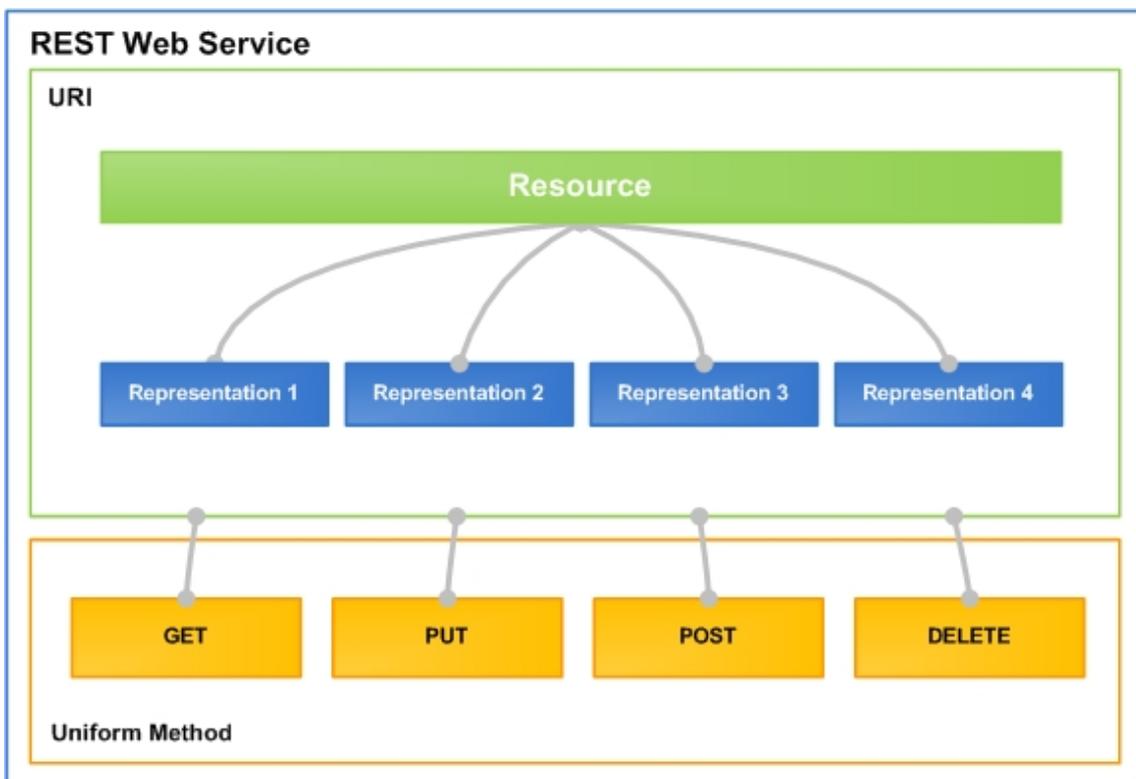


Ilustración 2-22 Arquitectura Servicios Web REST

3. PATRONES DE **INTERACCIÓN GESTUAL**

En este capítulo, vamos a definir lo que llamamos el marco abstracto de los gestos que utilizará nuestra librería de movimientos

Para esta definición de patrones de interacción gestual, lo primero es centrarse en la situación donde vamos a desarrollar estos patrones. Para este trabajo nos centramos en una vivienda, más concretamente en el control de los dispositivos que se encuentran en ella. Una vivienda con tiene muchos dispositivos los cuales son usados para distintas tareas y con los que se interactúa de maneras muy distintas para la vida cotidiana del usuario. Todos estos dispositivos estamos acostumbrados a utilizarlos y realizamos la interacción con ellos de forma natural y no nos paramos a pensar el porqué de esa forma de interactuar con ellos ya que están en nuestra vida cotidiana desde hace mucho tiempo. Por ejemplo en una vivienda podemos encontrarnos los siguientes dispositivos:

- **Puerta:** Lo habitual de las puertas es que se abren y se cierran mediante un pomo o una manivela, que es girado a un lado u otro si queremos cerrar o abrir. También contiene una cerradura que es la permite cerrar o abrir la puerta.
- **Persiana:** Las persianas, existen de distintos tipos.
 - Manivela: Este tipo de persianas, mediante un giro rotatorio a la derecha o izquierda suben o bajan.
 - Correa: Con este tipo de persianas, el usuario debe realizar un movimiento hacia abajo si desea abrirla o un movimiento hacia arriba si desea bajarla
 - Automáticas: Este tipo de persianas, disponen de un mecanismo que mediante una pulsación poseen un motor que sube o baja sin esfuerzo la misma.
- **Lámparas:** Este dispositivo contiene las siguientes variantes:
 - Pulsación: Este tipo es el más común en las viviendas, simplemente realizando un gesto de pulsación se enciende o se apaga el dispositivo.
 - Graduales: Para interactuar con este tipo de lámparas, el usuario dispone de un aparato que va conjuntamente la lámpara y mediante un movimiento hacia delante o hacia atrás, permite subir o bajar la intensidad de la luz que emite.
- **Climatizadores:** Los climatizadores se encienden o se apagan mediante un gesto de pulsación sobre un botón que está en el mando que los controla y posteriormente si se desea subir o bajar la temperatura se debe ir realizando pulsaciones para aumentarla o disminuirla.
- **Cafetera:** Este dispositivo existen de varios tipos:
 - Manuales: Estas son las de siempre, que requiere que el usuario realice todo el trabajo.
 - Automáticas: En este tipo, el usuario únicamente debe realizar la pulsación de un botón y el resto se hace de forma automática. Para apagarla o pararla simplemente debe realizar de nuevo una pulsación.

- **Televisión:** Para realizar la interacción con este dispositivo el usuario dispone de un mando a distancia en donde debe pulsar los botones asignados para cada función. Si desea encenderla o apagarla debe realizar la pulsación al botón correspondiente al igual que si desea cambiar el canal o subir el volumen.

Estos son algunos de los dispositivos que se encuentran en una vivienda estándar. Una vez analizados estos dispositivos, el siguiente paso para definirlos de forma abstracta es como llevar esta interacción al dispositivo móvil con el que poder capturarlos y mapearlos como acciones utilizando el mismo, para que a la hora de que el usuario interactúe con ellos mediante el dispositivo móvil sea de la forma más natural e intuitiva posible. Obviamente los gestos anteriormente identificados no pueden plasmarse en su totalidad o de la forma en la que se interactúa normalmente pero se ha intentado mantener en la medida de lo posible que sean lo más parecido posible con el mundo real. Vamos a analizar a continuación los dispositivos que hemos comentado anteriormente para así determinar los gestos abstractos asociados y como podrían ser:

- **Puerta :** Para el caso de la puerta, debido a que el gesto completo es el de giro rotatorio bien a la derecha o a la izquierda y gesto hacia delante o hacia atrás para abrir o cerrar la misma, con un dispositivo móvil no veía necesario realizar el gesto rotatorio completo seguido con un gesto hacia delante o hacia atrás, ya que resultaría muy dificultoso y no muy usable el que el usuario girara por completo el dispositivo ya que necesitaría ambas manos para realizar el gesto y posteriormente realizar el gesto de tirar o empujar con el móvil. Para solventar este inconveniente, resulta más usable el realizar un giro parcial hacia la derecha o a la izquierda para abrir o cerrar y obviar el segundo gesto ya que considero innecesario el realizarlo.
- **Persiana:** Para este dispositivo, nos encontramos otro claro ejemplo de que al querer abstraer el gesto natural con el que se interactúa normal, no podemos abstraerlo en su totalidad, al menos en el caso en el que la persiana sea del tipo de la manivela, ya que resultaría tedioso y muy poco práctico el realizar el gesto para el usuario. En este caso el gesto más natural para interactuar con el sería el de realizar el gesto de elevar o bajar el dispositivo para su subida o bajada ya que resulta un gesto muy sencillo e intuitivo de realizar.
- **Lámparas:** En el siguiente dispositivo sí que hemos podido abstraer el gesto y realizarlo lo más parecido posible a su interacción natural, ya que el gesto de pulsación podemos asociarlo a un gesto de agitar el dispositivo para encender o apagarlo y en el caso de graduar el intensidad, el gesto más adecuado para ello sería el de elevar o bajar el dispositivo para ajustarla. También para apagar el dispositivo podemos realizar el gesto de girar el dispositivo en posición boca abajo sobre cualquier superficie y esto apagaría la lámpara.

- **Climatizadores:** Con este dispositivo, al igual que con la lámpara, se ha podido abstraer su interacción sin mayor dificultad, ya que el encendido del dispositivo se realiza mediante una pulsación que es sustituida por el gesto de agitar el dispositivo y el de aumentar y disminuir la temperatura por el de elevar o bajar el dispositivo. Otro gesto que podría abstraerse es el de apagar el dispositivo que bien podría realizarse con la agitación del dispositivo o bien con el volteo del dispositivo boca abajo.
- **Cafetera:** En este caso, los gestos que podrían interpretarse serían en el caso del encendido del dispositivo como se realiza mediante una pulsación, lo asociamos con el agitado del dispositivo móvil y su apagado con otro gesto de agitar o bien volteando el dispositivo ya que ese volteo del dispositivo boca abajo recuerda a la pulsación de un botón.
- **Televisión:** Para este último caso a analizar, el encendido o apagado del mismo sería similar al de la cafetera o la lámpara. Para el cambio de canal o la subida o bajada del volumen serían los siguientes. Para cambiar de canal tendríamos el giro parcial a la derecha que pasaría al canal siguiente y el giro parcial a la izquierda que volvería al canal anterior y para la subida del volumen o bajada realizaríamos el gesto de elevación o de bajada del dispositivo. Se han elegido estos gestos ya que el usuario se sentirá cómodo en la realización de estos gestos y no supondría ningún problema a la hora de recordarlos ya que son lo más similar posible a la forma tradicional de utilización.

Una vez analizados los dispositivos y vistas sus características de funcionamiento y cómo podríamos abstraer sus gestos a otros que se puedan realizar con un dispositivo móvil, lo siguiente es definir un patrón de interacción con ellos mediante la creación de unos gestos básicos que se adapten a las necesidades de los mismos y no suponga una forma extraña de utilización. Estos gestos básicos se mapearán a unas acciones realizadas con el dispositivo siempre teniendo en cuenta una serie de factores que determinen si ese gesto es viable y satisfactorio para su inclusión en la librería de gestos. Los factores que se han empleado para determinarlos son los siguientes:

- **Gestos intuitivos:** Deben ser fácilmente mapeables en acciones reales que el usuario pueda realizar.
- **Facilidad al uso:** Es importante que sean gestos fáciles para el usuario y se puedan reproducir de forma continuada sin que el usuario deba prestar atención al dispositivo por si está realizando mal el gesto. Los gestos deben ser lo más sencillos posibles.
- **Debe ser con un dispositivo móvil:** Los gestos deben ser realizados con un dispositivo móvil que normalmente el usuario lleva en una mano, esto conlleva que no tiene por qué utilizar las dos manos para realizarlos ya que entonces no serían del todo sencillos y obligaría al usuario a utilizar ambas.

- **Solamente gestos espaciales:** El usuario no tiene la obligación de estar pendiente de la pantalla del dispositivo ni interactuar con ella todo el tiempo. Con estos gestos se pretende que el usuario no debe realizar pulsaciones en la pantalla salvo las imprescindibles para la selección del dispositivo.
- **Tienen que ser capturados mediante el dispositivo móvil:** Todos los gestos tienen que ser capturados por el dispositivo móvil que el usuario lleva en el dispositivo, no pueden ayudarse de elementos externos que no se encuentren en el dispositivo.
- **Liberar al usuario de elementos externos para el control de su vivienda:** Otra de las premisas que se intenta cumplir, es la de poner controlar todos los elementos mediante un único dispositivo y no disponer de un elemento de interacción por dispositivo. Con un único elemento poder controlar la totalidad de los mismos que componen la vivienda.
- **Familiarización que la sociedad tiene con los dispositivos móviles:** En la actualidad la sociedad está acostumbrada a los dispositivos móviles y no supone un cambio tan radical a la forma de interactuar con una casa, ya que día a día el mundo está cada vez más conectado y el disponer en la palma de tu mano del control total de tu vivienda es una idea cada vez menos fantástica.

Estos son los factores que han llevado a la elección de los gestos básicos que vamos a enumerar y las razones de por qué se ha elegido un dispositivo móvil para llevarlos a cabo. Los gestos son los siguientes:

- Gesto arriba
- Gesto abajo
- Gesto derecha
- Gesto izquierda
- Gesto shake(agitar)
- Gesto boca abajo
- Gesto boca abajo(Luminosidad)
- Gesto touch
- Gesto abrir puerta
- Gesto cerrar puerta

Una vez seleccionados estos gestos básicos, pueden ser necesarios otros gestos más complejos para controlar dispositivos que con estos gestos simples no satisface la interacción natural con algún dispositivo de la vivienda inteligente. Para ello, estos gestos simples pueden combinarse para crear otros más complejos y así disponer de una variedad más amplia de gestos espaciales con los que poder interactuar con la vivienda. Actualmente la librería dispone de tres gestos complejos implementados pero en una fase muy temprana y este listado de gestos puede crecer conforme se detecten más gestos complejos que cumplan los factores anteriormente mencionados.

Los gestos complejos son los siguientes:

- Gesto puente
- Gesto saludo
- Gesto pincel

Como se ha mencionado antes, este listado puede aumentar su tamaño en futuras ampliaciones de la librería para que poco a poco las posibilidades de la misma continúen creciendo tanto en variedad como en complejidad pero sin renunciar a los factores que determinan si un gesto es viable o no para ser incluido.

En las siguientes secciones del documento se proporcionara información más detallada y explícita sobre todos los gestos incluidos en la librería, tanto de los gestos simples como de los gestos complejos.

Definición gestual abstracta simple

Gesto arriba

Este gesto se activa cuando el usuario alza el dispositivo hacia arriba desde una posición horizontal. Se detecta de forma continua, es decir si se mantiene el gesto puede aplicarse para interacción con un dispositivo que requiera una acción continua por ejemplo. Aunque también puede utilizarse de forma unitaria siempre reconfigurando el gesto. Este gesto se detecta analizando la posición del eje de la y, ya que si el valor de la posición actual en el eje de las y es mayor que el valor anterior en el mismo eje, entonces se ha elevado el dispositivo y el gesto es capturado.



Ilustración 3-1 Gesto Arriba

Gesto abajo

Este gesto se activa cuando el usuario baja el dispositivo hacia abajo desde una posición horizontal. Se detecta de forma continua, es decir si se mantiene el gesto puede aplicarse para interacción con un dispositivo que requiera una acción continua por ejemplo. Aunque también puede utilizarse de forma unitaria siempre reconfigurando el gesto. Este gesto se detecta analizando la posición del eje de la y, ya que si el valor de la posición actual en el eje de las y es menor que el valor anterior en el mismo eje, entonces se ha bajado el dispositivo y el gesto es capturado.



Ilustración 3-2 Gesto Abajo

Gesto derecha

Este gesto se activa cuando el usuario inclina hacia la derecha desde una posición horizontal. Se detecta de forma continua, es decir si se mantiene el gesto puede aplicarse para interacción con un dispositivo que requiera una acción continua por ejemplo. Aunque también puede utilizarse de forma unitaria siempre reconfigurando el gesto. Este gesto se detecta analizando la posición del eje de la x, ya que si el valor de la posición actual en el eje de las x es menor que el valor anterior en el mismo eje, entonces se ha girado el dispositivo hacia la derecha y el gesto es capturado.



Ilustración 3-3 Gesto Derecha

Gesto izquierda

Este gesto se activa cuando el usuario inclina hacia la izquierda desde una posición horizontal. Se detecta de forma continua, es decir si se mantiene el gesto puede utilizarse por ejemplo para pasar bajar el canal del televisor, el giro de una luz movable, o como veremos en otro gesto, realizar el gesto de apertura de una puerta. Este gesto se detecta analizando la posición del eje de la x, ya que si el valor de la posición actual en el eje de las x es mayor que el valor anterior en el mismo eje, entonces se ha girado el dispositivo hacia la derecha y el gesto es capturado.



Ilustración 3-4 Gesto Izquierda

Gesto shake (agitar dispositivo)

Este gesto se activa cuando el usuario agita el dispositivo. Se detecta cuando ha ocurrido una variación en los valores de los ejes de coordenadas aplicando una fuerza determinada durante un tiempo determinado.



Ilustración 3-5 Gesto Shake

Gesto boca Abajo

Este gesto se activa cuando el usuario pone boca abajo el dispositivo desde una posición horizontal. Al igual que se puede asignar una acción al gesto boca abajo, también se le puede asignar una acción al gesto contrario una vez realizado este gesto, es decir asignar una acción al gesto boca arriba. Este gesto se detecta analizando la posición del eje de la z, ya que si el valor de la posición actual en el eje de la z es menor que el valor anterior en el mismo eje, entonces se ha colocado el dispositivo boca abajo y si por el contrario el valor de la posición actual en el eje de las z es mayor que el valor anterior en el mismo eje, entonces el dispositivo esta boca arriba.



Ilustración 3-6 Gesto Boca Abajo

Gesto boca abajo luminosidad

Este gesto se activa cuando el usuario pone boca abajo el dispositivo desde una posición horizontal. La diferencia con el anterior gesto, es que el anterior puede realizarse en cualquier lugar, pero este gesto es realizarlo poniendo la pantalla del dispositivo sobre cualquier objeto. Al igual que se puede asignar una acción al gesto boca abajo, también se le puede asignar una acción al gesto contrario una vez realizado este gesto, es decir asignar una acción al gesto boca arriba. Para detectar este gesto se utiliza la medición de la intensidad lumínica ambiental, si el valor de la intensidad actual es menor a la anterior entonces se detecta el gesto y viceversa.

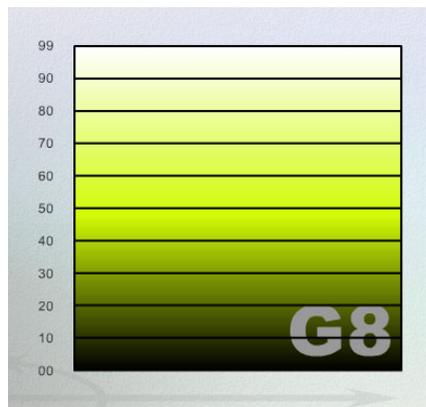


Ilustración 3-7 Gesto Boca Abajo Luminosidad

Gesto touch

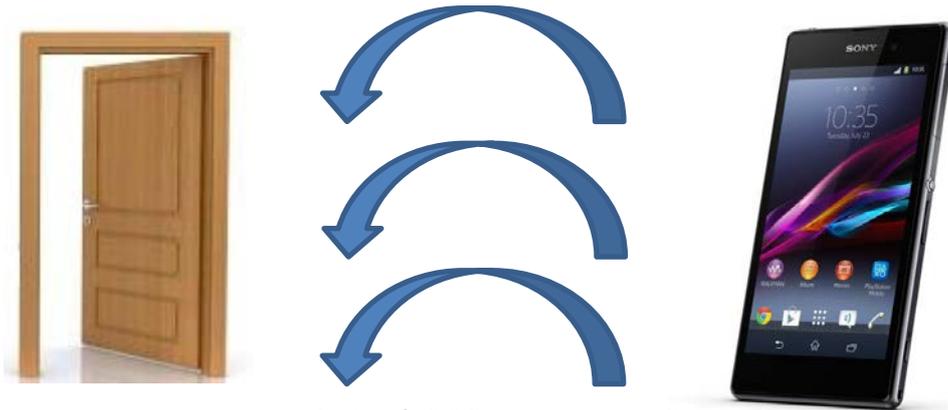
Este gesto se activa cuando el usuario acerca el dispositivo a un elemento. Este gesto se activa cuando la distancia del dispositivo con el elemento es menor a 1cm. Al igual que se puede asignar una acción cuando el dispositivo se acerca, también se puede asignar otra acción cuando el dispositivo se aleja a más de 10 cm.



Ilustración 3-8 Gesto touch

Gesto abrir puerta(X3)

Este gesto se activa cuando el usuario gira tres veces a la izquierda el dispositivo. Se detecta utilizando las mismas características que el gesto a la izquierda pero durante tres tiempos.



Gesto cerrar puerta(X3)

Este gesto se activa cuando el usuario gira tres veces a la derecha el dispositivo. Se detecta utilizando las mismas características que el gesto a la derecha pero durante tres tiempos.



Definición gestual abstracta compleja

El objeto del trabajo es que mediante los gestos simples donde se pueden capturar interacciones gestuales elementales. Claro que para construir interfaces gestuales con las que controlar una vivienda inteligente o algo más amplio una Smart City puede que no sea suficiente con estos gestos más simples. Con lo cual, mediante esta librería de gestos se pueden construir otros más complejos encadenando gestos simples para construirlos y poder asignarlos a diferentes dispositivos.

Los gestos complejos se pueden componer de varios movimientos espaciales simples o básicos que se deben utilizar en un orden determinado. Estos gestos complejos dan muchas posibilidades al usuario ya que pueden combinarse los movimientos simples para generar otros más complejos. Esta posibilidad que proporciona la librería abre a los futuros desarrolladores unas posibilidades muy amplias de construir interfaces gestuales muy elaboradas ya que al tener externalizados los gestos y poder ser asociados a cualquier dispositivo constituye un marco más generalizado que si estuvieran asignados por defecto a un dispositivo. A continuación describiré unos ejemplos de gestos complejos que pueden realizarse concatenando los gestos simples de la librería.

Gesto puente

Este gesto se activa cuando el usuario realiza los siguientes gestos en orden:

- **Gesto Arriba:** Este gesto se activa cuando el usuario alza el dispositivo hacia arriba desde una posición horizontal.
- **Gesto Derecha:** Este gesto se activa cuando el usuario inclina hacia la derecha desde una posición horizontal.
- **Gesto Abajo:** Este gesto se activa cuando el usuario baja el dispositivo hacia abajo una posición horizontal.

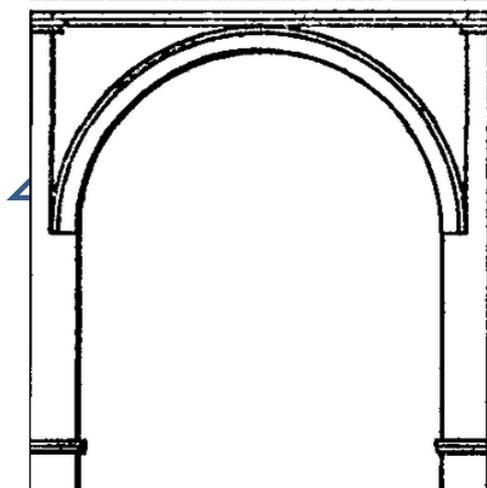


Ilustración 3-11 Gesto Puente

Gesto saludo

Este gesto se activa cuando el usuario realiza los siguientes gestos en orden:

- **Gesto Arriba:** Este gesto se activa cuando el usuario alza el dispositivo hacia arriba desde una posición horizontal.
- **Gesto Derecha:** Este gesto se activa cuando el usuario inclina hacia la derecha desde una posición horizontal.
- **Gesto Izquierda:** Este gesto se activa cuando el usuario inclina hacia la izquierda desde una posición horizontal.

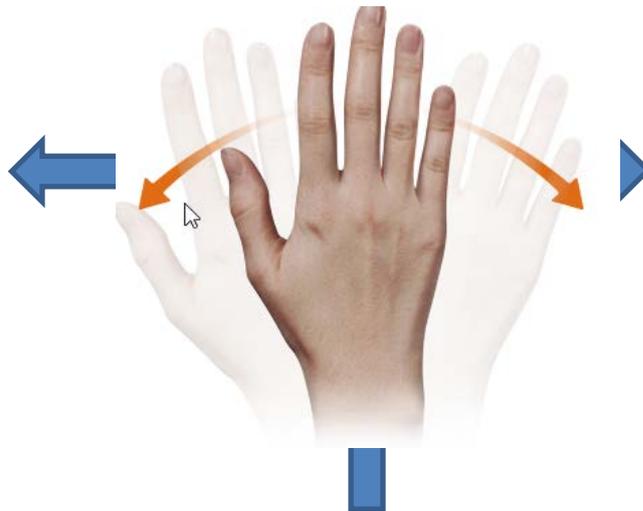


Ilustración 3-12 Gesto Saludo

Gesto pincel

Este gesto se activa cuando el usuario realiza los siguientes gestos en orden:

- **Gesto Abajo:** Este gesto se activa cuando el usuario baja el dispositivo hacia abajo una posición horizontal.
- **Gesto Arriba:** Este gesto se activa cuando el usuario alza el dispositivo hacia arriba desde una posición horizontal



Ilustración 3-15 Gesto Pincel

Alternativas existentes en el mercado

En este apartado se comentaran las alternativas existentes en el mercado, las distintas empresas y universidades, que tienen interés en seguir explotando la tecnología de gestos espaciales ya sea utilizando la tecnología de sensores o no.

Aplicaciones

Android Control Software R2

Esta aplicación de la empresa Creston (<http://www.creston.es/>). La solución que propone esta empresa es el control total de una smart home mediante una interfaz ya sea para Tablet o Smartphone. Dispone de aplicación tanto para Android como para iOS, ambas disponibles en los markets correspondientes. Los inconvenientes de esta aplicación a primera vista es una interfaz muy sobrecargada y sobria. Otro inconveniente es la no explotación de las posibilidades que ofrecen actualmente estos dispositivos como son el uso de los sensores que disponen.



Ilustración 3-14 Android Control Software R2

- <https://play.google.com/store/apps/details?id=com.id8group.r2&hl=es>

BeHome 247

Esta aplicación de la empresa BeHome247 (<http://www.behome247.com>). El objetivo de esta aplicación es poder tener el control total de tu casa conectada. Permite controlar la temperatura de toda la casa, monitorizarla y programar la temperatura. Otra funcionalidad que tiene es descartar las llaves físicas y poder asignar contraseñas de acceso a la casa y consultar en todo momento el acceso a la misma.



Ilustración 3-15 BeHome 247

Loxone

Esta aplicación permite controlar la totalidad de la smart home mediante tu Smartphone o Tablet. A primera vista de la aplicación se puede apreciar que la empresa ha optado por la sencillez en cuanto a la interfaz. Se nutre de botones para el control de la aplicación frente al uso de los gestos naturales que puede realizar el usuario. La aplicación está disponible tanto para Android como iOS, además de también disponer de una aplicación web para el control de la vivienda.



Ilustración 3-16 Loxone

- <https://play.google.com/store/apps/details?id=com.loxone.app&hl=es>

RedEyes

Esta aplicación desarrollada por la empresa ThinkFlood, se encontraba disponible al principio exclusivamente para iOS, pero posteriormente la empresa decidió portarla también a Android, debido a la gran aceptación del sistema operativo de Google.

El modo de operar con esta aplicación es realmente sencillo. Se trata simplemente de usar un hardware (escogido según la necesidad requerida), distribuido por la propia compañía, que mediante la aplicación nos permite controlar un sin fin de distintos dispositivos que van desde televisores o equipos de música hasta el propio Aire Acondicionado de la casa, la iluminación, la central de Alarmas o tu propio PC.



Ilustración 3-17 Red Eyes

- <https://play.google.com/store/apps/details?id=com.thinkflood.redeye&hl=es>

Magus Free 3D gestos, Launcher

Esta aplicación utiliza los gestos espaciales para controlar en este caso tu Smartphone, se basa en uno de los objetivos principales de este trabajo, ya que el usuario puede seleccionar que gesto quiere utilizar y a que acción puede asociarlo. Se activa automáticamente en el dispositivo y se ejecuta en segundo plano. "Es una idea bastante revolucionaria en realidad , ya que crea más opciones para aquellos que están buscando más formas de controlar sus dispositivos, o aquellos con alguna limitación que buscan como utilizar su dispositivo de otra forma a causa de alguna discapacidades."

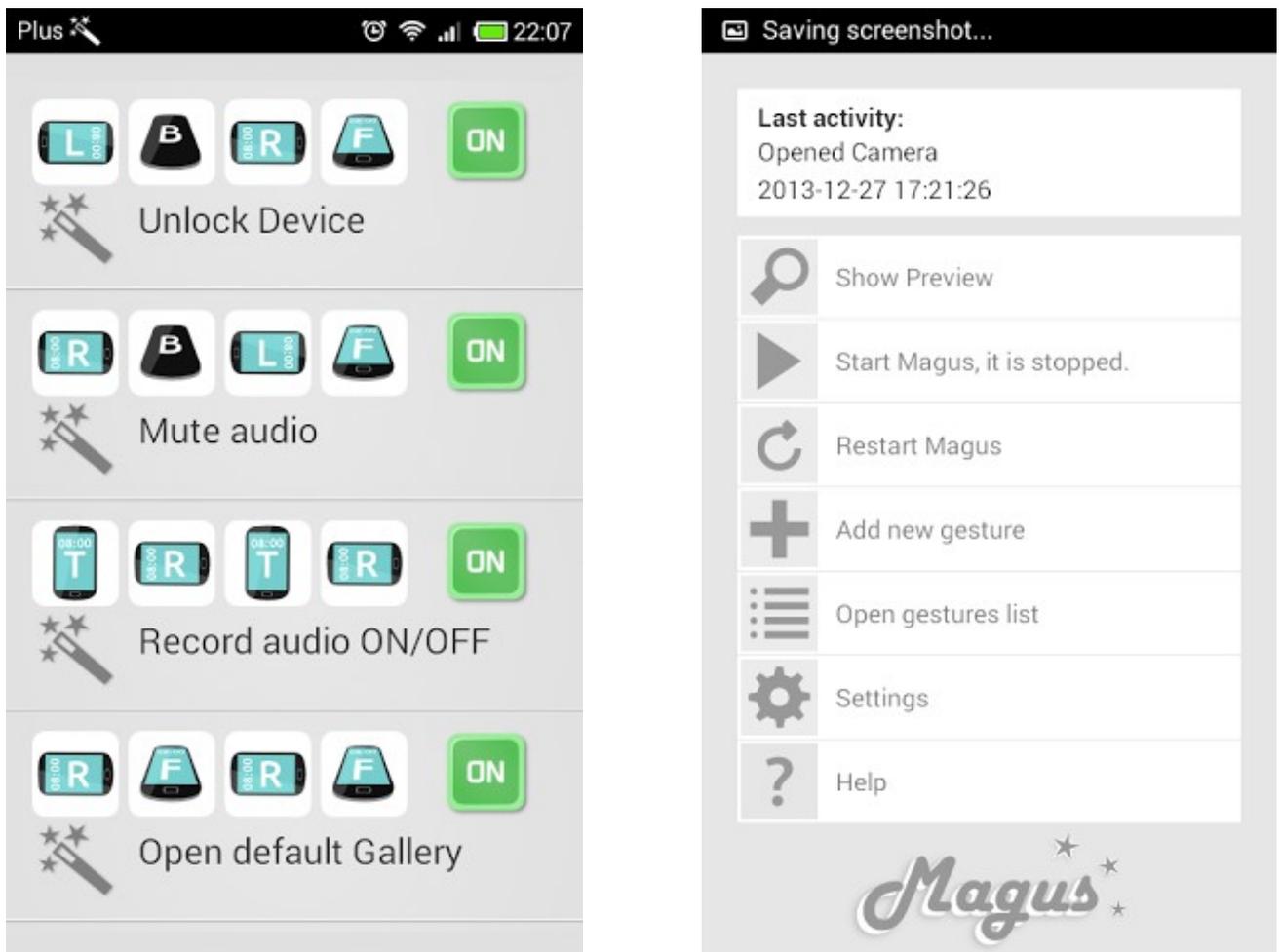


Ilustración 3-18 Magus Free 3D gestos, Launcher

- <https://play.google.com/store/apps/details?id=com.stmp.magusfree&hl=es>

Empresas

Actualmente existen muchas empresas cuyas investigaciones y proyectos van enfocados a explotar esta capacidad de los sistemas inteligentes gracias a la multitud de sensores de los que disponen y que permiten realizar un gran número de tipos de interacción con el usuario.

Logbar

Esta joven empresa o startup, presento en el Macworld/iWorld 2014 (Celebrado en marzo) un anillo inteligente (Ring). La novedad de este dispositivo es que es capaz de reconocer los movimientos del usuario.



Ilustración 3-19 Logbar

La idea es, que con un pequeño anillo se puede controlar varios dispositivos electrónicos, aplicaciones móviles y ordenadores mediante gestos espaciales. Con los incorporados sensores en accesorio, el usuario puede introducir información de texto, transmitir datos sobre los pagos electrónicos y recibir notificaciones de dispositivos móviles por vibración y retro iluminación LED.



Ilustración 3-20 Especificaciones Logbar

Ring está equipado con un botón especial, que activa el control. El usuario sólo tiene que poner el anillo en el dedo, presionar el botón y dibujar en el aire uno de los gestos. Por ejemplo, puede ser icono de la música para poner en marcha el reproductor de música, la cámara – para activar grabación de vídeo o un sobre – para comprobar los mensajes de texto. Configuración de los gestos se realiza con el cliente móvil.

Integrada función OCR (reconocer el texto) ofrece la oportunidad con gestos escribir texto en Twitter o Facebook, enviar mensajes SMS y de correo electrónico, e incluso cambiar los canales del televisor. El producto puede ser utilizado en conjunción con gafas Google Glass y el reloj inteligente iWatch. Declaran apoyo para los sistemas operativos iOS 7, Android y Windows Phone.

La empresa lanzo el proyecto en Kickstarter para financiarlo, y lo consigo el 4 de abril del 2014(<https://www.kickstarter.com/projects/1761670738/ring-shortcut-everything>).

Google

La empresa responsable de Android, con su último wereable, las Google Glass, según filtraciones actuales este dispositivo permite gestos espaciales para realizar distintas acciones.

```
CONNECTIVITY_BT_PAIRING_LOCAL_CANCEL
CONNECTIVITY_BT_PAIRING_REMOTE_CANCE
HEAD_GESTURES_HEAD_WAKE_ENABLED = ne
HEAD_GESTURES_HEAD_WAKE_DISABLED = n
HEAD_GESTURES_HEAD_WAKE_ANGLE_SETTIN
EYE_GESTURES_WINK_ENABLED = new User
EYE_GESTURES_WINK_DISABLED = new Use
EYE_GESTURES_WINK_CALIBRATION_SUCCES
EYE_GESTURES_WINK_CALIBRATION_FAILUR
EYE_GESTURES_WINK_TAKE_PHOTO = new U
BROWSER_PAGE_REQUESTED_EXPLICIT = ne
BROWSER_PAGE_REQUESTED_IMPLICIT = ne
BROWSER_PAGE_LOADED = new UserEventA
BROWSER_MENU_OPTION = new UserEventA
BROWSER_TWO_FINGER_ZOOM = new UserEv
BROWSER_ERROR = new UserEventAction(
GUEST_MODE_TURNED_ON = new UserEvent
GUEST_MODE_TURNED_OFF = new UserEven
COMPANION_SCREENCAST = new UserEvent
```

Ilustración 3-21 Funciones Google Glass

Los más interesantes son HEAD_GESTURES, EYE_GESTURES (incluyendo WINK_TAKE_PHOTO), BROWSER_TWO_FINGER_ZOOM, GUEST_MODE y COMPANION_SCREENCAST. Teniendo en cuenta que las dos primeras son las categorías, uno podría suponer que habrá varios tipos de cada uno, a pesar de que sólo hay una de cada lista en estos momentos.

VTOUCH (SAMSUNG)

Otra empresa que ve que el control por interfaces no es lo que debería imperar en el futuro, es Vtouch. Esta empresa esta especialidad en el control de dispositivos con gestor naturales realizados con las manos y con los ojos gracias a una cámara 3D que dispone el usuario y que calcula con exactitud dónde estamos apuntando.

VTouch es diferente de otras soluciones existentes. Todos los controladores de gestos en el mercado son de tipo ratón que requiere un cursor en la pantalla para seleccionar cierto punto. Los usuarios tienen que detectar primero el cursor y mover su cuerpo o manos para poner el cursor en el punto que se desea seleccionar.

Con la solución propuesta por esta empresa, el usuario puede señalar directamente lo que ven con sus dedos.

Una de la empresas que vio el potencial de esta empresa es Samsung, que en un trabajo conjunto. Prepara la inclusión de una tecnología en las televisiones inteligentes que permitirá controlar mediante gestos la interfaz del televisor sin la necesidad de utilizar un mando a distancia, además de gestionar la iluminación del hogar.

La compañía surcoreana quiere aumentar la calidad de sus televisores inteligentes para poder ofrecer algo más de lo que ofrecen todas las demás compañías en este sector, según ha publicado Wall Street Journal. Por ello, Samsung quiere trabajar en la inclusión de esta tecnología.

Para ello se hará uso de cámaras especiales para detectar los movimientos, de forma similar al dispositivo de Kinect de Microsoft.

La tecnología de VTouch utiliza algoritmos que hacen un seguimiento tanto de las manos como de los ojos de los usuario, lo que permite filtrar los comandos falsos de mejor manera que otras soluciones que combinan varias partes del cuerpo.

Según lo planeado por las compañías, esta tecnología llegará a los mercados de los televisores en 2016.



Ilustración 3-22 VTouch

LEAP MOTION

Esta empresa, fundada en 2010, fabrica y comercializa un dispositivo sensor que detecta movimientos de la mano y los dedos como si fuera un ratón convencional para el PC. Con este sensor no se requiere el contacto con el ordenador para utilizarlo.



Ilustración 3-23 Leap Motion

Es un pequeño dispositivo periférico USB que está diseñado para ser colocado en un escritorio físico, mirando hacia arriba. El uso de dos cámaras monocromáticas IR y tres LEDs infrarrojos, el dispositivo observa un área aproximadamente semiesférica, a una distancia de aproximadamente 1 metro (3 pies). Los LED generan un patrón 3D de puntos de luz IR y las cámaras generan cerca de 300 cuadros por segundo de datos reflejadas, que se envía a través de un cable USB al ordenador, donde es analizada por el software de controlador de salto de movimiento usando "complejo matemáticas" de una manera que no ha sido revelada por la empresa, de alguna manera la síntesis de los datos de posición 3D mediante la comparación de los marcos 2D generados por las dos cámaras.

El área de observación más pequeña y mayor resolución del dispositivo diferencia el producto de la Kinect, que es más adecuado para el seguimiento de todo el cuerpo en un espacio del tamaño de una sala de estar. En una demostración CNET, el salto ha demostrado. Realizar tareas como navegar por un sitio web, el uso de gestos de pellizcar para hacer zoom en los mapas, dibujos de alta precisión, y la manipulación de complejas visualizaciones de datos en 3D.

Aunque el dispositivo propone una idea muy ambiciosa, sí que es cierto que este producto con se desenvuelve muy bien con interfaces 3D, como podemos leer en un artículo del 30 de agosto del 2013(http://www.technologyreview.es/read_article.aspx?id=43792). Este dispositivo permite a la gente desarrollar sus propias aplicaciones mediante las herramientas de desarrollo que proporciona este dispositivo, pero la queja más generalizada es que lo más dificultoso es realizar aplicaciones en 3D y inventar controles de movimiento para que el usuario ya sea experto o no, entiendan como usarla.

LG

Este dispositivo de LG, se basa en uno de los objetivos principales de este trabajo, en el control por gestos espaciales. Se asimila a este trabajo ya que en nuestro caso utilizamos un dispositivo Android para el control de una SmartHome y en el caso de LG se utiliza un mando a distancia con sensores para controlar una Smart TV. El caso de LG es menos ambicioso que el nuestro, ya que solo se centra en sus dispositivos y nuestro trabajo es mucho más generalizado.



Ilustración 3-24 Magic Control

Este Magic Control permite el usuario dibujar gestos espaciales que el televisor entiende y realiza las acciones que el usuario quiere. Por ejemplo dibujar un número y automáticamente se pondrá el canal que está asociado a ese número. Si se desplaza el dispositivo de derecha a izquierda, se vuelve al canal anterior o si navegas por internet realizando un círculo podrás hacer zoom a la página web que estas consultando.

Otra de las características que añade, es el control por voz, hablándole al mando a distancia, puede realizarse también acciones que interpreta la Smart TV.

LG, al igual que Samsung, también piensa que el control gestual para los televisores es el camino más lógico. Por ello en sus últimos modelos también ha añadido cámaras que detectan al usuario y los gestos que realiza para así liberarlo también del omnipresente mando a distancia para controlar la Smart TV.



Ilustración 3-25 Smart TV LG

4. ARQUITECTURA DE UNA PROPUESTA DE SOLUCIÓN

Requisitos

Los requisitos para poder utilizar la aplicación son los siguientes:

- Un Smartphone o Tablet Android con la versión 4.0 o superior del sistema operativo.
- Un servidor central en la vivienda donde se conectaran todos los dispositivos de la vivienda. Existen muchas opciones en este punto, el usuario elegiría el que más se adapte a sus necesidades.
- Una conexión a internet tanto en la vivienda como en su dispositivo.



Ilustración 4-1 Smart Home conectada

Arquitectura general

La arquitectura como bien hemos comentado al comienzo del documento, puede ser :

- Arquitectura centralizada
- Arquitectura distribuida
- Arquitectura descentralizada

Para esta propuesta supondremos que disponemos de una arquitectura centralizada.

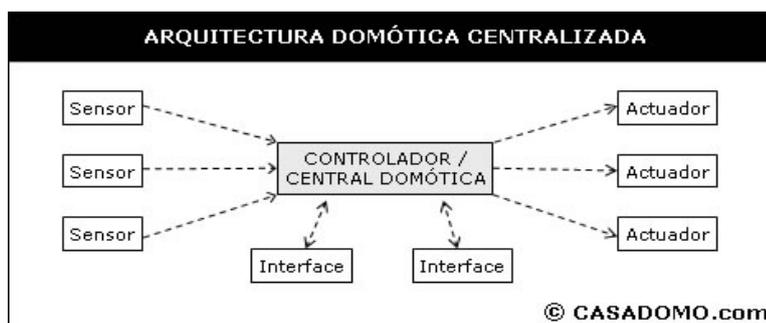


Ilustración 4-2 Arquitectura domótica centralizada

Dispondremos de un servidor central donde estarán conectados todos los sensores o dispositivos y mediante gestos espaciales con nuestro Smartphone o Tablet enviaremos al servidor central la acción que se desea realizar.

El usuario al entrar a su vivienda inteligente, elegiría el dispositivo que desea controlar y utilizaría el grupo de espaciales que le brinda la aplicación para realizar las distintas

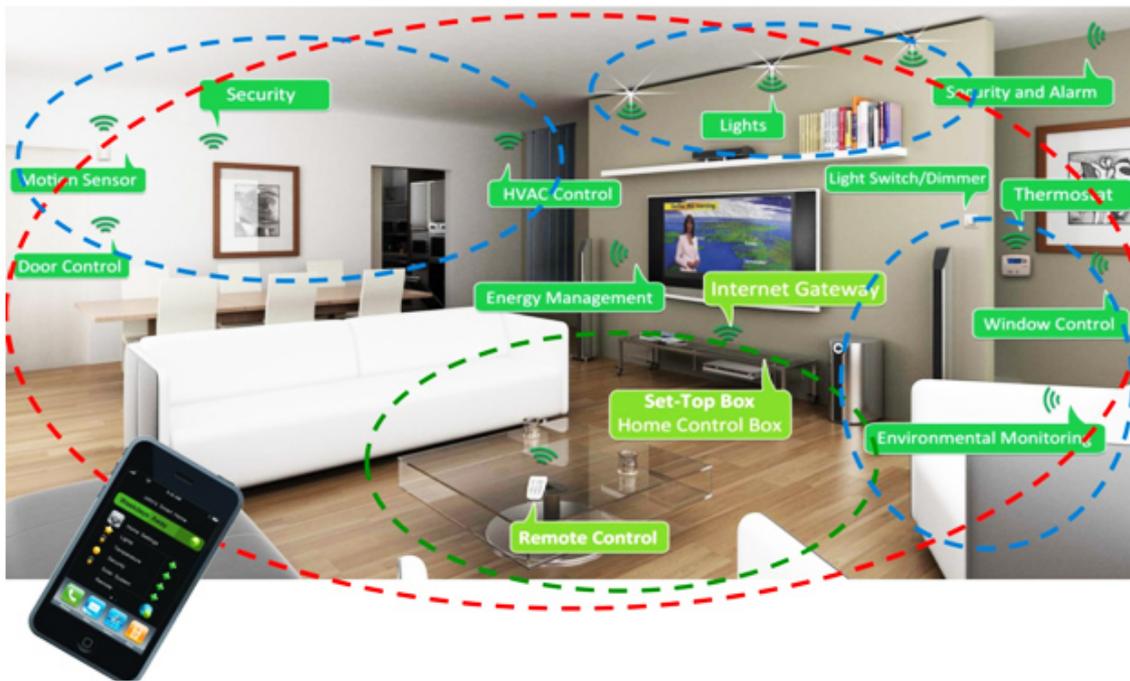


Ilustración 4-3 Interacción con una Smart Home

Sensores utilizados

Los sensores utilizados para la detección de los movimientos en dispositivos Android, son los siguientes:

Acelerómetro

Un acelerómetro es un elemento sensor que mide la aceleración, así como el ángulo de inclinación, la rotación, la vibración, el choque y la gravedad. Para ofrecer funcionalidad en un teléfono inteligente, el software del acelerómetro debe traducir los datos proporcionados por el sensor. Los teléfonos inteligentes utilizan varios tipos de acelerómetros, siendo el elemento sensor y el software las principales diferencias entre ellos. Cuando se aplica a un teléfono inteligente, un acelerómetro puede cambiar automáticamente la orientación del dispositivo de vertical a horizontal. Esta pequeña pieza de hardware se encuentra ya en casi todos los teléfonos inteligentes y viene en dos tipos: acelerómetro de dos dimensiones y de tres dimensiones. La diferencia entre éstos no solo radica en su estructura, sino en sus capacidades.

El más robusto de ambos es el acelerómetro de 3 dimensiones y es el más ampliamente utilizado. Así, este sensor puede entender la posición de nuestro dispositivo fácilmente para, por ejemplo, cambiar la pantalla de posición horizontal a posición vertical dependiendo de cómo lo sostengamos.

Es lógicamente uno de los sensores más utilizados por las aplicaciones, pues permite leer en modo vertical u horizontal entre diversas aplicaciones. Para la navegación web a veces es más fácil leer en modo horizontal por el tamaño de las letras, mientras que algunos desarrolladores bloquean el cambio de posición para que sus apps tengan una sola forma de visualizarse.

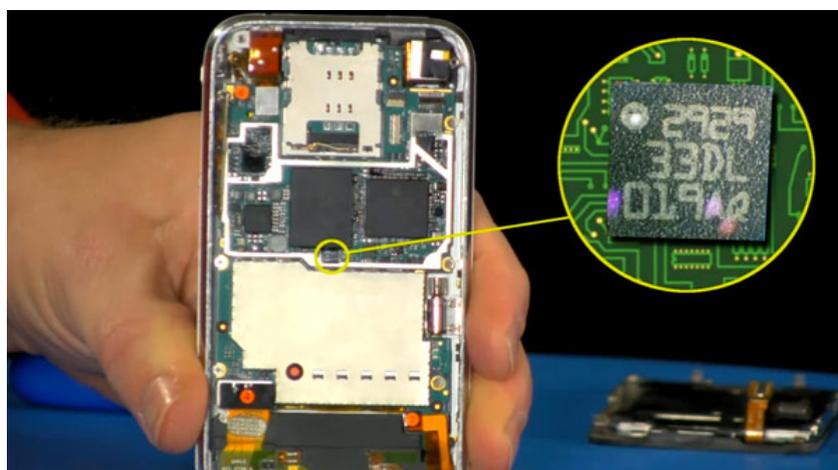


Ilustración 4-4 Sensor Acelerometro

El acelerómetro de nuestros dispositivos, trabaja con las tres dimensiones (X, Y, Z). Detecta las tres coordenadas y son las que utilizamos en este trabajo para detectar la mayoría de los gestos espaciales.

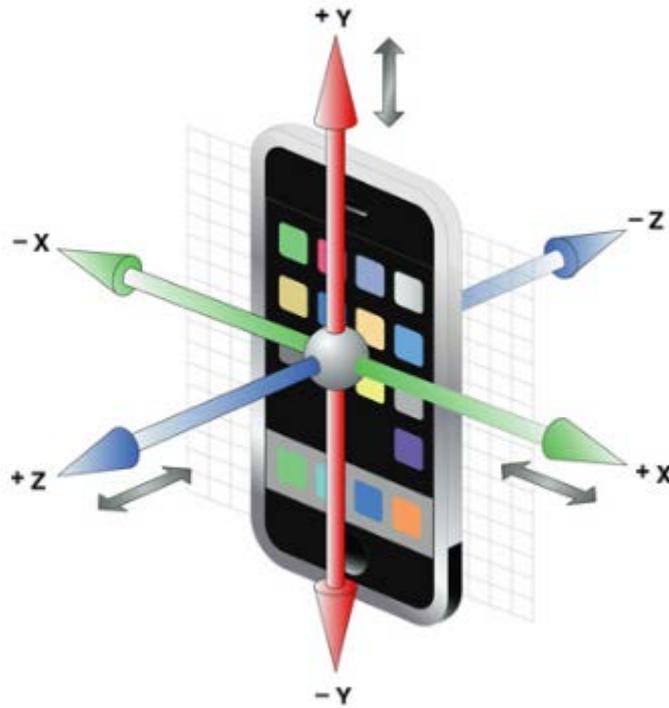


Ilustración 4-5 Eje de coordenadas

Sensor de luminosidad

Un sensor fotoeléctrico o fotocélula es un dispositivo electrónico que responde al cambio en la intensidad de la luz. Estos sensores requieren de un componente emisor que genera la luz, y un componente receptor que percibe la luz generada por el emisor. Todos los diferentes modos de sensado se basan en este principio de funcionamiento. Están diseñados especialmente para la detección, clasificación y posicionado de objetos; la detección de formas, colores y diferencias de superficie, incluso bajo condiciones ambientales extremas.

Los sensores de luz se usan para detectar el nivel de luz y producir una señal de salida representativa respecto a la cantidad de luz detectada. Un sensor de luz incluye un transductor fotoeléctrico para convertir la luz a una señal eléctrica y puede incluir electrónica para condicionamiento de la señal, compensación y formateo de la señal de salida.

El sensor de luz más común es el LDR -Light Dependant Resistor o Resistor dependiente de la luz-.Un LDR es básicamente un resistor que cambia su resistencia cuando cambia la intensidad de la luz. Existen tres tipos de sensores fotoeléctricos, los sensores por barrera de luz, reflexión sobre espejo o reflexión sobre objetos.

Sensor de proximidad

Un sensor de proximidad es un transductor que detecta objetos o señales que se encuentran cerca del elemento sensor.

Existen varios tipos de sensores de proximidad según el principio físico que utilizan. Los más comunes son los interruptores de posición, los detectores capacitivos, los inductivos y los fotoeléctricos, como el de infrarrojos.

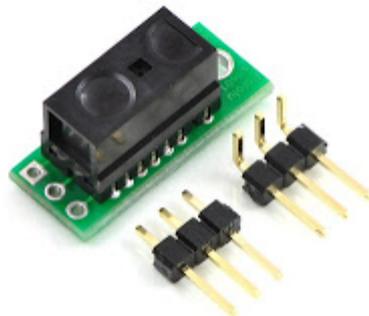


Ilustración 4-6 Sensor de proximidad

Para la propuesta que deseamos realizar en concreto, utilizaremos los sensores comentados anteriormente y el sistema operativo Android. Con esto, implementaremos una solución que desarrolle los patrones abstractos descritos en el capítulo anterior del documento.

Implementación de los patrones de interacción

En este capítulo del documento vamos implementar y explicar cómo han sido implementados los movimientos descritos anteriormente y que se ha utilizado para capturarlos y con qué sensores ha sido realizado.

Gesto arriba

Este gesto se activa cuando el usuario alza el dispositivo hacia arriba desde una posición horizontal. Para la detección de este movimiento se ha utilizado el acelerómetro que captura variaciones en las 3 coordenadas del espacio(X, Y, Z). Para ello la aplicación captura la variación de posición de la Y a partir de +5, ya que en estos tipos de sensores las coordenadas van desde -9 a +9 en las tres coordenadas. Se ha elegido que sea a partir de +5 ya que si capturara cualquier variación en la coordenada X hacia positivo, no sería totalmente fiable este gesto.

Gesto abajo

Este gesto se activa cuando el usuario baja el dispositivo hacia abajo desde una posición horizontal. Para la detección de este movimiento se ha utilizado el acelerómetro que captura variaciones en las 3 coordenadas del espacio(X, Y, Z). Para ello la aplicación captura la variación de posición de la Y a partir de -5, ya que en estos tipos de sensores las coordenadas van desde -9 a +9 en las tres coordenadas. Se ha elegido que sea a partir de +5 ya que si capturara cualquier variación en la coordenada X hacia negativo, no sería totalmente fiable este gesto.

Gesto derecha

Este gesto se activa cuando el usuario inclina hacia la derecha desde una posición horizontal. Para la detección de este movimiento se ha utilizado el acelerómetro que captura variaciones en las 3 coordenadas del espacio(X, Y, Z). Para ello la aplicación captura la variación de posición de la X a partir de +5, ya que en estos tipos de sensores las coordenadas van desde -9 a +9 en las tres coordenadas. Se ha elegido que sea a partir de +5 ya que si capturara cualquier variación en la coordenada X hacia positivo, no sería totalmente fiable este gesto.

Gesto izquierda

Este gesto se activa cuando el usuario inclina hacia la izquierda desde una posición horizontal. Para la detección de este movimiento se ha utilizado el acelerómetro que captura variaciones en las 3 coordenadas del espacio(X, Y, Z). Para ello la aplicación captura la variación de posición de la X a partir de -5, ya que en estos tipos de sensores las coordenadas van desde -9 a +9 en las tres coordenadas. Se ha elegido que sea a partir de +5 ya que si capturara cualquier variación en la coordenada X hacia negativo, no sería totalmente fiable este gesto.

Gesto shake (agitar dispositivo)

Este gesto se activa cuando el usuario agita el dispositivo. Este gesto también hace uso del acelerómetro para ser detectado y para ello, este gesto captura la variación de la posición en las tres coordenadas y tiene un mínimo de fuerza configurable donde se especifica a partir de qué nivel de fuerza se considera como que se ha realizado un gesto shake.

Gesto boca abajo

Este gesto se activa cuando el usuario pone boca abajo el dispositivo desde una posición horizontal. Al igual que se puede asignar una acción al gesto boca abajo, también se le puede asignar una acción al gesto contrario una vez realizado este gesto, es decir asignar una acción al gesto boca arriba. Para la detección de este movimiento se ha utilizado el acelerómetro que captura variaciones en las 3 coordenadas del espacio(X, Y, Z). Para ello la aplicación captura la variación de posición de la Z a partir de +5, ya que en estos tipos de sensores las coordenadas van desde -9 a +9 en las tres coordenadas. Se ha elegido que sea a partir de +5 ya que si capturara cualquier variación en la coordenada X hacia positivo, no sería totalmente fiable este gesto.

Gesto boca abajo luminosidad

Este gesto se activa cuando el usuario pone boca abajo el dispositivo desde una posición horizontal. Este gesto hace uso ya no del acelerómetro, si no del sensor de luminosidad del que disponen la gran mayoría de dispositivos. Para realizar la captura de este gesto, se ha configurada para que su implementación detecte cuando la luminosidad baja de 20(La luminosidad va desde 0 a 100). Si dicha luminosidad es menos a 20, el gesto es capturado.

Gesto touch

Este gesto se activa cuando el usuario acerca el dispositivo a un elemento. Este gesto se activa cuando la distancia del dispositivo con el elemento es menor a 1cm. Al igual que se puede asignar una acción cuando el dispositivo se acerca, también se puede asignar otra acción cuando el dispositivo se aleja a más de 10 cm.

Gesto abrir puerta(X3)

Este gesto se activa cuando el usuario gira tres veces a la izquierda el dispositivo. Los factores que determinan su implementación son los mismos que rigen la detección del gesto a la izquierda.

Gesto cerrar puerta(X3)

Este gesto se activa cuando el usuario gira tres veces a la derecha el dispositivo. Los factores que determinan su implementación son los mismos que rigen la detección del gesto a la derecha.

Los siguientes gestos complejos al no estar implementados directamente en la librería, lo que se indica a continuación es un ejemplo de cómo podría implementarse y cuál podría ser su uso, ya que el objetivo principal del trabajo se centraba en el desarrollo de una librería gestual simple pero viendo el potencial que proporciona, pues podríamos extenderla a un uso más complejo y atrevernos a diseñar e implementar gestos más elaborados y complejos si fueran necesarios.

Gesto puente

Este gesto se activa cuando el usuario realiza los siguientes gestos en orden:

- **Gesto Arriba:** Este gesto se activa cuando el usuario alza el dispositivo hacia arriba desde una posición horizontal.
- **Gesto Derecha:** Este gesto se activa cuando el usuario inclina hacia la derecha desde una posición horizontal.
- **Gesto Abajo:** Este gesto se activa cuando el usuario baja el dispositivo hacia arriba desde una posición horizontal.

Gesto saludo

Este gesto se activa cuando el usuario realiza los siguientes gestos en orden:

- **Gesto Arriba:** Este gesto se activa cuando el usuario alza el dispositivo hacia arriba desde una posición horizontal.
- **Gesto Derecha:** Este gesto se activa cuando el usuario inclina hacia la derecha.
- **Gesto Izquierda:** Este gesto se activa cuando el usuario inclina hacia la izquierda.

Gesto pincel

Este gesto se activa cuando el usuario realiza los siguientes gestos en orden:

- **Gesto Arriba:** Este gesto se activa cuando el usuario alza el dispositivo hacia arriba desde una posición horizontal.
- **Gesto Abajo:** Este gesto se activa cuando el usuario baja el dispositivo hacia arriba desde una posición horizontal.

5. APLICACIÓN AL ÁMBITO
DE LAS VIVIENDAS
INTELIGENTES

Implementación

Para este trabajo, se ha implementado una pequeña aplicación donde se utiliza una simulación de una Smart home donde mediante los gestos que hemos descrito en el capítulo anterior. Para ello se han implementado varios escenarios donde se interactuar con distintos dispositivos, entre ellos son las luces, el aire acondicionado, una cafetera o un estor.

A continuación comentaremos como se ha realizado la aplicación que utiliza algunos de los movimientos que se han especificado.

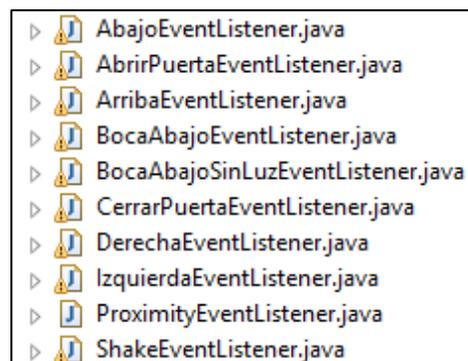


Ilustración 5-1 Listado de gestos espaciales

En la imagen de arriba, se muestran los diferentes gestos que se han implementado. Todos estos gestos son implementados idénticamente, pero su funcionalidad interna es muy distinta unos de otros. Todos los gestos implementan la clase **SensorEventListener**. Esta clase es la que se encarga de capturar y recibir los datos cuando el valor del sensor varía o se modifica.

```
public class ShakeEventListener implements SensorEventListener {
```

Ilustración 5-2 Declaración de la clase ShakeEventListener

En este diagrama, se muestra como todos los gestos implementados en la librería, implementan la interface **SensorEventListener**. Esta interface es propia de la librería Android y le proporciona a los gestos los métodos necesarios para obtener los datos de los distintos sensores que son los utilizados para detectar los distintos gestos espaciales que utilizamos. En esta interface el método que nos interesa es el **onSensorChanged**

```
/**
 * Called when sensor values have changed.
 * <p>See {@link android.hardware.SensorManager SensorManager}
 * for details on possible sensor types.
 * <p>See also {@link android.hardware.SensorEvent SensorEvent}.
 *
 * <p><b>NOTE:</b> The application doesn't own the
 * {@link android.hardware.SensorEvent event}
 * object passed as a parameter and therefore cannot hold on o it.
 * The object may be part of an internal pool and may be reused by
 * the framework.
 *
 * @param event the {@link android.hardware.SensorEvent SensorEvent}.
 */
public void onSensorChanged(SensorEvent event);
```

Ilustración 5-3 Método onSensorChanged de la interface SensorEventListener

Este método proporciona a cada gesto el atributo SensorEvent que es el que contiene los datos captados por el sensor indicado.

Cada gesto implementado dispone del método **onSensorChanged**, donde se capturan los datos del sensor que se desea controlar.

```
@Override
public void onSensorChanged(SensorEvent event) {

    Float x = event.values[0];
    Float y = event.values[1];
    Float z = event.values[2];
    incrementa = Boolean.FALSE;
    if(y.intValue() > 5) {
        if( antiguaY > 0 && y.intValue() == 9) {
            antiguaY = y;
            incrementa = Boolean.TRUE;
        } else if(antiguaY < y) {
            antiguaY = y;
            incrementa = Boolean.TRUE;
        }
    }
    mUpListener.toUp(incrementa);
}
}
```

Ilustración 5-4 Implementación del método onSensorChanged de la clase Shake

En este ejemplo que tenemos arriba, se encarga de capturar el gesto que realiza el usuario subiendo el dispositivo. Internamente solamente captura el movimiento en el eje de coordenadas Y, ya que es el único que nos interesa para este gesto en concreto. Los tres ejes de coordenadas X, Y, Z van desde -9 a +9 en todos los dispositivos. Para que el gesto sea detectado correctamente solamente se captura cuando el dispositivo pasa del valor 5 en el eje de la Y, ya que si no se dejara esa condición al menor movimiento con el dispositivo se capturaría el gesto cuando en la realidad no se correspondería con un movimiento intencionado.

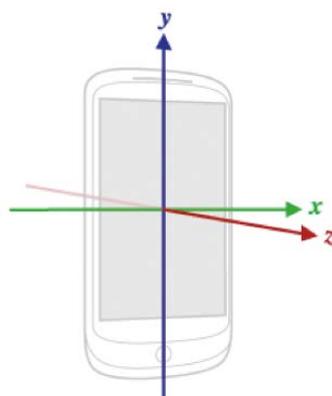


Ilustración 5-5 Eje de coordenadas dispositivo móvil

```

@Override
public void onSensorChanged(SensorEvent event) {

    Float z = event.values[2];
    bocaAbajo = Boolean.FALSE;
    Log.v("Intensidad Z", "Valor de la Z: "+ z);
    if(z.intValue() == -9) {
        antiguaZ = z;
        bocaAbajo = Boolean.TRUE;
    } else if(z.intValue() == 9) {
        antiguaZ = z;
        bocaAbajo = Boolean.FALSE;
    }

    mBocaAbajoListener.toBocAbajo(bocaAbajo);
}

```

Ilustración 5-6 Implementación del método onSensorChanged de la clase gesto Boca Abajo

Este otro ejemplo se corresponde con el gesto que realiza el usuario cuando pone el dispositivo boca abajo. Para ello en este caso se utiliza el eje de coordenadas Z donde se captura cuando llega a -9 ya que ese valor es cuando el dispositivo está totalmente girado hacia abajo.

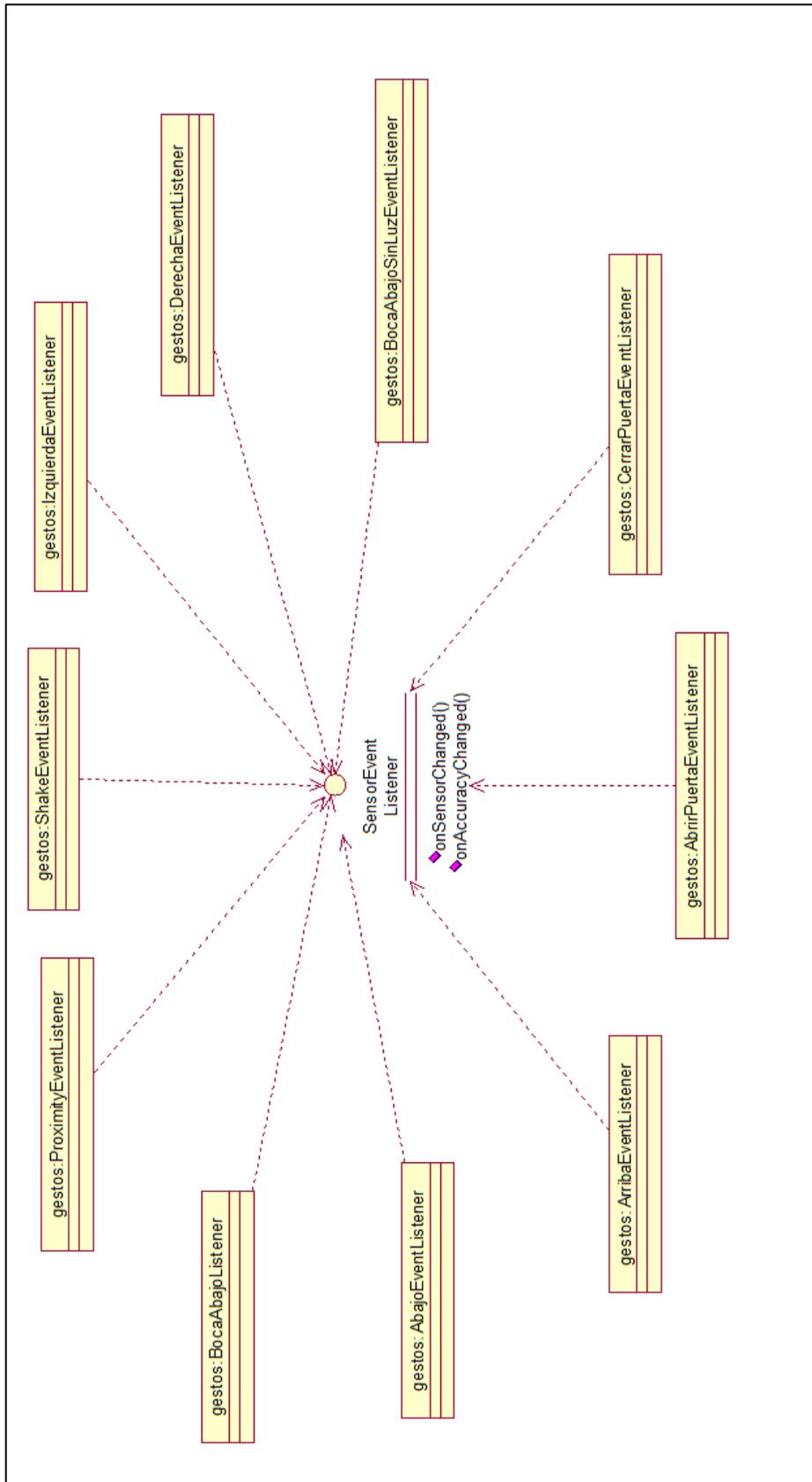


Ilustración 5-7 Diagrama de clases de la librería de gestos

Internamente cada gesto dispone de una interface que internamente tienen un método que es el que es utilizado en la clase principal donde se quiere capturar el movimiento. Es decir, cada gesto dispone de una interface con un método que es el utilizado en la clase principal que instancia el movimiento.

```
/**
 * Interface for shake gesture.
 */
public interface OnShakeListener {

    /**
     * Called when shake gesture is detected.
     */
    void onShake();
}
```

Ilustración 5-8 Interface propia de cada gesto

Como he comentado, esta interface es la se utiliza en cada clase que controle un dispositivo, este método de la interface se encarga de avisar a la clase que lo implementa si se ha realizado o no satisfactoriamente el gesto en cuestión. Para ello se pueden utilizar atributos Booleanos para que posteriormente la clase que lo implementa pueda utilizarlos para determinar si realizan o no la acción que desean asociar a ese gesto espacial. A continuación pondremos un ejemplo para explicar mejor todo el proceso y entender cómo funciona la estructura creada.

Lo realizaremos con el gesto de mover el dispositivo hacia arriba.

```
public interface OnUpListener {
    /**
     * Se llama cuando se detecta el movimiento de arriba
     */
    public void toUp(Boolean incrementa);
}
```

Ilustración 5-9 Interface propia del gesto arriba

Primeramente creamos la interface que en este caso se llama **OnUpListener** que contiene el método **toUp**, el cual tiene como parámetro de entrada un Booleano que determina si el dispositivo esta hacia arriba o no. Posteriormente declaramos en la clase un atributo de tipo privado donde decimos que es un **OnUpListener**, el cual podrá llamar al método **toUp** y enviarle el valor Booleano si se han cumplido las condiciones necesarias.

```
/** OnUpListener that is called when up is detected. */
private OnUpListener mUpListener;
```

Ilustración 5-10 Objeto OnUpListener

```

@Override
public void onSensorChanged(SensorEvent event) {

    Float y = event.values[1];
    incrementa = Boolean.FALSE;
    if(y.intValue() > 5) {
        if( antiguaY > 0 && y.intValue() == 9) {
            antiguaY = y;
            incrementa = Boolean.TRUE;
        } else if(antiguaY < y) {
            antiguaY = y;
            incrementa = Boolean.TRUE;
        }
    }
    mUpListener.toUp(incrementa);
}

```

Envió del valor Booleano que determina si ha sido correcto o no.

Ilustración 5-11 Implementación del método onSensorChanged

A continuación como la clase implementa la interface **SensorEventListener**, dispone del método **onSensorChanged** que recibe como parámetro un **SensorEvent** que es el utilizado para obtener las variaciones del sensor y determinar si el dispositivo está en la posición correcta para que se realice el gesto hacia arriba. Si se han cumplido todas las condiciones necesarias, al final se envía al método de la interface que hemos declarado dentro de la clase un valor Booleano que determina si el gesto ha sido correcto o no.

Este valor es recogido por la clase que está instanciando el gesto hacia arriba de la siguiente forma.

```

mArribaDetector.setmUpListener(new OnUpListener() {

    @Override
    public void toUp(Boolean incrementa) {
        if(incrementa) {
            subiendoPersiana = Boolean.TRUE;
            bajandoPersiana = Boolean.FALSE;
            new executeJson().execute(incrementa, Boolean.FALSE, subiendoPersiana, bajandoPersiana);
        }
    }
});

```

Obtiene el valor que ha sido enviado anteriormente por el método onSensorChanged.

Ilustración 5-12 Método que recibe los datos del gesto

Una vez obtenido el dato, la clase que desea controlar el dispositivo con el gesto hacia arriba, utilizando el Booleano recibido, puede realizar cualquier acción que desee asociar al gesto, ya que si ha sido satisfactorio el gesto, el valor del Booleano será verdadero o por el contrario falso si el gesto no ha sido satisfactorio. Una vez determinado si ha sido satisfactorio como es en este caso, se decide realizar una llamado REST a un servicio web para indicar al dispositivo que realice una acción determinada que en este caso sería el de subir la persiana.

Para la realización de las peticiones REST, se ha externalizado a una clase las peticiones GET y PUT. De este modo solo es necesario instanciar la clase donde se quiera utilizar.

- PUT
 - Funcionalidad → Se corresponde con el servicio al cual queremos interactuar
 - Código de Acción → La acción que se desea realizar.
 - Valor → Si la acción dispone de un campo valor, se especifica el valor al que queremos situarlo (Por ejemplo la temperatura de una habitación).

La respuesta contiene los datos sobre el estado del dispositivo en concreto y si la acción ha sido satisfactoria.

- GET
 - Funcionalidad → Se corresponde con el servicio al cual queremos interactuar

```
public JSONObject getOperation(String funcionalidad) {  
  
    JSONObject respuesta = null;  
    HttpClient httpClient = new DefaultHttpClient();  
    HttpGet get = new HttpGet(Constants.dirServidor+funcionalidad);  
    get.setHeader("content-type", "application/json");  
  
    HttpResponse response;  
  
    String respStr = null;  
    try {  
        response = httpClient.execute(get);  
        respStr = EntityUtils.toString(response.getEntity());  
        respuesta = new JSONObject(respStr);  
    } catch (Exception e1) {  
        e1.printStackTrace();  
    }  
    return respuesta;  
}
```

```
public JSONObject putOperation(String funcionalidad, String codigoAccion, String valor) {  
  
    JSONObject respuesta = null;  
    HttpClient httpClient = new DefaultHttpClient();  
    HttpPut put = new HttpPut(Constants.dirServidor+funcionalidad);  
    put.setHeader("content-type", "application/json");  
  
    HttpResponse response;  
  
    String respStr = null;  
    try {  
        JSONObject jsonObj = new JSONObject();  
  
        jsonObj.put("action", codigoAccion);  
        if(valor != null) {  
            jsonObj.put("value", valor);  
        }  
        StringEntity entity = new StringEntity(jsonObj.toString());  
        put.setEntity(entity);  
        response = httpClient.execute(put);  
        respStr = EntityUtils.toString(response.getEntity());  
        respuesta = new JSONObject(respStr);  
  
    } catch (Exception e1) {  
        e1.printStackTrace();  
    }  
    return respuesta;  
}
```

Ilustración 5-13 Implementación de los métodos GET y PUT de la clase OperationRest

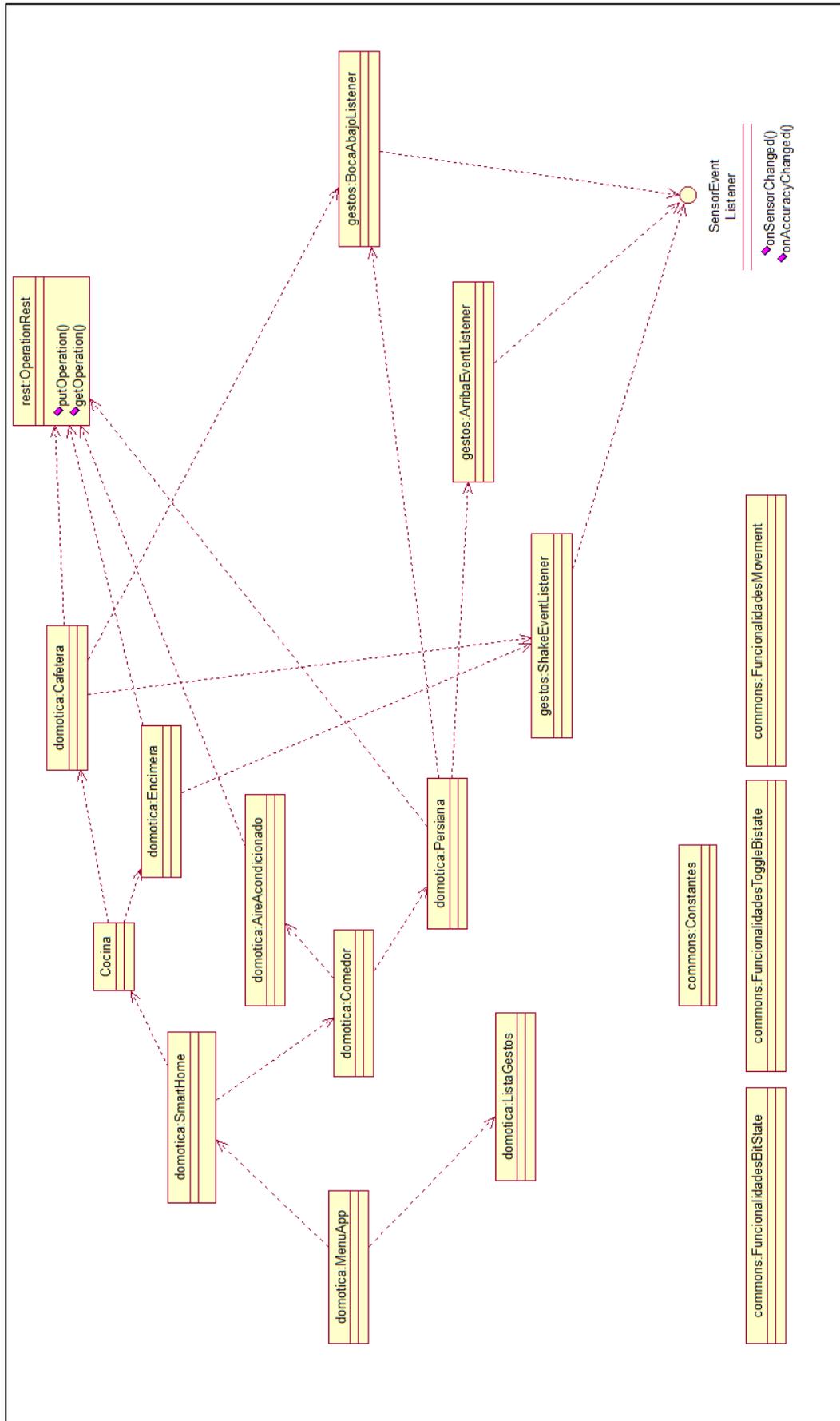


Ilustración 5-14 Diagrama de clases de la aplicación de ejemplo

Funcionalidades de la aplicación

En este apartado describiremos las distintas funcionalidades de la aplicación desarrolladas para defender este documento y en donde hacemos uso de algunos de los gestos implementados.

Primero disponemos de una aplicación donde están implementados todos los gestos de la librería y que únicamente los detecta y muestra un mensaje para la comprobación de que la detección del gesto es el correcto. A continuación vamos a exponer situaciones concretas donde creemos que los gestos definidos en nuestra librería encajarían perfectamente y sería considerado gestos naturales para el control de dichos dispositivos. Obviamente al tratarse de una librería de gestos, cada aplicación puede utilizar estos gestos como quisiera el usuario utilizar o bien como el desarrollador pensara que son la forma mas optima de utilización.

Control de iluminación

Esta funcionalidad permite al usuario controlar estos dispositivos de distintas formas. Una de estas funcionalidades es la de encender o apagar la luz de una habitación mediante simplemente agitando el Smartphone o Tablet. Otra de esas funcionalidades es la de poder controlar la iluminación de forma gradual mediante el gesto de abajo a arriba para incrementar la luminosidad o de arriba abajo para disminuirla. Y siempre que la habitación en concreto dispusiera de un sensor de luz ambiental, la aplicación puede automáticamente encender o apagar la luz dependiendo del nivel de luz que se disponga en ese momento.

Gestos asociados:

- Shake
- Gesto arriba
- Gesto Abajo

Estos gestos son los que serían considerados como más naturales para la interacción con el control de la iluminación.

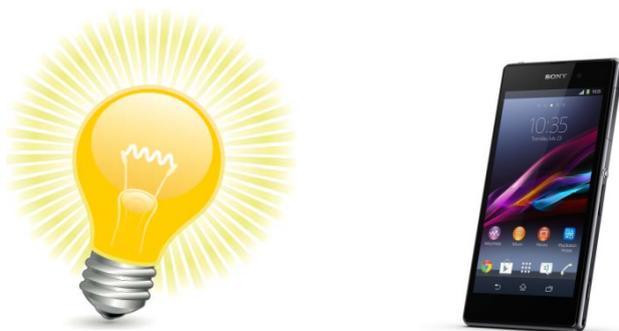


Ilustración 5-15 Control de iluminación

Control de las persianas

Otra de las opciones que permite la aplicación es la de poder subir o bajar las persianas de la habitación correspondiente mediante gestos naturales. Gestos asociados:

- Shake
- Gesto arriba
- Gesto Abajo



Ilustración 5-16 Control de las persianas

Control de la calefacción

Esta funcionalidad permite al usuario controlar la calefacción. El usuario podrá encender o apagar la calefacción del baño a conveniencia desde cualquier rincón de la vivienda, consultar en todo momento en qué estado se encuentra y subir la temperatura o bajarla. Gestos asociados:

- Gesto arriba
- Gesto Abajo

Estos gestos son los que serían considerados como más naturales para la interacción con el control de la iluminación.



Ilustración 5-17 Control de la calefacción

Control de encendido y apagado de la cafetera

Esta funcionalidad, permite al usuario encender o apagar remotamente desde cualquier sitio y consultar el estado de la misma. Gestos asociados:

- Shake
- Gesto Abajo



Ilustración 5-18 Control de la cafetera

6. TRABAJOS FUTUROS Y

MEJORAS

Este apartado de la memoria está dedicado a exponer en qué puntos puede mejorar este proyecto.

Librería para iOS

La mejora más interesante es la de poder portar la librería desarrollada para Android a los terminales con el sistema operativo de Apple. Aunque Android domina casi por totalidad el mercado de Smartphone, el segundo en el ranking es Apple con su iPhone y sería muy rentable realizarlo para todos estos usuarios que apuestan por iOS.



Ilustración 6-1 Aplicación para iOS

iPhone es una línea de teléfonos inteligentes diseñado y comercializado por Apple Inc. Ejecuta el sistema operativo móvil iOS, antes conocido como "iPhone OS" hasta mediados de 2010. El iPhone dispone de una cámara de fotos de 8 megapíxeles y un reproductor de música (equivalente al del iPod) además de software para enviar y recibir mensajes de texto y mensajes de voz. También ofrece servicios de Internet como leer correo electrónico, cargar páginas web y conectividad por Wi-Fi. La primera generación de teléfonos eran GSM cuatribanda con la tecnología EDGE; la segunda generación ya incluía UMTS con HSDPA.2; la sexta generación ya incluía LTE.

Apple Inc. anunció el iPhone en 2007,³ tras varios rumores y especulaciones que circulaban desde hacía meses.⁴ El iPhone se introdujo inicialmente en los Estados Unidos el 29 de junio de 2007. Fue nombrado «Invento del año» por la revista Time en el 2009.⁵ El 11 de julio de 2008 el iPhone 3G salió a la venta, estando disponible en 22 países⁶ y en 70 para

finales de año. Esta versión soportaba la transmisión de datos por 3G a una velocidad más rápida y tecnología AGPS.2.

Para poder desarrollar la aplicación para iPhone es necesario que el desarrollador conozca el lenguaje de programación Objective C. Es un lenguaje de programación basado en C, y que ha sido modificado para poder trabajar orientado a objetos, con lo cual pueden crearse clases, objetos, variables de instancias, métodos, encapsulación, etc.

También es indispensable el conocimiento de Cocoa, el Framework de desarrollo para iPhone. Cocoa Touch es un API para la creación de programas para el iPad, iPhone y iPod Touch de la compañía Apple Inc. Proporciona una capa de abstracción al sistema operativo iOS y se basa en el set de herramientas que proporciona el API de Cocoa para crear programas sobre la plataforma Mac OS X.

Librería para Windows Phone

Debido a la gran aceptación que está teniendo el sistema operativo de Microsoft desde Windows Phone 8. Sería muy interesante realizar esta aplicación para este sistema operativo.

Windows Phone 8 es la segunda generación del sistema operativo para móviles Windows Phone, como se confirmó oficialmente por Microsoft en un seminario de MSDN en agosto de 2011 y visto de antemano en Microsoft 'sneak peek' de Windows Phone el 20 de junio de 2012. Su versión RTM fue lanzada el 14 de septiembre de 2012 y fue liberado a los consumidores el 29 de octubre de 2012.



Ilustración 6-2 Dispositivo Windows Phone

El desarrollo de aplicaciones para Windows Phone puede hacerse empleando dos tipos de implementaciones:

- **Microsoft Silverlight** que permite realizar aplicaciones que contengan transiciones y efectos visuales. Silverlight permite el desarrollo de aplicaciones basadas en XAML. Silverlight para Windows Phone incluye el Microsoft .NET Compact Framework, que hereda de la arquitectura .NET Framework, el CLR y la ejecución de código administrado, soporta un subconjunto de las librerías de clases de .NET Framework y contiene clases diseñadas exclusivamente para .NET Compact Framework. Este soporte incluye el Base Class Library, una colección de clases que soportan lectura y escritura de ficheros, manipulación XML y manejo de gráficos. Cada aplicación que es ejecutada en Windows Phone OS 7.0 CTP se ejecuta dentro de un proceso en el motor de ejecución .NET Compact Framework.
- **Microsoft XNA Framework** es una implementación nativa de .NET Compact Framework que incluye un amplio conjunto de bibliotecas de clases, específicos para el desarrollo de juegos, por ejemplo para el manejo de dispositivos de entrada, tratamiento de sonidos y vídeos, carga de modelos y texturas, uso de ficheros de forma transparente a la plataforma en la que se ejecute, desarrollo de juegos online, etc... Permite desarrollar juegos para Windows Phone OS 7.0 CTP, Xbox 360, Zune HD y Windows 7.

Librería mas flexible

Otro trabajo interesante sería realizar la librería más flexible, es decir poder realizar una abstracción más elaborada sobre los gestos complejos y permitir al desarrollador que escogiera esta librería, el poder realizar de una forma más intuitiva y fácil el poder crear y gestionar los gestos complejos, ya que en el documento hemos comentado como podrían implementarse pero no se encuentran implementados en el interior de la librería. Actualmente pueden crearse pero sería encadenando desde fuera de la librería un gesto simple detrás de otro.

7. CONCLUSIONES

8. BIBLIOGRAFÍA

- <http://eopsoa.blogspot.com.es/2009/02/eclipse-model-framework-cosa-fina-oiga.html>
- [http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software))
- <http://androidcookbook.com/Recipe.seam;jsessionid=92EE6ACFBF67A2278AEC4A2208A1251D?recipeld=2871>
- <http://developer.android.com/reference/android/hardware/SensorEvent.html>
- <http://developer.android.com/training/gestures/scroll.html>
- <http://canalys.com/newsroom/smart-mobile-device-shipments-exceed-300-million-q1-2013>
- <http://www.casadomo.com/noticiasDetalle.aspx?c=14>
- <http://es.wikipedia.org/wiki/Dom%C3%B3tica>
- <http://www.andalucialab.org/blog/interaccion-natural-tecnologia-de-reconocimiento-de-gestos/>
- <http://www.buenastareas.com/ensayos/El-Futuro-De-La-Tecnolog%C3%ADa-Gestual/3739231.html>
- <http://www.ciens.ucv.ve:8080/genasig/sites/interaccion-humano-comp/archivos/clase10-IHCII2012-TecnicasInteraccion.pdf>
- <https://code.google.com/p/android-motion-detection/>
- <http://www.xataka.com/otros/mit-s-cityhome-con-gestos>
- <http://www.smartcities.es/about/>
- http://es.wikipedia.org/wiki/Ciudad_inteligente
- http://www.endesaeduca.com/Endesa_educa/recursos-interactivos/smart-city/smart-citizen
- <http://androtalk.es/2013/09/dymotics-domotica-con-sello-espanol-al-alcance-de-nuestro-smartphone/>

- <http://www.applesfera.com/ios/homekit-reunira-todas-las-soluciones-de-domotica-en-un-solo-lugar>
- http://www.elotrolado.net/noticia_nest-la-compania-de-domotica-de-google-abre-su-api-y-encuentran-la-primera-vulnerabilidad_24447
- <http://es.wikipedia.org/wiki/ZigBee>
- <https://www.iconfinder.com/>
- <http://developer.android.com/index.html>
- <http://stackoverflow.com/>
- <http://developer.android.com/training/gestures/movement.html>
- <http://4ndroid.com/thinkflood-domotica-en-tu-android-con-redeyes/>