



Universidad de Deusto

Marco de colaboración entre dispositivos inteligentes para la reducción de su consumo energético en espacios compartidos

Tesis doctoral presentada por Juan López de Armentia Mendizabal
dentro del Programa de Doctorado en Ingeniería Informática y
Telecomunicación

Dirigida por el Dr. Diego López de Ipiña González de Artaza

El doctorando

El director

Bilbao, junio de 2016

Marco de colaboración entre dispositivos inteligentes para la reducción de su consumo energético en espacios compartidos

Autor:

Juan López de Armentia Mendizabal

Director:

Dr. Diego López de Ipiña González de Artaza

De forma especial, a mis padres y a Esti.

Agradecimientos

El desarrollo de esta tesis doctoral ha constituido un largo trayecto lleno de altibajos, vulgarmente conocido en ciclismo como rompepiernas. De no ser por el apoyo de muchas personas que me han acompañado a lo largo de estos años, habría sido imposible pedalear hasta la meta de esta gran vuelta.

Quiero agradecer a Diego López de Ipiña la posibilidad que me ha dado de desarrollar el doctorado dentro del grupo MORElab y las condiciones en las que lo he podido llevar a cabo, ya que pese a participar en otras tareas, se me ha permitido dedicar la mayor parte del tiempo a mi tesis doctoral.

Aprovecho también estas líneas para agradecer a los miembros de MORElab, que tan a gusto me han hecho sentir desde el primer día y me han ayudado en todo lo que he necesitado. Por esos cafés de desconexión diarios. Me acuerdo especialmente de Dieguich, mi compañero de pedales durante estos cuatro años y medio, sin el que los repechos habrían sido mucho más pronunciados. Ha sido un largo camino, pero recorriéndolo junto a un amigo se ha hecho mucho más llevadero. También agradecer a Jonathan esas innumerables charlas de autobús y por esta amistad que seguro me llevo.

Por otra parte, quiero agradecer a mis amigos que siempre me acojan como lo hacen, pese a que con la realización de este trabajo mi tiempo de ocio se haya visto reducido de forma notable.

También me acuerdo en estos momentos de toda la gente del fútbol que tantas horas de desconexión me han ofrecido y con quien tantos buenos momentos he pasado y sigo pasando. Podría enumerar una lista interminable, pero ya sabéis todos quiénes sois.

No me quiero olvidar de mi familia, especialmente de mis padres, que siempre me han dado total libertad para estudiar lo que yo he querido y me han apoyado en mis decisiones, tanto en lo moral como en lo económico. Gracias de todo corazón, ya que sin vuestra educación no sería como soy.

Y por último, mi mayor agradecimiento a Esti, mi gran compañera de equipo en esta vida, que además de servirme de apoyo en todo momento, me ha facilitado el dedicar horas y más horas de nuestro tiempo libre al desarrollo de esta tesis. ¡Mil gracias por estar siempre ahí!

Y seguro que me dejo gente por recordar, porque son muchas las personas que con pequeños detalles han sumado a que haya finalizado con éxito esta gran vuelta. Gracias a todos.

Juan

Resumen

A lo largo de los últimos años, coincidiendo con el *boom* de las nuevas tecnologías de información, el número de dispositivos consumidores de energía ha ido creciendo de forma exponencial. Esto conlleva que el consumo energético haya incrementado siguiendo la misma tendencia, alcanzando en áreas residenciales y comerciales el 40 % de la energía total consumida en todo el mundo. Además, el consumo eléctrico de los equipamientos de oficina supone más del 15 % del total consumido en una oficina. Este aumento implica un incremento de las emisiones de dióxido de carbono, lo que supone un gran desequilibrio en la sostenibilidad del planeta que habitamos.

Por una parte, existe el problema del ineficiente diseño de los dispositivos en cuanto a energía se refiere. Y por otra parte, el uso inadecuado de los mismos, ya sea por dejadez, confort o desconocimiento. Este último problema, se acrecienta cuando se trata de dispositivos de uso colectivo, especialmente los que disponen del modo *Standby*. Algunos ejemplos de este tipo de objetos son las cafeteras, las impresoras, los proyectores o los sistemas de iluminación. En la mayoría de los casos, los usuarios carecen de toda la información necesaria para hacer un uso eficiente de estos objetos de uso colectivo.

Por lo tanto, esta tesis doctoral pretende mejorar la eficiencia energética de este tipo de dispositivos, dotando a los mismos de la capacidad de:

- (a) aprender automáticamente un modelo eficiente de funcionamiento (a través de un nuevo modo *Ecoaware*) y ,
- (b) comunicarse, de forma que puedan compartir entre ellos sus modos de funcionamiento, posibilitando un aprendizaje más rápido — disminución del problema de arranque en frío — (a través de un nuevo modo *Cooperative*).

Para ello se ha desarrollado un experimento de 12 meses de duración con 11 máquinas de café, que es el dispositivo que se ha utilizado como caso de uso. A lo largo de este tiempo, gracias a un módulo de captura de datos instalado en las propias cafeteras, éstas han estado recogiendo datos que luego han permitido calcular el consumo energético de los nuevos modos de funcionamiento propuestos. Estos cálculos demuestran que el uso del modo *Ecoaware* reduce el consumo energético de los dispositivos, mientras que también hacen prever que el uso del modo *Cooperative* sigue la misma línea.

Tabla de contenido

Lista de figuras	VIII
Lista de tablas	XI
Lista de acrónimos	XIII
CAPÍTULO 1 Introducción	1
1.1 Motivación	1
1.2 Hipótesis y objetivos	8
1.3 Metodología de investigación	11
1.4 Organización de la tesis	12
CAPÍTULO 2 Estado del arte	15
2.1 Introducción	15
2.2 Marco para la revisión	17
2.3 Dispositivos sostenibles	17
2.3.1 Análisis y conclusiones	21
2.4 Hardware para Internet de las Cosas	22
2.4.1 Plataformas hardware para Internet de las Cosas	25
2.5 Plataformas software para Internet de las Cosas	34
2.5.1 Arquitecturas para Internet de las Cosas	34
2.5.2 Redes sociales para Internet de las Cosas	46
2.6 Algoritmos de similitud de perfiles	54
2.6.1 Coeficientes de similitud para datos binarios	55
2.6.2 Coeficientes de similitud para valores cuantitativos	59
2.6.3 Análisis y conclusiones	62

2.7	Métodos de aprendizaje automático	63
2.7.1	Clasificador Naïve Bayes	64
2.7.2	Redes neuronales artificiales	64
2.7.3	Series temporales - Modelo ARIMA	67
2.7.4	Análisis y conclusiones	69
CAPÍTULO 3	Un nuevo enfoque inteligente y colaborativo	71
3.1	Introducción	71
3.2	Bases teóricas. De <i>On-Off</i> a <i>Standby</i>	72
3.3	Diseño conceptual	74
3.3.1	Diseño del modo <i>Ecoaware</i>	74
3.3.2	Diseño del modo <i>Cooperative</i>	77
3.4	Cálculo de la predicción de uso	80
3.5	Cálculo de similitud entre perfiles	83
3.5.1	Modelado de los perfiles de los dispositivos	83
3.5.2	Coeficiente de similitud de <i>wGower</i>	84
3.6	Análisis y conclusiones	89
CAPÍTULO 4	Implementación de la solución	91
4.1	Introducción	91
4.2	Requisitos del sistema	91
4.3	Captura y almacenamiento de datos	93
4.3.1	Hardware	93
4.3.2	Almacén de datos	97
4.3.3	Análisis y conclusiones	100
4.4	Plataforma de registro de dispositivos	101
4.4.1	Servicio de registro automático en Twitter	102
4.4.2	Detección de dispositivos similares	103
4.4.3	Mecanismo de compartición de datos entre dispositivos	105
4.4.4	Análisis y conclusiones	107
4.5	Implementación del método de aprendizaje y predicción de uso	107
4.5.1	Identificación del modelo	107
4.5.2	Validación del modelo	110
4.5.3	Análisis y conclusiones	111
CAPÍTULO 5	Evaluación y resultados	113

5.1	Introducción	113
5.2	Metodología de evaluación	114
5.2.1	Análisis y conclusiones	123
5.3	Plataforma hardware	130
5.3.1	Análisis y conclusiones	135
CAPÍTULO 6 Conclusiones y trabajo futuro		137
6.1	Conclusiones	137
6.2	Contribuciones de la tesis	142
6.3	Publicaciones	143
6.3.1	Revistas indexadas en JCR	143
6.3.2	Conferencias internacionales	144
6.3.3	Otras publicaciones relacionadas	145
6.4	Trabajo futuro	145
ANEXOS		147
ANEXO A	Modelado de la información energética en la BBDD	147
ANEXO B	Cálculo de consumo energético	149
ANEXO C	Archivo de configuración de los dispositivos	167
ANEXO D	Registro en Twitter y compartición de datos	169
D.1	Autenticación en Twitter	169
D.2	Registro automático en Twitter	170
D.3	Suscripción a lista de dispositivos	170
D.4	Intercambio de información entre dispositivos	170
ANEXO E	Cálculo de la predicción de uso	173
E.1	Cálculo de ARIMA	173
ANEXO F	Modelos teóricos de las funciones ACF y PACF	175
Bibliografía		179

Lista de figuras

1.1	Crecimiento del número de dispositivos conectados	2
1.2	Histórico del consumo de energía en EEUU (1776-2012). . . .	3
1.3	Crecimiento de las emisiones de CO_2	3
1.4	Energía consumida por una cafetera durante 10 días consecuti- vos desglosada en 3 tipos de consumo.	6
1.5	Apagado de las cafeteras en el trabajo en varios países	7
1.6	Metodología de investigación utilizada.	12
2.1	Evolución del hardware para Internet de las Cosas	23
2.2	Los 3 ejes del hardware para el Internet de las Cosas.	25
2.3	Arduino UNO.	26
2.4	Raspberry Pi B.	29
2.5	Intel Edison e Intel Galileo.	31
2.6	Placa iBoard Pro.	33
2.7	Arquitectura de HYDRA.	36
2.8	Arquitectura modular de SOFIA2.	37
2.9	La consulta y los resultados propagándose a través de la red. .	38
2.10	Esquema de Otsopack.	39
2.11	Arquitectura SENSEI.	40
2.12	Ecosistema propuesto por OpenIoT.	41
2.13	Arquitectura de Xively.	42
2.14	Caso de uso de Carriots.	43
2.15	Diagrama de un sistema con ThingSpeak.	44
2.16	Abstracciones de <i>Objeto</i> y <i>Evento</i>	50
2.17	Arquitectura de Thing Broker.	51
2.18	De la Web hacia la SWoT.	51

2.19	Modelo de neurona artificial.	65
2.20	Modelo de red neuronal artificial.	66
3.1	Arquitectura <i>Ecoaware</i>	74
3.2	Diagrama de secuencia del proceso de captura de datos energéticos.	75
3.3	Diagrama de secuencia del proceso de actualización de la predicción de uso.	76
3.4	Arquitectura <i>Cooperative</i>	77
3.5	Origen de los datos de entrenamiento para una cafetera nueva (A) durante las primeras 4 semanas. (ND = No hay datos) . . .	78
3.6	Diagrama de secuencia del proceso de compartición de datos de uso.	79
3.7	Metodología de Box-Jenkins.	81
4.1	Funcionamiento del módulo de captura y almacenamiento de datos.	93
4.2	Sensor de corriente (a) y (b).	94
4.3	PCB para adaptar el sensor de corriente (a) y (b).	94
4.4	Módulo hardware al completo.	95
4.5	Módulo hardware a medida.	97
4.6	Web de registro de dispositivos.	101
4.7	Registro de un dispositivo en la web y su correspondiente perfil generado en Twitter.	102
4.8	Lista de máquinas de café <i>Ecoaware</i> registradas en Twitter. . .	103
4.9	Serie temporal de cafés preparados durante 20 días hábiles en uno de los espacios bajo estudio.	108
4.10	Serie diferenciada y sus ACF y PACF para identificar el modelo	109
4.11	Los residuales y sus ACF y PACF asociados.	110
4.12	Predicción de cafés preparados durante 5 días.	111
5.1	Una de las cafeteras del experimento tras la intervención. . .	115
5.2	Período de captura de datos de las cafeteras (12 meses). . .	116
5.3	Cálculo del consumo en modo <i>On-Off</i>	118
5.4	Cálculo del consumo en modo <i>Auto-power Down</i>	119
5.5	Cálculo del consumo en modo <i>Ecoaware</i>	120

5.6	Cálculo del consumo en modo <i>Cooperative</i>	121
5.7	Consumo energético de las 4 cafeteras de la categoría <i>Ecoaware</i> en cada uno de los modos.	127
5.8	Consumo energético de 4 cafeteras de la categoría <i>On-Off</i> en cada uno de los modos.	127
5.9	Consumo energético de las 3 cafeteras restantes de la categoría <i>On-Off</i> en cada uno de los modos.	128
5.10	Equivalencias de emisiones de CO_2 (1).	129
5.11	Equivalencias de emisiones de CO_2 (2).	130
5.12	Fuente programable Agilent Technologies N6705B.	131
5.13	Consumo del hardware en la fase de arranque.	132
5.14	Consumo del hardware en la fase de captura de datos.	133
5.15	Consumo del hardware en la fase de envío de datos.	134
F.1	Patrones teóricos ACF y PACF para series temporales (1). . .	175
F.2	Patrones teóricos ACF y PACF para series temporales (2). . .	176
F.3	Patrones teóricos ACF y PACF para series temporales (3). . .	177

Lista de tablas

1.1	Funcionamiento de una máquina de café en cada uno de los diferentes modos de operación.	5
2.1	Productos comerciales.	20
2.2	Características de Arduino UNO y Arduino MEGA 2560. . . .	27
2.3	Comparativa de las placas hardware de Internet de las Cosas. .	32
2.4	Comparativa de plataformas software de Internet de las Cosas.	46
2.5	Dos muestras descritas por variables multiestado.	60
3.1	Variables del perfil de los dispositivos clasificadas por tipo de variable.	85
3.2	Variables del perfil de los dispositivos reclasificadas por tipo de variable.	86
3.3	Listado de espacios de trabajo.	88
3.4	Grados de similitud entre los diferentes tipos de espacio. . . .	88
3.5	Pesos asignados a las variables del perfil.	89
4.1	Datos de un documento generado debido a la realización de un café.	99
4.2	Datos de un documento generado debido al encendido de una cafetera.	99
4.3	Datos de un documento generado debido a un pico de consumo en modo <i>Standby</i>	99
4.4	Características de los espacios donde se ubican las cafeteras ID1 e ID3.	104
4.5	Cálculo del coeficiente de <i>wGower</i> entre todos los dispositivos.	105

5.1	Características de los espacios donde se ubican los dispositivos.	114
5.2	Cálculo del coeficiente de wGower entre todos los dispositivos.	123
5.3	Evolución del consumo energético de las cafeteras bajo estudio.	125
5.4	Consumo energético en kWh en cada espacio en modos <i>Eco-aware</i> y <i>Cooperative</i> (20 días, excepto ID12 e ID14).	126
5.5	Comparativa entre cafeteras de la proporción de consumo energético del hardware respecto al ahorro energético producido por los modos propuestos.	135
5.6	Comparativa entre cafeteras de la proporción de consumo energético del nuevo hardware respecto al ahorro energético producido por los modos propuestos.	136
6.1	Estado de cumplimiento de los objetivos definidos en la sección 1.2 y en qué artículo o sección se cumplen.	140

Lista de acrónimos

ACF	Autocorrelation Function. 82, 108-110, 175-177
API	Application Programming Interface. 35, 37, 42-45
ARIMA	Autoregressive Integrated Moving Average. 67-70, 80, 108, 109, 111, 173
BBDD	Base de datos. 118-121, 147, 148
CSV	Comma-Separated Values. 42, 46
DSL	Domain Specific Language. 38
EM	Embodied Microblogging. 48
HCI	Human Computer Interaction. 20, 21
HTTP	Hypertext Transfer Protocol. 43, 44, 48
IoT	Internet of Things. 16, 23, 34, 41
JCR	Journal Citation Reports. 143
JSON	JavaScript Object Notation. 42, 43, 45, 46, 98
Li-Fi	Light Fidelity. 25

M2M	Machine to Machine. 17, 43
MEM	Microelectromechanical Systems. 23
MQTT	MQ Telemetry Transport. 37
NFC	Near Field Communication. 23
NTP	Network Time Protocol. 95
OWL	Web Ontology Language. 35, 46
PACF	Partial Autocorrelation Function. 82, 108-110, 175-177
PCB	Printed Circuit Board. 94
PNG	Portable Network Graphics. 45, 46
RDF	Resource Description Framework. 40, 46
REST	Representational State Transfer. 37, 42, 43, 50
RF	Radio Frequency. 26, 35
RFID	Radio Frequency Identification. 16, 22, 23, 35, 49
RNA	Red Neuronal Artificial. 64, 69
SDK	Software Development Kit. 37, 38, 43
SIoT	Social Internet of Things. 49
SoC	System on a Chip. 23
SWoT	Social Web of Things. 51
XML	Extensible Markup Language. 40, 42, 43, 45, 46

1

Introducción

Cuando veo a un adulto en una bicicleta, no me desespero por el futuro de la humanidad
—H. G. WELLS.

ESTE CAPÍTULO expone una visión general de la investigación llevada a cabo en esta tesis doctoral. Se presentan las motivaciones que han impulsado esta investigación, así como las hipótesis y objetivos establecidos. Además se describe la metodología de investigación utilizada y se finaliza el capítulo explicando la estructura de esta tesis doctoral.

1.1 Motivación

A finales del siglo XIX dio comienzo la generación industrial de electricidad, período en el que la iluminación eléctrica se extendió por calles y casas. La rápida expansión de la tecnología eléctrica la convirtió en la columna vertebral de la sociedad industrial moderna (?). Un gran número de aplicaciones vieron la luz gracias a este tipo de energía, lo que la convirtió en uno de los pilares de la Segunda Revolución Industrial (?).

A primeros del siglo XX, aprovechando la energía eléctrica mediante baterías y líneas de corriente, se crearon los primeros electrodomésticos. El aire acondicionado en 1902, la aspiradora eléctrica en 1908 o la cafetera eléctrica en la década de 1920 son algunos ejemplos de ello. Estos novedosos objetos no eran más que un presagio del mundo del electrodoméstico en el que vivimos.

Con la aparición en los últimos años de nuevas tecnologías de información, el número de dispositivos consumidores de energía está creciendo de forma exponencial (ver figura 1.1). Muchos de estos dispositivos son denominados inteligentes por su capacidad de conexión a Internet y por sus nuevas formas de identificación y comunicación. Como se puede observar en la figura 1.2, se ha producido un aumento notable de la energía consumida en los últimos años. Otro ejemplo, entre varios en esta línea, es un informe (?) publicado por el gobierno de Japón, donde se publica que la energía consumida en áreas residenciales y comerciales se ha incrementado en un 30 % desde 1990. En Estados Unidos el consumo de energía en estas mismas áreas alcanzó en 2015 el 40 % de la energía total consumida en todo el país¹, al igual que ocurre a nivel mundial².

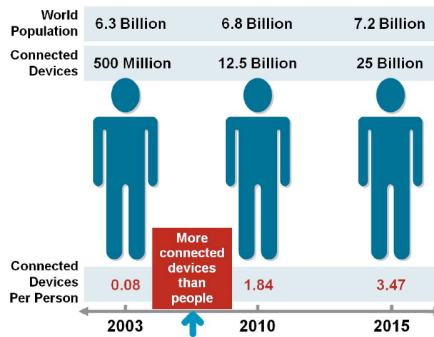


Figura 1.1: Crecimiento del número de dispositivos conectados³.

¹<http://www.eia.gov/tools/faqs/faq.cfm?id=86&t=1>

²<http://www.unep.org/sbci/AboutSBCI/Background.asp>

³<http://tiny.cc/ciscoineternet>

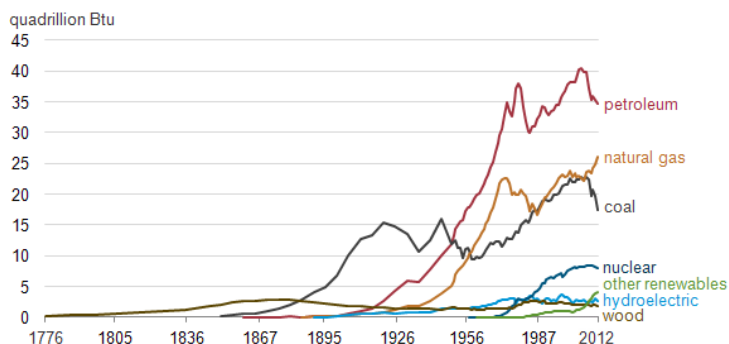


Figura 1.2: Histórico del consumo de energía en EEUU (1776-2012).

Según estudios realizados (?), se ha observado que el consumo eléctrico de los equipamientos de oficina suponen más del 15 % del consumo total de una oficina. Estos mismos estudios revelan que la creciente invasión de los ordenadores, los dispositivos electrónicos, los electrodomésticos y el equipamiento de oficina es uno de los factores que influyen considerablemente en el incremento de las emisiones de dióxido de carbono (ver figura 1.3), lo que supone un mayor desequilibrio en la sostenibilidad del planeta en el que vivimos.

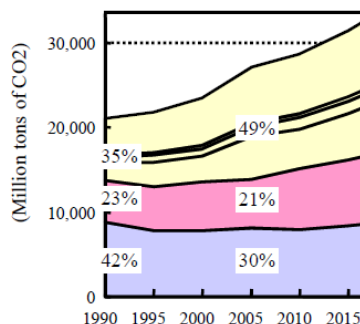


Figura 1.3: Crecimiento de las emisiones de CO_2 ⁴.

⁴<http://www.rite.or.jp/en/>

Esta situación llevó en 2008 a la Unión Europea a regular la situación de este tipo de dispositivos, creando el reglamento 1275/2008/EC⁵ que afecta a todos los países europeos. El objetivo de éste es resolver los requisitos de diseño ecológico aplicables al consumo de energía eléctrica de los electrodomésticos, equipos electrónicos y de oficina que no se encuentran en modo de funcionamiento “normal” (modo *On*). Esta norma establece que desde 2013 todos los aparatos deben permitir al menos uno de los dos modos de espera posibles:

- Modo *Standby*: aquél en que el equipo se halla conectado a la red eléctrica y depende de la energía procedente de dicha red para funcionar según los fines previstos y ejecuta únicamente las siguientes funciones, que se pueden prolongar por un tiempo indefinido:
 - (a) Función de reactivación (potencia máxima permitida de 0.5 W).
 - (b) Visualización de información o del estado (potencia máxima permitida de 1 W).
- Modo *Off*: aquél en que el equipo se halla conectado a la red eléctrica, si bien no está en funcionamiento o permanece totalmente inactivo (potencia máxima permitida de 1W).

Además, esta regulación decreta que cuando un equipo no esté ejecutando su actividad principal ofrecerá una función para conmutar automáticamente el equipo, tras un período breve de tiempo, al modo *Off* o al modo *Standby*. De aquí en adelante, cuando los dispositivos tengan activada esta función, se considerará que están funcionando en modo *Auto-power Down*.

En la tabla 1.1 se muestra, mediante diagramas de flujo y a modo de ejemplo, el comportamiento de una cafetera en cada uno de los modos descritos.

⁵<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2008:339:0045:0052:EN:PDF>

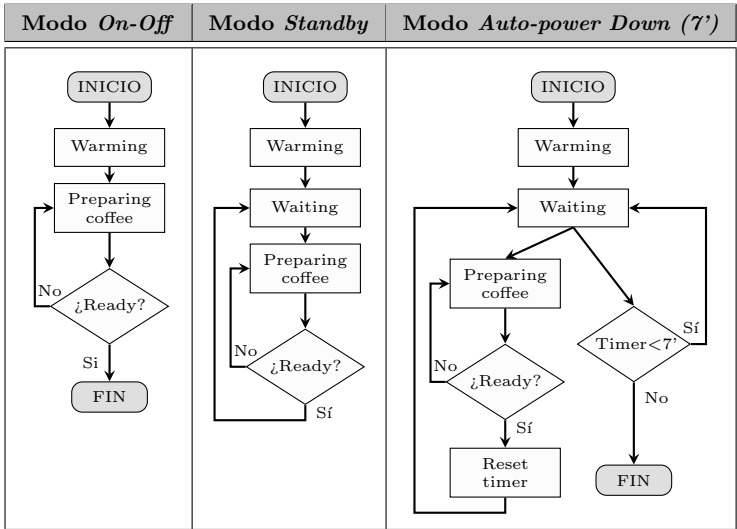


Tabla 1.1: Funcionamiento de una máquina de café en cada uno de los diferentes modos de operación.

Pero el problema a lo largo de estos años no se ha cimentado únicamente en el ineficiente diseño de los dispositivos, sino también en el uso inadecuado que se hace de los mismos. Uno muy frecuente consiste en dejar los dispositivos encendidos consumiendo energía constantemente cuando nadie los está utilizando. En un trabajo de exploración inicial (?) se plasmaba de forma gráfica un ejemplo de este tipo de comportamientos que tanto se repiten. Se registró el consumo energético de una cafetera ubicada en un laboratorio de investigación durante 30 días consecutivos. La figura 1.4 presenta una franja de tiempo de 10 días, donde se muestra claramente que los días correspondientes al fin de semana (cuarto y quinto día) la cafetera ha estado encendida sin realizar su función principal, que es la de preparar cafés.

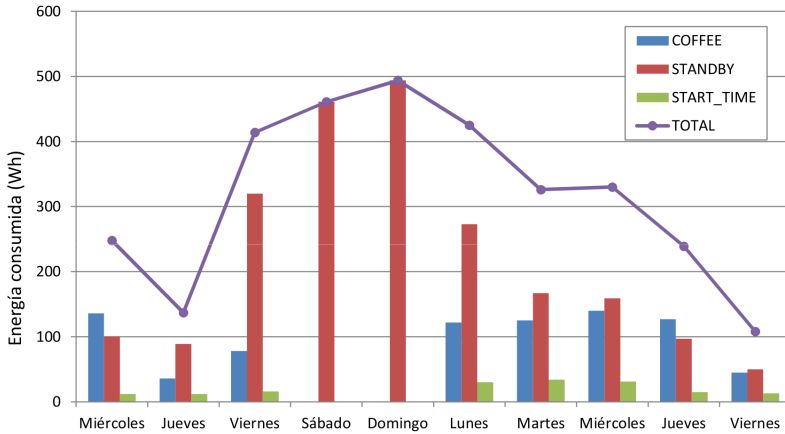


Figura 1.4: Energía consumida por una cafetera durante 10 días consecutivos desglosada en 3 tipos de consumo.

Yang et al. (?) identificaron la falta de atención de los usuarios y el confort como dos de las principales causas que motivan este tipo de comportamientos. Asimismo, debido en gran parte a la naturaleza intangible de la electricidad, las personas habitualmente se mantienen al margen de comprender el consumo energético de los dispositivos. Además, tal y como señalan Schwartz et al. (?), la conciencia sobre el consumo energético de objetos de uso colectivo es insignificante en comparación con la de los dispositivos personales. Por ejemplo, no es habitual ver trabajadores por cuenta ajena preocupándose de lo que consumen los objetos que están utilizando o encontrar una mayoría de ciudadanos concienciados con un uso energéticamente eficiente de los dispositivos de uso colectivo que tienen a su disposición. ¿O acaso no es frecuente encontrar encendidas las luces de los aseos de un lugar de trabajo o una universidad?

De modo que aunque se considera que la combinación de los modos de funcionamiento regulados por la Unión Europea en el reglamento 1275/2008/EC podría dar lugar a un funcionamiento energéticamente más eficiente, los usuarios de estos dispositivos, incluso los más sensibilizados con el medio ambiente, en la mayoría de los casos carecen de

toda la información necesaria para hacer un uso eficiente de los mismos.

De hecho, se puede afirmar que este comportamiento confuso es generalizado a nivel mundial. La figura 1.5 muestra comportamientos completamente diferentes en cuanto al apagado de las cafeteras en entornos de trabajo. Esto puede ser una evidencia de la ausencia de reglas para utilizar eficientemente las máquinas de café.

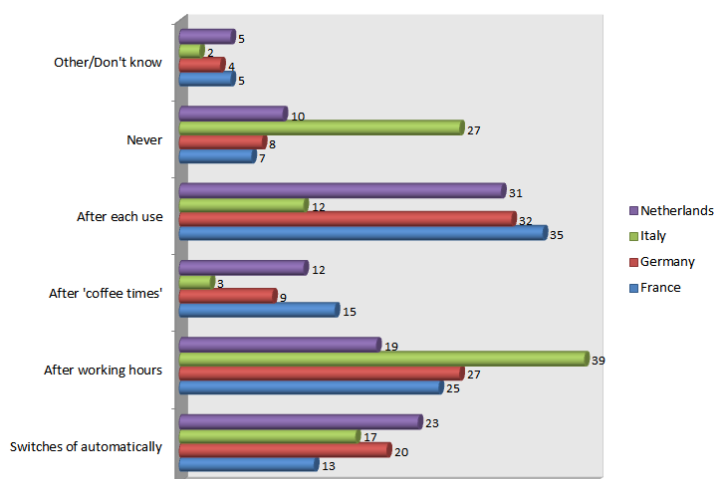


Figura 1.5: Apagado de las cafeteras en el trabajo en varios países (?).

Por lo tanto, el objetivo general de esta tesis es mejorar la eficiencia energética de los dispositivos provistos del modo *Standby*, concretamente de los ubicados en espacios comunes, en los que un uso energéticamente eficiente se antoja difícil de conseguir. Algunos ejemplos de este tipo de objetos son las cafeteras, las impresoras, los proyectores o los sistemas de iluminación. Se pretende lograrlo dotando a los mismos de la capacidad de:

- (a) aprender automáticamente un modelo eficiente de funcionamiento (a través de un nuevo modo *Ecoaware*) y ,
- (b) comunicarse, de forma que puedan compartir entre ellos sus modos de funcionamiento, posibilitando un aprendizaje más rápido — disminución del problema de arranque en frío⁶ — (a través de un nuevo modo *Cooperative*).

1.2 Hipótesis y objetivos

Acorde a lo ya expuesto, se desarrollan las hipótesis de la investigación. La primera de ellas se expresa en los siguientes términos:

H_A: *Una conmutación automática e inteligente del modo de funcionamiento de los dispositivos eléctricos, se traduce en una reducción significativa⁷ de su consumo energético.*

Considerando veraz la hipótesis H_A , se presenta la siguiente hipótesis:

H_B: *Una colaboración entre dispositivos eléctricos basada en la compartición de sus patrones de uso, se traduce en una aceleración del proceso de ahorro energético, lo que conlleva una reducción significativa de su consumo energético durante sus primeras semanas de uso.*

⁶Denominado en inglés *cold-start problem*, es un problema potencial de los equipos basados en sistemas de información, que implican un grado de modelado de datos automatizado. En concreto, se refiere a la cuestión donde el sistema no puede extraer inferencias para los usuarios o temas sobre los que aún no ha reunido suficiente información.

⁷La directiva relativa a la eficiencia energética (2012/27/UE) establece entre otros criterios la elaboración de medidas que garanticen un ahorro anual de energía del 1.5 % para los usuarios finales hasta 2020. Por lo tanto, se considera en esta tesis doctoral que una reducción significativa del consumo energético es un ahorro del 1.5 % de la energía consumida.

Como ya se ha explicado con anterioridad, la hipótesis H_B se basa en la mayor eficiencia energética de los dispositivos que tienen la capacidad de adaptar dinámicamente su modo de funcionamiento a su patrón de uso. Por lo tanto, si y sólo si se demuestra que la hipótesis H_A es verdadera, se podrá cumplir la hipótesis H_B .

Para conformar la visión planteada en la investigación y teniendo en cuenta las hipótesis de partida, se ha definido una serie de objetivos, tanto para la creación del modo de funcionamiento *Ecoaware*, como del modo *Cooperative*. Se enumeran primero los objetivos establecidos para crear satisfactoriamente el nuevo modo de funcionamiento *Ecoaware*, ya que su consecución es imprescindible para poder validar la hipótesis H_B :

- **Objetivo A.1:** definición del proceso que se va a llevar a cabo para analizar el funcionamiento de un electrodoméstico a nivel energético. Se debe medir el consumo de energía en el período de tiempo inmediato a alimentar con corriente el dispositivo, cuando se encuentra funcionando en modo *Standby* y cuando lo está haciendo en modo *On* desarrollando la función principal para la que ha sido diseñado.
- **Objetivo A.2:** diseño y desarrollo de un sistema, hardware y software, que debe medir y almacenar todas las acciones de consumo energético de un dispositivo. Este objetivo conlleva la definición del modelo de datos que permite describir cuantitativamente y cualitativamente la energía consumida por los dispositivos a lo largo del tiempo. A su vez, se ha de determinar el sistema de almacenamiento a utilizar.
- **Objetivo A.3:** definición y aplicación del método de aprendizaje automático que permite obtener una mayor precisión en la predicción de uso del dispositivo. Es necesario analizar diferentes métodos para discernir cuál es el que mejor se ajusta a la problemática.
- **Objetivo A.4:** validación de los módulos hardware y software desarrollados para comprobar que su funcionamiento se adecua a lo que se ha propuesto.

- **Objetivo A.5:** evaluación de la solución integral que propone cambiar el modo de funcionamiento de los dispositivos en función de su patrón de uso. Se mide cuantitativamente el ahorro energético de los dispositivos funcionando en modo *Ecoaware* y se compara respecto a los modos de funcionamiento existentes (*On-Off*, *Standby*, *Auto-power Down*).

A continuación, se exponen los objetivos definidos para crear el modo de funcionamiento *Cooperative* y evaluar su impacto:

- **Objetivo B.1:** definición de la arquitectura que debe englobar los dispositivos y los flujos de intercambio de datos entre los mismos.
- **Objetivo B.2:** determinación del algoritmo de similitud entre dispositivos. Para ello es necesario realizar un estudio del estado del arte de los algoritmos ya existentes y evaluar si alguno de ellos se ajusta a la problemática o es necesario definir uno nuevo que se adecue a la misma. Una parte esencial es el modelado del perfil de los dispositivos, ya que el grado de similitud entre dos objetos depende directamente de su perfil.
- **Objetivo B.3:** desarrollo del algoritmo de similitud definido anteriormente y su validación para evaluar su funcionamiento de acuerdo a los requisitos que se establezcan.
- **Objetivo B.4:** definir el modelo de comunicación entre los dispositivos. El primer paso debe ser definir qué información se quiere intercambiar y qué flujo de información se va a producir. Una vez realizada esta tarea, se ha de valorar tanto la idoneidad de los diferentes *middleware* existentes para atender las necesidades que se plantean, como la posibilidad de crear una solución adaptada.
- **Objetivo B.5:** desarrollo de la infraestructura de comunicación determinada anteriormente y su validación para determinar que responde a los requisitos de la solución demandada.
- **Objetivo B.6:** evaluación de la solución colaborativa expuesta en esta tesis doctoral para comprobar las ventajas y limitaciones de la

propuesta planteada. Se mide cuantitativamente el ahorro energético de los dispositivos funcionando en modo *Cooperative* y se compara respecto al modo de funcionamiento *Ecoaware*.

1.3 Metodología de investigación

En busca de alcanzar los objetivos plasmados en la sección 1.2 se ha definido una estrategia de investigación (ver figura 1.6). Ésta engloba las siguientes actividades:

1. **Revisión bibliográfica:** llevar a cabo una revisión sistemática de los estudios e investigaciones más relevantes, atendiendo a diversos marcos de revisión, relacionados con las áreas abordadas en la tesis doctoral.
2. **Definición de hipótesis y objetivos:** la revisión bibliográfica ha dado lugar a una línea de investigación abierta relacionada con la posibilidad de reducir el consumo energético de los dispositivos eléctricos capaces de funcionar en diferentes modos (*On/Off/Standby*). Por lo tanto, en esta tarea se definen las hipótesis de partida y se identifican los objetivos, tanto generales como específicos, perseguidos durante el desarrollo de la tesis doctoral.
3. **Diseño y desarrollo de la propuesta:** esta tarea consiste en el diseño y el desarrollo de la arquitectura propuesta, así como la de todos los elementos que la componen.
4. **Experimentación y evaluación:** evaluar las soluciones propuestas, tanto para los dispositivos que individualmente aprenden a funcionar de forma dinámica como para la solución global en la que los dispositivos se comunican entre sí. La experimentación ha permitido comprobar las aportaciones, ventajas y limitaciones de las propuestas realizadas en la investigación.
5. **Validación:** Difusión en la comunidad científica de los resultados obtenidos para la consideración y aprobación de la tesis.

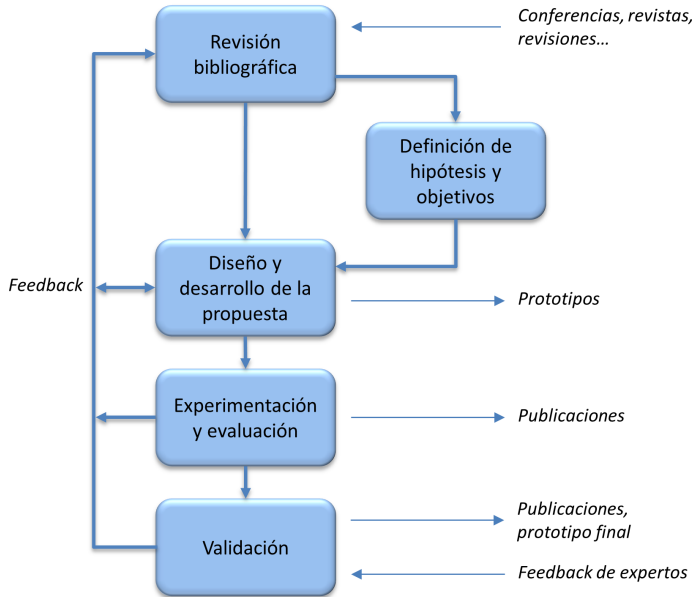


Figura 1.6: Metodología de investigación utilizada.

1.4 Organización de la tesis

Esta tesis doctoral está estructurada en seis capítulos, incluyendo el actual de introducción.

El capítulo 2 contiene una revisión bibliográfica de diferentes áreas relacionadas con esta tesis doctoral. Se revisan los diferentes dispositivos existentes diseñados para el ahorro de energía; las principales plataformas hardware en el ámbito de Internet de las Cosas; arquitecturas para Internet de las Cosas y redes sociales dentro del mismo contexto; algoritmos de similitud entre perfiles; y métodos de aprendizaje automático y predicción.

El capítulo 3 introduce de una manera formal cuál es el problema a resolver. Se presenta también la solución que se propone para hacer frente a la problemática descrita, describiendo la arquitectura basada en redes sociales que se ha adoptado.

En el capítulo 4 se describe cómo se ha implementado la solución adoptada. Se detalla el diseño y desarrollo de todos los elementos de la arquitectura, desde los propios dispositivos hasta la infraestructura que las comunica, pasando por los adaptadores hardware que se han incorporado a las cafeteras que forman parte del experimento, la plataforma web para registrar dispositivos y los módulos de cálculo, tanto de similitud entre los mismos, como de predicción de uso para aumentar su eficiencia energética.

El capítulo 5 contiene la evaluación y los resultados de la solución que se ha propuesto, permitiendo así conocer cuál es el alcance de las aportaciones que se han identificado.

Por último, el capítulo 6 recoge las conclusiones extraídas de la investigación realizada. Además, se enumeran las contribuciones de esta tesis doctoral y las publicaciones que la respaldan. Para finalizar, se presentan las posibles líneas futuras de investigación que permitan avanzar sobre el trabajo realizado y aquí expuesto.

2

Estado del arte

Yendo en bicicleta es como mejor se conocen los contornos de un país, pues uno suda ascendiendo a los montes y se desliza en la bajada.

—ERNEST HEMINGWAY.

2.1 Introducción

EN ESTE CAPÍTULO se recoge el estado del arte más relevante relacionado con las áreas tecnológicas en las que se basa esta tesis doctoral. En un primer lugar, se muestra una revisión de los dispositivos existentes para ahorrar energía en el hogar, oficinas o comercios. A continuación se presentan las principales plataformas hardware en el ámbito de Internet de las Cosas, así como las tecnologías de comunicación de las mismas y su consumo energético. Dada la naturaleza cooperativa de los objetos *Ecoaware*, se ha llevado a cabo una revisión bibliográfica de las arquitecturas existentes para el Internet de las Cosas. Además, en este mismo contexto se han revisado las redes sociales, haciendo hincapié en las que relacionan objetos. También se exponen los procedimientos de clasificación basados en similitud de perfiles

considerados más relevantes. Y por último, se hace una breve revisión bibliográfica de los métodos predictivos con el objetivo de discernir cuál obtiene una mejor predicción de uso de los dispositivos.

Antes de continuar con las diferentes secciones de este capítulo se va a definir qué es el Internet de las Cosas (Internet of Things – IoT) para poner en contexto al lector. Es un concepto propuesto por Kevin Ashton en el Auto-ID Center¹ del MIT² en 1999, donde se realizaban investigaciones sobre identificación por radiofrecuencia (RFID) y tecnología de sensores. Se refiere a la interconexión digital de objetos cotidianos a través de Internet.

La UIT³ (International Telecommunication Union – ITU) lo define como *“la infraestructura mundial de la sociedad de la información, que ofrece servicios avanzados interconectando cosas (físicas y virtuales) utilizando las tecnologías de la información y la comunicación compatibles existentes y en evolución”* (?).

A lo largo de los últimos años se ha ido generando cada vez una mayor atención en la industria para interactuar con diferentes tipos de dispositivos. El IoT puede tener una gran influencia, tanto en la industria como en la sociedad, mediante la integración de los objetos físicos en los sistemas de información (?). Desde la perspectiva de los usuarios a nivel personal, el IoT puede tener repercusión en campos como las casas u oficinas inteligentes, *e-health* o *assisted living*. Desde el punto de vista empresarial, el impacto lo tiene en campos tales como la automatización y la fabricación industrial, la logística, la gestión de procesos empresariales o los sistemas inteligentes de transporte de personas y mercancías (?). El Internet de las Cosas abarca toda la infraestructura, incluyendo software, hardware y servicios.

Según la empresa Gartner⁴, en 2020 habrá en el mundo aproximadamente 26 mil millones de dispositivos con un sistema de adaptación al

¹<http://autoidlabs.org/>

²<http://web.mit.edu/>

³<http://www.itu.int>

⁴<http://www.gartner.com/newsroom/id/2636073>

Internet de las Cosas. Abi Research⁵, por otro lado, asegura que para el mismo año existirán 30 mil millones de dispositivos inalámbricos conectados al Internet.

Actualmente, el término Internet de las Cosas se usa con una denominación de conexión avanzada de dispositivos, sistemas y servicios que va más allá del tradicional M2M (máquina a máquina) y cubre una amplia variedad de protocolos, dominios y aplicaciones (?).

2.2 Marco para la revisión

A raíz de la hipótesis inicial (H_A) de esta tesis doctoral, se plantea la siguiente pregunta de investigación:

¿Es posible modelar, diseñar y desarrollar una arquitectura que permita a objetos cotidianos ubicados en entornos con características similares comunicarse para que puedan rápidamente adaptar su funcionamiento a uno energéticamente más eficiente?

En torno a esta pregunta se ha realizado una revisión bibliográfica de los productos y trabajos ya existentes, recurriendo principalmente a las siguientes bases de datos:

- IEEE Xplore (<http://ieeexplore.ieee.org/Xplore/home.jsp>)
- ACM (<http://portal.acm.org>)
- Springer (<http://link.springer.com/>)
- Google Scholar (<http://scholar.google.es>)
- ScienceDirect (<http://www.sciencedirect.com/>)

Al final de cada sección se presenta el análisis y las conclusiones obtenidas en cada una de las áreas.

⁵<http://www.abiresearch.com/press/more%2Dthan%2D30%2Dbillion%2Ddevices%2Dwill%2Dwirelessly%2Dconnect>

2.3 Dispositivos sostenibles

Hoy en día, el número de dispositivos consumidores de energía crece exponencialmente (?) en la mayoría de los ámbitos en los que se mueve la humanidad, como el hogar, el trabajo o los espacios públicos. Este aumento de energía ha conllevado movimientos en beneficio del ahorro de energía. Por una parte, las compañías han visto como se ha abierto un nicho de mercado y han diseñado y desarrollado productos dirigidos a colaborar en la compleja tarea del ahorro de energía. A continuación, en la tabla 2.1 se describen algunos de estos productos:

Producto	Descripción
 Wattson Solar⁶	Wattson se trata de un monitor y gestor energético para el hogar. A través de un pedagógico sistema de luces de colores ayuda a los miembros del hogar a conocer y corregir su consumo eléctrico doméstico. Este dispositivo, además de monitorizar los consumos eléctricos y la producción de la instalación solar, permite conocer los excedentes producidos en cada instante, por lo que el usuario que recibe esa información puede aprovechar el sobrante de energía para cargar un teléfono, calentar agua o lo que considere oportuno.
 Energy Orb⁷	Energy Orb es una bola de cristal con conexión de red inalámbrica que proporciona datos en tiempo real sobre el consumo de energía, permitiendo a los clientes modificar su consumo de energía cuando lo consideren oportuno. Este dispositivo va cambiando desde el color verde hasta el color rojo en función de una demanda y precio de la energía baja o alta.

⁶http://www.biggreentechnology.com/images/Wattson_Solar_Plus_web.pdf

⁷<http://www.ambientdevices.com/>

Producto	Descripción
 Energy Joule ⁸	Este dispositivo, del mismo fabricante que el anterior, recoge datos del termostato traduciéndolos a tendencias de consumo y tarifas energéticas. Está provisto de conexión Zigbee, para no tener la necesidad obligatoria de estar conectado a una red Wi-Fi.
 Wattio ⁹	Wattio es un kit de ahorro energético con posibilidades de control remoto. Una vez instalado, permite conocer en tiempo real el consumo del hogar a través de la pantalla dedicada para ello (GATE), la plataforma web o el teléfono móvil. Además se puede consultar los históricos de forma mensual, diaria, etc.
 Nest ¹⁰	El termostato Nest se presenta como el dispositivo que ofrece un control responsable y eficiente del consumo energético de los sistemas de aire acondicionado de los hogares. Se trata de un termostato inteligente 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 esta forma, a medida que pasa el tiempo, es capaz de funcionar de forma autónoma, sin apenas interactuar con él para que la temperatura del hogar sea óptima. Además, mediante el uso de diferentes tecnologías Nest es capaz de guiar al usuario a temperaturas más eficientes y utiliza sensores que detectan la no presencia de personas en el hogar, permitiendo así bajar la temperatura y seguir ahorrando energía.

⁸<http://www.ambientdevices.com/>

⁹<http://www.wattio.com/>

¹⁰<https://nest.com/thermostat/meet-nest-thermostat/>

Producto	Descripción
 Momit Thermostat¹¹	Se trata de un termostato inteligente que realiza tareas de almacenamiento de datos de consumo y utiliza un sensor para saber cuándo hay alguien en el hogar y de este modo reducir el uso de energía, llegando incluso a apagar la calefacción. Además, indica de forma visual cuándo se están produciendo ahorros.

Tabla 2.1: Productos comerciales.

La mayoría de los productos comerciales que se han desarrollado en esta línea del ahorro energético han sido diseñados para mostrar a las personas la energía consumida en tiempo real. A partir de esa información, el usuario se encuentra en una mejor disposición de tomar decisiones para reducir el consumo energético. Entre los productos presentados en la tabla 2.1 se encuentran los termostatos inteligentes Nest y Momit Smart Thermostat. A diferencia de los anteriores, estos dispositivos tienen la capacidad de adaptar el funcionamiento de la calefacción en función de los hábitos de los miembros del hogar el primero y en función de la ocupación del hogar el segundo. En tiempo real y en función de determinadas variables adecuan la actividad de un objeto, en este caso la calefacción, logrando un mayor confort y un ahorro energético más que evidente, más de un 20 % de energía en el caso de Nest y hasta un 30 % en el caso de Momit Smart Thermostat.

Por otra parte, la comunidad científica ha visto cómo se han abierto múltiples líneas de investigación en busca de la reducción del consumo energético. En la misma dirección de los termostatos inteligentes que se han presentado previamente está el TherML (?). Éste se trata de un algoritmo predictivo de ocupación que utiliza el GPS de los teléfonos móviles de los usuarios para controlar automáticamente la temperatura del hogar. Una línea de investigación muy extendida con el objetivo de disminuir el consumo de energía es la interacción humano-computadora (HCI), que en esta área energética promueve principalmente el cambio

¹¹<http://www.momit.com/smartthermostat.html>

de comportamiento y la concienciación de los humanos. Hay múltiples trabajos que se han desarrollado en esta línea (???) y comercialmente ya se han presentado algunos ejemplos en la tabla 2.1.

Sin embargo, alejándose del ámbito del HCI y más focalizado en la automática y los objetos cotidianos, las soluciones que se proponen son principalmente estáticas: modo *Standby* o modo *Auto-power Down*. Es decir, se han incorporado a los objetos cotidianos nuevos modos de funcionamiento energéticamente más eficientes que independientemente de cuánto, cuándo y cómo sean utilizados, se comportan de la misma forma: *a)* tras su funcionamiento permanecen en un modo de bajo consumo hasta su siguiente uso (modo *Standby*); o *b)* tras su funcionamiento permanece en modo *Standby* y cuando ha discurrido un período de tiempo determinado se apaga completamente (modo *Auto-power Down*).

2.3.1 Análisis y conclusiones

Como ya ha sido demostrado (?), los modos no siempre ofrecen los resultados deseados y en función de ciertas variables el presunto ahorro energético puede convertirse en un consumo mayor. Habitualmente, cuantos más usuarios tenga un dispositivo, mayor dificultad tiene el conocer cuál es el modo de funcionamiento adecuado para cada momento. Por lo que dentro de un ámbito personal o en un hogar con un número reducido de personas es más viable hacer un uso energéticamente eficiente del dispositivo. En un espacio mayor, como pueden ser oficinas, tiendas o áreas públicas, esta tarea adquiere una mayor dificultad. Esto hace palpable la necesidad de añadir un modo de funcionamiento dinámico que adapte la forma de operar del dispositivo al modo más eficiente. En una línea similar se encuentra *Google Nest*, el termostato inteligente ya presentado previamente en la tabla 2.1. A diferencia de los termostatos convencionales que permiten la programación por horas de toda la semana, este dispositivo ejecuta un algoritmo de predicción y adapta el funcionamiento de la calefacción a sus resultados.

A lo largo de la revisión bibliográfica se han categorizado los objetos en tres grupos:

- Los nuevos objetos que se han introducido en el ecosistema que esencialmente aportan funcionalidades de información del consumo de energía, visualización y concienciación energética.
- Los objetos cotidianos que disponen de alguna funcionalidad estática de ahorro de energía.
- Los objetos cotidianos que disponen de alguna funcionalidad dinámica de ahorro de energía y funcionan de forma autónoma.

En esta revisión se han echado en falta objetos cotidianos que compartan entre ellos información útil para reducir el consumo de energía. Por lo tanto, en esta tesis doctoral se propone añadir una nueva categoría. En ella quedarían englobados los objetos cotidianos aumentados con funciones colaborativas, de modo que así sean capaces de compartir información con otros dispositivos con el objetivo de que éstos reduzcan su consumo de energía.

2.4 Hardware para Internet de las Cosas

A lo largo de los últimos años han sido numerosas las empresas y centros que, tal y como muestra la figura 2.1, han centrado su innovación hardware en etiquetas RFID y sensores. Hoy en día la evolución del hardware está marcada por su miniaturización, la nanotecnología y el bajo consumo energético del mismo.

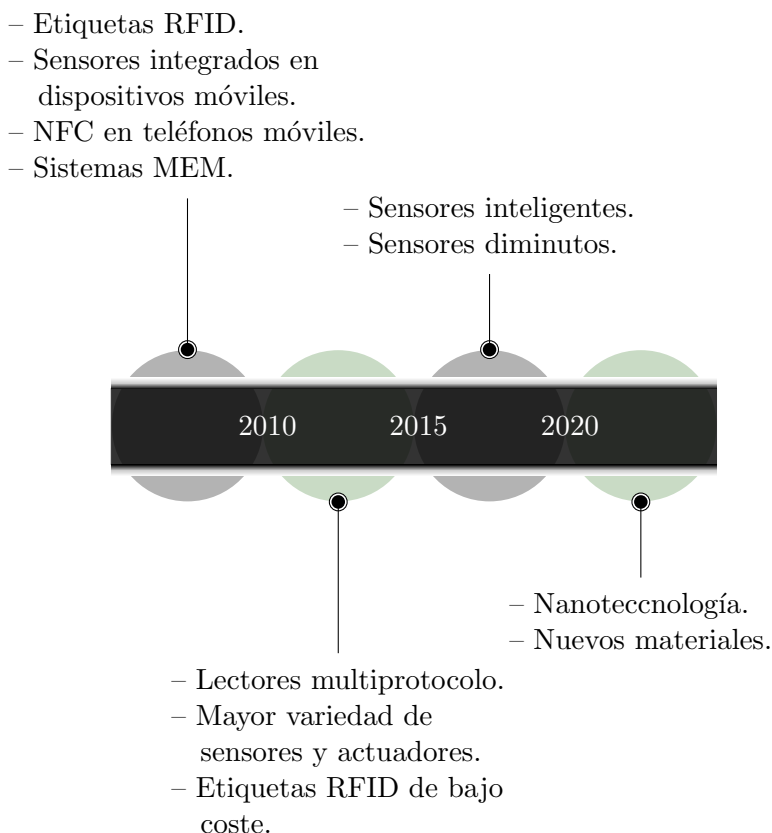


Figura 2.1: Evolución del hardware para Internet de las Cosas (?).

Esta evolución del hardware hacia dispositivos pequeños y de bajo consumo posibilita el progreso de Internet de las Cosas. Ejemplo de ello son los procesadores con formato SoC¹² que cumplen con las dos características mencionadas. ARM ofrece un amplio catálogo (Cortex-A, Cortex-R, Cortex-M) de SoC para dispositivos de Internet de las Cosas. Otro de los grandes fabricantes como es Intel también ha apostado por este mercado presentando Quark, su SoC para IoT. Debido a la evolu-

¹²https://en.wikipedia.org/wiki/System_on_a_chip

ción creciente de Internet de las Cosas, otros fabricantes como Mediatek¹³ (MT7687) o Qualcomm¹⁴ (QCA401x, QCA4531) han empezado a desarrollar productos en este ámbito. Además de éstos, los procesadores RISC¹⁵ de Atmel se han hecho un hueco dentro del ámbito de Internet de las Cosas de la mano de Arduino, una plataforma de prototipos de código abierto, que los lleva en la mayoría de sus placas.

Los sensores y actuadores no se han quedado atrás en esta evolución. Éstos son los elementos hardware que interactúan entre la tecnología y el entorno, capturando los datos deseados. Antiguamente, la electrónica estaba enfocada casi exclusivamente a un ámbito profesional, pero la llegada de Arduino y placas similares han extendido su uso de manera notable. El bajo coste de los sensores y actuadores y el enorme catálogo de accesorios disponibles ha sido un elemento importante en este crecimiento. Hoy en día, es relativamente fácil disponer de sensores táctiles, acelerómetros, potenciómetros, de inclinación, de humedad, de temperatura, de altitud, de presión, de luminosidad, etc. y hacerlos funcionar rápidamente con unos conocimientos muy básicos de electrónica.

El último pilar dentro de esta evolución del hardware para Internet de las Cosas es la comunicación. Una vez se dispone de pequeñas computadoras que recogen datos del entorno a través de los sensores, en muchos casos es necesario procesarlos de forma rápida y éstos no son los suficientemente potentes para hacerlo. Por lo tanto, es inevitable enviar esa información a otro ordenador a través de algún canal de comunicación. Por una parte, están los protocolos tradicionales que continúan vigentes en el Internet de las Cosas, tales como conexiones de red local vía Ethernet o de transmisión inalámbrica (GPS, 3G, 4G, GPRS, Wi-Fi, Bluetooth, etc.). Y por otra parte en los últimos años han aparecido nuevos protocolos enfocados a Internet de las Cosas. Ejemplo de éstos son NFC

¹³<http://www.mediatek.com/en/news-events/mediatek-news/mediatek%2Daddresses%2Diot%2Ddevelopers%2Dpower%2Defficiency%2Dand%2Dsecurity%2Ddemands%2Dwith%2Dhighly%2Dintegrated%2Dwifi%2Dsoc/>

¹⁴<http://www.forbes.com/sites/greatspeculations/2015/05/19/qualcomm%2Dstrengthens%2Dits%2Diot%2Dfootprints%2Dwith%2Dduo%2Dof%2Dwifi%2Dchips%2Dfor%2Dconnected%2Ddevices/>

¹⁵http://www.atmel.com/technologies/cpu_core/avr.aspx

o Bluetooth 4.0 LE 'Low Energy', precisamente diseñado para ser implementado en sistemas con baterías reducidas. Esta línea es un foco importante de investigación para los fabricantes, que buscan estándares de comunicación de muy bajo consumo. Algunos como Li-Fi¹⁶ (transmisión de datos a través de la luz), sobre el que se lleva años investigando, irán llegando en un futuro cercano.

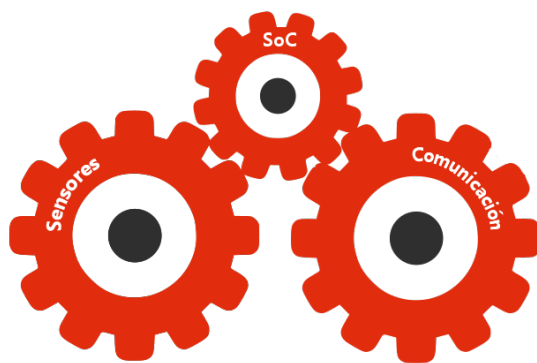


Figura 2.2: Los 3 ejes del hardware para el Internet de las Cosas.

2.4.1 Plataformas hardware para Internet de las Cosas

En esta subsección se van a presentar las plataformas hardware más relevantes en el ámbito de Internet de las Cosas y que incorporan las tecnologías descritas.

Arduino

Arduino¹⁷ es definido por sus creadores como *"una plataforma de prototipos electrónica de código abierto basada en hardware y software flexibles y fáciles de usar"*. Está ideado para artistas, diseñadores y cualquier interesado en crear objetos o entornos interactivos.

¹⁶<https://en.wikipedia.org/wiki/Li-Fi>

¹⁷<https://www.arduino.cc/>

Arduino puede obtener información del entorno a través de las entradas de múltiples sensores y puede actuar sobre el mismo mediante el control de luces, motores, sonidos y otros artefactos. Tiene un entorno de desarrollo basado en Processing¹⁸ y el microcontrolador de la placa se programa en un lenguaje basado en Wiring¹⁹.

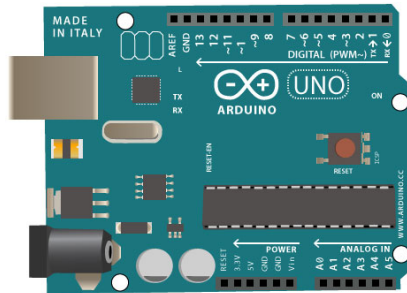


Figura 2.3: Arduino UNO.

El software se puede descargar de forma gratuita y los diseños de referencia del hardware están disponibles bajo licencia Creative Commons²⁰, por lo que cada uno es libre para adaptarlos en función de sus intereses o necesidades.

El hardware consiste en una placa con un microcontrolador, en la mayoría de ellas Atmel AVR (Atmega168, Atmega328, Atmega1280, ATmega8), y puertos de entrada/salida analógicos y digitales. Pese a la aparición de nuevos modelos de Arduino con microcontroladores CortexM3 de ARM de 32 bits, se pueden programar con el mismo entorno de desarrollo y compilarlos del mismo modo. A diferencia de la mayoría de las placas con AVR que generalmente funcionan a 5V, las que utilizan los ARM funcionan a 3.3V. Además hay múltiples placas, denominadas *shields*, que permiten añadir funcionalidades a los Arduino, tales como interfaces de comunicación (Ethernet, Wi-Fi, RF, GPRS, Bluetooth, XBee, etc.),

¹⁸<https://processing.org/reference/environment/>

¹⁹<http://wiring.org.co/>

²⁰<https://creativecommons.org/>

pantallas, cámaras de vídeo, reproductor de MP3, movimiento de motores, etc.

Uno de los puntos fuertes de esta plataforma es la gran comunidad de usuarios que hay detrás de la misma y que provoca que la curva de aprendizaje sea muy rápida. Tal ha sido la extensión/impregnación de este producto, que incluso Google colaboró en el desarrollo del Kit Android ADK, una placa Arduino capaz de comunicarse directamente con teléfonos móviles inteligentes bajo el sistema operativo Android para controlar desde el propio dispositivo sensores y actuadores conectados a Arduino.

Entre las placas oficiales, se pueden encontrar multitud de modelos. Todas especialmente pensadas para un fin, compatibles con las *shield* oficiales, así como con el entorno de desarrollo de Arduino. A continuación, se detallan las principales características de dos de las placas más extendidas: Arduino UNO, que es la primera que salió al mercado y la mayoría de las placas restantes implementan sus características; y Arduino MEGA, que es una placa que ofrece mayor capacidad de proceso, más entradas/salidas digitales y analógicas, más memoria y un mayor número de puertos serie.

	UNO	MEGA 2560
Microcontrolador	ATmega328P	ATmega2560
Voltaje de operación	5 V	5 V
V. de entrada recomendado	7 - 12 V	7 - 12 V
Pines E/S digitales	14	54
Pines E/S PWM digitales	6	15
Pines E analógicos	6	16
Memoria Flash	32 KB	256 KB
SRAM	2 KB	8 KB
EEPROM	1 KB	4 KB
Velocidad del reloj	16 MHz	16 MHz
Longitud	68.6 mm	101.52 mm
Anchura	53.4 mm	53.3 mm
Peso	25 g	37 g
Consumo energético	50 mA	60 mA
Precio	25 €	45 €

Tabla 2.2: Características de Arduino UNO y Arduino MEGA 2560.

Los requisitos del proyecto a desarrollar (número de pines analógicos y digitales, tamaño del código fuente, voltajes a manejar o interfaces de comunicación) van a determinar qué tipo de Arduino se utiliza.

La virtud de Arduino de que cualquier persona con unos conocimientos básicos de electrónica y programación pueda diseñar e implementar sus ideas, ha provocado un impulso en el Internet de las Cosas. Hay innumerables proyectos de Arduino (4.780.000 resultados²¹), de los cuales al menos el 25 % son relacionados con Internet de las Cosas²². Entre ellos se encuentran sensores para el hogar, circuitos de control y gestión de cámaras de videovigilancia o termostatos inteligentes al estilo *Nest*.

Raspberry Pi

Raspberry Pi es una placa computadora de bajo coste desarrollada por la fundación *Raspberry Pi Foundation* con el objetivo de estimular la enseñanza en las escuelas del campo de ciencias de la computación.

Se trata de una placa base de 85 x 54 milímetros que en su última versión (*Raspberry Pi 3*) contiene un procesador ARM Cortex-A53 de 4 núcleos a 1,2 GHz, un procesador gráfico VideoCore IV 3D y 1 Gbyte de memoria RAM. Además de una ranura para microSD, donde se almacena el sistema operativo, dispone de un puerto de salida de vídeo HDMI y otro tipo RCA, minijack de audio, 4 puertos USB, un puerto Ethernet y 17 pines E/S. En cuanto a las interfaces de comunicación, dispone de un puerto Ethernet y otro de Wi-Fi integrado, así como de Bluetooth 4.1 LE. A diferencia de Arduino, que no tiene un consumo energético excesivamente alto, el de la Raspberry Pi aumenta notablemente hasta los 800 mA.

²¹<https://www.google.es/search?q=arduino%20projects&rct=j>

²²<https://www.google.es/search?q=arduino%20projects&rct=j#q=arduino+projects+iot>

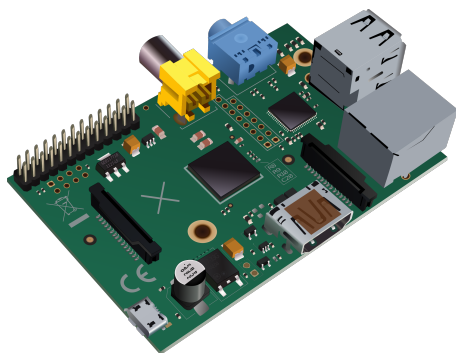


Figura 2.4: Raspberry Pi B.

En lo que se refiere al sistema operativo, Raspberry Pi utiliza mayoritariamente sistemas operativos basados en Linux. Windows, en su apuesta por el Internet de las Cosas, ofrece una versión de Windows 10 (IoT Core), que alejada de la versión de escritorio, está orientada a dispositivos limitados por sus recursos.

Al igual que Arduino, el gran éxito y la gran comunidad que se ha formado alrededor de esta placa han hecho que se disponga en el mercado de diversos accesorios y módulos para aumentar sus capacidades. Incluso hay un módulo que permite montar las *shields* de Arduino sobre la Raspberry Pi.

Pese a seguir siendo el segmento educativo una parte fundamental de la Fundación Raspberry Pi, las mejoras llevadas a cabo en la última versión la han convertido en un gran candidata a entrar como herramienta de desarrollo de soluciones para el Internet de las Cosas, ya que le permiten interactuar de forma más fácil con el mundo físico. Tiene un precio aproximado de 40 €.

Intel

Con el objetivo de paliar los riesgos que deben asumir muchas empresas en fases tempranas de desarrollo de producto y aplicaciones, Intel lanzó al mercado las plataformas Intel Galileo e Intel Edison. Se tratan de dos placas de desarrollo asequibles para crear proyectos. Haciendo frente a las plataformas que montan un chip ARM, Intel ha apostado por Quark para tratar de convertirse en el soporte principal de Internet de las Cosas, donde se busca la conexión del mundo digital y el mundo real.

Cada una de las plataformas tienen sus peculiaridades. Intel Galileo es la primera de una familia de placas de desarrollo y para prototipos certificada por Arduino y basada en arquitectura Intel. Proporciona un entorno de desarrollo de software y hardware de código abierto y es compatible con diversas *shields* de Arduino UNO. Contiene un procesador de aplicaciones Intel Quark SoC X1000 de 32 bits, 256 MB de memoria RAM, SRAM integrada de 512 kb, memoria Flash de 8 MB, EEPROM de 8 kb y admite una tarjeta microSD de hasta 32 GB. También proporciona varias interfaces de E/S, tales como Ethernet, USB y mini-PCI Express. Se alimenta entre 7 y 15 V y además de ser programable mediante el IDE de Arduino, tiene soporte para C, C++, Python y Node.js/Javascript. Su precio aproximado es de 90 € y tiene un consumo de alrededor de 500 mA.

Por otra parte, Intel Edison se trata de una placa del tamaño de una tarjeta SD (ver figura 2.5) enfocada a las tecnologías vestibles²³. Se busca la posibilidad de utilizar una gran cantidad de sensores y la facilidad para realizar desarrollos con un coste limitado. Contiene un chip Intel Atom de doble núcleo a 500 MHz y también un microcontrolador a 100 MHz que permite recopilar y preprocesar datos en un estado de bajo consumo y trasladar los datos filtrados a la CPU para su análisis. Incluye 1GB de memoria RAM, 4GB de almacenamiento interno, interfaces de Wi-Fi y Bluetooth 4.0 y admite 40 GPIO con múltiples configuraciones. El sistema operativo es Yocto Linux 1.6. Esta placa tiene un precio de alrededor de 65 €.

²³https://en.wikipedia.org/wiki/Wearable_technology



Figura 2.5: Intel Edison e Intel Galileo.

Análisis y conclusiones

Las requisitos de hardware básicos de la propuesta que se hace en esta tesis doctoral son los siguientes:

- Capacidad de almacenamiento de 1 MB, además de lo que ocupa el propio sistema.
- Interfaz de comunicación Ethernet o Wi-Fi.
- 1 pin de entrada analógica para conectar el sensor de corriente.
- Reloj en tiempo real.
- 1 pin de salida digital para controlar un relé que posibilite el apagado automático de la cafetera.

Teniendo en cuenta las necesidades básicas descritas, rápidamente se deduce que cualquiera de las placas presentadas serían candidatas a formar parte del sistema que se propone. Pero dado que el objetivo final de la infraestructura es el ahorro de energía, se debe hacer hincapié en este apartado. De entre las cuatro, Arduino es la que menos energía consume, siendo su consumo del orden de 50 mA. Teniendo en cuenta que debe estar a diario conectado durante 24 horas y que habría que añadir una *shield* de comunicación Ethernet, el consumo energético ya no sería tan insignificante. Además del microcontrolador, el Arduino integra un regulador de tensión, un conversor Serie-USB, leds, etc. que consumen energía. Algunos de estos módulos serían prescindibles, pero implicaría

que el uso de Arduino no sería tan sencillo y el objetivo era diseñar una plataforma para facilitar el acceso a la computación física por parte de artistas, aficionados, etc. Por lo tanto, tiene prioridad la sencillez y comodidad sobre la eficiencia energética. Haciendo referencia a cuestiones monetarias, un Arduino oficial con interfaz Ethernet tiene un precio aproximado de 50 €.

La Raspberry Pi se ha descartado debido al alto consumo energético que produce en relación con el ahorro que genera la infraestructura propuesta en esta tesis doctoral. El consumo de 800 mA durante las 24 horas del día se considera inasumible. Por este mismo motivo, se descarta la placa Intel Galileo, que consume 500 mA.

Intel Edison tiene un consumo menor que Raspberry Pi e Intel Galileo. Consume, según pruebas experimentales²⁴, 50 mA, que asciende a 120 mA enviando datos a través de Wi-Fi. Además de no tener un consumo excesivamente bajo, el otro inconveniente radica en su precio, ya que incrementaría el valor de los objetos cotidianos de forma notoria. Añadir que tanto Intel Edison, como Intel Galileo tienen poca documentación en relación con la que hay disponible para Arduino y Raspberry Pi.

En la tabla 2.3 se muestra una comparativa de las cuatro plataformas hardware presentadas:

Plataformas	Finalidad	Ventajas	Desventajas
Arduino	Proyectos de propósito simple	Mucha documentación. Ideal para iniciarse en electrónica. Barato.	No válido para soluciones que requieren mucha capacidad de cómputo.
Raspberry Pi	Proyectos multimedia	Mucha documentación. Gran relación capacidad de cómputo / Precio.	Limitado para proyectos de electrónica.
Galileo	Internet de las Cosas	Compatible con shields de Arduino. Soporta múltiples lenguajes.	Poca documentación. Precio alto.
Edison	Internet de las Cosas	Tamaño. Soporta múltiples lenguajes.	Poca documentación. Precio alto.

Tabla 2.3: Comparativa de las placas hardware de Internet de las Cosas.

²⁴<http://www.helios.de/heliosapp/edison/>

Tras hacer esta valoración, se ha considerado que Arduino es la plataforma económicamente más asequible para realizar un despliegue en varios dispositivos como parte del experimento principal de la tesis. Después de una búsqueda entre las diferentes placas oficiales y no oficiales de Arduino, se ha seleccionado la placa iBoard Pro²⁵, ya que tiene una interfaz Ethernet que aporta mayor fiabilidad a la conexión que la interfaz Wi-Fi, dispone de reloj en tiempo real y no tiene un precio excesivamente alto, 35 €.

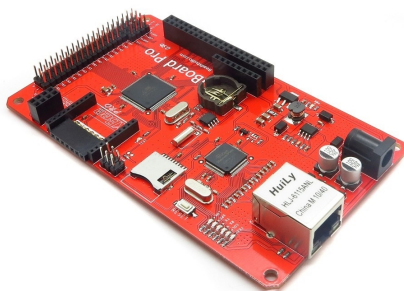


Figura 2.6: Placa iBoard Pro.

Como se puede ver más adelante (sección 4.3.1), en la tesis se propone utilizar una solución hardware a medida que utilice únicamente los componentes imprescindibles. Se trata de este modo, que el consumo energético sea insignificante.

²⁵<https://www.itead.cc/development-platform/arduino/arduino-compatible-mainboard/iboard-pro.html>

2.5 Plataformas software para Internet de las Cosas

Ante la falta de objetos cotidianos que colaboran a través de la red y con el objetivo de ofrecer una solución que los habilite para ello, se ha llevado a cabo una revisión bibliográfica de las arquitecturas software para Internet de las Cosas que se han propuesto y/o desarrollado en los últimos años para ver cómo podrían encajar en la propuesta que se hace en esta tesis doctoral.

El término Internet de las Cosas se refiere a un amplio conjunto de aplicaciones y áreas de investigación, tales como la computación distribuida y la gestión del conocimiento. Gran parte de las investigaciones llevadas a cabo sobre IoT se relacionan con el campo de la computación ubicua. World Wide Web Consortium (W3C) define la computación ubicua como *"un mundo donde las personas estarían rodeadas de sistemas informáticos que estarían conectados a través de redes, y que nos proporcionaría ayuda en cada cosa que lleváramos a cabo"*. La computación ubicua²⁶ trata de facilitar la vida al ser humano mediante la explotación de las tecnologías informáticas disponibles.

Por una parte, en esta sección se van a presentar las infraestructuras software desarrolladas para habilitar el Internet de las Cosas y la computación ubicua, y por otra parte se va a hacer referencia a las investigaciones llevadas a cabo en torno a las redes sociales de objetos dentro del Internet de las Cosas.

2.5.1 Arquitecturas para Internet de las Cosas

Tal y como afirma Bandyopadhyay et al. (?), los middleware son esenciales para hacer realidad el Internet de las cosas. A continuación se enumeran los principales motivos:

²⁶<http://www.w3.org/wiki/PervasiveComputing>

- Dificultad para definir y hacer cumplir un estándar común entre todos los dispositivos que pertenecen al dominio de Internet de las Cosas.
- El middleware actúa como nexo entre componentes heterogéneos.
- Aplicaciones de diversos dominios demandan una capa de abstracción/adaptación.
- El middleware ofrece una API (en inglés *application programing interface*) para, independientemente de su naturaleza, comunicar la capa física y los servicios necesarios para las aplicaciones.

HYDRA

Hydra (?) es uno de los middleware más relevantes dentro del Internet de las Cosas. Nació fruto de un proyecto europeo del Sexto Programa Marco y fue desarrollado para tres dominios de aplicación: automatización, asistencia sanitaria y agricultura (?). Hydra es un software inteligente que media entre las aplicaciones y el sistema operativo. Este middleware proporciona una interfaz de servicios web para interactuar con cualquier dispositivo físico, actuadores, sensores o subsistemas, independientemente de sus tecnologías de red, ya sean Bluetooth, RF, Zig-Bee, RFID, Wi-Fi, etc.

Hydra ha sido diseñado para facilitar la interacción entre dispositivos de diferente naturaleza abstrayéndolos de la información detallada de estos dispositivos y sus redes. Considera cada objeto como un servicio y utiliza lenguajes de ontologías, tales como OWL, OWL-S y SAWSDL, para describirlos semánticamente. Además, provee una capa de servicio que permite a los usuarios finales interactuar con estos dispositivos sin tener en cuenta qué tecnologías de comunicación soporta cada uno de ellos.

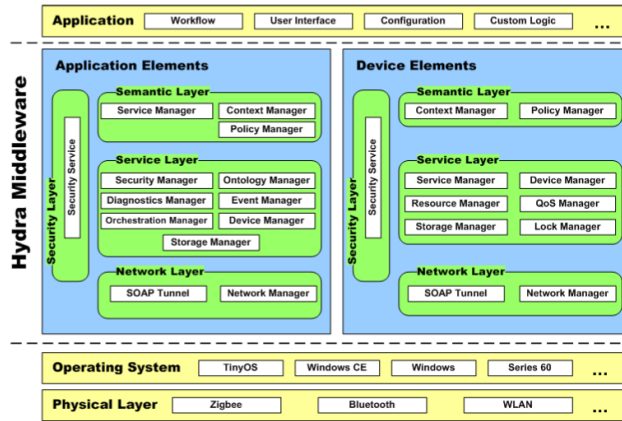


Figura 2.7: Arquitectura de HYDRA.

Este software ya ha sido utilizado en proyectos de eficiencia energética. Jahn et al. (?) propusieron un hogar inteligente comprometido con el ahorro de energía basado en el middleware HYDRA. Su sistema integraba medidores de energía y proveía interfaces para monitorizar y controlar diferentes dispositivos.

SOFIA2

SOFIA2²⁷ es un middleware desarrollado por Indra²⁸ que tiene su origen en el proyecto europeo SOFIA²⁹. Permite la interoperabilidad de múltiples sistemas y dispositivos, ofreciendo una plataforma semántica que permite poner información del mundo real a disposición de aplicaciones inteligentes (Internet de las Cosas). Es multilenguaje y multiprotocolo, permitiendo así la interconexión de dispositivos heterogéneos. Proporciona mecanismos de publicación y suscripción, facilitando la orquestación de sensores y actuadores para monitorizar y actuar sobre el

²⁷<http://sofia2.com/>

²⁸<http://www.indracompany.com/>

²⁹<http://www.indracompany.com/sostenibilidad%2De%2Dinnovacion/proyectos%2Dinnovacion/sofia%2Dobjetos%2Dinteligentes%2Dpara%2Daplicaciones%2Dinteli>

entorno. Además es multiplataforma y multidispositivo, a través de su SDK, API's y mecanismos de extensión que permite su integración con cualquier tipo de dispositivo.

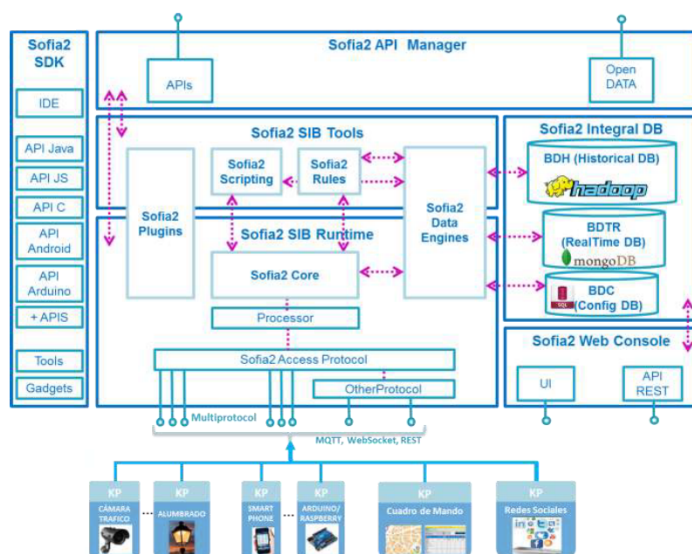


Figura 2.8: Arquitectura modular de SOFIA2.

A continuación se describen algunos de los diferentes componentes de SOFIA2 que se muestran en la figura 2.8:

- **SIB Runtime:** es el core de la plataforma, el que publica interfaces (MQTT, WebSockets, REST,...) para publicar y consultar información.
- **SIB API Manager:** es la pieza que cuando se publican APIs REST sobre las ontologías (o APIs externas) permite acceder a ellas.
- **SIB Web Console:** consola Web + API Web para gestionar/administrar la plataforma.
- **SIB Tools:** se encarga de lanzar las reglas y otros procesos como el proceso de paso BDTR a BDH.

- SDK: ofrece APIS y herramientas para trabajar con Sofia2.

TinyDB

TinyDB (?) fue uno de los primeros proyectos que propuso la idea de abstraerse de los dispositivos. Este middleware, que fue diseñado para extraer datos de redes de sensores inalámbricos, busca la interacción de los usuarios finales con los dispositivos sin conocer detalladamente la especificación de los dispositivos, tales como los protocolos de comunicación soportados por los mismos. TinyDB provee un lenguaje específico del dominio (en inglés *Domain Specific Language - DSL*) para interactuar con los objetos. Este DSL permite al usuario final obtener información sobre el tiempo, el lugar, el tipo y el método de muestreo en un entorno de sensores embebidos.

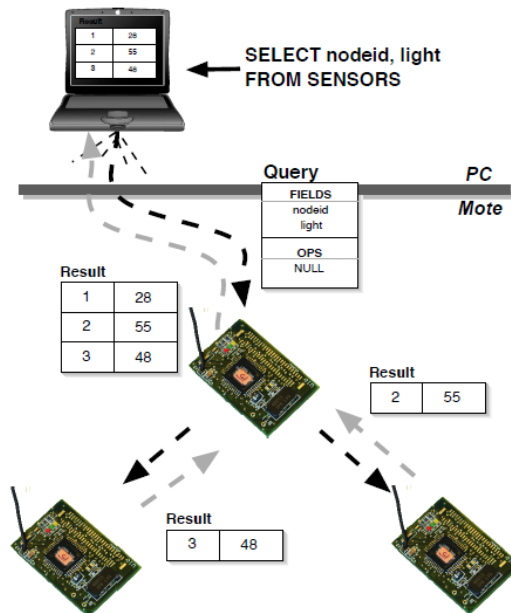


Figura 2.9: La consulta y los resultados propagándose a través de la red.

Otsopack

Otsopack (?) es un middleware para entornos ubicuos basado en una implementación descentralizada de Triple Space³⁰. El middleware está ideado para ser desplegado en dispositivos con recursos computacionales limitados, como móviles o dispositivos embebidos, que comparten su información contextual a través de la red formando el Internet de las Cosas.

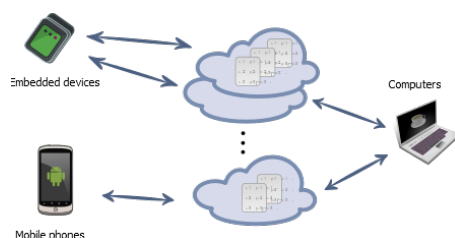


Figura 2.10: Esquema de Otsopack.

SENSEI

Fruto de un proyecto europeo del Séptimo Programa Marco se diseñó la arquitectura abierta SENSEI (???) para integrar el mundo físico en el mundo digital. Se trata de una arquitectura en capas que representa el mundo físico en dos niveles de abstracción: el nivel de recursos y el de contexto. En el primero de ellos se permite recoger datos, en bruto o con meta-datos, de un gran número de dispositivos distribuidos globalmente (WS&AN - Redes de sensores y actuadores). Al mismo tiempo, el nivel de contexto proporciona servicios de red y gestiona la información necesaria para permitir una recuperación fiable y precisa de contexto e interacción con el medio físico. La información se puede representar en diversos formatos, tales como XML (*eXtensible Markup Language*), RDF

³⁰La computación Triple Space es un paradigma de coordinación basado en la computación *tuplespace*, donde se utilizan técnicas de web semántica para definir el conocimiento intercambiado.

(Resource Description Framework) o OWL (Web Ontology Language).

La figura 2.11 presenta la arquitectura de SENSEI:

- **Resources:** sensores, procesadores y actuadores que proveen información del mundo físico o permiten interactuar con él.
- **Support Services:** ofrece servicios de descubrimiento, consulta y composición de recursos.
- **Community Management:** ofrece gestión de usuarios, gestión de identidad y mecanismos de seguridad y privacidad.

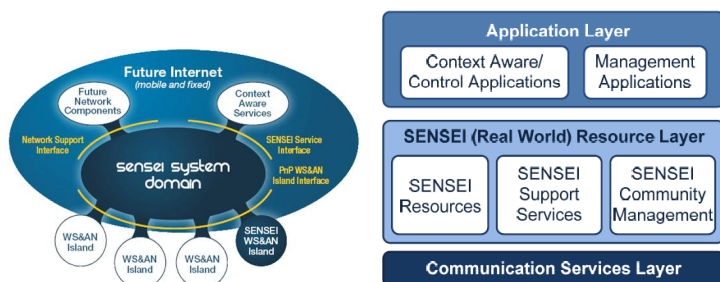


Figura 2.11: Arquitectura SENSEI.

OpenIoT

OpenIoT³¹ (?) propone un *framework*³² abierto para Internet de las Cosas basado en software de código abierto. Establece un ecosistema que engloba dispositivos y software relacionados con Internet de las Cosas. En la figura 2.12 se muestra la visión que persigue la propuesta OpenIoT:

1. Los desarrolladores de hardware crean dispositivos para Internet de las cosas y registran sus APIs (Interfaz de programación de aplicaciones) en el *Open API Portal*.

³¹<http://www.openiot.eu/>

³²https://en.wikipedia.org/wiki/Software_framework

2. Haciendo uso de las API registradas, los desarrolladores de software crean aplicaciones que se comunican con los dispositivos. Éstas se registran en tiendas de aplicaciones, tales como Apple Store³³ o Play Store³⁴.
3. Los proveedores de servicios compran dispositivos IoT y los registran en el sistema, donde pueden ser monitorizados y gestionados de forma eficiente.
4. Los consumidores podrían encontrar y conectar de manera sencilla con los dispositivos utilizando servicios de búsqueda.

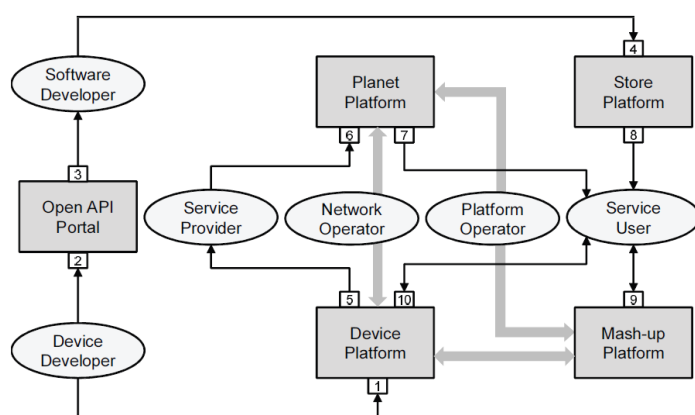


Figura 2.12: Ecosistema propuesto por OpenIoT.

OpenIoT hace uso de tecnologías de datos enlazados abiertos (en inglés *Open Linked Data*) y de ontologías, modelos y anotaciones semánticas para representar objetos.

Xively

³³<https://itunes.apple.com/es/genre/ios/id36?mt=8>

³⁴<https://play.google.com/store>

Previamente conocido como Pachube y Cosm, Xively³⁵ proporciona la posibilidad de subir datos de objetos haciendo uso de las librerías que proporcionan. Un usuario puede subir a su perfil los datos de diferentes dispositivos y a éstos se les puede añadir disparadores, para que, tras cumplirse una determinada condición, realice una llamada a un servicio web determinado. Este sistema está habilitado para trabajar en cualquier plataforma, ya sea un desarrollo en Linux, Android, un sistema operativo Embedded C, Arduino, etc.

Xively dispone de una API basada en los principios REST (en inglés *Representational State Transfer*) que actúa como interfaz para leer y escribir datos en Xively. Acepta formatos como JSON, XML o CSV. Además, dispone de un servicio comercial orientado a empresas que requieran un soporte dedicado para su propia Internet de las Cosas.

A continuación, en la figura 2.13, se presenta la arquitectura de Xively.

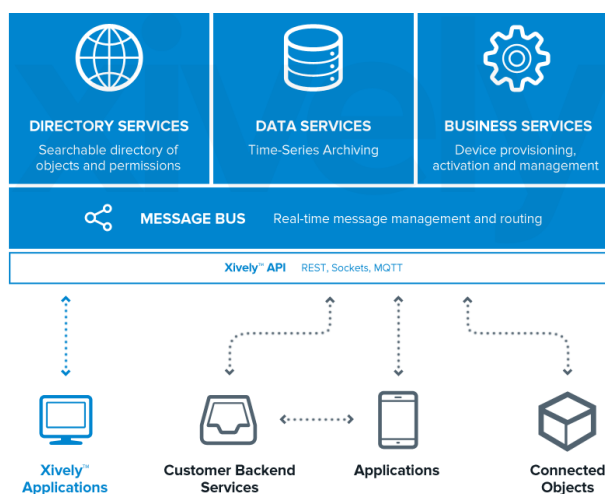


Figura 2.13: Arquitectura de Xively.

³⁵<https://xively.com/>

Carriots

Carriots³⁶ es una plataforma para Internet de las Cosas que pretende proporcionar una plataforma M2M para la integración fácil, rápida y barata de los dispositivos. Se creó en 2010 para gestionar los datos de millones de dispositivos y en 2013 se puso en marcha la versión comercial. La versión gratuita para desarrolladores ofrece la posibilidad de gestionar hasta 10 dispositivos.

Ofrece un SDK (*Software Development Kit*) basado en Groovy³⁷ para el desarrollo de software en dispositivos externos. Estos dispositivos interactúan con la plataforma Carriots a través de la API REST que proporciona. Las peticiones y respuestas HTTP soportan formatos JSON y XML.

En la figura 2.14 se muestra un caso de uso³⁸ de Carriots.

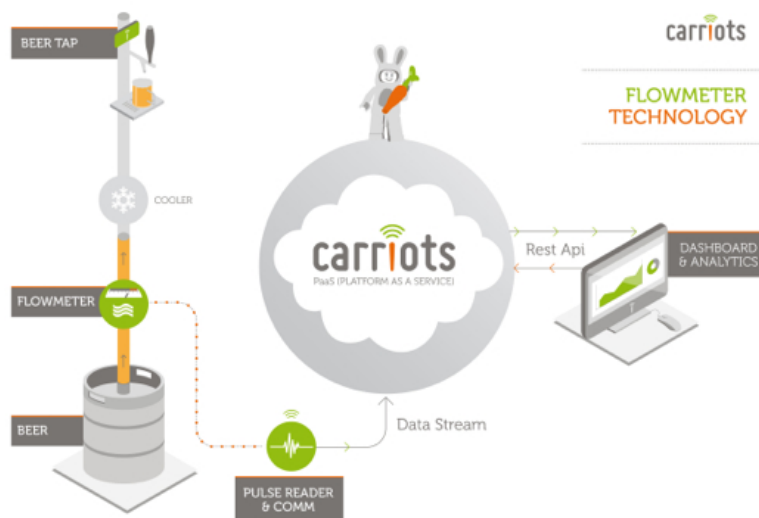


Figura 2.14: Caso de uso de Carriots.

³⁶<https://www.carriots.com/>

³⁷<http://www.groovy-lang.org/>

³⁸https://www.carriots.com/cool_project/flowmeter

ThingSpeak

ThingSpeak³⁹ es una plataforma web que ofrece servicios para la recolección de datos en tiempo real de los dispositivos, procesamiento de los datos, visualizaciones de datos para una amplia gama de aplicaciones, tales como monitoreo de sensores o control de la energía. Almacena los datos en canales ThingSpeak y éstos se representan como una interfaz web donde se publican los datos almacenados. Proporciona una API de código abierto para la conexión a la infraestructura ThingSpeak. Las aplicaciones cliente están integradas en los dispositivos físicos que pueden leer y escribir a un canal ThingSpeak mediante peticiones HTTP a la API de ThingSpeak. Los canales soportan los formatos JSON, XML y CSV. Además dispone de una Chart API para mostrar los datos almacenados en los canales ThingSpeak en gráficos estáticos o dinámicos, que pueden presentar datos numéricos y pueden ser embebidos en sitios web externos.

La figura 2.15 presenta el diagrama de un sistema⁴⁰ que hace uso de ThingSpeak para controlar un dispositivo físico a través de la web.

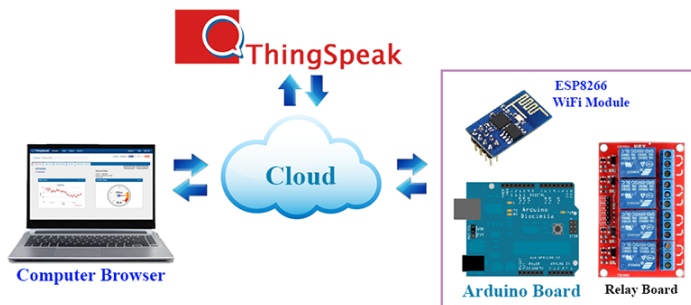


Figura 2.15: Diagrama de un sistema con ThingSpeak.

³⁹<https://thingspeak.com/>

⁴⁰<http://microcontrollerkits.blogspot.com.es/2015/05/arduino-esp8266-iot-control-thingspeak.html>

Análisis y conclusiones

Analizando las arquitecturas presentadas, se observa que se pueden categorizar en dos tipos. Por una parte están las que son un middleware al uso para Internet de las Cosas. Éstos son HYDRA, SOFIA2, TinyDB, Otsopack, WISemid, SENSEI y OpenIoT. Por otra parte están las que se acercan a lo que podría ser una red social (Carriots, Xively y ThingSpeak), permitiendo a los desarrolladores conectar sensores a la Web para crear aplicaciones. Sin embargo, no permiten a los objetos crear grupos de forma autónoma, por lo que se hace imprescindible la intervención del ser humano.

En cuanto a los middleware se refiere, se ha observado que algunos de ellos tienen muy poca documentación y no están extendidos (HYDRA, OpenIoT). SOFIA se trata de la más completa de las plataformas observadas y tiene API para Arduino, pero para ofrecer una alta disponibilidad es necesaria una infraestructura dedicada y hacer frente a un coste económico como contrapartida a su uso. Otsopack y TinyDB son los más ligeros, pero actualmente ninguno de los dos dispone de soporte estable para Arduino. Hay que tener en cuenta que en la propuesta de esta tesis doctoral, la interoperabilidad entre dispositivos no es primordial, ya que todos los que se van a utilizar son de la misma familia (Arduino), y por lo tanto utilizarán el mismo protocolo de comunicación.

Carriots, Xively y ThingSpeak son bastante similares respecto al trato de la información, usando cada una de las plataformas su API correspondiente. Algunos de los formatos de datos, como XML y JSON, son soportados por todas las plataformas, en cambio, el formato PNG sólo Xively lo soporta. ThingSpeak y Carriots son las plataformas más recientes y la documentación es escasa en relación con la que proporciona Xively, que es una de las primeras plataformas de este tipo (originariamente Pachube). Con el modelo de negocio actual de Xively, únicamente soporta 10 sensores de forma gratuita. En cambio, tanto Carriots como ThingSpeak permiten un número indefinido de sensores. ThingSpeak tiene la desventaja respecto a las otras dos plataformas de que soporta menos dispositivos hardware (Arduino, Raspberry Pi y Electric Imp).

En la Tabla 2.4 se resumen las ventajas y desventajas de cada una de las infraestructuras presentadas.

Plataformas	Formatos	Ventajas	Desventajas
HYDRA	OWL, OWL-S, SAWSDL	Abstracción de los dispositivos.	No está extendido. No almacena los datos de forma segura.
SOFIA	JSON	Buena documentación. Multilenguaje. Multiplataforma.	Las versiones con alta disponibilidad requieren una infraestructura dedicada y son de pago.
TinyDB	Por defecto JSON	Bajo consumo. Rapidez en la recuperación de datos. Código abierto.	Sólo soporta dispositivos basados en TinyOS. No abstrae al usuario final de la aplicación.
Otsopack	RDF	Escalable. Información descentralizada y actualizada siempre. Código abierto.	Poca viabilidad para dispositivos con recursos muy limitados. Soporta pocos tipos de dispositivos.
SENSEI	XML, RDF, OWL	Promete interoperabilidad con muchos dispositivos.	No hay ningún desarrollo.
OpenIoT	--	Plataforma colaborativa. Código abierto.	Muy incipiente. Poca documentación
Xively	JSON, XML, CSV, PNG	Mucha documentación.	Hasta 10 sensores.
Carriots	JSON, XML	Uso de librería de Clicarriots ⁴¹ .	Poca documentación.
ThingSpeak	JSON, XML, CSV	Interfaz amigable de los canales.	Poca documentación. Poco hardware compatible.

Tabla 2.4: Comparativa de plataformas software de Internet de las Cosas.

2.5.2 Redes sociales para Internet de las Cosas

Boyd y Ellison (?) crearon la siguiente definición para una red social tal y como la conocemos hoy en día: *"Se trata de un servicio Web que permite a individuos construir un perfil público o semi-público dentro de un sistema compartido, organizar una lista de usuarios del sistema con los que compartir una conexión y ver y navegar a través de la lista de conexiones propias y de las del resto de usuarios"*.

Por otra parte, Gallego (?) define las redes sociales como *"conjuntos de individuos que se encuentran relacionados entre sí, por motivos de muy diversa índole, que van desde los negocios hasta la amistad"*. En el ámbito de la

⁴¹<https://github.com/sdeancos/clicarriots>

informática, "la red social hace alusión al sitio web que estas personas utilizan para generar su perfil, compartir información, colaborar en la generación de contenidos y participar en movimientos sociales".

Berners-Lee expone en el libro *Weaving the Web* (?) su sueño de que la web se convierta en un potente medio de colaboración entre personas, donde éstas no sólo accedan a la web en busca de contenidos, sino con el fin de crearlos. Además añade, que la colaboración debería extenderse a los ordenadores, de modo que éstos sean capaces de analizar los contenidos de la web.

En definitiva, se considera que las redes sociales son plataformas de comunidades virtuales que proporcionan información e interconectan personas con afinidades comunes.

En los últimos años son comunes las investigaciones en las que se añaden los mundos de Internet de las Cosas y de las redes sociales (????????).

Holmquist et al. (?) fueron los primeros en hacer referencia a la socialización de objetos. El enfoque de esa investigación se centra en analizar el comportamiento colectivo de unos dispositivos aumentados con capacidad de proceso, sensores y comunicación inalámbrica para realizar tareas de percepción colectiva del entorno. Este trabajo fue una evolución de una investigación previa en la que los dispositivos adquirían información del contexto de forma individual. En 2001, año de publicación de este artículo, tanto el concepto de Internet de las Cosas como el de las redes sociales eran muy incipientes.

Una investigación más reciente (?) augura una nueva generación de objetos llamados con el neologismo *Blogjects*, que intervienen en las actividades diarias de los seres humanos. Son objetos inteligentes con capacidad de interacción entre sí, algo que años atrás era inconcebible. Bleecker et al. defendían que estos objetos en un futuro cercano participarían en todo el entramado que ahora es la web social, y por lo tanto en el Internet de las Cosas. La característica más peculiar de los *Blogjects* es que participan en el intercambio de ideas. No sólo publican información, sino que son capaces de mantener una conversación.

Nazzi et al. (?) introdujeron el concepto teórico *Embodied Microblogging* (EM), que desafía en parte a la visión actual de Internet de las Cosas.

Definen EM para informar del diseño de objetos cotidianos aumentados que facilitan a las personas mayores el anuncio de sus actividades diarias dentro de su comunidad. En vez de centrarse en las interacciones objeto–objeto u objeto–humano, propone dos nuevos roles que jugarán los objetos cotidianos aumentados: *i)* mediar en la comunicación entre seres humanos y *ii)* ofrecer diferentes modos para anunciar actividades de la vida cotidiana.

Vázquez y López-de-Ipiña (?) introdujeron el concepto de *dispositivo social* y transformaron los objetos cotidianos en artefactos autónomos que se comunican con otros objetos, servicios o personas, incluso formando comunidades de dispositivos que imitan el comportamiento social. Apuestan por que el Internet de las Cosas se asemeje al Internet de las Personas. Al igual que un individuo utiliza Internet para buscar soluciones a un problema concreto, su visión es que los *dispositivos sociales* deberían funcionar de la misma manera. Cuando un objeto hace frente a un problema, debe “hablar” con otros dispositivos para adquirir sus experiencias o información que le pueda ayudar a encontrar una solución.

Ding et al. (?) señalan tres elementos clave de la sociedad: la información, los objetos y las personas. Cada uno de ellos se refleja en las siguientes redes: Internet, Internet de las Cosas y las redes sociales respectivamente. Aunque cada vez hay más excepciones, habitualmente estas redes trabajan de forma independiente. Este trabajo propone la idea de una plataforma para que las agrupe y además discute futuras aplicaciones de la misma.

Guinard et al. (?) investigaron la explotación de las redes sociales en el contexto de Internet de las Cosas. Propusieron una solución basada en las redes sociales (p. ej., Facebook, LinkedIn, Twitter, etc.) y sus APIs que permite el uso compartido de las cosas. La utilización de las redes sociales ya existentes habilita a los usuarios para compartir objetos conectados a Internet con personas que conocen y de confianza sin la necesidad de crear una nueva red social o un nuevo servicio en línea. A este enfoque se le identifica como *Web of Things* (?).

Kranz et al. (?) investigaron el potencial de combinar las redes sociales (p. ej., Twitter) con las redes tecnológicas (p. ej., una red de sensores)

para ofrecer servicios, tanto a los seres humanos como a los objetos. Discuten sobre las implicaciones de las redes socio-tecnológicas en el contexto de Internet de las Cosas.

En la misma línea de socializar los objetos de Internet de las Cosas introducida por Kranz et al. (?), Atzori et al. (?) propusieron un innovador paradigma de interacción entre objetos. La base del trabajo era la definición de una red social de objetos inteligentes a la que denominaron *Social Internet of Things (SIoT)*.

Realizaron un análisis de los distintos tipos de relaciones sociales que los objetos podían crear. La primera forma de socialización entre objetos que detectaron fue la que denominaron *Parental Object Relationship*. Esta engloba las relaciones entre objetos similares que han sido construidos en el mismo período y por el mismo fabricante. Además, los objetos podrían establecer relaciones *Co-location Object Relationship* y *Co-work Object Relationship*, al igual que las personas comparten experiencias de ámbito personal o público (p. ej., laboral). Estas relaciones se crean siempre que los objetos (sensores, actuadores o etiquetas RFID) están ubicados constantemente en el mismo lugar (p. ej., para ofrecer servicios de automatización en el hogar o la industria) o cooperan de forma periódica para proporcionar servicios de Internet de las Cosas, tales como respuestas de emergencia. Para los objetos pertenecientes al mismo usuario (*smartphones, tablets, etc.*) crearon un nuevo tipo de relación, denominada *Owner Object Relationship*. Por último, se definió la relación *Social Object Relationship* para los objetos que están en contacto debido a la relación que existe entre sus dueños.

Al igual que las personas intercambian información con sus contactos, los dispositivos autorizados intercambian sus perfiles sociales de forma autónoma. La idea principal es que los dispositivos con características y perfiles similares puedan compartir "buenas prácticas" para resolver problemas a los que ya se hayan enfrentado dispositivos "amigos".

Perez de Almeida et al. (?) presentaron la plataforma Thing Broker para la *Web of Things* propuesta por Guinard (?). A diferencia de otras plataformas (Open.Sen.Se, ThingSpeak o Xively) que proporcionan un servi-

cio en la *nube* con una interfaz de usuario para la gestión y visualización de sensores y sus datos, Thing Broker es una plataforma para el desarrollo de aplicaciones. Proporciona interfaces REST para, siguiendo un modelo de comunicación basado en Twitter (*following/followed*), comunicarse de forma abstracta con los objetos. Las principales abstracciones son *Objetos* y *Eventos* (ver figura 2.16). Los objetos contienen información sobre ellos mismos y sus relaciones con otros, y los eventos están asociados a los objetos y contienen los datos que éstos producen, desde texto plano hasta contenido multimedia. Su arquitectura se puede observar en la figura 2.17.

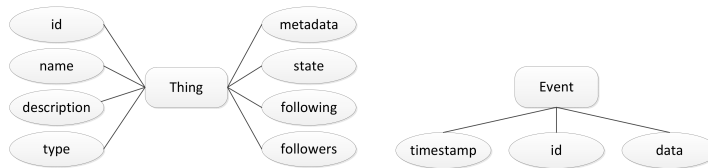


Figura 2.16: Abstracciones de *Objeto* y *Evento*.

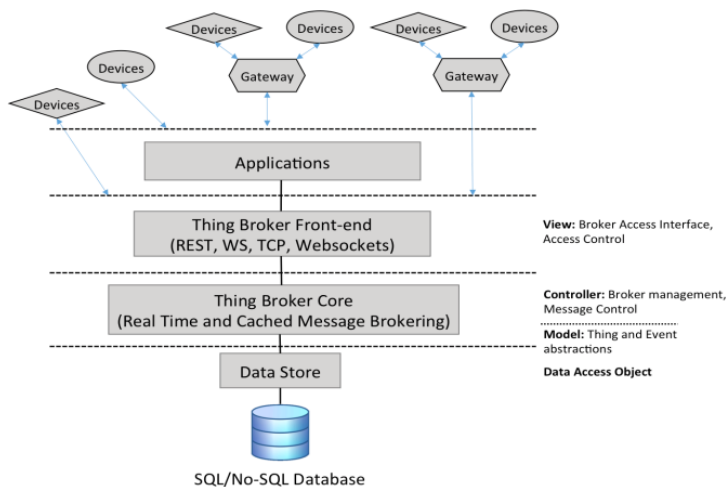


Figura 2.17: Arquitectura de Thing Broker.

Ciortea et al. (??), como se muestra en la figura 2.18, presentaron su visión de la evolución de la Web hacia una Web social de objetos (SWoT – *Social Web of Things*). Dentro de la *Web of Things* echaban en falta una nueva dimensión: la proactividad. Por lo tanto, propusieron extender y transformar la redes sociales integrando en ellas objetos autónomos y proactivos.

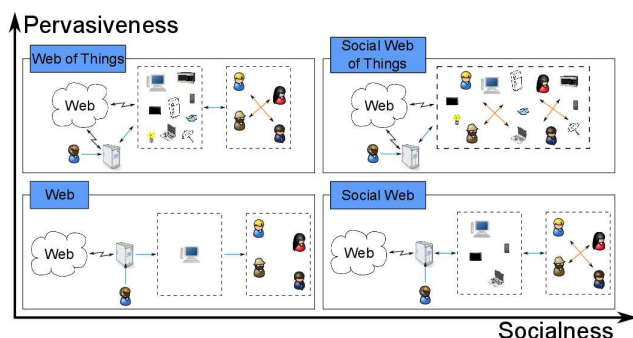


Figura 2.18: De la Web hacia la SWoT.

Análisis y conclusiones

Una realidad es que las relaciones personales de los seres humanos se han convertido más visibles y fácilmente cuantificables debido a las redes sociales. Los amigos se pueden contabilizar, organizar en grupos, se pueden buscar amigos de amigos, etc. Pero estas personas no sólo tienen relaciones de amistad entre ellas a través de estas redes, sino que también las tienen con objetos, libros, películas, *gadgets*⁴², etc. Al mismo tiempo, en el ámbito de Internet de las Cosas se han desarrollado objetos que comunican su estado y/o funcionalidad. Entre algunos de éstos está la lavadora que envía *tweets* cuando ha terminado su tarea⁴³, la cafetera que notifica en las redes sociales la energía consumida y desperdiciada (?), o el teléfono móvil que publica *tweets* con las llamadas realizadas recientemente (?).

Hay evidencias científicas de que la colaboración de un gran número de individuos es capaz de proporcionar respuestas más precisas a problemas complejos que la de un sólo individuo (?). Este fenómeno ha despertado el interés de la comunidad científica y hay múltiples trabajos que lo relacionan con Internet (??????).

En esta sección se han presentado trabajos en los que se relacionan las redes sociales y el Internet de las Cosas, y también se ha expuesto el principio de sabiduría de los grupos (*The Wisdom Crowds*), publicado por Surowiecki (?).

En esta tesis doctoral, aunando estas dos vertientes, se propone una colaboración entre objetos de Internet de las Cosas a través de las redes sociales para resolver con mayor eficacia el problema de arranque en frío⁴⁴ en sistemas de aprendizaje (*Cold Start problem*). De esta forma se pretende demostrar que si los electrodomésticos comparten datos relativos al uso que se hace de ellos con otros dispositivos del mismo tipo, la eficiencia energética de los mismos sería mayor, debido a que el período que los electrodomésticos necesitarían para poder operar de forma efi-

⁴²<https://es.wikipedia.org/wiki/Gadget>

⁴³<http://latimesblogs.latimes.com/technology/2009/01/twitter-washing.html>

⁴⁴https://es.wikipedia.org/wiki/Arranque_en_frio

ciente, se podría reducir si los electrodomésticos disponen inicialmente de un conjunto de datos adecuado.

A diferencia de los trabajos presentados por Pérez de Almeida et al. (?) y Ciorte et al. (?), y con el objetivo de aprovechar las infraestructuras ya existentes y en explotación, se pretende utilizar una red social ya en funcionamiento que se adapte a las necesidades de la propuesta presentada.

Dado que el propósito final de la propuesta de esta tesis doctoral es el ahorro de energía y pese a no abordar esta línea de investigación en esta tesis doctoral, una de las características que se considera deberían tener los objetos eco-aware es la capacidad de colaboración con las personas (en forma de consejos de uso, advertencias, etc.), además de hacerlo con otros dispositivos. Por lo tanto, se ha orientado la solución a utilizar una red social muy popular entre los seres humanos y registrar los objetos en la misma.

Twitter⁴⁵ es una de las redes sociales más influyentes del mundo. Con más de 300 millones de usuarios y alrededor de 500 millones de tweets al día⁴⁶, Twitter es el líder indiscutible de los sistemas de microblogging⁴⁷. La comunidad investigadora ha explotado esta red social con diversos propósitos, como la predicción de tendencias (?), detección de incidencias (?) o la influencia de los usuarios (?).

Tras observar las funcionalidades de Twitter, se considera que puede adaptarse a nuestras necesidades de forma satisfactoria. Éstas se especifican a continuación:

- Tener una identidad con un perfil asociado: es inmediato al registrar un usuario.
- Encontrar objetos del mismo tipo (p. ej., luminarias): Twitter permite crear listas de usuarios, por lo que se podrían crear listas de diferentes tipos de objetos (p. ej., luminarias, cafeteras o impreso-

⁴⁵<https://twitter.com/>

⁴⁶<https://about.twitter.com/es/company>

⁴⁷<https://es.wikipedia.org/wiki/Microblogging>

ras) e interactuarían entre sí sólo los dispositivos pertenecientes a una misma lista.

- Enviar información privada entre objetos: a través de mensajes privados.

Otras redes sociales, por ejemplo Facebook, también podrían cubrir las necesidades de la propuesta de esta tesis doctoral, pero Twitter es más abierta y ofrece un proceso de búsqueda e interacción más sencillo. En la propia web de Twitter hay un apartado⁴⁸ con múltiples librerías para comunicarse con esta red social.

2.6 Algoritmos de similitud de perfiles

En la literatura se encuentran una gran variedad de métodos para calcular la similitud, disimilitud o matriz de distancias⁴⁹ entre individuos de una población (???). Según Legendre et al. (?), en sus inicios los coeficientes de similitud se desarrollaron para datos binarios y más tarde se generalizaron a variables multiestado siempre y cuando su cálculo lo permite.

Clifford et al. (?) señalan que diferentes circunstancias han conducido al desarrollo de una gran variedad de coeficientes de similitud. A continuación se presentan brevemente varios de ellos.

⁴⁸<https://dev.twitter.com/overview/api/twitter-libraries>

⁴⁹https://en.wikipedia.org/wiki/Distance_matrix

2.6.1 Coeficientes de similitud para datos binarios

Legendre et al. (?) sostienen que de forma simple, la similitud de dos muestras se fundamenta en la presencia o ausencia de características en las mismas. Por lo tanto, existen cuatro posibilidades en la comparación entre dos muestras A y B:

- **a** es el número de variables para el cual las dos muestras tienen el valor 1.
- **b** es el número de variables para el cual la muestra A tiene el valor 1 y la muestra B tiene el valor 0.
- **c** es el número de variables para el cual la muestra A tiene el valor 0 y la muestra B tiene el valor 1.
- **d** es el número de variables para el cual las dos muestras tienen el valor 0.

El número total de variables adquiere el valor n .

Coeficiente de Igualdad Simple

Según Legendre et al. (?), un método sencillo para establecer la similitud entre dos muestras es dividir el sumatorio del número de variables que contienen en común entre el número total de variables.

$$S_1(x_A, x_B) = \frac{a + d}{n} = \frac{a + d}{a + b + c + d} \quad (2.1)$$

Donde x_A = muestra A y x_B = muestra B.

Coeficiente de Rogers y Tanimoto

El coeficiente de Rogers y Tanimoto (?) es una variante del Coeficiente de Igualdad Simple, en la cual las diferencias son más importantes que las semejanzas.

$$S_2(x_A, x_B) = \frac{a + d}{a + 2b + 2c + d} \quad (2.2)$$

Este índice también produce valores entre el cero y el uno.

Coeficiente de Sokal y Sneath

Sokal et al. (?) proponen, entre otros, tres nuevos índices que toman en cuenta los dobles ceros.

$$S_3(x_A, x_B) = \frac{2a + 2d}{2a + b + c + 2d} \quad (2.3)$$

En esta medida las presencias o ausencias conjuntas tienen un doble valor. El índice proporciona valores del cero al uno.

El siguiente índice pondera doblemente a las discordancias y no se tienen en cuenta las ausencias conjuntas.

$$S_4(x_A, x_B) = \frac{a}{a + 2b + 2c} \quad (2.4)$$

Los valores que proporciona son entre 0 y 1.

En el siguiente índice se comparan las concordancias con las diferencias en una medida que varía entre el 0 y el infinito.

$$S_5(x_A, x_B) = \frac{a + d}{b + c} \quad (2.5)$$

Coeficiente de Russel y Rao

Russel et al. (?) propusieron un índice que toma en cuenta las concordancias y las divide entre el número total de ausencias y concurrencias. Se trata de la medida por defecto para los datos de similaridad binarios.

$$S_6(x_A, x_B) = \frac{a}{a + b + c + d} \quad (2.6)$$

Este índice proporciona valores de 0 a 1. A tener en cuenta es la situación cuando se mide la similitud entre dos muestras iguales, ya que el valor no será 1, a menos que $d = 0$.

Coeficiente de Hamann

Janson et al. (?) afirman que este coeficiente fue propuesto por Hamann en 1961. Es muy parecido al Coeficiente de Igualdad Simple, salvo que en el numerador se hace la diferencia entre ocurrencias y ausencias.

$$S_7(x_A, x_B) = \frac{(a + d) - (b + c)}{a + b + c + d} \quad (2.7)$$

Este índice produce valores que oscilan entre -1 y 1.

Coeficiente de Jaccard

El coeficiente de Jaccard (??) es el índice binario más conocido que excluye las ausencias conjuntas. También se conoce como *razón de similitud*.

$$S_8(x_A, x_B) = \frac{a}{a + b + c} \quad (2.8)$$

Este índice pondera de igual modo a las concordancias y a las discordancias, pero omite las ausencias conjuntas. Pondera de igual modo a todos los términos y su valor oscila entre 0 y 1.

Coeficiente de Dice

También conocido como medida de Czekanowski (?) o de Sørensen (?), el coeficiente de Dice (?) es una variante del índice de Jaccard en el que las coincidencias tienen un doble valor.

$$S_9(x_A, x_B) = \frac{2a}{2a + b + c} \quad (2.9)$$

Se considera que la presencia de una característica proporciona mayor información que su ausencia. Provee valores que oscilan entre 0 y 1.

Coeficiente de similitud

Este coeficiente es una variante de los anteriores que pondera triplemente la doble presencia.

$$S_{10}(x_A, x_B) = \frac{3a}{3a + b + c} \quad (2.10)$$

Coeficiente de Kulczynski (1927)

Según indican Legendre et al.(?), Kulczynski propone un coeficiente en el que se divide el número de presencias conjuntas por todas las discordancias.

$$S_{11}(x_A, x_B) = \frac{a}{b + c} \quad (2.11)$$

Su valor oscila entre el 0 y el infinito.

Coeficiente de Kulczynski (1928)

Este segundo índice propuesto por Kulczynski está basado en la probabilidad condicional de que una característica esté presente en una muestra, siempre que esté presente en otra. Para calcularlo se hace el promedio de los distintos valores para cada elemento que actúa como predictor del otro.

$$S_{12}(x_A, x_B) = \frac{a}{2} \left(\frac{1}{a+b} + \frac{1}{a+c} \right) \quad (2.12)$$

Este coeficiente, a diferencia del anterior, presenta un valor de 0 a 1.

Coeficiente de Ochiai

Ochiai (?) utiliza como medida de similitud el cociente de a entre la media geométrica de $(a+b)$ y $(a+c)$.

$$S_{13}(x_A, x_B) = \frac{a}{\sqrt{(a+b)(a+c)}} \quad (2.13)$$

Este coeficiente produce valores que oscilan entre cero y uno.

2.6.2 Coeficientes de similitud para valores cuantitativos

A menudo, los atributos presentan más de dos estados y Legendre et al. (?) afirman que el uso de muchos de los coeficientes binarios puede extenderse a variables multiestado. Sin embargo, el uso de dichos coeficientes suele provocar una pérdida de información.

Coeficiente de similitud simple

Legendre et al. (?) señalan que el Coeficiente de Similitud Simple se utiliza del siguiente modo:

$$S_{14}(x_A, x_B) = \frac{\text{conurrencia}}{n} \quad (2.14)$$

Donde la *conurrencia* corresponde al número de características para las cuales ambas muestras presentan el mismo valor. A continuación, la tabla 2.5 presenta dos muestras descritas por variables multiestado y el cálculo de la *conurrencia* correspondiente.

	Descriptores (variables)										
Muestra A	7	4	1	5	5	7	3	2	0	1	
Muestra B	2	4	5	5	3	7	1	8	0	1	
<i>conurrencia</i>	0	1	0	1	0	1	0	0	1	1	=5

Tabla 2.5: Dos muestras descritas por variables multiestado.

El coeficiente de similitud que se obtendría en este caso sería el siguiente:

$$S(x_A, x_B) = \frac{\text{conurrencia}}{n} = \frac{5}{10} = 0,5 \quad (2.15)$$

Coeficiente de Gower

El coeficiente de Gower (?) es un índice de similitud que permite la combinación de diferentes tipos de variables. Para ello las variables deben haber sido procesadas previamente. Para datos binarios cualitativos o semicuantitativos el índice queda de la siguiente forma:

$$S_{15}(x_A, x_B) = \frac{1}{n} \sum_{i=1}^n s_{ABi} \quad (2.16)$$

Donde: $s_i = 1$, cuando se presenta una concurrencia en los datos.
 $s_i = 0$, cuando no se presenta una concurrencia en los datos.
 n = número de variables.

Por lo tanto, la similitud entre dos muestras es la media de los valores de similitud para las n variables. En el caso de dobles ausencias, Gower recomienda utilizar $s_i = 0$, sin descartar el uso del valor $s_i = 1$ en función de si se desea o no admitir las dobles ausencias como una base válida de semejanza entre muestras.

En cambio, las variables cuantitativas se manejan del siguiente modo:

$$s_{ABi} = 1 - \frac{|y_{Ai} - y_{Bi}|}{R_i} \quad (2.17)$$

Donde: y_{Ai} = valor de la variable i para la muestra A.
 y_{Bi} = valor de la variable i para la muestra B.
 R_i = desviación máxima de la variable i .

Además, al coeficiente calculado en S_{15} se le añade la delta de Kronecker (δ_{AB}) que se utiliza para ponderar las variables de las muestras. De esta forma, el coeficiente queda de la siguiente manera:

$$S_{16}(x_A, x_B) = \frac{\sum_{i=1}^n \delta_{ABi} s_{ABi}}{\sum_{i=1}^n \delta_{ABi}} \quad (2.18)$$

Donde: $\delta_{ABi} = 1$, cuando se tienen los valores de la variable i para ambas muestras. Si se especifica el peso de dicha variable i , δ_{ABi} adquiere el valor de su peso.
 $\delta_{ABi} = 0$, cuando no se tienen los valores de la variable i para cualquiera de las dos muestras.

Coficiente de Estabrook y Rogers

De análogas características al coeficiente de Gower es el de Estabrook y Rogers (?).

$$S_{17}(x_A, x_B) = \frac{\sum_{i=1}^n \delta_{ABi} s'_{ABi}}{\sum_{i=1}^n w_{ABi}} \quad (2.19)$$

Sin embargo, la medida de similitud que proponen para cada variable es diferente.

$$s_{ABi}^1 = f(d_{ABi}, k_i) = \frac{2(k+1-d)}{2k+2+dk} \quad \text{cuando } d \leq k$$

$$= 0 \quad \text{cuando } d > k$$

Donde d es la distancia entre los valores de las dos muestras x_A y x_B para la variable i , es decir $|y_{Ai} - y_{Bi}|$; y k es un parámetro establecido por el usuario para cada variable.

Coefficiente de similitud

Este coeficiente fue utilizado por Moreay et al. (?).

$$S_{18}(x_A, x_B) = 1 - \left(\frac{1}{n}\right) \sum_{i=1}^n \left(\frac{|y_{Ai} - y_{Bi}|}{y_{Ai} + y_{Bi}}\right) \quad (2.20)$$

Donde: y_{Ai} = valor de la variable i en la muestra A.
 y_{Bi} = valor de la variable i en la muestra B.
 n = número total de variables.

2.6.3 Análisis y conclusiones

Una vez presentado un amplio número de métodos de similitud, se deduce que no en todas las ocasiones el método más adecuado va a ser el mismo. Básicamente se debe considerar la variedad de tipos de datos que se dispone: binarios, cuantitativos y/o cualitativos.

Por ejemplo, si todas las variables son de tipo binario, existen múltiples coeficientes de similitud (?) tales como el índice de Jaccard, que podría ofrecer buenos resultados dependiendo del número de variables. Mientras que si todas las variables son de tipo cualitativo, el procedimiento debe ser diferente. En este caso, Everitt et al. (?) recomiendan seguir un proceso de conversión de las variables cualitativas en binarias. De este modo, el problema se convierte en un ejercicio de similitud binaria.

Sin embargo, cuando se manejan datos de tipo binario, cuantitativo y cualitativo, se debe optar por otros procedimientos. Gordon (?) sugiere convertir las variables al mismo tipo de datos (binario o cuantitativo). Una vez unificados los tipos de datos, el uso de los métodos de similitud es más sencillo.

Actualmente, el coeficiente general de similitud de Gower (?) es uno de los métodos más populares para calcular la similitud entre dos muestras con varios tipos de datos. Lim et al. (?) argumentan que, debido a sus propiedades, el coeficiente de Gower para evaluar la similitud entre dos muestras es uno de los métodos de análisis con un mayor potencial.

En un contexto diferente al de esta tesis, pero con la misma problemática si se reduce a meros datos, Fontecha (?) aplica con éxito algoritmos de similitud basados en el coeficiente de Gower para evaluar la similitud entre personas mayores. Las variables que utiliza son las que influyen en la evaluación de la fragilidad, que son de tipo cualitativo, cuantitativo y binario.

Por tanto, tras esta fase exploratoria se considera que el coeficiente de Gower puede determinar con acierto el grado de similitud entre dos dispositivos representados por datos binarios (p. ej., la variable que determina si el dispositivo está en un lugar dedicado para él), cualitativos (p. ej., el tipo de espacio donde está ubicado el dispositivo) y cuantitativos (p. ej., el número de personas que utiliza el dispositivo) en sus perfiles.

2.7 Métodos de aprendizaje automático

En esta subsección se presentan algunos métodos de aprendizaje automático y predicción más relevantes que podrían aplicarse en esta tesis doctoral para realizar predicciones de uso de los dispositivos con la mayor precisión posible.

2.7.1 Clasificador Naïve Bayes

El clasificador Naïve Bayes (?) es el modelo de red bayesiana más sencillo y uno de los algoritmos de aprendizaje automático y minería de datos más efectivo. Asume que todos los atributos son independientes unos de otros, de modo que todos contribuyen de forma independiente al valor de la probabilidad. A pesar de ser un algoritmo sencillo, se ha comportado bien en situaciones complejas. Zhang (?) presenta razones teóricas que fundamentan la eficacia aparentemente irracional de los clasificadores Naïve Bayes. Sin embargo, Caruana et al. (?), en una investigación posterior, realizaron una comparación de varios métodos de clasificación mostrando que el clasificador de Bayes es superado por enfoques más actuales.

Hay dos condiciones principales que se tienen que cumplir para aplicar este método:

- Disponer de un gran conjunto de datos.
- Los atributos deben ser independientes.

2.7.2 Redes neuronales artificiales

Las redes neuronales artificiales (RNA), dentro del campo del aprendizaje automático, son una familia de modelos de aprendizaje estadísticos inspirados en las redes neuronales biológicas. McCulloch et al. (?) desarrollaron el primer modelo de red neuronal, un modelo binario que ha servido de base para modelos posteriores (???).

Una red neuronal es un grupo de nodos interconectados donde cada uno recibe una serie de entradas y emite una salida. Usando la analogía con las redes neuronales biológicas, los nodos que componen la RNA y las conexiones unidireccionales que las comunican se denominan neuronas y sinapsis respectivamente. Las neuronas se organizan en capas que generalmente están totalmente conectadas por sinapsis. Cada una de las sinapsis tiene un peso asociado, que se va modificando durante proceso

de aprendizaje y pondera el efecto de la neurona correspondiente sobre el modelo. Los datos se propagan a través de la red neuronal en forma de señales y la salida de las neuronas viene dada por las siguientes funciones (ver figura 2.19):

- *Función de propagación o de red*: generalmente consiste en el sumatorio de cada entrada multiplicado por el peso de su interconexión.
- *Función de activación*: define la salida de un nodo dada una entrada o un conjunto de entradas. Algunas de las más frecuentes son la función *sigmoidea*⁵⁰, para obtener valores entre 0 y 1, o la tangente hiperbólica⁵¹, para obtener valores entre -1 y 1.

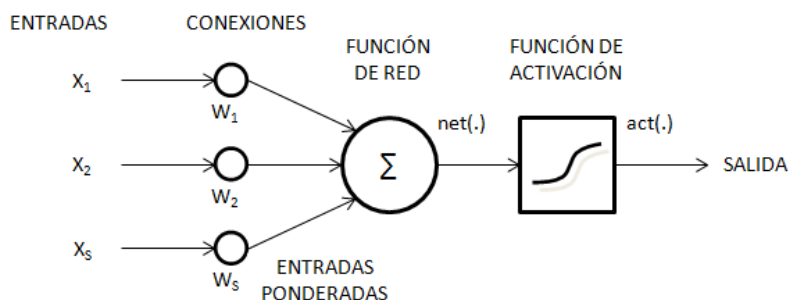


Figura 2.19: Modelo de neurona artificial.

El modelo general de una red neuronal frecuentemente tiene tres capas (ver figura 2.20), de las que una de ellas es la capa oculta, con una única salida. Se representa del siguiente modo:

$$o(x) = f\left(w_0 + \sum_{j=1}^q w_j \cdot f\left(w_{0j} + \sum_{i=1}^n w_{ij} x_i\right)\right) \quad (2.21)$$

⁵⁰https://en.wikipedia.org/wiki/Sigmoid_function

⁵¹https://en.wikipedia.org/wiki/Hyperbolic_function

Donde: w_{ij} ($i=1,2,\dots,n$; $j=1,2,\dots,q$) y w_j son los pesos de las sinapsis.
 n es el número de nodos de entrada.
 q es el número de nodos ocultos.
 f es la función de activación.

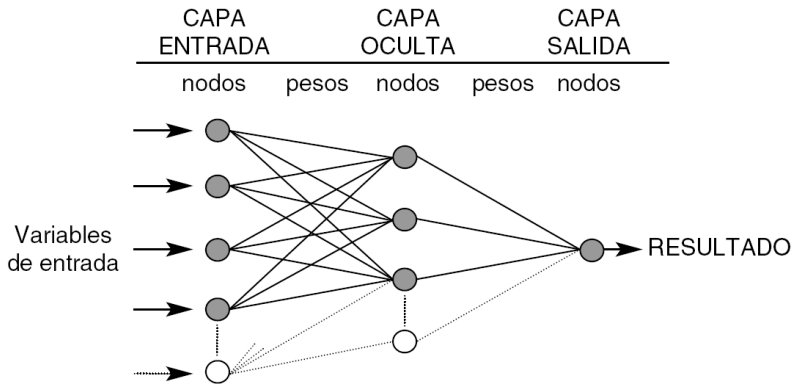


Figura 2.20: Modelo de red neuronal artificial.

La definición de un modelo predictivo basado en redes neuronales consiste en determinar, a través del uso de algoritmos de aprendizaje, los pesos de las sinapsis que proporcionan el mejor ajuste de los datos reales (?).

Hay dos fases en el proceso de modelización con redes neuronales:

- *Fase de entrenamiento:* en esta fase se usa un conjunto de datos de entrenamiento para determinar los pesos que definen el modelo de red neuronal.
- *Fase de prueba:* para evitar que el modelo se ajuste en demasía a las particularidades del conjunto de datos de entrenamiento, es aconsejable utilizar otro grupo de datos de validación para observar como evoluciona el proceso de aprendizaje.

2.7.3 Series temporales - Modelo ARIMA

Una serie temporal es un conjunto de datos, observaciones o valores ($y_1, y_2, \dots, y_t$) medidos en instantes de tiempo determinados (t) y ordenados cronológicamente. Estas observaciones pueden estar espaciadas a intervalos iguales de tiempo (periodo – horas, días, meses, años...) o desiguales. La base de las series temporales es que asume que patrones que se han dado en el pasado se volverán a dar de nuevo en un futuro, y por lo tanto son predecibles (?). Los patrones se corresponden con el valor que una variable va adquiriendo a lo largo de una secuencia de tiempo.

Hay diferentes modelos de predicción de series temporales, tales como Holt-Winters o el modelo autorregresivo integrado de media móvil ARIMA. Nau (?) indica que el uso de modelos ARIMA es adecuado cuando:

- (a) la serie de tiempo es estacionaria o puede llegar a serlo mediante diferenciaciones y transformaciones matemáticas, y
- (b) se dispone de una cantidad de datos suficiente, de al menos 50 observaciones.

Si, por el contrario, la serie temporal no se puede convertir en estacionaria por diferenciación (p. ej., es muy irregular o cambia su comportamiento a lo largo del tiempo) o no se dispone de datos suficientes, se recomienda utilizar otros métodos que hacen uso de técnicas de ajuste estacional y suavizado. Un ejemplo de este tipo de métodos es Holt-Winters (?).

Atendiendo a estos criterios, se decide utilizar el modelo ARIMA, ya que se cumplen las dos premisas necesarias para que su uso sea adecuado. Se dispone de datos suficientes y la serie se puede convertir a estacionaria mediante diferenciación (ver sección 4.5).

El modelo ARIMA fue introducido inicialmente por Box et al. (?) para pronosticar una serie temporal que puede convertirse en estacionaria por diferenciación. Esta conversión es imprescindible, ya que de no ser

así no se podría aplicar la metodología de Box-Jenkins (?) por ser una condición inecuánime la estacionalidad de la serie temporal, condición que raramente se cumple inicialmente.

El modelo ARIMA no estacionario se expresa como "*ARIMA*(p, d, q)", donde:

- p es la componente autorregresiva (AR), que expresa el número de observaciones retrasadas de la serie temporal analizada que intervienen en la ecuación.
- q es la media móvil (MA), que expresa el número de errores correspondientes a períodos precedentes, ponderados convenientemente.
- d es la integrada (I), que corresponde al número de diferencias necesarias para convertir la serie original en estacionaria.

La ecuación 2.22 muestra que el modelo ARIMA es una combinación lineal de polinomios de grado p y q respectivamente. Está basado en que las series temporales son una función de valores del pasado.

$$\phi_p(L)(1 - L)^d y_t = \theta_q(L)\varepsilon_t \quad (2.22)$$

Donde: y_t es la serie.

ε_t es el error.

L es el operador retardo, que aplicado al valor de una variable en t , da como resultado el valor de esa misma variable en $t - 1$.

ϕ_p es el polinomio autorregresivo (AR) de orden p .

θ_q es el polinomio de media móvil (MA) de orden q .

$(1 - L)^d$ es el número de diferencias necesarias para convertir la serie en estacionaria.

2.7.4 Análisis y conclusiones

Tras analizar brevemente algunos de los métodos de aprendizaje y predicción más utilizados, no parece evidente decantarse por uno, pese a que se tienen algunos indicios.

Adebiyi et al. (?) indican que las redes neuronales son el modelo de predicción más preciso y extendido en diferentes áreas, tales como los negocios, la ingeniería o las finanzas. Sin embargo, en el estudio que llevan a cabo los resultados que ofrecen ARIMA y RNA son similares.

Khashei et al. (?) afirman que los modelos ARIMA son relativamente más robustos y eficientes que modelos con estructuras más complejas en relación con la predicción a corto plazo. Además, Castro et al. (?) señalan que en general, el proceso de búsqueda de un modelo basado en redes neuronales es más complejo que el desarrollo de un modelo ARIMA, ya que implica calcular, a prueba y error, un mayor número de parámetros y la estabilización de la red neuronal requiere mayores recursos computacionales y de tiempo.

Respecto a la utilización de Naïve Bayes, Caruana et al. (?) realizó una comparación de varios métodos de clasificación mostrando que los resultados del clasificador de Bayes se encontraban entre los peores ofrecidos por diferentes algoritmos de aprendizaje supervisado, entre los que se encontraban las redes neuronales.

Ya que ARIMA y RNA parecían ofrecer resultados similares, en un trabajo previo (?) se realizó una comparación entre varios modelos de RNA con distintas configuraciones y un modelo ARIMA aplicados a la problemática de esta tesis doctoral. Pese a que en algún caso la predicción ofrecida por las redes neuronales era más precisa que la de ARIMA, en la mayoría de los casos se daba la situación contraria. Sin embargo, se constató que la precisión proporcionada era similar en todos los casos.

Si bien es cierto que, una vez definido el modelo, las RNA son más ligeras que ARIMA y, por tanto, más adecuadas para ejecutarse en microcomputadores, tales como Arduino, en esta tesis doctoral se ha optado por utilizar ARIMA como método de predicción. Esto se debe a que:

- los recursos computacionales no son críticos, ya que todos los procesos relativos al aprendizaje y predicción se van a ejecutar en un servidor y no van a requerir una alta computación. Además el tiempo de ejecución no es tampoco un punto crítico;
- el desarrollo de un modelo ARIMA es más sencillo. Pese a no ser crítico, requiere menos recursos computacionales y de tiempo que encontrar un modelo preciso con redes neuronales.

3

Un nuevo enfoque inteligente y colaborativo

Se me ocurrió cuando estaba andando en mi bici.
—ALBERT EINSTEIN, sobre la Teoría de la Relatividad.

3.1 Introducción

ESTE CAPÍTULO está dedicado a la descripción detallada y justificación de la infraestructura de objetos cotidianos, conformando uno de los capítulos principales de la presente tesis doctoral. Como prueba de concepto de estos dispositivos se ha escogido la máquina de café, un objeto cotidiano de uso compartido muy extendido en entornos laborales que dispone del modo *Standby*.

El capítulo se divide en cinco secciones. En primer lugar (sección 3.2), se presentan las bases teóricas sobre las que se sustenta la investigación. En segundo lugar (sección 3.3), se abordará el diseño conceptual de la arquitectura, profundizando en el diseño de los modos *Ecoaware* y *Cooperative*, siendo ambos aportaciones de esta tesis doctoral. En tercer lugar

(sección 3.4), se explica el modelo de series temporales utilizado para calcular la predicción de uso de los dispositivos. En la sección 3.5, se expone lo relacionado con el cálculo de similitud entre distintos perfiles de dispositivos, concretando el modelado de estos perfiles y el índice de similitud utilizado, éste muy similar al coeficiente de Gower. Por último, en la sección 3.6, se presenta un análisis de la propuesta y conclusiones extraídas de la misma.

3.2 Bases teóricas. De *On-Off* a *Standby*

Para demostrar de forma teórica que hay períodos propensos al uso del modo *Standby*, se han realizado los siguientes cálculos matemáticos.

En primer lugar se ha definido el período D de actividad laboral, de 7:00 h a 19:00 h. Éste se ha dividido en franjas de tiempo de T_{sl} segundos cada una. El objetivo es determinar el número mínimo de cafés (n), que se deben preparar en cada franja de tiempo para que merezca la pena el uso del modo *Standby* en vez del modo *On-Off*.

Para ello, tal y como se representa en la ecuación 3.1, se ha igualado la energía consumida a lo largo de una jornada laboral (D) si la cafetera sólo hubiera funcionado en modo *On-Off* ($E_{nf|T_{sl}}$) y si únicamente lo hubiera hecho en modo *Standby* ($E_{sb|T_{sl}}$).

$$E_{nf|T_{sl}} = E_{sb|T_{sl}} \quad (3.1)$$

La energía consumida en modo *On-Off*, que se calcula mediante la ecuación 3.2, sólo depende de n y el valor de T_{sl} no tiene influencia.

$$E_{nf} = n(\overline{T_{st}} + \overline{T_{cf}}) \quad (3.2)$$

Donde: T_{cf} es el valor medio de preparación de un café.

T_{st} es el tiempo medio de arranque de una máquina de café.

En cambio, la energía consumida en modo *Standby* se determina me-

diante la ecuación 3.3.

$$E_{sb|T_{sl}} = n(\overline{T_{cf}}) + (n_{peak} * \overline{T_{peak}}) \quad (3.3)$$

Donde: T_{cf} es el valor medio de preparación de un café.
 T_{peak} es el tiempo medio de los picos de consumo en modo *Standby*.
 n_{peak} es el número de picos de consumo que se producen en un período de tiempo cuando una cafetera se mantiene en modo *Standby*. Viene definido por la ecuación 3.4.

A diferencia de la ecuación 3.2, el cálculo de n_{peak} depende del valor de T_{sl} . Por lo tanto, el de $E_{sb|T_{sl}}$ también.

$$n_{peak} = \frac{T_{sl} - n(\overline{T_{acf}} + \overline{T_{cf}})}{\overline{T_{sb}}} + 1 \quad (3.4)$$

Donde: T_{sb} es el tiempo medio que discurre entre los picos de consumo en modo *Standby*.
 T_{acf} es el tiempo medio que necesita la cafetera para estabilizar su consumo tras la preparación de un café.
 T_{cf} es el valor medio de preparación de un café.

Haciendo uso de estas formulas, se está en disposición de extraer el valor del número mínimo de cafés n que se deben preparar en un período de tiempo T_{sl} para cambiar de un modo de funcionamiento *On-Off* al modo *Standby* como modo más eficiente de operación. Por ejemplo, si se fragmenta una jornada laboral en franjas de tiempo de 1 hora ($T_{sl} = 3600 \text{ seconds}$), el valor de n para una cafetera Dolce Gusto Melody asciende a 2,3 cafés. Como los cafés no son divisibles, se considera que se deberían hacer 3 cafés durante una hora para rentabilizar en términos energéticos el uso del modo *Standby* a lo largo de ese período temporal.

Es pertinente aclarar que los parámetros temporales T_{sb} , T_{cf} , T_{acf} , T_{peak} y T_{st} son cantidades escalares medidas en segundos y que se han

extraído tras un proceso de captura de datos de consumo energético para cada modelo de cafetera que forma parte del experimento.

3.3 Diseño conceptual

En esta sección se presenta el diseño conceptual de la arquitectura propuesta para habilitar el uso de los modos *Ecoaware* y *Cooperative* en las máquinas de café. Se identifican todos los elementos que forman parte de la arquitectura y se establecen las relaciones existentes entre ellos, así como las funcionalidades que desempeñan. Bajo un nivel de abstracción relativamente alto, se hace referencia a la estructura de la arquitectura y sus elementos, sin entrar en detalle en cómo éstos realizan cada una de las funcionalidades.

3.3.1 Diseño del modo *Ecoaware*

A continuación se va a describir la arquitectura que permite a una máquina de café común funcionar en un modo dinámico *Ecoaware*.



Figura 3.1: Arquitectura *Ecoaware*.

Observando la figura 3.1, que muestra una perspectiva general de la

arquitectura, se identifican los siguientes elementos:

1. Cafetera de cápsulas.
2. Módulo hardware.
3. Servidor cloud.
 - a) Base de datos.
 - b) Servicio de predicción de uso.

El módulo hardware (elemento 2 de la figura 3.1) es el encargado de detectar cuándo se produce un consumo energético, capturarlo, diferenciar de qué tipo es y posteriormente enviarlo a través de TCP/IP a un servidor (elemento 3 de la figura 3.1) para su almacenamiento en una base de datos (elemento 3a de la figura 3.1).

En el diagrama de secuencia de la figura 3.2 se puede observar de forma gráfica la interacción entre las distintas entidades mencionadas para el proceso de captura de datos y su posterior almacenamiento.

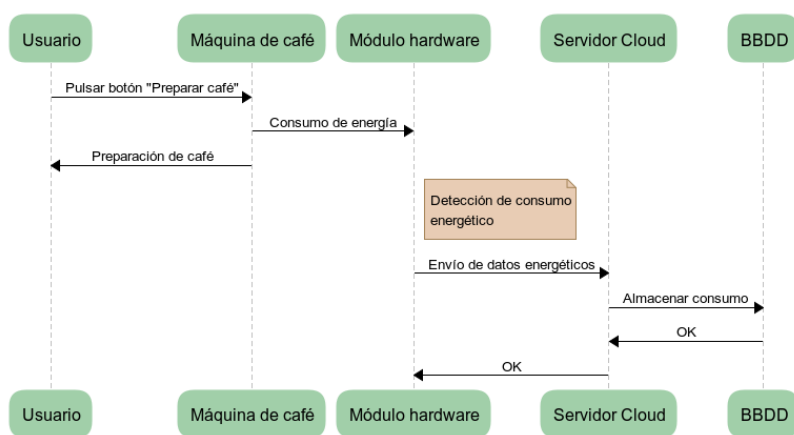


Figura 3.2: Diagrama de secuencia del proceso de captura de datos energéticos.

Además de disponer de la funcionalidad de captura y envío de datos energéticos, el módulo hardware es el encargado de gestionar los modos

de funcionamiento en función de una predicción de uso calculada en el servidor. Este proceso de predicción (elemento 3b de la figura 3.1) se alimenta con los datos de uso almacenados en la base de datos energéticos ubicada en el servidor.

La figura 3.3 presenta, mediante un diagrama de secuencia, el proceso de actualización de la predicción de uso de la cafetera. Como se puede observar, éste se da de forma periódica semanal.

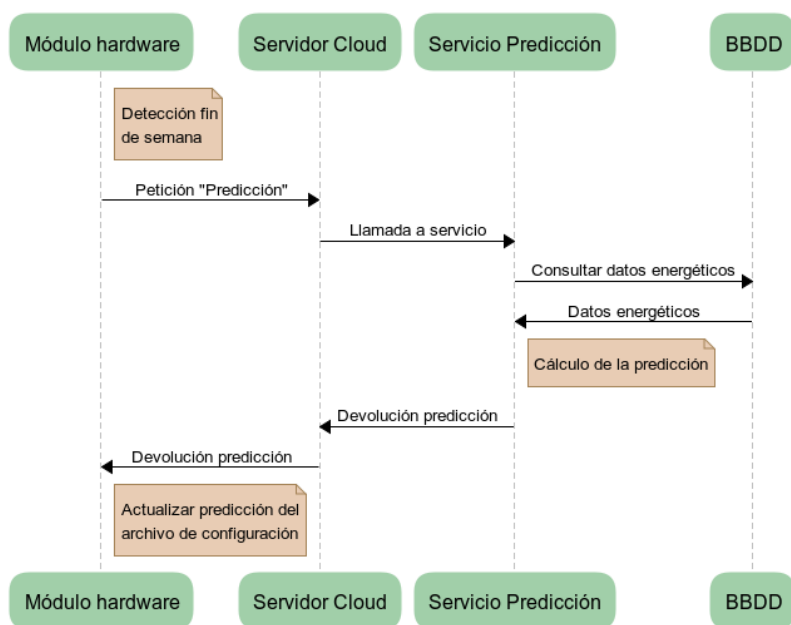


Figura 3.3: Diagrama de secuencia del proceso de actualización de la predicción de uso.

Tras este proceso, el módulo hardware dispone de la información necesaria para decidir en qué franjas de tiempo a lo largo de la semana debe apagar automáticamente la cafetera tras la preparación de un café o debe mantenerla en modo *Standby*.

3.3.2 Diseño del modo *Cooperative*

En este apartado se va a describir la arquitectura que permite a una máquina de café común funcionar en un modo dinámico *Cooperative*. Este modo parte de la premisa de que los dispositivos son capaces de funcionar en modo *Ecoaware* y nace con el objetivo de aliviar el problema de arranque en frío del recién mencionado modo *Ecoaware*. Inicialmente un dispositivo no dispone de datos previos de uso, por lo que no existe la posibilidad de prever un patrón de uso del mismo. Por tanto, se puede deducir que inicialmente se hace imprescindible un período de aprendizaje, en el cual el dispositivo no está habilitado para funcionar en modo *Ecoaware*.

A esta debilidad del modo *Ecoaware* es a la que el modo *Cooperative* pretende hacer frente. Para ello, se propone que una cafetera (de color blanco en la figura 3.4) que se enciende por primera vez adquiera los datos de uso de otra cafetera previamente ya en funcionamiento (de color rojo en la figura 3.4), para así poder realizar una predicción de uso y funcionar en base a la información de consumo de otra máquina de café ubicada en un entorno compartido similar.

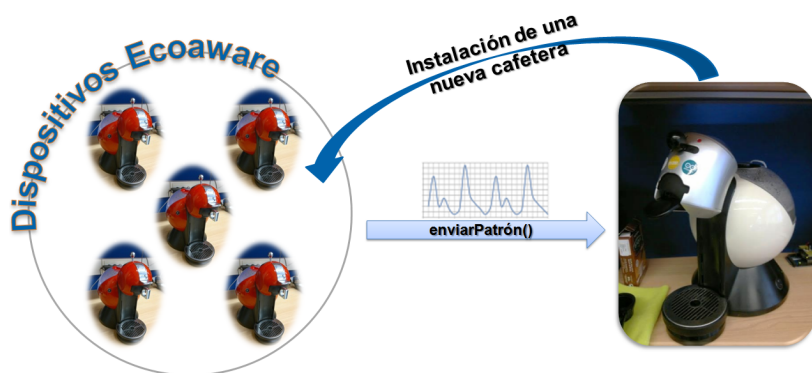


Figura 3.4: Arquitectura *Cooperative*.

La principal cuestión a resolver es cómo decidir los datos de qué dispositivos son válidos para un funcionamiento energéticamente eficiente del nuevo dispositivo y cuáles no lo son. Esta se aborda en la sección 3.5.

Una vez que se dispone de un dispositivo (*B* en la figura 3.5) candidato a compartir su patrón de uso con el nuevo dispositivo que se va a poner en marcha (*A* en la figura 3.5), este último realiza una predicción de uso para la primera semana con los datos del dispositivo *B*. Tras adaptar el dispositivo *A* su funcionamiento a la predicción calculada durante los primeros 7 días, realiza una nueva predicción haciendo uso de datos de *B* y los datos capturados por ella durante la primera semana de funcionamiento. Y así sucesivamente durante las primeras 4 semanas (ver figura 3.5). A partir de la quinta semana, la predicción será calculada en base a datos capturados por el propio dispositivo durante el mes anterior de funcionamiento.

<i>A</i> _{1stWeek}	ND	ND	ND	ND	ND	ND	ND	ND	ND	ND	ND
<i>B</i> _{1stWeek}	1	2	3	...	...	18	19	20	ND	ND	ND
<i>A</i> _{2ndWeek}	ND	ND	ND	ND	ND	ND	21	22	23	24	25
<i>B</i> _{2ndWeek}	...	6	7	...	19	20	21	22	23	24	25
<i>A</i> _{3rdWeek}	ND	ND	ND	ND	ND	21	22	...	...	29	30
<i>B</i> _{3rdWeek}	...	11	12	...	20	21	22	...	...	29	30
<i>A</i> _{4thWeek}	ND	ND	ND	ND	ND	21	22	...	...	34	35
<i>B</i> _{4thWeek}	...	16	17	...	20	21	22	...	...	34	35

Figura 3.5: Origen de los datos de entrenamiento para una cafetera nueva (*A*) durante las primeras 4 semanas. (ND = No hay datos)

La figura 3.6 muestra el diagrama de secuencia de todo el proceso de compartición de datos entre dispositivos distintos, pero del mismo tipo al fin y al cabo (p. ej., cafeteras), donde el dispositivo A, al igual que en la figura anterior, se corresponde con el dispositivo nuevo a instalar.

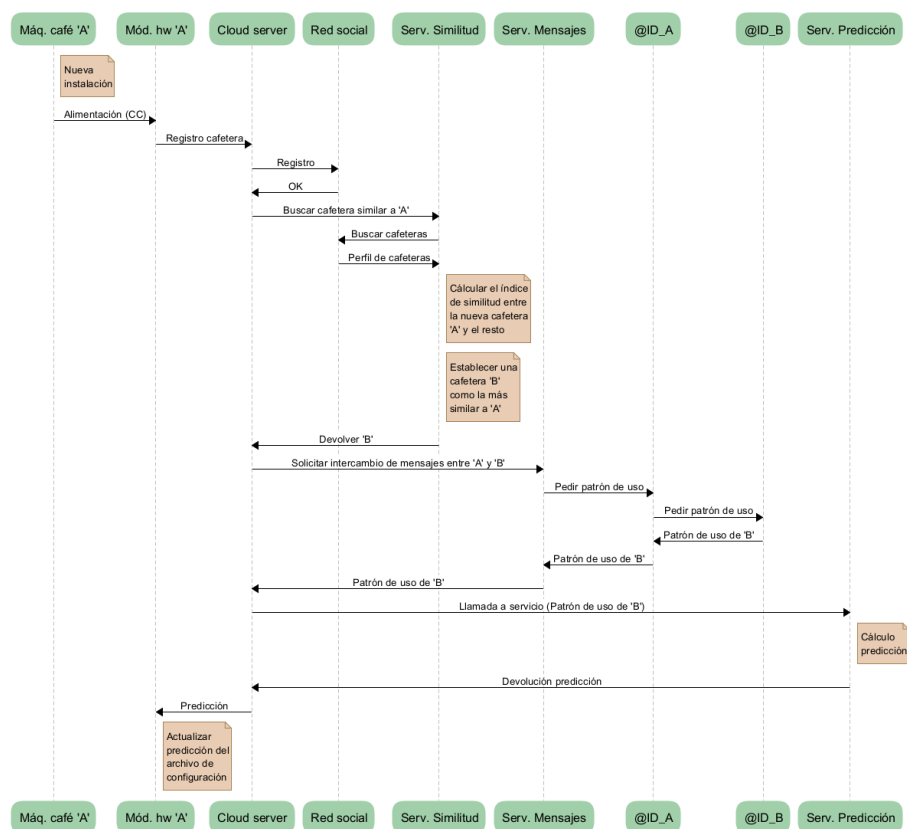


Figura 3.6: Diagrama de secuencia del proceso de compartición de datos de uso.

3.4 Cálculo de la predicción de uso

Tal y como se ha explicado en la sección 2.7, se ha optado por utilizar ARIMA como método de predicción de uso de los dispositivos.

Antiguamente, las series temporales se analizaban desde un punto de vista determinista. Sin embargo, actualmente se estudian desde un punto de vista estocástico, haciendo uso de métodos más complejos y series más largas.

Box et al. (?) desarrollaron modelos estadísticos para series temporales en los que cada observación en un instante de tiempo es modelada en función de valores anteriores. Estos modelos se conocen con el nombre de ARIMA (*AutoRegresive Integrated Moving Average*), que proviene de sus tres componentes: AR (Autoregresivo), I(Integrado) y MA (Medias Móviles).

Mediante el modelo ARIMA se puede describir un valor como una función lineal de datos previos y errores. Además, puede incluir un componente cíclico o estacional, por lo que es necesario un número de datos suficiente para poder describir este fenómeno. Box et al.(?) recomiendan al menos 50 observaciones en la serie temporal.

La metodología de Box y Jenkins, representada en la figura 3.7, se divide en cuatro fases.

- **1ª fase:** identificar el modelo ARIMA que sigue la serie temporal, lo que requiere:
 - Aplicar transformaciones para convertir la serie en una serie estacionaria.
 - Determinar un modelo ARMA para la serie estacionaria, estableciendo los órdenes p y q de su estructura autorregresiva y de media móvil.
- **2ª fase:** se estiman los parámetros AR y MA del modelo y se obtienen sus errores estándar y los residuos del modelo.
- **3ª fase:** es el diagnóstico. Se comprueba que los residuos no tienen

estructura de dependencia y siguen un proceso de ruido blanco. Si tienen estructura de dependencia, se modifica el modelo y se repiten las etapas anteriores hasta obtener un modelo adecuado.

- **4ª fase:** es la predicción. Una vez que se ha obtenido un modelo adecuado se realizan predicciones con el mismo.

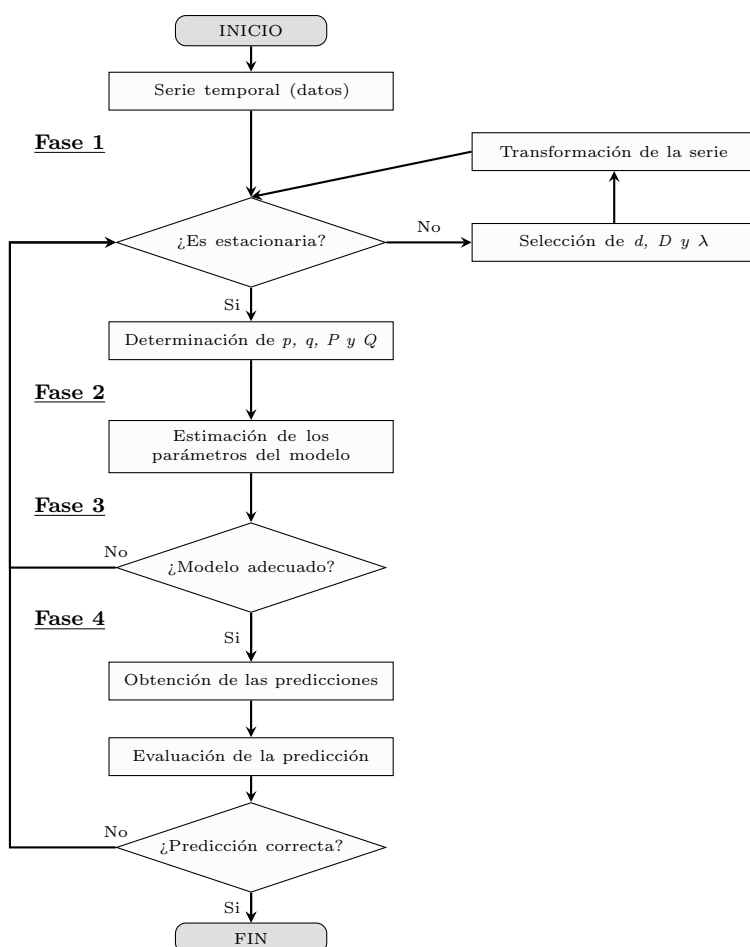


Figura 3.7: Metodología de Box-Jenkins.

Pasos a seguir para el análisis de datos

A continuación, se enumeran los pasos a seguir para el análisis de los datos de uso capturados.

1. **Captura de datos:** se recomienda disponer de al menos 50 datos.
2. **Representación gráfica:** es recomendable disponer de un gráfico de la serie de tiempo para comprobar su estacionariedad. Además, se pueden utilizar medias y desviaciones típicas por subperíodo para dicha comprobación.
3. **Transformación previa de la serie:** si la serie de tiempo no es estacionaria en varianza, es necesario realizar una transformación logarítmica.
4. **Eliminación de la tendencia:** analizando el gráfico de la serie se observa si existe o no tendencia. De existir una tendencia lineal, que será el caso más frecuente, se corregirá diferenciando una vez la serie. De ser una tendencia no lineal, se suelen utilizar dos diferencias como mucho.
5. **Identificación del modelo:** se trata de determinar el modelo más adecuado a la serie, es decir, el orden de los procesos autorregresivos de medias móviles de las componentes regular y estacional. Esta decisión se basa en las funciones de autocorrelación (ACF y PACF). En caso de duda, se pueden seleccionar varios modelos que serán contrastados posteriormente, para finalmente definir el modelo adecuado.
6. **Estimación de los coeficientes del modelo:** una vez se tiene el modelo, se estiman sus parámetros.
7. **Contraste de validez del modelo:** para comprobar si el modelo es adecuado, se pueden utilizar distintos procedimientos, tales como covarianzas entre estimadores, coeficiente de correlación, contraste de significación de parámetros o suma de cuadrados de errores entre otros.

8. **Análisis detallado de los errores:** se contrastan los valores reales con los estimados por el modelo. Hay que analizar si existen errores especialmente significativos.
9. **Selección del modelo:** en función de los resultados de los dos pasos anteriores, se decide sobre la validez del modelo adoptado.
10. **Predicción:** finalmente, el modelo seleccionado será el utilizado para realizar las predicciones.

En el siguiente capítulo se describen los pasos que se han ido dando para los datos de los que se dispone en esta tesis doctoral.

3.5 Cálculo de similitud entre perfiles

En esta sección se describe el servicio *Similitud* al que se hace referencia en la figura 3.6. Éste se ejecuta cada vez que se instala un nuevo dispositivo *Ecoaware* (A) con el objetivo de buscar otro dispositivo (B) del mismo tipo que disponga de datos de uso que sean adecuados para el entrenamiento del nuevo objeto. Para que estos datos puedan considerarse adecuados, deben tener cierto grado de semejanza con los que posteriormente generará el nuevo dispositivo, permitiendo así que éste pueda realizar predicciones de uso fiables desde el mismo momento en el que es instalado.

3.5.1 Modelado de los perfiles de los dispositivos

No todos los dispositivos van a tener un mismo modelo de perfil, ya que las características de cada tipo de objeto van a determinar su modelado. De este modo, una cafetera, una fotocopidora o un sistema de iluminación tendrán una estructura de perfil diferente en función de sus características. Como ya se ha indicado en la introducción de este capítulo, como prueba de concepto de estos dispositivos se ha escogido la máquina de café, por lo tanto en este apartado se presenta la estructura de su perfil.

Los datos que se integran dentro del modelo del perfil de una cafetera son los siguientes:

- *Identificador*: es el nombre que el responsable de la cafetera le asigna a la misma y debe ser único.
- *Número de personas*: se corresponde con el número de individuos que hacen uso de la máquina de café.
- *Lugar dedicado*: indica si hay o no un lugar dedicado para la ubicación de la cafetera (p. ej., cocina o sala de café).
- *Jornada*: hace referencia al número de horas que el dispositivo es susceptible de ser usado.
- *Horario*: indica el tipo de horario del lugar donde está ubicada la cafetera (continuo, partido, turnos)
- *Ubicación*: es el lugar geográfico donde está ubicada la máquina de café.
- *Tipo de espacio*: describe el espacio donde está ubicada la cafetera (p. ej., Centro de investigación, Administración o Comercio). Su valor se elige de una lista previamente definida.
- *Modelo de cafetera*: es el modelo de la máquina de café.

3.5.2 Coeficiente de similitud de wGower

Tal y como se ha explicado en la sección 2.6.3 se ha considerado que ante la diversidad de tipos de datos de los que dispone el perfil de un dispositivo (binarios, cuantitativos y cualitativos), el cálculo de similitud entre dispositivos puede llevarse a cabo mediante el coeficiente de Gower.

Atendiendo al apartado *Coeficiente de Gower* de la sección 2.6.3, el valor del coeficiente de similitud se obtiene aplicando la siguiente fórmula:

$$S(x_A, x_B) = \frac{\sum_{i=1}^n \delta_{ABi} s_{ABi}}{\sum_{i=1}^n \delta_{ABi}} \quad (3.5)$$

Donde para variables binarias y cualitativas:

- $s_{ABi} = 1$, cuando se presenta una concurrencia en los datos.
- $s_{ABi} = 0$, cuando no se presenta una concurrencia en los datos.
- n = número de variables.

Y para variables cuantitativas:

$$s_{ABi} = 1 - \frac{|y_{Ai} - y_{Bi}|}{R_i}$$

(3.6)

- y_{Ai} = valor de la variable i para la muestra A.
- y_{Bi} = valor de la variable i para la muestra B.
- R_i = rango de valores de la variable i .
- n = número de variables.

En la tabla 3.1 se observa que entre las variables cualitativas hay una que representa el tipo de espacio en el que está ubicado el dispositivo (p. ej., Universidad, Centro de idiomas o Comercio). Atendiendo a la ecuación 3.5, por ser una variable cualitativa, s_{esp} adquiriría el valor 1 cuando se presenta una concurrencia en los datos y 0 en caso contrario.

Binarias	Cualitativas	Cuantitativas
Lugar dedicado (<i>Sí / No</i>),	Horario (<i>Continua / Partida / turnos</i>), Ubicación, Tipo de espacio (<i>Centro de investigación / Administración / Universidad / Centro de idiomas / Bufete de abogados / Empresa tecnológica / Co-working</i>)	Número de personas, Jornada

Tabla 3.1: Variables del perfil de los dispositivos clasificadas por tipo de variable.

En esta tesis doctoral, se presenta una modificación del coeficiente de Gower, que se ha denominado coeficiente de *wGower* (*Weighted Gower*). En esta modificación las variables cualitativas se dividen en dos categorías. Por una parte las cualitativas tradicionales, cuya relación entre los diferentes valores que puede adquirir una de esas variables no existe. De ahí que s_{ABi} adquiera el valor 1 cuando se presenta una concurrencia en los datos y adquiera el valor 0 cuando no se presenta esta concurrencia. Y por otra parte están las cualitativas relacionales, en las que cada valor a adquirir por esas variables tienen un grado de similitud (φ) con el resto de esos valores. De este modo, se considera que si no se da una concurrencia en la variable, el valor de s_{ABi} será entre 0 y 1, dependiendo del grado de similitud (φ) entre los distintos valores que pueda adquirir dicha variable.

Así, la tabla 3.1 se transforma en la siguiente tabla:

Binarias	Cualitativas tradicionales	Cualitativas relacionales	Cuantitativas
Lugar dedicado (Sí / No),	Horario (<i>Continua</i> / <i>Partida</i> / <i>turnos</i>), Ubicación	Tipo de espacio (<i>Centro de investigación</i> / <i>Administración</i> / <i>Universidad</i> / <i>Centro de idiomas</i> / <i>Bufete de abogados</i> / <i>Empresa tecnológica</i> / <i>Co-working</i>)	Número de personas, Jornada

Tabla 3.2: Variables del perfil de los dispositivos reclasificadas por tipo de variable.

De manera que, según lo explicado, se considera que si no se da una concurrencia en la variable *Tipo de espacio*, el valor de s_{esp} será entre 0 y 1, dependiendo de los grados de similitud (φ) entre tipos de espacio que se hayan definido previamente (ver siguiente apartado). De este modo, si se busca el grado de similitud entre un perfil A, cuyo tipo de espacio es *Centro de investigación*, y un perfil B, cuyo tipo de espacio es *Empresa tecnológica*, el valor de s_{esp} será de 0.8 en vez de 0 si este es el valor que se ha definido previamente. Esto indicaría que previamente se ha establecido que los hábitos de consumo de café en una empresa tecnológica y en un centro de investigación tienen un alto grado de similitud (0.8).

Por lo tanto, aplicando el coeficiente de wGower a dos máquinas de café A y B, la fórmula matemática quedaría de la siguiente forma:

$$S(x_A, x_B) = \frac{\delta_{nPers} s_{ABnPers} + \delta_{dedEsp} s_{ABdedEsp} + \delta_{jorn} s_{ABjorn} + \delta_{horario} s_{ABhorario} + \delta_{ubic} s_{ABubic} + \delta_{esp} s_{ABesp}}{\delta_{nPers} + \delta_{dedEsp} + \delta_{jorn} + \delta_{horario} + \delta_{ubic} + \delta_{esp}} + \frac{\delta_{horar} s_{ABhorar} + \delta_{ubic} s_{ABubic} + \delta_{esp} s_{ABesp}}{\delta_{nPers} + \delta_{dedEsp} + \delta_{jorn} + \delta_{horar} + \delta_{ubic} + \delta_{esp}} \quad (3.7)$$

Donde:

$$\begin{aligned} s_{ABnPers} &= 1 - \frac{|y_{AnPers} - y_{BnPers}|}{R_{nPers}} \\ s_{ABdedEsp} &= 0 \text{ (si } y_{AdedEsp} \neq y_{BdedEsp}) \text{ ó } 1 \text{ (si } y_{AdedEsp} = y_{BdedEsp}) \\ s_{ABjorn} &= 1 - \frac{|y_{Ajorn} - y_{Bjorn}|}{R_{jorn}} \\ s_{ABhorar} &= 0 \text{ (si } y_{Ahorar} \neq y_{Bhorar}) \text{ ó } 1 \text{ (si } y_{Ahorar} = y_{Bhorar}) \\ s_{ABubic} &= 0 \text{ (si } y_{Aubic} \neq y_{Bubic}) \text{ ó } 1 \text{ (si } y_{Aubic} = y_{Bubic}) \\ s_{ABesp} &= \varphi_{AB} \text{ (si } y_{Aesp} \neq y_{Besp}) \text{ ó } 1 \text{ (si } y_{Aesp} = y_{Besp}) \end{aligned}$$

En la sección 4.4.2 del siguiente capítulo se presenta el cálculo de este coeficiente con datos de todos los dispositivos que forman parte del experimento.

Definición de grados de similitud (φ) entre tipos de espacio

El grado de similitud (φ) entre tipos de espacio, como se ha explicado ya, establece el grado de similitud que tienen los patrones de uso de dos dispositivos del mismo tipo ubicados en dichos tipos de espacio. El valor de esta medida fluctuará entre 0, si no hay similitud, y 1, si el patrón de uso de esos dispositivos es semejante.

Revisando la literatura se observa que existen varias categorías de tipos de trabajo (??). Sin embargo se echa en falta una de tipos de lugares de trabajo. En esta tesis doctoral se ha elaborado un listado de los tipos de lugares de trabajo que forman parte de la experimentación (ver tabla 3.3).

Lugar de trabajo	Abreviatura
Administración	ADM
Empresa tecnológica	TECN
Universidad	UNI
Centro de investigación	INV
Centro de idiomas	IDIOM
Bufete de abogados	ABOG
Startup	STARTUP

Tabla 3.3: Listado de espacios de trabajo.

A continuación, en la tabla 3.4, se presentan los grados de similitud entre tipos de espacio que se han ajustado de forma experimental tras observar la similitud de los patrones de uso de los dispositivos bajo experimento.

Tipos de espacio	ADM	TECN	UNI	INV	IDIOM	ABOG	STARTUP
ADM	1.0	0.2	0.6	0.8	0.2	0.2	0.2
TECN	0.2	1.0	0.6	0.6	0.2	0.6	0.8
UNI	0.6	0.6	1.0	0.4	0.8	0.2	0.4
INV	0.8	0.6	0.4	1.0	0.2	0.5	0.6
IDIOM	0.2	0.2	0.8	0.2	1.0	0.2	0.4
ABOG	0.2	0.6	0.2	0.5	0.2	1.0	0.4
STARTUP	0.2	0.8	0.4	0.6	0.4	0.4	1.0

Tabla 3.4: Grados de similitud entre los diferentes tipos de espacio.

Obtención de pesos

La función de los pesos es ponderar la importancia de cada variable durante el proceso de cálculo de similitud entre dos dispositivos. Inicialmente, con el objetivo de discernir qué variables tenían una mayor representatividad, se utilizaron técnicas como el análisis de componentes principales o la prueba χ^2 . Sin embargo, todas ellas consideraban que las variables cualitativas eran de tipo tradicional. Por lo tanto, se optó por hacer uso de la primera de estas técnicas sin tener en cuenta la variable cualitativa relacional *Tipo de espacio*, de forma que se obtuvieron unos pesos para el resto de variables eliminando el ruido de dicha variable. Partiendo de estos pesos iniciales y añadiendo la variable *Tipo de espacio*

y los grados de similitud entre tipos de espacio calculados en el apartado anterior, se han realizado varias iteraciones hasta ajustar los pesos de forma que el patrón recomendado a cada dispositivo ha sido adecuado para ahorrar energía desde la primera semana de funcionamiento de cada dispositivo. A continuación, en la tabla 3.5, se presentan los pesos resultantes.

Variable	Valor (δ)
nPers	0,90
dedEsp	0,05
jorn	0
horar	0
ubic	0,05
esp	0,40

Tabla 3.5: Pesos asignados a las variables del perfil.

Se puede observar que a algunas variables no se les ha concedido ningún peso. Esto se debe a que estas adquieren el mismo valor para las 11 cafeteras que forman parte del experimento principal de esta tesis doctoral, de modo que no son representativas.

Además de para el cálculo de s_{ABi} , los pesos son necesarios en el denominador de la fórmula de *wGower*, donde se debe proporcionar el sumatorio de todos ellos.

3.6 Análisis y conclusiones

En esta sección se han presentado las bases teóricas sobre las que se sustenta la investigación, el diseño conceptual de la arquitectura para cada uno de los modos de funcionamiento propuestos y los procedimientos para calcular la predicción de uso de los dispositivos y la similitud entre distintos perfiles de dispositivos.

Una limitación existente para la arquitectura cooperativa, en concreto en el proceso de cálculo de similitud entre perfiles, es la dificultad de encontrar un patrón adecuado entre un número de dispositivos no eleva-

do. Este es el caso de la experimentación de esta tesis doctoral, en la que se dispone de datos de 11 máquinas de café, ya que sería inviable en términos económicos y de mantenimiento instalar en cientos de cafeteras el módulo de captura de datos. Para una muestra mayor de dispositivos, se prevé que los pesos calculados y los grados de similitud entre tipos de espacio sufrirían variaciones para un tamaño de muestra mayor, y en este caso se considerarían valores representativos.

4

Implementación de la solución

La vida es como andar en bicicleta. Para mantener el equilibrio debes estar en movimiento.
—ALBERT EINSTEIN.

4.1 Introducción

EN ESTE CAPÍTULO se profundiza sobre los aspectos más interesantes relacionados con la implementación de los distintos componentes software y hardware de la infraestructura propuesta en esta tesis doctoral. En primer lugar, y previo al desarrollo e implementación del sistema, la sección 4.2 expone los requisitos funcionales y no funcionales de la propia infraestructura. Las secciones posteriores se corresponden con los diferentes módulos del sistema: el de captura de datos (sección 4.3), el de registro de los dispositivos (sección 4.4), el de compartición de datos entre dispositivos (sección 4.4.3) y el módulo de predicción.

4.2 Requisitos del sistema

El diseño y desarrollo de la infraestructura debe ir alineado con una serie de requisitos previamente especificados. Éstos se han dividido en requisitos funcionales (RF) y requisitos no funcionales (RNF).

A continuación se presentan los requisitos funcionales de la infraestructura.

- **RF1: Uso común.** Los usuarios de los dispositivos podrán utilizarlos del mismo modo que los utilizaban previamente.
- **RF2: Captura de datos energéticos.** Los dispositivos tendrán la capacidad de detectar sus propios consumos de energía y enviarlos a través de Internet para su posterior almacenamiento.
- **RF3: Registro de dispositivos.** El principal responsable del dispositivo podrá registrar su dispositivo en la nube.
- **RF4: Cálculo de similitud.** Los dispositivos podrán detectar otros ubicados en entornos similares de entre todos los registrados.
- **RF5: Compartición de datos.** Los dispositivos podrán compartir datos de uso de forma transparente.
- **RF6: Comportamiento dinámico.** Los dispositivos podrán funcionar en un modo dinámico dependiente de la predicción de uso que realizarán.

Por otra parte, se ha definido el siguiente requisito no funcional:

- **RNF1: Reducción del consumo energético.** Los dispositivos, aún sumando la energía que consume el hardware añadido, tendrán un consumo energético menor a antes de ser instrumentados.

4.3 Captura y almacenamiento de datos

Esta sección describe el módulo de captura y almacenamiento de datos, profundizando en la plataforma hardware que recoge toda la información y en el software que gestiona dichos datos. A continuación, en la figura 4.1, se muestra un diagrama que describe de forma gráfica el funcionamiento de este módulo.

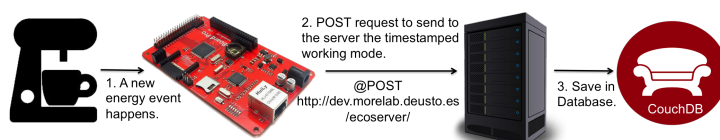


Figura 4.1: Funcionamiento del módulo de captura y almacenamiento de datos.

4.3.1 Hardware

Por motivos ya expuestos en la sección 2.4.1, se ha optado por utilizar una placa electrónica compatible con Arduino, en concreto la placa iBoard Pro, que se alimenta a 7V. Con el objetivo de cumplir con el requisito RF2, se ha seleccionado un sensor de corriente alterna capaz de medir hasta 30 A. Tal y como se puede ver en la figura 4.2b, la conexión del sensor con la cafetera solo requiere la introducción del conductor neutro (cable azul) del cable que alimenta la cafetera en la abertura del sensor que luego se cierra con un sistema de pinza.

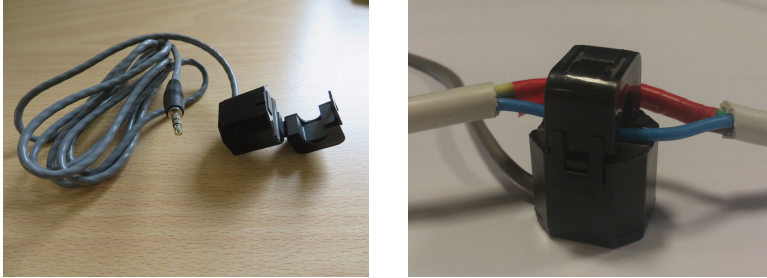


Figura 4.2: Sensor de corriente (a) y (b).

Para dotar de robustez a la plataforma y minimizar errores de hardware, se ha diseñado e imprimido una PCB (*Printed Circuit Board*) para conectar el sensor de corriente a la placa a través de un conector *Jack* de 3.5 mm. El resultado de este diseño se puede ver en la figura 4.3, así como su correspondiente fichero *board* de *Eagle*¹.

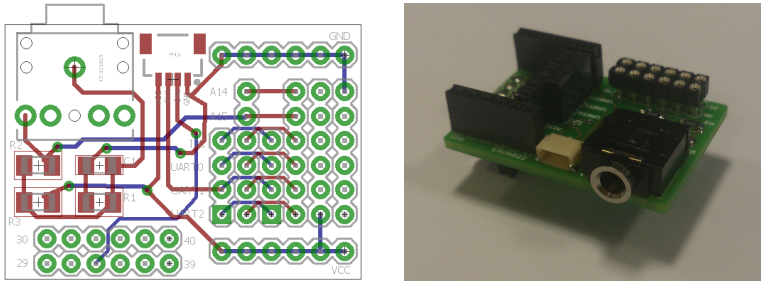


Figura 4.3: PCB para adaptar el sensor de corriente (a) y (b).

En la figura 4.4 se puede ver el ensamblaje de todos los componentes que forman el módulo hardware. Observándola con detalle, se puede apreciar una pila de botón y una tarjeta μ SD.

La primera cumple la función de alimentar el reloj en tiempo real del que dispone la placa.

La segunda almacena un archivo de configuración que contiene el identificador único del dispositivo (*identifier*), información necesaria pa-

¹Software de diseño de PCBs: <http://www.cadsoftusa.com/>

ra establecer la conexión a Internet (*network* y *static*), la dirección IP de un servidor NTP (*Network Time Protocol*) para actualizar el reloj en tiempo real y la predicción de uso calculada para la semana correspondiente. En el anexo C se muestra un ejemplo de este archivo de configuración.

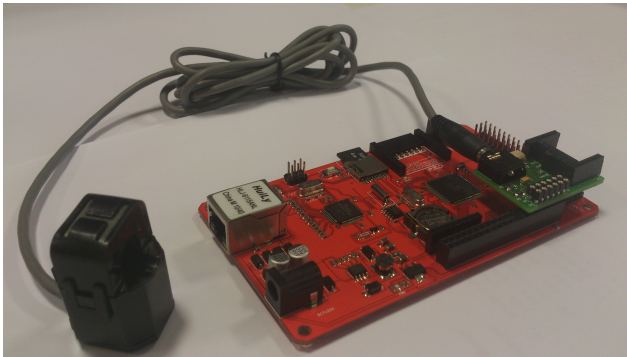


Figura 4.4: Módulo hardware al completo.

Solución a medida *Low Energy*

Era previsible que la solución hardware utilizada en el experimento no iba a ser la de menor consumo posible, ya que Arduino no destaca por su eficiencia energética en la mayoría de sus placas. En la sección 5.3 se realiza un análisis del consumo de esta plataforma que confirma las previsiones, por lo que se ha optado por diseñar una solución *Low Energy* a medida seleccionando exclusivamente los componentes requeridos para cumplir con los requisitos, tanto funcionales, como no funcionales.

En el apartado de análisis de la sección 2.4 se han presentado las características que el hardware debe cumplir, que son las siguientes:

- Capacidad de almacenamiento de 1 MB, además de lo que ocupa el propio sistema.
- Interfaz de comunicación Ethernet o Wi-Fi.
- 1 pin de entrada analógica para conectar el sensor de corriente.

- Reloj en tiempo real.
- 1 pin de salida digital para controlar un relé que posibilite el apagado automático de la cafetera.

Además, con el objetivo de optimizar el rendimiento energético de los dispositivos, se pretende mantenerlos en modo *Sleep* mientras no detectan ningún consumo energético. Generalmente, los microcontroladores que disponen de esta función abandonan el modo *Sleep* cuando se da una de estas condiciones:

- Se activa de forma externa el *Reset* del circuito integrado.
- Por una señal temporal (hora y fecha o ha discurrido un tiempo determinado).
- Cambio de estado de una entrada digital.

Al ser el sensor de corriente de tipo analógico, es necesario incorporar un comparador analógico entre el microcontrolador y el propio sensor, de modo que una vez superada una tensión umbral establecida, el comparador cambia de estado despertando al microcontrolador.

Tras una revisión de los microcontroladores que se ajustan a las necesidades presentadas, se ha optado por hacer uso del PIC18F25K20 de la línea XLP (*eXtreme Low Power*) de *Microchip* (elemento 5 de la figura 4.5). Su consumo es de 800 nA, disminuyendo a 100 nA en modo *Sleep*. Además, el propio circuito integrado dispone de un comparador analógico.

Respecto al módulo Wi-Fi, se ha escogido el RN171 de *Microchip* (elemento 2 de la figura 4.5), que consume 4 μ A en modo *Sleep*, 40 mA recibiendo datos y 120 mA enviándolos. Asimismo, incluye un reloj en tiempo real.

Para almacenar temporalmente los datos de consumo energético capturados, se ha seleccionado una memoria EEPROM 25AA1024 de *Microchip* (elemento 3 de la figura 4.5). Su consumo es muy reducido, 5 mA en escritura y 7 mA en lectura, mientras que en modo *Sleep* se reduce hasta 1 μ A.

Con el fin de detectar cuándo se produce un consumo energético del dispositivo, cafetera en este caso, se ha escogido un sensor de corriente alterna que permite medir intensidades de hasta 30 A y devuelve una tensión entre 0 y 1 V (elemento 4 de la figura 4.5).

Por último, para completar el módulo hardware y a falta de elementos electrónicos pasivos (resistencias, condensadores, etc.) y el circuito de alimentación, se ha optado por integrar el relé RM85 de TME (elemento 6 de la figura 4.5), que va a habilitar al propio módulo para apagar la cafetera.

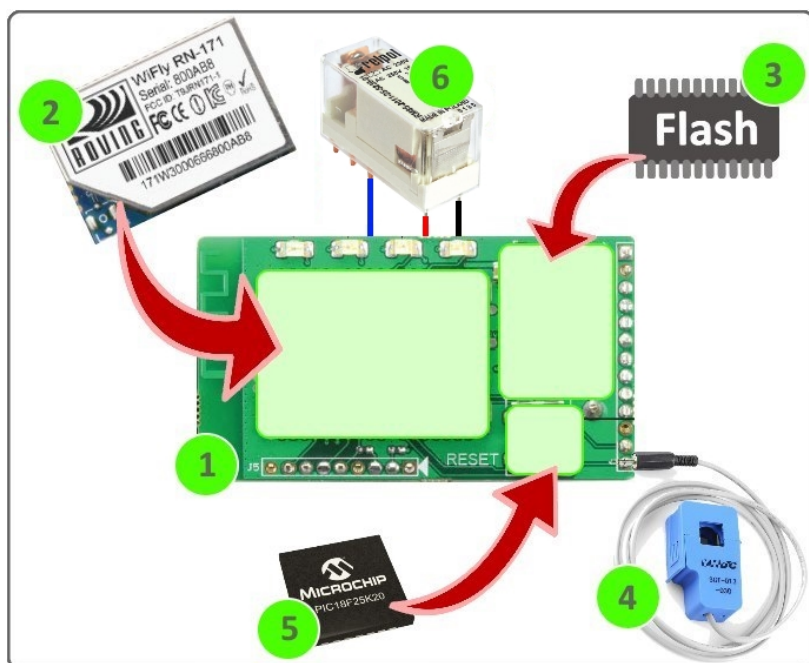


Figura 4.5: Módulo hardware a medida.

En la sección 5.3 se evalúa el rendimiento energético de esta solución a nivel teórico, haciendo uso de los datos recogidos de las hojas de especificaciones.

4.3.2 Almacén de datos

Todos los datos capturados por el módulo hardware son enviados a un servidor donde son almacenados. Dentro de las alternativas existentes para este fin, se diferencian dos categorías: las bases de datos relacionales y las no relaciones (NoSQL). La principal diferencia entre ambas radica en cómo guardan los datos. Las bases de datos relacionales particionan la información en diferentes tablas diseñadas previamente. En cambio, las bases de datos NoSQL almacenan la información en formatos como clave-valor, mapeo de columnas, documentos o grafos sin la necesidad de tener un esquema predefinido de los datos.

Dentro de las bases de datos NoSQL, se ha optado por utilizar CouchDB, una base de datos *Open Source* orientada a documentos JSON (*JavaScript Object Notation*) compuestos por uno o más pares campo/valor. Su accesibilidad mediante una interfaz RESTful permite realizar toda la gestión de la base de datos a través de HTTP. Esta característica simplifica de manera notable la tarea de almacenar datos desde plataformas hardware con recursos limitados sin tener que desarrollar un módulo intermedio en el servidor que ejecute la tarea de almacenamiento en la base de datos.

A continuación, en las tablas 4.1, 4.2 y 4.3, se muestran tres ejemplos de la información que se almacena en la base de datos para cada uno de los tipos de consumo (realización de café, encendido de la cafetera y picos de consumo en *Standby*). Estos mismos ejemplos, en formato JSON, se pueden consultar en el anexo A.

Clave	Valor
ID del dispositivo	ID1
Tipo del dispositivo	Cafetera
Energía consumida	14.41
Tipo de consumo	Café
Tiempo de consumo (seg)	48.23
Fecha	2016-05-04
Hora	14:38:16

Tabla 4.1: Datos de un documento generado debido a la realización de un café.

Clave	Valor
ID del dispositivo	ID3
Tipo del dispositivo	Cafetera
Energía consumida	7.21
Tipo de consumo	Encendido
Tiempo de consumo (seg)	20.07
Fecha	2016-05-12
Hora	08:19:56

Tabla 4.2: Datos de un documento generado debido al encendido de una cafetera.

Clave	Valor
ID del dispositivo	ID3
Tipo del dispositivo	Cafetera
Energía consumida	0.39
Tipo de consumo	Standby
Tiempo de consumo (seg)	2.07
Fecha	2016-05-13
Hora	09:23:38

Tabla 4.3: Datos de un documento generado debido a un pico de consumo en modo *Standby*.

4.3.3 Análisis y conclusiones

Como ya se ha explicado a lo largo de la sección 4.3.1, este módulo de captura y almacenamiento de datos está dividido en dos partes: la placa electrónica que captura los datos de consumo energético y el software de almacenamiento de dichos datos.

Respecto a la primera parte, se han diferenciado dos diseños diferentes. Uno de ellos, el que utiliza la placa iBoard Pro como base, se ha utilizado para realizar los experimentos, ya que es una solución sencilla y barata para desplegar sobre un número considerable de dispositivos. El segundo diseño ha sido realizado a medida y además de proveer las mismas funcionalidades que el primero, tiene un consumo energético insignificante (ver sección 5.3).

Respecto a la segunda parte, que se corresponde con el almacenamiento de los datos, destacar que la posibilidad de acceder a la base de datos mediante una interfaz RESTful reduce la complejidad en el envío de datos desde dispositivos embebidos, ya que muchos de ellos implementan la pila de protocolos TCP/IP. Por lo tanto, el almacenamiento en la base de datos se convierte en un problema trivial.

4.4 Plataforma de registro de dispositivos

En esta sección se describe la plataforma web desarrollada² (ver figura 4.6) para que tras la adquisición de un nuevo dispositivo *Ecoaware* las personas responsables de los mismos puedan registrarlos.



Figura 4.6: Web de registro de dispositivos.

Inicialmente, los propios responsables se registrarán en la plataforma, para a continuación introducir los datos descriptivos del nuevo dispositivo que se va a instalar. La descripción de un dispositivo incluye campos como el número de personas, el tipo de espacio donde está ubicado y el número de horas que el dispositivo está disponible. A partir de aquí, la responsabilidad de los usuarios reside en hacer un uso corriente del dispositivo.

²<http://apps.morelab.deusto.es/ecoaware>

4.4.1 Servicio de registro automático en Twitter

Este servicio se ejecuta inmediatamente después del registro de un dispositivo en la web. Teniendo en cuenta que el flujo de información entre dispositivos se va a producir a través de Twitter, es imprescindible que dispongan de un perfil en la citada red social. Con el fin de que el proceso sea transparente para los usuarios, este servicio genera automáticamente un perfil en Twitter correspondiente a cada uno de los dispositivos. A modo de ejemplo, a continuación se puede observar el modelado de los datos de un dispositivo en su respectivo perfil de Twitter.

The screenshot displays the 'Ecoaware Everyday Things' website. On the left, a 'New device' form is visible with the following fields and values:

- ID: ID7
- Space's name: Espacio 7
- Type of the device: Coffee-maker
- Email: ecoawareth+id7@gmail.com
- Type of the space: Administration
- Number of people: 4
- Place separated for the device: No
- Country: Spain
- Province: Bizkaia
- City: Bilbao
- Type of working schedule: Split working day
- Number of working hours per day: 12
- Device's model: DG Melody

A 'Register' button is located below the form. To the right of the form, a generated Twitter profile is shown for 'ID7 CoffeeM' (@id7coffeem). The profile includes a bio with JSON data: `ecoawth: { cat: coffeeM, sp: { tipo: Administration, nPerson: 4, ded: 0, hor: Partido, mod: DG Melody } }` and a location tag for 'Bilbao, Spain'. The footer of the page indicates 'Copyright © 2014 MORElab'.

Figura 4.7: Registro de un dispositivo en la web y su correspondiente perfil generado en Twitter.

Además, también de forma automática, este nuevo perfil se suscribe a una lista de dispositivos de su mismo tipo (p. ej., máquinas de café), entre cuyos miembros se deberá seleccionar el que tenga un patrón de uso similar al nuevo dispositivo. En la figura 4.8 se puede observar una lista de Twitter, en la que los dispositivos suscritos son cafeteras.

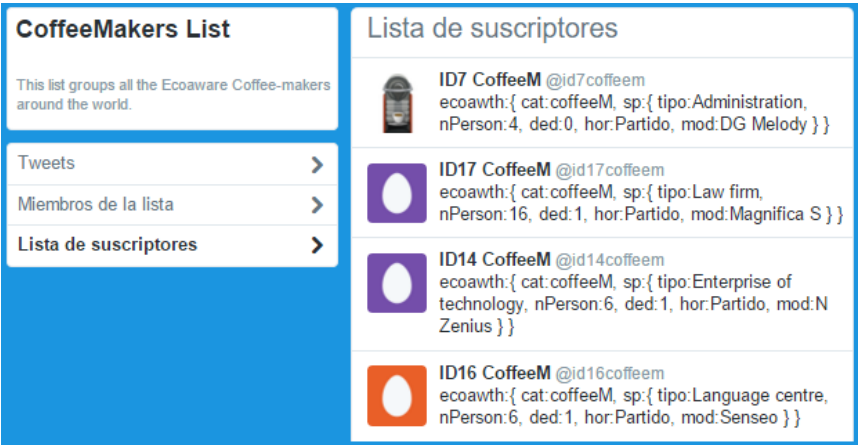


Figura 4.8: Lista de máquinas de café *Ecoaware* registradas en Twitter.

En el anexo D se puede observar el código fuente de este servicio, que está desarrollado en Python y hace uso de las librerías *mechanize*³ y *Tweepy*⁴.

4.4.2 Detección de dispositivos similares

Tal y como se muestra en el diagrama de secuencia de la figura 3.6 del capítulo anterior, una vez registrado el dispositivo en la web, se pone en marcha el servicio *Similitud*. Éste se ejecuta cada vez que se instala un nuevo dispositivo *Ecoaware* (A) con el objetivo de buscar otro dispositivo (B) del mismo tipo que disponga de datos de uso que sean adecuados para el entrenamiento del nuevo objeto. En este apartado se describe el proceso de detección de ese dispositivo B, en el que para ello se ejecuta el coeficiente de similitud de *wGower* descrito en 3.5.2.

A modo de ejemplo y con el fin de clarificar el proceso de detección de dispositivos similares, se va a desarrollar el proceso de cálculo del coeficiente de *wGower* entre dos de los dispositivos bajo experimento (ID1,

³<http://wwwsearch.sourceforge.net/mechanize/>

⁴<http://www.tweepy.org/>

ID3), cuyos datos de perfil se pueden observar en la tabla 4.4.

Identificador	Nº de personas	Lugar dedicado	Jornada (h)	Horario	Ubicación	Tipo de espacio	Modelo de cafetera
ID1	6	No	12	Partido	Bilbao	Centro de investigación	Dolce Gusto Melody
ID3	9	No	12	Partido	Bilbao	Centro de investigación	Dolce Gusto Melody

Tabla 4.4: Características de los espacios donde se ubican las cafeteras ID1 e ID3.

A continuación se muestra la fórmula del coeficiente de wGower entre los dos dispositivos escogidos (ID1, ID3).

$$S(x_1, x_3) = \frac{\delta_{nPers}s_{13nPers} + \delta_{dedEsp}s_{1ID3dedEsp} + \delta_{jorn}s_{13jorn} + \delta_{horar}s_{13horar} + \delta_{ubic}s_{13ubic} + \delta_{esp}s_{13esp}}{\delta_{nPers} + \delta_{dedEsp} + \delta_{jorn} + \delta_{horario} + \delta_{ubic} + \delta_{esp}} + \frac{\delta_{horar}s_{13horar} + \delta_{ubic}s_{13ubic} + \delta_{esp}s_{13esp}}{\delta_{nPers} + \delta_{dedEsp} + \delta_{jorn} + \delta_{horar} + \delta_{ubic}} \quad (4.1)$$

Dado que algunas variables del perfil (*jorn* y *horar*) adquieren el mismo valor para las 11 cafeteras que forman parte del experimento principal de esta tesis doctoral, se han omitido considerando que únicamente iban a generar ruido al realizar el cálculo del algoritmo de similitud. Por lo tanto, la fórmula simplificada queda de la siguiente forma:

$$S(x_1, x_3) = \frac{\delta_{nPers}s_{13nPers} + \delta_{dedEsp}s_{13dedEsp} + \delta_{ubic}s_{13ubic} + \delta_{esp}s_{13esp}}{\delta_{nPers} + \delta_{dedEsp} + \delta_{ubic} + \delta_{esp}} \quad (4.2)$$

En la siguiente tabla, de acuerdo a lo presentado en la sección 3.5.2, se muestran los cálculos realizados para obtener los diferentes factores que determinan el valor final del coeficiente de wGower.

Factor	Fórmula	Desarrollo de la fórmula
$s_{13nPers}$	$1 - \frac{ y_{1nPers} - y_{3nPers} }{R_{nPers}}$	$1 - \frac{ 6-9 }{19} = 0,84$
$s_{13dedEsp}$	$0 \text{ (si } y_{1dedEsp} \neq y_{3dedEsp}) \text{ ó } 1 \text{ (si } y_{1dedEsp} = y_{3dedEsp})$	1
s_{13ubic}	$0 \text{ (si } y_{1ubic} \neq y_{3ubic}) \text{ ó } 1 \text{ (si } y_{1ubic} = y_{3ubic})$	1
s_{13esp}	$\varphi_{13} \text{ (si } y_{1esp} \neq y_{3esp}) \text{ ó } 1 \text{ (si } y_{1esp} = y_{3esp})$	1

Donde R_{nPers} adquiere el valor 19, ya que el rango de valores de la variable $nPers$ para las 11 cafeteras es dicho valor.

Por lo tanto, el valor del coeficiente de wGower sería el siguiente:

$$S(x_1, x_3) = \frac{\delta_{nPers} s_{13nPers} + \delta_{dedEsp} s_{13dedEsp} + \delta_{ubic} s_{13ubic} + \delta_{esp} s_{13esp}}{\delta_{nPers} + \delta_{dedEsp} + \delta_{ubic} + \delta_{esp}} = \frac{0,9 * 0,84 + 0,05 * 1 + 0,05 * 1 + 0,4 * 1}{0,9 + 0,05 + 0,05 + 0,4} = 0,89 \quad (4.3)$$

Este proceso llevado a cabo entre las cafeteras ID1 e ID3, se ha realizado entre todas las máquinas de café que han formado parte del experimento, obteniendo los siguientes resultados:

Dispositivos	ID	ID2	ID3	ID5	ID7	ID12	ID13	ID14	ID15	ID16	ID17
ID1	1,00	0,67	0,89	0,80	0,87	0,26	0,48	0,88	0,74	0,75	0,48
ID2	0,67	1,00	0,78	0,48	0,54	0,59	0,80	0,55	0,41	0,43	0,81
ID3	0,89	0,78	1,00	0,69	0,76	0,37	0,59	0,77	0,63	0,64	0,59
ID5	0,80	0,48	0,69	1,00	0,75	0,25	0,46	0,87	0,82	0,74	0,38
ID7	0,87	0,54	0,76	0,75	1,00	0,06	0,28	0,68	0,88	0,68	0,32
ID12	0,26	0,59	0,37	0,25	0,06	1,00	0,78	0,38	0,18	0,13	0,62
ID13	0,48	0,80	0,59	0,46	0,28	0,78	1,00	0,60	0,40	0,35	0,84
ID14	0,88	0,55	0,77	0,87	0,68	0,38	0,60	1,00	0,80	0,75	0,51
ID15	0,74	0,41	0,63	0,82	0,88	0,18	0,40	0,80	1,00	0,87	0,32
ID16	0,75	0,43	0,64	0,74	0,68	0,13	0,35	0,75	0,87	1,00	0,39
ID17	0,48	0,81	0,59	0,38	0,32	0,62	0,84	0,51	0,32	0,39	1,00

Tabla 4.5: Cálculo del coeficiente de wGower entre todos los dispositivos.

Estos coeficientes son los que van a permitir simular el consumo en modo *Cooperative* siguiendo el proceso que se explica en la sección 5.2. En la siguiente sección del capítulo 5, la 5.2.1, se presentan los resultados obtenidos a nivel de consumo energético con el modo *Cooperative* activado. Además se compararán con los del modo *Ecoaware* y los del resto de modos ya existentes previos a esta tesis doctoral.

4.4.3 Mecanismo de compartición de datos entre dispositivos

Para que un dispositivo pueda compartir sus datos de uso con los de otro, se ha desarrollado un mecanismo de compartición de datos entre dispositivos, denominado *Servicio Mensajes* en la figura 3.6. El intercambio de información, tanto la petición del patrón de uso como el envío del

mismo, se produce a través de mensajes directos en la red social Twitter. Esto permite que el flujo de información entre dispositivos no sea de dominio público y se limite a un ámbito privado.

La estructura del mensaje para pedir los datos de uso a otro dispositivo será la siguiente:

"Could you send me the last N days' usage data?"

Donde N es el número de días de los que se requieren los datos de uso.

La respuesta a este mensaje contendrá el número de cafés realizados en cada franja de una hora durante los últimos N días solicitados. La respuesta más larga, tal y como se deduce en la sección 5.2, contendrá los datos de 20 días. Por lo tanto, en el caso más extremo de que el dispositivo estuviera ubicado en un lugar en el que hay actividad las 24 horas del día, el número de franjas de una hora a contabilizar asciende a 480 (20×24). Se puede pensar que introducir esta cantidad de caracteres en un mensaje de Twitter puede ser un problema, pero desde el 12 de agosto de 2015⁵ no existe limitación alguna en este sentido para los mensajes directos.

A continuación, se va a mostrar un ejemplo de mensaje cuyo contenido es el número de cafés realizados en cada franja horaria durante 5 días en un lugar en el que la actividad es de 7:00 h a 19:00 h, es decir de 12 horas:

"0,0,2,2,6,0,0,1,4,2,0,0 0,0,1,2,5,0,0,0,6,1,0,0 0,0,2,1,7,0,0,1,5,1,0,0
0,0,2,2,4,0,0,2,5,1,0,0 0,0,2,1,7,0,0,0,6,0,0,0"

El código fuente de este servicio se puede observar en el anexo D y, al igual que el servicio anterior, está desarrollado en *Python* haciendo uso de la librería *Tweepy*.

⁵<https://blog.twitter.com/es/2015/eliminamos%2Del%2Dl%2Dmite%2Dde%2D140%2Dcaracteres%2Dde%2Dlos%2Dmensajes%2Ddirectos%2D0>

4.4.4 Análisis y conclusiones

A lo largo de la sección 4.4 se han descrito los pasos que se darían una vez se instala un nuevo dispositivo *Ecoaware*. Estos pasos se resumen a continuación:

1. Registro del dispositivo en la web por parte de un usuario.
2. Registro automático del dispositivo en Twitter.
3. Detección de dispositivos similares.
4. Compartición de datos entre dispositivos similares.

El tercero de estos pasos es la principal aportación de esta sección, ya que es determinante para que un nuevo dispositivo solicite los datos de uso al dispositivo existente más adecuado para utilizarlos en su proceso de aprendizaje. En el siguiente capítulo se comprobará si el uso de esta información acelera el aprendizaje de nuevas cafeteras y, por tanto, se traduce en una reducción del consumo energético.

4.5 Implementación del método de aprendizaje y predicción de uso

En esta sección se presenta el modelo ARIMA seleccionado de acuerdo a la metodología de Box-Jenkins (?). Además en el anexo E se puede observar el código fuente que realiza el cálculo de la predicción de uso en base a datos de uso previos.

4.5.1 Identificación del modelo

Antes de identificar el modelo, el primer paso es comprobar si la serie temporal de cafés (ver figura 4.9) es estacionaria o no. En caso de no

serlo, hay que transformar la serie original de forma que la nueva serie transformada sí lo sea.

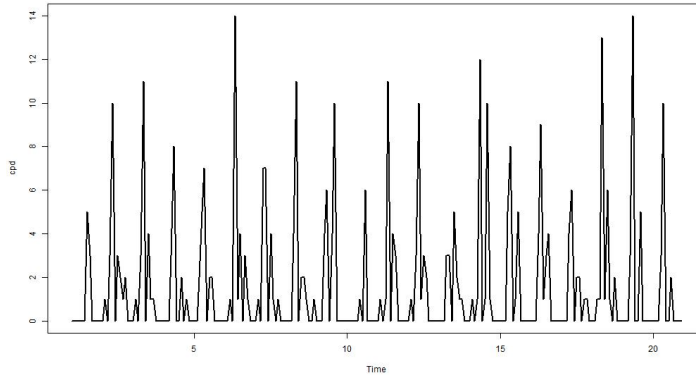


Figura 4.9: Serie temporal de cafés preparados durante 20 días hábiles en uno de los espacios bajo estudio.

Para realizar esta verificación, Kwiatkowski et al. (?) desarrollaron la *Prueba KPSS* (Kwiatkowski, Phillips, Smichdt y Shin), en la que proponen contrastar como hipótesis nula la hipótesis de estacionalidad en tendencia. El resultado de esta prueba para los datos de uso recogidos ha sido $p\text{-value}=0.0219$. Está establecido que un valor menor a 0.05, es suficiente para rechazar la hipótesis nula de que la serie temporal es estacionaria en tendencia.

La figura 4.10 muestra la serie temporal de cafés diferenciada ($d = 1$) y sus gráficas ACF (Función de auto-correlación) y PACF (Función de auto-correlación parcial) asociadas. Para seleccionar el modelo apropiado se han de identificar las funciones de auto-correlación obtenidas (ACF y PACF) con alguno de los modelos teóricos que se presentan en el anexo F. Tras comparar los modelos teóricos con los empíricos, se podría considerar que el modelo $ARIMA(3, 1, 1)$ se aproxima al obtenido con los datos de uso disponibles.

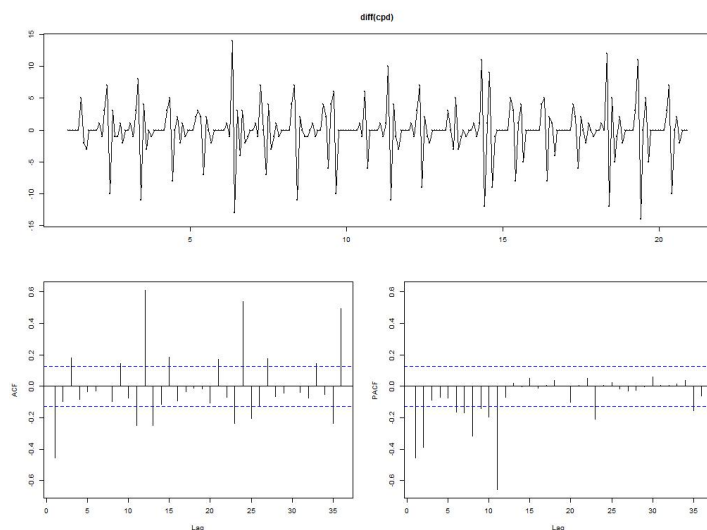


Figura 4.10: Serie diferenciada y sus ACF y PACF para identificar el modelo

Sin embargo, para seleccionar el mejor ajuste de ARIMA se han probado diferentes modelos. La elección del mismo se ha basado en el criterio de información de Akaike (AICE) (?). Este criterio se utiliza para comparar varios modelos de ajuste sobre la misma serie temporal. El modelo con el criterio de información más pequeño es el que mejor se ajusta a los datos. Finalmente, el modelo seleccionado ha sido el modelo $ARIMA(3, 1, 1)(2, 0, 2)$. El último conjunto de este modelo se refiere a la parte estacional de la ARIMA: P es el número de términos autorregresivos estacionales (SAR), Q es el número de términos de promedio móvil estacional (SMA) y D es el número de diferencias estacionales. La parte estacional de un modelo ARIMA, también llamado SARIMA, tiene la misma estructura que la parte regular (no-estacional).

Finalmente, se han representado en la figura 4.11 los residuales de la regresión para determinar si el modelo ARIMA utilizado describe apropiadamente las características de la serie temporal. Esta figura muestra que no hay autocorrelaciones significantes en los residuales del modelo $ARIMA(3, 1, 1)(2, 0, 2)$. Esto se afirma ya que los coeficientes de au-

tocorrelacion se concentran dentro de la ventana de Bartlett⁶. Además, realizando la prueba de Ljung-Box (?) se obtiene un $p\text{-value}=0.3438$, un valor mucho mayor que el error de confianza $\alpha = 0.05$ mencionado previamente. Este hecho demuestra que el modelo seleccionado se ajusta a los datos de uso de los que se dispone, encontrando así residuales aleatorios y que no presentan autocorrelación. Por último, se destaca que los residuales siguen una distribución normal $\mathcal{N}(0, \sigma)$ con ruido gaussiano⁷.

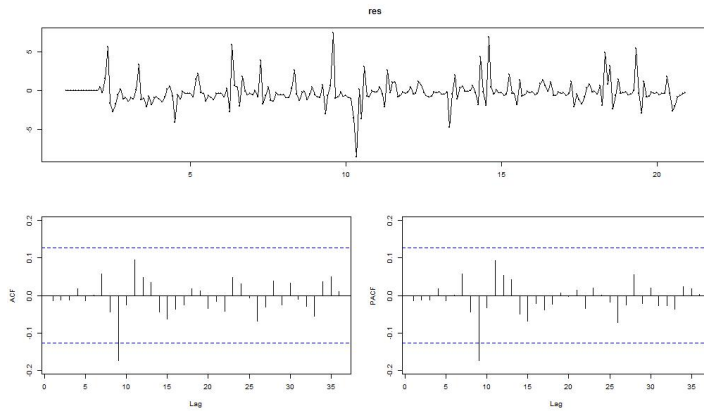


Figura 4.11: Los residuales y sus ACF y PACF asociados.

4.5.2 Validación del modelo

Para validar el modelo seleccionado, se ha comparado la predicción realizada para 5 días de funcionamiento de la máquina de café con los valores reales registrados durante ese mismo período de 5 días.

La figura 4.12 muestra en su línea azul la predicción realizada para 5 días. La línea verde se corresponde con los datos reales capturados durante esos 5 días y la línea roja discontinua define los límites de con-

⁶https://en.wikipedia.org/wiki/Window_function#Triangular_window

⁷https://media4.obspm.fr/public/VAU/instrumentacion/observer/analizar/ruido-gaussiano/OBSERVER_3.html

fianza, tanto superior como inferior. El límite de error superior se ha calculado con un 95 % de confianza, mientras que el inferior se ubica en 0 cafés. De acuerdo a la predicción, el número de cafés preparados nunca debería superar el límite superior ni va a presentar valores negativos.

Analizando los datos de la gráfica de la figura 4.12 se obtiene que ha habido un error del 26 % en la predicción respecto a lo ocurrido en la realidad. Sin embargo, lo realmente relevante es el porcentaje de error al predecir si la máquina de café se debe quedar encendida o no en función del número de cafés predichos. En este caso, el error disminuye a un 15 %, por lo que el modelo seleccionado se considera adecuado.

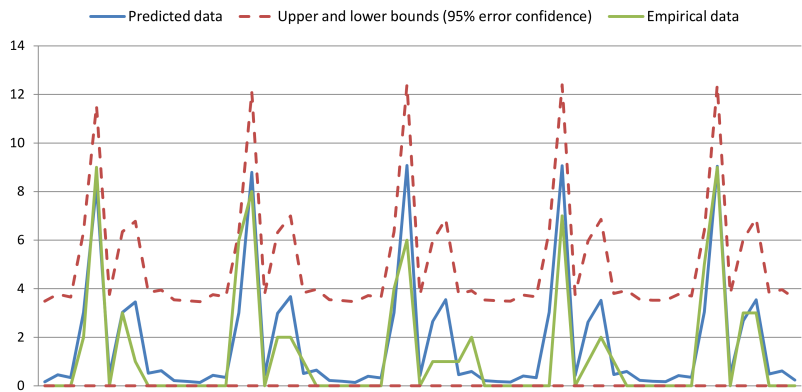


Figura 4.12: Predicción de cafés preparados durante 5 días.

4.5.3 Análisis y conclusiones

En esta sección se han explicado los pasos que se han ido dando para identificar un modelo de ARIMA que se ajuste a los datos de de uso capturados. La gráfica de la figura 4.12 hace indicar que la elección del modelo ha sido la correcta, ya que presenta un error del 15 % entre la predicción de uso y los datos reales recogidos. Además, esta decisión está avalada por los datos positivos que se desprenden de las gráficas comparativas de consumos que se recogen en la sección 5.2.1.

5

Evaluación y resultados

El ciclismo es un importante elemento del futuro. Algo no marcha bien en una sociedad que va en coche al gimnasio.

—BILL NYE

5.1 Introducción

COMO SE HA EXPLICADO a lo largo de los capítulos anteriores, en esta tesis doctoral se propone añadir a los electrodomésticos que disponen de modo *Standby* y que son susceptibles de ser utilizados en entornos públicos y/o semipúblicos, la capacidad de adaptar su funcionamiento de una forma inteligente al modo más eficiente en cada momento en cuanto a energía se refiere (modo *Ecoaware*). Además, se propone también que los electrodomésticos compartan con otros dispositivos del mismo tipo datos correspondientes al uso que se hace de ellos para aumentar la eficiencia energética de los mismos, minimizando así el período de aprendizaje que los electrodomésticos necesitarían para poder operar de manera inteligente (modo *Cooperative*).

Para evaluar la eficiencia energética de estos dos modos, *Ecoaware* y *Cooperative*, se ha llevado a cabo una evaluación cuantitativa que se describe a continuación.

5.2 Metodología de evaluación

Con el objetivo de validar las hipótesis presentadas en esta tesis doctoral, se ha llevado a cabo un experimento que una vez finalizado ha permitido realizar un análisis comparativo. Este estudio se ha llevado a cabo en 18 entornos de trabajo y en cada uno de ellos se ha medido el patrón de uso de un dispositivo de uso compartido, concretamente de una cafetera. Pese a estar anonimizados por motivos de privacidad, en la tabla 5.1 se pueden ver las características de cada uno de los espacios donde se ubican estas cafeteras y el modelo de las mismas.

ID	Nº de personas	Lugar dedicado	Jornada (h)	Horario	Ubicación	Tipo de espacio	Modelo de cafetera
ID1	6	No	12	Partido	Bilbao	Centro de investigación	Dolce Gusto Melody
ID2	15	No	12	Partido	Bilbao	Centro de investigación	Dolce Gusto Piccolo
ID3	9	No	12	Partido	Bilbao	Centro de investigación	Dolce Gusto Melody
ID4	4	No	12	Partido	Bilbao	Administración	Dolce Gusto Piccolo
ID5	4	No	12	Partido	Bilbao	Startup	Nespresso Essenza
ID6	4	No	12	Partido	Bilbao	Centro de investigación	Nespresso Le Cube
ID7	3	No	12	Partido	Bilbao	Administración	Dolce Gusto Melody
ID8	8	Sí	12	Partido	Bilbao	Coworking	Nespresso Essenza
ID9	6	Sí	12	Partido	Madrid	Coworking	De goteo
ID10	9	Sí	12	Partido	Bilbao	Coworking	De goteo
ID11	5	Sí	12	Partido	Tres Cantos	Coworking	Nespresso Pixie
ID12	23	Sí	12	Partido	Madrid	Empresa tecnológica	Nespresso Zenius
ID13	17	Sí	12	Partido	Madrid	Empresa tecnológica	Nespresso Zenius
ID14	6	Sí	12	Partido	Madrid	Empresa tecnológica	Nespresso Zenius
ID15	4	Sí	12	Partido	Madrid	Universidad	Dolce Gusto Melody
ID16	6	Sí	12	Partido	Bilbao	Centro de idiomas	Philips Senseo
ID17	16	Sí	12	Partido	Madrid	Bufete de abogados	DeLonghi Magnifica S
ID18	6	Sí	24	Continuo	Madrid	Laboratorio ciudadano	Dolce Gusto Melody

Tabla 5.1: Características de los espacios donde se ubican los dispositivos.

Como ya se ha explicado en anteriores capítulos de esta tesis doctoral, a cada una de las cafeteras que forma parte del experimento se le ha añadido un dispositivo de medición de la corriente para poder detectar los consumos de la misma y una interfaz Ethernet para poder almacenarlos en el servidor. Una vez incorporado este módulo de captura de datos,

las cafeteras han continuado ubicadas en el mismo lugar y funcionando del mismo modo que lo hacían hasta ese momento. Se puede observar en la figura 5.2 que los primeros datos de consumo energético se recogieron en marzo de 2014 para algunas de las cafeteras y en marzo de 2015 los últimos, pero no todas han estado recogiendo datos durante el mismo tiempo. Esto se debe a la dificultad de añadir a las cafeteras el módulo de captura de datos al mismo tiempo, ya que la ubicación entre las mismas no era cercana. Además, la instalación del hardware en las cafeteras ha sido ardua, ya que con el objetivo de que sea prácticamente transparente para el usuario, cada instalación supuso un esfuerzo importante. En la figura 5.1 se puede observar una de estas cafeteras tras la instalación del módulo hardware. Detrás del dispositivo se puede apreciar una caja que oculta dicho hardware y que pasa casi desapercibida.



Figura 5.1: Una de las cafeteras del experimento tras la intervención.

El número de cafeteras que finalmente han formado parte del experimento se ha visto reducido a 11, ya que problemas diversos (hardware, conexión de red, cortes de alimentación, etc.) han motivado una deficiente captura de datos de 7 de estas máquinas. Destacar que a lo largo del mes de octubre de 2014 no se recogieron datos por diversos cambios que se hicieron en el servidor.

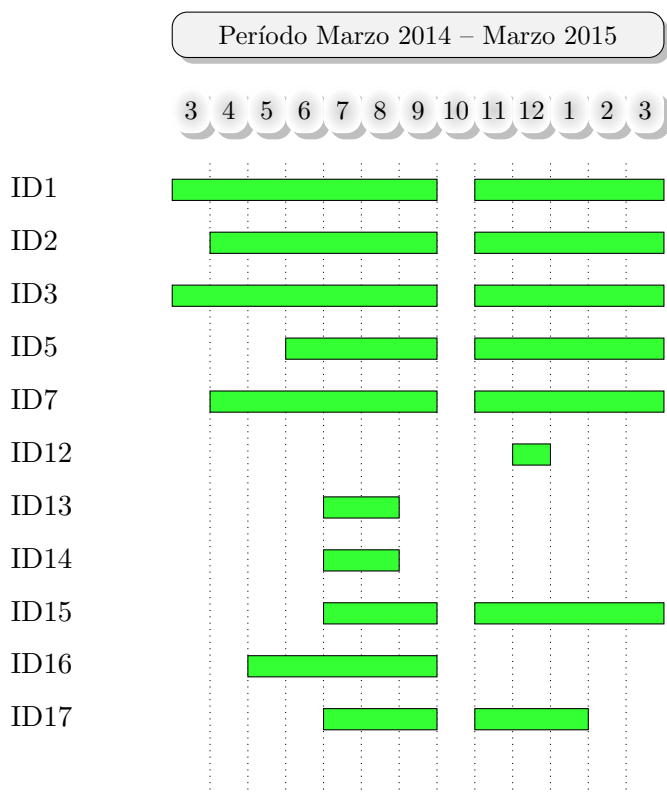


Figura 5.2: Período de captura de datos de las cafeteras (12 meses).

Estos datos recogidos a lo largo de estos 12 meses son los que se han tomado como punto de referencia al evaluar la eficiencia energética de los modos *Ecoaware* y *Cooperative* propuestos en esta tesis doctoral. Este punto de referencia, que se ha denominado consumo real, ha sido com-

parado con los consumos simulados que se presentan a continuación. Para cada uno de los modos de funcionamiento (*On-Off*, *Auto-power Down*, *Ecoaware* y *Cooperative*), se presenta un diagrama de flujo que representa qué proceso se ha seguido para el cálculo de su consumo.

Donde: **energy** representa un consumo energético leído de la BD.
datetime contiene la fecha y hora de un consumo energético.
type es el tipo de un consumo leído de la BD. Adquiere los siguientes valores: *START_TIME*, *STANDBY* o *COFFEE*.
coffeeMade es una variable lógica que adquiere el valor *Verdadero* cuando se lee un consumo de tipo *COFFEE*.
totalEnergy representa el consumo energético total que consume la cafetera en cada uno de los modos.
deltaTime representa el tiempo que transcurre desde que se prepara un café hasta que se prepara el siguiente.
lastCoffeeTime se corresponde con la hora del último consumo de tipo *COFFEE* leído.
timeBetweenSB es el tiempo medio entre 2 picos *Standby*.
startTimeConsumption es el consumo energético medio al encender la máquina de café.
APDTime es el tiempo determinado para la función de auto-apagado.
APDConsumption representa la suma de los picos de consumo *Standby* en modo *Auto-power Down*.
nSB adquiere el número de picos de consumo *Standby* en un período de tiempo (*APDTime* o *deltatime*).
SBCconsumption es el consumo energético medio de un pico *Standby*.
ecoawareHours representa las franjas de tiempo en las que es favorable mantener la cafetera en modo *Standby*.
timeToEnd es el tiempo que transcurre entre la preparación del último café en una franja horaria incluida en *ecoawareHours* y la hora de finalización de dicha franja.

■ Consumo On-Off.

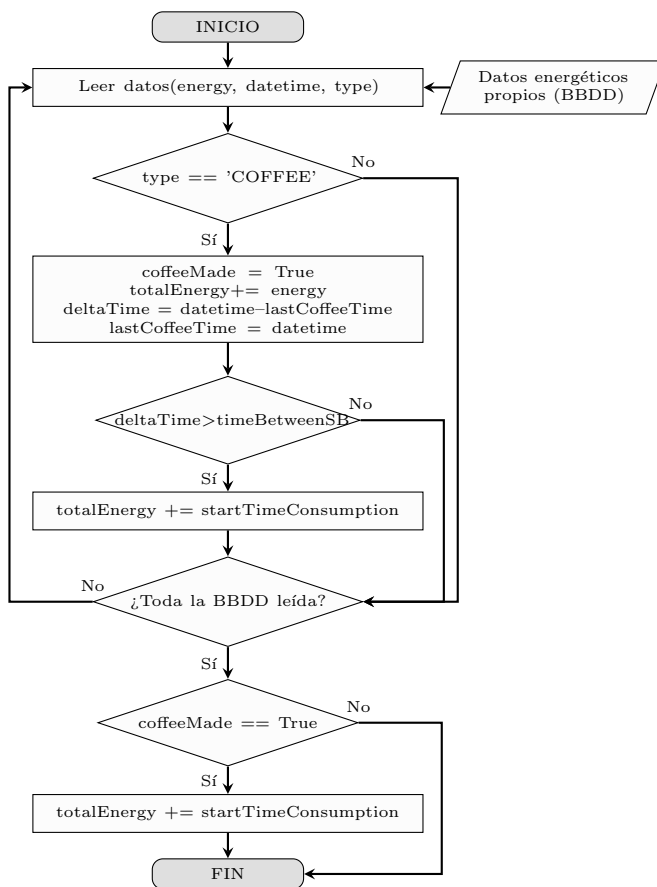


Figura 5.3: Cálculo del consumo en modo On-Off.

■ Consumo *Auto-power down*.

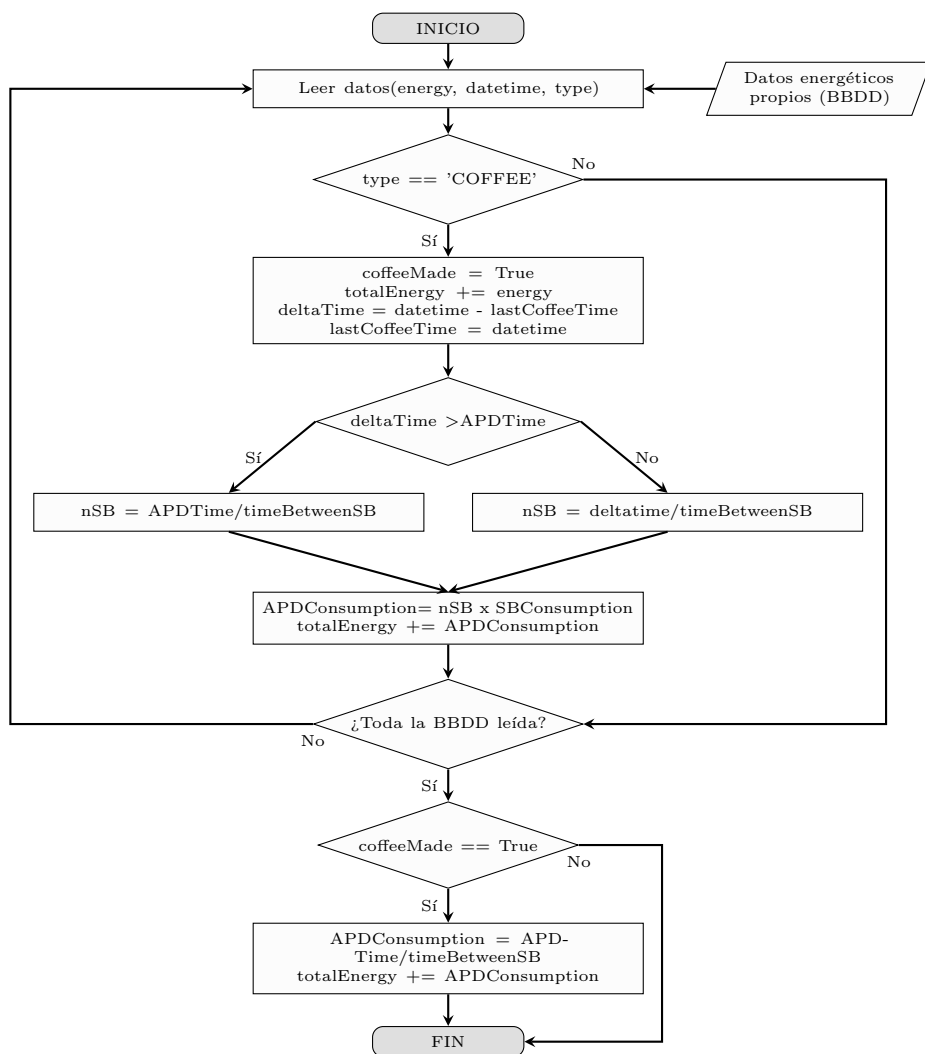


Figura 5.4: Cálculo del consumo en modo *Auto-power Down*.

■ Consumo Ecoaware.

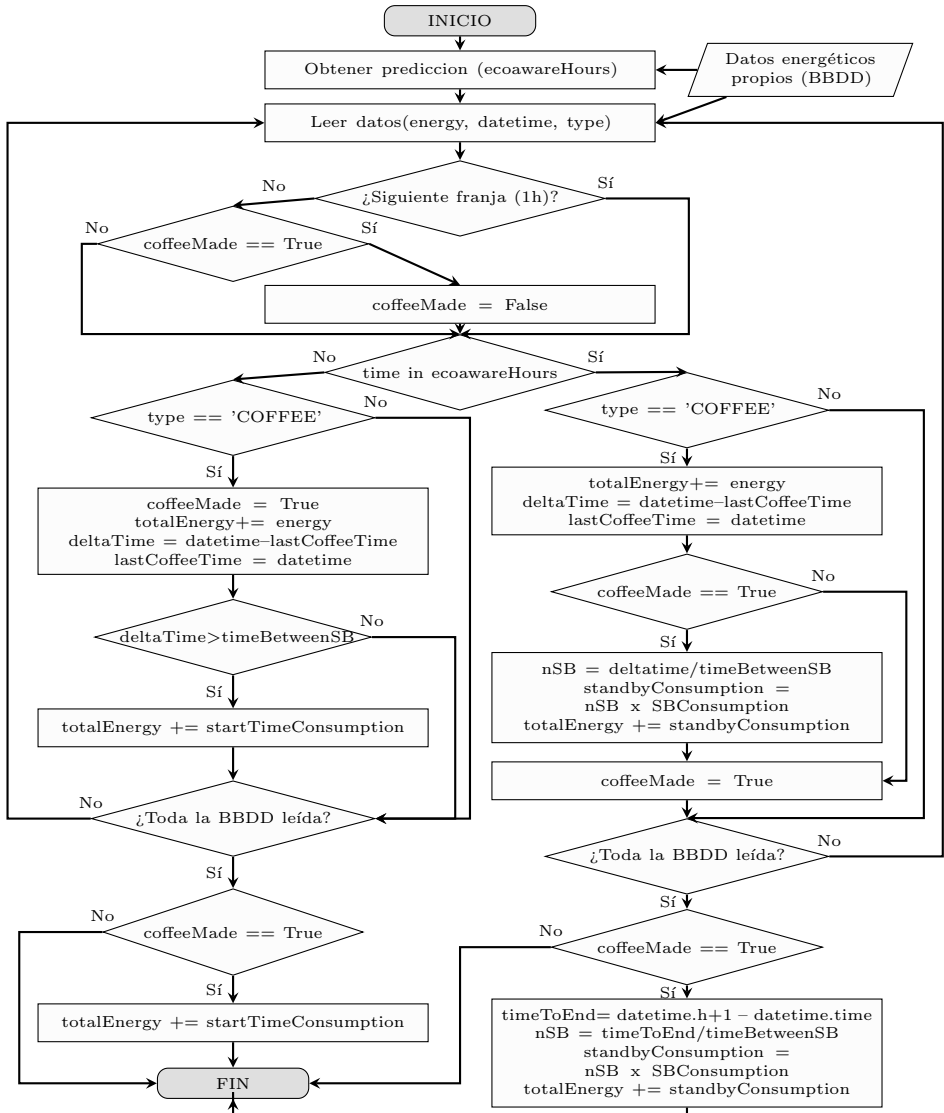


Figura 5.5: Cálculo del consumo en modo *Ecoaware*.

■ Consumo Cooperative.

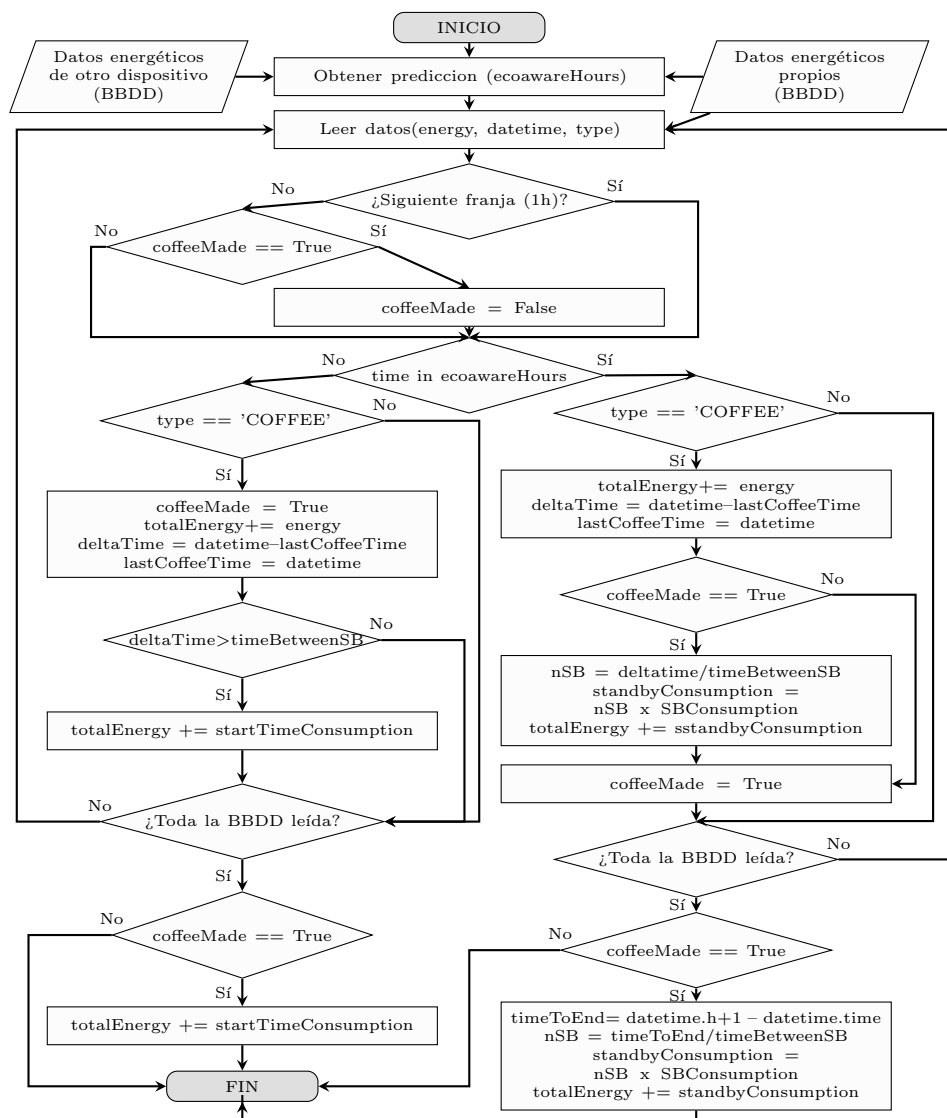


Figura 5.6: Cálculo del consumo en modo Cooperative.

Estos cuatro consumos, pese a que se consideran simulados por no ser consumos reales que se hayan producido, sí que son muy aproximados a los que se habrían originado en caso de haber funcionado las cafeteras en cada uno de los modos de funcionamiento correspondientes. Esto se debe a que los consumos energéticos generados de los cafés realizados se han extraído de la base de datos y los consumos derivados del modo *Standby* o de arrancar el electrodoméstico son medias de consumos que han generado las propias máquinas de café de forma real. En un trabajo previo (?) se aumentó una máquina de café, dotándola de la capacidad de funcionar en modo *Ecoaware*, pero para esta experimentación se ha descartado, ya que habría habido que manipular cafeteras ajenas, incluso algunas que eran cedidas temporalmente por Nespresso.

Para simular el consumo en modo *Cooperative* de una cafetera A, como ya se ha explicado, es necesario utilizar datos de uso de una cafetera B. Para seleccionar cuál debe ser esa máquina de café, se ha propuesto utilizar el coeficiente de *wGower*. Así, se detecta el dispositivo previamente existente que tiene un mayor grado de similitud con el nuevo a instalar.

La tabla que se muestra a continuación, contiene los mismos valores presentados en la tabla 4.5, pero hay ciertos matices diferentes. Por una parte se puede observar que hay algunos valores que se encuentran tachados. Esto se debe a que cuando algunos dispositivos se han puesto en funcionamiento, otros no estaban en marcha, por lo tanto no hay datos previos de los mismos. Por ello, estos valores no se han tenido en cuenta en el momento de seleccionar el dispositivo más afín. Por otra parte, los resaltados en negrita indican entre qué máquinas de café existe un mayor grado de similitud, de modo que han sido las utilizadas para realizar la simulación de consumo energético en modo *Cooperative*.

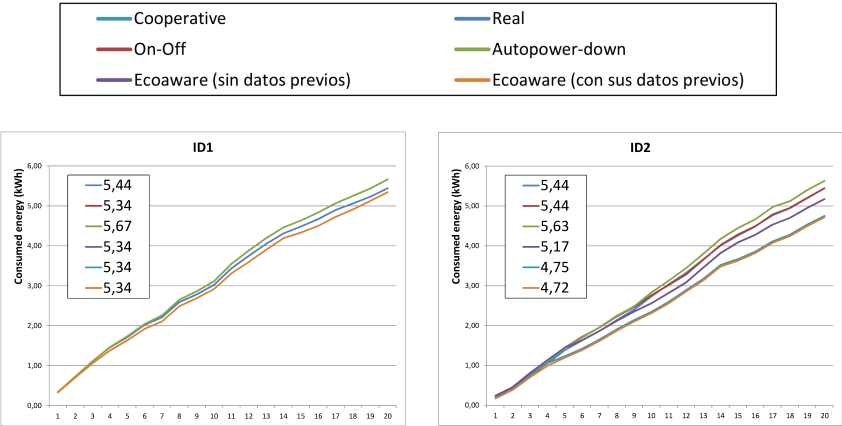
Dispositivos	ID	ID2	ID3	ID5	ID7	ID12	ID13	ID14	ID15	ID16	ID17
ID1	1,00	0,67	0,89	0,80	0,87	0,26	0,48	0,88	0,74	0,75	0,48
ID2	0,67	1,00	0,78	0,48	0,54	0,59	0,80	0,55	0,41	0,43	0,81
ID3	0,89	0,78	1,00	0,69	0,76	0,37	0,59	0,77	0,63	0,64	0,59
ID5	0,80	0,48	0,69	1,00	0,75	0,25	0,46	0,87	0,82	0,74	0,38
ID7	0,87	0,54	0,76	0,75	1,00	0,06	0,28	0,68	0,88	0,68	0,32
ID12	0,26	0,59	0,37	0,25	0,06	1,00	0,78	0,38	0,18	0,13	0,62
ID13	0,48	0,80	0,59	0,46	0,28	0,78	1,00	0,60	0,40	0,35	0,84
ID14	0,88	0,55	0,77	0,87	0,68	0,38	0,60	1,00	0,80	0,75	0,54
ID15	0,74	0,41	0,63	0,82	0,88	0,18	0,40	0,80	1,00	0,87	0,32
ID16	0,75	0,43	0,64	0,74	0,68	0,13	0,35	0,75	0,87	1,00	0,39
ID17	0,48	0,81	0,59	0,38	0,32	0,62	0,84	0,54	0,32	0,39	1,00

Tabla 5.2: Cálculo del coeficiente de wGower entre todos los dispositivos.

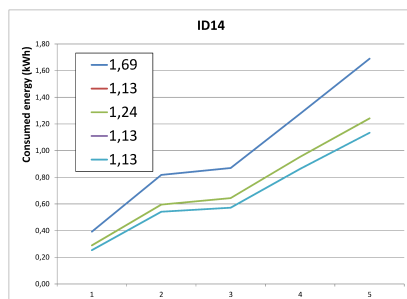
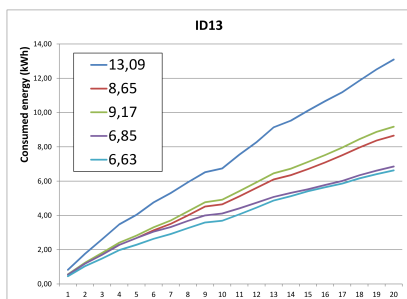
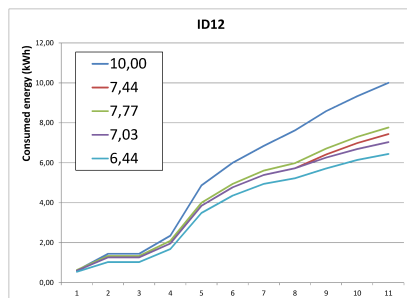
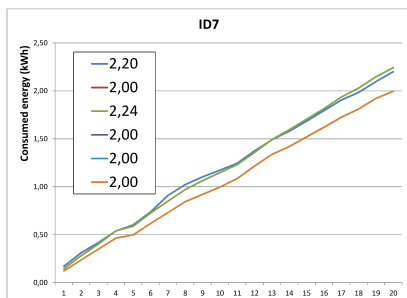
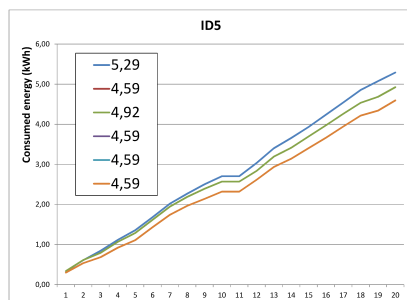
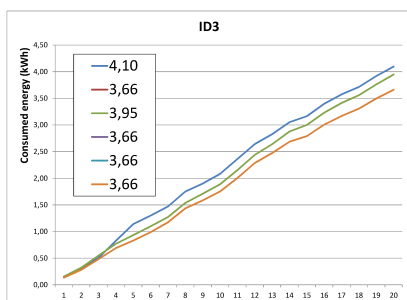
5.2.1 Análisis y conclusiones

En las gráficas que se presentan a continuación, se muestra la evolución del consumo energético real de cada una de las 11 cafeteras bajo experimento en comparación con los modos de funcionamiento previamente existentes (*On-Off* y *Auto-power Down*) y con los nuevos propuestos en esta tesis doctoral (*Ecoaware*, sin datos y con datos propios previos, y *Cooperative*). En los dispositivos *ID12*, *ID13*, *ID14* e *ID16*, no se ha calculado el consumo en modo *Ecoaware* con datos propios previos, ya que no se dispone de datos suficientes.

Gráficas comparativas del consumo energético de las 11 cafeteras.



Gráficas comparativas del consumo energético de las 11 cafeteras.



Gráficas comparativas del consumo energético de las 11 cafeteras.

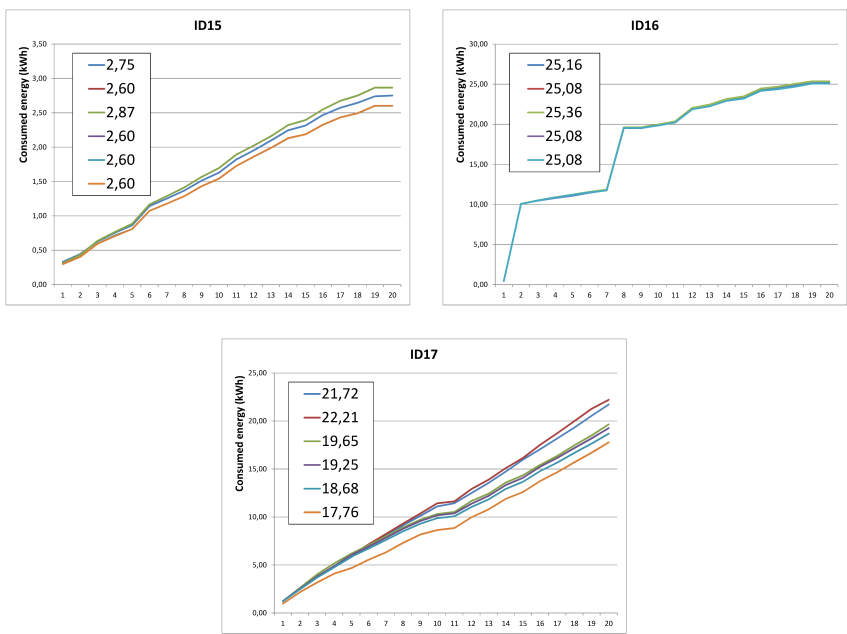


Tabla 5.3: Evolución del consumo energético de las cafeteras bajo estudio.

En las gráficas de la tabla 5.3 se puede observar que con los nuevos modos de funcionamiento propuestos, todos los dispositivos han reducido el consumo energético en mayor o menor medida, exceptuando el ID16 en el que se ha mantenido. Analizando las mismas, se han detectado dos categorías de dispositivos, que al igual que los modos de funcionamiento, se han denominado *On-Off* y *Ecoaware*. En la primera de estas categorías se encuentran las cafeteras para las que el consumo en modo *Ecoaware* coincide con el del modo *On-Off*. En cambio, en la segunda categoría se encuentran las máquinas de café para las que el modo dinámico *Ecoaware* es energéticamente más eficiente que el modo *On-Off*. Es destacable que los 4 dispositivos que se ubican en esta categoría son también los que más usuarios tienen.

La tabla 5.4 resume los datos de las gráficas presentadas. En color verde se han resaltado los dispositivos agrupados dentro de la categoría *Ecoaware*. Las cafeteras que proporcionalmente más han reducido su consumo de energía han sido *ID12*, *ID13* e *ID14*, sobrepasando el 30 % de ahorro energético y llegando una de ellas a casi un 50 %. Por el contrario, las que menos mejora han logrado han sido *ID1*, *ID2*, *ID15* e *ID16*, que no superan el 5 % de ahorro de energía. Esto se debe a que las máquinas de café *ID12*, *ID13* e *ID14* se dejan encendidas durante toda la jornada laboral, por lo que el margen de mejora, como se puede observar, es muy alto.

ID	Consumo real	Consumo <i>Ecoaware</i> (sin datos)	Ahorro <i>Ecoaware</i> (sin datos)	Consumo <i>Cooperative</i>	Ahorro <i>Cooperative</i>	Consumo <i>Ecoaware</i> (con datos)	Ahorro <i>Ecoaware</i> (con datos)
ID1	5.44	5.34	2 %	5.34	2 %	5.34	2 %
ID2	5.44	5.17	5 %	4.75	13 %	4.72	13 %
ID3	4.10	3.66	11 %	3.66	11 %	3.66	11 %
ID5	5.29	4.59	13 %	4.59	13 %	4.59	13 %
ID7	2.20	2.00	9 %	2.00	9 %	2.00	9 %
ID12	10.00	7.03	30 %	6.44	36 %	—	—
ID13	13.09	6.85	48 %	6.63	49 %	—	—
ID14	1.69	1.13	33 %	1.13	33 %	—	—
ID15	2.75	2.60	5 %	2.60	5 %	2.60	5 %
ID16	25.16	25.08	0 %	25.08	0 %	—	—
ID17	21.72	19.25	11 %	18.68	14 %	17.76	18 %

Tabla 5.4: Consumo energético en kWh en cada espacio en modos *Ecoaware* y *Cooperative* (20 días, excepto *ID12* e *ID14*).

A continuación, en las figuras 5.7, 5.8 y 5.9, se presenta el consumo energético de los dispositivos, separándolos por categorías. En el primer gráfico (5.7) se ve claramente como el uso de los modos dinámicos propuestos en esta tesis doctoral, conlleva un ahorro energético respecto al consumo real que se ha generado y al que se habría producido haciendo uso del modo *On-Off*.

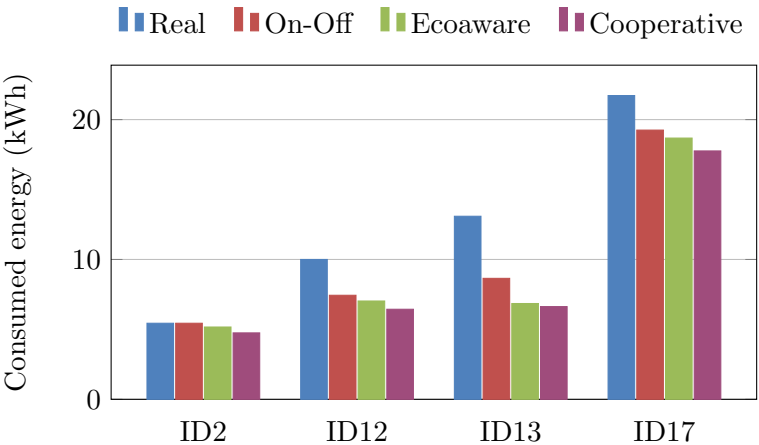


Figura 5.7: Consumo energético de las 4 cafeteras de la categoría *Ecoaware* en cada uno de los modos.

En los gráficos de las figuras 5.8 y 5.9 se aprecia de forma notoria que los modos *Ecoaware* y *Cooperative* adquieren el comportamiento del modo *On-Off*, ya que para estas máquinas de café es el modo más eficiente en términos energéticos.

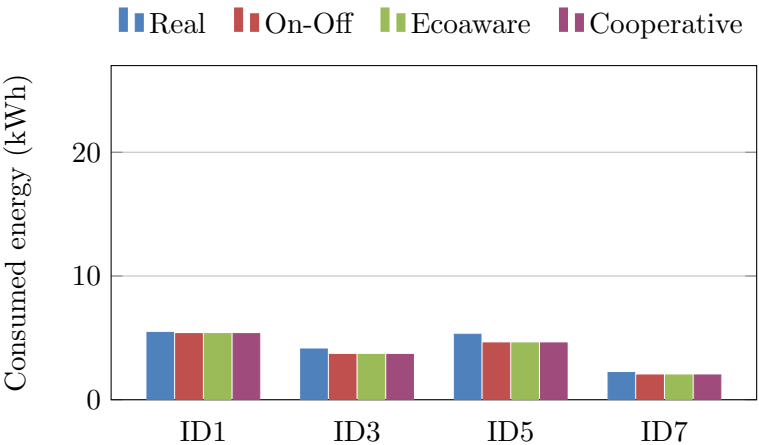


Figura 5.8: Consumo energético de 4 cafeteras de la categoría *On-Off* en cada uno de los modos.

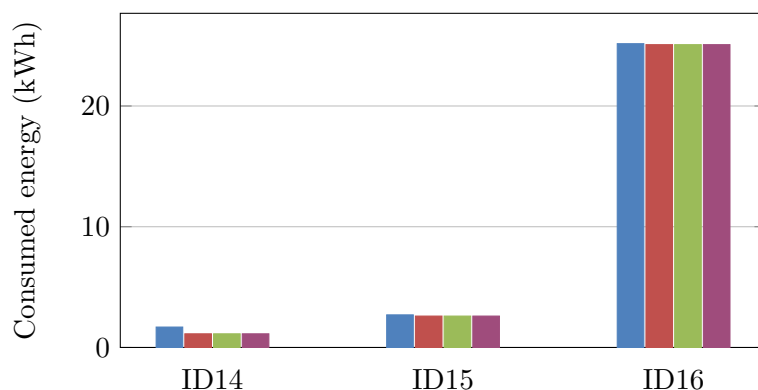


Figura 5.9: Consumo energético de las 3 cafeteras restantes de la categoría *On-Off* en cada uno de los modos.

En aras de entender con mayor claridad lo que suponen estas cantidades de ahorro energético, se van a presentar algunas equivalencias de energía.

Sumando el consumo energético de todas las cafeteras, se obtiene que el consumo real de todas ellas ha sido 110.13 kWh durante 20 días de funcionamiento¹. Si hubieran funcionado durante ese período en modo *Ecoaware* sin datos previos, con ellos, o en modo *Cooperative*, el consumo energético habría sido 91.84, 88.61 u 89.56 kWh respectivamente.

Haciendo referencia a los datos correspondientes al modo *Ecoaware* con datos previos (es el modo de funcionamiento propuesto una vez completado el período inicial en modo *Cooperative*), el ahorro total de energía durante los 20 días es 21.52 kWh. Si se extrapola este dato a un año completo (5 días laborables semanales), el consumo energético sería 258.24 kWh menor, lo que supone un ahorro anual medio por cafetera de 23.48 kWh. A simple vista, puede parecer una cantidad sin importancia, pero si la multiplicamos por los millones de cafeteras que hay instaladas en espacios compartidos a lo largo del mundo, la cifra adquiere

¹Para calcular el consumo energético de las cafeteras *ID12* e *ID14* durante 20 días se ha aplicado una regla de tres directa.

re una magnitud importante. Sin ir más lejos, un estudio² del año 2014 reveló que el número de empresas vivas en España, una vez descartados los empresarios individuales, asciende a 1.435.263. No es posible saber la cantidad de máquinas de café que hay en esas empresas, pero hay que tener en cuenta que en algunas de ellas habrá varias de ellas, mientras que en otras no habrá ninguna. No parece descabellado pensar que hay un millón de cafeteras ubicadas en espacios de trabajo en este país. Por lo tanto, el ahorro total alcanzaría los 23 millones de kWh. Según la agencia de protección del medioambiente de Estados Unidos³(EPA), el CO_2 emitido para generar esta cantidad de energía es equivalente al emitido por el consumo de casi 7 millones de litros de gasolina o por la electricidad de más de 2200 hogares durante un año.

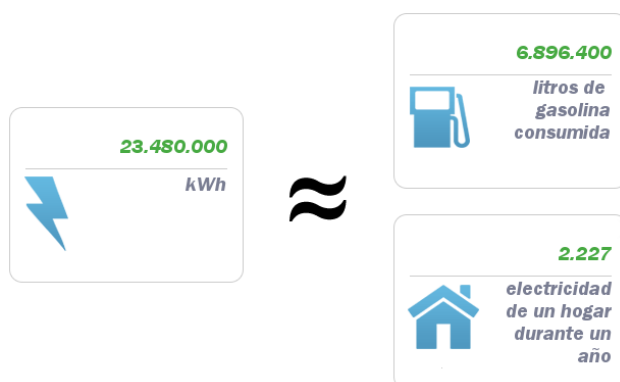


Figura 5.10: Equivalencias de emisiones de CO_2 (1).

A continuación se va a evaluar el ahorro energético de las máquinas de café en modo *Cooperative* con respecto a si hubieran funcionado en modo *Ecoaware* durante el primer mes de funcionamiento, es decir sin disponer de datos de uso propios previos.

²<http://www.empresaactual.com/2014%2D05%2D14%2Dcuantas%2Dempresas%2Dhay%2Den%2Despana/>

³<http://www.epa.gov/energy/greenhouse%2Dgas%2Dequivalencies%2Dcalculator>

Dado que en máquinas cuyo funcionamiento óptimo sea el modo *On-Off*, el modo *Cooperative* no presenta ahorro energético sobre el modo *Ecoaware*, se han desarrollado los cálculos teniendo en cuenta los 4 dispositivos (36 % del total) de la categoría *Ecoaware*. El ahorro de energía total entre ambos modos durante el primer mes (20 días laborables) es 2.28 kWh, lo que se traduce en 0.57 kWh por dispositivo. De nuevo, analizando este dato de forma unitaria, se puede considerar insignificante. Sin embargo, si este ahorro se produce en el 36 % de las cafeteras que hay en las empresas de este país, el consumo energético se reduciría en 205.200 kWh. Según la EPA, el CO_2 emitido para su generación se corresponde con el emitido por el consumo de 60.000 litros de gasolina, por la electricidad de 20 hogares durante un año o por el cultivo de 3628 árboles durante 10 años.



Figura 5.11: Equivalencias de emisiones de CO_2 (2).

5.3 Plataforma hardware

Por último, en esta sección se va a medir el rendimiento en términos energéticos del módulo hardware incorporado a cada una de las cafeteras que forman parte del experimento. Esta medición tiene como objetivo comprobar que el consumo energético derivado de esta plataforma hardware no excede el ahorro energético debido a los nuevos modos de funcionamiento propuestos (*Ecoaware* y *Cooperative*) en esta tesis doctoral.

En primer lugar el análisis se va a centrar en la plataforma que se ha utilizado en el experimento, a sabiendas de que esta no se caracteriza por su eficiencia energética, sino por sus características funcionales. Todas las mediciones se han hecho haciendo uso de una fuente programable de *Agilent Technologies* (ver figura 5.12).



Figura 5.12: Fuente programable Agilent Technologies N6705B.

Las mediciones se han dividido en tres fases:

- *Fase de arranque:* corresponde al período en el que el hardware se alimenta y se pone en marcha.
- *Fase de captura de datos:* se corresponde con el período en el que el hardware está leyendo datos del sensor de corriente.
- *Fase de envío de datos:* es la etapa en la que se envían los datos al servidor a través de la interfaz Ethernet.

Fase de arranque

El impacto que tiene esta fase en el consumo energético total de la placa es despreciable, ya que solo se produce al arrancar el dispositivo y ante algún reinicio inesperado (p. ej., un corte de alimentación) que se pudiera producir.

Como se muestra en la figura 5.13, en el período de arranque se produce un pico de corriente, lo que conlleva un mayor consumo energético durante menos de 4 segundos.

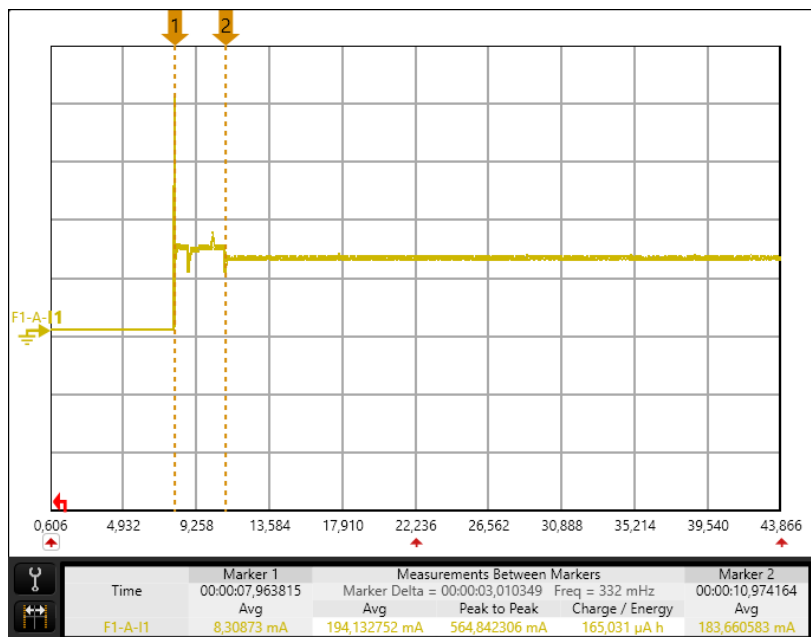


Figura 5.13: Consumo del hardware en la fase de arranque.

Fase de captura de datos

El consumo energético que se produce en esta fase es el que más repercusión tiene sobre el total. Alrededor del 99 % del tiempo, el dispositivo se encuentra leyendo datos del sensor analógico de corriente.

En el gráfico de la figura 5.14, se muestra que el consumo de corriente es de 130 mA constantes a lo largo del tiempo. Por lo tanto, considerando una alimentación de 7 V, supone un consumo diario aproximado de 0.022 kWh.

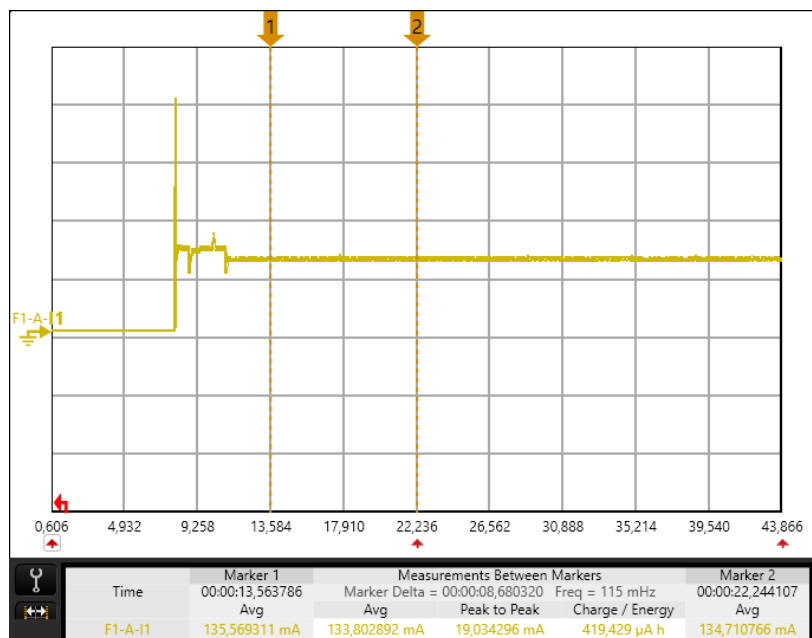


Figura 5.14: Consumo del hardware en la fase de captura de datos.

Fase de envío de datos

En función de la magnitud del consumo energético de esta fase, se puede variar la frecuencia en el envío de los datos.

La figura 5.15 confirma que el envío de datos no supone un aumento significativo del consumo energético, por lo que se ha procedido a enviar los datos en tiempo real.

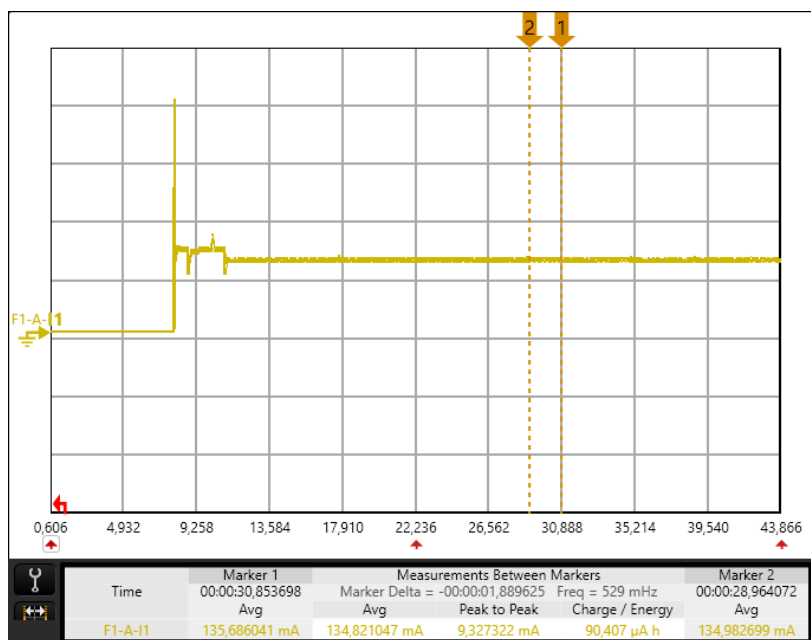


Figura 5.15: Consumo del hardware en la fase de envío de datos.

5.3.1 Análisis y conclusiones

Haciendo referencia a las observaciones expuestas, se asume que el consumo energético de la plataforma hardware utilizada se resume únicamente en el consumo que se desprende de la fase de captura de datos, es decir 0.022 kWh diarios. En la tabla 5.5 se presenta lo que significa ese consumo respecto al ahorro energético diario en cada uno de los espacios. En algunos casos (*ID12*, *ID13*, *ID14* e *ID17*) ese consumo no supera el 20 %. Sin embargo, en otros (*ID1*, *ID7*, *ID15* e *ID16*) es mayor el consumo del hardware que el ahorro energético de los modos propuestos en esta tesis doctoral.

Identificador	Ahorro energético Ecoaware vs Real (kWh/día)	'Consumo hw' respecto 'Ahorro energético'
ID1	0,005	440 %
ID2	0,036	61 %
ID3	0,022	100 %
ID5	0,035	63 %
ID7	0,010	220 %
ID12	0,324	7 %
ID13	0,323	7 %
ID14	0,112	19 %
ID15	0,008	275 %
ID16	0,004	550 %
ID17	0,198	11 %

Tabla 5.5: Comparativa entre cafeteras de la proporción de consumo energético del hardware respecto al ahorro energético producido por los modos propuestos.

Tras unos resultados en los que se considera que la plataforma hardware consume demasiado en proporción con el beneficio obtenido, se plantea recalcular los resultados haciendo uso de la plataforma hardware a medida que se ha descrito en la sección 4.3.1. Con este cambio de componentes y teniendo en cuenta que el relé solo estará activado cuando la máquina de café deba estar encendida (se estima un tiempo de 4 horas diarias, que en ningún caso se ha alcanzado en la experimentación), los 130 mA de consumo de corriente se convierten, según las hojas de especificaciones, en 30 μ A con el relé desactivado y en 3.2 mA con él ac-

tivado. Por lo tanto, los 0.022 kWh diarios pasan a ser $9,5 \times 10^{-5}$ kWh y la tabla 5.5 quedaría de la siguiente forma (tabla 5.6):

Espacio	Ahorro energético Ecoaware vs Real (kWh/día)	'Consumo hw' respecto 'Ahorro energético'
ID1	0,005	1,9 %
ID2	0,036	0,26 %
ID3	0,022	0,43 %
ID5	0,035	0,27 %
ID7	0,010	0,95 %
ID12	0,324	0,029 %
ID13	0,323	0,029 %
ID14	0,112	0,085 %
ID15	0,008	1,18 %
ID16	0,004	2,37 %
ID17	0,198	0,048 %

Tabla 5.6: Comparativa entre cafeteras de la proporción de consumo energético del nuevo hardware respecto al ahorro energético producido por los modos propuestos.

Tras el cambio de hardware, se puede asumir que el consumo del mismo es despreciable, ya que la proporción de la energía que se consume respecto a la que se ahorra es en 8 de los casos menor a un 1 %, mientras que en los 3 restantes, aunque mayor al 1 %, no supera el 2.37 %. Puede surgir la duda sobre la fiabilidad de los datos de las hojas de especificaciones, pero incluso si el consumo del hardware fuera el doble, en ninguno de los casos la proporción superaría el 5 % sobre el ahorro energético logrado.

6

Conclusiones y trabajo futuro

No sería en absoluto extraño si la historia llegara a la conclusión de que el perfeccionamiento de la bicicleta es el incidente más grande del siglo XIX.

—ANÓNIMO.

6.1 Conclusiones

EL DESARROLLO de esta tesis doctoral pretende mejorar la eficiencia energética de los dispositivos provistos del modo *Standby*, concretamente de los ubicados en espacios comunes, en los que un uso energéticamente eficiente se antoja difícil de conseguir.

Para ello, el objetivo es dotar a dichos dispositivos de la capacidad de:

- (a) aprender automáticamente un modelo eficiente de funcionamiento (a través del modo *Ecoaware*) y ,
- (b) comunicarse, de forma que puedan compartir entre ellos sus modos de funcionamiento, posibilitando un aprendizaje más rápido —

disminución del problema de arranque en frío — (a través del modo *Cooperative*).

Así, partiendo de las hipótesis expuestas en la sección 1.2:

H_A: *Una conmutación automática e inteligente del modo de funcionamiento de los dispositivos eléctricos, se traduce en una reducción significativa de su consumo energético, y*

H_B: *Una colaboración entre dispositivos eléctricos basada en la compartición de sus patrones de uso, se traduce en una aceleración del proceso de ahorro energético, lo que conlleva una reducción significativa de su consumo energético durante sus primeras semanas de uso,*

se propusieron una serie de objetivos con el fin de validar dichas hipótesis. A continuación, en la tabla 6.1, se presentan estos objetivos indicando si se han cumplido totalmente (✓), parcialmente (⊙) o no se han cumplido (×), y de haberse cumplido, en qué sección está explicado.

Objetivos	✓ × ⊙
A.1: definición del proceso que se va a llevar a cabo para analizar el funcionamiento de un electrodoméstico a nivel energético. Se debe medir el consumo de energía en el período de tiempo inmediato a alimentar con corriente el dispositivo, cuando se encuentra funcionando en modo <i>Standby</i> y cuando lo está haciendo en modo <i>On</i> desarrollando la función principal para la que ha sido diseñado.	⊙
A.2: diseño y desarrollo de un sistema, hardware y software, que debe medir y almacenar todas las acciones de consumo energético de un dispositivo. Este objetivo conlleva la definición del modelo de datos que permite describir cuantitativamente y cualitativamente la energía consumida por los dispositivos a lo largo del tiempo. A su vez, se ha de determinar el sistema de almacenamiento a utilizar. (Sección 4.3)	✓

Objetivos	✓ × ○
A.3: definición y aplicación del método de aprendizaje automático que permite obtener una mayor precisión en la predicción de uso del dispositivo (Sección 4.5). Es necesario analizar diferentes métodos para discernir cuál es el que mejor se ajusta a la problemática (?).	✓
A.4: validación de los módulos hardware y software desarrollados para comprobar que su funcionamiento se adecua a lo que se ha propuesto. (Sección 5.2)	✓
A.5: evaluación de la solución integral que propone cambiar el modo de funcionamiento de los dispositivos en función de su patrón de uso. Se mide cuantitativamente el ahorro energético de los dispositivos funcionando en modo <i>Ecoaware</i> y se compara respecto a los modos de funcionamiento existentes (<i>On-Off</i> , <i>Standby</i> , <i>Auto-power Down</i>). ((?) y sección 5.2)	✓
B.1: definición de la arquitectura que debe englobar los dispositivos y los flujos de intercambio de datos entre los mismos. (Sección 3.3)	✓
B.2: determinación del algoritmo de similitud entre dispositivos. Para ello es necesario realizar un estudio del estado del arte de los algoritmos ya existentes y evaluar si alguno de ellos se ajusta a la problemática o es necesario definir uno nuevo que se adecue a la misma. Una parte esencial es el modelado del perfil de los dispositivos, ya que el grado de similitud entre dos objetos depende directamente de su perfil. (Sección 2.6 y 3.5.1)	✓
B.3: desarrollo del algoritmo de similitud definido anteriormente y su validación para evaluar su funcionamiento de acuerdo a los requisitos que se establezcan. (Sección 3.5)	✓

Objetivos	✓ × ⊗
B.4: definir el modelo de comunicación entre los dispositivos. El primer paso debe ser definir qué información se quiere intercambiar y qué flujo de información se va a producir. Una vez realizada esta tarea, se ha de valorar tanto la idoneidad de los diferentes <i>middleware</i> existentes para atender las necesidades que se plantean, como la posibilidad de crear una solución adaptada. (Sección 2.5 y 4.4.3)	✓
B.5: desarrollo de la infraestructura de comunicación determinada anteriormente y su validación para determinar que responde a los requisitos de la solución demandada. (Sección 4.4.3)	✓
B.6: evaluación de la solución colaborativa expuesta en esta tesis doctoral para comprobar las ventajas y limitaciones de la propuesta planteada. Se mide cuantitativamente el ahorro energético de los dispositivos funcionando en modo <i>Cooperative</i> y se compara respecto al modo de funcionamiento <i>Ecoaware</i> . (Sección 5.2)	✓
Tabla 6.1: Estado de cumplimiento de los objetivos definidos en la sección 1.2 y en qué artículo o sección se cumplen.	

El objetivo A.1, señalado como cumplido parcialmente, hace referencia a la definición de un proceso para analizar el funcionamiento de un electrodoméstico a nivel energético. Haciendo una revisión bibliográfica, se ha observado que Nipkow (?) definió un método para medir el consumo de electricidad de una máquina de café en cada uno de sus modos de operación. Habiendo tomado una cafetera como elemento principal del caso de uso de esta tesis doctoral, se ha utilizado este método como guía. Dicho documento es susceptible de ser adaptado con el fin de medir el consumo energético de otros dispositivos. En esta tesis doctoral no se ha profundizado en definir un método general, ya que cada dispositivo tiene unas particularidades diferentes.

Como muestra la tabla 6.1, se ha diseñado y desarrollado un sistema

software y hardware que permite almacenar los consumos energéticos de un dispositivo (objetivos A.2 y A.4); se ha determinado y aplicado el método de aprendizaje a utilizar para poder predecir el uso de los dispositivos *Ecoaware* (objetivo A.3); y se ha integrado todo para que el dispositivo, la máquina de café en este caso, de forma autónoma adapte su modo de funcionamiento a la predicción de uso calculada semanalmente.

Mediante la ejecución de la tarea que se plantea en el objetivo A.5, se ha verificado si la hipótesis H_A es cierta o no. Los resultados presentados en la sección 5.2.1 indican que es válida, ya que haciendo uso del modo *Ecoaware*, el consumo energético disminuye en mayor o menor medida en el 91 % de los casos observados, no aumentando en el caso restante. Por lo tanto, con la realización de esta tesis doctoral, se concluye que una conmutación automática e inteligente del modo de funcionamiento de los dispositivos eléctricos, se traduce en una reducción significativa de su consumo energético.

Por otra parte, se ha diseñado una arquitectura en la que los dispositivos *Ecoaware* se comunican a través de la red social Twitter mediante mensajes directos (objetivos B.1, B.4 y B.5); y se ha seleccionado y adaptado el coeficiente de similitud de Gower para discernir qué dispositivo de los ya existentes tiene un patrón de uso más parecido al del nuevo objeto que se va a instalar (objetivos B.2 y B.3).

La ejecución de la tarea que se extrae del objetivo B.6 es la que permite determinar si la hipótesis H_B es válida o no. Tal y como se puede observar en la sección 5.2.1, hay indicios de que esta hipótesis pueda ser válida, ya que de nuevo se produce un ahorro energético en el 91 % de los casos. Sin embargo, se considera que la muestra de 11 cafeteras que finalmente han formado parte de la investigación llevada a cabo en esta tesis doctoral, no es lo suficientemente amplia para poder generalizar los ajustes realizados para obtener el grado de similitud entre dispositivos a un alto porcentaje de las cafeteras distribuidas por el mundo. Por lo tanto, tampoco se puede afirmar con rotundidad que la hipótesis H_B sea válida. Para poder hacerlo sería imprescindible un extenso despliegue de módulos de captura de datos energéticos en máquinas de café de uso compartido. Por el motivo que se acaba de señalar, esta segunda par-

te de la investigación se considera una fase exploratoria que, como luego se presenta en la sección 6.4, abre nuevas vías de investigación.

Otro de los aspectos a destacar en este punto está relacionado con la falta de propuestas similares en la literatura. Durante la revisión bibliográfica no se han encontrado trabajos en los que se propongan modos dinámicos e inteligentes de funcionamiento como alternativa al modo *Standby* o *Auto-power Down*, por lo tanto, la comparación se ha realizado con estos dos modos y con el modo *On-Off*.

6.2 Contribuciones de la tesis

A continuación, se presentan las principales contribuciones de esta tesis doctoral.

- Creación de un nuevo modo de funcionamiento (modo *Ecoaware*), mediante el cual un dispositivo aprende automáticamente un modelo energéticamente eficiente de funcionamiento y conmuta dinámicamente de modo en función del mismo.
- Diseño y desarrollo de un módulo hardware y software que captura datos energéticos, los almacena y ejecuta el modo *Ecoaware* cambiando los modos de funcionamiento de los dispositivos de forma automática e inteligente.
- Diseño y desarrollo de una variante del coeficiente de similitud de Gower, que se ha denominado coeficiente de *wGower*, en el que las variables cualitativas se dividen en dos categorías: las tradicionales y las relacionales.
- Definición de una arquitectura que engloba dispositivos *Ecoaware* y habilita la compartición de datos de uso entre los mismos para afrontar el problema de arranque en frío en sistemas de aprendizaje.

6.3 Publicaciones

El desarrollo y ejecución de esta tesis doctoral ha dado lugar a la publicación de varios artículos de investigación que se presentan a continuación.

6.3.1 Revistas indexadas en JCR

- Juan López-de-Armentia, Diego Casado-Mansilla, Diego López-de-Ipiña. *Making social networks a means to save energy*. Journal of Network and Computer Applications, ISSN 1084-8045, Volume 59, pp. 237-246, 2016. Impact Factor (2014): 2.229, Q1 (Computer Science: Artificial Intelligence, Interdisciplinary Applications).
<http://dx.doi.org/10.1016/j.jnca.2014.09.020>
- Juan López-de-Armentia, Diego Casado-Mansilla, Sergio López-Pérez, Diego López-de-Ipiña. *Reducing energy waste through eco-aware everyday things*. Journal on Mobile Information Systems, ISSN 1574-017x, Volume 10, Issue 1, pp. 79-103, 2014. Impact Factor (2014): 0.949, Q3 (Computer Science: Hardware & Architecture, Interdisciplinary Applications, Software Engineering).
<http://dx.doi.org/10.3233/MIS-130172>
- Diego Casado-Mansilla, Juan López-de-Armentia, Daniela Ventura, Pablo Garaizar, Diego López-de-Ipiña. *Embedding Intelligent Eco-aware Systems within Everyday Things to Increase People's Energy Awareness*. Soft Computing Journal, ISSN 1432-7643, Volume 20, Issue 5, pp. 1695-1711, 2016. Impact Factor (2014): 1.271, Q3 (Computer Science: Information Systems; Telecommunications).
<http://dx.doi.org/10.1007/s00500-015-1751-0>

6.3.2 Conferencias internacionales

- Daniela Ventura, Diego Casado-Mansilla, Juan López-de-Armentia, Pablo Garaizar, Diego López-de-Ipiña, Vincenzo Catania. *ARIIMA: A Real IoT Implementation of a Machine-Learning Architecture for Reducing Energy Consumption*. In Proceedings of 8th International Conference on Ubiquitous Computing and Ambient Intelligence - UCAml2014. Lecture Notes in Computer Science Volume 8867, 2014, pp. 444-451. December, 2014 – Belfast, Ireland.

http://dx.doi.org/10.1007/978-3-319-13102-3_72.

- Juan López-de-Armentia, Diego Casado-Mansilla, Diego López-de-Ipiña. *Saving Energy Through Collaborative Eco-aware Everyday Things*. In proceedings of The Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (Extending Seamlessly to the Internet of Things - esIoT2013 - workshop), pp. 489-493. July, 2013 – Taichung, Taiwan.

<http://dx.doi.org/10.1109/IMIS.2013.88>

- Diego Casado-Mansilla, Juan López-de-Armentia, Diego López-de-Ipiña. *Will Eco-aware Objects Help to Save the World?*. In Proceedings of 6th International Conference on Ubiquitous Computing and Ambient Intelligence - UCAml2012. Lecture Notes in Computer Science 2012, pp. 298-305. ISBN: 978-3-642-35376-5. December, 2012 – Vitoria-Gasteiz, Spain.

https://dx.doi.org/10.1007/978-3-642-35377-2_3

- Juan López-de-Armentia, Diego Casado-Mansilla, Diego López-de-Ipiña. *Fighting against Vampire Appliances through Eco-aware Things*. In proceedings of The Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (Extending Seamlessly to the Internet of Things - esIoT2012 - workshop), pp. 868-873. July, 2012 – Palermo, Italy.

<http://dx.doi.org/10.1109/IMIS.2012.112>

6.3.3 Otras publicaciones relacionadas

- Diego Casado-Mansilla, Juan López-de-Armentia, Pablo Garaizar, Diego López-de-Ipiña. *Team up with Eco-aware Everyday Things to Green your Workplace!*. In Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 409-414. July, 2014 – Birmingham, UK.

<http://dx.doi.org/DOI%2010.1109/IMIS.2014.55>.

- Diego Casado-Mansilla, Juan López-de-Armentia, Pablo Garaizar, Diego López-de-Ipiña. *To switch off the coffee-maker or not: that is the question to be energy-efficient at work*. In Extended Abstracts on Human Factors in Computing Systems (CHI 2014), pp. 2425-2430. April, 2014 – Toronto, Canada.

<http://dx.doi.org/10.1145/2559206.2581152>

6.4 Trabajo futuro

Tras la realización de esta tesis doctoral, se han identificado ciertas limitaciones y oportunidades de mejora que abren nuevas líneas de trabajo con el fin de dar continuidad a la propuesta realizada. A continuación se exponen algunas de estas líneas.

- Añadir nuevas variables a tener en cuenta en la predicción, tales como la meteorología, estaciones del año o cargas de trabajo. Se intuye que puede existir una relación entre el patrón de uso de las cafeteras y estas variables (p. ej. a mayor carga de trabajo, mayor número de cafés preparados). Para ello, es imprescindible disponer de datos de uso de un largo período de tiempo.
- Aumentar el tamaño y diversidad de la muestra de máquinas de café. Esto permitirá, por una parte, un ajuste de pesos más preciso

en la aplicación del coeficiente de similitud de *wGower*. Y por otra parte, permitirá que aumenten las posibilidades de que una nueva cafetera encuentre otra con un perfil similar.

- Habilitar los dispositivos para que una vez transcurrido un período de tiempo en funcionamiento, ofrezcan al sistema un determinado coeficiente de similitud que indique en qué grado se parecen el patrón de uso adquirido y el suyo propio. De esta forma, se podrían excluir patrones atípicos y mejorar así el proceso de búsqueda de un dispositivo cuyo modo de uso sea similar.
- Extender los modos, tanto *Ecoaware* como *Cooperative*, a otros dispositivos distintos, tales como sistemas de iluminación, máquinas expendedoras, fotocopadoras o paneles de publicidad en las calles. Para cada uno de ellos habría que realizar un proceso para conocer el comportamiento energético de cada uno de ellos. A continuación, habría que realizar unos cálculos similares a los que se presentan en la sección 3.2. Además, como se ha indicado en la sección 3.5.1, las características de cada tipo de dispositivo determinarán el modelado de sus perfiles.
- Optimización, montaje y evaluación de la solución hardware a medida *Low Energy* presentada en la parte final de la sección 4.3.1.
- Diseño y desarrollo de una solución *Plug-and-Play*, que facilite la instalación del hardware en los dispositivos.



Modelado de la información energética en la BBDD

Código A.1: Representación en la BBDD del consumo derivado de la realización de un café.

```
1 {  
2   "_id": "70e462030a214aebf462a41724d0a9ed",  
3   "_rev": "1-5554e64024dd55f2a58960c4baef8714",  
4   "energy_consumption_Wh": 14.41,  
5   "consumption_type": "COFFEE",  
6   "consumption_time_in_secs": 48.23,  
7   "deviceID": "ID1",  
8   "device_type": "COFFEE-MAKER",  
9   "time": "14:38:16",  
10  "date": "2016-05-04"  
11 }
```

Código A.2: Representación en la BBDD del consumo derivado del encendido de la cafetera.

```
1 {  
2   "_id": "70e462030a214aebf462a41724d5fgg58",  
3   "_rev": "1-f6a1d2d4bf980109096d412ca2af17c6",  
4   "energy_consumption_Wh": 7.21,  
5   "consumption_type": "START_TIME",  
6   "consumption_time_in_secs": 20.07,  
7   "deviceID": "ID3",  
8   "device_type": "COFFEE-MAKER",  
9   "time": "08:19:56",  
10  "date": "2016-05-12"  
11 }
```

Código A.3: Representación en la BBDD de un consumo en modo *Standby*.

```
1 {  
2   "_id": "70e462030a214aebf462a41724d05e9h",  
3   "_rev": "1-1846f5d11c882ce6efd485c96073cceb",  
4   "energy_consumption_Wh": 0.39,  
5   "consumption_type": "STANDBY",  
6   "consumption_time_in_secs": 2.07,  
7   "deviceID": "ID3",  
8   "device_type": "COFFEE-MAKER",  
9   "time": "09:23:38",  
10  "date": "2016-05-13"  
11 }
```

B

Cálculo de consumo energético

Este anexo presenta los *script* desarrollados en lenguaje *Python* para calcular los consumos energéticos de los dispositivos en los diferentes modos, para luego compararlos con el consumo real, que es el punto de referencia.

Código B.1: Cálculo de consumo energético en modo *On-Off*.

```
1 def getConsumptionsOnOff(dbView, database):
2     consumptionType = ''
3     consumptionWh = 0.0
4     rowID = 0
5     identifier = 0
6     energyTemp = 0.0
7     dateTemp = ''
8     timeTemp = ''
9     index = 0
10    STEnergy = 0.0
11    STEnergyTemp = 0.0
12    totalStartTimes = 0
13    nDatesAdded = 0
14    lastCoffee = datetime.datetime.now()
15    lastWasST = False
16    global nDays
17
```

```

18  ### Reset de lists
19  del on_offAccumulated[:]
20  del energyConsumption[:]
21  del on_off[:]
22
23  for itemDate in dates:
24      date1 = datetime.datetime.strptime(itemDate.split('/')
          [0], "%Y-%m-%d").date()
25      date2 = datetime.datetime.strptime(itemDate.split('/')
          [1], "%Y-%m-%d").date()
26      nDaysAux = date2 - date1
27      nDays = nDaysAux.days + 1
28      nDatesAdded = 0
29      for _ in range(len(energyConsumption), len(
          energyConsumption)+nDays):
30          relDate = date1 + datetime.timedelta(days=nDatesAdded
          )
31          if ((relDate.weekday()!=5) and (relDate.weekday()!=6)
          and (relDate not in holidays)):
32              energyConsumption.append({'id': index, 'date':
          relDate, 'wastedEnergy': 0.0, 'stEnergy': 0.0,
          'effectiveEnergy': 0.0, 'numCoffees':0})
33              index += 1
34              nDatesAdded +=1
35
36  ### lastDate stores the more recent day ww are going to
compute
37  lastDate = datetime.datetime.strptime(dates[len(dates)
          -1].split('/')[1], "%Y-%m-%d").date()
38
39  for row in database.view(dbView, descending=False): ###
          We use a couchdb's view
40      info = database.get(row.id)
41      consumptionType = info[typeKey]
42      consumptionWh = info[consumptionKey]
43      dateTemp = info[dateKey]
44      timeTemp = info[timeKey]
45      dateTime = dateTemp + " " + timeTemp
46
47      hour = int(timeTemp[0:2])
48      dDate = datetime.datetime.strptime(dateTemp, "%Y-%m-%d"
          ).date()
49      if dDate > lastDate:

```

```

50     break
51
52     if (any (d['date'] == dDate for d in energyConsumption)
53         ) and (dDate.weekday() != 5) and (dDate.weekday() != 6)
54         and (dDate not in holidays):
55     for lines in energyConsumption:
56         if (dDate == lines['date']):
57             rowID = int(lines['id'])
58             break
59     if(consumptionType == "COFFEE"):
60         if lastWasST:
61             lastWasST = False
62             energyTemp = float(energyConsumption[rowID]['
63                 effectiveEnergy']) + float(consumptionWh)
64             identifier = energyConsumption[rowID]['id']
65             timedeltaLastCoffee = datetime.datetime.strptime(
66                 dateTime, "%Y-%m-%d %H:%M:%S") - lastCoffee
67             if timedeltaLastCoffee.seconds > secsBetweenSB:
68                 energyConsumption[rowID]=({'id': identifier, '
69                     date': energyConsumption[rowID]['date'], '
70                     wastedEnergy': energyConsumption[rowID]['
71                     wastedEnergy'], 'stEnergy': energyConsumption
72                     [rowID]['stEnergy'], 'effectiveEnergy':
73                     energyTemp, 'numCoffees': energyConsumption[
74                     rowID]['numCoffees']+1})
75             elif (0<timedeltaLastCoffee.seconds < secsBetweenSB
76                 ):
77                 energyConsumption[rowID]=({'id': identifier, '
78                     date': energyConsumption[rowID]['date'], '
79                     wastedEnergy': energyConsumption[rowID]['
80                     wastedEnergy'], 'stEnergy': energyConsumption
81                     [rowID]['stEnergy'], 'effectiveEnergy':
82                     energyTemp, 'numCoffees': energyConsumption[
83                     rowID]['numCoffees']})
84             lastCoffee = datetime.datetime.strptime(dateTime, "
85                 %Y-%m-%d %H:%M:%S")
86
87 for energies in energyConsumption:
88     if len(on_offAccumulated)>0:
89         ### ON/OFF MODE CONSUMPTION
90         on_off.append(energies['effectiveEnergy'] + energies[
91             'numCoffees']*ST_consumption)
92     on_offAccumulated.append(on_offAccumulated[-1] +

```

```
        on_off[-1])
74     else:
75         ### ON/OFF MODE CONSUMPTION
76         on_off.append(energies['effectiveEnergy'] + energies[
            'numCoffees']*ST_consumption)
77         on_offAccumulated.append(on_off[-1])
```

Código B.2: Cálculo de consumo energético en modo *Autopower-Down*.

```

1 def getConsumptionsAPD(dbView, database):
2     consumptionType = ''
3     consumptionWh    = 0.0
4     rowID             = -1
5     identifier       = 0
6     energyTemp       = 0.0
7     dateTemp         = ''
8     timeTemp         = ''
9     index            = 0
10    lastCoffee        = datetime.datetime.now()
11    lastWasST         = False
12    lastWasSB         = False
13    STEnergy          = 0.0
14    STEnergyTemp      = 0.0
15    nStartTimes       = 0
16    nStartTimesAPD    = 0
17    totalStartTimes   = 0
18    dayBefore         = datetime.date.today()
19    nDatesAdded       = 0
20    nSB_duringAPD     = 0
21    global nDays
22
23    ### Reset de lists
24    del energyConsumption[:]
25    del autopowerAccumulated[:]
26    del autopower[:]
27
28    for itemDate in dates:
29        date1 = datetime.datetime.strptime(itemDate.split('/')[-1], "%Y-%m-%d").date()
30        date2 = datetime.datetime.strptime(itemDate.split('/')[0], "%Y-%m-%d").date()
31        nDaysAux = date2 - date1
32        nDays = nDaysAux.days + 1
33        nDatesAdded = 0
34        for i in range(len(energyConsumption), len(
35            energyConsumption)+nDays):
36            relDate = date1 + datetime.timedelta(days=nDatesAdded)
37            if ((relDate.weekday()!=5) and (relDate.weekday()!=6)
38                and (relDate not in holidays)):
39                energyConsumption.append({'id': index, 'date':

```

```

        relDate, 'wastedEnergy': 0.0, 'stEnergy': 0.0,
        'effectiveEnergy': 0.0, 'numST':0})
38     index += 1
39     nDatesAdded +=1
40
41
42     ### lastDate stores the more recent day ww are going to
compute
43     lastDate = datetime.datetime.strptime(dates[len(dates)
        -1].split('/')[1], "%Y-%m-%d").date()
44
45     for row in database.view(dbView, descending=False): ###
        We use a couchdb's view
46         info = database.get(row.id)
47         consumptionType = info[typeKey]
48         consumptionWh = info[consumptionKey]
49         dateTemp = info[dateKey]
50         timeTemp = info[timeKey]
51         dateTime = dateTemp + " " + timeTemp
52
53         dDate = datetime.datetime.strptime(dateTemp, "%Y-%m-%d"
        ).date()
54         if dDate > lastDate:
55             if nStartTimesAPD>0:
56                 energyConsumption[rowID]=({'id': identifier, 'date'
                    : energyConsumption[rowID]['date'], '
                    wastedEnergy': energyConsumption[rowID]['
                    wastedEnergy'], 'stEnergy': energyConsumption[
                    rowID]['stEnergy'] + nSB_duringAPD*
                    SB_consumption, 'effectiveEnergy':
                    energyConsumption[rowID]['effectiveEnergy'], '
                    numST': energyConsumption[rowID]['numST'] +
                    nStartTimesAPD + 1})
57             else:
58                 energyConsumption[rowID]=({'id': identifier, 'date'
                    : energyConsumption[rowID]['date'], '
                    wastedEnergy': energyConsumption[rowID]['
                    wastedEnergy'], 'stEnergy': energyConsumption[
                    rowID]['stEnergy'] + nSB_duringAPD*
                    SB_consumption, 'effectiveEnergy':
                    energyConsumption[rowID]['effectiveEnergy'], '
                    numST': energyConsumption[rowID]['numST'] +
                    nStartTimesAPD})

```

```

59     break
60
61     if (any (d['date'] == dDate for d in energyConsumption)
        ) and (dDate.weekday() != 5) and (dDate.weekday() != 6)
        and (dDate not in holidays):
62         if dDate != dayBefore:
63             dayBefore = dDate
64             if rowID != -1:
65                 if nStartTimesAPD > 0:
66                     energyConsumption[rowID] = ({'id': identifier, '
                        date': energyConsumption[rowID]['date'], '
                        wastedEnergy': energyConsumption[rowID]['
                        wastedEnergy'], 'stEnergy':
                        energyConsumption[rowID]['stEnergy'] +
                        nSB_duringAPD * SB_consumption, '
                        effectiveEnergy': energyConsumption[rowID][
                        'effectiveEnergy'], 'numST':
                        energyConsumption[rowID]['numST'] +
                        nStartTimesAPD + 1})
67                 else:
68                     energyConsumption[rowID] = ({'id': identifier, '
                        date': energyConsumption[rowID]['date'], '
                        wastedEnergy': energyConsumption[rowID]['
                        wastedEnergy'], 'stEnergy':
                        energyConsumption[rowID]['stEnergy'] +
                        nSB_duringAPD * SB_consumption, '
                        effectiveEnergy': energyConsumption[rowID][
                        'effectiveEnergy'], 'numST':
                        energyConsumption[rowID]['numST'] +
                        nStartTimesAPD})
69             nStartTimesAPD = 0
70             nSB_duringAPD = 0
71         for lines in energyConsumption:
72             if (dDate == lines['date']):
73                 rowID = int(lines['id'])
74                 break
75         if (consumptionType == "COFFEE"):
76             timedeltaLastCoffee = datetime.datetime.strptime(
                dateTime, "%Y-%m-%d %H:%M:%S") - lastCoffee
77             lastCoffee = datetime.datetime.strptime(dateTime, "
                %Y-%m-%d %H:%M:%S")
78             if timedeltaLastCoffee.seconds > APD_secs:
79                 nStartTimesAPD += 1

```

```

80         else:
81             nSB_duringAPD += timedeltaLastCoffee.seconds/
                        secsBetweenSB
82             energyTemp = float(energyConsumption[rowID]['
                        effectiveEnergy']) + float(consumptionWh)
83             identifier = energyConsumption[rowID]['id']
84             energyConsumption[rowID]=({'id': identifier, 'date'
                        : energyConsumption[rowID]['date'], '
                        wastedEnergy': energyConsumption[rowID]['
                        wastedEnergy'], 'stEnergy': energyConsumption[
                        rowID]['stEnergy'], 'effectiveEnergy':
                        energyTemp, 'numST': energyConsumption[rowID]['
                        numST']})
85
86     for energies in energyConsumption:
87         if len(autopowerAccumulated)>0:
88             autopower.append(energies['effectiveEnergy'] +
                        energies['numST']*(ST_consumption+SB_consumption*
                        nSB))
89             autopowerAccumulated.append(autopowerAccumulated[-1]
                        + autopower[-1])
90         else:
91             autopower.append(energies['effectiveEnergy'] +
                        energies['numST']*(ST_consumption+SB_consumption*
                        nSB))
92             autopowerAccumulated.append(autopower[-1])

```

Código B.3: Cálculo de consumo energético en modo *Ecoaware*.

```

1 def getConsumptionsArima(dbView, database):
2     hourSB          = [] # Contiene las franjas horarias que
        debe estar en Standby
3     consumptionType = ''
4     consumptionWh   = 0.0
5     rowID           = -1
6     identifier      = 0
7     energyTemp      = 0.0
8     dateTemp        = ''
9     timeTemp        = ''
10    secsTemp        = 0.0
11    index            = 0
12    STEnergy         = 0.0
13    STEnergyTemp     = 0.0
14    totalStartTimes  = 0
15    nDatesAdded      = 0
16    totalSecCoffees  = 0.0
17    dayBefore        = datetime.date.today()
18    lastCoffee       = datetime.datetime.now()
19    firstCoffeeMade  = False
20    global hourSB
21    indexDates       = 0
22    coffeePeriods    = 0
23    nCoffees         = 0
24    firstCoffeeTime  = 0
25    global nDays
26    week             = 0
27    weekAux          = -1
28
29    ### Reset de lists
30    del arima[:]
31    del arimaAccumulated[:]
32    del energyConsumption[:]
33
34    for itemDate in dates:
35        date1 = datetime.datetime.strptime(itemDate.split('/')[
36            [0], "%Y-%m-%d").date()
37        date2 = datetime.datetime.strptime(itemDate.split('/')[
38            [1], "%Y-%m-%d").date()
39        nDaysAux = date2 - date1
40        nDays = nDaysAux.days + 1
41        nDatesAdded = 0

```

```

40     for _ in range(len(energyConsumption), len(
41         energyConsumption)+nDays):
42         relDate = date1 + datetime.timedelta(days=nDatesAdded
43         )
44         if ((relDate.weekday()!=5) and (relDate.weekday()!=6)
45             and (relDate not in holidays)):
46             energyConsumption.append({'id': index, 'date':
47                 relDate, 'energy': 0.0, 'numCoffees':0})
48             index += 1
49             nDatesAdded +=1
50     ### lastDate stores the more recent day we are going to
51     compute
52     lastDate = datetime.datetime.strptime(dates[len(dates)
53         -1].split('/')[1], "%Y-%m-%d").date()
54     for row in database.view(dbView, descending=False): ###
55         We use a couchdb's view
56         info = database.get(row.id)
57         consumptionType = info[typeKey]
58         consumptionWh = info[consumptionKey]
59         dateTemp = info[dateKey]
60         timeTemp = info[timeKey]
61         secsTemp = info[secsKey]
62         dateTime = dateTemp + " " + timeTemp
63         coffeePeriods = 0
64         hour = int(timeTemp[0:2])
65         minute = int(timeTemp[3:5])
66
67         dDate = datetime.datetime.strptime(dateTemp, "%Y-%m-%d"
68             ).date()
69
70         if dDate > lastDate:
71             for h in hourSB[0]:
72                 if not (h['nullCoffees']):
73                     coffeePeriods += 1
74             if coffeePeriods > 0:
75                 energyTemp = coffeePeriods*ST_consumption + (((
76                     coffeePeriods*3600 - totalSecCoffees -
77                     firstCoffeeTime*60 - nCoffees*secsBetweenSB*2)/
78                     secsBetweenSB)-1)*SB_consumption
79                 energyConsumption[rowID]=({'id': identifier, 'date'
80                     : energyConsumption[rowID]['date'], 'energy':

```

```

        energyConsumption[rowID]['energy']+energyTemp,
        'numCoffees': energyConsumption[rowID]['
        numCoffees']})
71     break
72
73
74     if (any (d['date'] == dDate for d in energyConsumption)
        ) and (dDate.weekday() != 5) and (dDate.weekday() != 6)
        and (dDate not in holidays):
75         week = getWeek(dDate)
76         if week != weekAux:
77             getHourSB(dDate, predictions)
78             hourSB = hourSBFixed[:]
79             weekAux = week
80         if dDate != dayBefore:
81             dayBefore = dDate
82             if rowID != -1:
83                 for h in hourSB[0]:
84                     if not (h['nullCoffees']):
85                         coffeePeriods += 1
86                     if coffeePeriods > 0:
87                         energyTemp = coffeePeriods*ST_consumption + (((
                            coffeePeriods*3600 - totalSecCoffees -
                            firstCoffeeTime*60 - nCoffees*secsBetweenSB
                            *2)/secsBetweenSB)-1)*SB_consumption
88                         energyConsumption[rowID]={'id': identifier, '
                            date': energyConsumption[rowID]['date'], '
                            energy': energyConsumption[rowID]['energy'
                            ]+energyTemp, 'numCoffees':
                            energyConsumption[rowID]['numCoffees']})
89                         totalSecCoffees = 0
90                         nCoffees = 0
91                         coffeePeriods = 0
92                         firstCoffeeTime = 0
93                         hourSB = hourSBFixed[:]
94                     for lines in energyConsumption:
95                         if (dDate == lines['date']):
96                             rowID = int(lines['id'])
97                             break
98                 if any(d['h'] == hour for d in hourSB[0]):
99                     if (consumptionType == "COFFEE"):
100                         if ({'h':hour,'nullCoffees':True} in hourSB[0]):
101                             index = hourSB[0].index({'h':hour,'nullCoffees'

```

```

:True}))
102     hourSB[0][index] = {'h':hour,'nullCoffees':
        False}
103     firstCoffeeTime += minute
104     totalSecCoffees += float(secsTemp)
105     nCoffees += 1
106     energyTemp = float(energyConsumption[rowID]['
        energy']) + float(consumptionWh)
107     identifier = energyConsumption[rowID]['id']
108     energyConsumption[rowID]=({'id': identifier, '
        date': energyConsumption[rowID]['date'], '
        energy': energyTemp, 'numCoffees':
        energyConsumption[rowID]['numCoffees']})
109     else:
110         if(consumptionType == "COFFEE"):
111             energyTemp = float(energyConsumption[rowID]['
                energy']) + float(consumptionWh)
112             identifier = energyConsumption[rowID]['id']
113             timedeltaLastCoffee = datetime.datetime.strptime(
                dateTime, "%Y-%m-%d %H:%M:%S") - lastCoffee
114             if timedeltaLastCoffee.seconds > secsBetweenSB:
115                 energyConsumption[rowID]=({'id': identifier, '
                    date': energyConsumption[rowID]['date'], '
                    energy': energyTemp, 'numCoffees':
                    energyConsumption[rowID]['numCoffees']+1})
116             elif (0<timedeltaLastCoffee.seconds <
                secsBetweenSB):
117                 energyConsumption[rowID]=({'id': identifier, '
                    date': energyConsumption[rowID]['date'], '
                    energy': energyTemp, 'numCoffees':
                    energyConsumption[rowID]['numCoffees']})
118                 lastCoffee = datetime.datetime.strptime(dateTime,
                    "%Y-%m-%d %H:%M:%S")
119
120     for energies in energyConsumption:
121         if len(arimaAccumulated)>0:
122             arima.append(energies['energy'] + energies['
                numCoffees']*ST_consumption)
123             arimaAccumulated.append(arimaAccumulated[-1] + arima
                [-1])
124         else:
125             arima.append(energies['energy'] + energies['
                numCoffees']*ST_consumption)

```

126	<code>arimaAccumulated.append(arima[-1])</code>
-----	---

Código B.4: Cálculo de consumo energético en modo *Cooperative*.

```

1 def getConsumptionsCoop(dbView, database):
2     hourSBCoop          = [] # Contiene las franjas horarias
        que debe estar en Standby
3     consumptionType = ''
4     consumptionWh    = 0.0
5     rowID            = -1
6     identifier       = 0
7     energyTemp       = 0.0
8     dateTemp        = ''
9     timeTemp        = ''
10    secsTemp         = 0.0
11    index            = 0
12    STEnergy         = 0.0
13    STEnergyTemp     = 0.0
14    totalStartTimes  = 0
15    nDatesAdded      = 0
16    totalSecCoffees   = 0.0
17    dayBefore        = datetime.date.today()
18    lastCoffee        = datetime.datetime.now()
19    firstCoffeeMade   = False
20    global hourSBCoop
21    indexDates        = 0
22    coffeePeriods     = 0
23    nCoffees          = 0
24    firstCoffeeTime   = 0
25    global nDays
26    week = 0
27    weekAux = -1
28
29    ### Reset de lists
30    del coop[:]
31    del coopAccumulated[:]
32    del energyConsumption[:]
33
34    for itemDate in dates:
35        date1 = datetime.datetime.strptime(itemDate.split('/')[
36            [0], "%Y-%m-%d").date()
37        date2 = datetime.datetime.strptime(itemDate.split('/')[
38            [1], "%Y-%m-%d").date()
39        nDaysAux = date2 - date1
40        nDays = nDaysAux.days + 1
41        nDatesAdded = 0

```

```

40     for _ in range(len(energyConsumption), len(
41         energyConsumption)+nDays):
42         relDate = date1 + datetime.timedelta(days=nDatesAdded
43             )
44         if ((relDate.weekday()!=5) and (relDate.weekday()!=6)
45             and (relDate not in holidays)):
46             energyConsumption.append({'id': index, 'date':
47                 relDate, 'energy': 0.0, 'numCoffees':0})
48             index += 1
49             nDatesAdded +=1
50     ### lastDate stores the more recent day we are going to
51     compute
52     lastDate = datetime.datetime.strptime(dates[len(dates)
53         -1].split('/')[1], "%Y-%m-%d").date()
54
55     for row in database.view(dbView, descending=False): ###
56         We use a couchdb's view
57         info = database.get(row.id)
58         consumptionType = info[typeKey]
59         consumptionWh = info[consumptionKey]
60         dateTemp = info[dateKey]
61         timeTemp = info[timeKey]
62         secsTemp = info[secsKey]
63         dateTime = dateTemp + " " + timeTemp
64         coffeePeriods = 0
65         hour = int(timeTemp[0:2])
66         minute = int(timeTemp[3:5])
67
68         dDate = datetime.datetime.strptime(dateTemp, "%Y-%m-%d"
69             ).date()
70
71         if dDate > lastDate:
72             for h in hourSBCoop[0]:
73                 if not(h['nullCoffees']):
74                     coffeePeriods += 1
75             if coffeePeriods > 0:
76                 energyTemp = coffeePeriods*ST_consumption + (((
77                     coffeePeriods*3600 - totalSecCoffees -
78                     firstCoffeeTime*60 - nCoffees*secsBetweenSB*2)/
79                     secsBetweenSB)-1)*SB_consumption
80                 energyConsumption[rowID]=({'id': identifier, 'date'
81                     : energyConsumption[rowID]['date'], 'energy':

```

```

        energyConsumption[rowID]['energy']+energyTemp,
        'numCoffees': energyConsumption[rowID]['
        numCoffees']})
71     break
72
73
74     if (any (d['date'] == dDate for d in energyConsumption)
        ) and (dDate.weekday() != 5) and (dDate.weekday() != 6)
        and (dDate not in holidays):
75         week = getWeek(dDate)
76         if week != weekAux:
77             getHourSB(dDate, predictionsCoop)
78             hourSBCoop = hourSBFixed[:]
79             weekAux = week
80         if dDate != dayBefore:
81             dayBefore = dDate
82             if rowID != -1:
83                 for h in hourSBCoop[0]:
84                     if not (h['nullCoffees']):
85                         coffeePeriods += 1
86                 if coffeePeriods > 0:
87                     energyTemp = coffeePeriods*ST_consumption +(((
                        coffeePeriods*3600 - totalSecCoffees -
                        firstCoffeeTime*60 - nCoffees*secsBetweenSB
                        *2)/secsBetweenSB)-1)*SB_consumption
88                     energyConsumption[rowID] = ({'id': identifier, '
                        date': energyConsumption[rowID]['date'], '
                        energy': energyConsumption[rowID]['energy'
                        ]+energyTemp, 'numCoffees':
                        energyConsumption[rowID]['numCoffees']})
89                     totalSecCoffees = 0
90                     nCoffees = 0
91                     coffeePeriods = 0
92                     firstCoffeeTime = 0
93                     hourSBCoop = hourSBFixed[:]
94                 for lines in energyConsumption:
95                     if (dDate == lines['date']):
96                         rowID = int(lines['id'])
97                         break
98             if any(d['h'] == hour for d in hourSBCoop[0]):
99                 if (consumptionType == "COFFEE"):
100                     if ({'h': hour, 'nullCoffees': True} in hourSBCoop
                        [0]):

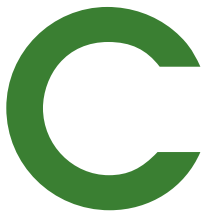
```

```

101         index = hourSBCoop[0].index({'h':hour,'
102             nullCoffees':True})
103         hourSBCoop[0][index] = {'h':hour,'nullCoffees':
104             False}
105         firstCoffeeTime += minute
106         totalSecCoffees += float(secsTemp)
107         nCoffees += 1
108         energyTemp = float(energyConsumption[rowID]['
109             energy']) + float(consumptionWh)
110         identifier = energyConsumption[rowID]['id']
111         energyConsumption[rowID]=({'id': identifier, '
112             date': energyConsumption[rowID]['date'], '
113             energy': energyTemp, 'numCoffees':
114             energyConsumption[rowID]['numCoffees']})
115     else:
116     if(consumptionType == "COFFEE"):
117         energyTemp = float(energyConsumption[rowID]['energy
118             ']) + float(consumptionWh)
119         identifier = energyConsumption[rowID]['id']
120         timedeltaLastCoffee = datetime.datetime.strptime(
121             dateTime, "%Y-%m-%d %H:%M:%S") - lastCoffee
122         if timedeltaLastCoffee.seconds > secsBetweenSB:
123             energyConsumption[rowID]=({'id': identifier, '
124                 date': energyConsumption[rowID]['date'], '
125                 energy': energyTemp, 'numCoffees':
126                 energyConsumption[rowID]['numCoffees']+1})
127         elif (0<timedeltaLastCoffee.seconds < secsBetweenSB
128             ):
129             energyConsumption[rowID]=({'id': identifier, '
130                 date': energyConsumption[rowID]['date'], '
131                 energy': energyTemp, 'numCoffees':
132                 energyConsumption[rowID]['numCoffees']})
133         lastCoffee = datetime.datetime.strptime(dateTime, "
134             %Y-%m-%d %H:%M:%S")
135
136     for energies in energyConsumption:
137         if len(coopAccumulated)>0:
138             coop.append(energies['energy'] + energies['numCoffees
139                 ']*ST_consumption)
140             coopAccumulated.append(coopAccumulated[-1] + coop
141                 [-1])
142         else:
143             coop.append(energies['energy'] + energies['numCoffees

```

```
        ']*ST_consumption)  
126    coopAccumulated.append(coop[-1])
```



Archivo de configuración de los dispositivos

Código C.1: Archivo de configuración del dispositivo ID1.

```
# Name of your device. It will be used for
# identification (e.g. ID1, ID2, ID3)
[identifier]
deviceID = ID1

# DHCP service state. Write 'true' if your dhcp
# service is enabled in your network, else write 'false'.
[network]
mac = 01:23:45:67:89:FF
dhcpEnabled = true

# Static ip configuration. Fill the fields only if
# dhcp service is "not enabled" in your local network.
[static]
ip = 10.48.1.200
gateway = 10.48.1.254
netmask = 255.255.254.0
dnsServer = 130.206.100.2

# Ip to retrieve Unix time from a NTP server.
# Set to another server if it fails.
[NTPserver]
```

```
ntpIP = 172.16.250.11

# Preditions by day of the week.
# The data is filled every week.
[predictions]
monday      = 000110001000
tuesday     = 000110001000
wednesday   = 000100001000
thursday    = 000110011000
friday      = 001110010000
```



Registro en Twitter y compartición de datos

D.1 Autenticación en Twitter

Código D.1: Autenticación en Twitter.

```
1 def authentication(consumer_key, consumer_secret,  
    access_token, access_token_secret):  
2     auth = tweepy.OAuthHandler(consumer_key, consumer_secret)  
3     auth.set_access_token(access_token, access_token_secret)  
4     api = tweepy.API(auth)  
5  
6     return api
```

D.2 Registro automático en Twitter

Código D.2: Registro automático en Twitter.

```
1 def registration(profile):
2     br = mechanize.Browser()
3     br.open("https://twitter.com/signup")
4     br.select_form(predicate=lambda f: f.attrs.get('id', None)
5         == 'phx-signup-form')
6     br.form['user[name]'] = profile.id + " " + deviceType
7     br.form['user[email]'] = 'ecoawarethings+twitterid@gmail.
8         com'
9     br.form['user[user_password]'] = '#####'
10
11 def updateProfile(screen_name, location, description):
12     api.update_profile(screen_name, location, description)
```

D.3 Suscripción a lista de dispositivos

Código D.3: Suscripción a lista de dispositivos.

```
1 def subscribeList(api):
2     listToReturn = []
3     api.subscribe_list('juanarmentia', 'coffeeMakers-list')
4     list_subscribers = api.list_subscribers('juanarmentia',
5         'coffeeMakers-list')
6     myId = api.me()._json['id']
7     for suscriber in list_subscribers:
8         if suscriber.id != myId:
9             api.create_friendship(suscriber.id)
10             user = suscriber.screen_name + "|" + suscriber.
11                 location + "|" + suscriber.description
12             listToReturn.append(user)
13
14     return listToReturn
```

D.4 Intercambio de información entre dispositivos

Código D.4: Petición de los datos de uso a otro dispositivo.

```
1 def getPattern(api, nDays, idTwitter):
2     dm = api.send_direct_message(screen_name=idTwitter, text="
        Could you send me the last "+str(nDays)+" days' usage
        data?")
3     api.destroy_direct_message(dm.id)
```

Código D.5: Lectura de la petición a través de un mensaje directo.

```
1 def readDirectMessages(api):
2     today = datetime.datetime.now().strftime('%Y-%m-%d')
3     direct_messages = api.direct_messages(today)
4     msg = direct_messages[0].text
5     nDays = msg.replace("Could you send me the last ", "")
6     nDays = nDays.replace(" days' usage data?", "")
7     data = {'idTwitter': direct_messages[0].sender.
            screen_name, 'nDays':int(nDays), 'idMsg':
            direct_messages[0].id}
8
9     return data
```

Código D.6: Envío de los datos de uso a otro dispositivo.

```
1 def sendPattern(api, pattern, idTwitter, idMsg):
2     dm = api.send_direct_message(screen_name=idTwitter, text=
        pattern)
3     api.destroy_direct_message(idMsg)
4     api.destroy_direct_message(dm.id)
```

Código D.7: Lectura del patrón recibido a través de un mensaje directo.

```
1 def readPattern(api):  
2     today = datetime.datetime.now().strftime('%Y-%m-%d')  
3     direct_messages = api.direct_messages(today)  
4     pattern = direct_messages[0].text  
5     api.destroy_direct_message(direct_messages[0].id)  
6  
7     return pattern
```



Cálculo de la predicción de uso

E.1 Cálculo de ARIMA

Código E.1: Código para calcular la predicción de uso en base a datos de uso previos.

```
1 require(forecast)
2 require(stats)
3
4 # Read the csv file
5 cpdata = read.csv("20days.csv",header=F)
6 # Create the time series with frequency 12 (7am-7pm)
7 # We're going to create a learning time of 20 days so 20*12
  = 240 (2400 lines of in the csv)
8 cpd = ts(cpdata[1:240,1],start=1,freq=12)
9 summary(cpd)
10
11 fit <- Arima(cpd, order=c(3,1,1), seasonal=c(2,0,2))
12 summary(fit)
13
14 # We perform the prediction for the next 5 days (Mon-Fry)
  12*5-1 = 59
15 frcast <- forecast(fit,h=59)
16 summary(frcast)
17
```

```
18 #Show the prediction for h=12
19 jpeg("./Images/serieTemporalForecast.jpg",width=1024,height
      =600)
20 plot(frcast)
21 graphics.off()
22 # write in a new csv the prediction with the confidence
   intervals
23 write.csv(frcast, "forecast.csv", row.names = FALSE)
```

F

Modelos teóricos de las funciones ACF y PACF

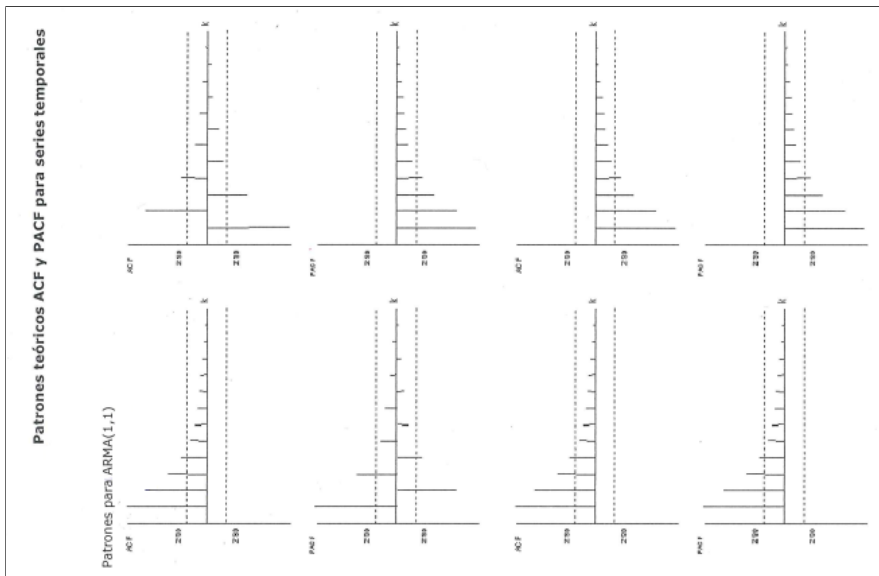


Figura F.1: Patrones teóricos ACF y PACF para series temporales (1).

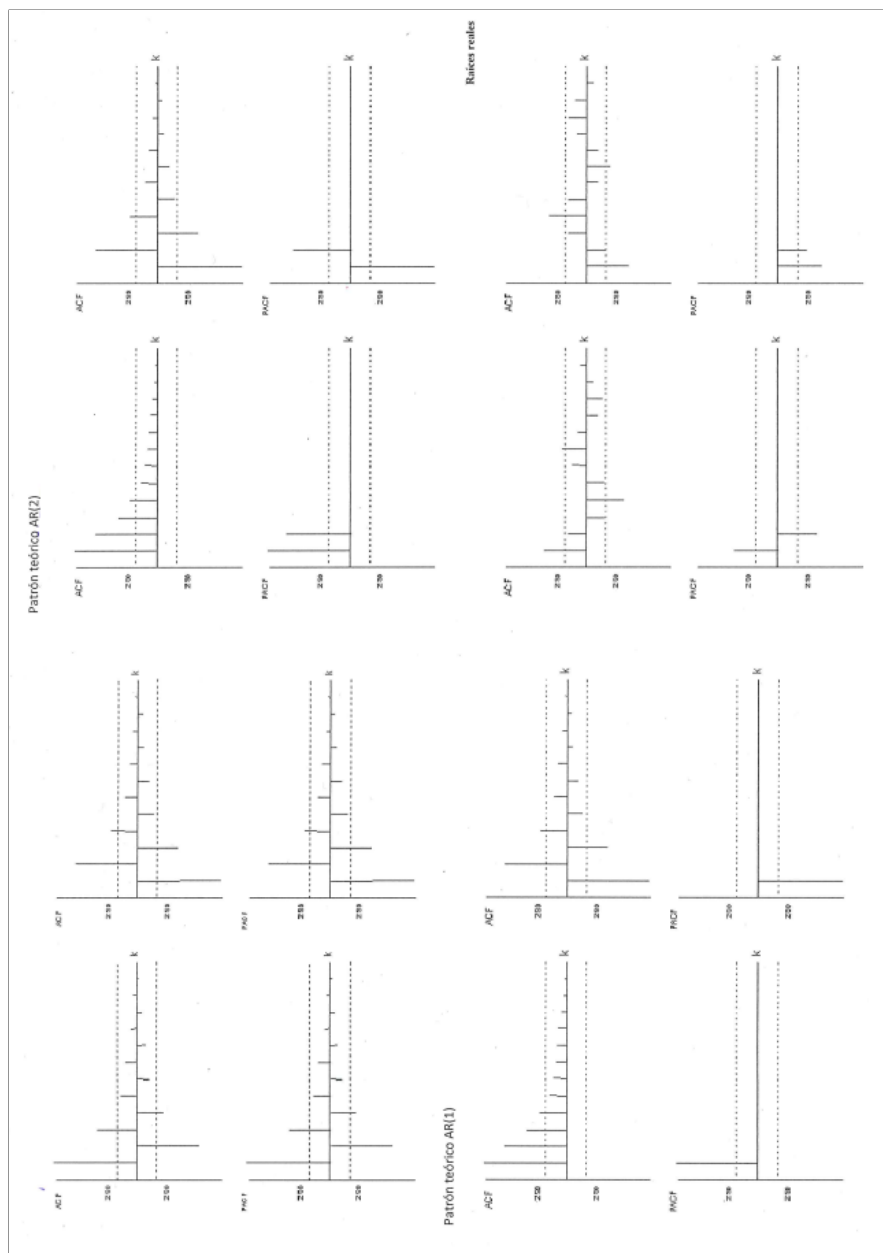


Figura F.2: Patrones teóricos ACF y PACF para series temporales (2).

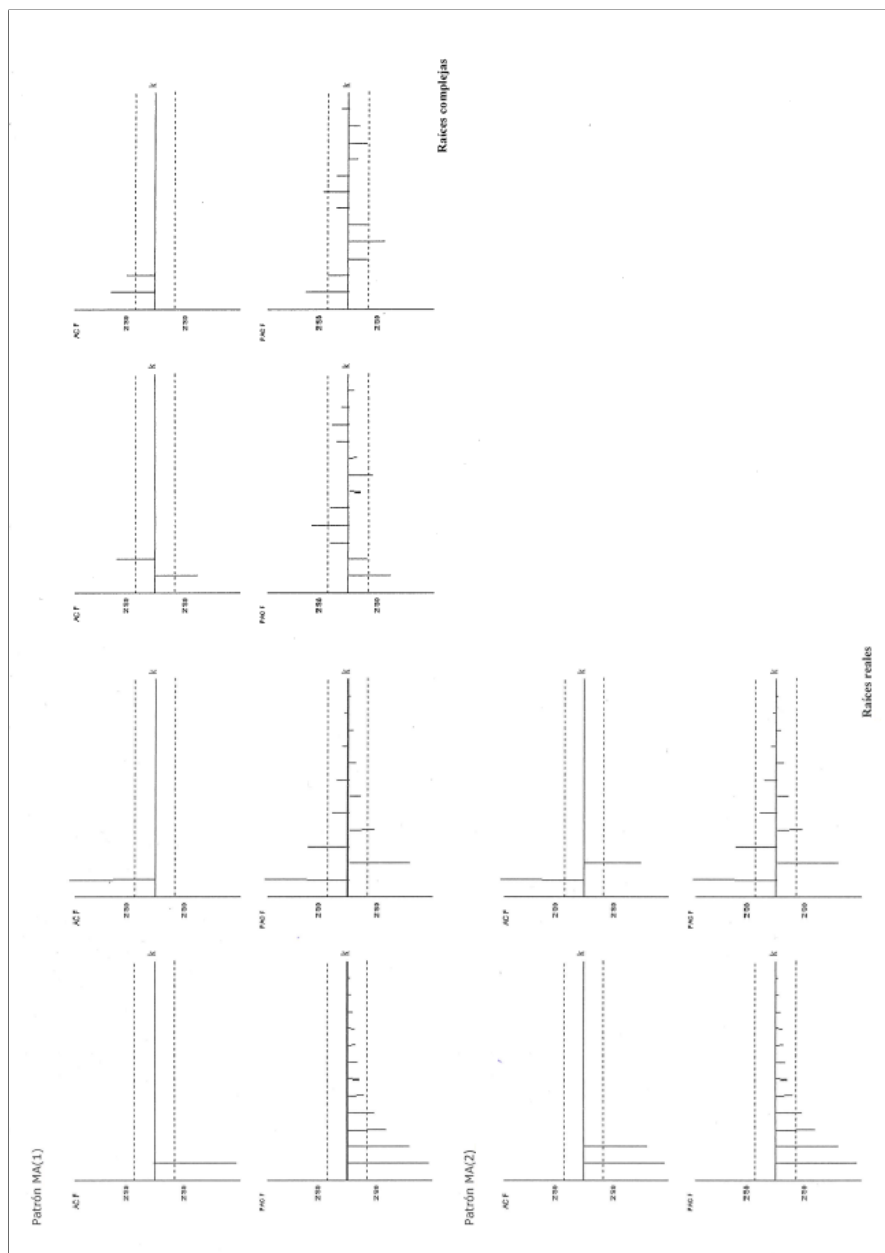


Figura F.3: Patrones teóricos ACF y PACF para series temporales (3).

Bibliografía

- Adebiyi, A. A., Adewumi, A. O., and Ayo, C. K. (2014). Comparison of arima and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics*, 2014.
- Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723.
- Atzori, L., Iera, A., and Morabito, G. (2011). Siot: Giving a social structure to the internet of things. *Communications Letters, IEEE*, 15(11):1193–1195.
- Bandyopadhyay, D. and Sen, J. (2011). Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58(1):49–69.
- Bandyopadhyay, S., Sengupta, M., Maiti, S., and Dutta, S. (2011). Role of middleware for internet of things: A study. *International Journal of Computer Science & Engineering Survey (IJCSES)*, 2(3):94–105.
- Berners-Lee, T., Fischetti, M., and Foreword By-Dertouzos, M. L. (2000). *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*. HarperInformation.

- Bleecker, J. (2006). A manifesto for networked objects—cohabiting with pigeons, arphids and aibos in the internet of things.
- Box, G. E. P. and Jenkins, G. (1990). *Time Series Analysis, Forecasting and Control*, volume Holden-Day, Incorporated. John Wiley & Sons, Inc.
- boyd, d. m. and Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230.
- CarbonTrust (2010). Office based companies. maximising energy savings in an office environment.
- Caruana, R. and Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM.
- Casado-Mansilla, D., López-de-Armentia, J., Ventura, D., Garaizar, P., and López-de-Ipiña, D. (2015). Embedding intelligent eco-aware systems within everyday things to increase people’s energy awareness. *Soft Computing Journal. Springer Berlin Heidelberg*.
- Castro, A. M. and Torres, I. C. (2008). Pronósticos mediante redes neuronales artificiales y modelos arima: el caso de los cetes en México. *Comercio exterior*, 58(11):794–802.
- Cha, M., Haddadi, H., Benevenuto, F., and Gummadi, P. K. (2010). Measuring user influence in twitter: The million follower fallacy. *ICWSM*, 10(10-17):30.
- Choi, S.-S., Cha, S.-H., and Tappert, C. C. (2010). A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1):43–48.
- Ciortea, A., Boissier, O., Zimmermann, A., and Florea, A. M. (2013). Reconsidering the social web of things: position paper. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 1535–1544. ACM.

- Ciortea, A., Boissier, O., Zimmermann, A., and Florea, A. M. (2014). Open and interoperable socio-technical networks. In *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2014 16th International Symposium on*, pages 251–257. IEEE.
- Clifford, H. T., Stephenson, W., et al. (1975). *An introduction to numerical classification*, volume 229. Academic Press New York.
- Comission, T. E. (2013). Regulation 801/2013: Non-tertiary coffee machines (lot 25).
- Costa, P., Mascolo, C., Musolesi, M., and Picco, G. P. (2008). Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 26(5):748–760.
- Costanza, E., Ramchurn, S. D., and Jennings, N. R. (2012). Understanding domestic energy consumption through interactive visualisation: a field study. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 216–225. ACM.
- Czekanowski, J. (1913). Overview over the application of statistical methods for anthropology. *Prace Towarzystwa Naukowego Warszawskiego*, 5.
- Danova, T. (2014). The connected-home report: Forecasts and growth trends for the leading 'internet of things' market.
- Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.
- Ding, L., Shi, P., and Liu, B. (2010). The clustering of internet, internet of things and social network. In *Knowledge Acquisition and Modeling (KAM), 2010 3rd International Symposium on*, pages 417–420. IEEE.
- Eisenhauer, M., Rosengren, P., and Antolin, P. (2010). Hydra: A development platform for integrating wireless devices and sensors into ambient intelligence systems. In *Giusto, D., Iera, A., Morabito, G., and*

- Atzori, L., editors, *The Internet of Things*, pages 367–373. Springer New York.
- Estabrook, G. F. and Rogers, D. J. (1966). A general method of taxonomic description for a computed similarity measure. *BioScience*, 16(11):789–793.
- Everitt, B. S., Landau, S., and Leese, M. (2009). *Cluster Analysis*. Wiley Publishing, 4th edition.
- Fast, A., Jensen, D., and Levine, B. N. (2005). Creating social networks to improve peer-to-peer networking. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 568–573. ACM.
- Fontecha Diezma, J. (2013). Sistema móvil para la detección y valoración del síndrome de fragilidad en el adulto mayor.
- Froehlich, J., Findlater, L., and Landay, J. (2010). The design of eco-feedback technology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1999–2008. ACM.
- Gallego Vázquez, J. A. (2012). *Comunidades Virtuales y Redes Sociales*. Wolters Kluwer.
- Girardello, A. and Michahelles, F. (2010). Twiphone: Sharing communication behavior on twitter. In *IoT for Citizen Workshop@ Pervasive*, volume 10.
- Gómez-Goiri, A. (2014). *Semantic Tuple Spaces for Constrained Devices: A Web-compliant Vision*. Ph.d., University of Deusto, Bilbao, Spain.
- Gordon, A. (1990). Constructing dissimilarity measures. *Journal of Classification*, 7(2):257–269.
- Gower, J. C. (1971). A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871.
- Group, I.-T. S. (2012). Acm turing award honors founders of automatic verification technology.

- Guinard, D. (2011). *A Web of Things Application Architecture – Integrating the Real-World into the Web*. PhD thesis, ETH Zurich, Zurich, Switzerland.
- Guinard, D., Fischer, M., and Trifa, V. (2010). Sharing using social networks in a composable web of things. In *Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2010 8th IEEE International Conference on, pages 702–707. IEEE.
- Guinard, D., Trifa, V., Mattern, F., and Wilde, E. (2011). From the internet of things to the web of things: Resource-oriented architecture and best practices. In Uckelmann, D., Harrison, M., and Michahelles, F., editors, *Architecting the Internet of Things*, pages 97–129. Springer Berlin Heidelberg.
- Hamilton, J. D. (1994). *Time series analysis*, volume 2. Princeton university press Princeton.
- HECSU (2013). Hecsu's type of work categories.
- Höller, J., Tsiatsis, V., Mulligan, C., Karnouskos, S., Avesand, S., and Boyle, D. (2014). *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*. Elsevier.
- Holmquist, L. E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., and Gellersen, H.-W. (2001). Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *Ubicomp 2001: Ubiquitous Computing*, pages 116–122. Springer.
- Irani, D., Webb, S., Pu, C., and Li, K. (2010). Study of trend-stuffing on twitter through text classification. In *Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*.
- Jaccard, P. (1901). Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:241 – 272.
- Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New phytologist*, 11(2):37–50.

- Jahn, M., Jentsch, M., Prause, C., Pramudianto, F., Al-Akkad, A., and Reiners, R. (2010). The energy aware smart home. In *Future Information Technology (FutureTech)*, 2010 5th International Conference on, pages 1–8.
- Janson, S. and Vegelius, J. (1981). Measures of ecological association. *Oecologia*, 49(3):371–376.
- JapanGovernment (2010). Energy in japan.
- Jones, D. A. (1991). Electrical engineering: the backbone of society. In *Science, Measurement and Technology, IEE Proceedings A*, volume 138, pages 1–10. IET.
- Kalekar, P. S. (2004). Time series forecasting using holt-winters exponential smoothing. *Kanwal Rekhi School of Information Technology*, 4329008:1–13.
- Khashei, M. and Bijari, M. (2010). An artificial neural network (p, d, q) model for timeseries forecasting. *Expert Systems with applications*, 37(1):479–489.
- Kim, J. and Lee, J.-W. (2014). Openiot: An open service framework for the internet of things. In *Internet of Things (WF-IoT)*, 2014 IEEE World Forum on, pages 89–93. IEEE.
- Klein, B., Laiseca, X., Casado-Mansilla, D., López-de-Ipiña, D., and Nespral, A. P. (2012). Detection and extracting of emergency knowledge from twitter streams. In *Ubiquitous Computing and Ambient Intelligence*, pages 462–469. Springer.
- Koehler, C., Ziebart, B. D., Mankoff, J., and Dey, A. K. (2013). Therml: Occupancy prediction for thermostat control. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pages 103–112, New York, NY, USA. ACM.
- Kranz, M., Roalter, L., and Michahelles, F. (2010). Things that twitter: social networks and the internet of things. In *What can the Internet of*

- Things do for the Citizen (CloT) Workshop at The Eighth International Conference on Pervasive Computing (Pervasive 2010)*, pages 1–10.
- Kwiatkowski, D., Phillips, P. C., Schmidt, P., and Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of econometrics*, 54(1):159–178.
- Legrende, P. and Legrende, L. (1983). Numerical ecology. *Developments in Environmental Modelling*, 2:853.
- Lewis, D. M. and Janeja, V. P. (2011). An empirical evaluation of similarity coefficients for binary valued data. *IGI Global*, pages 44–66.
- Lim, T. and Khoo, H. (1985). Sampling properties of gower’s general coefficient of similarity. *Ecology*, pages 1682–1685.
- Ljung, G. M. and Box, G. E. (1978). On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303.
- López-de-Armentia, J., Casado-Mansilla, D., López-Pérez, S., and López-de-Ipiña, D. (2014). Reducing energy waste through eco-aware everyday things. *Mob. Inf. Syst.*, 10(1):79–103.
- López-de-Armentia, J., Casado-Mansilla, D., and López-de-Ipiña, D. (2012). Fighting against vampire appliances through eco-aware things. In *The Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 868–873, Palermo, Italy. IEEE Computer Society.
- Madden, S. R., Franklin, M. J., Hellerstein, J. M., and Hong, W. (2005). Tinydb: An acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.*, 30(1):122–173.
- Marti, S., Ganesan, P., and Garcia-Molina, H. (2005). Sprout: P2p routing with social networks. In *Current Trends in Database Technology-EDBT 2004 Workshops*, pages 425–435. Springer.

- McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.
- Mei, A., Morabito, G., Santi, P., and Stefa, J. (2011). Social-aware stateless forwarding in pocket switched networks. In *Infocom, 2011 Proceedings IEEE*, pages 251–255. IEEE.
- Mislove, A., Gummadi, K. P., and Druschel, P. (2006). Exploiting social networks for internet search. In *5th Workshop on Hot Topics in Networks (HotNets06)*. Citeseer, page 79.
- Moreau, G. and Legendre, L. (1979). Relation entre habitat et peuplements de poissons: essai de définition d’une méthode numérique pour des rivières nordiques. *Hydrobiologia*, 67(1):81–87.
- Nau, R. (2014). Statistical forecasting: notes on regression and time series analysis.
- Nazzi, E. and Sokoler, T. (2011). Walky for embodied microblogging: sharing mundane activities through augmented everyday objects. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 563–568. ACM.
- Ning, H. and Wang, Z. (2011). Future internet of things architecture: like mankind neural system or social organization framework? *Communications Letters, IEEE*, 15(4):461–463.
- Nipkow, J. (2009). Measuring method and calculation formula for the electricity consumption of coffee machines for household use. *Top-ten International Group TIG, Paris, France*.
- Ochiai, A. (1957). Zoogeographic studies on the soleoid fishes found in japan and its neighbouring regions. *Bull. Jpn. Soc. Sci. Fish*, 22(9):526–530.
- Perez de Almeida, R. A., Blackstock, M., Lea, R., Calderon, R., do Prado, A. F., and Guardia, H. C. (2013). Thing broker: A twitter for things. In

- Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 1545–1554. ACM.
- Recruiter (2012). Careers and occupations.
- Rish, I. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM New York.
- Rodríguez-Salazar, M. E., Alvarez-Hernández, S., and Bravo-Núñez, E. (2001). *Coeficientes de asociación*. Plaza y Valdés Editores.
- Rogers, D. J. and Tanimoto, T. T. (1960). A computer program for classifying plants. *Science*, 132(3434):1115–1118.
- Rosenberg, N. (1998). The role of electricity in industrial development. *Energy Journal*, 19(2):7–24.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.
- Russell, P. F., Rao, T. R., et al. (1940). On habitat and association of species of anopheline larvae in south-eastern madras. *Journal of the Malaria Institute of India*, 3(1):153–178.
- Schwartz, T., Betz, M., Ramirez, L., and Stevens, G. (2010). Sustainable energy practices at work: Understanding the role of workers in energy conservation. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, pages 452–462, New York, NY, USA. ACM.
- Sokal, R. R., Sneath, P. H., et al. (1963). Principles of numerical taxonomy. *Principles of numerical taxonomy*.
- Sørensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biol. Skr.*, 5:1–34.

- Sundmaeker, H., Guillemin, P., Friess, P., and Woelfflé, S. (2010). Vision and challenges for realising the internet of things.
- Surowiecki, J. (2005). *The wisdom of crowds*. Anchor.
- Tsiatsis, V., Gluhak, A., Bauge, T., Montagut, F., Bernat, J., Bauer, M., Villalonga, C., Barnaghi, P. M., and Krco, S. (2010). The sensei real world internet architecture. In *Future Internet Assembly*, pages 247–256.
- Tsiatsis, V., Höller, J., and Gold, R. (2011). Sensor and actuator resource architecture.
- Ventura, D., Casado-Mansilla, D., López-de-Armentia, J., Garaizar, P., López-de-Ipiña, D., and Catania, V. (2014). ARIIMA: a real IoT implementation of a machine-learning architecture for reducing energy consumption. In *Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services*, pages 444–451. Springer.
- Villa, C. S. M., Rotondi, D., Comolli, M., Houyou, A. M., P., H. H., Kloukinas, C., Trsek, H., and Claessens, J. (2010). Wp 1 – plug&work iot requirement assessment and architecture. Technical report.
- Villalonga, C., Bauer, M., Huang, V., Bernat, J., and Barnaghi, P. (2010). Modeling of sensor data and context for the real world internet. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 1–6. IEEE.
- Vázquez, J. I. and López-de-Ipiña, D. (2008). Social devices: autonomous artifacts that communicate on the internet. In *The Internet of Things*, pages 308–324. Springer.
- Werbos, P. (1974). Beyond regression: New tools for prediction and analysis in the behavioral sciences.
- Widrow, B. et al. (1960). *Adaptive "adaline" Neuron Using Chemical "memistors"*.

- Yang, R. and Newman, M. W. (2013). Learning from a learning thermostat: lessons for intelligent systems for the home. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 93–102. ACM.
- Yu, H., Kaminsky, M., Gibbons, P. B., and Flaxman, A. (2006). Sybilguard: defending against sybil attacks via social networks. *ACM SIGCOMM Computer Communication Review*, 36(4):267–278.
- Zhang, H. (2004). The optimality of naive bayes. *AA*, 1(2):3.

Esta tesis fue terminada de escribir en junio de 2016.

