



UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DOCTORADO EN CIENCIAS DE LA INGENIERÍA CON MENCIÓN EN
INGENIERÍA ELÉCTRICA

Modelo de asignación de recursos en un ambiente inteligente con restricciones energéticas

Profesor supervisor: PhD. Jorge Pezoa Núñez
Departamento de Ingeniería Eléctrica
Facultad de Ingeniería
Universidad de Concepción

Tesis para optar al grado académico de Doctor en Ciencias
de la Ingeniería con mención en Ingeniería Eléctrica

SILVIA ELENA RESTREPO MEDINA
CONCEPCIÓN - CHILE
2016

Prefacio

Esta tesis es presentada como parte de los requisitos para optar al grado académico de Doctor en Ingeniería Eléctrica con mención en Ingeniería Eléctrica, de la Universidad de Concepción, Chile, y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otras. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el Departamento de Ingeniería Eléctrica, durante el período comprendido entre el año 2012 y 2016, bajo la dirección del Doctor Jorge Pezoa Núñez.

Silvia Elena Restrepo Medina
srestrepo@udec.cl
Departamento de Ingeniería Eléctrica
Facultad de Ingeniería
Universidad de Concepción
Concepción, 2016

A mis padres Celina y Alonso....
A mis hermanos Angela y Alejo....
A mi esposo Tavo....



Agradecimientos

Esta tesis se debe a muchas personas y entidades que de una u otra forma aportaron en su exigente producción. Entre ellos, agradezco a:

CONICYT (Comisión Nacional de Investigación Científica y Tecnológica), quien me apoyó para realizar mis estudios de doctorado y favoreció financieramente con la Beca Nacional de Doctorado/2014-63140151 (2014-2016).

El centro de investigación CEFOP y el Departamento de Ingeniería Eléctrica de la Universidad de Concepción por el apoyo académico y financiero brindado para mi trabajo investigativo.

Jorge Pezoa, profesor supervisor de esta tesis, quien con su apoyo constante y enseñanzas académicas, profesionales y personales permitió que este trabajo de investigación fuera motivadora y fructífera.

Además, agradezco a los profesores Demetrio Arturo Ovalle y Sergio Torres sus colaboraciones académicas y personales. Por otra parte, Mahshid R. Naeini, quien con su apoyo durante la pasantía realizada en Texas Tech University (USA) contribuyó significativamente en la comprensión y futuro de este trabajo investigativo.

A mis profesores y compañeros por los valiosos aportes en su realización. También, a mi familia chilena: profe Pezoa, Angely, Don Exe, Glima, Pauli, Andre, Pablo, Vivi, Tebo, Philippe, Marta, Romi, Dardo, Telma, Wallon, Bruna y Guix por el continuo acompañamiento y colaboración en todos los aspectos de mi vida. Gracias por hacerme sentir como en casa.

A mis amigos de siempre, mi hermana putativa Johanna y familiares (incluidos tíos, primos, cuñaditos y suegros), les agradezco por motivarme a cumplir mis metas y ser un apoyo constante. A pesar de la distancia siempre los tengo presente en mi vida.

Finalmente y no menos importantes (“el que ríe de último ríe mejor”), a las personas que son el motor de mi vida y me han ayudado a llegar a este momento de mi vida; a mis padres Alonso y Celina, por el apoyo incondicional, afecto y motivación que me brindan día a día, a mis hermanos Ángela y Alejo por su apoyo fraternal, y a una persona muy especial, Tavo, quien gracias a su compañía ha facilitado mi crecimiento tanto personal como académico, y me impulsa diariamente a lograr más allá de lo que me propongo.



Tabla de Contenido

Índice de figuras	xi
Índice de tablas	xiii
Resumen	xv
1. Introducción	1
1.1. Descripción del problema	2
1.2. Hipótesis	3
1.3. Objetivos	3
1.4. Trabajos previos de la autora de esta tesis	5
1.4.1. Asignación de tareas o recursos	5
1.4.2. Inteligencia Ambiental	5
1.5. Contribuciones de tesis	5
1.6. Organización general de tesis	6
2. Inteligencia Ambiental y Asignación de recursos	7
2.1. Estado del arte	7
2.1.1. Inteligencia Ambiental	7
2.1.2. Asignación de recursos	10
2.2. Conceptos y fundamentos: Inteligencia Ambiental	11
2.2.1. Agentes de Software	12
2.2.2. Redes de Sensores Inalámbricas (WSN)	14
2.3. Conceptos y fundamentos: Asignación de recursos	15
2.4. Modelo de Inteligencia Ambiental basado en la integración de WSN y Agentes inteligentes	17

2.5. Asignación de Imágenes para el procesamiento distribuido de video considerando el consumo energético	20
3. Modelo de Asignación de Servicios	25
3.1. Definición del problema	25
3.2. Descripción del ambiente inteligente	26
3.3. Modelamiento matemático para el problema de Asignación de Servicios	28
3.4. Algoritmo de búsqueda para la Asignación de Servicios	31
3.4.1. Enfoque Proactivo	32
3.4.2. Enfoque reactivo	32
4. Evaluación de Modelo de Asignación de Servicios	37
4.1. Consideraciones de implementación del Modelo de Asignación de Servicios	37
4.2. Evaluación de estrategias de Asignación de servicios	41
4.3. Análisis comparativo	44
4.3.1. Enfoque <i>Greedy</i>	44
4.3.2. Comparación de escenarios	45
5. Conclusiones y Trabajo Futuro	51
5.1. Trabajo futuro	52
Bibliografía	53



Índice de figuras

1.1. Esquema de control del problema de asignación. Fuente: Elaboración propia.	3
1.2. Esquema representativo del ambiente inteligente. Fuente: Elaboración propia.	4
2.1. Arquitectura de la aplicación del laboratorio inteligente [1]. Fuente: Elaboración propia.	19
2.2. Para la secuencia de video 640×480 [2]: (a) Asignación de imágenes por nodo. (c) Energía disponible por nodo. Fuente: Recuperado de [2].	24
3.1. Esquema general de control del problema planteado. Fuente: Elaboración propia.	26
3.2. Representación del ambiente inteligente. Fuente: Elaboración propia.	27
3.3. Justificación de orden topológico de grafo. Fuente: Elaboración propia.	30
4.1. Representación del grafo de servicios del ambiente inteligente. Fuente: MATLAB.	38
4.2. Representación de los subgrafos de servicios asociados a tres usuarios. Fuente: MATLAB.	39
4.3. Representación del subgrafo asociado a nueva solicitud. Fuente: Elaboración propia.	40

4.4. <i>Enfoque proactivo</i> . Tiempo mínimo de respuesta, Asignación de Servicios, y consumo de energía de agentes para el sistema AmI considerado. Fuente: MATLAB.	42
4.5. <i>Enfoque reactivo</i> . Tiempo mínimo de respuesta, Asignación de Servicios, y consumo de energía de agentes para el sistema AmI considerado. Fuente: MATLAB.	43
4.6. <i>Enfoque Greedy</i> . Asignación de Servicios, y consumo de energía de agentes para el sistema AmI considerado. Fuente: MATLAB.	46
4.7. Asignación de Servicios en escenario comparativo. Fuente: MATLAB.	49



Índice de tablas

2.1. Comparación de trabajos relacionados con AmI. Fuente: Elaboración propia.	9
2.2. Tiempo de procesamiento de video para diferentes criterios de asignación. Fuente: Recuperado de [2].	22
3.1. Ejemplos de solicitud de servicios. Fuente: Elaboración propia.	31
4.1. Valores de los parámetros del sistema AmI. Fuente: Elaboración propia.	40
4.2. Tiempo mínimo de respuesta y consumo energético de enfoques proactivo y reactivo. Fuente: Elaboración propia.	41
4.3. Tiempo mínimo de respuesta y consumo energético de enfoque <i>greedy</i> . Fuente: Elaboración propia.	45
4.4. Tiempo mínimo de respuesta y consumo energético de escenario comparativo. Fuente: Elaboración propia.	47
4.5. Evaluación del desempeño de modelo propuesto. Fuente: Elaboración propia.	48

Resumen

La Inteligencia Ambiental o Ambient Intelligence (AmI), en inglés, es un novedoso paradigma computacional cuyo objetivo es proporcionar ambientes que, por un lado, respondan a los usuarios que los habitan y que, por otro, se adapten a diferentes tipos de situaciones. Para lograr esto de una manera amigable con los usuarios, la Inteligencia Ambiental (AmI) debe integrar dispositivos de hardware y entidades software que interactúen de manera transparente con las personas.

Producto de la integración de varias áreas tecnológicas novedosas tales como la computación ubícua, los sistemas adaptivos, las redes de sensores inalámbricos, los agentes inteligentes, entre otros, la AmI se ha convertido en un tema de investigación y desarrollo de mucho interés en los últimos años.

Un sistema AmI que pretenda ser aplicado en situaciones reales debe ser diseñado teniendo en cuenta un cierto contexto y una serie de capacidades que le permitan adaptarse a las condiciones dinámicas del ambiente y, simultáneamente, a los requerimientos de los usuarios. Para lograr esto, es necesario que un sistema AmI sea capaz de abstraer, dinámicamente, el entorno y sus usuarios mediante algún modelo apropiado, y a su vez, es necesario asignar los servicios y coordinar adecuadamente los elementos que están encargados de proporcionar la inteligencia para llegar a una solución acertada en el menor tiempo posible y/o con el menor costo posible.

Esta tesis, se focaliza en el problema de la asignación eficiente de servicios en un ambiente inteligente, donde la satisfacción del usuario está garantizada por la entrega de todos los servicios solicitados por los usuarios. Para esto, se propone un modelo matemático para representar un problema de asignación de recursos fun-

damental en un entorno AmI, el cual busca satisfacer a los usuarios que habitan el ambiente de acuerdo a los servicios proporcionados. El modelo asume que existen L agentes en el sistema, los cuales tienen la capacidad de proveer una serie de N servicios a un total de M usuarios. Luego, un problema de optimización binaria es formulado para minimizar el tiempo de respuesta al proporcionar los servicios a los usuarios del ambiente inteligente. El problema de optimización considera un conjunto de restricciones que pretenden utilizar eficientemente los agentes disponibles y su energía disponible. Además, para resolver el problema de optimización se propone un algoritmo de asignación de servicios, el cual lleva a cabo un búsqueda heurística y probabilística basada en un criterio según la siguiente razón: los servicios deberían ser asignados con mayor probabilidad a aquellos agentes que procesan más rápido. Finalmente, se implementan tres estrategias para la asignación de servicios: (i) el enfoque proactivo asigna adecuadamente los servicios a todos los usuarios dado un conjunto de agentes; (ii) el enfoque reactivo mientras recibe una nueva solicitud, actualiza el conjunto de solicitudes y continúa la asignación; y (iii) el enfoque *greedy* asigna los servicios únicamente entre los agentes disponibles más rápidos y es usado para fines de comparación.



Introducción

Con el crecimiento tecnológico actual, es factible ofrecer aplicaciones y/o servicios orientados a la asistencia y el mejoramiento de la calidad de vida de las personas a partir del aprovechamiento máximo de las potencialidades de las TIC (Tecnologías de Información y Comunicación). La AmI como tecnología en pleno desarrollo y producto de la convergencia de otras áreas tecnológicas como agentes de software y Wireless Sensor Network (WSN), está enfocada en realizar una variedad de aplicaciones para muchos dominios con el objetivo de crear ambientes inteligentes en donde se trata de mejorar la interacción hombre-máquina y las actividades cotidianas de las personas, y en la cual muchas áreas y grupos de investigación, empresas y universidades están esforzándose por mostrar el futuro de esta fascinante ciencia. Además, el uso de WSN y agentes juegan un papel significativo en la obtención de información del contexto y la toma de decisiones en dichas aplicaciones.

En el área de la Inteligencia Ambiental se busca desarrollar sistemas o aplicaciones que logren la satisfacción de las personas que habitan un determinado espacio, a partir de la proporción adecuada de los servicios brindados por estos. La mayoría de sistemas inteligentes desarrollados hasta el momento no siguen modelos para afrontar el problema de asignación de servicios de una forma eficiente y se basan únicamente en la ejecución de acciones sobre el entorno basados en reglas predefinidas. Además, mientras que estos servicios pretenden exclusivamente proporcionar confort a los usuarios, es posible que se desprecien largos tiempos de respuesta y el alto consumo de recursos computacionales y/o energéticos.

Para proveer diferentes servicios en un ambiente inteligente, es necesario que varios factores trabajen juntos de manera coordinada. Estos factores incluyen la abstracción dinámica del entorno y sus habitantes, usando modelos apropiados

[1]; la asignación de servicios a sus usuarios y la coordinación adecuada de todos los elementos que proveen la inteligencia a dicho entorno. Además, la asignación y provisión de servicios a sus usuarios debe ser realizada en un tiempo mínimo. Idealmente, el aprovisionamiento de servicios debe anticiparse a las solicitudes de los usuarios para que se sientan en un ambiente efectivamente inteligente.

La asignación de servicios es un problema fundamental en ambientes inteligentes, donde los componentes individuales del sistema colaboran para alcanzar un objetivo común [3, 4]. Una asignación de servicios conveniente consiste en completar una tarea asignada, preferiblemente en un tiempo predefinido, usando eficientemente los recursos o componentes del sistema, como agentes, robots, sensores, etc [5]. Por consiguiente, el problema de asignación de servicios debe considerar las prioridades de los servicios, las capacidades y las ubicaciones de los dispositivos que interactúan con las personas, así como los servicios solicitados directamente por los usuarios o aquellos proporcionados por el ambiente inteligente anticipándose a los deseos o comportamientos de los usuarios.

1.1. Descripción del problema

El problema de asignación de servicios en un ambiente inteligente busca alcanzar la satisfacción del usuario a partir de la proporción adecuada de los servicios solicitados por este. Por lo tanto, en esta tesis se propone un modelo matemático que formaliza el problema de asignación de recursos (servicios) en un entorno AmI, minimizando el tiempo total tomado por el entorno para proporcionar un conjunto de servicios solicitados. Para esto, se busca proporcionar una serie de N servicios a un total de M usuarios usando L agentes, con el fin de optimizar el tiempo de respuesta en proveer los servicios. El modelo propuesto considera: (i) las solicitudes de servicios requieren una coordinación adecuada para lograr la satisfacción de los usuarios; (ii) los servicios son ofrecidos por elementos inteligentes, es decir, agentes de software; y (iii) los agentes exhiben diferentes capacidades en términos de velocidad de procesamiento, consumo energético y energía disponible.

En la figura 1.1 se muestra el esquema del sistema de control que representa el problema que se quiere resolver. El set point representa el ambiente inteligente, es decir, es el resultado que se quiere alcanzar producto del control efectuado por los L agentes, donde dicho control está representado por la función de transferencia (F. de T.) $C(s)$. La F. de T. $A(s)$ hace referencia a los actuadores implementados, tales como electrodomésticos, sistema de iluminación, etc., los cuales adaptan el ambiente (representado por el proceso con F. de T. $H(s)$) a través de la ejecución

de acciones. Finalmente, los sensores de la WSN son el elemento de realimentación (con F. de T. $S(s)$) y el perfil de usuario (preferencias) es la entrada del sistema.

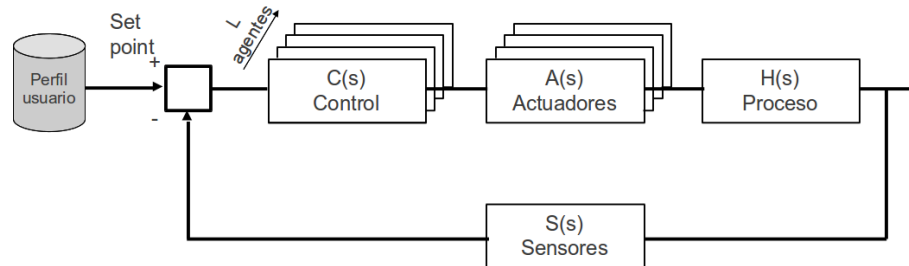


Figura 1.1: Esquema de control del problema de asignación. Fuente: Elaboración propia.

Por lo tanto, el modelo de asignación de servicios consiste en asignar de manera eficiente los agentes del bloque de control de la Figura 1.1 usando el esquema de ambiente inteligente mostrado en la Figura 1.2. El ambiente inteligente considera un Sistema Multi-Agente (SMA) para distribuir los servicios solicitados por los usuarios, y una WSN para recolectar la información del entorno.

1.2. Hipótesis

La provisión óptima de servicios, enfocados en la satisfacción de los usuarios en un entorno de AmI, se puede modelar matemáticamente como un problema fundamental de asignación de recursos donde se pretende proveer una serie de N servicios a un total de M usuarios, minimizando el tiempo total de respuesta que toma proporcionar estos N servicios.

1.3. Objetivos

Objetivo general:

Desarrollar un nuevo modelo de optimización y algoritmos de asignación de servicios, para un entorno inteligente, donde se pretende proporcionar una serie de N servicios a un total de M usuarios, minimizando el tiempo de respuesta.

Objetivos específicos:

1. Formular el problema de optimización para asignar de manera adecuada los servicios a los usuarios, basado en la información del contexto y de cada

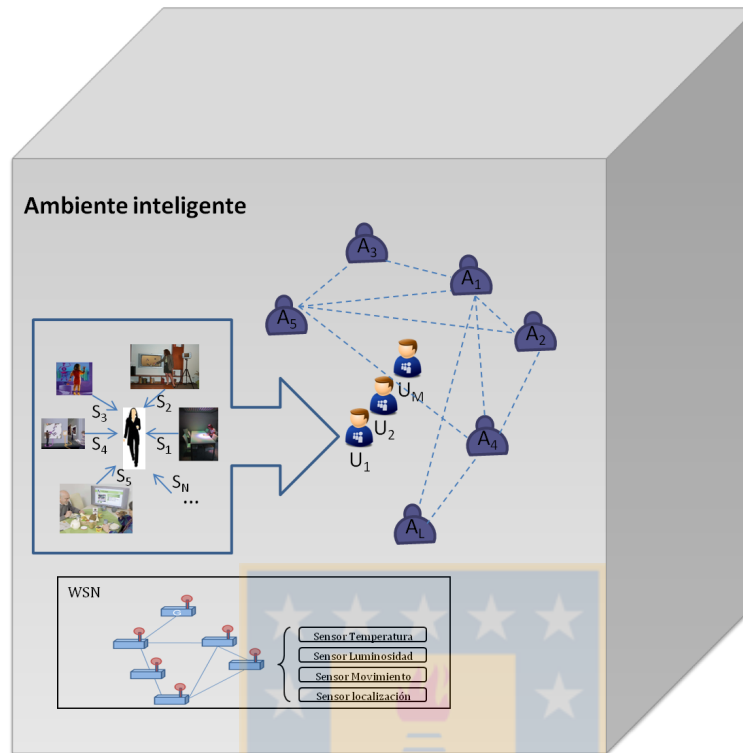


Figura 1.2: Esquema representativo del ambiente inteligente. Fuente: Elaboración propia.

usuario.

2. Formular restricciones relevantes para el problema de optimización que consideren: un número fijo de recursos, tales como agentes de software, sensores inalámbricos y energía disponible en los dispositivos, los perfiles de usuario y el contexto del ambiente inteligente.
3. Investigar heurísticas capaces de resolver el problema de optimización planteado.
4. Explorar y comparar estrategias reactivas y proactivas para la solución del problema de optimización planteado.
5. Desarrollar un método de optimización basado en la información del contexto y de cada usuario que permita una asignación de servicios óptima en problemas de AmI.
6. Evaluar el método de optimización en un ambiente inteligente.

1.4. Trabajos previos de la autora de esta tesis

En esta sección se introducen los trabajos realizados previamente, los cuales se presentarán detalladamente en el Capítulo 2.

1.4.1. Asignación de tareas o recursos

A partir del concepto de computación paralela y distribuida, se implementa en [2] un algoritmo para minimizar el tiempo de procesamiento de una secuencia de video en un sistema de computación distribuido basado en dispositivos portátiles como tablets y smartphones. El algoritmo considera el consumo de energía y la energía restante de los nodos, y realiza una asignación basada en el método de Monte-Carlo de acuerdo a las siguientes razones: las imágenes que componen el video deberían asignarse con mayor probabilidad a aquellos nodos que se comunican y procesan más rápido, consumen menos energía y poseen más depósitos energéticos. Los resultados obtenidos muestran que estas simples ideas permiten al algoritmo de asignación tratar eficientemente con la compensación fundamental entre el tiempo de respuesta y la energía.

1.4.2. Inteligencia Ambiental

En trabajos previos desarrollados por la autora de este tesis [1], se propone un metamodelo genérico de AmI que incluye los modelos más relevantes para el diseño e implementación de sistemas AmI reales, los cuales contienen información relacionada con el estado del entorno y los usuarios, los servicios adaptados a éstos y el conocimiento relacionado a ellos. Además, basado en el metamodelo se diseña e implementa una arquitectura de AmI para un laboratorio inteligente ubicado en un ambiente universitario. Esta arquitectura aprovecha el meta-modelo AmI propuesto para abstraer eficientemente el ambiente y sus usuarios, utiliza una WSN para adquirir las características contextuales del entorno y ejecutar acciones sobre el ambiente, y usa agentes inteligentes para proveer la adaptación requerida para ofrecer servicios personalizados. La evaluación del desempeño de la arquitectura AmI se realizó a través de métricas cualitativas y cuantitativas estándar de un sistema de AmI.

1.5. Contribuciones de tesis

Las principales contribuciones que permitieron la difusión de esta investigación fueron:

- S. E. Restrepo, P. Pinaud, J. E. Pezoa, and S. Sobarzo. “Energy-aware image allocation for distributed video processing on handheld devices”. In Proc. of the 32nd International Performance, Computing, and Communications Conference (IPCCC 2013), San Diego, CA, USA, December 6–8, 2013.
- S. E. Restrepo, J. E. Pezoa, and D. A. Ovalle-Carranza. “An adaptive architecture for ambient intelligence based on meta-modeling, smart agents, and wireless sensor networks”. *IEEE Latin America Transactions*, 12(8):1508–1514, 2014.
- J. E. Pezoa, S. E. Restrepo. “Impact of Correlated Failures on Wireless Sensor Networks Reliability”. Chapter 13 of *The Handbook of Sensor Networking: Advanced Technologies and Applications*. John R. Vacca (Ed). Chapman and Hall/CRC 2015, ISBN: 978-1-4665-6971-3 CRC Press.
- S. E. Restrepo, J. E. Pezoa, M. R. Naeini. “A Model for Optimal Service Allocation in a Smart Environment”. *Enviado a: Pervasive and Mobile Computing* (Elsevier).

1.6. Organización general de tesis

Este documento de tesis está organizado de la siguiente forma. El Capítulo 2 presenta el estado del arte y algunos conceptos claves relacionados con el paradigma de Inteligencia Ambiental y la asignación de recursos, como también algunos trabajos previos en el área. El Capítulo 3 describe el modelo propuesto para la asignación de servicios así como el algoritmo usado para su implementación. El Capítulo 4 presenta los resultados de la implementación y evaluación del modelo propuesto. Finalmente, el Capítulo 5 presenta algunas conclusiones y trabajo futuro.

Inteligencia Ambiental y Asignación de recursos

Este capítulo presenta la revisión de literatura relacionada con los conceptos de AmI y asignación de recursos. Además, se describe el paradigma de Inteligencia Ambiental, así como las herramientas que contribuyen a expandir las posibilidades de éste: Agentes Inteligentes y Redes de Sensores Inalámbricas; seguido de una breve descripción del concepto de asignación de recursos. En esta tesis, los recursos a distribuir corresponden a los servicios ofrecidos dentro de un ambiente inteligente. Finalmente, se presentan algunos trabajos previos desarrollados en el marco de esta tesis.

2.1. Estado del arte

2.1.1. Inteligencia Ambiental

Como se mencionó anteriormente, el objetivo de la AmI es lograr que diferentes dispositivos puedan interactuar con las personas de manera que los beneficie de forma transparente, en donde se facilitan algunas actividades cotidianas o se delegan otras que consumen mucho tiempo y dinero. Lo anterior se puede conseguir a través del uso de WSN debido a que estas permiten el acceso en lugares donde el cableado sería muy costoso o difícil de instalar y tienen la cualidad de proveer portabilidad; y además de la integración de agentes que pueden analizar los datos recibidos, tomar decisiones en un dominio de aplicación específico y realizar diferentes tareas en paralelo [6]. Adicional a la adaptación del entorno según los usuarios y el estado de este, es necesario proporcionar los servicios de una manera óptima de modo que también se ahorren recursos y se distribuyan las tareas de la mejor forma posible entre los agentes que están encargados de la provisión de los servicios.

Por tal motivo, se han encontrado varios trabajos que abarcan en cierta parte lo que se ha mencionado en las secciones anteriores y las cuales serán descritas a continuación.

A la fecha se han desarrollado diversas aplicaciones de AmI en domótica, cuidado de la salud y agricultura [7], [8], [9], [10], [11]. El proyecto AMIGO propone una arquitectura middleware interoperable donde se habilita el concepto de AmI en un entorno de red doméstica. AMIGO conecta electrodomésticos con dispositivos de computación móvil y personal, con el fin de automatizar el hogar. El proyecto propone una arquitectura para cada dispositivo de la red doméstica variando el tipo de aplicación dependiendo de la tarea que éste realiza. Además, implementa el middleware que facilita un entorno en red con todo lo relacionado al manejo de los servicios que facilitan un entorno de AmI [7].

El proyecto MyHeart propone diseñar y desarrollar dispositivos electrónicos y servicios, que ayuden al usuario a controlar su propia salud desde cualquier lugar, creando de esta forma entornos inteligentes que mejoren su calidad de vida [9]. Este sistema se ha centrado en la gestión de problemas cardíacos a partir del monitoreo de señales vitales de pacientes, a través de sensores implantados en su ropa. En caso de alguna anomalía, los sensores envían advertencias a los usuarios y al personal médico.

Para proveer interacción entre personas y sistemas, Huynh *et al.* diseñaron un sistema de iluminación LED controlable creado bajo el paradigma de Inteligencia Ambiental a partir de la implementación de una WSN inteligente distribuida, el cual optimiza y controla el sistema de iluminación de forma eficiente y adaptado a las preferencias del usuario y a las condiciones de iluminación in y outdoor [12].

Recientemente, se han incorporando agentes inteligentes a entornos de AmI para ejecutar funciones autónomas. Como iniciativa de un sistema de AmI, Hernández *et al.* presentaron un Sistema Multi-Agente (SMA) híbrido para la detección y control de incendios. La idea es que el sistema genere predicciones del tipo de incendio que puede generarse en una zona específica mediante un razonamiento basado en reglas, o bien que genere alarmas (de forma reactiva) en situaciones consideradas de riesgo basadas en la medición de variables físicas como temperatura y presencia de humo [13].

Sebbak *et al.* integraron agentes inteligentes a la plataforma Open Service Gateway initiative (OSGi) y desarrollaron un framework de servicios ubicuos sensibles al contexto [14]. Este framework está compuesto por varias capas (Multi-Agente, OSGi y física), permite la interoperabilidad entre diferentes aplicaciones,

es dinámicamente configurable y extensible debido a la heterogeneidad y a la movilidad tanto de los dispositivos inteligentes como de los usuarios.

Por otra parte, con el objetivo de optimizar ciertas actividades o servicios dentro de un ambiente inteligente, algunos investigadores han aplicado varios métodos o algoritmos de optimización para la ejecución o proporción de estos [15]. En [16] desarrollan un método de toma de decisiones para un escenario de entrenamiento de *running* de múltiples etapas basado en el método de optimización de programación dinámica y formulado como un proceso de decisión de Markov, el cual permite a atletas realizar programas de entrenamiento heterogéneos con varios niveles de intensidad del ejercicio. Para lograr esto, tanto el ambiente como los atletas son monitoreados por medio de una WSN desplegada sobre un circuito de carreras a campo abierto, y por elementos móviles integrados en los usuarios que sensan su ritmo cardíaco. A partir de esto, se busca seleccionar una trayectoria de entrenamiento para un determinado usuario que optimice el ritmo cardíaco de acuerdo a un programa de entrenamiento predefinido.

La Tabla 2.1 compara algunos de los trabajos citados anteriormente acá y en el capítulo 2, usando criterios asociados a las características más relevantes en el área de AmI. El factor común entre todos estos trabajos es que, a nuestro saber, muy pocos garantizan que la solución al problema de asignar ciertos servicios a un determinado número de usuarios sea óptima. Además, no existe un formalismo conceptual que integre todos los modelos y/o información requerida dentro del paradigma de AmI.

	[17]	[18]	[16]	[7]	[9]	[12]	[13]	[14]
Contexto	?	+	+	+	+	+	+	+
Perfil de usuario	-	-	+	+	+	+	-	-
Adaptación al contexto	-	-	+	+	+	+	+	+
Adaptación al usuario	-	-	+	+	+	+	-	-
WSN	+	+	+	-	+	+	-	-
Agentes inteligentes	-	-	-	?	-	-	+	+
Asig. servicios óptima	+	-	+	?	-	-	-	-

Tabla 2.1: Comparación de trabajos relacionados con AmI. Fuente: Elaboración propia.

Los símbolos (+), (-) y (?) indican, respectivamente, que el trabajo incluye, no incluye o no se sabe si incluye el criterio mencionado.

2.1.2. Asignación de recursos

Debido a la variedad y cantidad de servicios ofrecidos y usuarios presentes en un ambiente inteligente, es de vital importancia coordinar y distribuir adecuadamente los servicios a los correspondientes usuarios, análogo a como ocurre en el caso de los sistemas distribuidos con la asignación de recursos o tareas en los dispositivos que los componen. Una de las áreas donde es estudiada ampliamente la asignación de recursos son las WSNs [19], donde un claro ejemplo es el trabajo realizado en [17]. Pezoa propone un algoritmo para la asignación de recursos en una WSN, donde se pretende maximizar la probabilidad de completar una misión (o tarea) en la WSN, de acuerdo a un puntaje o marca que pondera los recursos, la disponibilidad y la probabilidad de que los nodos sensores sean afectados por fallas correlacionadas geográficamente. De esta manera, el algoritmo desarrollado favorece la asignación de tareas a los nodos sensores más rápidos con mayor cantidad de energía disponible y penaliza la asignación a aquellos sensores que probablemente fallen de una manera correlacionada geográficamente.

Por otro lado, en [18] presentan una posible solución al problema de asignación de recursos en aplicaciones distribuidas presentes en el internet de las cosas (IoT), cuya visión es pretender conectar dispositivos con diferentes capacidades tales como sensores, actuadores, objetos inteligentes (como *smartphones*) y servidores, dentro de la misma red heterogénea. Para esto, proponen un protocolo de consenso donde los objetos de la red cooperan entre sí con el fin de lograr la misión de la aplicación y distribuyen las tareas de tal forma que los recursos sean compartidos adecuadamente.

Al mismo tiempo, en [3] proponen un sistema multiagente para la gestión de recursos en sistemas domóticos, donde la asignación de recursos depende de los parámetros de comportamiento de los dispositivos individuales y de ciertas prioridades. Por otra parte, en [4] diseñaron agentes con motivación intrínseca para tratar con la incertidumbre en el ambiente y la diversidad de requisitos. Este tipo de agentes pueden mostrar gran adaptabilidad para ayudar a los usuarios a rastrear, predecir y utilizar el conjunto de recursos compartidos, y proporcionar información más confiable a través de la colaboración entre múltiples agentes. Además, investigadores de ATR Intelligent Robotics and Communication Laboratories han desarrollado un sistema de robots en red con el fin de permitir la interacción humano-robot en espacios públicos como un centro comercial [20]. El sistema considera una asignación de servicios según las prioridades de los servicios y las características de los robots, así como algunos tipos de servicios.

Recientemente, se han aplicado metaheurísticas en un gran número de proble-

mas de optimización con el fin de estudiar problemas de gran complejidad, como la asignación de recursos, de una manera sencilla y de obtener soluciones suficientemente buenas en tiempos razonables. En [21] se presentan varios modelos y heurísticas para sistemas de computación distribuida y paralela que generan asignaciones de recursos robustas, considerando la incertidumbres que pueden dar lugar a variaciones en los tiempos de ejecución de dichos sistemas. Además, en [22] evalúan el desempeño de varias técnicas de búsqueda metaheurística en problemas de asignación de recursos y planificación (*scheduling*) en proyectos de desarrollo de software. El análisis comparativo de las metaheurísticas sugiere un mejor desempeño de los algoritmos genéticos sobre el recocido simulado (*Simulated Annealing*) y la búsqueda tabú.

2.2. Conceptos y fundamentos: Inteligencia Ambiental

La Inteligencia Ambiental consiste en la creación de espacios donde los usuarios interactúan de forma natural y sin esfuerzo con los diferentes sistemas, gracias a que las tecnologías de computación y comunicación se vuelven invisibles para el usuario, al estar siempre presentes e integradas en los objetos cotidianos del mismo [23]. Esta visión ha despertado un creciente interés por utilizar las tecnologías de la computación en la construcción de ambientes inteligentes que soporten las actividades de la vida diaria de forma más eficiente. De esta forma, dentro de estos espacios se abarcan cualidades, como el reconocimiento y la acomodación de los diversos componentes que los constituyen, la personalización y la adaptatividad del sistema, el entendimiento del estado en determinada situación, la colaboración y la cooperación entre componentes del sistema y las características autónomas como auto-administración y auto-reparación [24].

Entre las características principales de un sistema de AmI se encuentran las siguientes:

- El entorno debe ser sensible al contexto, es decir, tener capacidad para adaptarse a la información proveniente del mismo. El contexto se refiere a cualquier información que puede ser utilizada para caracterizar la situación de cualquier persona, lugar u objeto, la cual se considera relevante para los servicios proporcionados por el ambiente [25].
- El acceso a la información, comunicación y servicios, debe realizarse de forma ubicua, inalámbrica y transparente para el usuario.

- La interacción hombre-máquina (usuario-sistema) debe realizarse de forma natural y no intrusiva.

Según E. Aarts [26], las cinco características principales de la AmI son:

- Embebida: muchos dispositivos conectados en red son integrados en el ambiente.
- Sensible al contexto: estos dispositivos pueden reconocer a las personas y la situación de su contexto en un momento determinado.
- Personalizada: estos dispositivos pueden adaptarse a las necesidades de las personas, realizando ajustes en el ambiente dependiendo de cada persona en particular.
- Adaptativa: el comportamiento de estos dispositivos pueden cambiar en respuesta a las personas y a su contexto.
- Anticipativa: estos dispositivos pueden anticipar los deseos de los usuarios y realizar ajustes apropiados sobre el ambiente.

El desarrollo de la AmI se debe a algunos avances tecnológicos que se han logrado en varias áreas tecnológicas, como es el caso de la informática, donde actualmente se cuenta con nuevas formas de interacción hombre-máquina que se enfocan en la sistematización de los sentidos o características del ser humano, el incremento de las capacidades de procesamiento y el surgimiento de sistemas de procesamiento y comunicación más simples y amigables para el hombre. En el área de la electrónica se han desarrollado gran cantidad de sensores y algunos dispositivos electrónicos se han miniaturizado de tal manera que son fáciles de introducir en elementos cotidianos; por otro lado, el desarrollo de redes de comunicación, especialmente las móviles e inalámbricas, permiten interconectar todos los sistemas e imprimirles portabilidad.

Todo este entramado tecnológico se puede lograr a través de tres tecnologías claves: computación ubicua, comunicación ubicua e interfaces inteligentes [27] y a partir de la integración de sistemas adaptivos, redes de sensores inalámbricas, agentes inteligentes, entre otros [26, 23].

2.2.1. Agentes de Software

Un agente es una entidad de software que puede actuar por sí mismo con el fin de alcanzar unos objetivos que se ha fijado inicialmente y además está caracterizado por una o varias cualidades tales como capacidad de razonamiento (inteligencia),

percepción de su entorno y actuación con base a ciertas circunstancias (reactividad) [28]. Además, tiene la capacidad de actuar en forma proactiva, es decir, actúa sin necesidad de darle una orden y la habilidad de desplazarse de un lugar a otro (movilidad). Otras características con que puede contar son autonomía, modularidad, adaptación, habilidad social, continuidad temporal, orientación hacia el objetivo final, benevolencia, colaboración [29].

Según Wooldridge y Jennings (1995), un agente es un sistema de computador que está ubicado en algún ambiente, y está capacitado de acciones autónomas en este ambiente para cumplir sus objetivos planteados. La primera definición de agente más general dada por G. W. Lecky - Thompson es: “Un agente es una pieza de software que ejecuta una tarea dada usando información obtenida de su entorno para actuar de forma apropiada para completar la tarea exitosamente. El software debe ser capaz de adaptarse a sí mismo en base a los cambios que ocurren en su entorno, de modo que ante un cambio en algún evento o situación pueda aún obtener el resultado deseado” [30].

Los agentes pueden clasificarse de muchas maneras, dependiendo de sus características, la tecnología que los soporta, la función que desarrollan o una combinación de estas. Según su movilidad pueden ser agentes estacionarios o estáticos, o agentes móviles; por su característica fundamental se dividen en agentes reactivos, agentes inteligentes, agentes autónomos y agentes adaptativos. Finalmente, según una combinación de sus características pueden ser agentes de interfaz, agentes colaborativos, agentes de información o agentes de aprendizaje.

Un sistema que consiste de un grupo de agentes que pueden interactuar potencialmente uno con otro para lograr un objetivo común es llamado Sistema Multi-Agente (SMA) y debido a que éste está constituido por varias entidades independientes (agentes) necesita definir muy bien cómo están organizados, cómo se comunican, coordinan y cooperan, y finalmente cómo se controlan estos agentes para lograr un objetivo común.

Para el análisis y diseño de SMA se han creado varias metodologías, algunas de las más utilizadas son GAIA, MAS-CommonKADS e INGENIAS, las cuales proporcionan modelos para capturar los principales rasgos del sistema y su entorno.

Por otro lado, los agentes pueden usarse para realizar tareas en el ambiente basados en un razonamiento o mecanismo de inteligencia, gracias a que algunas de sus características les permite proveer servicios y adaptar adecuadamente el entorno para el beneficio del usuario y la toma de decisiones en un dominio de

aplicación específico. Estas características son: (i) proactividad, donde el agente actúa sin necesidad de darle una orden; (ii) adaptatividad, este es capaz de ajustarse a sí mismo a los hábitos, métodos de trabajo y preferencias de su usuario, así como a su contexto; (iii) autonomía o semiautonomía, agente opera sin o con poca intervención directa de los humanos; (iv) reactividad, agentes perciben su entorno y responden en un tiempo razonable a los cambios que ocurren en él; (v) inteligencia o racionalidad, el agente siempre actuará para lograr sus metas y no actuará de una forma que evite la consecución de las mismas. Además, los agentes pueden usarse en AmI como abstracciones para modelar e implementar tanto funcionalidades y dispositivos de un sistema AmI (manejo de sensibilidad al contexto, dinámica de los sistemas), encapsular técnicas de inteligencia artificial y coordinar los diferentes elementos que componen la aplicación [31].

2.2.2. Redes de Sensores Inalámbricas (WSN)

Las Redes de Sensores Inalámbricas o WSN, en inglés, son redes inalámbricas compuestas por varios dispositivos autónomos distribuidos llamados nodos sensores, las cuales son capaces de crear sistemas de monitoreo y adquisición de datos del entorno donde son configuradas. Las WSN son un área emergente de los sistemas embebidos que tienen el potencial para revolucionar nuestras vidas en la casa y en el trabajo, con aplicaciones de gran escala, incluyendo monitoreo y conservación ambiental (por ejemplo en invernaderos), control industrial, administración de negocios, monitoreo estructural y sísmico, transporte, salud y domótica [32]. En una WSN, cada nodo sensor posee capacidades de procesamiento, almacenamiento y sensado, y una comunicación radio entre nodos. Además, cada nodo está equipado con uno o más dispositivos sensores, tales como sensores de luz visible o infrarroja, campos magnéticos, resistencia eléctrica, aceleración o vibración, pH, humedad o temperatura, micrófonos acústicos y/o video o cámaras fotográficas [33].

Las WSN poseen varios retos que deben ser tratados antes de desplegarlas e implementarlas a gran escala [34]. Entre ellos están: la conservación de energía, las comunicaciones de baja calidad, la operación en ambientes hostiles, los recursos de cómputo limitados, el procesamiento de información reducido, la escalabilidad, y la escasez de aplicaciones de fácil comercialización. Aunque los nodos sensores poseen varias limitaciones y los desarrolladores se enfrentan a varios desafíos de diseño, existen varias ventajas para la instrumentación de un área con una red de sensores [35]. Entre ellas se resaltan un alto nivel de tolerancia a fallas en la WSN debido al despliegue denso de un gran número de nodos; una mejora en la calidad de sensado, combinando lecturas de múltiples sensores independientes y la prevención de cualquier factor ambiental en el entorno que interfiera con la observación del fenómeno deseado debido a que los nodos son desplegados muy

cerca al evento sentido.

Debido a la funcionalidad de las WSN como medio de adquisición de información del contexto y gracias a ciertos atributos que poseen, estas pueden beneficiar al ambiente donde son implementadas de dos formas, recopilando información del ambiente y actuando sobre éste. Algunas de las características de las WSN que permiten modelar un ambiente inteligente son: (i) integración de sensores en el ambiente y en el vestuario o accesorios de los usuarios que adquieran información acerca del estado de estos; (ii) el uso de actuadores controlados por la WSN que harán posible la realización de acciones en el ambiente; (iii) transmitir los datos de los sensores al elemento del sistema donde está el mecanismo de inteligencia; y (iv) integración e interoperabilidad de la WSN en el ambiente y a nivel de red personal, local y global.

2.3. Conceptos y fundamentos: Asignación de recursos

La asignación de recursos es un tema de investigación que ha sido abordado desde muchas áreas del conocimiento, en especial, en aquellas donde se tienen diferentes entes u objetos heterogéneos que se relacionan entre sí y comparten un conjunto de recursos, servicios o tareas, donde es necesario distribuir estos elementos en los entes individuales con el fin de alcanzar un objetivo final.

La asignación de recursos puede realizarse según varios métodos:

- **Programación lineal:** es un método de optimización para la solución de problemas que implican la maximización o minimización de una función lineal de un conjunto de variables no negativas (función objetivo), sujeta a restricciones (conjunto de igualdades y/o desigualdades también lineales), que relacionan a las variables de decisión [36]. Surgió a partir del desarrollo de métodos para la asignación de recursos óptima y es de gran utilidad debido a que permite tomar decisiones óptimas en situaciones complejas tales como la asignación de recursos hospitalarios [37], la distribución equitativa de tareas en un sistema distribuido [38], entre otros. El problema de programación lineal general puede ser establecido en la siguiente forma estándar [39]:

$$\text{mín } f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

sujeto a las restricciones

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \\
x_1 &\geq 0 \\
x_2 &\geq 0 \\
&\vdots \\
x_n &\geq 0
\end{aligned}$$

donde c_j, b_j y a_{ij} ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$) son constantes conocidas y x_j son las variables de decisión.

- **Heurísticas:** son una estrategia de búsqueda informada, la cual utiliza el conocimiento específico del problema además de la definición del problema para encontrar soluciones de una manera eficiente [28]. Como por ejemplo, el conocimiento obtenido de la experiencia.

A la aproximación general se le considera “búsqueda primero el mejor” (caso particular del algoritmo de búsqueda basada en grafos), en el cual se selecciona un nodo para la búsqueda basada en una función de evaluación. Tradicionalmente, se selecciona el nodo con la evaluación más baja para realizar la búsqueda, debido a que la evaluación mide la distancia al objetivo. La función de evaluación puede definirse de formas diferentes, donde la función heurística $h(n)$ es clave en este tipo de algoritmos debido a que es la forma más común de transmitir el conocimiento adicional del problema al algoritmo de búsqueda.

La asignación de recursos basada en heurísticas es usada en muchas áreas, algunos ejemplos son la asignación de recursos humanos y organizacionales en ambientes universitarios [40], asignación de recursos (tareas y mensajes) en sistemas de control automatizado distribuido [41], entre otros.

- **MAS:** La asignación de recursos basada en MAS (Multi-Agent System) o SMA, en español, consiste en el desarrollo de soluciones al problema de distribuir un número de tareas y/o recursos entre múltiples agentes [42]. Este tipo de asignación ha sido ampliamente desarrollado en los últimos años y en diferentes áreas, incluyendo la producción de una empresa, el enrutamiento de redes, el transporte público, el comercio electrónico, la construcción industrial, entre otras [43]. En este caso, se proporciona un algoritmo de búsqueda que asegura que los recursos sean distribuidos de acuerdo a las

preferencias específicas de los agentes e intenta maximizar el bienestar social de éstos. La preferencia del agente representa el grado de satisfacción por determinada asignación, la cual puede tomar un valor explícito o una relación que revela la alternativa más satisfactoria. Además, su evaluación es expresada generalmente como una función de utilidad, la cual el agente usa para comunicar su grado de satisfacción.

2.4. Modelo de Inteligencia Ambiental basado en la integración de WSN y Agentes inteligentes

Un sistema AmI que pretenda ser aplicado en situaciones reales debe ser diseñado teniendo en cuenta un cierto contexto y una serie de capacidades que le permitan adaptarse a las condiciones dinámicas del ambiente y, simultáneamente, a los requerimientos de los usuarios [26, 23]. Para lograr esto, es necesario que un sistema AmI sea capaz de abstraer, dinámicamente, el entorno y sus usuarios mediante algún modelo apropiado. En [1] se presenta un meta-modelo genérico de AmI que incluye los modelos más relevantes para el diseño e implementación de sistemas AmI reales.

El meta-modelo AmI propuesto ha sido desarrollado usando como requerimiento las principales ideas del paradigma de AmI. A saber: dado un contexto y un conjunto de usuarios en un sistema de AmI, existe un conjunto de dispositivos de computación embebidos, dispositivos de sensado, y una configuración de perfil de usuario que permiten al sistema adaptarse a las condiciones dinámicas del ambiente y a los requerimientos de los usuarios [44], y además debe tener la capacidad de proporcionar servicios de AmI [11].

Por lo tanto, los modelos definidos en el meta-modelo son:

- **Modelo de Contexto:** caracteriza la situación actual del usuario por medio de los siguientes submodelos: (i) de contexto tecnológico, que identifica los distintos dispositivos tecnológicos (dispositivos móviles, electrodomésticos, WSNs, red SMA, etc.) presentes en el entorno y que permiten la captura, transmisión y análisis de la información de éste; (ii) de contexto espacio-temporal, que localiza las entidades (personas u objetos) que habitan el ambiente y registra los tiempos en que ocurren sus eventos o acciones; (iii) de contexto social, que adquiere información contextual sobre los usuarios y su entorno social; y (iv) de contexto del entorno, que sensa variables físicas del entorno (temperatura, humedad relativa, luminosidad, etc.), registra los

objetos involucrados en la interacción usuario-entorno y describe el tipo de entorno.

- **Modelo del Usuario:** identifica y abstrae las necesidades del usuario, lo que permite que el meta-modelo se adapte a sus necesidades y proporcione la interactividad requerida por un sistema de AmI. Este modelo está definido por las preferencias de usuario (afinidad o inclinación hacia algo), sus gustos (definidos a largo plazo, siendo un deleite que se experimenta por algún motivo o se recibe de cualquier cosa), sus intereses (que están influenciados por la moda, el cambio de estaciones y clima, edad y otros aspectos a corto plazo) y su información personal.
- **Modelo de Dominio de Aplicación:** almacena la información necesaria para definir adecuadamente los componentes del sistema de AmI, y lograr su adaptación al área o dominio de aplicación. Esto permite focalizar de manera precisa el tipo de servicios que se requieren y describir información relacionada con el contexto y el usuario. Además, incluye información sobre los actuadores que realizan la adaptación del ambiente y sobre otros aspectos del dominio de aplicación que no son abarcados por los Modelos de Contexto y Usuario.
- **Modelo de Adaptación:** representa la forma cómo interactúan los usuarios con el entorno de AmI y se enfoca en proporcionar el mayor grado de satisfacción a los usuarios. En este caso, la adaptación del entorno es controlada por una red de agentes inteligentes que busca alcanzar un objetivo determinado por los eventos que están ocurriendo en el sistema de AmI. Este modelo define las formas de adaptación que se pueden obtener con el objetivo de ofrecer servicios considerando las características del usuario y/o algún tipo de contexto.

Además de los cuatro modelos anteriores, el meta-modelo AmI propuesto considera un conjunto de servicios de inteligencia ambiental y de dispositivos como sensores y actuadores. Los primeros corresponden a un conjunto de N ambientes y capacidades proporcionadas en tiempo real por el sistema AmI a sus usuarios. Por su parte, el conjunto de dispositivos sensores y actuadores del ambiente son considerados acá como una WSN integrada en el ambiente, un conjunto de dispositivos integrados en los usuarios, y un número determinado de agentes inteligentes.

Por otra parte, dicho modelo es aprovechado para diseñar e implementar una arquitectura de AmI para un laboratorio inteligente (SmartLab). La arquitectura, representada en la Figura 2.1, está dividida en una capa física, una de agentes, una de dominio de aplicación y una de servicios.

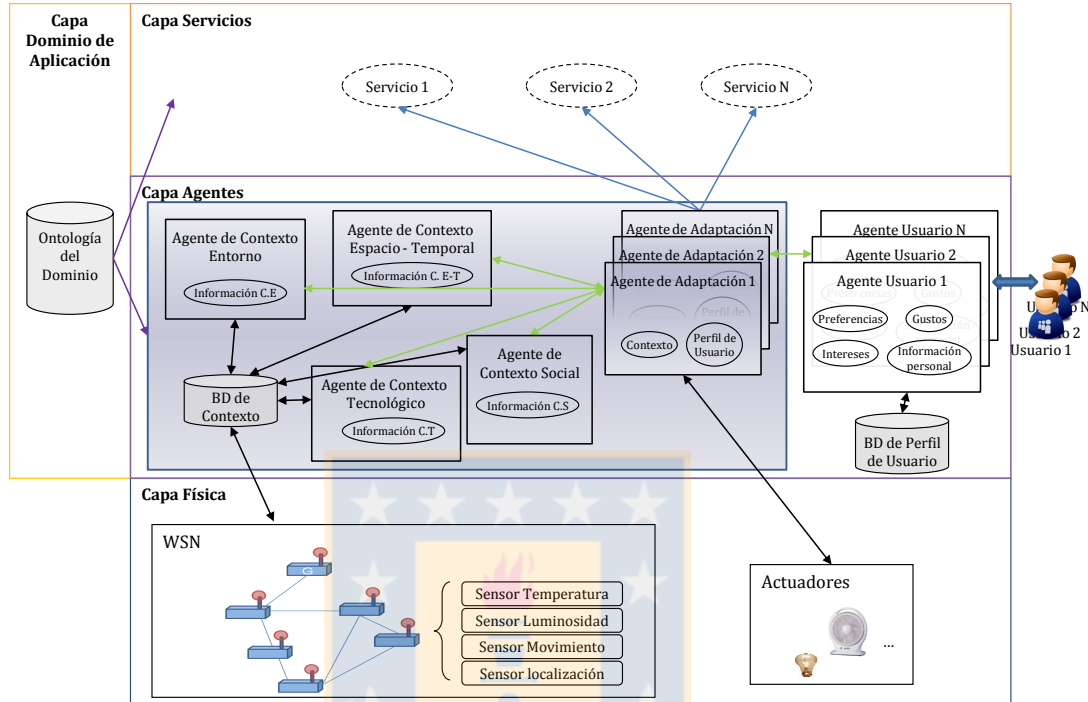


Figura 2.1: Arquitectura de la aplicación del laboratorio inteligente [1]. Fuente: Elaboración propia.

La *capa física* está compuesta por una WSN que sensa ciertas variables físicas ambientales y realiza el seguimiento de los usuarios. Esta capa incluye además algunos actuadores que modifican el entorno. La capa física interactúa con la capa de agentes de dos formas: (i) alimentando la base de datos de contexto; y (ii) por medio de los agentes inteligentes que adaptan el entorno de acuerdo a las aplicaciones activadas por los servicios.

La *capa de agentes* implementa el Modelo de Contexto, el Modelo del Usuario y el Modelo de Adaptación usando varios agentes inteligentes. La capa de agentes especifica un agente inteligente para cada submodelo del Modelo del Contexto, N agentes para el Modelo de Adaptación y otros N agentes para el Modelo del Usuario. Estos últimos agentes inteligentes recuperan y almacenan información en la base de datos de perfil de usuario.

La *capa de dominio de aplicación* formaliza y transfiere el conocimiento adquirido por el sistema en un lenguaje que puede ser interpretado por los agentes inteligentes. Esta capa es clave en la arquitectura AmI propuesta y ha sido implementada como una ontología, la cual contiene las relaciones entre conceptos del dominio que son intercambiados por los agentes.

Finalmente, la *capa de servicios* implementa en software los servicios de AmI proporcionados a los usuarios. La especificación de los servicios de AmI define un mecanismo de razonamiento basado en reglas en donde cada servicio es ofrecido como consecuencia del cumplimiento de condiciones previas. Algunos de los servicios implementados en el sistema inteligente son: proporción de información, adaptación del entorno de acuerdo a la temperatura y luminosidad y activación de sistemas (karaoke y audio).

La arquitectura propuesta ha sido programada utilizando las herramientas SQLite, Jade, Protégé, y CLIPs, y ha sido simulada en el ambiente de desarrollo Solarium de Oracle. Dicha arquitectura además de tener la fortaleza de integrar WSNs con Agentes Inteligentes, posee una eficiente clasificación del contexto que permite deducir más fácilmente la situación actual del entorno y su consecuente adaptación. La evaluación de esta implementación, en términos de varias métricas cualitativas y cuantitativas, ha sido satisfactoria y ayuda a demostrar la capacidad del meta-modelo para proporcionar sistemas AmI adaptativos.

2.5. Asignación de Imágenes para el procesamiento distribuido de video considerando el consumo energético

La asignación de aplicaciones paralelas en dispositivos portátiles como smartphones y tablets, es un tema de mucho interés debido al avance tecnológico actualmente [45, 46, 47]. En [2], se propone resolver el problema de asignación de recursos, en este caso imágenes, minimizando el tiempo de procesamiento de una secuencia de video, compuesta por N imágenes, a través de la realización de ciertos cálculos en paralelo, sobre un sistema distribuido compuesto por M nodos.

Dicho sistema distribuido está compuesto por un nodo de servidor y un conjunto de M nodos de computación portables. El nodo servidor no procesa ninguna imagen, pero realiza las siguientes tareas: i) almacena la secuencia de video; ii)

encuentra la mejor asignación de imágenes entre los nodos portables; y iii) coordina el intercambio de imágenes hacia y desde los nodos de computación. Estos nodos poseen diferentes capacidades dependiendo del procesador, por lo tanto, el tiempo que toma un nodo para procesar una imagen dependerá del tiempo de procesamiento del nodo.

Se define T_j^{proc} como el tiempo tomado por el nodo j para procesar una imagen, T_j^{tx} el tiempo para transferir una imagen del servidor al nodo j y T_j^{rx} el tiempo para transferir una imagen del nodo j al servidor. Por lo tanto, el tiempo de respuesta de una imagen cuando es procesada por el nodo j será:

$$T_j = T_j^{rx} + T_j^{proc} + T_j^{tx}, \quad (2.1)$$

donde $j \in \{1, \dots, M\}$ y $T_j \in \mathbb{R}^+$. Puesto que la energía en dispositivos portables es un recurso escaso, la energía disponible de un nodo no puede estar por debajo del umbral E_{\min} como consecuencia del procesamiento del video. Además, si la asignación de imágenes es especificada por $\mathbf{I} = ((I_{i,j}))_{N \times M}$, cuyo elemento (i, j) toma el valor de “1” si la imagen i es procesada en el nodo j , y en el caso contrario, el valor de dicho elemento es “0”, el tiempo de procesamiento de la secuencia completa del video por el sistema distribuido puede ser escrito como:

$$T(\mathbf{I}) = \max_{j \in \{1, \dots, M\}} \sum_{i=1}^N T_j I_{i,j}. \quad (2.2)$$

con $i = 1, \dots, N$. Por consiguiente, el tiempo mínimo de procesamiento de una secuencia completa de video, $T(\mathbf{I}^*)$, es la solución al siguiente problema de minimización con restricciones:

$$T(\mathbf{I}^*) = \min_{\mathbf{I} \in \{0,1\}^{N \times M}} \max_{j \in \{1, \dots, M\}} \sum_{i=1}^N T_j I_{i,j}, \quad (2.3)$$

sujeto a:

$$\sum_{j=1}^M I_{i,j} = 1, \quad i = 1, \dots, N, \quad (2.4)$$

$$E_j - \sum_{i=1}^N e_j I_{i,j} \geq E_{\min}, \quad j = 1, \dots, M. \quad (2.5)$$

donde E_j corresponde a la energía disponible en el nodo j ($E_j \in \mathbb{R}^+$) y e_j la energía total utilizada por el nodo j para procesar y comunicar una sola imagen

$(e_j \in \mathbb{R}^+)$.

Con el fin de resolver el problema de asignación de imágenes descrito en las ecuaciones (2.3) - (2.5), se propone un algoritmo de búsqueda de Monte-Carlo. Dicho algoritmo aprovecha la restricción de procesar una imagen únicamente en un nodo y realiza la búsqueda imagen por imagen. Por lo tanto, para cada imagen, el algoritmo asigna aleatoriamente dicha imagen a uno de los nodos que satisface la restricción de mínima energía. La búsqueda además usa un criterio de aptitud para la asignación, el cual induce una función de probabilidad o pmf (probability mass function) con el fin de seleccionar aleatoriamente un nodo. Dicho criterio sigue la base lógica de que una imagen debería ser asignada a aquellos nodos que se comunican y procesan más rápido, consumen menos energía y poseen mayor energía disponible. La Tabla 2.2 muestra varios criterios considerados para tres tipos de imágenes con diferentes tamaños (640×480 , 800×600 y 1920×1080), donde los tiempos de procesamiento fueron caracterizados usando una aplicación de Android y los siguientes valores: $N = 100$, $M = 10$ y $E_{min} = 20$. En este caso, s_j es la velocidad de transferencia del nodo j y M_i es el número de nodos disponibles para la asignación de la imagen i .

Tabla 2.2: Tiempo de procesamiento de video para diferentes criterios de asignación. Fuente: Recuperado de [2].

Criterio	pmf	T^* [s]		
		640×480	800×600	1920×1080
Referencia		30.825	48.514	251.840
Velocidad de procesamiento	$p(j) = \frac{1/T_j}{\sum_{\ell=1}^{M_i} 1/T_\ell}$	45.210	75.080	373.743
Energía disponible	$p(j) = \frac{E_j}{\sum_{\ell=1}^{M_i} E_\ell}$	47.265	76.669	375.865
Vel. Procesamiento & Energía disponible	$p(j) = \frac{E_j/T_j}{\sum_{\ell=1}^{M_i} E_\ell/T_\ell}$	47.265	76.669	392.821
Energía consumida	$p(j) = \frac{1/e_j}{\sum_{\ell=1}^{M_i} 1/e_\ell}$	51.375	84.900	428.532
Vel. Procesamiento & energía consumida	$p(j) = \frac{1/T_j e_j}{\sum_{\ell=1}^{M_i} 1/T_\ell e_\ell}$	51.375	86.253	427.135
Vel. Comunicación	$p(j) = \frac{s_j}{\sum_{\ell=1}^{M_i} s_\ell}$	55.657	86.607	471.153
Vel. Comunicación & energía consumida	$p(j) = \frac{s_j/e_j}{\sum_{\ell=1}^{M_i} s_\ell/e_\ell}$	57.222	93.023	493.875

Los resultados del algoritmo de asignación de imágenes son registrados en la Tabla 2.2, y las Figs. 2.2(b) and (c) muestran el número de imágenes procesadas por cada nodo y el consumo energético para la secuencia de video 640×480 . Se

usa como criterio de referencia, aquel que no considera restricciones energéticas y donde todos los nodos tienen la misma probabilidad para procesar una determinada imagen. A partir de los resultados se muestra que: (i) únicamente para el caso de referencia, todos los nodos procesan imágenes; (ii) en todas las asignaciones (excepto el caso de referencia), los nodos 5 y 9 nunca procesan imágenes debido a que no cumplen con el requisito energético; y (iii) en todas las asignaciones, los nodos 7, 8 y 10 procesan el mayor número de imágenes debido a que poseen los más bajos tiempos de respuesta y valores de consumo energético.

Como consecuencia, puede observarse que los resultados concuerdan con la idea de que las imágenes deberían ser asignadas con mayor probabilidad a aquellos nodos que comunican y procesan con mayor rapidez y poseen mayor energía disponible; donde para todos los casos, el mejor criterio de aptitud es la velocidad de procesamiento.



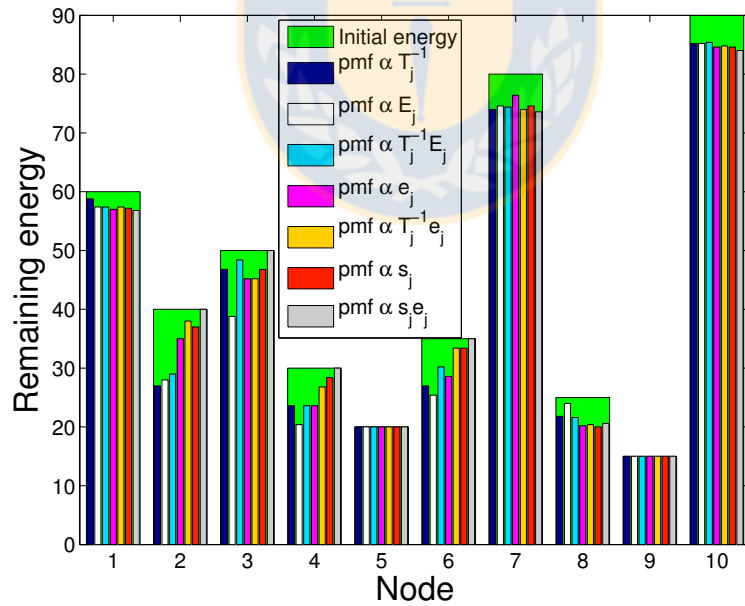
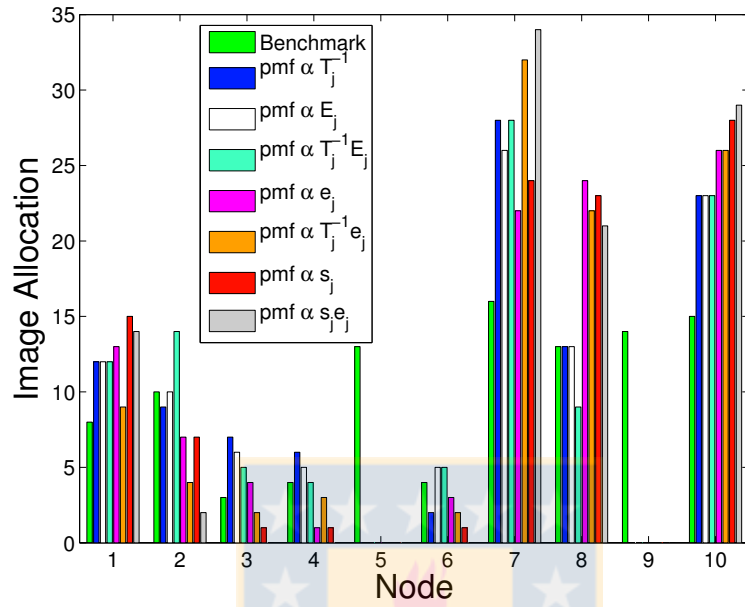


Figura 2.2: Para la secuencia de video 640×480 [2]: (a) Asignación de imágenes por nodo. (c) Energía disponible por nodo. Fuente: Recuperado de [2].

Modelo de Asignación de Servicios

En este capítulo se describe el modelo propuesto, para el cual se tomaron como herramienta las potencialidades del algoritmo de asignación de imágenes descrito en la Sección 2.5 para el modelamiento de la asignación de servicios, y del modelo de Inteligencia Ambiental descrito en la Sección 2.4 para la conceptualización del problema en un entorno AmI.

3.1. Definición del problema

El problema de asignación de servicios en un ambiente inteligente busca alcanzar la satisfacción del usuario a partir de la proporción adecuada de los servicios solicitados por este. Para asignar de manera eficiente los servicios, se busca minimizar el tiempo total tomado por el entorno para proporcionar un conjunto de servicios solicitados. Por lo tanto, se propone un modelo matemático que formaliza el problema de asignación de recursos en un entorno AmI, el cual considera: (i) las solicitudes de servicios requieren una coordinación adecuada para lograr la satisfacción de los usuarios; (ii) los servicios son ofrecidos por elementos inteligentes, es decir, agentes de software; y (iii) los agentes exhiben diferentes capacidades en términos de velocidad de procesamiento, consumo energético y energía disponible.

Con el fin de reducir el tiempo total tomado por el entorno AmI para proveer un conjunto de N servicios requeridos por M usuarios, un SMA compuesto por L agentes proporciona de manera paralela dichos servicios. Cada usuario especifica una solicitud de servicios, la cual está compuesta por un subconjunto de N_i servicios del conjunto total de N servicios. El entorno AmI debe asignar (o mapear) un total de $N' = N_1 + N_2 + \dots + N_M$ servicios entre los L agentes. Además, se consideran las siguientes suposiciones: (i) las solicitudes de servicios de los usuarios introducen dependencias entre los servicios requeridos, debido a

que un usuario puede especificar que ciertos servicios sean suministrados antes que otros o que el ambiente inteligente debe entregar los servicios en un determinado orden; (ii) el provisionamiento de cada servicio es una tarea atómica, es decir, no puede dividirse en servicios más pequeños; y (iii) un servicio solo puede ser asignado a un agente disponible, donde la disponibilidad del agente depende de su estado (ocupado/libre) o su energía disponible.

En la Figura 3.1 se muestra el esquema del sistema de control que representa el problema que se quiere resolver. El set point representa el ambiente inteligente, es decir, es el resultado que se quiere alcanzar producto del control efectuado por los L agentes, varios actuadores implementados que adaptan el ambiente (proceso $H(s)$), los sensores de la WSN como el elemento de realimentación y el perfil de usuario (preferencias) como entrada del sistema.

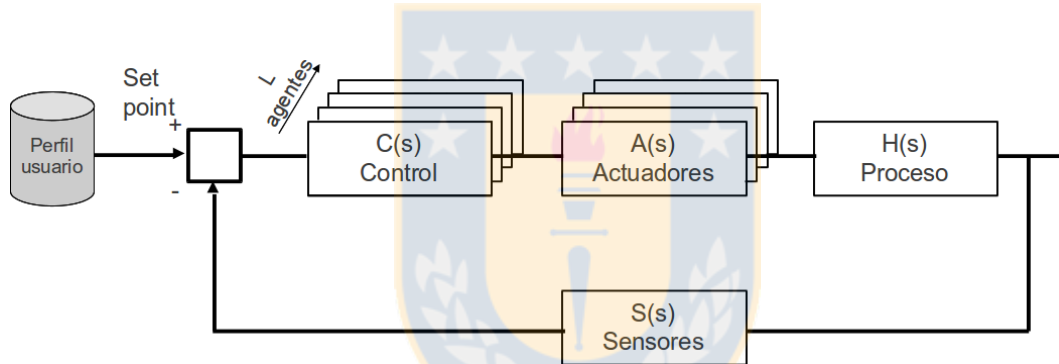


Figura 3.1: Esquema general de control del problema planteado. Fuente: Elaboración propia.

Por lo tanto, se denota la asignación de los N' servicios solicitados por los M usuarios soportada por los L agentes, usando la matriz binaria tridimensional $\mathbf{I} = ((I_{ijk}))_{M \times N \times L}$ donde el elemento (i, j, k) es definido como “1” si el servicio j solicitado por el usuario i es suministrado por el agente k , de lo contrario dicho elemento es definido como “0”. Por lo tanto, se busca obtener la mejor asignación de servicios \mathbf{I}^* a los agentes tal que el tiempo de respuesta del sistema AmI sea minimizado.

3.2. Descripción del ambiente inteligente

El modelo de AmI descrito en la Sección 2.4 es usado como herramienta para especificar la información asociada al perfil de usuario y el contexto del usuario i

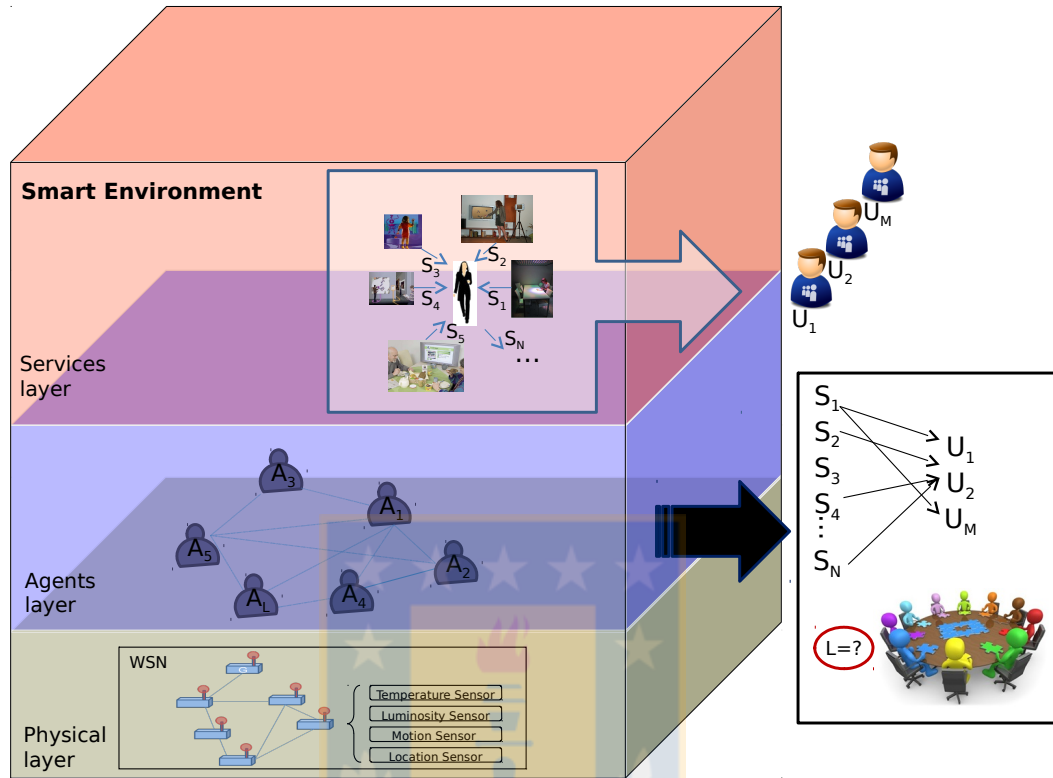


Figura 3.2: Representación del ambiente inteligente. Fuente: Elaboración propia.

representada por las variables pu_i y c_i , respectivamente.

La Figura 3.2 muestra el ambiente inteligente propuesto, el cual está dividido en tres capas: la capa física, la capa de agentes y la capa de servicios. La capa física está compuesta por una WSN que recolecta la información del entorno (por ejemplo, temperatura, luminosidad, etc.). La capa de agentes contiene el SMA, el cual es usado para distribuir los servicios solicitados por los usuarios. Debe tenerse en cuenta que como el ambiente inteligente es dinámico, el número de agentes disponibles varía dependiendo de los usuarios y los servicios a proveer, y la energía disponible de los agentes. Finalmente, la capa de servicios considera un conjunto de servicios AmI ofrecidos por el sistema a los usuarios, los cuales son especificados usando el meta-modelo genérico de AmI propuesto en [1].

Los agentes pueden exhibir diferentes capacidades dependiendo del dispositivo de cómputo donde son implementados (por ejemplo, computador de escritorio, computador portátil, *tablet*, teléfono inteligente, etc.). Además, se puede notar que existen ciertos desafíos prácticos debido a que los agentes puede ser imple-

mentados en dispositivos operados con baterías. Por lo tanto, el tiempo que toma un agente para proveer un servicio depende del tiempo de respuesta del agente y además debido a que existen diferentes servicios, dicho tiempo de respuesta dependerá de las capacidades del agente. Se define T_{ijk} como el tiempo tomado por el agente k para proporcionar el servicio j al usuario i , donde $i \in \{1, \dots, M\}$, $j \in \{1, \dots, N\}$, y $k \in \{1, \dots, L\}$. Aunque el tiempo tomado por un agente para proveer cierto servicio no depende del usuario que lo está solicitando, se ha conservado el subíndice i en la variable T_{ijk} para mantener la notación consistente.

Adicionalmente, se asume que los agentes implementados en dispositivos con capacidades de procesamiento más potentes, como un computador portátil, exhiben menores tiempos de respuesta pero un alto consumo energético, y aquellos agentes implementados en dispositivos con capacidades moderadas, como *tablets* y *smartphones*, manifiestan mayores tiempos de respuesta pero el consumo energético es menor. Teniendo en cuenta que la energía es un recurso escaso, se asume que la energía disponible en un dispositivo donde se encuentra alojado un agente no puede caer por debajo del límite ε_{\min} después de proporcionar un servicio. Se define ε_k como la energía disponible, en porcentaje, en el dispositivo donde el agente k está ejecutándose, y ϵ_k como la energía consumida, en porcentaje, por el dispositivo donde el agente k proporciona un servicio, donde $k \in \{1, \dots, L\}$. Además, la solicitud de servicios de cada usuario es representada por la colección R_i con $i \in \{1, \dots, M\}$. Finalmente, la variable binaria A_{jk} es usada para indicar la disponibilidad del agente k para proveer el servicio j en forma paralela, donde $A_{jk} = 1$ denota la disponibilidad del agente k para proporcionar el servicio j , y $A_{jk} = 0$ denota lo contrario.

3.3. Modelamiento matemático para el problema de Asignación de Servicios

La provisión óptima de servicios, la cual está enfocada en la satisfacción de los usuarios en un entorno AmI, se puede modelar matemáticamente como un problema fundamental de asignación de recursos. En primer lugar, se define el tiempo de respuesta para cada usuario como:

$$T_i(\mathbf{I}) = \sum_{j=1}^N \sum_{k=1}^L T_{ijk} I_{ijk}, \quad (3.1)$$

donde i, j, k representan al usuario, servicio y agente, respectivamente. Además, considerando la asignación de servicios especificada por \mathbf{I} , el tiempo de respuesta

del ambiente inteligente compuesto por M usuarios va a estar limitado por el agente que tarde más tiempo en proveer los servicios que tiene asignados, lo cual se puede escribir como:

$$\max_{i \in \{1, \dots, M\}} \{T_i(\mathbf{I})\} \quad (3.2)$$

Por lo tanto, el modelo de asignación de servicios busca satisfacer a los usuarios que habitan el ambiente de acuerdo a los servicios proporcionados. Para esto, este cuenta con L agentes capaces de proporcionar una serie de N servicios a un total de M usuarios, optimizando el tiempo de respuesta y considerando el uso eficiente de los recursos disponibles tales como el consumo energético y/o la cantidad de agentes inteligentes. Matemáticamente equivale a resolver el siguiente problema de minimización binaria con restricciones:

$$T(\mathbf{I}^*) = \min_{\mathbf{I} \in \{0,1\}^{M \times N \times L}} \max_{i \in \{1, \dots, M\}} \left(\sum_{j=1}^N \sum_{k=1}^L T_{ijk} I_{ijk} \right), \quad (3.3)$$

sujeto a:

$$\sum_{k=1}^L I_{ijk} = 1, \text{ para cada } i \text{ \& } j, \quad (3.4)$$

$$\sum_{k=1}^L A_{jk} \geq 1, \text{ para cada } j, \quad (3.5)$$

$$\varepsilon_k - \sum_{i=1}^M \sum_{j=1}^N \varepsilon_k I_{ijk} \geq \varepsilon_{\min}, \text{ para cada } k, \quad (3.6)$$

$$R_i = G_i(V_i, E_i), \quad \therefore G_i \subseteq G_S. \quad (3.7)$$

el cual está sujeto a cumplir con las solicitudes de los usuarios R_i , las cuales están asociadas al contexto c_i y al perfil de usuario pu_i . Vale la pena notar, que las MN restricciones en la ecuación (3.4) indican que los servicios solicitados por cada usuario pueden ser proporcionados por un solo agente; las N restricciones en la ecuación (3.5) indican que debe haber al menos un agente disponible para proporcionar los servicios; y las L restricciones en la ecuación (3.6) indican que la energía disponible de los agentes no puede estar por debajo del límite ε_{\min} .

El problema de minimización expresado en las ecuaciones (3.3)–(3.7) pertenece a la clase de problemas NP-completo debido a la naturaleza combinatoria del espacio binario de búsqueda [48].

Adicionalmente, debido a que los servicios pueden ejecutarse de forma paralela o secuencial según ciertas restricciones de procedencia o dependencias, es necesario aplicar teoría de grafos para determinar adecuadamente la forma de ejecución de ellos teniendo en cuenta el tiempo de respuesta [49, 50, 51]. Por lo tanto, se propone el digrafo de servicios $G_S = (V_S, E_S)$, donde V_S representa el conjunto de todos los servicios disponibles dentro del ambiente inteligente y E_S es el conjunto de dependencias que representan el orden topológico de los servicios. Dicho orden topológico depende de la aplicación y de los tipos de servicios, por ejemplo, si se tratara de una casa inteligente, es lógico que el ambiente inteligente antes de ofrecer cualquier otro servicio proveerá el servicio de acceso a la casa. Dado que cada usuario tiene una petición particular R_i , es necesario definir un nuevo grafo G_i asociado a la petición del usuario i ; de hecho, G_i es un subgrafo de G_S , el cual está representado por la ecuación (3.7).

En la Figura 3.3, se muestran dos ejemplos de subgrafos, los cuales están asociados a peticiones de la forma R_i que se realizan dentro de un ambiente inteligente en un entorno doméstico. En la Tabla 3.1 se definen los servicios que pueden representar cada servicio j en los subgrafos (a) y (b). Puede observarse que para el subgrafo (a) asociado a la petición del usuario 1, R_1 , es necesario que se asignen los servicios $S2$ y $S3$ antes de $S4$, mientras que para la petición del usuario 2, R_2 , los servicios $S8$ y $S9$ son independientes.



Figura 3.3: Justificación de orden topológico de grafo. Fuente: Elaboración propia.

Dado que un ambiente inteligente se caracteriza por ser reactivo y proactivo, el modelo propuesto aborda el problema de asignación de servicios desde los enfoques proactivo y reactivo.

- **Enfoque proactivo o estático:** asigna adecuadamente los servicios a todos los usuarios dado el conjunto de agentes y considera que todos los agentes

Tabla 3.1: Ejemplos de solicitud de servicios. Fuente: Elaboración propia.

Usuario	Servicio	Descripción
1	1	Encender luces de habitación a las 7 a.m
	2	Encender cafetera
	3	Encender sistema de calentamiento de ducha
	4	Encender TV
2	5	Acceso a ambiente inteligente
	6	Recomendación de habitación tipo 1
	7	Recomendación de habitación tipo 2
	8	Adecuación de temperatura en hab. 1
	9	Adecuación de luminosidad en hab. 2

están disponibles al comienzo de la asignación (con excepción de aquellos agentes alojados en dispositivos que no cumplan con el requisito de mínima energía), ya que se parte de un estado inicial fijo para el sistema AmI. Una vez iniciada la asignación, los agentes pueden estar disponibles u ocupados.

- **Enfoque reactivo o dinámico:** en este caso, se considera que el sistema AmI está realizando una determinada asignación de servicios justo en el mismo momento en que llega una nueva solicitud. Por lo tanto, debe responder a dicha solicitud sin reasignar los servicios distribuidos previamente usando los recursos (agentes y energía) disponibles.

3.4. Algoritmo de búsqueda para la Asignación de Servicios

Con el propósito de resolver el problema de asignación de servicios, se propone un algoritmo que asigna aleatoriamente un conjunto de servicios entre los agentes que cumplen la restricción de mínima energía. Por simplicidad, se consideran los deseos del usuario especificados en pu_i y el contexto asociado c_i como una petición de servicios, ya sean servicios solicitados directamente o indirectamente por el usuario, la cual está representada por R_i con $i \in \{1, \dots, M\}$, donde el elemento j en R_i es establecido como “1” si el usuario i demanda el servicio j , o de lo contrario, tal elemento es fijado como “0”.

El algoritmo de asignación de servicios lleva a cabo una búsqueda heurística basada en un criterio de acuerdo a la siguiente base lógica: los servicios deberían ser asignados con mayor probabilidad a aquellos agentes que procesen con mayor rapidez, es decir, que ejecuten los servicios en menor tiempo. Por lo tanto, el

algoritmo de asignación determina la probabilidad de elegir aleatoriamente un conjunto de agentes, quienes son responsables de coordinar la provisión de servicios. Dicho criterio selecciona un agente de acuerdo al tiempo de respuesta relativo de los agentes para proporcionar servicios, el cual induce la función de probabilidad o pmf para seleccionar aleatoriamente un agente:

$$p_K(k) = \frac{T_{ijk}^{-1}}{\sum_{\ell=1}^{L_i} T_{i,j,\ell}^{-1}}. \quad (3.8)$$

donde $k = 1, \dots, L_i$ con $L_i \leq L$ el número de agentes que satisfacen la restricción de mínima energía. Este criterio debe ser actualizado durante el proceso de búsqueda debido a que algunos agentes pueden dejar de cumplir con las restricciones de energía, por lo tanto ellos no deberían ser incluidos en la búsqueda.

3.4.1. Enfoque Proactivo

En el Algoritmo 1 se muestra un pseudocódigo para el algoritmo de búsqueda correspondiente al enfoque proactivo, en el cual se tienen unas condiciones fijas iniciales para todos los componentes del ambiente inteligente. De acuerdo al Algoritmo 1, la búsqueda es ejecutada Z veces o hasta que se satisfaga una condición de parada, y en caso de que exista más de una búsqueda con el mismo tiempo de respuesta mínimo, aquella asignación con menor consumo energético será seleccionada como la mejor asignación de servicios. La condición de parada consiste en calcular los outliers asociados al tiempo de respuesta del sistema AmI y una vez que el número de outliers sea menor a un umbral establecido previamente, la búsqueda es finalizada. Además, es importante aclarar que el orden en el cual el servicio j es elegido para asignarlo posteriormente, dependerá de la solicitud de servicios R_i del usuario i y sus dependencias con otros servicios, definidos en el grafo G_S . Para determinar el servicio que se desea asignar en un determinado momento, se realiza una búsqueda en anchura (o en inglés, *BFS - Breadth First Search*) teniendo en cuenta la variable j' para recorrer los diferentes subgrafos, donde $j' = 1, 2, \dots, N$.

3.4.2. Enfoque reactivo

El algoritmo de búsqueda correspondiente al enfoque reactivo está basado en el algoritmo asociado al enfoque proactivo, pero presenta algunas variaciones principales: (i) en el enfoque proactivo, se considera un conjunto inicial y fijo de solicitudes de varios usuarios $R = \{R_1, R_2, \dots, R_M\}$, mientras que para el enfoque reactivo este conjunto es dinámico ya que ante una nueva solicitud R'_i de un usuario i el conjunto de solicitudes es actualizado a $R = \{R, R'_i\}$ conforme llegan las solicitudes; y (ii) el enfoque proactivo busca la mejor asignación de servicios

Algorithm 1 Algoritmo heurístico para la Asignación de Servicios: Enfoque proactivo

Require: $T_{ijk}, R_i, A_{jk}, G_S, \varepsilon_k, \epsilon_k, i = 1, \dots, M, j = 1, \dots, N, k = 1, \dots, L$

Ensure: $\mathbf{I}^*, T(\mathbf{I}^*)$

Set $T(\mathbf{I}^*) \leftarrow \infty, j' \leftarrow 1$

Set $\varepsilon_k^0 \leftarrow \varepsilon_k \forall k, p_X(x)$ según un criterio de aptitud

for $z \leftarrow 1, Z$ **do**

Set $\mathbf{I} \leftarrow \mathbf{0}, \varepsilon_k \leftarrow \varepsilon_k^0 \forall k, \varepsilon_T \leftarrow 0, G_i \subseteq G_S$

while $j' \leq N$ **do**

for $i \leftarrow 1, M$ **do**

while True **do**

j \leftarrow según *BFS*

k \leftarrow un conjunto de muestras aleatorias según $p_X(x), G_i, A_{jk}$

if $\varepsilon_k - \epsilon_k \geq \varepsilon_{\min}$ **then**

$\varepsilon_k \leftarrow \varepsilon_k - \epsilon_k, I_{ijk} \leftarrow 1, \varepsilon_T \leftarrow \varepsilon_T + \epsilon_k$

break

else

$p_X(k) \leftarrow 0$, normalizar $p_X(x)$.

end if

end while

$j' \leftarrow j' + 1$

end for

end while

if $T(\mathbf{I}^*) > \max_{i \in \{1, \dots, M\}} \sum_{j=1}^N \sum_{k=1}^L T_{ijk} I_{ijk}$ **then**

$\mathbf{I}^* \leftarrow I_{ijk}$ and $\varepsilon_T^* \leftarrow \varepsilon_T$

$T(\mathbf{I}^*) \leftarrow \max_{i \in \{1, \dots, M\}} \sum_{j=1}^N \sum_{k=1}^L T_{ijk} I_{ijk}$

else if $T(\mathbf{I}^*) = \max_{i \in \{1, \dots, M\}} \sum_{j=1}^N \sum_{k=1}^L T_{ijk} I_{ijk} \ \& \ \varepsilon_T^* > \varepsilon_T$ **then**

$\mathbf{I}^* \leftarrow I_{ijk}$ and $\varepsilon_T^* \leftarrow \varepsilon_T$

$T(\mathbf{I}^*) \leftarrow \max_{i \in \{1, \dots, M\}} \sum_{j=1}^N \sum_{k=1}^L T_{ijk} I_{ijk}$

end if

end for

basado en el conjunto inicial de solicitudes, mientras que en el enfoque reactivo luego de actualizar el conjunto de solicitudes continúa buscando la mejor asignación sin modificar el plan de asignación que tenía en el momento de recibir la nueva solicitud. En Algoritmo 2 se muestra el pseudocódigo asociado al algoritmo de búsqueda para el enfoque reactivo.

De acuerdo a la complejidad del algoritmo de búsqueda, este está limitado por $\mathcal{O}(MNL)$, debido a que su complejidad aumenta linealmente en el número

Algorithm 2 Algoritmo heurístico para la Asignación de Servicios: Enfoque reactivo

Require: $T_{ijk}, R_i, A_{jk}, G_S, \varepsilon_k, \epsilon_k, i = 1, \dots, M, j = 1, \dots, N, k = 1, \dots, L$

Ensure: $\mathbf{I}^*, T(\mathbf{I}^*)$

Set $T(\mathbf{I}^*) \leftarrow \infty, j' \leftarrow 1$

Set $\varepsilon_k^0 \leftarrow \varepsilon_k \forall k, p_X(x)$ según un criterio de aptitud

for $z \leftarrow 1, Z$ **do**

Set $\mathbf{I} \leftarrow \mathbf{0}, \varepsilon_k \leftarrow \varepsilon_k^0 \forall k, \varepsilon_T \leftarrow 0, G_i \subseteq G_S$

while $j' \leq N$ **do**

for $i \leftarrow 1, M$ **do**

if existe una nueva solicitud R'_i **then**

$R = \{R, R'_i\}$, actualizar demás variables asociadas

end if

while True **do**

$\mathbf{j} \leftarrow$ según *BFS*

$\mathbf{k} \leftarrow$ un conjunto de muestras aleatorias según $p_X(x), G_i, A_{jk}$

if $\varepsilon_k - \epsilon_k \geq \varepsilon_{\min}$ **then**

$\varepsilon_k \leftarrow \varepsilon_k - \epsilon_k, I_{ijk} \leftarrow 1, \varepsilon_T \leftarrow \varepsilon_T + \epsilon_k$

break

else

$p_X(k) \leftarrow 0$, normalizar $p_X(x)$.

end if

end while

$j' \leftarrow j' + 1$

end for

end while

if $T(\mathbf{I}^*) > \max_{i \in \{1, \dots, M\}} \sum_{j=1}^N \sum_{k=1}^L T_{ijk} I_{ijk}$ **then**

$\mathbf{I}^* \leftarrow I_{ijk}$ and $\varepsilon_T^* \leftarrow \varepsilon_T$

$T(\mathbf{I}^*) \leftarrow \max_{i \in \{1, \dots, M\}} \sum_{j=1}^N \sum_{k=1}^L T_{ijk} I_{ijk}$

else if $T(\mathbf{I}^*) = \max_{i \in \{1, \dots, M\}} \sum_{j=1}^N \sum_{k=1}^L T_{ijk} I_{ijk} \ \& \ \varepsilon_T^* > \varepsilon_T$ **then**

$\mathbf{I}^* \leftarrow I_{ijk}$ and $\varepsilon_T^* \leftarrow \varepsilon_T$

$T(\mathbf{I}^*) \leftarrow \max_{i \in \{1, \dots, M\}} \sum_{j=1}^N \sum_{k=1}^L T_{ijk} I_{ijk}$

end if

end for

de usuarios, servicios y agentes. Lo anterior se puede ver en los algoritmos para ambos enfoques, ya que cada algoritmo posee tres ciclos anidados. El ciclo interior asociado al agente k es ejecutado L veces con el fin de hallar un agente candidato para la asignación de un determinado servicio; el bucle intermedio corresponde a los usuarios haciendo que se repita lo anterior M veces; y el ciclo exterior

corresponde a los servicios, el cual hace que se ejecute lo anterior N veces más.



Evaluación de Modelo de Asignación de Servicios

En este capítulo se presentan los resultados del modelo propuesto en el Capítulo 3, tanto para el enfoque proactivo como para el reactivo. Además se implementa otra alternativa para la asignación de servicios con el fin de compararla con el modelo propuesto.

4.1. Consideraciones de implementación del Modelo de Asignación de Servicios

Para evaluar la capacidad de los algoritmos diseñados en la sección 3.4, se usó un conjunto de servicios asociado a tres usuarios ($M = 3$) usando un conjunto de L agentes. Para esto, se usa un grafo compuesto por $N = 100$, donde cada nodo representa un servicio en particular dentro del ambiente inteligente como se muestra en la Figura 4.1(a). Por ejemplo, $j = 46$ es el nodo raíz y corresponde al servicio de acceder al ambiente inteligente, mientras que $j = 78$ es un nodo hijo y podría corresponder a la adecuación de temperatura de un lugar específico dentro del ambiente inteligente. Para cada usuario i (con $i = 1, 2, 3$) se tiene una determinada solicitud de servicios R_i asociada al subgrafo G_i , donde $N_1 = 40$, $N_2 = 32$ y $N_3 = 20$ corresponden al número de servicios solicitados por el usuario 1, 2 y 3, respectivamente. Los subgrafos G_1 , G_2 y G_3 son ilustrados en 4.2(b)-(d), donde para cada subgrafo los nodos son resaltados en color rojo, cian y amarillo, respectivamente.

Adicionalmente, se considera que los servicios pueden ser asignados a los agentes en diferentes intervalos de tiempo que no se superponen llamados *time slots* (TS), en consecuencia se define el tiempo tomado por cualquier agente para proporcionar un determinado servicio como nTS , donde $n \in \mathbb{N}$. Los valores de los

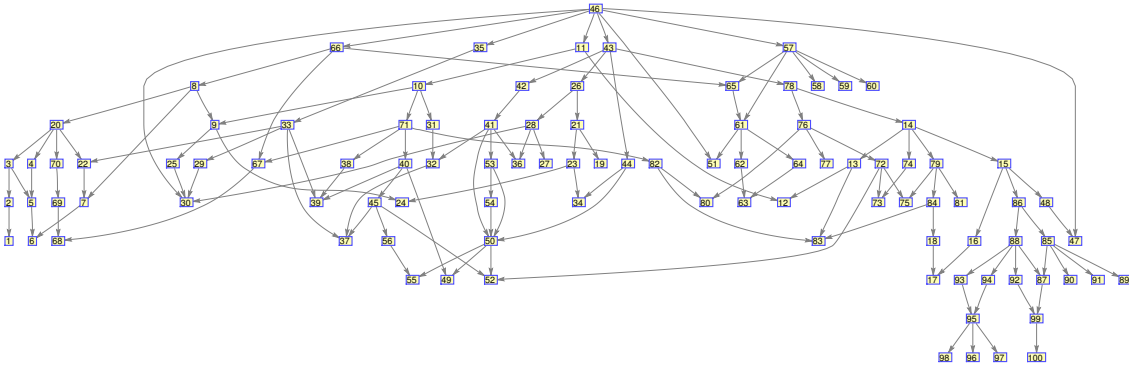


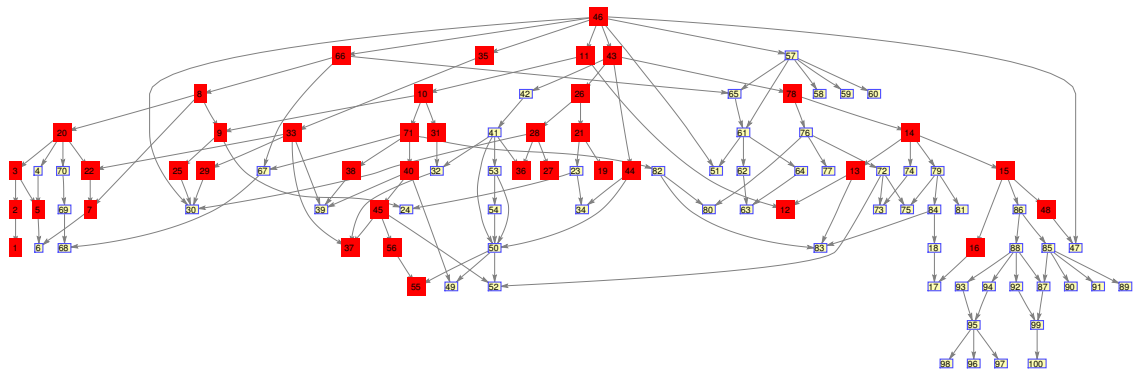
Figura 4.1: Representación del grafo de servicios del ambiente inteligente. Fuente: MATLAB.

parámetros usados en nuestros cálculos tales como el tiempo de respuesta de los agentes y su energía disponible en $t = 0$ son definidos arbitrariamente como se registra en la Tabla 4.1. Además, se asume que el agente k toma el mismo tiempo T_{ijk} para proveer todos los servicios, donde $TS = 1s$ (asociado al tiempo del agente más rápido), y el consumo de energía de cada agente depende de las capacidades de procesamiento de los dispositivos donde ellos son implementados y es independiente del tipo de servicio. Por esta razón, tres tipos de dispositivos son considerados: computador portátil, *tablet*, y teléfono inteligente, con un consumo energético basado en los valores típicos de dispositivos electrónicos in [52] de 0.16 kWh, 0.04 kWh y 0.02 kWh, respectivamente (correspondiente a un consumo normalizado de 1.6 %, 0.4 % y 0.2 %, respectivamente). Por lo tanto, dispositivos con óptimas capacidades de procesamiento, como un computador portátil, exhiben tiempos de respuesta más cortos pero consumen mayor energía y dispositivos con capacidades de procesamiento moderadas, como un teléfono inteligente, manifiesta tiempos de respuesta mayores pero consume menos energía. Finalmente, los servicios podrán ser asignados a un agente si y solo si su energía disponible excede el límite de energía correspondiente a $\varepsilon_{\min} = 20\%$.

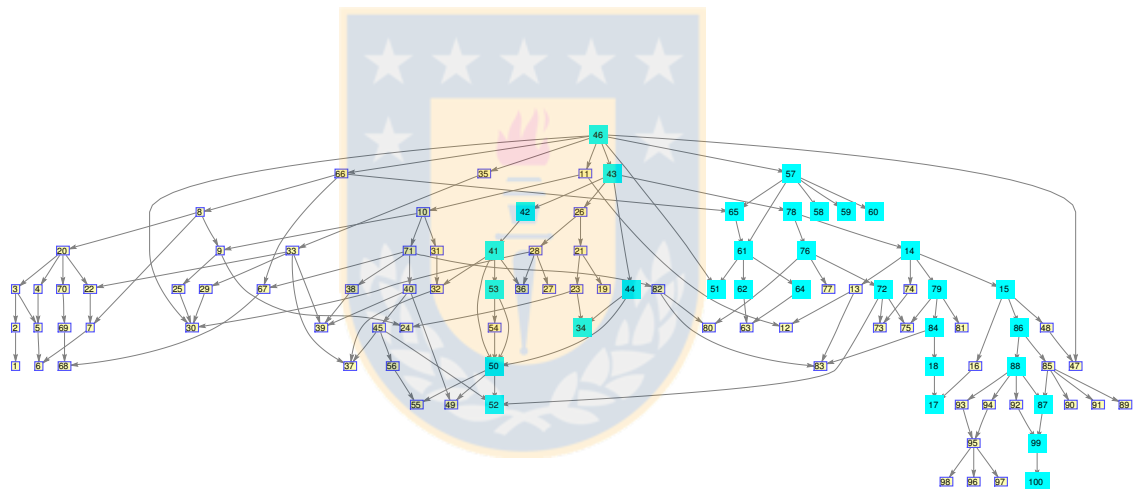
Por otra parte, para el enfoque reactivo, además de los parámetros indicados en la Tabla 4.1, se consideró la nueva solicitud R'_1 ilustrada en la Figura 4.3, que representa que el usuario 1 solicita una nueva solicitud en un determinado momento.

A modo de ejemplo, el modelo genérico de asignación de servicios se particulariza en una aplicación de domótica, la cual está basada en una casa inteligente. En este caso, según la Sección 2.4, el perfil de usuario podría estar conformado por la siguiente información:

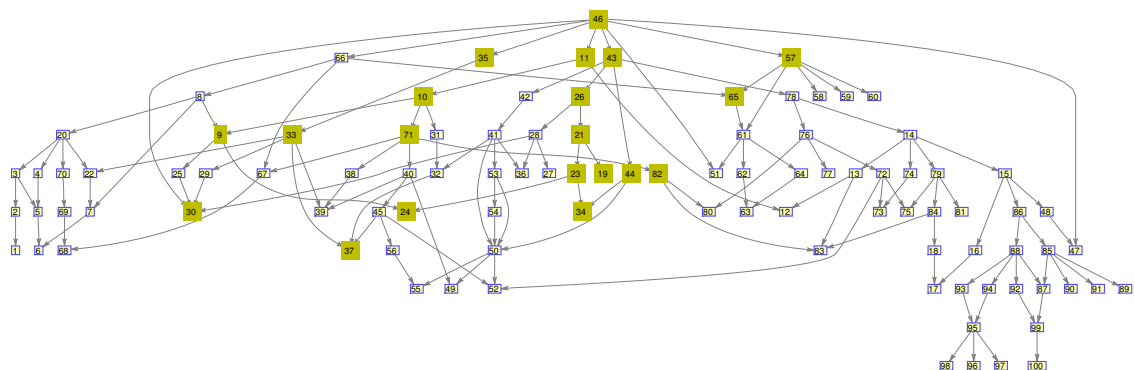
- Gustos: $pu_i = \{\text{escuchar música, ver TV, tipo música, tipo programa tele-}$



(a) Usuario 1



(b) Usuario 2



(c) Usuario 3

Figura 4.2: Representación de los subgrafos de servicios asociados a tres usuarios.
Fuente: MATLAB.

Tabla 4.1: Valores de los parámetros del sistema AmI. Fuente: Elaboración propia.

Agente	T_{ijk} [s]	ε_j %	ϵ_j %
1	1	60	1.6
2	2	40	0.4
3	1	50	1.6
4	1	30	1.6
5	3	20	0.2
6	3	35	0.2
7	1	80	1.6
8	2	25	0.4
9	1	15	1.6
10	2	90	0.4

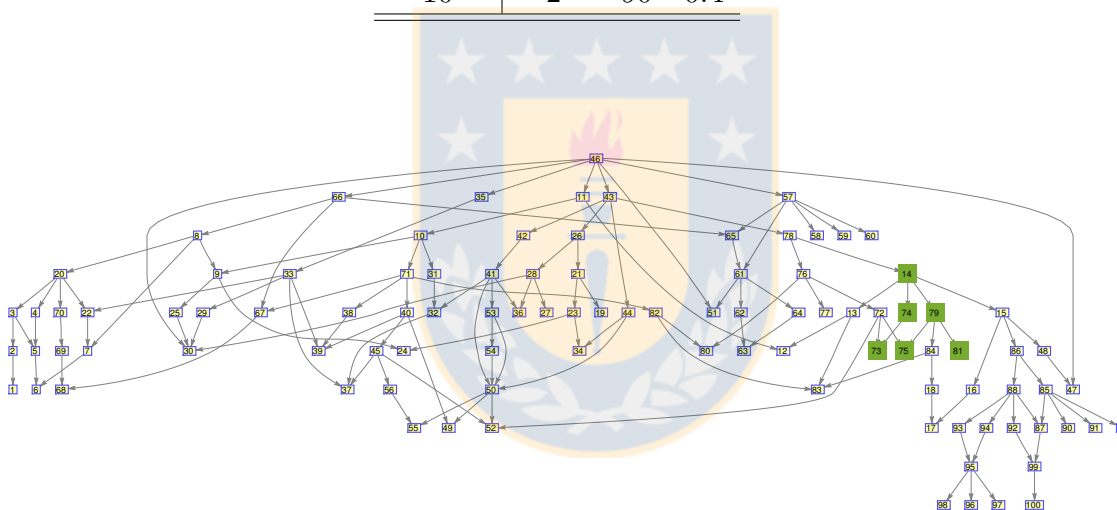


Figura 4.3: Representación del subgrafo asociado a nueva solicitud. Fuente: Elaboración propia.

visivo, tipo comida}.

- Preferencias e intereses: $pu_i = \{\text{nivel de calefacción (cálido, normal, frío), nivel de luminosidad (brillante, normal, tenue), tipo de ambiente (ameno, relajante, informado)}\}$.
- Información personal: $pu_i = \{\text{RFID, profesión, edad}\}$.

Mientras que el contexto contiene información acerca de:

- Contexto tecnológico: $c_i = \{\text{tipo de dispositivo de usuario (laptop, celular, tablet)}\}$.

- Contexto espacio-temporal: $c_i = \{\text{ubicación usuario, tiempo}\}$.
- Contexto social: $c_i = \{\text{categoría social (estudiante, profesional, jubilado)}\}$.
- Contexto del entorno: $c_i = \{\text{medición de temperatura, medición de luminosidad}\}$.

Por lo tanto, basados en estos perfiles de usuario pu_i y contextos c_i , algunos de los N servicios que puede proveer el ambiente inteligente planteado en la Figura 4.1 son:

- Adecuación de iluminación (brillante, normal, tenue)
- Adecuación de temperatura (cálido, normal, frío)
- Proporción de música (clásica, rock, pop)
- Despliegue de información personalizada dependiendo de perfil de usuario o contexto
- Ingreso a la casa inteligente

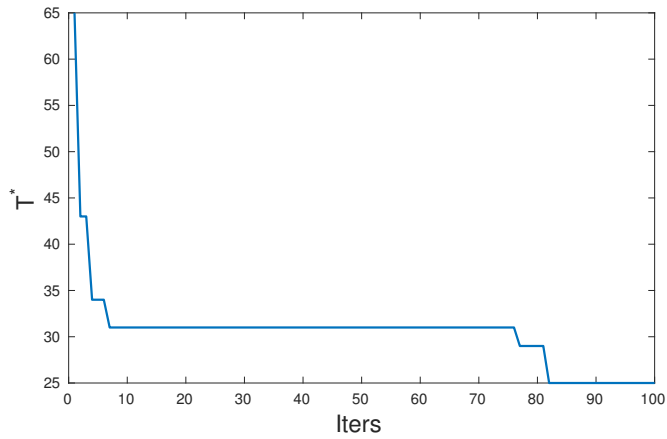
4.2. Evaluación de estrategias de Asignación de servicios

El algoritmo de asignación de servicios para cada estrategia o enfoque descrita en la sección 3.4 ha sido implementado en MATLAB y los resultados son registrados en la Tabla 4.2, la cual incluye el tiempo de respuesta para cada usuario i y el sistema AmI y la energía consumida por el sistema completo. Además, la Figura 4.4 y la Figura 4.5 muestran el tiempo de respuesta, el número de servicios asignados a cada agente y el consumo energético del sistema AmI para ambas estrategias.

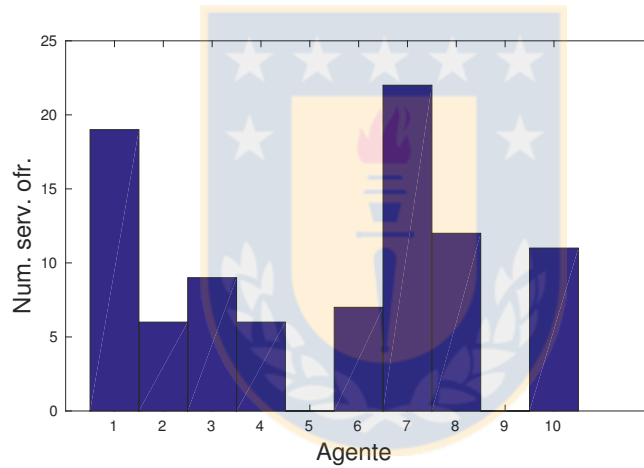
Tabla 4.2: Tiempo mínimo de respuesta y consumo energético de enfoques proactivo y reactivo. Fuente: Elaboración propia.

Enfoque	$T(\mathbf{I}^*)$ [s]				Energía Cons. kWh
	u_1	u_2	u_3	Sist. AmI	
Proactivo	25	23	16	25	10.26
Reactivo	30	28	19	30	11.16

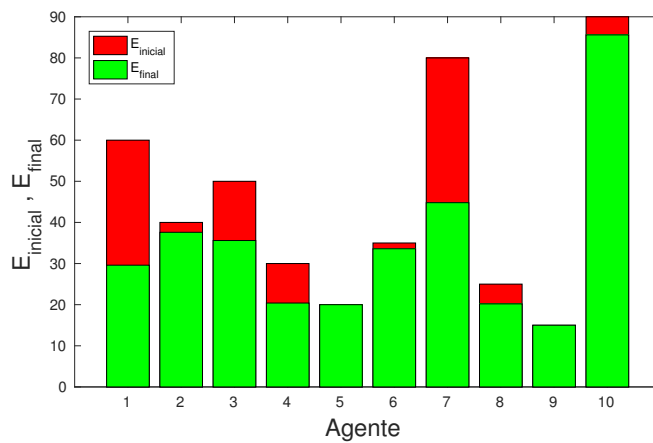
Como era de esperarse, el tiempo de respuesta para el enfoque reactivo fue mayor al enfoque proactivo debido a la presencia de una nueva solicitud, y a pesar de que no exige una reasignación, esta debe sumarse al conjunto de servicios a asignar. Lo anterior se traduce en mayor número de servicios para distribuir y así un mayor tiempo de respuesta y consumo energético del sistema AmI.



(a) Tiempo de respuesta del sistema

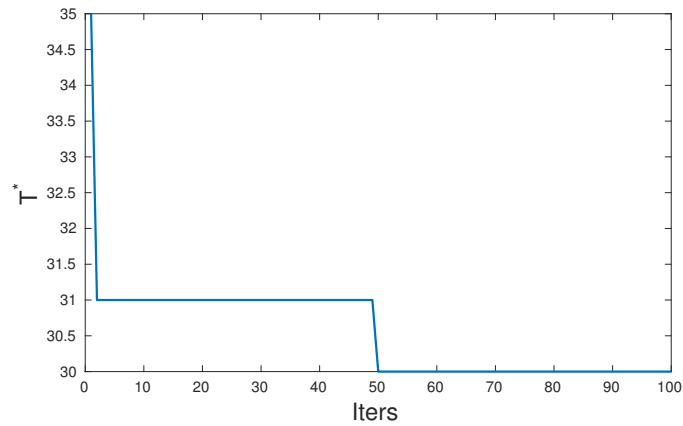


(b) Asignación de Servicios

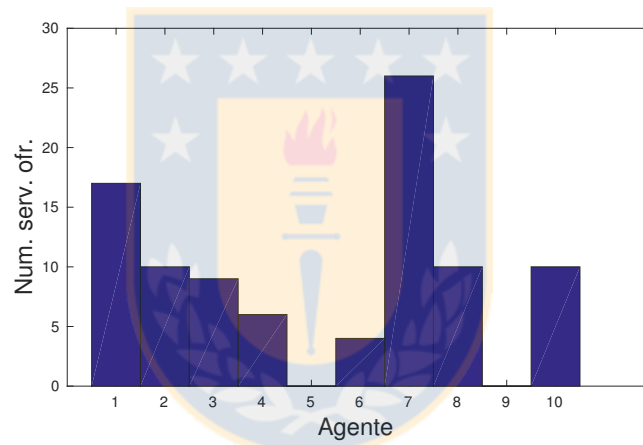


(c) Consumo de energía del sistema

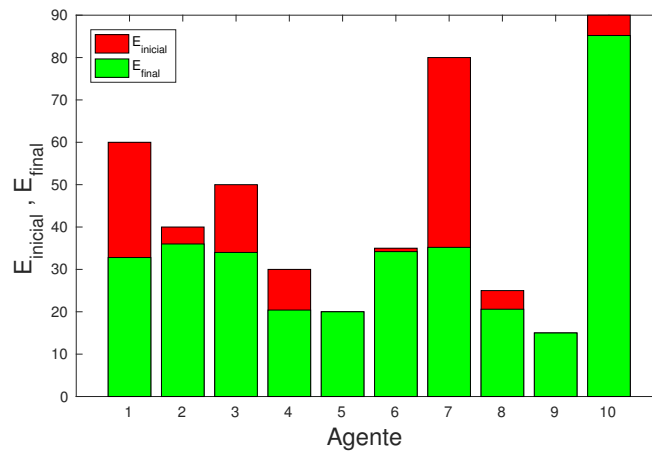
Figura 4.4: *Enfoque proactivo.* Tiempo mínimo de respuesta, Asignación de Servicios, y consumo de energía de agentes para el sistema AmI considerado. Fuente: MATLAB.



(a) Tiempo de respuesta del sistema



(b) Asignación de Servicios



(c) Consumo de energía del sistema

Figura 4.5: *Enfoque reactivo.* Tiempo mínimo de respuesta, Asignación de Servicios, y consumo de energía de agentes para el sistema AmI considerado. Fuente: MATLAB.

4.3. Análisis comparativo

4.3.1. Enfoque *Greedy*

Para fines de comparación, se implementó un algoritmo *greedy* para resolver el problema de asignación de recursos, el cual toma decisiones en función de la información que está disponible en cada momento pero no siempre garantiza alcanzar la solución óptima. El enfoque *greedy* asigna los servicios a los agentes más rápidos, es decir, aquellos que tengan los mínimos tiempos de respuesta, siempre y cuando estén disponibles. Este algoritmo es análogo al algoritmo proactivo, pero tiene dos principales diferencias: (i) el algoritmo *greedy* es ejecutado solo una vez (no es probabilístico); (ii) el agente que proporciona un determinado servicio es escogido al azar del conjunto de agentes más rápidos del sistema AmI. En Algoritmo 3 se muestra el pseudocódigo para el algoritmo *greedy*.

Algorithm 3 Algoritmo *Greedy* para la Asignación de Servicios

Require: $T_{ijk}, R_i, A_{jk}, G_S, \varepsilon_k, \epsilon_k, i = 1, \dots, M, j = 1, \dots, N, k = 1, \dots, L$

Ensure: \mathbf{I}^* and $T(\mathbf{I}^*)$

Set $j' \leftarrow 1$

Set $\varepsilon_k^0 \leftarrow \varepsilon_k \forall k$

Set $\mathbf{I} \leftarrow \mathbf{0}, \varepsilon_k \leftarrow \varepsilon_k^0 \forall k, \varepsilon_T \leftarrow 0$, and $G_i \subseteq G_S$

while $j' \leq N$ **do**

for $i \leftarrow 1, M$ **do**

while True **do**

$j \leftarrow$ según *BFS*

$k \leftarrow$ un agente seleccionado del conjunto de agentes más rápidos, según G_i and A_{jk}

if $\varepsilon_k - \epsilon_k \geq \varepsilon_{\min}$ **then**

$\varepsilon_k \leftarrow \varepsilon_k - \epsilon_k, I_{ijk} \leftarrow 1$, and $\varepsilon_T \leftarrow \varepsilon_T + \epsilon_k$

break

else

$A_{jk} \leftarrow 0 \quad \forall j$

end if

end while

$j' \leftarrow j' + 1$

end for

end while

$\mathbf{I}^* \leftarrow I_{ijk}$ and $\varepsilon_T^* \leftarrow \varepsilon_T$

$$T(\mathbf{I}^*) \leftarrow \max_{i \in \{1, \dots, M\}} \sum_{j=1}^N \sum_{k=1}^L T_{ijk} I_{ijk}$$

Los resultados del algoritmo *greedy* de asignación de servicios son registrados en la Tabla 4.3, la cual incluye el tiempo de respuesta para cada usuario i y el sistema AmI y la energía consumida por el sistema completo. Además, la Figura 4.6

muestra el número de servicios asignados a cada agente y el consumo energético del sistema AmI.

Tabla 4.3: Tiempo mínimo de respuesta y consumo energético de enfoque *greedy*. Fuente: Elaboración propia.

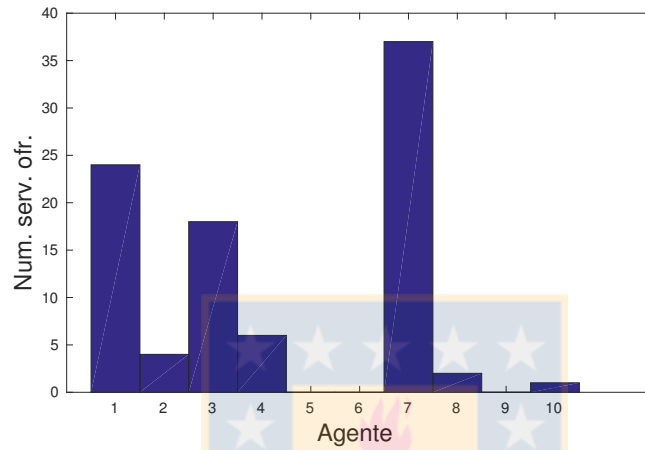
Enfoque	$T(\mathbf{I}^*)$ [s]				Energía Cons.
	u_1	u_2	u_3	Sist. AmI	kWh
<i>Greedy</i>	38	35	22	38	13.88

A partir de los resultados, puede observarse lo siguiente: (i) los servicios son distribuidos entre los agentes más aptos, es decir, aquellos agentes más rápidos y con óptimas características para proveer servicios proveerán más servicios que los demás; (ii) los tiempos de respuesta mejoran considerablemente cuando las tareas son divididas entre los agentes más competentes; (iii) es claro que los agentes 5 y 9 no tienen servicios asignados debido a que su energía inicial está por debajo del límite fijado; (iv) en el enfoque proactivo y reactivo, aunque es más probable asignar los servicios a los agentes más rápidos (1 y 7), si existieran otros agentes disponibles con suficiente energía (8 y 10) podrían ser también utilizados en la asignación; (v) se espera que los servicios sean asignados a los agentes 1, 3, 4 y 7 en el enfoque *greedy*, de acuerdo a la Tabla 4.1 pero a medida que los agentes más rápidos no superan el límite de energía, el algoritmo elige agentes intermedios (por ejemplo, 2, 8 y 10) para continuar asignando los servicios solicitados; (vi) el tiempo de respuesta y el consumo energético del enfoque reactivo son directamente proporcionales al tamaño del conjunto total de solicitudes de servicios; y (vii) el enfoque proactivo y reactivo ofrecen mejores resultados que el *greedy* respecto a menores tiempos de respuesta y consumo energético del sistema AmI.

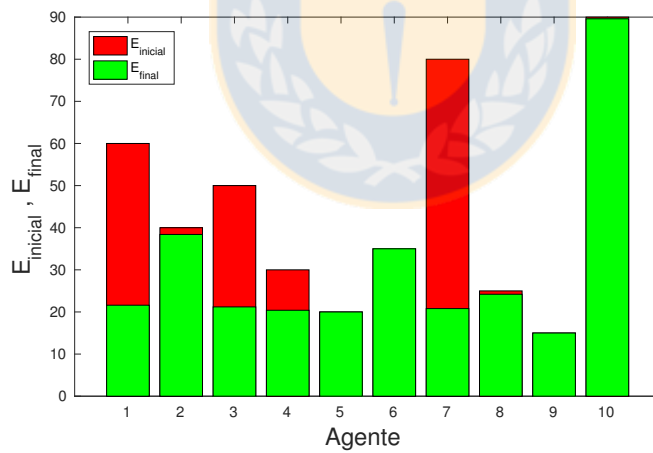
4.3.2. Comparación de escenarios

Para destacar la capacidad de ahorro energético que presenta el modelo de asignación de servicios propuesto, se efectúa una comparación con un escenario que no presenta restricciones energéticas. Los resultados para los tres enfoques en el nuevo escenario son registrados en la Tabla 4.4, la cual incluye el tiempo de respuesta para cada usuario i y el sistema AmI y la energía consumida por el sistema completo. Se usan los mismos parámetros que en la Tabla 4.1, modificando únicamente la energía inicial $\varepsilon_k = \infty$.

Comparando los resultados de la Tabla 4.4 con los obtenidos en las Tablas 4.2 y 4.3 se puede ver que aunque el modelo de asignación propuesto arroja tiempos



(a) Asignación de Servicios



(b) Consumo de energía del sistema

Figura 4.6: *Enfoque Greedy*. Asignación de Servicios, y consumo de energía de agentes para el sistema AmI considerado. Fuente: MATLAB.

Tabla 4.4: Tiempo mínimo de respuesta y consumo energético de escenario comparativo. Fuente: Elaboración propia.

Enfoque	$T(\mathbf{I}^*)$ [s]				Energía Cons. kWh
	u_1	u_2	u_3	Sist. AmI	
Proactivo	18	19	10	19	12.22
Reactivo	19	20	14	20	12.34
<i>Greedy</i>	26	26	13	26	14.72

de respuesta mayor, la energía consumida es considerablemente menor. Según lo anterior, la consideración de energía por el modelo propuesto beneficia el ahorro energético en sistemas donde esta es escasa o costosa. Además, como es de esperarse, en este caso todos los agentes pueden tener servicios asignados, a excepción del enfoque *greedy* que elige los agentes más rápidos como se muestra en la Figura 4.7.

Según [53], la evaluación a través de métricas de un determinado sistema o proceso resulta ser un método adecuado para medir su calidad o desempeño de forma cuantitativa. Por lo tanto, para evaluar el desempeño del modelo de asignación de servicios propuesto se usa el índice de equidad de Jain [54]. Esta métrica mide que tan equitativa es la asignación de recursos a un conjunto de usuarios, cuyo k usuario recibe una asignación x_k , y se define como:

$$\mathcal{J}(x_1, x_2, \dots, x_L) = \frac{(\sum_{k=1}^L x_k)^2}{L \sum_{k=1}^L x_k^2}. \quad (4.1)$$

En este caso x_k representa los servicios asignados a cada agente k . El valor del índice de equidad oscila entre 0 y 1, donde 1 (100 %) significa una asignación totalmente justa. Además, se calcula la varianza, la cual también es utilizada en la literatura como métrica para determinar que tan equitativa es cierta asignación en un sistema distribuido, donde un valor alto de varianza representa una mayor dispersión y una asignación poco equitativa. En la tabla 4.5 pueden observarse las métricas evaluadas para el enfoque proactivo y greedy en ambos escenarios. Es de esperarse, que según los resultados obtenidos anteriormente, el enfoque reactivo presente unos valores similares al enfoque proactivo.

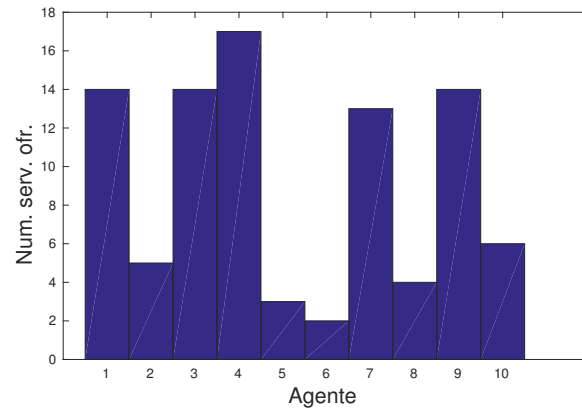
En general, el enfoque proactivo obtuvo mejores resultados, donde el índice de equidad presenta un valor cercano a 1, lo que verifica que el modelo de asignación de servicios es equitativo a la hora de distribuir los servicios entre los agentes. Además, la varianza fue mucho menor que la del enfoque *greedy*, lo que implica que la asignación trata de ser justa teniendo en cuenta las restricciones del problema de asignación. Por lo tanto, los resultados obtenidos de la implementación del modelo de asignación de servicios propuesto, en términos de varias métricas, han sido

Tabla 4.5: Evaluación del desempeño de modelo propuesto. Fuente: Elaboración propia.

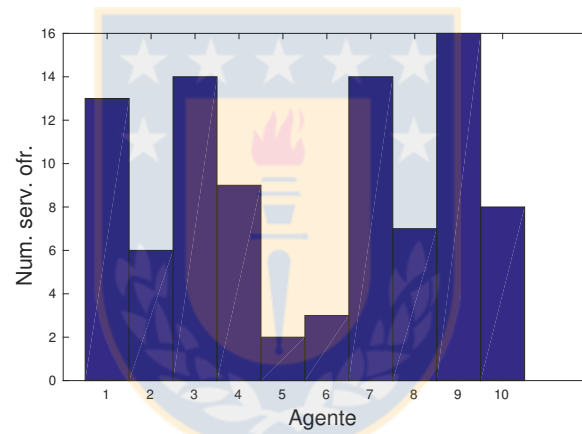
Enfoque	Índice de equidad (\mathcal{J})	Varianza	Restricciones energéticas
Proactivo	0.65	51.73	✓
	0.75	32.18	×
<i>Greedy</i>	0.36	164.4	✓
	0.49	98.62	×

satisfactorios y ayudan a demostrar la capacidad del modelo para proporcionar una asignación óptima en sistemas AmI.

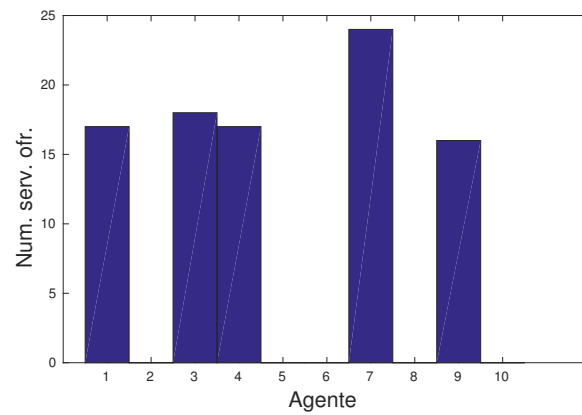




(a) Enfoque proactivo



(b) Enfoque reactivo



(c) Enfoque greedy

Figura 4.7: Asignación de Servicios en escenario comparativo. Fuente: MATLAB.

Conclusiones y Trabajo Futuro

Un sistema AmI que pretenda ser aplicado en situaciones reales debe ser diseñado teniendo en cuenta un cierto contexto y una serie de capacidades que le permitan adaptarse a las condiciones dinámicas del ambiente y, simultáneamente, a los requerimientos de los usuarios. Para lograr esto, es necesario que un sistema AmI sea capaz de abstraer, dinámicamente, el entorno y sus usuarios mediante algún modelo apropiado, y a su vez, es necesario asignar los servicios y coordinar adecuadamente los elementos que están encargados de proporcionar la inteligencia para llegar a una solución acertada en el menor tiempo posible y/o con el menor costo posible.

En esta tesis de doctorado se propone un modelo matemático que representa un problema de asignación de recursos fundamental en un ambiente inteligente, el cual tiene como propósito mejorar la provisión de servicios en ambientes donde la variedad y cantidad de servicios aumentan conforme crecen las necesidades de las personas. Este modelo pretende satisfacer a los usuarios quienes habitan el ambiente de acuerdo a los servicios proporcionados por un conjunto de elementos inteligentes representados como agentes y quienes son útiles para soportar la dinámica de un ambiente inteligente. Para esto, se exploran tres estrategias: el enfoque proactivo y reactivo están basados en una búsqueda heurística mientras que el enfoque greedy es determinista y es usado para evaluar el modelo propuesto.

A pesar de la complejidad del problema de asignación de servicios (asociada a su naturaleza combinatorial), este puede ser resuelto a través de heurísticas, métodos probabilísticos y/o determinísticos. Además, soluciones a este problema de asignación de servicios ofrecen avances científicos al novedoso paradigma de la AmI, lo cual genera aportes significativos para el sensado y control de variables físicas, para el mapeo y localización de usuarios y para la toma de decisiones y

adaptación del entorno.

Con los resultados del modelo de asignación de servicios se ha demostrado que los servicios son distribuidos entre los agentes más aptos, es decir, aquellos más rápidos y con suficiente energía disponible para proveer los servicios a los usuarios. Además, se destaca que la importancia del problema resuelto en este trabajo consiste en lograr una adecuada coordinación y distribución de servicios a los usuarios, apoyados por elementos inteligentes, para mejorar la prestación de servicios en entornos AmI.

El trabajo investigativo propuesto es relevante actualmente debido a que considera las dependencias de los servicios y la disponibilidad de los componentes dentro del entorno AmI. Además, este conduciría a mejorar el confort y la eficiencia energética en ambientes inteligentes, reduciendo los tiempos de respuesta y la sobrecarga del sistema. Finalmente, el problema de asignación de servicios se aborda desde un modelo general lo que involucra independencia del dominio de aplicación, extrapolación a tipos similares de sistemas, entre otros aspectos.

5.1. Trabajo futuro

Como trabajo futuro se tiene previsto validar el modelo a través de la implementación de un prototipo real, implementando una WSN compuesta por dispositivos hardware reales, la cual efectúe el monitoreo y actuación, en tiempo real; y un SMA en alguna plataforma para la creación y alojamiento de agentes inteligentes (por ejemplo, JADE). Con lo anterior, se busca proveer un conjunto de servicios en un dominio de aplicación determinado, donde interactúen naturalmente varios usuarios con el ambiente inteligente, teniendo en cuenta sus preferencias y el estado del entorno, y aprovechando además las características de distribución, comunicación y paralelismo de los agentes.

Además, se pretende desarrollar e implementar nuevos algoritmos y/o modelos de predicción en ambientes inteligentes para modelar la proporción de servicios anticipados a habitantes del ambiente inteligente, los cuales se complementen con el modelo de asignación de servicios propuesto en esta tesis.

Bibliografía

- [1] S. E. Restrepo, J. E. Pezoa, and D. A. Ovalle-Carranza. An adaptive architecture for ambient intelligence based on meta-modeling, smart agents, and wireless sensor networks. *IEEE Latin America Transactions*, 12(8):1508–1514, 2014.
- [2] S. E. Restrepo, P. Pinaud, J. E. Pezoa, and S. K. Sobarzo. Energy-aware image allocation for distributed video processing on handheld devices. In *32nd International Performance, Computing, and Communications Conference (IPCCC 2013)*, San Diego, CA, USA, December 2013.
- [3] G. Conte, G. Morganti, A. M. Perdon, and D. Scaradozzi. Multi-agent system theory for resource management in home automation systems. *Journal of Physical Agents*, 3(2), 2009.
- [4] G. Zhao, Ailiya, and Z. Shen. Intrinsically motivated agent for service management in smart homes. In *2012 Southeast Asian Network of Ergonomics Societies Conference (SEANES 2012)*, Langkawi, Kedah, 2012.
- [5] Joo-Hyung Kim, Dong Won Kim, Bum-Jae You, and Gwi-Tae Park. A design of framework for smart services of robots in intelligent environment. In *Control and Automation*, pages 55–61. Springer, 2009.
- [6] D. I. Tapia, R. S. Alonso, F. De la Prieta, C. Zato, S. Rodríguez, E. Corchado, J. Bajo, and J. M. Corchado. Sylph: An ambient intelligence based platform for integrating heterogeneous wireless sensor networks. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–8, July 2010.
- [7] Maddy D. Janse. Amigo in hindsight - lessons learned. In *2012 Workshop on Ambient Intelligence Infrastructures (WAmI)*, pages 3–6, Pisa, Italy, 2012.

- [8] S. Makonin, L. Bartram, and F. Popowich. A smarter smart home: Case studies of ambient intelligence. *Pervasive Computing, IEEE*, 12(1):58–66, Jan 2013.
- [9] Myheart project. [Online] <http://www.hitech-projects.com/euprojects/myheart/>, 2004. Último acceso: Feb. 2014.
- [10] P. Mikulecky. Ambient intelligence in water management. In *Intelligent Environments (IE), 2011 7th International Conference on*, pages 302–305, 2011.
- [11] J.C. Augusto, R. Wichert, R. Collier, D.V. Keyson, A. A. Salah, and A.–H. Tan, editors. *Ambient Intelligence - 4th International Joint Conference, Aml 2013, Dublin, Ireland, December 3-5, 2013. Proceedings*, volume 8309 of *Lecture Notes in Computer Science*. Springer, 2013.
- [12] T. P. Huynh, Y.K. Tan, and K. J. Tseng. Energy-aware wireless sensor network with ambient intelligence for smart led lighting system control. In *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, pages 2923–2928, Nov 2011.
- [13] J. Hernández, S. E. Restrepo, T. Muñoz, and D. Ovalle. Modelo de un sistema híbrido multi-agente para la detección y control de incendios ambientales simulados: STOP-FIRE. In *V Congreso Colombiano Computación (5CCC)*, 2010.
- [14] F. Sebbak, A Mokhtari, A Chibani, and Y. Amirat. Context-aware ubiquitous framework services using jade-osgi integration framework. In *Machine and Web Intelligence (ICMWI), 2010 International Conference on*, pages 48–53, Algiers, Algeria, Oct 2010.
- [15] V. Zamudio, R. Baltazar, M. A. Casillas, and V. Callaghan. c-inpres: Coupling analysis towards locking optimization in ambient intelligence. In *Intelligent Environments (IE), 2010 Sixth International Conference on*, pages 68–73, 2010.
- [16] J. Vales-Alonso, P. López-Matencio, J.J. Alcaraz, J.L. Sieiro-Lomba, E. Costa-Montenegro, and F.J. González-Castaño. *Human-Computer Systems Interaction: Backgrounds and Applications 2*, volume 98, chapter A Dynamic Programming Approach for Ambient Intelligence Platforms in Running Sports Based on Markov Decision Processes, pages 165–181. Springer Berlin Heidelberg, 2012.

- [17] J. E. Pezoa. Optimizing mission allocation in wireless sensor networks under geographically correlated failures. In *11th ACM Conference on Embedded Networked Sensor Systems (SenSys 2013)*, Rome, Italy, November 2013.
- [18] Giuseppe Colistra, Virginia Pilloni, and Luigi Atzori. The problem of task allocation in the internet of things and the consensus-based approach. *Computer Networks*, 73(0):98 – 111, 2014.
- [19] Kevin L. Mills. A brief survey of self-organization in wireless sensor networks. *Wireless Communications and Mobile Computing*, 7(7):823–834, 2007.
- [20] D. F. Glas, S. Satake, F. Ferreri, T. Kanda, and N. Hagita. The network robot system: enabling social human-robot interaction in public spaces. *Journal of Human-Robot Interaction*, 1(2):5–32, 2012.
- [21] D. L. Janovy, J. Smith, H. J. Siegel, and A. A. Maciejewski. Models and heuristics for robust resource allocation in parallel and distributed computing systems. In *2007 IEEE International Parallel and Distributed Processing Symposium*, pages 1–5, March 2007.
- [22] Andy M. Connor and Amit Shah. Resource allocation using metaheuristic search. *CoRR*, abs/1605.01855, 2016.
- [23] N. Carretero and A. Bermejo. *Inteligencia ambiental*. Centro de Difusión de Tecnologías, Universidad Politécnica de Madrid, 2005.
- [24] G. M. P. O’Hare, Michael J. O’Grady, Stephen Keegan, Donal O’Kane, Richard Tynan, and David Marsh. Intelligent agile agents: active enablers for ambient intelligence. In *Ambient Intelligence and Scientific Discovery Workshop at CHI 2004 Conference on Human Factors in Computing Systems*, Vienna, Austria, 2004.
- [25] Anind Kumar Dey. *Providing Architectural Support for Building Context-aware Applications*. PhD thesis, Atlanta, GA, USA, 2000.
- [26] E. Aarts and S. Marzano. *The New Everyday: Views on Ambient Intelligence*. 010 Publishers, Rotterdam, The Netherlands, 2003.
- [27] J. A. Corrales, F. Torres, and F. A. Candelas. *Tecnologías en la inteligencia ambiental*, 2006.
- [28] S.J. Russell and P. Norvig. *Inteligencia artificial: un enfoque moderno*. Colección de Inteligencia Artificial de Prentice Hall. Pearson Educación, 2004.

- [29] Michael Wooldridge. *An Introduction to MultiAgent Systems*. Wiley Publishing, 2nd edition, 2009.
- [30] J. A. Pérez. *Sahara: Arquitectura de seguridad integral para sistemas de agentes móviles basados en Java*. PhD thesis, Universidad de Oviedo, Departamento de Informática, Oviedo, España, Febrero 2000.
- [31] Inmaculada Ayala, Mercedes Amor, and Lidia Fuentes. A model driven engineering process of platform neutral agents for ambient intelligence devices. *Autonomous Agents and Multi-Agent Systems*, pages 214–255, Mar 2014.
- [32] E. Cheong. *Actor-oriented Programming for Wireless Sensor Networks*. University of California, Berkeley, 2007.
- [33] Feng Zhao and Leonidas Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [34] Paolo Santi. *Topology Control in Wireless Ad Hoc and Sensor Networks*. John Wiley & Sons Ltd., England, 2005.
- [35] Ivan Stojmenovic. *Handbook of Sensor Networks: Algorithms and Architectures*. John Wiley & Sons Inc., Canadá, 2005.
- [36] George B. Dantzig and Mukund N. Thapa. *Linear Programming 1: Introduction*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag New York, 1997.
- [37] Patricio Antonio Wolff Rojas. Modelos de programación matemática para asignación de pabellones quirúrgicos en hospitales públicos. Master's thesis, Universidad de Chile, Santiago, Chile, 2011.
- [38] J. E. Pezoa. *Theory of Resource Allocation for Robust Distributed Computing*. PhD thesis, University of New Mexico, Albuquerque, New Mexico, USA, 2010.
- [39] Singiresu S. Rao. *Engineering Optimization: Theory and Practice, Fourth Edition*. John Wiley & Sons, Inc., New Jersey, USA, 2009.
- [40] M. Tounsi. A heuristic-based technique for university resource allocation problems. In *GCC Conference (GCC), 2006 IEEE*, pages 1–6, March 2006.
- [41] J. Youn, I. Park, and M. Sunwoo. Heuristic resource allocation and scheduling method for distributed automotive control systems. *International Journal of Automotive Technology*, 14(4):611–624, 2013.

- [42] S. S. Fatima and M. Wooldridge. Adaptive task resources allocation in multi-agent systems. In *Proceedings of the Fifth International Conference on Autonomous Agents*, AGENTS '01, pages 537–544, 2001.
- [43] Yang Liu and Y. Mohamed. Multi-agent resource allocation (mara) for modeling construction processes. In *Simulation Conference, 2008. WSC 2008. Winter*, pages 2361–2369, Dec 2008.
- [44] G.M.P. O'Hare, R. Collier, M. Dragone, M.J. O'Grady, C. Muldoon, and A. Montoya. *Agent Based Approaches to Ambient Intelligence*, chapter Embedding Agents within Ambient Intelligent Applications. IOS Press, 2012.
- [45] A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning. Saving portable computer battery power through remote process execution. *SIGMOBILE Mob. Comput. Commun. Rev.*, 2(1):19–26, jan 1998.
- [46] M. Othman and S. Hailes. Power conservation strategy for mobile computers using load sharing. *SIGMOBILE Mob. Comput. Commun. Rev.*, 2(1):44–51, jan 1998.
- [47] U. Kremer, J. Hicks, and J. Rehg. A compilation framework for power and energy management on mobile computers. In *Proc. 14th Int. Conf. on Languages and Compilers for Parallel Computing*, pages 115–131, 2003.
- [48] C. A. Floudas and X. Lin. Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications. *Annals of Operations Research*, 139(1):131–162, 2005.
- [49] S. V. Hashemian and F. Mavaddat. A graph-based approach to web services composition. In *The 2005 Symposium on Applications and the Internet*, pages 183–189, 2005.
- [50] C. Cadere, D. Barth, and S. Vial. Virtualization and allocation of network service resources using graph embedding. In *Computer and Information Sciences, 2008. ISCIS '08. 23rd International Symposium on*, pages 1–6, 2008.
- [51] B. Wu, C. H. Chi, and Z. Chen. Resource allocation based on workflow for enhancing the performance of composite service. In *IEEE International Conference on Services Computing (SCC 2007)*, pages 552–559, 2007.
- [52] Apple. [Online] <http://www.apple.com/>, 2016. Último acceso: Oct. 2016.
- [53] M. Pastrana and D. A. Ovalle. Método de comparación utilizando métricas de calidad para protocolos de negociación electrónica en sistemas multi-agente. *DYNA*, 75(154), 2008.

- [54] Raj Jain, Dah-Ming Chiu, and William R Hawe. *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*, volume 38. Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984.

