



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

**Departamento de Computación**

Disponibilidad de recursos compartidos en un  
ambiente colaborativo con carácter ubicuo

Tesis que presenta

**Elvia Kimberly García García**

para obtener el Grado de

**Maestro en Ciencias  
en Computación**

Directora de Tesis

**Dra. Sonia Guadalupe Mendoza Chapa**

México, D.F.

Enero 2009



# Resumen

La mayoría de los trabajos de investigación desarrollados en el campo del cómputo ubicuo permanecen en el dominio de los sistemas mono-usuario, los cuales hacen suposiciones como: “nadie interfiere”, “nadie observa” y “nadie apura”. Además, estos sistemas ignoran las contribuciones de terceras personas y no fomentan los consensos.

Esta tesis propone un sistema para administrar la disponibilidad de recursos distribuidos, inmersos en un ambiente cooperativo con carácter ubicuo. La disponibilidad de un recurso se determina tomando en cuenta los siguientes parámetros: sus restricciones de uso, permisos y dependencias con otros recursos en términos de pertenencia, presencia, localización e incluso disponibilidad. Todos estos parámetros no han sido considerados de manera conjunta en trabajos precedentes.

El sistema propuesto permite que los colaboradores: 1) publiquen los recursos que desean compartir con sus colegas y 2) se suscriban cuando necesiten utilizar algún recurso. El proceso de publicación de un recurso consiste en: 1) describirlo en términos de sus capacidades o características técnicas, 2) definir sus restricciones de uso o acceso y 3) asignar permisos a los colaboradores con quienes se desea compartirlo. La definición de restricciones de uso y la asignación de permisos ayudan a resguardar a los recursos de abusos. Por otra parte, el proceso de suscripción estriba en describir el recurso que se requiere utilizar en términos de sus capacidades o características técnicas. El sistema utilizará esta información para buscar algún recurso que técnicamente satisfaga las necesidades del colaborador. Además, en caso de encontrar el recurso adecuado, el sistema verificará su disponibilidad con el fin de ofrecer respuestas que sean realmente útiles.

Para mantener actualizada la información que permite conocer la disponibilidad de un recurso, el sistema propuesto ofrece: 1) aplicaciones de usuario y 2) un sistema de reconocimiento de caras. Las aplicaciones de usuario hacen posible que los colaboradores modifiquen en cualquier momento el estado de sus recursos compartidos, con el fin de mantenerlos protegidos. Por su parte, el sistema de reconocimiento de caras es capaz de identificar y localizar automáticamente a los colaboradores en su ambiente de trabajo. Este sistema de reconocimiento también ayuda a permitir o denegar el acceso a la información de acuerdo al contexto.

**Palabras clave:** disponibilidad de recursos distribuidos, administración de recursos, percepción, ambientes cooperativos ubicuos, reconocimiento de caras, localización automática de usuarios.



# Abstract

Most research works in ubiquitous computing remain developed in the domain of monouser systems, which make assumptions such as: “nobody interferes”, “nobody observes” and “nobody hurries up”. In addition, these systems ignore third-part contributions and do not encourage consensus achievement.

This thesis work proposes a system for managing availability of distributed shared resources in ubiquitous cooperative environments. A resource availability is determined according to several parameters: usage restrictions, permissions, and dependencies with other resources in terms of ownership, presence, location, and even availability. All this parameters have not been considered as a whole in previous works.

More specifically, the proposed system allows collaborators: 1) to publish resources that are intended to be shared with their colleagues and 2) to subscribe to resources depending on their interest in accessing or using them. In the publish process collaborators can: 1) describe a resource in terms of its capabilities or technical characteristics, 2) define usage or access restrictions for the resource and 3) assign permissions to collaborators in order to share the resource with them. A collaborator can protect his resources from abuse defining usage restrictions and assigning permissions. In the subscription process, a collaborator that is interested in using a resource, describes his resource needs in terms of its capabilities or its technical characteristics. The system uses this information to look for resources that fill the collaborator’s requirements and also that are available in that moment for that specific collaborator.

To keep updated the information that allows to know the availability of a resource, the proposed system offers users applications and a human face recognition system. The users applications allow collaborators to modify the state of their shared resources at anytime in order to protect them. On the other hand, the human face recognition system is capable of identifying collaborators and to locate them within the cooperative environment. This human face recognition system also helps to permit or deny access to context-aware information.

**Keywords:** Availability of distributed resources, resource management, perception, ubiquitous cooperative environments, human face recognition, automatic user localization.



*A mis queridos asesores mi más profundo agradecimiento y admiración ...*

*Dra. Sonia Mendoza:*

*Gracias por todo el tiempo, dedicación, esfuerzo y sacrificio que ha implicado dirigir este trabajo. Le agradezco infinitamente que haya compartido sus experiencias conmigo, me han hecho admirarla cada día más. Le doy las gracias por preocuparse por mí, por entenderme no sólo como alumna, sino también como persona (sé que no ha sido sencillo). Pero sobre todo, le doy las gracias por confiar en mí, porque sus palabras siempre me transmiten seguridad.*

*Dr. Dominique Decouchant:*

*Le agradezco todos los momentos felices que me ha regalado, porque con su buen humor es imposible dejar de sonreír. Gracias por interesarse en mí, por dejarme ver la maravillosa persona que es usted, por sus buenos consejos y por las experiencias inolvidables que he vivido.*

*Dr. Gustavo Olague:*

*Jamás olvidaré la primera vez que estuve lejos de mi familia, no lo haré porque lo conocí, porque las pláticas que tuve con usted han sido de las más enriquecedoras de mi vida. Le agradezco cada una de sus palabras, ya que me han ayudado a ver las cosas de forma distinta.*

*Le doy gracias a Dios por haberme rodeado de personas tan admirables, trabajadoras e íntegras como ustedes, le agradezco porque para mi ustedes no sólo son profesores, ustedes son mis profesores.*

*Gracias al Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional y al Consejo Nacional de Ciencia y Tecnología por permitirme el logro de muchos sueños y metas.*





*Gracias a mi maravillosa familia por estar siempre junto a mi ...*

*Mami:*

*Un simple gracias no demuestra lo infinitamente agradecida que estoy contigo, porque tu confianza, amor y seguridad me han ayudado a ser yo. Gracias por el ejemplo de lucha y trabajo que siempre has sido para mí. Te agradezco todas las noches de pláticas que me han hecho sentir querida e importante ¡Te amo!*

*Papi:*

*Te agradezco mucho haberme enseñado a cumplir mis sueños. Muchas gracias por tu atención, amor y confianza. Gracias por darme seguridad al tomar mis propias decisiones. Gracias por ser parte de cada etapa de mi vida, yo se que no ha sido fácil, pero eres el mejor ¡Te amo!*

*Hermanita:*

*Te agradezco porque siempre has estado para escucharme (desde que eramos jóvenes), para darme un abrazo y para hacerme reír, sobre todo gracias por darme todo tu amor ¡Te amo!*

*Hermanito:*

*Gracias por tu cariño, porque las cosas más cotidianas y simples se vuelven divertidas junto a ti. Gracias por darme todo tu apoyo, por levantarme el ánimo y por siempre preocuparte por mí. Gracias porque siempre me has demostrado que somos hermanos de verdad ¡Te amo!*

*I'll always be there for you ...*

*Gracias por ser lo más importante en mi vida.  
“Los amo”, sin ustedes nada de esto sería posible.*



# Índice general

Índice de figuras	xii
Índice de tablas	xiv
<b>1 Introducción</b>	<b>1</b>
1.1 Contexto de investigación	1
1.2 Planteamiento del problema	3
1.3 Objetivos del proyecto	6
1.4 Organización del documento	6
<b>2 Trabajo relacionado y conceptos relevantes</b>	<b>9</b>
2.1 Conceptos relevantes	9
2.1.1 “Observabilidad” publicada y filtrada	10
2.1.2 Modelo de publicación/subscripción	11
2.2 Sistemas colaborativos	13
2.2.1 ProxyLady	13
2.2.2 CHIS	15
2.2.3 ABC	17
2.3 Protocolos de descubrimiento de servicios	18
2.3.1 <i>Service Location Protocol</i>	19
2.3.2 <i>Ninja Service Discovery Service</i>	20
2.3.3 Jini	21
2.4 Discusión sobre el trabajo relacionado	21
<b>3 Análisis y diseño del sistema RAMS</b>	<b>25</b>
3.1 Escenario	25
3.2 Tipos de recursos	28
3.3 Esquema de aplicación	29
3.3.1 Agentes	30
3.3.2 Eventos	31
3.3.3 Componentes del sistema RAMS	32
3.4 Broker	35
3.4.1 Módulos de publicación y de subscripción	35

3.4.2	Filtro de información . . . . .	39
3.5	Motor de inferencia . . . . .	40
3.6	Sensores . . . . .	43
3.6.1	Sistema de reconocimiento de caras . . . . .	43
3.6.2	Sistema de localización del recurso más cercano . . . . .	44
3.7	Sistema de notificación . . . . .	46
3.7.1	Información de estado de los recursos físicos . . . . .	46
3.7.2	Información de estado de los recursos humanos . . . . .	48
3.7.3	Información de estado de los recursos informáticos . . . . .	50
3.8	Esquema de distribución . . . . .	51
<b>4</b>	<b>Implementación del sistema RAMS</b>	<b>55</b>
4.1	Descripción de recursos . . . . .	55
4.1.1	Herramienta para describir recursos . . . . .	56
4.1.2	Descripción de recursos físicos . . . . .	56
4.1.3	Descripción de recursos informáticos . . . . .	60
4.1.4	Descripción de recursos humanos . . . . .	60
4.2	Motor de inferencia . . . . .	61
4.3	Sistema de reconocimiento de caras . . . . .	63
4.3.1	Biblioteca <i>OpenCV</i> . . . . .	64
4.3.2	Análisis de Componentes Principales . . . . .	66
4.3.3	<i>Support Vector Machines</i> . . . . .	68
4.3.4	Experimento . . . . .	69
<b>5</b>	<b>Conclusiones y trabajo futuro</b>	<b>73</b>
5.1	Recapitulación del planteamiento del problema . . . . .	73
5.2	Conclusiones . . . . .	74
5.3	Trabajo futuro . . . . .	75
	<b>Publicación del autor</b>	<b>77</b>
	<b>Referencias</b>	<b>79</b>

# Índice de figuras

1.1	Contexto de investigación de la presente tesis . . . . .	2
1.2	Caracterización de la percepción . . . . .	3
1.3	El sistema de teleconferencia <i>VideoWindow</i> . . . . .	4
1.4	Portholes de Xerox EuroPARC . . . . .	5
1.5	Organización del documento . . . . .	7
2.1	Principio de la “observabilidad” publicada y filtrada . . . . .	11
2.2	Modelo de publicación/subscripción . . . . .	12
2.3	Ventanas del sistema <i>Proxy Lady</i> . . . . .	14
2.4	Uso de un despliegue público por medio de telepunteros y un PDA . . . . .	16
2.5	Uso compartido de una aplicación . . . . .	16
2.6	Un médico comparte una actividad síncrona . . . . .	17
2.7	Una enfermera utiliza la funcionalidad de <i>roaming</i> . . . . .	18
3.1	Ambiente colaborativo . . . . .	26
3.2	Casos de uso para un agente productor . . . . .	32
3.3	Casos de uso para un agente consumidor . . . . .	33
3.4	Arquitectura de <i>software</i> del sistema RAMS . . . . .	34
3.5	Funciones de los módulos de publicación y de subscripción . . . . .	35
3.6	Diagrama de clases de las entidades del sistema RAMS . . . . .	42
3.7	Fases de aprendizaje y prueba del sistema de reconocimiento de caras . . . . .	44
3.8	Mapa de la planta alta del Departamento de Computación . . . . .	45
3.9	Diagrama de estado de un recurso físico . . . . .	46
3.10	Diagrama de estado de un recurso humano . . . . .	49
3.11	Diagrama de estado de un recurso informático . . . . .	50
3.12	Arquitectura de distribución del sistema RAMS . . . . .	52
4.1	Estructura jerárquica de un documento XML DOM . . . . .	57
4.2	Descripción en XML de una impresora . . . . .	58
4.3	Descripción en XML de un escáner . . . . .	59
4.4	Descripción en XML de una sala de juntas . . . . .	59
4.5	Descripción en XML de un video . . . . .	60
4.6	Descripción en XML de un colaborador . . . . .	61
4.7	Procesos que se realizan en el motor de inferencia . . . . .	63

4.8	Construcción del conjunto de entrenamiento . . . . .	65
4.9	Detección de una cara humana . . . . .	66
4.10	Detección simultánea de varias caras humanas . . . . .	66
4.11	Análisis de Componentes Principales . . . . .	68
4.12	Hiperplano construido por SVM . . . . .	69
4.13	Colaboradores que reconoce el sistema para este experimento . . . . .	70
4.14	Imágenes de prueba . . . . .	70
4.15	Resultados que arroja el sistema de reconocimeinto de caras . . . . .	71

# Lista de Tablas

3.1	Descripción de un recurso humano . . . . .	36
3.2	Descripción de un recurso físico . . . . .	37
3.3	Descripción de un recurso informático . . . . .	38





# Capítulo 1

## Introducción

### 1.1 Contexto de investigación

El concepto de ubicuidad se refiere en general a la presencia de una entidad en todas partes, pero en ciencias computacionales adquiere además un carácter invisible. Así, el campo de investigación del Cómputo Ubicuo tiene como objetivo incrementar el uso de sistemas computacionales a través del entorno físico, haciéndolos disponibles pero también invisibles al usuario [Weiser, 1993]. Este campo de investigación ha sido posicionado como el tercer paradigma de la computación, el cual hoy en día ya cuenta con múltiples adeptos, tanto en el ámbito científico como en el industrial, quienes procuran su desarrollo. Por lo tanto, se pretende diseñar e implementar sistemas computacionales inteligentes que se adapten al usuario y que se utilicen de forma intuitiva. La integración de dispositivos de cómputo y de comunicación en el entorno físico requiere habilitar sus capacidades, así como las de la información que albergan, en todo momento y en cualquier lugar.

Este trabajo de investigación también se inscribe en el campo multidisciplinario denominado Trabajo Cooperativo Asistido por Computadora (TCAC), el cual estudia tanto los aspectos sociales de las actividades individuales y colectivas, como los aspectos tecnológicos de la información y de las comunicaciones, con el fin de soportar la colaboración entre personas. Respecto a este carácter multidisciplinario, esta tesis toma la perspectiva computacional del TCAC (i.e., desarrollo de sistemas colaborativos) con el fin de diseñar e implementar un sistema que administre la percepción de la información de conciencia de grupo. Esta información se refiere al estado (e.g., presencia, ubicación, acciones y capacidades) de los colaboradores y de sus recursos compartidos. Dourish y Bellotti definen “conciencia de grupo” como “el entendimiento de las actividades de otros, que nos proporciona un contexto en el cual encuadrar nuestras propias actividades” [Dourish y Bellotti, 1992]. La información de conciencia de grupo está presente de manera natural cuando los colaboradores interactúan cara a cara, pero se pierde cuando trabajan a distancia. En particular, el tipo de información relevante para este estudio concierne la disponibilidad de recursos humanos, físicos e informáticos en un entorno de colaboración ubicua (ver Figura 1.1).

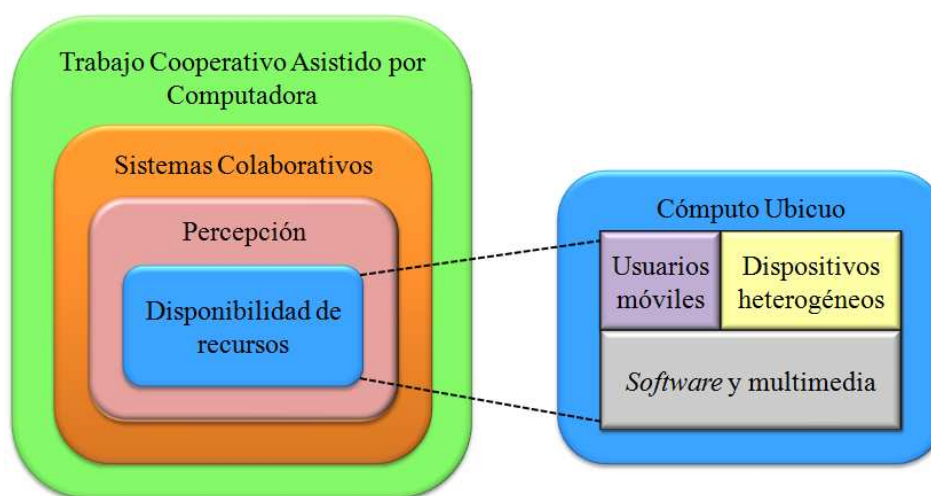


Figura 1.1: Contexto de investigación de la presente tesis

Según las dimensiones de espacio-tiempo propuestas por Ellis et al. [Ellis et al., 1991] para caracterizar la interacción humana asistida por computadora, la llamada interacción colocalizada (mismo lugar - mismo tiempo) parece constituir el punto de partida para explorar el concepto de ubicuidad en sistemas colaborativos. Miller propone un sistema de control remoto que permite a un grupo de individuos compartir un espacio de despliegue, así como la aplicación que se está ejecutando en ese espacio [Miller, 2003]. Así mismo, Markarian et al. desarrollaron un sistema colaborativo móvil que pretende soportar: 1) la colaboración colocalizada, 2) el uso compartido de aplicaciones con base en la proximidad y 3) el control remoto de dispositivos heterogéneos [Markarian et al., 2006].

En general, los trabajos propuestos en el campo del TCAC sobre ubicuidad ponen en evidencia la necesidad de los integrantes de un equipo de trabajo de acceder a recursos informáticos (e.g., *software* y *multimedia*) desde cualquier lugar. Con el fin de ilustrar este requerimiento, considere el siguiente escenario: “un grupo de trabajo se reúne para tratar algún tema en particular. Durante la reunión, uno de los colaboradores recuerda que posee información relevante que puede compartir con sus colegas. Sin embargo, dicha información sólo se encuentra en su computadora personal de escritorio. En consecuencia, el nulo carácter móvil de este dispositivo de cómputo lo obliga a trasladarse a su oficina para imprimir la información o para enviarla al despliegue de la sala de juntas, donde se encuentra el grupo de trabajo”.

Resulta evidente que esta forma de compartir información entorpece la interacción entre los miembros del grupo durante la reunión. Tales problemas han motivado el estudio de la potencial utilidad de los dispositivos móviles en sistemas colaborativos para trasladar de forma transparente recursos informáticos entre dispositivos de cómputo heterogéneos. Otra de las motivaciones de hacer converger los campos de investigación del Cómputo Ubicuo y del TCAC, en la presente tesis, es que el primero hace avanzar la comprensión de la teoría y práctica del segundo, mientras que el TCAC presenta tópicos de investigación que lanzan retos al Cómputo Ubicuo, tópicos que posiblemente no serían explorados de

otra manera.

El ámbito de estudio de esta tesis es el entorno de una organización, cuyos recursos humanos son colaboradores potenciales que pueden estar localizados en varios edificios. Múltiples dispositivos heterogéneos (e.g., computadoras personales fijas y portátiles, servidores, impresoras, *plotters*, pizarrones interactivos y organizadores personales) están distribuidos en lugares privados (e.g., oficinas) y en lugares públicos (e.g., salas, pasillos, corredores o aulas) de la organización. Ya que resulta imposible que una persona disponga en su oficina de cada uno de los recursos físicos existentes en la organización, el uso compartido se vuelve una necesidad evidente. De manera similar, técnicamente no es factible duplicar todos los recursos informáticos de los colaboradores en cada uno de los dispositivos de almacenamiento.

Por lo tanto, para soportar el uso compartido de los recursos inmersos en un entorno con carácter ubicuo (i.e., potencialmente accesibles desde cualquier lugar y en todo momento) es necesario administrar la disponibilidad de los colaboradores y de sus recursos.

## 1.2 Planteamiento del problema

El concepto de percepción se asocia generalmente a los sistemas colaborativos en los que no se presenta la interacción colocalizada. Debido a que la distancia física disminuye drásticamente la comunicación entre colaboradores, se vuelve necesario crear mecanismos que provean información sobre la actividad del grupo [McDaniel, 1996]. La percepción se refiere a la información sobre la información, así que se han distinguido dos tendencias de clasificación: la primera corresponde al tipo de objeto de información y la segunda concierne a la forma de capturar esta información (ver Figura 1.2).

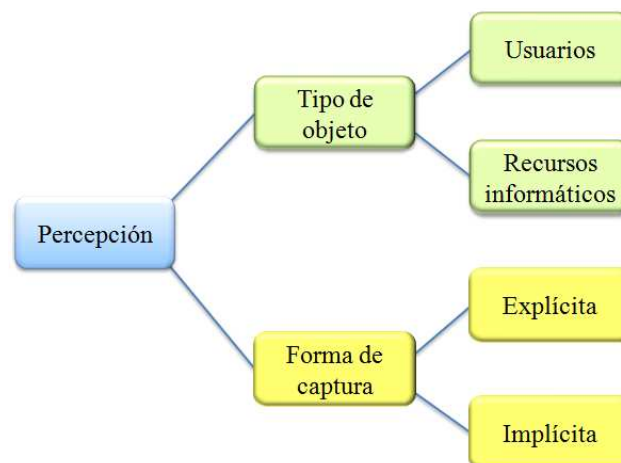


Figura 1.2: Caracterización de la percepción

En cuanto al objeto de información, se pueden distinguir dos clases: la **percepción de usuarios** y la **percepción de recursos informáticos**. La **percepción de usuarios** provee información sobre los miembros de grupo, e.g. presencia y actividades en una

sesión, mientras que la **percepción de recursos informáticos** suministra información referente a los cambios efectuados sobre los recursos informáticos. Respecto a la forma de capturar la información, la percepción puede ser **implícita** o **explícita**. La información se puede percibir implícitamente mediante cámaras de video, altavoces, chapas activas, sensores infrarrojos, teclados, ratones o cualquier método que no requiera la introducción explícita de información por parte del usuario. Por el contrario, la percepción explícita es proveída por los usuarios en forma de calendarios, horarios o letreros como “no molestar”.

Por una parte, esta caracterización de la percepción, propuesta por Susan E. McDaniel en 1996, no toma en consideración (por obvias razones) un tipo de objeto relevante en el cómputo ubicuo: los recursos físicos, e.g., Dahlberg y Sanneblad introducen un estilo de interacción mediante dispositivos móviles que combina parámetros referentes al estado y a la proximidad de los dispositivos y de las personas [Dahlberg y Sanneblad, 2000]. Por otra parte, esta caracterización de la percepción pone en evidencia un requerimiento esencial en el diseño de todo sistema de cómputo ubicuo: la percepción implícita de la información. Si bien, este campo de investigación trata de evitar la percepción explícita, i.e. la introducción de información por parte del usuario, este mecanismo de captura es necesario en sistemas colaborativos por cuestiones de privacidad e intrusión.

### Violación de privacidad e intrusión

La mayoría de los trabajos de investigación que abordan la ubicuidad forma parte del dominio de los sistemas monousuario, los cuales hacen suposiciones tales como “nadie interfiere”, “nadie observa” y “nadie apura”, entre otras. Además, ignoran aportes de terceras personas y no estimulan la formación de consensos. Uno de los principales requerimientos que se ha puesto en evidencia en el diseño de estos sistemas mono-usuario es la necesidad de rastrear la localización de los dispositivos móviles portados por los usuarios, e.g., sistemas como CHIS (*Context Aware Hospital Information System*) [Markarian et al., 2006] han mostrado que la satisfacción de este requerimiento es indispensable para manejar la información de un hospital. Sin embargo, esta capacidad de rastreo podría ocasionar problemas de privacidad e intrusión en el dominio de los sistemas colaborativos.



Figura 1.3: El sistema de teleconferencia *VideoWindow*

Los sistemas de espacios de medios (*mediaspaces*), desarrollados a principios de la década de los noventas, permitieron comprender algunos efectos tecnológicos y sociales de la provisión de conexiones de video y audio para percibir a las personas que se encuentra en otros lugares y eventualmente entablar una conversación. Algunos de estos sistemas, e.g., VideoWindow [Abel, 1990] y Portholes [Dourish y Bly, 1992], utilizaban cámaras para captar no sólo áreas comunes (e.g., cafetería), sino también áreas privadas (e.g., oficinas). En aquellos tiempos, algunas de las preocupaciones principales de los diseñadores de estos sistemas eran las necesidades excesivas de tecnología, e.g., vídeo, audio y ancho de banda (ver figuras 1.4 y 1.3). Si bien, hoy en día, estos problemas de carácter tecnológico están relativamente resueltos, las violaciones de privacidad e intrusión son temas de investigación abiertos.

Estos problemas de carácter social se deben a que los diseñadores de sistemas colaborativos carecen de una metodología que determine con precisión la información necesaria para facilitar el contacto entre las personas [Greenberg y Johnson, 1997]. En consecuencia, el diseño inapropiado de un sistema colaborativo desembocar en la provisión de: 1) información excesiva, la cual puede generar problemas sociales (e.g., privacidad e intrusión) y problemas tecnológicos (e.g., escalabilidad y ancho de banda) o 2) información escasa, la cual puede ocasionar problemas de contactos inapropiados o pérdida de oportunidades. Esta es la razón principal por la cual los diseñadores de sistemas colaborativos tienden a confiar principalmente en la perspicacia y en la creatividad.



Figura 1.4: Portholes de Xerox EuroPARC

Además, el control del uso compartido de los recursos y la privacidad de la información asociada pueden cambiar, debido a que las necesidades de un grupo pueden variar con el tiempo [Berard y Coutaz, 1997]. Para hacer frente a esta dinámica, la pregunta obligada es ¿quién debería encargarse de estas tareas?, ¿el sistema?, ¿el propietario de los recursos privados?, ¿el grupo? o ¿un colaborador en particular?.

El presente trabajo de investigación toma estas preguntas como punto de partida para

definir los parámetros que influirán en la administración de la percepción del estado tanto de los colaboradores como de sus recursos físicos e informáticos compartidos. Todos estos recursos son indispensables en un entorno de colaboración ubicua, pero sus capacidades, sus limitaciones de uso y sus relaciones con otros recursos no han sido consideradas de manera comprehensiva en la determinación de su grado de disponibilidad.

## 1.3 Objetivos del proyecto

### Objetivo general

Diseñar la arquitectura de un sistema que, por un lado, permita a los miembros de una organización compartir sus recursos físicos e informáticos y que, por otro lado, proporcione a cada colaborador información sobre el estado, en términos de presencia, ubicación y disponibilidad, de un conjunto de recursos accesibles que puede satisfacer sus requerimientos técnicos y de uso.

### Objetivos específicos

1. Diseñar e implementar los módulos que permitirán a los colaboradores publicar y suscribirse a recursos compartidos. Estos módulos ofrecerán funciones para asistir al dueño de un recurso: a) y también al solicitante, en la descripción de las características técnicas o capacidades del recurso; b) en el otorgamiento o la denegación de permisos a los usuarios potenciales del recurso, y c) en la definición de restricciones de uso del recurso.
2. Definir un mecanismo para determinar el grado de disponibilidad de un recurso en base a los siguientes parámetros: a) sus características técnicas, b) sus limitaciones de uso definidas por el dueño, y c) sus dependencias con otros recursos en términos de pertenencia, presencia, localización, permisos e incluso disponibilidad.
3. Definir un mecanismo para calcular el recurso más próximo al solicitante a partir del conjunto de recursos más adecuados para satisfacer sus necesidades.
4. Adaptar un sistema de reconocimiento de nebulosas planetarias (proporcionado por el Dr. Gustavo Olague del Departamento de Computación del CICESE) para que identifique personas por medio del reconocimiento de sus características faciales.

## 1.4 Organización del documento

Este documento está estructurado en cinco capítulos (ver Figura 1.5). Después de haber presentado el contexto de investigación donde se ubica la presente tesis y de haber planteado el problema que se pretende resolver, así como los objetivos que se persiguen, se expone el estado del arte en el capítulo 2. Particularmente, se analizan las principales

características, ventajas y desventajas de dos clases de sistemas propuestos en los campos de investigación del Trabajo Cooperativo Asistido por Computadora y del Cómputo Ubicuo: a) los protocolos de descubrimiento de servicios, los cuales ofrecen soporte para el uso compartido de aplicaciones y, en algunos casos, de dispositivos y b) las aplicaciones conscientes de contexto, las cuales han sido utilizadas en entornos reales (e.g., hospitales e instituciones) para proveer información y servicios a los usuarios según su ámbito actual (e.g., localización).

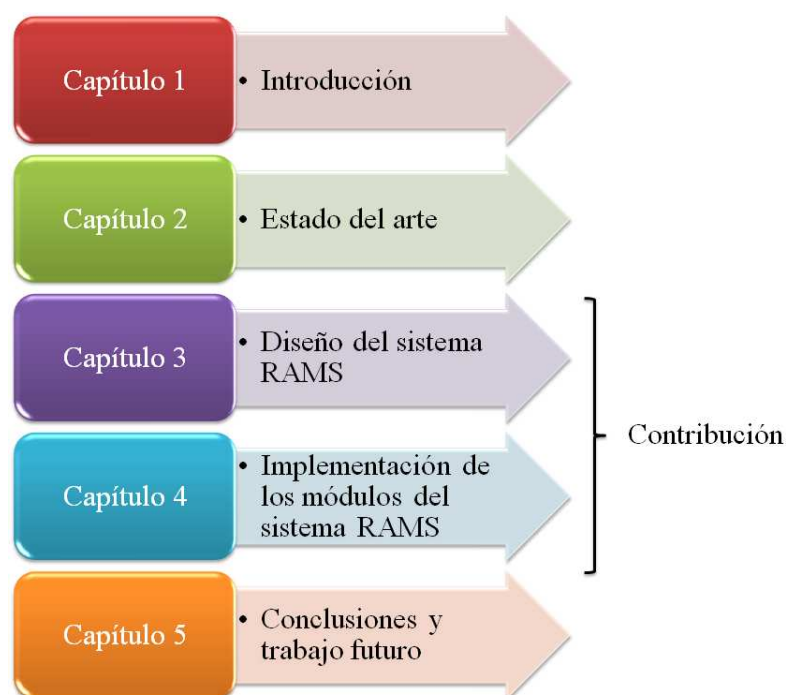


Figura 1.5: Organización del documento

En los Capítulos 3 y 4 se describe la contribución de este trabajo de investigación. Específicamente, en el Capítulo 3, se detalla el diseño del sistema RAMS (*Resource Availability Management System*), el cual promueve el uso compartido de recursos físicos e informáticos entre los miembros de una organización. En primer lugar, se presenta un escenario que pone en evidencia los beneficios que el sistema propuesto ofrecería a sus potenciales usuarios. Posteriormente, se describen los módulos que componen al sistema RAMS: a) el **broker** ofrece funciones a los usuarios para facilitar la descripción de recursos, ya sea para fines de registro o búsqueda, b) el **motor de inferencia** determina la disponibilidad actual de cada recurso accesible a un usuario específico, c) los **sensores** para el reconocimiento de caras humanas y la localización del recurso más cercano se ocupan de alimentar al motor de inferencia con información actual sobre el estado de los recursos que están registrados en el sistema RAMS y d) el **sistema de notificación** da a conocer, a los usuarios interesados, las respuestas provenientes del motor de inferencia. Finalmente, se describe la arquitectura de distribución del sistema RAMS.

En el Capítulo 4, se detalla la implementación de algunos módulos del sistema RAMS. En particular, se presentan las estructuras de datos que utiliza el módulo de publicación/subscripción para permitir a los colaboradores: a) describir los recursos que compartirá con otros colaboradores y b) expresar sus requerimientos con la intención de encontrar un recurso que los satisfaga. Adicionalmente, se detallan las reglas que sigue el motor de inferencia para seleccionar un conjunto de recursos adecuados y determinar su disponibilidad. Por último, se describe la implementación de un sistema de reconocimiento de caras humanas que se realizó en conjunto con el Dr. Olague del Departamento de Computación de CICESE.

Finalmente, en el Capítulo 5, se presentan las conclusiones de este trabajo de investigación así como algunas posibles prolongaciones a futuro.



## Capítulo 2

# Trabajo relacionado y conceptos relevantes

En este capítulo se describen dos conceptos importantes que se explotan en el desarrollo de esta tesis (sección 2.1). En primer lugar, se detalla el concepto de “observabilidad”, el cual permite obtener información sobre las actividades personales de los colaboradores pero, al mismo tiempo, ayuda a cuidar su privacidad. Posteriormente, se explica el modelo de publicación/subscripción tradicional. Aún cuando algunas de las características de este modelo están enfocadas a sistemas mono-usuario, puede ser utilizado como base para construir sistemas colaborativos. Adicionalmente, en este capítulo se presentan algunos de los trabajos más importantes que están relacionados con el sistema que se propone en esta tesis. Dichos trabajos se dividen en dos tipos: sistemas colaborativos y protocolos de descubrimiento de servicios. Los sistemas colaborativos que han sido analizados además de apoyar la colaboración entre usuarios, también están inmersos en el área del cómputo ubicuo, ya que proporcionan mecanismos para acercar información relevante a los usuarios (sección 2.2). Por otra parte, los protocolos de descubrimiento de servicios, que han sido estudiados, se seleccionaron porque son representantes significativos del ámbito académico o empresarial en el que fueron desarrollados (sección 2.3). El objetivo de los protocolos de descubrimiento es promover el uso de servicios (*hardware* o *software*) que están disponibles en un sistema en red. Algunos de estos protocolos no se enfocan en el cómputo ubicuo, pero es posible explotarlos en esta área. Por último, se presenta una discusión sobre las características de los sistemas colaborativos y de los protocolos de descubrimiento estudiados (sección 2.4).

### 2.1 Conceptos relevantes

En esta sección se describen dos conceptos importantes que se debe tener en mente, ya que han sido utilizados en el desarrollo de esta tesis. En la sección 2.1.1 se describe el concepto

de “observabilidad” <sup>1</sup> (*observability*) publicada y filtrada. Este concepto es imponente en este trabajo de investigación, ya que propone una forma de controlar la información que se obtiene de un usuario. A diferencia de los sistemas distribuidos, los sistemas colaborativos requieren información sobre las actividades que realizan los usuarios, pero deben cuidar la forma y la cantidad de información que obtienen. Demasiada información puede ocasionar que el colaborador se sienta invadido, mientras que muy poca información puede dificultar la interacción entre los colaboradores. En la sección 2.1.2, se detalla el modelo tradicional de publicación/subscripción. Este modelo propone una forma de interacción que involucra usuarios que publican eventos y usuarios que se subscriben para recibir notificaciones sobre dichos eventos. Aunque el modelo publicación/subscripción ha sido utilizado en muchos áreas, requiere algunas modificaciones para adecuarlo a los requerimientos de los sistemas colaborativos. Este modelo constituye la base para el desarrollo del sistema que se propone en esta tesis.

### 2.1.1 “Observabilidad” publicada y filtrada

La “observabilidad” [Berard y Coutaz, 1997] es la capacidad de un sistema para hacer perceptibles a un usuario las variables de estado internas, pertinentes para la tarea en curso. La pertinencia se estudia y se define a partir del análisis de la tarea. En una actividad de grupo, la “observabilidad” se opone al concepto de protección del espacio privado, el cual se define como un conjunto de variables de estado “personales” (e.g., el hecho de que un usuario X esté en su oficina leyendo su correo electrónico).

El carácter personal de una variable se define a partir del análisis del problema. Obviamente, la “observabilidad” de las variables personales puede ser pertinente para el grupo, pero potencialmente contraria al principio de respeto de la intimidad. Al respecto, es conveniente introducir el concepto de “observabilidad publicada” [Coutaz et al., 1997]: las variables de estado personales de un colaborador sólo pueden ser observables para otros, si el propietario autoriza su publicación. La autorización de publicación puede ser estática o dinámica.

La dualidad autorización/rechazo de la publicación de una variable personal es conveniente para el carácter binario de las computadoras, pero sólo refleja parcialmente la complejidad de las relaciones sociales. Por esta razón, se contempla el concepto de “filtro de publicación” [Coutaz et al., 1997], el cual consiste en asociar un filtro a una variable personal publicable. Como se muestra en la Figura 2.1, una variable personal publicable P es puesta en correspondencia con uno o varios conceptos representativos R, en este caso R y R'. A su vez, R puede ser percibida por uno o varios objetos de presentación. La dependencia entre P y R es mantenida por el filtro. La naturaleza del filtro puede depender de P, del rol del propietario, del rol del observador, de sus relaciones socio-profesionales, etc.

Si se retoma el ejemplo del usuario X, quien se encuentra ocupado leyendo un correo electrónico, entonces su “actividad actual” (leer el correo) es una variable de estado per-

---

<sup>1</sup>El término de “observabilidad” no existe en español ni en inglés, pero nos hemos tomado el atrevimiento de hacer la traducción al español del término en inglés

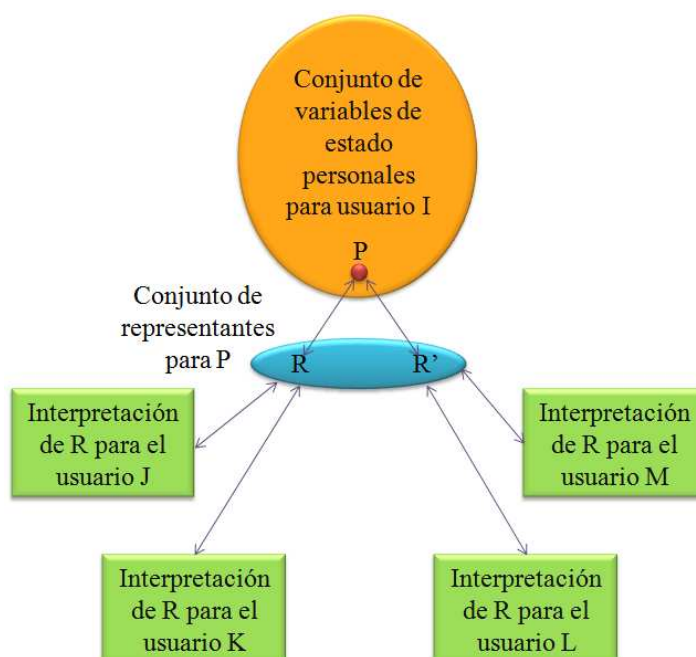


Figura 2.1: Principio de la “observabilidad” publicada y filtrada

sonal. El usuario X elige publicar dicha variable para los miembros de su equipo. Adicionalmente, él decide no compartir con los colegas, que no forman parte de su equipo, información sobre la actividad que está realizando, e.g., el concepto representativo de la actividad actual podría ser un índice de ocupación. En estas condiciones, los colegas que no forman parte del equipo del usuario X saben que él está presente y no disponible, pero ignoran el detalle de sus actividades. Así, un filtro de publicación permite que los usuarios decidan explícitamente qué información desean compartir y con quién.

### 2.1.2 Modelo de publicación/subscripción

El modelo de publicación/subscripción [Eugster et al., 2003] se ha vuelto muy popular ya que las entidades que participan (i.e., publicadores o subscriptores) no están acopladas. Los subscriptores pueden expresar su interés en algún evento, con la intención de recibir notificaciones cuando este se genere. Por otra parte, los publicadores están encargados de generar eventos. Un evento se propaga de forma asíncrona a todos los subscriptores que han registrado su interés en él. La interacción basada en eventos se centra en el completo desapego en tiempo, espacio y sincronización entre publicadores y subscriptores. El modelo de publicación/subscripción proporciona un servicio de notificación de eventos, el cual debe proveer almacenamiento y administración de los subscriptores y entrega eficiente de los mensajes.

La entidad que hace posible la interacción entre publicadores y subscriptores se denomina servicio de eventos. Esta entidad constituye un mediador neutral entre los publicadores

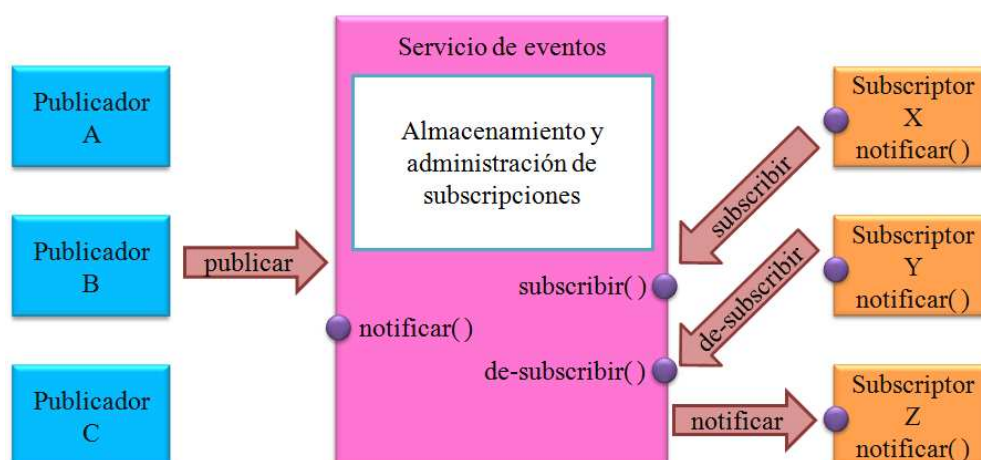


Figura 2.2: Modelo de publicación/subscripción

que actúan como productores de eventos y los suscriptores que actúan como consumidores de eventos. Como se muestra en la Figura 2.2, los suscriptores registran su interés en algún evento mediante la operación `suscribir()`. Dicha operación es proporcionada por el servicio de eventos. No es necesario que los suscriptores conozcan al publicador de los eventos en los que están interesados. A su vez, la información de la suscripción sólo se almacena en el servicio de eventos sin necesidad de enviarla a los publicadores. El método simétrico `de-suscribir()` termina el proceso de suscripción. Para generar un evento, un publicador llama a la operación `publicar()`. Una vez generado el evento, el servicio de eventos se encarga de propagarlo a todos los suscriptores interesados en él. Cada suscriptor será notificado cada vez que se genere un evento que sea de su interés. Los productores pueden especificar la naturaleza de sus futuros eventos por medio de la operación `anunciar()`. Esta información puede ser útil para que: a) el servicio de eventos pueda ajustarse al flujo de eventos esperado y b) los suscriptores estén conscientes cuando un nuevo tipo de información esté disponible. El desacople que el servicio de eventos provee a los productores y suscriptores puede ser dividido en las siguientes dimensiones:

- **Desacoplamiento de espacio:** los publicadores y suscriptores no necesitan conocerse. Los primeros publican eventos por medio del servicio de eventos, mientras que los suscriptores reciben estos eventos del servicio de eventos. Los publicadores usualmente no tienen referencias de los suscriptores, ni tampoco saben la cantidad de suscriptores que participan en la interacción. De forma similar, los suscriptores no tienen ninguna información sobre los publicadores.
- **Desacoplamiento de tiempo:** los publicadores y suscriptores no necesitan participar activamente al mismo tiempo. En particular, los primeros pueden publicar algunos de sus eventos, mientras los suscriptores están desconectados. Por su parte, los suscriptores pueden ser notificados sobre algún evento de su interés, aunque el publicador esté desconectado.

- **Desacoplamiento de sincronización:** la producción y el consumo de eventos carecen de un control de flujo entre los publicadores y suscriptores. El desacoplamiento de la producción y del consumo de los eventos incrementa la escalabilidad, ya que remueve todas las dependencias explícitas entre los participantes.

## 2.2 Sistemas colaborativos

A continuación se describen algunos sistemas colaborativos con carácter ubicuo que han sido desarrollados recientemente. Estos sistemas han sido analizados ya que están orientados a funcionar en organizaciones donde la colaboración entre colegas es vital. En la sección 2.2.1 se analiza el sistema Proxy Lady, el cual ha sido desarrollado para una organización donde existen usuarios que requieren colaborar y compartir recursos informáticos (e.g., correos electrónicos y documentos). Este sistema está basado en el concepto de comunicación oportunista mediante el uso de la tecnología *bluetooth*. Posteriormente, en la sección 2.2.2 se estudia el sistema CHIS, el cual se enfoca al ámbito hospitalario. Este sistema ofrece a sus usuarios información contextual y herramientas que facilitan el uso compartido de algunos recursos físicos públicos. Adicionalmente, el sistema CHIS proporciona funciones que apoyan la colaboración entre colegas que se encuentran en un mismo lugar. Por último, en la sección 2.2.3 se describe el sistema ABC, el cual ofrece apoyo al ambiente hospitalario mediante el concepto de cómputo basado en actividades. Este concepto permite que los usuarios tengan disponible información de su interés en cualquier dispositivo que se encuentre dentro del hospital. Además, el sistema ABC permite que los usuarios compartan actividades, con el objetivo de facilitar la colaboración entre colegas tanto de forma síncrona como asíncrona.

### 2.2.1 ProxyLady

Cuando se inició este proyecto, los dispositivos con tecnología *bluetooth* aún no eran comerciales. Por esta razón, el equipo decidió diseñar su propio radiotransmisor con características similares a las de la tecnología *bluetooth*. Esta tecnología tiene un radio de cobertura muy pequeña, así que sólo se pueden detectar dispositivos que se encuentren muy cerca. Proxy Lady [Dahlberg y Sanneblad, 2000] utiliza tanto la tecnología *bluetooth* como el concepto de notificaciones basadas en la proximidad para acercar información a un usuario móvil. El área de proximidad de un usuario se define como el espacio que lo circunscribe, el cual puede variar de acuerdo al lugar donde el usuario se encuentre, e.g., el área de proximidad de un usuario en un edificio es menor que el área de proximidad de un usuario que camina por la ciudad.

Las notificaciones basadas en la proximidad son mensajes que reciben los usuarios de acuerdo a su ubicación. Estas notificaciones realizan un análisis del estatus de los eventos y de la información contextual. El análisis del estatus de los eventos se refiere a las personas y a los objetos que se encuentran en la proximidad de un usuario móvil. Por su parte, la información contextual se refiere al filtrado y al envío de información de interés

para un colaborador de acuerdo a su ubicación. Cuando se filtra información relevante para el contexto actual de un colaborador, este dispondrá de información sobre lo que está pasando en ese momento.

Proxy Lady es un cliente móvil implementado en un PDA Casio Cassiopeia E-105 con un radiotransmisor que fue creado por el propio equipo. Este sistema fue diseñado para soportar la comunicación oportunista en el trabajo móvil por medio de notificaciones basadas en la proximidad. Este tipo de comunicación requiere que un colaborador manifieste anticipadamente su interés en interactuar con otro usuario, lo cual puede ocurrir cuando ambos colaboradores se encuentran en un área de proximidad [Whittaker et al., 1994].

En el contexto de este sistema, una “unidad de información” puede ser la base para iniciar una comunicación oportunista porque permite guardar y transferir información entre los componentes del sistema. Además, una unidad de información es una asociación entre personas y entidades de información (i.e., correo electrónico o tarea) establecida por los usuarios.

Cuando un colaborador anticipa un requerimiento de interacción con otro colaborador, crea una unidad de información. Así, cuando un “candidato a la interacción” (i.e., el colaborador con el que se desea interactuar) se encuentra en el área de proximidad del interesado, Proxy Lady le notifica mediante una alerta (e.g., un mensaje, un sonido o una luz palpitante) que un colega desea interactuar con él. Posteriormente, le proporciona una entidad de información. Después de esta notificación, los colaboradores pueden iniciar una sesión de comunicación informal.



Figura 2.3: Ventanas del sistema *Proxy Lady*

Este sistema está compuesto por: un servidor llamado *Proxy Server*, un componente de escritorio llamado *Proxy Desktop* y una aplicación para PDA llamada *Proxy Lady Client*, o simplemente *Proxy Lady*. El componente *Proxy Server* administra la lista de usuarios del sistema, la cual contiene un identificador, un nombre y una dirección de correo electrónico. El identificador es un número único que el sistema asigna a los usuarios cuando se registran. Este identificador es utilizado por el componente *Proxy Lady Client* en el proceso de autenticación de usuarios. El componente *Proxy Desktop* intercambia información con el componente *Proxy Server*, con el fin de obtener actualizaciones sobre las nuevas peticiones de interacción. Posteriormente, mediante una carpeta compartida,

el *Proxy Desktop* se sincroniza con el PDA, cuando este se conecta a la computadora. Por su parte, el componente *Proxy Lady Client* contiene una interfaz de componentes que permite desplegar, modificar y notificar candidatos a interacción, e.g., cuando un usuario de *Proxy Lady* desea discutir un correo electrónico con uno de sus colegas, crea una unidad de información mediante una copia del mensaje, la cual coloca en un directorio dedicado. El *software* del *Proxy Lady* de la computadora de escritorio transfiere este mensaje al cliente móvil una vez que se conecta. Posteriormente, cuando el candidato a la interacción se encuentra en el área de proximidad del usuario interesado en comunicarse con él, el sistema le notificará por medio de una alerta. Si el usuario decide aceptar la invitación, entonces recibirá un correo electrónico que contiene la información que será discutida. Alternativamente, el usuario puede activar el modo invisible para esconder su presencia.

A continuación se presentan algunas pantallas del *Proxy Lady Client*. La ventana principal de este cliente móvil enlista los candidatos a interacción (ver 2.3 ref. A), los cuales pueden estar dentro del área de proximidad de un usuario. Cuando este selecciona el nombre de algún candidato a interacción, también puede observar las unidades de información que le ha asociado (ver Figura 2.3 ref. B) o crear una nueva unidad de información (ver Figura 2.3 ref. C). De esta manera, si el usuario se encuentra cerca del candidato a interacción, este último recibe una “alerta de proximidad” (ver Figura 2.3 ref. D).

### 2.2.2 CHIS

El sistema CHIS (*Context aware Hospital Information System*) está orientado a la administración de la información y al soporte de la colaboración en un ambiente hospitalario [Markarian et al., 2006]. El sistema CHIS pretende satisfacer los requerimientos de los usuarios en este ambiente, donde se maneja un gran volumen de información, la cual debe estar actualizada, organizada y disponible en todo momento. Además, los médicos y enfermeras requieren llevar a cabo sesiones de trabajo colaborativo, con el fin de discutir las enfermedades de sus pacientes.

Uno de los principales elementos que componen al sistema CHIS es un mecanismo que permite localizar personas y recursos. La localización de personas permite que los usuarios reciban información dependiente del contexto, e.g., cuando una enfermera se encuentra en el cuarto de un paciente, ella recibirá información sobre los medicamentos que debe suministrarle. Las personas son localizadas por medio de un dispositivo móvil (PDA) que transportan a donde quiera que van dentro del hospital. El sistema de localización utiliza redes neuronales y triangulación de señales WiFi para calcular la posición de dichos dispositivos (PDAs). Los recursos que toma en cuenta el sistema CHIS incluyen dispositivos físicos que se encuentran en áreas públicas, e.g., impresoras y despliegues. Cuando un usuario requiere conocer información de estado de alguno de sus colegas o de un recurso, consulta un mapa que contiene información sobre la presencia y la ubicación de las personas y de los recursos dentro del hospital.

En vez de enviar información dependiente del contexto a una persona determinada, el

sistema CHIS utiliza el concepto de rol. Este concepto resulta útil porque las enfermeras y los médicos a cargo de los pacientes no son siempre los mismos, debido a su horario o disponibilidad. Adicionalmente, este sistema toma en cuenta el tiempo de entrega de la información. Este aspecto es importante porque los usuarios del sistema deben recibir información relevante en el momento adecuado. El sistema de notificación de información dependiente del contexto utiliza la mensajería instantánea, la cual ha sido extendida para soportar la localización de personas, el envío de mensajes retardatarios y la entrega de mensajes basada en roles.



Figura 2.4: Uso de un despliegue público por medio de telepunteros y un PDA

Para apoyar la interacción entre los colaboradores que se encuentran en un mismo lugar al mismo tiempo, el sistema CHIS incluye un control remoto de despliegues públicos mediante un PDA. La intención de este control es lograr que la participación de los colaboradores se realice de forma natural. Así, una persona puede estar frente al despliegue público apuntando con un telepuntero, mientras otra persona puede resaltar alguna información que se encuentra en el despliegue público desde su PDA (ver Figura 2.4)



Figura 2.5: Uso compartido de una aplicación

Mediante el sistema CHIS, también es posible compartir aplicaciones en base a la proximidad. Un colaborador puede compartir la aplicación que está ejecutando, con el objetivo



de entablar una discusión con los colaboradores que se encuentran cerca sobre la información que se está analizando, e.g., un médico que está recorriendo el cuarto de sus pacientes, junto con algunos residentes, desea compartir con estos una aplicación que se está ejecutando en su PDA para discutir una radiografía; una vez que ha compartido la aplicación, el médico dibuja un cuadrado sobre la radiografía para analizar una parte específica (ver Figura 2.5); esta acción se ve reflejada en cada uno de los dispositivos (PDA) de los residentes (ver Figura 2.5 ref. A).

### 2.2.3 ABC

Al igual que el sistema CHIS, el *framework* ABC [Bardram y Christensen, 2007] está enfocado a un ambiente hospitalario. Algunas de las características que motivaron el desarrollo de este *framework* son: 1) los médicos utilizan grandes cantidades de información diversa, 2) las actividades de los trabajadores (e.g., médicos y enfermeras) requieren de la colaboración de sus colegas (e.g., dos médicos intercambian puntos de vista para decidir el mejor tratamiento para un paciente) y 3) los médicos y las enfermeras generalmente no tienen un escritorio asignado, ya que la naturaleza de su trabajo hace que estén en constante movimiento, e.g., un médico que se encuentra frente a la cama de un paciente ingresa notas sobre su estado de salud en su expediente electrónico; posteriormente, se dirige a una reunión con sus colegas radiólogos para discutir el tratamiento apropiado para su paciente; durante esta reunión, el médico consulta los catálogos de medicamentos; más tarde el laboratorio le envía los resultados de un estudio de sangre que se le practicó a su paciente. El ejemplo anterior pone en evidencia la necesidad de un médico de acceder a muchas aplicaciones que le muestren información diversa de forma adecuada. Para hacer más sencilla esta actividad, ABC implementa un concepto llamado “computo basado en actividades”. Este concepto permite que los usuarios organicen varias aplicaciones, datos y recursos en espacios de trabajo denominados “actividades computacionales”. Estas pueden ser recreadas y compartidas, además de permitir que un usuario cambie fácilmente de contexto mediante cualquier dispositivo que tenga a la mano.



Figura 2.6: Un médico comparte una actividad síncrona

Para apoyar la colaboración entre colegas, ABC permite a los usuarios compartir con sus colegas aplicaciones asíncronas o síncronas. Una aplicación asíncrona es útil cuando un usuario requiere observar lo que ha estado haciendo alguno de sus colegas. Por su parte, una aplicación síncrona permite que los colaboradores interactúen simultáneamente, con la intención de intercambiar puntos de vista sobre un caso específico. En la Figura 2.6, se muestra un médico en una actividad síncrona de colaboración.

ABC ofrece la posibilidad de pausar actividades para reiniciarlas más tarde. Cuando un médico requiere iniciar una nueva actividad, puede pausar la que está en curso, crear una actividad nueva y reiniciar la actividad precedente más tarde. Adicionalmente, ABC ofrece una funcionalidad llamada *roaming*, la cual permite que un colaborador pueda reiniciar una actividad en cualquier dispositivo que tenga a la mano. Como se muestra en la Figura 2.7, una enfermera pausa una actividad en el cuarto de un paciente, pero la reiniciará más tarde en el almacén de medicamentos.



Figura 2.7: Una enfermera utiliza la funcionalidad de *roaming*

Este *framework* no está enfocado a dispositivos móviles con capacidades limitadas (e.g., PDA y smartphones) porque, desde el punto de vista de sus diseñadores, este tipo de dispositivos no es adecuado para consultar información médica, ya que el tamaño de sus pantallas es muy pequeño y sus capacidades de procesamiento son bajas.

## 2.3 Protocolos de descubrimiento de servicios

Actualmente, los usuarios ocupan mucho tiempo en buscar servicios y en configurar manualmente dispositivos y programas. Algunas veces, la tarea de configuración requiere habilidades o conocimientos especiales, los cuales no tienen nada que ver con las tareas que se desean realizar. Los protocolos de descubrimiento de servicios permiten minimizar las tareas administrativas e incrementar la usabilidad de los servicios de red, e.g., *hardware* o *software* de cualquier tipo.

En el pasado, muchas organizaciones diseñaron y desarrollaron protocolos de descubrimiento de servicios [Zhu et al., 2005]. Algunos ejemplos de protocolos realizados en la

academia son: el *International Naming System* desarrollado en el Instituto Tecnológico de Massachusetts y el *Ninja Service Discovery Service* de la Universidad de California en Berkeley. En el área comercial, se encuentran *Jini Network Technology* de *Sun Microsystems*, *Universal Plug and Play* (UPnP) de *Microsoft* y *Rendezvous* de *Apple*. Por su parte, algunas organizaciones de normalización también han diseñado protocolos de descubrimiento de servicios, como el *Service Location Protocol* (SLP) de *Internet Engineering Task Force* y *Bluetooth SDP* de *Bluetooth Special Interest Group*. Todos estos protocolos pueden descubrir servicios, pero la mayoría de ellos está orientado a la interacción entre programas de cómputo distribuidos que actúan como clientes o servidores. La integración de la interacción entre personas y servicios es uno de los principales retos del cómputo ubicuo, que nosotros hemos intentado abordar en este trabajo de investigación.

### 2.3.1 *Service Location Protocol*

El protocolo de localización de servicios (*Service Location Protocol*, SLP) [Guttman et al., 1999] comprende un *framework* que toma en cuenta aplicaciones y servicios. Dichas aplicaciones están modeladas como clientes llamados “agentes de usuario” y como servidores denominados “agentes de servicio”. Una tercera entidad llamada “agente de directorio” permite que este protocolo sea escalable. SLP funciona en redes empresariales que comparten servicios, pero no necesariamente es escalable a un área mayor (e.g., descubrimiento de servicios a través de Internet).

Cuando un agente de usuario requiere un servicio, realiza una petición indicando las características que necesita. El agente de usuario recibirá una respuesta, la cual especifica la ubicación de todos los servicios en la red que satisfacen su petición. Una vez que este agente de usuario conoce la ubicación de los servicios útiles, puede ponerse en contacto con ellos. En redes pequeñas, los agentes de usuario pueden realizar peticiones directamente a los agentes de servicio por medio de una solicitud *multicast*. Dichos agentes de servicio envían una respuesta *unicast* a los agentes de usuario en la que detallan su ubicación.

En redes más grandes, se utiliza uno o más “agentes de directorio”, los cuales funcionan como una memoria caché en la que los agentes de servicio se registran para anunciar los servicios que ofrecen. Estos anuncios se refrescan periódicamente para confirmar que dichos servicios aún se encuentran disponibles. Los agentes de usuario envían peticiones *unicast* a un agente de directorio en lugar de enviarla a un agente de servicio.

Los agentes de usuarios y agentes de servicio descubren agentes de directorio de dos formas: 1) envían una petición de servicio *multicast* para encontrar un agente de directorio o 2) el agente de directorio envía un anuncio no solicitado a los agentes de usuario para informarles sobre los servicios disponibles.

Los agentes de servicio están agrupados en ámbitos (*scopes*), los cuales contienen un nombre que identifica a un grupo de servicios, su ubicación, su proximidad y el tipo de servicio. Tanto a los agentes de servicio como a los agentes de directorio se les asigna un ámbito. Un agente de usuario también puede tener un ámbito, sin embargo él solo podrá descubrir los servicios que se encuentren en un grupo particular. Si un agente de usuario es declarado sin ámbito, podrá descubrir todos los servicios sin importar el grupo al que

pertenecen.

SLP proporciona algunas características adicionales que permiten a los agentes de usuario tomar en cuenta no sólo las aplicaciones sino también las personas, quienes podrán seleccionar servicios mediante una interfaz de navegación. Estas características pueden ser implementadas, pero SLP no está específicamente orientado a la interacción entre personas y servicios.

### 2.3.2 Ninja *Service Discovery Service*

En el ámbito del sistema Ninja SDS [Hodes et al., 2002], un servicio puede ser tanto *hardware* como *software*. El *software* que se contempla son aplicaciones que realizan cálculos o acciones en favor de sus clientes, e.g., un servicio puede ser una aplicación que permite que un usuario controle las luces de un cuarto. Cualquier *hardware* puede ser un servicio e.g., una impresora y un fax. En un ambiente, donde existen cientos o quizá miles de servicios disponibles a los usuarios finales, puede ser un reto para dichos usuarios localizar el servicio más apropiado para realizar una tarea. Un servicio apropiado será aquel que satisfaga las especificaciones de un usuario (e.g., localización y accesibilidad). El sistema Ninja SDS permite que los clientes (usuarios) hagan búsquedas efectivas de servicios.

Ninja SDS es un repositorio de datos seguro, escalable y tolerante a fallas, que está formado por tres componentes principales: servicios, clientes y servidores SDS. Como se mencionó, un servicio puede ser *hardware* o *software*. Los clientes son usuarios que necesitan descubrir servicios. Los servidores SDS permiten que los clientes hagan búsquedas de recursos. Un servidor SDS solicita información sobre cada servicio, que utiliza posteriormente para determinar los servicios que satisfacen las necesidades del cliente. El servidor SDS está organizado en una estructura jerárquica. La información sobre los servicios y las búsquedas de los clientes están asignadas a un servidor SDS particular. Si un servidor SDS se sobrecarga, entonces un nuevo servidor hijo se iniciará. Esta estructura permite que el sistema Ninja SDS sea escalable.

Ninja SDS utiliza documentos XML tanto en las descripciones de los servicios (i.e., la información provista por un servicio) como en las búsquedas de los clientes. La descripción de un servicio incluye su localización, sus capacidades y la dirección Java RMI donde se encuentra almacenado. Una vez que ha sido descrito, el servicio se registra en un servidor SDS. Por su parte, la búsqueda de un cliente contiene la especificación de sus requerimientos de servicio por medio de plantillas XML.

Un servicio debe realizar tres tareas para poder formar parte del sistema Ninja SDS. La primera consiste en escuchar continuamente los anuncios de los servidores SDS en un canal *multicast* global, con el objetivo de determinar cuál es el servidor SDS apropiado para dicho servicio. La búsqueda del servidor SDS correcto se realiza varias veces, ya que los servidores SDS pueden fallar o nuevos servidores pueden ser agregados al sistema. Posteriormente, el servicio envía al servidor SDS, donde desea registrarse, un mensaje *multicast* que contiene su descripción. Finalmente, el servicio contacta a un administrador de capacidades para definir apropiadamente el tipo de acceso permitido a los usuarios

individuales. El administrador de capacidades es un mecanismo de control de acceso, que permite a los propios servicios controlar a los usuarios que podrán descubrirlos.

### 2.3.3 Jini

La tecnología Jini [Keith, 2000] permite construir sistemas distribuidos que están compuestos por federaciones de servicios de red y clientes. La tecnología Jini puede ser utilizada para crear sistemas de red adaptativos, escalables y flexibles, los cuales son necesarios en ambientes computacionales dinámicos. Esta tecnología está compuesta por tres elementos principales: una infraestructura, un modelo de programación y un conjunto de servicios. La infraestructura está formada de componentes que permiten construir un sistema de Jini. Los servicios representan entidades de *hardware* o *software*, que pueden ser utilizadas por una aplicación, un dispositivo o incluso una persona. En términos de programación, un servicio está implementado mediante objetos Java que cuentan con una interfaz, la cual define las operaciones que dicho servicio puede realizar. Por su parte, el modelo de programación es un conjunto de interfaces que permite construir servicios confiables (representados por objetos Java).

La infraestructura está integrada por tres protocolos y tres participantes. Los protocolos son de descubrimiento, búsqueda y unión, mientras que los participantes son un cliente, un proveedor de servicios y un servicio *look up*. Tanto los protocolos como los participantes trabajan juntos para hacer funcionar un sistema Jini. Un cliente es el que busca un servicio, un proveedor es aquel que genera servicios y un servicio *look up*, se encarga de registrar los servicios que conforman un sistema en Jini.

Los protocolos de descubrimiento y unión permiten que un nuevo servicio se agregue al sistema. Así, cuando un nuevo servicio se conecta, busca un servicio *look up* donde pueda registrarse. Esta búsqueda se realiza mediante una petición *multicast*. Una vez que el nuevo servicio encuentra un servicio *look up*, el protocolo de unión le ayuda a registrarse. Este registro consiste en hacer una copia del objeto Java que representa al nuevo servicio en el servicio *look up*.

El protocolo de descubrimiento se utiliza cuando un cliente necesita encontrar o invocar un servicio. Dicho cliente hace una petición al servicio *look up*, el cual busca en los servicios registrados. Si encuentra un servicio que satisfaga la petición del cliente, entonces el servicio *look up* transfiere al cliente el objeto Java que representa el servicio. De esta forma, el cliente podrá conectarse al servicio o incluso utilizarlo remotamente, dependiendo de la naturaleza de dicho servicio.

## 2.4 Discusión sobre el trabajo relacionado

Los sistemas colaborativos que han sido analizados ofrecen herramientas que permiten crear un espacio ubicuo, donde los usuarios reciben información relevante de acuerdo a su contexto. El sistema CHIS proporciona un mecanismo basado en el principio de mensajería instantánea, el cual tiene el objetivo de enviar información contextual a sus usua-

rios. Este mecanismo trabaja en conjunto con un algoritmo de triangulación de señales WiFi por medio del cual se localiza el dispositivo móvil (PDA) que porta cada colaborador. Por su parte, el sistema Proxy Lady implementa la comunicación oportunista para el envío de información contextual por medio de la tecnología *bluetooth*. Esta tecnología ayuda a localizar el PDA del colaborador con quien se desea interactuar. El sistema ABC proporciona la funcionalidad de *roaming*, la cual permite que un colaborador pueda pausar una actividad y reiniciarla en cualquier dispositivo de la organización donde trabaja (en este caso un hospital).

Tanto el sistema CHIS como el sistema Proxy Lady ofrecen una forma implícita de localizar dispositivos móviles para enviar información contextual. Los dispositivos móviles representan a los colaboradores, ya que se asume que un usuario siempre está junto a su PDA. Aunque resulta práctico localizar dispositivos móviles, la representación de los colaboradores por medio de PDAs puede reducir las ventajas de estos sistemas (e.g., si un usuario no lleva consigo su PDA, el sistema proporcionará información incorrecta). Aunque los sistemas CHIS y Proxy Lady proporcionan información sobre la presencia y la ubicación de los colaboradores, ninguno ofrece información sobre su disponibilidad. Por su parte, el sistema ABC proporciona información dependiente del contexto mediante el concepto de actividades, las cuales pueden ser iniciadas y reanudadas por un colaborador en cualquier dispositivo que se encuentre dentro de la organización (e.g., un hospital). Este sistema descarta los dispositivos móviles que poseen características limitadas de procesamiento y despliegue. Por esta razón, el sistema ABC no proporciona ningún mecanismo para localizar dispositivos móviles, ni herramientas para ubicar colaboradores.

Los tres sistemas colaborativos analizados proporcionan información sobre los recursos que se encuentran dentro de la organización donde operan. El sistema CHIS ofrece información sobre la ubicación de recursos físicos públicos (i.e., sin restricciones de uso) y permite compartir recursos informáticos entre los usuarios. Mediante el sistema Proxy Lady, los colaboradores pueden compartir recursos informáticos privados. De forma similar, el sistema ABC permite compartir recursos informáticos bajo el concepto de actividades. Aunque estos sistemas toman en cuenta el uso compartido de recursos, ninguno se centra en la importancia de proporcionar información sobre su disponibilidad. Este requerimiento involucra múltiples parámetros relativos a los recursos de carácter privado, ya que se deben cumplir condiciones estáticas que define el dueño del recurso (e.g., restricciones de uso) y condiciones dinámicas que delatan el estado del recurso (e.g., no disponible por mantenimiento).

Por su parte, los protocolos de descubrimiento de servicios, que han sido estudiados, presentan características comunes ya que, en general, su objetivo es promover el uso compartido de servicios (*hardware* o *software*) que están disponibles en un sistema en red. Desafortunadamente, estos protocolos de descubrimiento no están completamente enfocados a clientes humanos. El protocolo SLP sólo proporciona algunas herramientas opcionales, las cuales pueden ser implementadas para permitir que los clientes humanos hagan búsquedas de servicios. Por su parte, JINI provee algunas interfaces limitadas que ayudan a los usuarios en la especificación de sus requerimientos de servicios. El protocolo Ninja SDS permite que dichas necesidades sean descritas mediante plantillas

XML. Los protocolos de descubrimiento de servicios están desarrollados para un ambiente de trabajo desacoplado, donde los clientes no saben quien es el proveedor del servicio, sino simplemente utilizan el servicio que esté disponible. En este contexto, la disponibilidad no se refiere necesariamente a las restricciones impuestas por el dueño del servicio, sino a la vigencia del registro de este servicio.





## Capítulo 3

# Análisis y diseño del sistema RAMS

El objetivo del presente capítulo es describir un sistema que promueve el uso compartido de recursos, en un ambiente colaborativo y ubicuo, por medio de herramientas que facilitan su administración. Primero, se presenta un escenario que pone en evidencia los potenciales beneficios de la implantación de este sistema en una organización (sección 3.1). Posteriormente se describen las categorías de recursos que pueden ser administrados por el sistema (sección 3.2) así como la modularización de éste en subsistemas especializados (sección 3.3). El primer subsistema que se presenta es el *broker*, el cual está integrado por: a) un módulo de publicación/subscripción de recursos y b) un filtro de información (sección 3.4). Después se describe el motor de inferencia, que actúa como un puente entre la información estática (e.g., características técnicas) y la información dinámica (e.g., ubicación actual de un colaborador) sobre el estado de los recursos (sección 3.5). En particular, una parte de la información dinámica es capturada implícitamente por un conjunto de sensores que mantienen actualizadas las ubicaciones absolutas y relativas de los recursos (sección 3.6). Posteriormente se detalla el sistema de notificación, el cual está encargado de dar a conocer los recursos disponibles que podrían satisfacer las necesidades de los colaboradores (sección 3.7). Finalmente se describe el esquema de distribución que ha sido diseñado con la intención de contar con un sistema que funcione de forma descentralizada (sección 3.8).

### 3.1 Escenario

En una organización con múltiples recursos distribuidos en varias oficinas (que a su vez pueden estar localizadas en distintos edificios) es prácticamente imposible que un colaborador tenga todos los recursos físicos de la organización (e.g., impresoras, pizarrones interactivos y laboratorios) en su propio cubículo ya que, además de resultar incosteable, las características usuales de una oficina no incluyen amplios espacios. Tampoco resulta práctico duplicar todos los recursos informáticos (e.g., multimedia y *software*) de la organización en algún dispositivo de almacenamiento de cada colaborador, debido a que algunos de ellos son costosos en cuanto a espacio de almacenamiento y/o licencia de instalación.

La nula practicidad de poseer individualmente todos los recursos físicos e informáticos de una organización, delata la posibilidad de que un colaborador necesite utilizar, en algún momento, un recurso que no le pertenece. En una situación común, este colaborador tendría que preguntar a sus colegas: 1) si cuentan con un recurso que posea las características técnicas que él requiere y además 2) si están dispuestos a compartirlo en un momento dado. Esta búsqueda podría consumir demasiado tiempo y esfuerzo al colaborador que requiere el recurso y también podría ocasionar interrupciones e incluso situaciones incómodas a los colegas que eventualmente poseen el recurso. Para tratar de solventar los problemas ocasionados por situaciones similares a ésta, se propone el sistema RAMS (*Resource Availability Management System*), el cual promueve el uso compartido de recursos al ayudar a sus dueños a administrarlos y al proveer, a sus potenciales usuarios, información actual sobre el estado de los recursos compartidos.

Suponga que dentro de su ambiente de trabajo físico (ver Figura 3.1), el Prof. Ruiz decide compartir con sus colegas un *plotter* de alta calidad que imprime gráficos técnicos en formatos grandes. Para administrar la disponibilidad de su recurso, el Prof. Ruiz utiliza el sistema RAMS, el cual ofrece funciones de publicación que le permiten detallar las principales características técnicas del *plotter* (e.g., resolución, velocidad y tamaño del papel), así como la forma en que éste será compartido (e.g., cómo, cuándo y con quién).



Figura 3.1: Ambiente colaborativo

Adicionalmente, el Prof. Ruiz puede definir restricciones de uso, roles y permisos asociados a su recurso (e.g., sus estudiantes tienen el derecho de imprimir  $x$  páginas por mes). Como el *plotter* se encuentra en su oficina, el Prof. Ruiz puede definir un horario estimado para acceder al recurso, con el fin de evitar que los usuarios del *plotter* lo interrumpan en cualquier momento.

En un momento dado el Prof. García, quien colabora con el Prof. Ruiz, necesita

imprimir el diseño de un circuito VLSI (*Very Large Scale Integration*), así que se suscribe al sistema RAMS para describir sus requerimientos.

El sistema RAMS busca un conjunto de recursos adecuados para el Prof. García, en base a:

1. las características técnicas y restricciones de uso proporcionadas cuando se publicaron los recursos,
2. los requerimientos que el Prof. García especificó cuando se suscribió al sistema RAMS,
3. los permisos que se atribuyeron al Prof. García sobre los recursos registrados en el sistema.

Al terminar la búsqueda, el sistema RAMS informa al Prof. García que el *plotter* de alta calidad del Prof. Ruiz es capaz de satisfacer sus necesidades, desde el punto de vista técnico. Además le proporciona información sobre la localización del *plotter*, su horario de acceso y disponibilidad actual. A pesar de que el horario de acceso autoriza al Prof. García a utilizar el *plotter* en ese momento, el sistema RAMS le indica que está temporalmente no disponible, ya que el Prof. Ruiz no se encuentra en su oficina.

Algunos minutos después, uno de los estudiantes del Prof. Ruiz, la Srita. López, va a la oficina de su asesor para probar la aplicación que está desarrollando en un pizarrón interactivo. Ella no tiene restricciones de acceso a la oficina cuando su asesor está ausente. Así, al entrar a la oficina, el sistema RAMS es capaz de reconocerla y automáticamente inferir su nueva localización (ya que anteriormente estaba en su cubículo). La Srita. López requiere quedarse dos horas en la oficina del Prof. Ruiz para realizar sus pruebas. Como no necesita demasiada concentración, decide declararse disponible, lo que significa que otras personas pueden interactuar con ella en cualquier momento. Aunque el Prof. Ruiz permanece ausente, el sistema RAMS infiere y notifica al Prof. García que el *plotter* del Prof. Ruiz ya está disponible. Por supuesto que si la Srita. López se hubiera declarado no disponible, el *plotter* de su asesor también lo estaría.

Aunque el Prof. García recibió la notificación del sistema RAMS, la cual indica que el *plotter* del Prof. Ruiz ya está disponible, el Prof. García decide esperar a que su colega Ruiz regrese a su oficina, antes de enviar su diseño al *plotter*.

Minutos más tarde el Prof. Ruiz arriba a su oficina, así que el sistema RAMS infiere su nueva ubicación y la notifica a todos los interesados. Al recibir la notificación, el Prof. García envía su diseño del circuito VLSI al *plotter* y se desplaza a la oficina de su colega Ruiz a recoger su diseño impreso y a discutirlo con él.

Los beneficios que ofrece el sistema RAMS a sus usuarios son:

- facilita el uso compartido de recursos por medio de la definición de restricciones de uso y la asignación/denegación de roles y permisos a usuarios y grupos específicos;
- permite modificar, en cualquier momento, las preferencias de uso y el estado de sus recursos;
- ofrece información relevante y actualizada sobre la presencia, la localización y la disponibilidad de los recursos compartidos.

Cabe mencionar que cuando un colaborador decide compartir un recurso, éste no se volverá público ya que solo podrá ser utilizado o accedido por las personas autorizadas, siempre y cuando cumplan con las restricciones de uso del recurso.

## 3.2 Tipos de recursos

El sistema RAMS administra información de estado correspondiente a tres tipos de recursos:

- **Recursos humanos:** son aquellos colaboradores que deciden interactuar entre ellos por medio del sistema RAMS.

Las características específicas de las personas incluyen constante movilidad y variable disponibilidad. En consecuencia, para poder administrar la información de estado asociada a este tipo de recursos, el sistema RAMS utiliza un sistema de reconocimiento facial (cf. sección 3.6.1) que determina la presencia y la localización de los colaboradores. Además, el sistema RAMS toma en cuenta eventos de aplicación que permiten a cada colaborador cambiar su estado, e.g., de “disponible” a “no disponible” cuando no quiera ser interrumpido.

- **Recursos físicos:** son todos aquellos dispositivos computacionales (e.g., computadoras, *PDA*s, escáneres, cámaras digitales y proyectores) e incluso recursos no computacionales (e.g., salas de reuniones) que los colaboradores pueden compartir con sus colegas.

A fin de poder compartir un recurso físico, el sistema RAMS ofrece un módulo de publicación *ad-hoc* (cf. sección 3.4.1) que permite, al dueño del recurso, definir sus características técnicas, sus restricciones de uso y la asignación/denegación de permisos a sus colegas.

Los recursos físicos se dividen en públicos y privados. Los recursos públicos (e.g., salas de reunión y auditorios) generalmente son poseídos por recursos no humanos (e.g., un departamento de la organización). Por el contrario, los recursos privados (e.g., impresoras y computadoras) pertenecen a un colaborador o a un grupo.

Una de las principales diferencias entre recursos públicos y recursos privados es que estos últimos generalmente están localizados en un área restringida (e.g., oficina). En el caso de los recursos físicos públicos es posible que el acceso sea libre o que exista un responsable que permita o deniegue el acceso. En consecuencia, es necesario que el dueño o el responsable se encuentre presente para poder hacer uso de los recursos.

- **Recursos informáticos:** son aquellas entidades (e.g., multimedia y *software*) que están alojadas en dispositivos de almacenamiento masivo.

Con el fin de poder compartir un recurso informático, el sistema RAMS también ofrece un módulo de publicación *ad-hoc* (cf. sección 3.4.1) que permite, al dueño del recurso, especificar sus características técnicas, los sitios en los que está replicado, sus condiciones de acceso (e.g., nombre de usuario y contraseña), sus restricciones de uso y la asignación/denegación de permisos a sus colegas.

Los recursos informáticos también pueden ser públicos o privados. Los recursos públicos generalmente son controlados por algún responsable, pero pertenecen a la organización o a uno de sus departamentos. El *software* (e.g., antivirus, aplicaciones de oficina y sistemas operativos) que se pone a disposición de los colaboradores que trabajan en la organización forma parte de los recursos informáticos públicos. Sin embargo, aún cuando los recursos sean públicos, pueden existir restricciones de uso e instalación (e.g., número limitado de licencias de instalación).

Al igual que los recursos físicos privados, los recursos informáticos privados son aquellos que pertenecen a un colaborador o a un grupo.

Una de las características importantes que diferencia a los recursos informáticos del resto es que un mismo recurso puede estar almacenado simultáneamente en varios sitios.

Los recursos que se comparten, mediante el sistema RAMS, tienen características diferentes, así que se deben administrar individualmente, con el fin de ofrecer información especializada sobre el estado de cada recurso.

### 3.3 Esquema de aplicación

El uso compartido de recursos entre los colaboradores de una organización requiere del manejo de información actualizada, que provenga de herramientas automatizadas y de los propios colaboradores. El sistema RAMS realiza la recopilación, la administración y el almacenamiento de esta información por medio de los subsistemas que lo integran.

En esta sección, se describen los conceptos de agente productor y de agente consumidor (sección 3.3.1), los cuales se refieren a los diferentes roles que un usuario del sistema RAMS puede jugar. Posteriormente se introduce el concepto de evento (sección 3.3.2), el cual contiene información sobre las acciones que realiza un usuario cuando cambia el estado de los recursos administrados por el sistema RAMS. Por último, se describe brevemente la arquitectura de *software* del sistema RAMS, así como los subsistemas que la conforman.

### 3.3.1 Agentes

El sistema RAMS se basa en el modelo asíncrono de publicación/subscripción [Muehl et al., 2007]. Se eligió este modelo, por encima de otros modelos de mensajes (e.g., la cola de mensajes), porque la mayoría de sus principios pueden ser aplicados a las entidades del sistema RAMS. Así, los colaboradores pueden jugar los roles de productores y/o consumidores de eventos relacionados con el estado (i.e., presencia, localización y disponibilidad) de los recursos compartidos. Además, este modelo de mensajes define funciones que permiten a los colaboradores: a) publicar sus recursos para compartirlos con algunos de sus colegas y b) suscribirse para buscar los recursos que necesitan utilizar.

Sin embargo, a diferencia de un sistema de publicación/subscripción tradicional (i.e., en el que los productores no conocen a los consumidores de sus mensajes ni viceversa), un sistema de publicación/subscripción cooperativo necesita identificar a sus usuarios para permitirles definir filtros que controlen el alcance de difusión y de recepción de los eventos. En respuesta a este requerimiento, el sistema RAMS asignará un identificador único a cada colaborador al momento de registrarse. El proceso de asignación de identificadores se basa en el concepto de agente.

La definición de agente incluye información para identificar no solo la entidad activa (i.e., un usuario) que ejecuta una acción sobre el estado de los recursos compartidos, sino también la fuente de la acción (i.e., un sitio y una aplicación).

*Definición de agente:* sean  $u$  un usuario que pertenece al conjunto de colaboradores  $C_{col}$  y  $f$  una fuente compuesta por un sitio  $sI$  y una aplicación  $aP$ . Un agente  $C_a$  (ver expresión 3.1) es un conjunto de parejas  $(u, f)$  tal que  $u \in C_{col}$  y  $f \in sI \times aP$ .

$$C_a = \{(u, f) \mid u \in C_{col}, f \in sI \times aP\} \quad (3.1)$$

La función  $C_a$  está representada por:

$$C_a : C_{col} \rightarrow (sI \times aP)$$

Esta definición de agente permite: a) controlar las acciones que los colaboradores realizan sobre los recursos compartidos y b) definir un conjunto de roles que especifique la forma en que cada agente puede actuar sobre cada recurso. Cabe mencionar que el concepto de rol (cf. sección 3.4.1) es esencial para expresar la organización social del grupo de trabajo y para resguardar a los recursos de acciones no autorizadas.

Gracias a la definición de agente, el sistema RAMS permite adaptar las funciones de filtrado y notificación del modelo tradicional de publicación/subscripción a los requerimientos específicos de los sistemas cooperativos. De hecho, la identidad y el rol del agente constituyen dos parámetros fundamentales de estas funciones.

El sistema RAMS maneja dos tipos de agentes:

1. **productores:** son los agentes que publican algunos de sus recursos, con el fin de compartirlos con algunos de sus colegas. Por medio del sistema RAMS, estos agentes pueden incluso generar eventos que cambien el estado de sus recursos;

2. **consumidores:** son los agentes que se subscriben al sistema RAMS, con el fin de recibir eventos sobre el estado de los recursos a los que tiene derecho de uso o de acceso.

### 3.3.2 Eventos

El sistema RAMS ofrece a los agentes la posibilidad de modificar explícitamente la información referente al estado actual de un recurso, ya que un agente podría necesitar cambiar esta información por cuestiones de privacidad, mantenimiento o actualización. Para poner en evidencia esta funcionalidad del sistema RAMS, considere el siguiente escenario:

El Prof. García publicó una cámara digital en el sistema RAMS y permitió el libre uso de este recurso al Prof. Jimenez. El Prof. García está trabajando en un reporte que tiene que entregar por la noche, así que necesita mucha concentración y decide declarar sus recursos físicos compartidos no disponibles hasta el día siguiente. En respuesta a este cambio, el sistema RAMS genera el evento correspondiente y lo notifica a los interesados. Un par de horas después, el Prof. Jimenez necesita utilizar la cámara de su colega, pero no podrá hacerlo porque el estado actual de este dispositivo es no disponible.

El sistema RAMS propone funciones que permiten a los colaboradores: a) modificar su propia disponibilidad y b) modificar la presencia, disponibilidad y ubicación de sus recursos compartidos, en cualquier momento. La información de estado es transmitida vía eventos de una aplicación a otra.

Un evento es producido por un agente que decide cambiar el estado actual (e.g., disponible) de un recurso a otro estado (e.g., no disponible o en mantenimiento) por un tiempo determinado.

*Definición del conjunto de recursos:* el conjunto de recursos  $C_r$  (ver expresión 3.2) está compuesto por uno o varios recursos, tal que cada recurso está definido por un tipo de recurso  $tR$ , que pertenece a la lista  $lR$  de los tipos de recursos definidos (i.e., humano, físico o informático), un identificador de publicación  $nR$ , que forma parte de la lista  $lP$  de los recursos publicados y un estado actual  $eA$ , que puede ser alguno de los estados posibles  $C_e$  (e.g., disponible, no disponible, presente o ausente).

$$C_r = \{(tR, nR, eA)^* \mid tR \in lR, nR \in lP, eA \in C_e\} \quad (3.2)$$

*Definición de evento:* sean  $ag$  un agente que pertenece a  $C_a$ ,  $ob$  una pareja formada por un recurso  $rec \in C_r$  y un periodo de tiempo  $t \in T$  (ver expresión 3.3) y  $C_e$  un conjunto de estados validos. Un evento  $E$  (ver expresión 3.4) es un conjunto de pares  $(ag, ob)$ , tal que  $ag \in C_a$  y  $ob \in O$ .

$$O = \{(rec, t) \mid rec \in C_r, t \in T\} \quad (3.3)$$

$$O = C_r \times T$$

$$E = \{(ag, ob) \mid ag \in C_a, ob \in O\} \quad (3.4)$$

La función  $E$  está representada por:

$$E : (C_a \times O) \longrightarrow C_e$$

### 3.3.3 Componentes del sistema RAMS

Con el fin de ilustrar la interacción entre los agentes y el sistema RAMS se presentan los siguientes casos de uso.

Un agente productor puede realizar las siguientes operaciones (ver Figura 3.2):

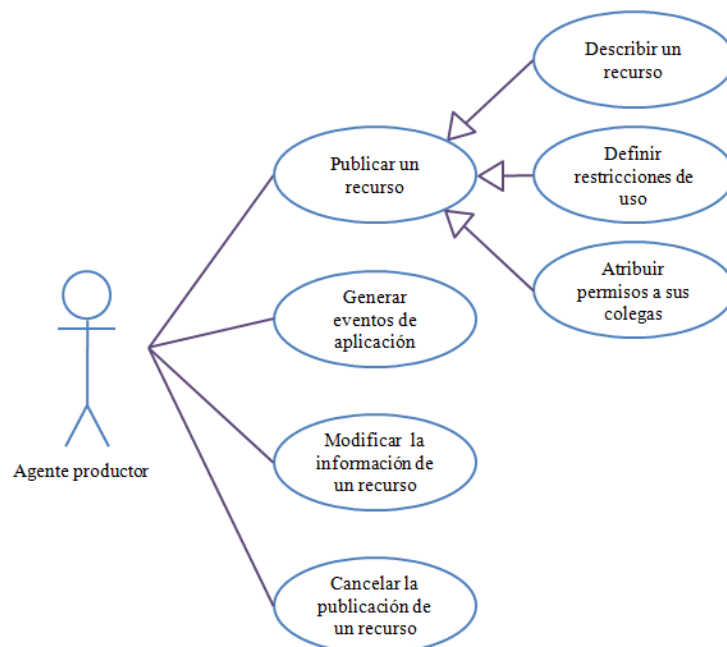


Figura 3.2: Casos de uso para un agente productor

1. **Publicar un recurso:** si un agente productor desea compartir un recurso con algunos de sus colegas, entonces podrá publicarlo mediante funciones que le permiten:
  - a) describir el recurso en términos de sus capacidades o características técnicas,
  - b) definir las limitantes que restringirán su uso o acceso,
  - c) asignar los permisos a los colaboradores con quienes desea compartirlo.
2. **Generar eventos de aplicación:** un agente productor podrá cambiar el estado de sus recursos en cualquier momento mediante aplicaciones específicas (e.g., activar el modo “no molestar” o poner un recurso físico temporalmente “no disponible”);



3. **Modificar la información de un recurso:** un agente productor podrá modificar las características técnicas de un recurso, sus restricciones de uso o acceso y los permisos atribuidos a sus colegas sobre dicho recurso, aún después de haber sido publicado.
4. **Cancelar la publicación de un recurso:** un agente productor podrá, en cualquier momento, dejar de compartir alguno de los recursos que previamente publicó.

Por su parte, un agente consumidor puede efectuar las siguientes operaciones (ver Figura 3.3):

1. **Subscribirse:** si un agente consumidor requiere un recurso, entonces proporcionará información al sistema RAMS mediante funciones que le permiten:
  - a) describir las características técnicas del recurso solicitado,
  - b) activar y desactivar filtros de información, con el fin de recibir únicamente notificaciones de los recursos en los que el agente consumidor está interesado.
2. **Cancelar una suscripción:** un agente consumidor podrá, en cualquier momento, dejar de recibir información sobre alguno de los recursos a los que se suscribió.

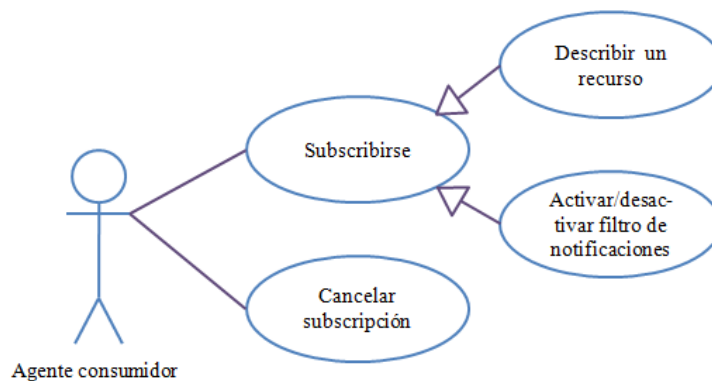


Figura 3.3: Casos de uso para un agente consumidor

El sistema RAMS se compone de cuatro subsistemas esenciales (ver Figura 3.4): 1) un *broker*, 2) un motor de inferencia, 3) un conjunto de sensores y 4) un sistema de notificación. Cada uno de estos subsistemas cumple importantes funciones para que el sistema RAMS pueda proporcionar un soporte adecuado, tanto a los colaboradores que desean compartir sus recursos como a los colaboradores que buscan un recurso que satisfaga sus necesidades.

El *broker* está encargado de controlar la interacción entre los agentes y el sistema RAMS. Un agente productor interactúa con el sistema RAMS por medio del módulo de publicación (ver Figura 3.4 ref. #1), el cual le permite describir alguno de sus recursos,

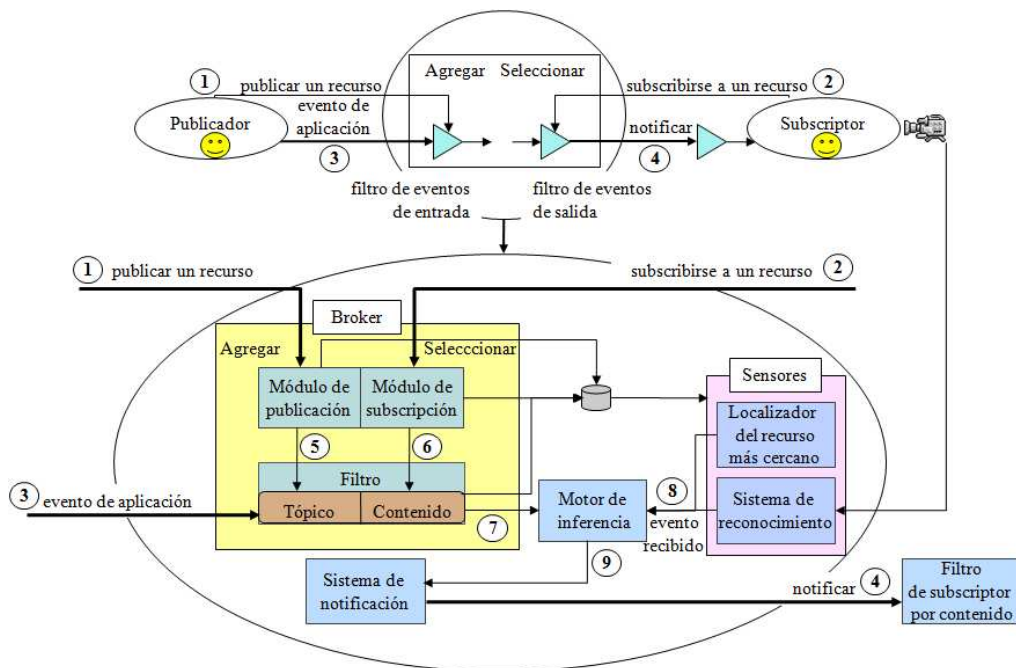


Figura 3.4: Arquitectura de *software* del sistema RAMS

definir sus restricciones de uso y atribuir permisos a los colaboradores con quienes desea compartirlo. El módulo de publicación envía esta información a un filtro, el cual se encarga de clasificar el recurso en el tópico adecuado, e.g., impresoras, videos y salones de clase (ver Figura 3.4 ref. #5).

Un agente consumidor interactúa con el sistema RAMS mediante el módulo de suscripción (ver Figura 3.4 ref. #2), el cual le permite describir el recurso que le interesa o que necesita utilizar. El módulo de suscripción envía esta información a un filtro por contenido (ver Figura 3.4 ref. #6), el cual selecciona solo aquellos recursos cuyos atributos corresponden con la descripción técnica que el agente consumidor especificó. De este proceso de filtrado, se obtiene un conjunto de recursos que eventualmente podría satisfacer las necesidades del agente consumidor.

Posteriormente, el filtro envía este conjunto de recursos al motor de inferencia (ver Figura 3.4 ref. #7), el cual determinará el recurso más adecuado para satisfacer la petición del agente consumidor. Para tomar esta decisión, el motor de inferencia utiliza información estática y dinámica de los recursos.

La información estática es aquella que proporciona un agente productor cuando publica un recurso. Esta información incluye las características técnicas del recurso, sus restricciones de uso y los permisos otorgados a los colaboradores.

La información dinámica se obtiene de: 1) un conjunto de sensores (ver Figura 3.4 ref. #8), los cuales proporcionan información sobre la disponibilidad, presencia y ubicación de los recursos y 2) aplicaciones de usuario (ver Figura 3.4 ref. #3), las cuales permiten que un agente productor modifique explícitamente la presencia, disponibilidad o ubicación de

sus recursos.

En particular, el conjunto de sensores está compuesto por: a) un sistema de reconocimiento de caras y b) un sistema de localización del recurso más cercano. Estos dos sensores son utilizados por el motor de inferencia para determinar la disponibilidad de un recurso físico. El sistema de reconocimiento de caras también es útil para localizar un recurso humano, ya que es capaz de identificar a los colaboradores por medio de su apariencia física.

Una vez que el motor de inferencia realizó el proceso de selección de los recursos más apropiados, envía este resultado al sistema de notificación (ver Figura 3.4 ref. #9), el cual se encarga de transmitirlo a un filtro (ver Figura 3.4 ref. #4) que el agente consumidor definió al momento de hacer la petición. Dicho filtro permite restringir las notificaciones que un agente recibe del sistema RAMS. De esta forma, solo se le notificará información de recursos que realmente le interesan.

## 3.4 Broker

El *broker* es un subsistema compuesto por los módulos de publicación y de suscripción y por un filtro de información. Este subsistema recibe información tanto de los agentes productores como de los agentes consumidores y la filtra para alimentar al motor de inferencia.

### 3.4.1 Módulos de publicación y de suscripción

Los módulos de publicación y de suscripción permiten respectivamente a los agentes productores y consumidores describir recursos en términos de sus características técnicas (ver Figura 3.5). Además los agentes productores pueden definir restricciones de uso y atribuir permisos a los agentes consumidores, con el fin de proteger sus recursos de abusos.



Figura 3.5: Funciones de los módulos de publicación y de suscripción

A continuación se describen las funciones que los agentes productores y consumidores realizan respectivamente mediante los módulos de publicación y de suscripción.

## Descripción de recursos

Cuando un agente productor decide compartir un recurso con sus colegas, es necesario que lo describa utilizando funciones dedicadas que proporciona el sistema RAMS. Obviamente esta descripción no es igual para todos los recursos, ya que el sistema RAMS necesita información específica para determinar la mejor opción que satisfaga los requerimientos de cada agente consumidor. La información que se requiere de un recurso depende no solo del tipo al que pertenece (e.g., recursos físicos) sino también del subtipo (e.g., impresoras, escáneres y pizarrones interactivos). A continuación se describe un ejemplo de las características esenciales que componen la descripción de cada tipo de recurso.

Un recurso humano (i.e., un colaborador), que decide interactuar con sus colegas por medio del sistema RAMS, proporciona la siguiente información: 1) su estatus social, 2) su ubicación por omisión y 3) su agenda de trabajo. Como se observa en la Tabla 3.1 (a), la descripción del Prof. Ruiz está compuesta por su nombre, su afiliación, su área de investigación y su ubicación por omisión (e.g., oficina). Adicionalmente, el Prof. Ruiz puede proporcionar su horario de oficina, así como su plan de cursos y reuniones de trabajo (ver Tabla 3.1 (b)). El sistema RAMS utiliza esta información para saber si es posible que un colaborador específico se encuentre en un momento dado dentro de la organización, pero comprueba su presencia por medio de un sensor (cf. sección 3.6.1).

Descripción de un colaborador	
a) Nombre	Prof. Juan Ruiz
Afiliación	CINVESTAV
Área de investigación	<i>Groupware</i>
Ubicación	Oficina B201

Agenda de trabajo		
b) Horario de oficina	Lunes - Viernes	9 - 19 hrs.
Junta de profesores	Lunes	12 - 14 hrs.
Curso de programación	Martes y Jueves	15 - 17 hrs.
...		

Tabla 3.1: Descripción de un recurso humano

Puesto que un recurso humano puede desplazarse regularmente de un lugar a otro dentro de la organización, es imposible asumir que se encuentra en su oficina durante todo su horario de trabajo. Considerando el carácter móvil de este tipo de recursos, el sistema RAMS utiliza un sensor para determinar su localización (cf. sección 3.6.1).

Por otra parte, cada subtipo de recurso físico posee características técnicas propias, las cuales generalmente son muy diferentes de las de otros subtipos (e.g., una *laptop* no puede ser descrita con los mismos atributos que una cámara Web). Por esta razón, una descripción apropiada de cada subtipo de recurso permitirá que el sistema RAMS disponga de la información necesaria para satisfacer las necesidades de cada agente consumidor.

La descripción de un recurso físico contiene: 1) sus características técnicas, 2) su ubicación por omisión e 3) información sobre el dueño o responsable del recurso. En la Tabla

3.2 (a) se muestran las características técnicas de un *plotter*, entre las que se encuentran su marca y modelo, su tecnología de impresión, su resolución, etc. Adicionalmente, el agente productor (en este caso, el Prof. Ruiz) indica que su *plotter* se encuentra en su oficina y provee información sobre él mismo, ya que es el responsable de permitir el acceso a este recurso.

Descripción de <i>plotter</i>	
Marca y modelo	HP DesignJet 4500
Tecnología	Laser
Impresión por ambos lados	No
Color o blanco y negro	Blanco y negro
Resolución	Hasta 508 x 508 ppi
Puerto USB	Si
Ubicación	Oficina B201
Dueño	Prof. Ruiz

Asignación de permisos	
Estudiantes	Imprimir
Prof. García	Imprimir
Administrativos	Sin permisos

Restricciones de uso	
Estudiantes	10 páginas por mes
Prof. García	Sin restricciones

Tabla 3.2: Descripción de un recurso físico

Finalmente, la descripción de un recurso informático comprende: 1) sus características técnicas, 2) sus requerimientos de ejecución, 3) datos de control de acceso en caso de ser necesario, 4) su ubicación e 5) información sobre el dueño del recurso. En la Tabla 3.3 (a) se presenta la descripción de un video de un curso de “Ingeniería de *Software*”. Entre sus características técnicas se encuentran su formato, su duración y el tipo de reproductor multimedia que permite visualizar este recurso. Además, el agente productor (en este caso, el Prof. García) define un nombre de usuario y una contraseña de acceso y proporciona información sobre la ubicación del video (que incluye el protocolo de comunicación requerido) y sobre él mismo, ya que es el dueño de este recurso.

A diferencia de los recursos físicos (que están asociados a una ubicación fija) y de los recursos humanos (quienes pueden cambiar regularmente de ubicación), los recursos informáticos pueden estar replicados en diferentes lugares (i.e., dispositivos de almacenamiento). Esta característica permite que los recursos informáticos sean más accesibles que los recursos físicos, ya que si un sitio no está disponible en un momento dado, es posible acceder a una réplica del recurso almacenada en otro sitio.

Descripción de un video	
Título	“Ingeniería de <i>Software</i> ”
Autor	Prof. García
Formato	mpeg
Duración	60 horas
Requerimientos de ejecución	Reproductor de multimedia (Windows Media Player, Real Player, QuickTime)
Usuario	Circuits232
Contraseña	*****
Ubicación	http://compu/videoIng.mpeg
Dueño	Prof. García

a)

Asignación de permisos	
Prof. Ruiz	Descarga y modificación
Estudiantes	Visualización

b)

Restricciones de uso	
Prof. Ruiz	Sin restricciones
Estudiantes	Lun y Jue de 16 - 18 hrs.

c)

Tabla 3.3: Descripción de un recurso informático

### Otorgamiento y negación de permisos

Los permisos se refieren a las operaciones que un agente consumidor puede realizar sobre un recurso compartido. La asignación de permisos hace posible que un agente productor restrinja en mayor o menor medida el uso de sus recursos compartidos para resguardarlos de abusos. Por esta razón, el sistema RAMS ofrece a los agentes productores la posibilidad de modificar, en cualquier momento, los permisos que hayan asignado a los agentes consumidores sobre sus recursos.

Los permisos se pueden otorgar o negar a un agente consumidor específico o a un grupo. Si el proceso de atribución se realiza a nivel de grupos, el agente productor podrá otorgar o denegar permisos, de forma rápida y sencilla, a un grupo de agentes consumidores que pertenecen a una comunidad particular. Pero si se requiere una atribución más específica, el agente productor podrá otorgar o negar permisos a nivel de agentes individuales.

En la Tabla 3.2 (b) se muestran los permisos de los agentes consumidores sobre el *plotter* del Prof. Ruiz. Se puede observar que los **estudiantes** y el **Prof. García** pueden imprimir, pero los **administrativos** no tiene ningún derecho. Por otra parte, en la Tabla 3.3 (b) se muestran los permisos de los agentes consumidores sobre el video del curso de ingeniería de *software*. En este caso, los **estudiantes** solo pueden visualizar el video, mientras que el **Prof. Ruiz** puede descargarlo e incluso modificarlo.

Mediante estos ejemplos se ilustra claramente que un agente productor tiene la libertad de permitir o negar el uso de sus recursos como a él le convenga.

## Restricciones de uso

El sistema RAMS también permite a los agentes productores limitar el uso de sus recursos por medio de la definición de restricciones. Así, un agente productor puede definir restricciones específicas para cada agente consumidor o para un grupo. Las restricciones de uso pueden expresarse en términos de tiempo o de resultados.

Algunos recursos físicos públicos, así como una gran parte de los recursos informáticos, pueden tener restricciones de uso definidas en términos de tiempo (e.g., horario de acceso u horas máximas de uso). En la Tabla 3.3 (c) se muestran las restricciones de uso para el video del curso de ingeniería de *software*. En este caso, el Prof. Ruiz puede descargar o modificar libremente el recurso, pero los **estudiantes** sólo pueden visualizarlo en su horario de clases (lunes y jueves de 16 a 18 hrs).

Sin embargo, para otros recursos resulta más práctico definir restricciones de uso en términos de resultados. En la Tabla 3.2 (c) se puede observar que las restricciones de uso del *plotter* están definidas en términos de resultados. En este caso, el Prof. García puede utilizar el *plotter* libremente, mientras que los **estudiantes** sólo pueden imprimir un máximo de 10 páginas por mes.

### 3.4.2 Filtro de información

El filtrado es el proceso de seleccionar la información que será difundida y recibida por los agentes. Un agente productor puede definir filtros para limitar la propagación de la información que publica. De forma similar, un agente consumidor puede definir filtros que le permitan recibir sólo la información que es de su interés. Este proceso de selección de la información puede llevarse a cabo de dos formas:

- **Por tópico:** la información es publicada en canales lógicos designados. Así, un agente consumidor recibirá tanto la información publicada en los canales a los que está suscrito, como la información sobre los nuevos recursos que se publicarán en dichos canales, siempre y cuando los agentes productores le otorguen permisos de acceso.

Los agentes productores son responsables de la definición de las clases de información que corresponden a cada canal, e.g., considere los siguientes tópicos: “Impresoras”, “Escáneres” y “Videos”. Si un agente consumidor está suscrito al tópico “Impresoras”, entonces recibirá información sobre todas las impresoras publicadas (e.g., impresoras de alta y baja resolución, monocromáticas y a colores) aún cuando sólo esté buscando impresoras de alta resolución y a color.

- **Por contenido:** la información es enviada a un agente consumidor, siempre que sus requerimientos concuerden con el contenido de los atributos definidos por los agentes productores. Este tipo de filtrado es más fino, ya que la información no sólo se clasifica por tópicos, sino también se realiza un cotejo de las características provistas por los agentes productores con la petición realizada por el agente consumidor para decidir si se le envía o no la información.

Los agentes consumidores son responsables de la clasificación de eventos, e.g., si un agente consumidor especifica algunos atributos (e.g., alta resolución, color y dispositivo de salida) sobre el recurso que requiere, él recibirá información sobre impresoras de alta resolución a color, pero también sobre despliegues de alta resolución a color, pantallas de computadora, etc., ya que estos dispositivos también son de salida.

Con el objetivo de combinar las ventajas de ambos tipos de filtros, el sistema RAMS utiliza un filtrado híbrido. Por medio de este mecanismo, los agentes productores publican información por tópico y los agentes consumidores se subscriben por contenido a uno o más tópicos. En el caso de las suscripciones, primero se descartan los recursos que se encuentran en otros canales y posteriormente, se hace una búsqueda más fina sobre los canales que le interesan al agente consumidor, e.g., si un agente consumidor está suscrito al tópico “Videos” y también define el atributo “computación”, entonces sólo recibirá información sobre videos de computación.

## 3.5 Motor de inferencia

El motor de inferencia del sistema RAMS está encargado de conjuntar la información estática y la información dinámica de los recursos, con el fin de decidir si un recurso es o no adecuado para satisfacer una petición.

Los agentes productores proporcionan información relativamente estática sobre sus recursos cuando los publican en el sistema RAMS. Cabe recordar que este tipo de información comprende características técnicas, restricciones de uso y permisos otorgados a los agentes consumidores (cf. sección 3.4.1). Por su parte, las aplicaciones de usuario y los sensores del sistema RAMS proveen información dinámica sobre los recursos que han sido publicados. Este tipo de información abarca la presencia y la localización de los agentes (cf. sección 3.6).

El motor de inferencia utiliza ambos tipos de información para establecer dependencias entre recursos, las cuales le ayudan a inferir el recurso que mejor satisface una petición, e.g., el escáner de un colaborador se encuentra en su oficina y está disponible para otros siempre y cuando: a) el dueño esté en su oficina, b) el dueño esté disponible, c) el escáner esté libre y d) el horario de acceso autorice el uso del recurso.

### Dependencias entre recursos

Como se mencionó en la sección 3.2, el sistema RAMS administra tres tipos de recursos: humanos, físicos e informáticos. Así, de la clase “Recurso” heredan las clases “Recurso humano” y “Recurso no humano” (ver Figura 3.6). La entidad “Recurso no humano” es una generalización de las clases “Recurso físico” y “Recurso informático”.

Las entidades “Agente productor” y “Agente consumidor” heredan de la clase “Recurso humano”, ya que un colaborador se convierte en agente productor cuando publica alguno de sus recursos en el sistema RAMS. Por su parte, un colaborador se convierte en agente consumidor cuando se suscribe al sistema RAMS para encontrar un recurso.



La definición de dependencias entre recursos está basada en tres tipos de relaciones (ver Figura 3.6):

- la **relación de posesión**: establece una relación “m a n” entre las clases “Agente productor” y “Recurso no humano” (entidades a usar o acceder), ya que un agente productor puede poseer múltiples recursos físicos e informáticos y, a su vez, un recurso no humano puede estar asociado a varios agentes productores que fungen como responsables de dicho recurso.
- la **relación de localización** crea:
  1. una asociación “m a 1” entre las clases “Recurso humano” y “Lugar”. Un recurso humano sólo puede encontrarse en un lugar a la vez, pero un lugar puede albergar al mismo tiempo a varios recursos humanos;
  2. una asociación “m a n” entre las clases “Recurso informático” y “Lugar” (i.e., dispositivos de almacenamiento). Un recurso informático puede estar replicado en varios dispositivos de almacenamiento y, a su vez, un dispositivo de almacenamiento puede guardar varios recursos informáticos;
  3. una asociación “m a 1” entre las clases “Recurso físico” y “Lugar”. Un recurso físico generalmente está asociado a una ubicación, en la que pueden encontrarse varios recursos físicos.

La ubicación de un recurso informático o físico puede ser modificada explícitamente por el dueño o responsable de dicho recurso. Por otra parte, debido a que un colaborador puede desplazarse de un lugar a otro dentro de la organización, la ubicación de un recurso humano es capturada implícitamente por un sistema de reconocimiento de caras (cf. sección 3.6.1).

- la **relación de colaboración**: establece una asociación “m a n” entre las clases “Recurso humano” y “Grupo”, ya que un recurso humano puede formar parte de varias agrupaciones, e.g., un investigador puede pertenecer al grupo de profesores de sistemas distribuidos y al mismo tiempo ser parte del colegio de profesores. Por su parte, un grupo está compuesto por varios recursos humanos.

Cada “Recurso humano” tiene una “Agenda de trabajo”, la cual sirve como guía para saber si es probable encontrar a un colaborador dentro de la organización.

Por su parte, las clases “Recurso no humano” y “Operación” tienen una relación “m a n”. Una operación es una acción sintáctica y semánticamente correcta que se puede realizar sobre un recurso (e.g., acceder a un video, descargar un *software* y usar un escáner). Cada recurso no humano está asociado a un conjunto de operaciones y, a su vez, una operación puede estar definida en varios recursos no humanos (e.g., se puede descargar un antivirus y también, es posible descargar un video de una clase).

Entre las clases “Recurso no humano” y “Restricción de uso” se establece una relación “m a n”, ya que un recurso no humano puede tener una o varias restricciones que limiten

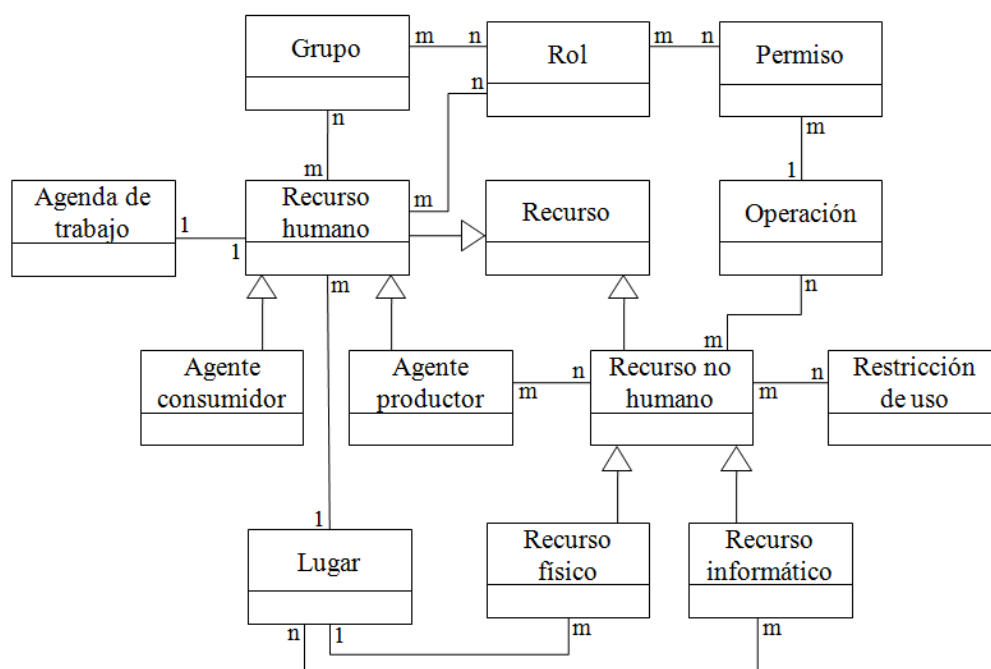


Figura 3.6: Diagrama de clases de las entidades del sistema RAMS

su uso o acceso. A su vez, una restricción de uso puede estar definida en varios recursos no humanos.

Las clases “Permiso” y “Operación” tienen una relación “m a 1”, ya que un permiso se refiere a la autorización para realizar una operación sobre un recurso (e.g., modificar un documento compartido). Por su parte, una operación puede estar definida en uno o varios permisos.

Entre las clases “Rol” y “Permiso” existe una relación “m a n”. Un rol puede estar asociado a uno o varios permisos y, a su vez, un permiso puede existir en diferentes roles e.g., el rol de *investigador* tiene los permisos de *acceder* y *modificar* un documento HTML, mientras que el rol de *estudiante* sólo puede *visualizarlo*.

Las clases “Recurso humano” y “Rol” tienen una relación “m a n”, ya que un agente puede tener distintos roles asignados e.g., un colaborador puede jugar el rol de *profesor* en el Departamento de Computación y el rol de *coordinador académico* en la Secretaría Académica. A su vez, un rol puede estar asignado a varios colaboradores.

De forma similar, se establece una relación “m a n” entre las clases “Grupo” y “Rol”, ya que un grupo de colaboradores puede jugar un rol en un momento dado y jugar otro rol en otro momento (e.g., un grupo de colaboradores del Departamento de Computación tiene el rol de *investigador*, pero cuando ellos se reúnen para decidir los cursos que se impartirán el siguiente año, el grupo juega el rol de *profesor*). A su vez, un mismo rol puede corresponder a varios grupos de colaboradores (e.g., aunque los grupos de las materias de programación y de arquitecturas de computadoras tienen distintos miembros, ambas agrupaciones juegan el rol de *estudiante*).

## 3.6 Sensores

El sistema RAMS utiliza un conjunto de sensores que capturan información dinámica, la cual permite determinar la disponibilidad de los recursos. Los sensores propuestos ayudan a automatizar algunas actividades que, si se delegaran a los usuarios del sistema, podrían resultar repetitivas e incluso molestas. Este conjunto de sensores se compone de: a) un sistema de reconocimiento de caras y b) un sistema de localización del recurso más cercano.

### 3.6.1 Sistema de reconocimiento de caras

El sistema RAMS se centra en la localización de los colaboradores, con el fin de acercarlos la información. Para determinar su ubicación, se propone un sistema de reconocimiento de caras, el cual tiene la finalidad de identificar a los colaboradores (en vez de rastrear a sus dispositivos móviles) ya que nada garantiza que un colaborador se encuentre en el mismo lugar que su dispositivo. Un usuario puede olvidarlo en el baño o dentro de su oficina, lo cual provocaría que el sistema RAMS ofrezca información equivocada sobre su ubicación. Gracias a este sistema de reconocimiento de caras, dichos descuidos pierden relevancia porque es imposible que una persona olvide su cara.

Nuestra propuesta no pretende negar el interés real de localizar dispositivos móviles mediante cualquier algoritmo e.g., triangulación de señales WiFi [Markarian et al., 2006]. Desde nuestro punto de vista, este enfoque constituye una forma complementaria de localizar dispositivos que podría ser útil en otras situaciones (e.g., cuando es necesario que un usuario reciba, en su dispositivo móvil, información de acuerdo a su ubicación). Por el contrario, nuestra propuesta pretende resaltar las ventajas que un sistema de reconocimiento de caras puede proveer al trabajo cooperativo móvil.

El sistema de reconocimiento de caras se compone de dos fases: una de aprendizaje y una de prueba o funcionamiento. La fase de aprendizaje se realiza sólo una vez, mientras que la fase de prueba se realiza cada vez que una cara humana es capturada por una cámara (ver Figura 3.7).

La fase de aprendizaje utiliza un algoritmo que ayuda a diferenciar una cara humana del resto de los objetos captados por una cámara. Este algoritmo además permite construir una base de datos (ver Figura 3.7 ref #1), la cual está compuesta de un conjunto de imágenes de cada colaborador que se desea reconocer. Cada conjunto de imágenes contiene fotografías de la cara de cada colaborador en diversas poses y gesticulaciones.

Posteriormente, la base de datos es analizada mediante un método matemático llamado *Eigenfaces* [Hernandez et al., 2007]. Este método permite obtener las características más significativas que diferencian a un colaborador de otro. Los resultados de este análisis matemático permiten crear un modelo de clasificación (ver Figura 3.7 ref #2), el cual ayudará a identificar a qué colaborador pertenecen las imágenes obtenidas durante la fase de prueba. Dichas imágenes serán capturadas por cámaras localizadas en lugares específicos.

La fase de prueba inicia cuando el algoritmo, que diferencia caras humanas de otros

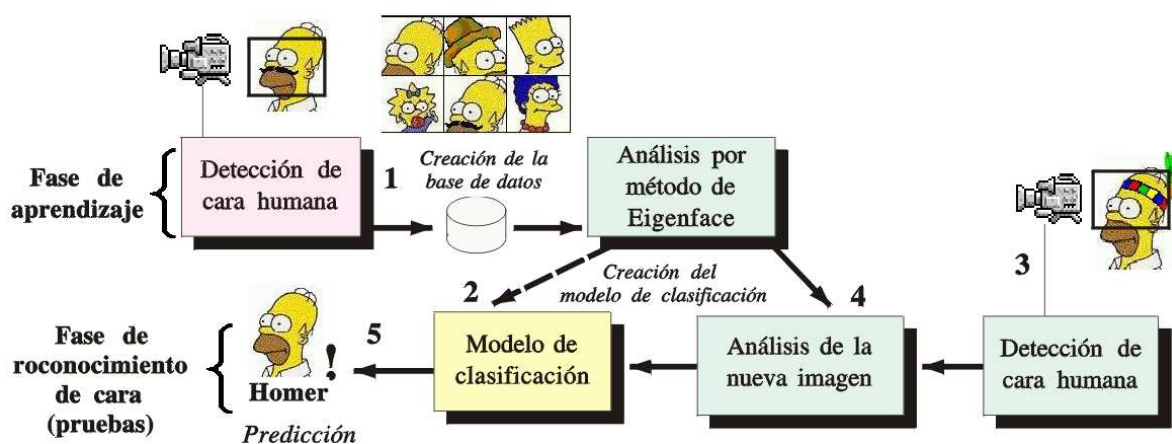


Figura 3.7: Fases de aprendizaje y prueba del sistema de reconocimiento de caras

objetos, detecta una cara humana y captura una imagen (ver Figura 3.7 ref #3). Esta imagen es analizada tomando en cuenta tanto la información que se obtuvo del análisis de las imágenes de la base de datos, como la información de la imagen recién capturada (ver Figura 3.7 ref #4). Mediante este análisis se obtienen las características principales de la nueva imagen, las cuales son evaluadas por el modelo de clasificación que se creó en la fase de aprendizaje (ver Figura 3.7 ref #5). Este modelo se encarga de hacer una predicción sobre la identidad del colaborador al que se le tomó dicha fotografía.

No obstante que este sistema de reconocimiento de caras tiene como objetivo informar acerca de la presencia y de la localización de los colaboradores, es importante considerar el problema de privacidad. Así, el sistema RAMS también permite que los colaboradores se declaren invisibles para todos o solo algunos de sus colegas. De esta manera, un colaborador puede ocultar su presencia dentro del ambiente cooperativo.

### 3.6.2 Sistema de localización del recurso más cercano

Cuando un agente consumidor hace una petición, particularmente de recursos físicos, el sistema RAMS puede encontrar más de un recurso que satisfaga las características técnicas especificadas en su petición. Para determinar cuál de esos recursos se encuentra más cerca del agente consumidor, se propone un sensor de localización.

El sensor de localización del recurso más cercano toma en cuenta: 1) la ubicación de los recursos que satisfacen los requerimientos técnicos de una petición, 2) la ubicación del agente consumidor que realiza la petición e 3) información sobre las distancias que existen entre las oficinas de la organización donde se encuentran los recursos.

La información sobre la ubicación de los recursos es proporcionada por los agentes productores cuando publican sus recursos en el sistema RAMS. Por su parte, la información sobre la ubicación del solicitante del recurso se obtiene del sensor de reconocimiento de caras. Por último, la información sobre las distancias entre las oficinas de la organización se conoce desde que se instala el sistema RAMS en dicha organización.

Una vez que se obtiene tanto la ubicación de cada uno de los recursos que satisfacen la petición de un agente consumidor, como la ubicación de dicho agente, el sensor decidirá cuál es el recurso más cercano. Para tomar esta decisión, el sensor evalúa ambas ubicaciones con respecto a la información sobre la distancia en metros que existe entre las oficinas de la organización.

En el siguiente escenario se describe la utilidad del sensor de localización del recurso más cercano.

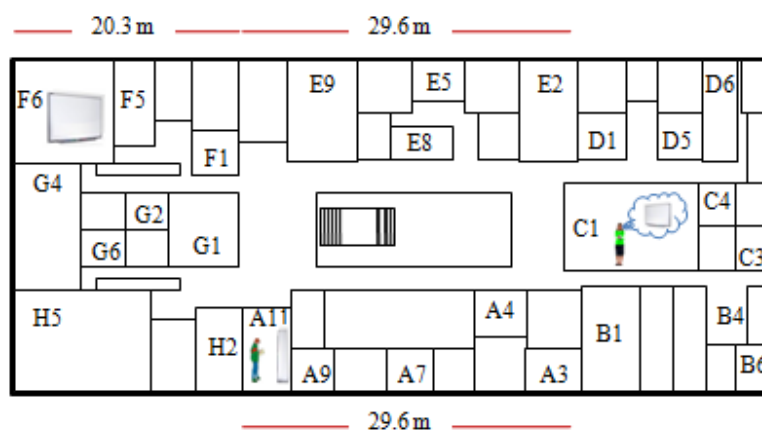


Figura 3.8: Mapa de la planta alta del Departamento de Computación

Dentro del edificio del Departamento de Computación del CINVESTAV (precisamente en la planta alta) se encuentran múltiples recursos físicos distribuidos.

En un momento dado, una estudiante de maestría, la Srita. Mendez, necesita utilizar un pizarrón interactivo para probar una tarea, así que utiliza el sistema RAMS para encontrar un dispositivo que satisfaga sus necesidades. El sistema RAMS encuentra dos pizarrones interactivos. Para determinar qué pizarrón está más cerca de la Srita. Mendez, el sistema RAMS utiliza la información del sensor de reconocimiento de caras, el cual detectó que ella está en la oficina *C1* (ver Figura 3.8). Esta información se envía al sensor de localización del recurso más cercano, el cual determina que el pizarrón interactivo más cercano está ubicado a menos de 30 metros de la estudiante (en la oficina *A11*), mientras que el otro pizarrón se encuentra a 50 metros aproximadamente de ella (en la oficina *F6*). Esta información se transmite al motor de inferencia, el cual determina que el pizarrón de la oficina *A11* está siendo utilizado por otro estudiante. Así, el sistema RAMS notifica a la Srita. Mendez que el pizarrón de la sala *F6* está disponible en ese momento, ya que el pizarrón más próximo está ocupado.

### 3.7 Sistema de notificación

Una vez que el motor de inferencia evaluó los requerimientos de un agente consumidor y tomó una decisión sobre el recurso que mejor satisface dichos requerimientos, el sistema RAMS se encarga de dar a conocer esta decisión por medio del sistema de notificación. A continuación se describen las condiciones que deben cumplirse para que el sistema RAMS proporcione la respuesta adecuada.

#### 3.7.1 Información de estado de los recursos físicos

Un recurso físico publicado en el sistema RAMS será **perceptible** (ver Figura 3.9 estado #2) por un agente consumidor, si las siguientes condiciones son verdaderas:

- el agente consumidor tiene permisos de acceder o utilizar el recurso (condición  $C_1$ ), y
- las características técnicas del recurso satisfacen los requerimientos del agente consumidor (condición  $C_2$ ).

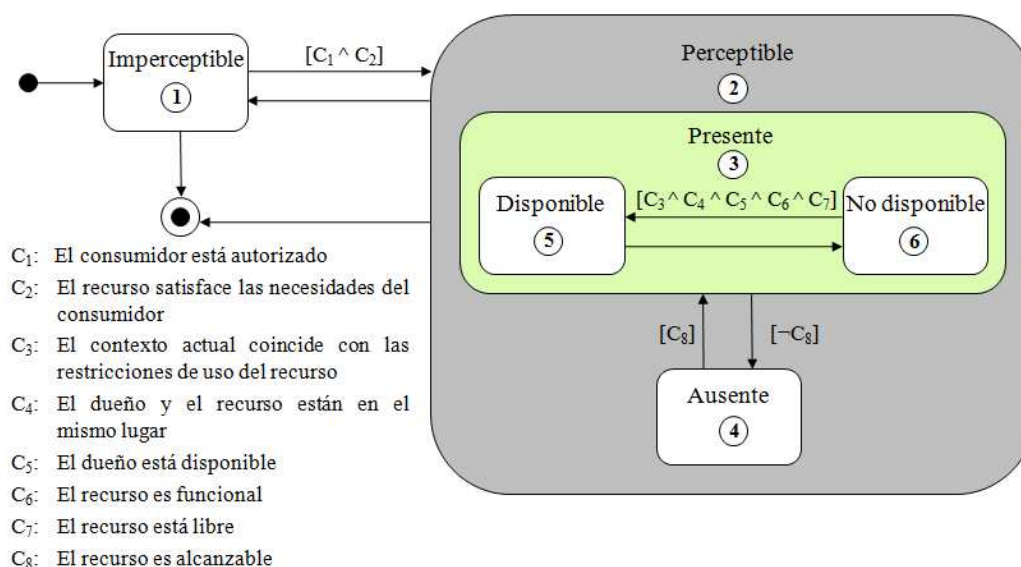


Figura 3.9: Diagrama de estado de un recurso físico

Si alguna de estas condiciones no se cumple, el recurso será **imperceptible** (ver Figura 3.9 estado #1) por dicho agente consumidor.

Cuando el recurso es **perceptible**, el sistema RAMS utiliza funciones de red (e.g., en el caso de una impresora en red) o eventos de aplicaciones de usuario (e.g., en el caso de una sala de juntas) para determinar si el recurso está presente o no. Así, un recurso físico puede encontrarse en alguno de los siguientes subestados:

- a) **ausente** (ver Figura 3.9 estado #4) cuando el recurso es inalcanzable, e.g., la impresora está fuera de línea por reparación o la sala de juntas está cerrada por mantenimiento (condición  $C_8$ );
- b) **presente** (ver Figura 3.9 estado #3) cuando el recurso es alcanzable, e.g., la impresora está en línea o la sala de juntas puede accederse (condición  $\neg C_8$ ).

Cuando un recurso se encuentra en el subestado **presente**, el sistema RAMS es capaz de informar si el recurso está disponible o no para un agente consumidor. Dado que algunos recursos físicos pueden ser utilizados o accedidos remotamente (e.g., un *cluster*), su disponibilidad depende principalmente de las restricciones de uso definidas por los agentes productores.

Por otra parte, cuando un recurso se encuentra en un área restringida (e.g., una oficina), puede ser necesaria la presencia de su dueño para accederlo. Consecuentemente, el motor de inferencia toma en cuenta información de ambas entidades (recurso y dueño) para determinar la disponibilidad de dicho recurso. Así, el sistema RAMS notificará al agente consumidor que el recurso está **disponible** (ver Figura 3.9 estado #5), si las siguientes condiciones son verdaderas:

- el contexto actual coincide con las restricciones de uso definidas por el agente productor al publicar el recurso, e.g., la hora actual corresponde con el horario que el dueño estipuló para autorizar el uso de su impresora (condición  $C_3$ );
- el dueño y el recurso se encuentran en el mismo lugar (condición  $C_4$ ),
- el dueño está disponible (condición  $C_5$ ),
- el recurso es funcional, e.g., la impresora está encendida y conectada a la red y además tiene hojas y tinta suficiente (condición  $C_6$ ), y
- el recurso no está siendo utilizado por ningún otro colaborador (condición  $C_7$ ).

La información referente a las condiciones  $C_3$ ,  $C_4$  y  $C_7$  es capturada implícitamente por el sistema RAMS, mientras que la información relativa a la condición  $C_5$  es capturada explícitamente por los agentes productores.

Por su parte, la información correspondiente a la condición  $C_6$  puede adquirirse tanto implícita como explícitamente. Algunos recursos físicos proveen información sobre sus condiciones actuales de forma implícita, e.g., una impresora en red será capaz de enviar un aviso cuando carezca de tóner. Sin embargo, otros recursos físicos no tienen esta capacidad, por lo tanto es necesario que el agente productor proporcione dicha información de forma explícita, e.g., cuando el proyector de una sala de juntas esté en mantenimiento; en este caso, aunque la sala de juntas es alcanzable, podría no ser complementamente funcional porque uno de sus componentes está fuera de servicio. Así, el sistema RAMS permite a los agentes productores: a) indicar su propia disponibilidad y b) declarar sus recursos no funcionales.

Si alguna de las cinco condiciones anteriores es falsa, el sistema RAMS notificará al agente consumidor que el estado del recurso es **no disponible** (ver Figura 3.9 estado #6). En el siguiente escenario, se describe cómo el sistema RAMS es capaz de inferir el estado de un recurso, evaluando las condiciones descritas anteriormente:

Suponga que el Prof. Ruiz posee una cámara digital que guarda en su oficina. En un momento dado, el sensor de reconocimiento de caras detecta que el Prof. Ruiz se encuentra en su oficina. Además, por medio de aplicaciones de usuario, él se declara disponible. A partir de esta información, el sistema RAMS determina que su cámara digital también está disponible e informa a los agentes consumidores interesados en este estado. Más tarde, el Prof. Ruiz se dirige al cubículo de una secretaria, así que el sistema de reconocimiento de caras detecta su nueva ubicación. Consecuentemente, el motor de inferencia deduce que el nuevo estado de la cámara digital es no disponible. Esta información es enviada al sistema de notificación, el cual se encarga de difundirla a los agentes consumidores interesados.

El motor de inferencia también implementa un método más sofisticado, que infiere la disponibilidad de los recursos en base a las relaciones de colaboración entre los usuarios. En el siguiente escenario se muestra cómo dichas relaciones hacen posible que un agente, que no es el dueño del recurso, permita utilizarlo:

De acuerdo con el escenario descrito en la Sección 3.1, suponga que el Prof. Ruiz está ausente, pero la Srita. López se declara disponible mientras trabaja en la oficina del Prof. Ruiz. En consecuencia, los recursos que pertenecen a este último pueden ser accedidos por los colegas que en ese momento tengan permisos.

### 3.7.2 Información de estado de los recursos humanos

Un recurso humano es **peceptible** (ver Figura 3.10 estado #2) por un agente consumidor, si se cumplen las siguientes condiciones:

- ambos colaboradores, el observador y el observado, tienen una relación de colaboración, e.g., son miembros de un grupo de trabajo (condición  $C_1$ ), y
- el colaborador observador otorgó permisos sobre sus recursos al colaborador observado (condición  $C_2$ ), o
- el colaborador observador tiene derecho de percibir al colaborador observado (condición  $C_3$ ).

Si ninguna de estas condiciones se cumple, el colaborador será **imperceptible** (ver Figura 3.10 estado #1) por su colega.

Cuando un colaborador es **perceptible** por otro, el sistema RAMS además puede ofrecer a este último (observador) información sobre la presencia o ausencia del primero (observado) en la organización. Así, un recurso humano puede encontrarse en alguno de los siguientes subestados:



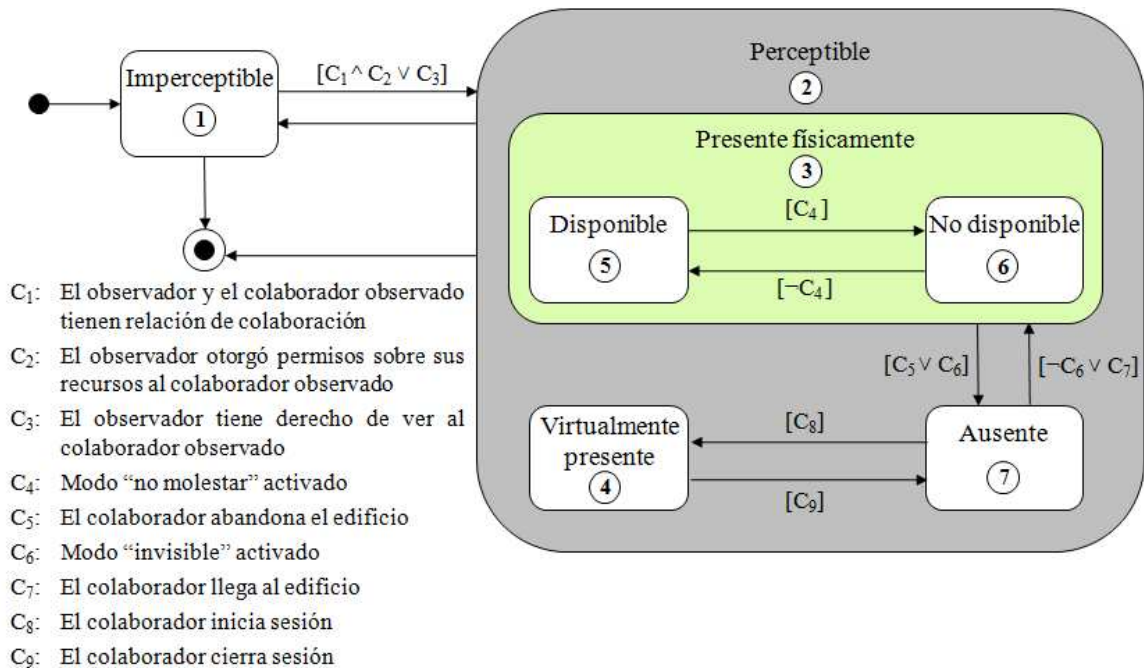


Figura 3.10: Diagrama de estado de un recurso humano

- **ausente** (ver Figura 3.10 estado #7) si el colaborador observado: a) es detectado por el sistema de reconocimiento de caras cuando está abandonando el edificio (condición  $C_5$ ) o b) activa explícitamente el modo “invisible” con el fin de ocultar su presencia física (condición  $C_6$ ) o c) cierra la sesión en curso para salir del sistema RAMS (condición  $C_9$ );
- **presente físicamente** (ver Figura 3.10 estado #3) si el colaborador observado: a) desactiva explícitamente el modo “invisible” (condición  $\neg C_6$ ) o b) es detectado por el sistema de reconocimiento de caras cuando está llegando al edificio (condición  $C_7$ ).
- **virtualmente presente** (ver Figura 3.10 estado #4) cuando un colaborador, que no se encuentra en la organización, inicia remotamente una sesión en el sistema RAMS (condición  $C_8$ ). De esta forma, sus colegas pueden ponerse en contacto con él, por medio de herramientas dedicadas que les ayuden a establecer una sesión de comunicación o colaboración.

Cuando un recurso humano está **presente físicamente** en la organización, el sistema RAMS le permite declararse:

- **no disponible** (ver Figura 3.10 estado #6) si el colaborador observado activa de forma explícita el modo “no molestar” (condición  $C_4$ );
- **disponible** (ver Figura 3.10 estado #5) si el colaborador observado desactiva el modo “no molestar” (condición  $\neg C_4$ ). Adicionalmente, el sistema RAMS le permite

declararse disponible para algunos de sus colegas (e.g., jefe) pero no disponible para otros (e.g., colegas).

### 3.7.3 Información de estado de los recursos informáticos

Para poder utilizar o acceder a un recurso informático, el agente consumidor no requiere la presencia del agente productor en el lugar donde se encuentra, ya que a diferencia de los recursos físicos, los recursos informáticos pueden utilizarse o accederse remotamente.

Un recurso informático es **perceptible** (ver Figura 3.11 estado #2) por un agente consumidor, si las siguientes condiciones son verdaderas:

- el agente consumidor tiene permisos de acceder al recurso (condición  $C_1$ ), y
- las características técnicas del recurso satisfacen los requerimientos del agente consumidor (condición  $C_2$ ).

Si alguna de las condiciones anteriores no se cumple, el recurso será **imperceptible** (ver Figura 3.11 estado #1) por el agente consumidor.

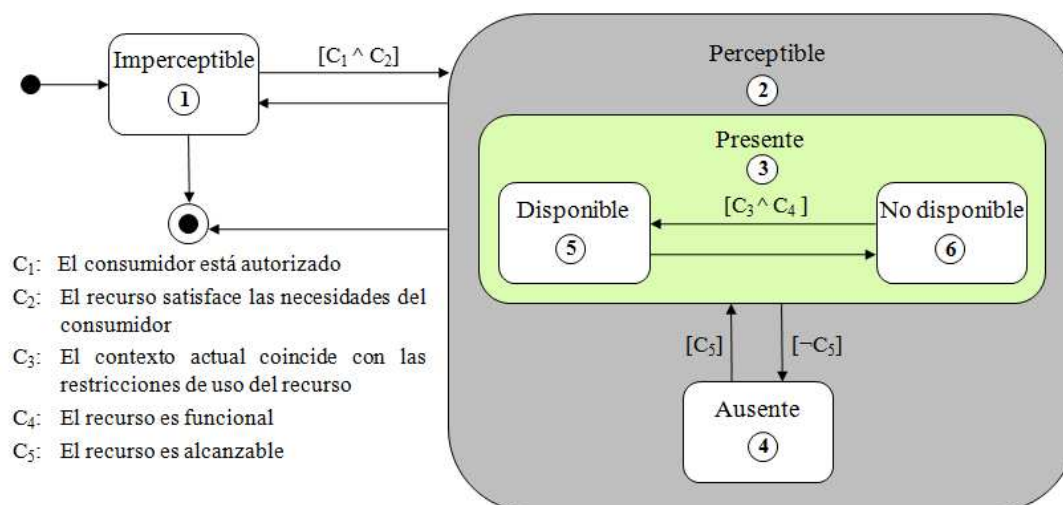


Figura 3.11: Diagrama de estado de un recurso informático

Un recurso informático **perceptible** puede encontrarse en alguno de los siguientes subestados:

- **ausente** (ver Figura 3.11 estado #4) si el recurso es inalcanzable, e.g., una página Web es inaccesible porque está siendo actualizada (condición  $\neg C_5$ );
- **presente** (ver Figura 3.11 estado #3) si el recurso es alcanzable, e.g., el sitio que lo almacena puede accederse (condición  $C_5$ ).

Cuando se determina que un recurso está **presente**, el agente consumidor podrá percibirlo como **disponible** (ver Figura 3.11 estado #5) si las siguientes condiciones son verdaderas:

- a) las restricciones de uso permiten que el agente consumidor utilice o acceda al recurso (condición  $C_3$ );
- b) el recurso es funcional (condición  $C_4$ ).

Si alguna de las condiciones anteriores no se cumple, entonces el recurso informático estará **no disponible** (ver Figura 3.11 estado #6) para el agente consumidor.

### 3.8 Esquema de distribución

En esta sección se describe el esquema de distribución del sistema RAMS. Este esquema tiene el objetivo de distribuir los diferentes módulos de un sistema, de manera que se obtenga un balance entre la cantidad de mensajes enviados por la red y el aseguramiento de las propiedades que se requiere ofrecer, e.g., disponibilidad de información o tolerancia a fallas. El esquema de distribución del sistema RAMS comprende cuatro tipos de sitios que albergan diferentes componentes. Existen dos tipos de sitios que juegan el rol de cliente:

- el sitio de tipo A aloja al módulo de publicación y a un filtro de información por tópico (ver Figura 3.12 ref A);
- el sitio de tipo B alberga al módulo de suscripción y a un filtro de información por contenido (ver Figura 3.12 ref B).

Puede existir varios sitios de los tipos anteriores, ya que cada agente productor y cada agente consumidor tendrán acceso a un sitio que les permitirá respectivamente publicar alguno de sus recursos (sitio de tipo A) y suscribirse para hacer una petición (sitio de tipo B).

Por otra parte, existen dos tipos de sitios que juegan el rol de servidor:

- el sitio de tipo C mantiene los siguientes módulos: un filtro de información por tópico, un filtro de información por contenido, el motor de inferencia, el sistema de localización del recurso más cercano y el sistema de notificación. Adicionalmente, este tipo de sitio almacena información sobre: los recursos publicados, las peticiones y los eventos que se generan a través de aplicaciones de usuario (ver Figura 3.12 ref C).

En el caso de una organización que consta de un solo edificio, podría colocarse un solo servidor de tipo C, sin considerar por supuesto la propiedad de tolerancia a fallas. Sin embargo, si el sistema RAMS es implantado en una organización que cuenta con varios edificios, podría ser necesario colocar un sitio de tipo C en cada edificio. La replicación de los componentes albergados en este tipo de sitio es necesaria sobre

todo cuando los edificios están físicamente distantes, como en el caso de implantar el sistema RAMS en los departamentos de computación del CINVESTAV y del CICESE.

- el sitio de tipo D almacena el sistema de reconocimiento de caras y la información que produce (ver Figura 3.12 ref D). Puede existir varios sitios de este tipo, uno por cada cámara ubicada en la organización.

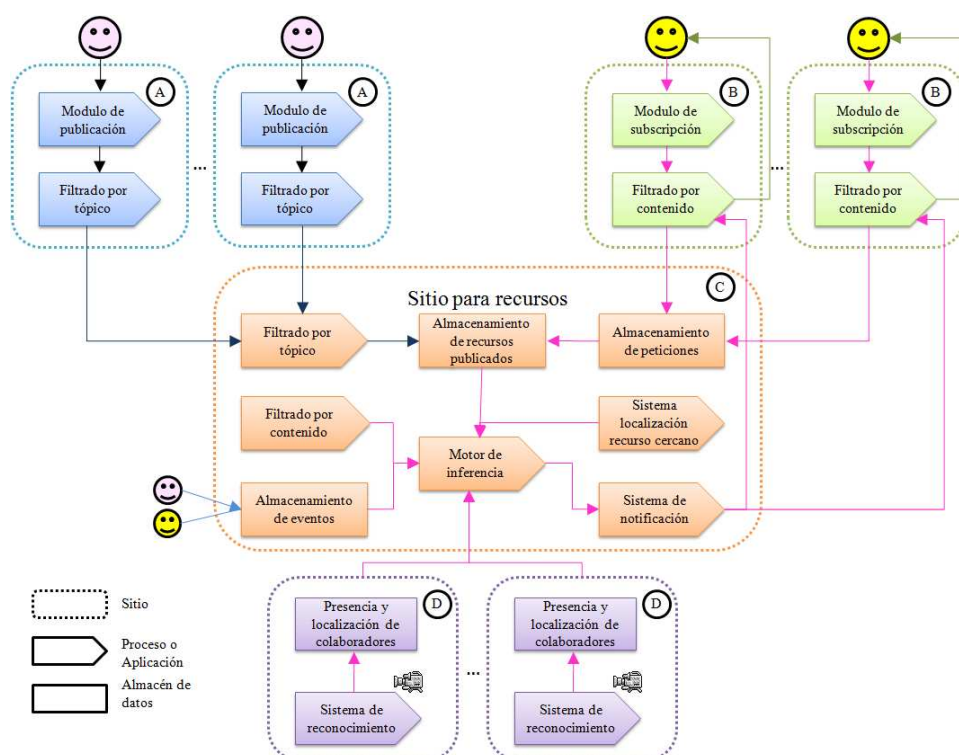


Figura 3.12: Arquitectura de distribución del sistema RAMS

El módulo de publicación (sitio de tipo A) permite que un agente productor describa alguno de sus recursos en términos de sus capacidades o características técnicas, con la finalidad de compartirlo con sus colegas. La información que proporciona dicho agente pasa por un módulo de filtrado por tópico para determinar el tipo de recurso al que pertenece (i.e., recurso físico, humano o informático). Posteriormente, esta información se envía a un servidor (sitio de tipo C), donde vuelve a pasar por un filtro de información por tópico, esta vez con el propósito de clasificar el recurso en el canal o canales correspondientes.

Cuando un agente consumidor necesita utilizar un recurso, accede al módulo de suscripción (sitio de tipo B), el cual le permite describir sus requerimientos. La petición del agente consumidor se envía primero a un filtro de información por tópico y después a un sitio de tipo C, donde será almacenada y eventualmente procesada. El procesamiento consiste en filtrar por contenido la información de los recursos que han sido publicados, con el objetivo de seleccionar solo aquellos que cumplan los requerimientos del agente

consumidor. La información filtrada de estos recursos se envía al motor de inferencia, el cual utiliza el sistema de reconocimiento de caras (sitio de tipo D), el sistema de localización del recurso más cercano y los eventos de aplicación para obtener la disponibilidad de dichos recursos. A partir de esta información, el motor de inferencia es capaz de tomar una decisión sobre el recurso que mejor satisface la petición del agente consumidor. Esta decisión se envía al sistema de notificación, el cual la transmite al filtro (sitio de tipo B) que el agente consumidor activó. De esta forma, él sólo recibirá las notificaciones que le interesan.



## Capítulo 4

# Implementación del sistema RAMS

En este capítulo, se describen los módulos del sistema RAMS que han sido implementados. Como se mencionó en el capítulo anterior, su arquitectura está compuesta de cuatro subsistemas principales: a) un *broker*, b) un motor de inferencia, c) un conjunto de sensores y d) un sistema de notificación. El *broker* contiene los módulos de publicación y de suscripción que permiten respectivamente, a los agentes productores y consumidores, describir recursos ya sea para compartirlos o para buscar aquellos que necesitan utilizar o acceder. Esta descripción de recursos requiere de estructuras de datos, las cuales han sido implementadas mediante documentos XML (sección 4.1). Las etiquetas que conforman estos documentos permiten representar adecuadamente las características relevantes de los recursos que son administrados por el sistema RAMS.

Asimismo, se describe la implementación inicial del motor de inferencia, el cual por ahora sólo utiliza información estática para tomar una decisión sobre el recurso que mejor satisface los requerimientos de un agente consumidor (sección 4.2). Por último, se detalla la implementación del sistema de reconocimiento de caras, la cual ha sido dividida en dos fases: aprendizaje y prueba (sección 4.3). En la fase de aprendizaje, este sistema crea un conjunto de entrenamiento, a partir de imágenes de los colaboradores que se requiere identificar; posteriormente, este sistema analiza dicho conjunto de entrenamiento para obtener las características principales de los colaboradores y crea finalmente un modelo de clasificación. Por su parte, en la fase de prueba, este sistema analiza una imagen recién tomada y realiza una predicción sobre la identidad del colaborador al que pertenece dicha imagen.

### 4.1 Descripción de recursos

El sistema RAMS tiene como objetivo administrar el uso compartido de recursos entre las personas que trabajan en una organización, donde pueden existir múltiples recursos físicos e informáticos heterogéneos. El sistema RAMS utiliza el modelo asíncrono de publicación y suscripción, el cual hace posible que los colaboradores: a) publiquen sus recursos para compartirlos con sus colegas y b) se suscriban con la intención de buscar un recurso que satisfaga sus necesidades.

El proceso de publicación de un recurso consiste en posibilitar que un agente productor: 1) describa las características técnicas del recurso que desea compartir, 2) defina las restricciones de uso o acceso del recurso y 3) atribuya permisos a los colaboradores con quienes desea compartir el recurso. Por su parte, el proceso de suscripción consiste en propiciar que un agente consumidor describa las características técnicas del recurso que necesita.

### 4.1.1 Herramienta para describir recursos

Como se mencionó anteriormente, la descripción de recursos se realiza tanto en el proceso de publicación como en el proceso de suscripción. Por esta razón, se requiere una herramienta que pueda emplearse en ambos procesos. XML (*Extensible Markup Language*) es un lenguaje de etiquetas que, gracias a su flexibilidad, puede ser utilizado para describir de forma sencilla distintos tipos de recursos. Algunas características de esta herramienta pueden ser explotadas en la implementación de los módulos de publicación y de suscripción. Particularmente, estas características se refieren a la facilidad que ofrecen los documentos XML para: 1) construir una gran variedad de aplicaciones, 2) ser legibles por cualquier persona y 3) ser diseñados de forma fácil y rápida.

Conforme a la implementación actual, cuando un colaborador desea publicar un recurso, el módulo correspondiente le solicitará un conjunto de características predefinidas, las cuales han sido seleccionadas en base a los tipos y subtipos de recursos soportados por el sistema RAMS (cf. sección 3.2). Las propiedades de facilidad y rapidez en el diseño de documentos XML también favorecen la incorporación de nuevas estructuras de datos cuando se requiera publicar un nuevo tipo o subtipo de recurso.

La implementación de los módulos de publicación y de suscripción ha sido realizada en C++ utilizando la biblioteca Xerces-C++ de Apache, la cual permite compilar, generar, manipular y validar documentos XML en lenguaje C++. Esta biblioteca soporta los APIs: DOM, SAX y SAX2. El API que resultó más conveniente para el desarrollo de estos módulos del sistema RAMS es XML DOM (*Document Object Model*), ya que define una forma estándar de acceder y manipular documentos XML. Este API organiza dichos documentos en una estructura de árbol, cuyos nodos pueden ser elementos, atributos y texto.

Como se observa en la Figura 4.1, XML DOM crea un esqueleto jerárquico donde existen nodos padre e hijo. En este caso, existe un elemento raíz llamado “recursos”, del cual se desprende el nodo “recurso humano” que, a su vez, está compuesto de otros elementos (i.e., nombre, afiliación y oficina). Cada elemento del árbol es un nodo.

### 4.1.2 Descripción de recursos físicos

Los recursos físicos poseen múltiples características técnicas, pero no todas son útiles porque, debido a su particularización, distan mucho de pertenecer al dominio público. Con la intención de obtener sólo información significativa, se han escogido algunas de las características más comunes de cada subtipo de recurso físico. Estas características han



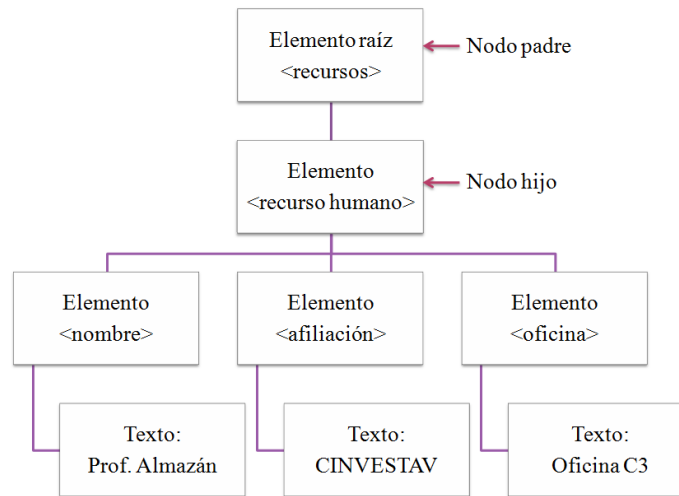


Figura 4.1: Estructura jerárquica de un documento XML DOM

sido seleccionadas mediante un sondeo que recogió las opiniones vertidas, por alumnos de computación y personas ajenas al campo, sobre los atributos que se buscarían en cada uno de los subtipos de recursos físicos contemplados por el sistema RAMS.

Por medio del módulo de publicación, el agente productor puede definir roles con el fin de atribuirlos a los agentes consumidores con quienes desea compartir su recurso. Asimismo, el agente productor puede asociar, a cada rol, un conjunto de restricciones de uso o acceso del recurso.

A continuación se describen algunos ejemplos que muestran el tipo de información que un agente productor debe proporcionar, cuando desea publicar un recurso físico en el sistema RAMS. La Figura 4.2 representa la descripción de una impresora mediante un árbol, de cuyo nodo raíz “impresora” se desprenden tres categorías principales de datos:

- **Características técnicas:** son proporcionadas por el agente productor al publicar un recurso en el sistema RAMS. En el caso de una impresora, estas características son: marca, modelo, tecnología (e.g., laser, inyección de tinta o matriz de punto), color y tipo de conexión. Estos campos proveen información técnica del dispositivo que no es demasiado especializada;
- **Datos del dueño:** incluyen el nombre y la ubicación del agente productor. En el caso de una impresora, el sistema RAMS infiere que su ubicación es la misma que la de su dueño. Adicionalmente, el agente productor puede definir colaboradores asociados, que son colegas con derecho de permitir el acceso a un recurso físico, aún cuando el dueño no esté presente (e.g., un alumno puede permitir que la impresora de su asesor sea utilizada, aunque este último no se encuentre en su oficina);
- **Roles:** el agente productor puede definir roles asociando, a cada rol, un nombre, permisos y restricciones de uso. En el caso de una impresora, el agente productor

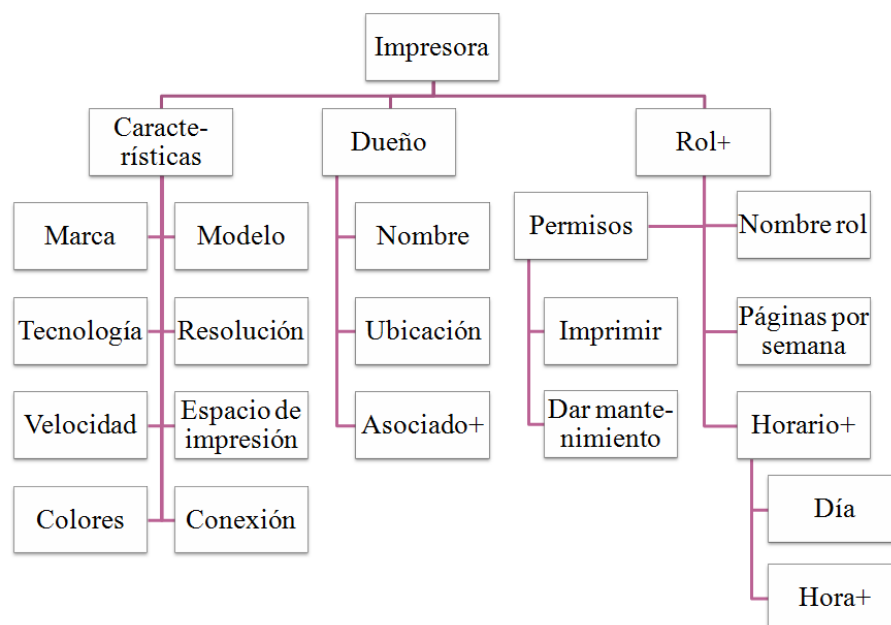


Figura 4.2: Descripción en XML de una impresora

puede asignar los permisos de `imprimir` y `dar mantenimiento` y definir sus restricciones de uso en términos de resultados, e.g., número de páginas impresas por semana. Además, el agente productor puede establecer el horario en el que cada rol podrá utilizar la impresora.

Todo elemento que termina con el signo “+” indica que puede existir uno o más elementos de ese tipo (e.g., rol).

A partir de las descripciones 4.2 y 4.3, se aprecia que las características técnicas difieren de un subtipo de recurso físico a otro. Asimismo, las restricciones de uso asociadas a cada rol son distintas. En el caso de un escáner, resulta más adecuado definir sus restricciones de uso en términos del horario en el que puede ser utilizado. De forma similar a la descripción de una impresora, el agente productor puede definir colaboradores asociados, quienes podrían permitir el uso del escáner cuando el dueño esté ausente.

Por último, la Figura 4.4 presenta la descripción de una sala de juntas. A diferencia de otros recursos físicos, una sala de juntas no tiene necesariamente un dueño, sino un responsable o administrador que se encarga de permitir el acceso al lugar (e.g., un intendente). Entre las características de una sala de juntas se encuentran la capacidad y el equipo con el que cuenta (e.g., proyector, pizarrón y equipo de videoconferencia). Adicionalmente, la descripción de este recurso contiene el nombre y la ubicación del responsable y de los colaboradores asociados. Al igual que otros recursos físicos, el responsable puede definir roles con el fin de asignarlos a los potenciales agentes consumidores (e.g., esta funcionalidad es útil en el caso de un profesor que requiere utilizar la sala todos los viernes para impartir su curso). Además cada rol puede estar asociado a permisos y restricciones de uso, las cuales se definen en términos de tiempo para este recurso.

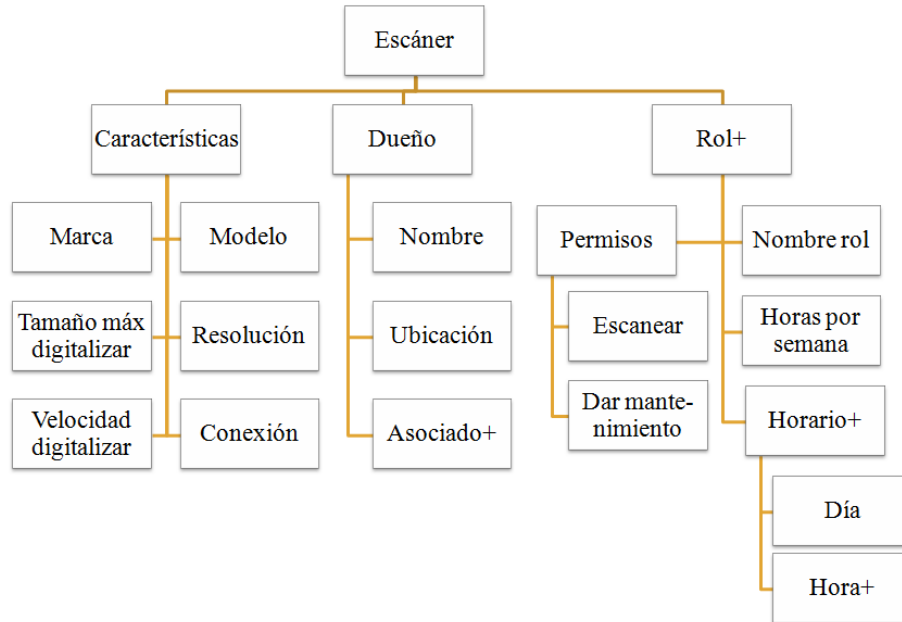


Figura 4.3: Descripción en XML de un escáner

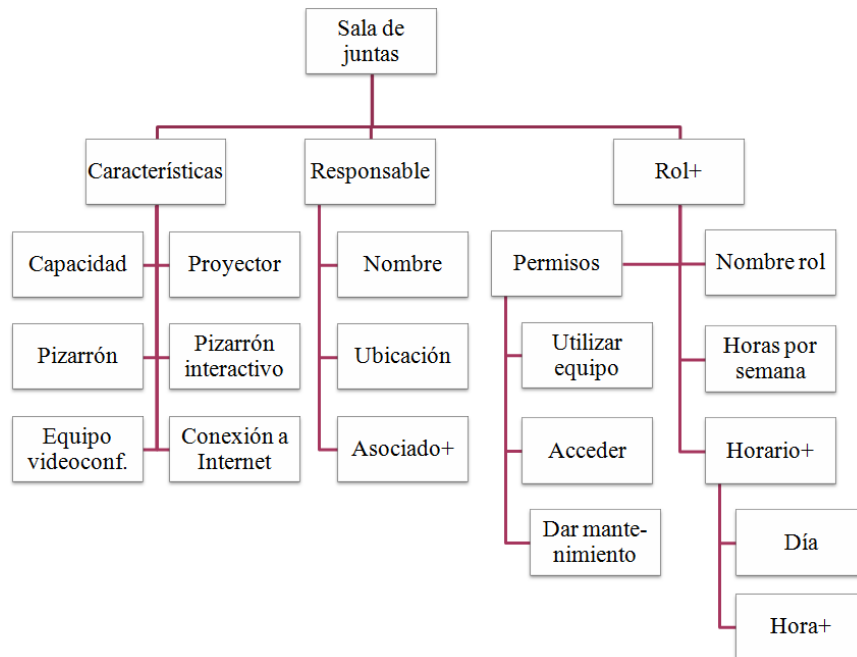


Figura 4.4: Descripción en XML de una sala de juntas

### 4.1.3 Descripción de recursos informáticos

De forma similar a la publicación de un recurso físico, un agente productor puede describir alguno de sus recursos informáticos, con el fin de compartirlo con sus colegas. Por medio del módulo de publicación dicho agente productor: 1) proporciona las características técnicas de su recurso, 2) provee información sobre él mismo (dueño) y 3) define roles y asigna permisos a los agentes consumidores.

En la Figura 4.5 sólo se detallan las características técnicas de un video (e.g., título, autor y formato) y la definición de roles, ya que el resto de la información es muy similar a la presentada en las figuras que ilustran la descripción de recursos físicos.

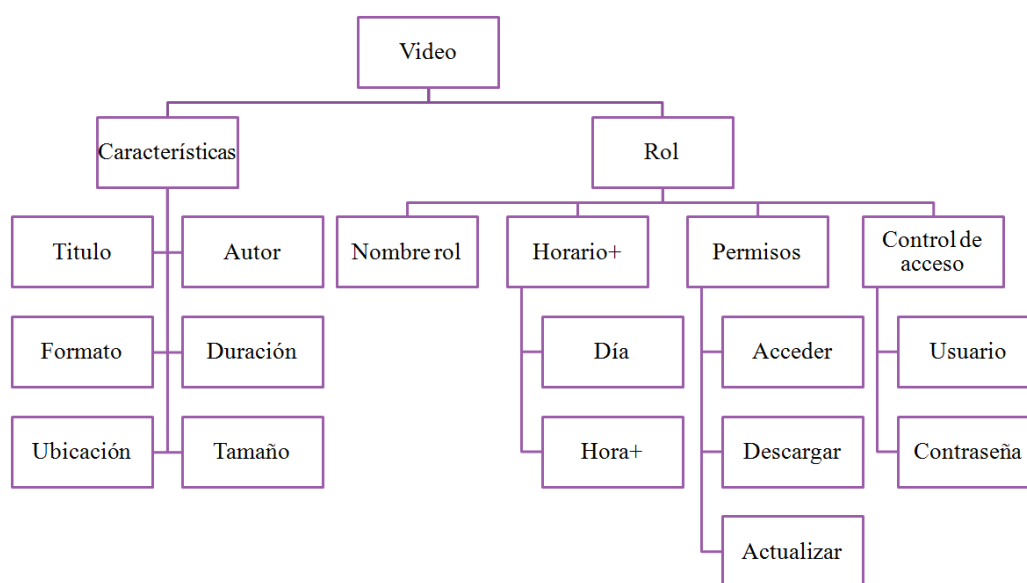


Figura 4.5: Descripción en XML de un video

En este caso, la definición de un rol incluye el nombre del rol, horario de uso, permisos y parámetros de control de acceso. Un rol determinado puede estar asociado a varios permisos e.g., al rol “estudiante” se puede otorgar el permiso de **acceder** el recurso, mientras que al rol “profesor”, se puede atribuir los permisos de **acceder** y **descargar** dicho recurso. Por seguridad, cada rol puede tener asignado un control de acceso compuesto por un nombre de usuario y una contraseña.

### 4.1.4 Descripción de recursos humanos

La descripción de un recurso humano es más compacta que la de un recurso físico o informático. La Figura 4.6 muestra la información que un agente productor proporciona sobre él mismo por medio del módulo de publicación. Esta información incluye: a) sus datos personales, e.g., nombre, afiliación, área de investigación y ubicación y b) un horario tentativo que puede ser enriquecido con una agenda de trabajo. Sin embargo, es

bien sabido que un colaborador podría no permanecer constantemente en un lugar determinado ni cumplir necesariamente con un horario preestablecido. Con el fin de capturar implícitamente la presencia y la ubicación de los colaboradores, se propone un sistema de reconocimiento de caras que actualiza constantemente esta información.

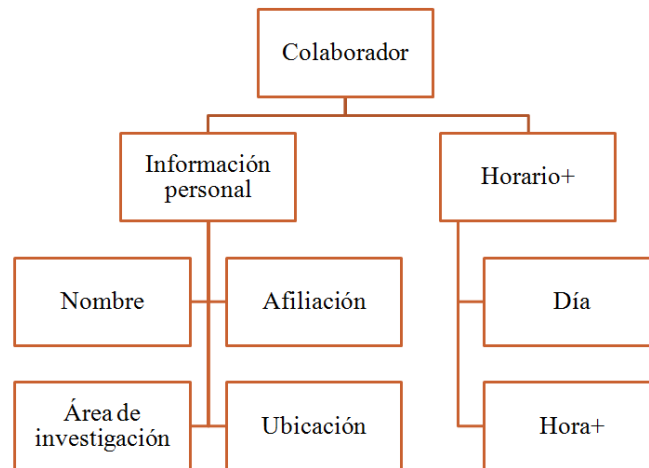


Figura 4.6: Descripción en XML de un colaborador

Finalmente, cuando un agente consumidor requiere un recurso, utilizará el módulo de suscripción para especificar todas o sólo algunas de sus características técnicas. A partir de esta información, el sistema RAMS buscará un recurso adecuado y disponible para dicho agente consumidor.

## 4.2 Motor de inferencia

Como se ha explicado anteriormente (cf. sección 3.5), el motor de inferencia evalúa información estática y dinámica sobre los recursos publicados en el sistema RAMS, con el fin de ofrecer una respuesta que satisfaga la petición de un agente consumidor.

La información estática que proporciona el agente productor al publicar alguno de sus recursos incluye: a) las capacidades o características técnicas del recurso, b) la definición de restricciones de uso y c) la asignación de permisos a los agentes consumidores sobre dicho recurso. Por su parte, la información dinámica, que utiliza el motor de inferencia para determinar la presencia, la ubicación y la disponibilidad de los recursos, es provista por: a) un sistema de reconocimiento de caras, b) un sistema de localización del recurso más cercano y c) aplicaciones de usuario.

La implementación del motor de inferencia, descrita en esta sección, es una propuesta inicial ya que únicamente toma en cuenta información estática. Por el momento, el motor de inferencia es capaz de decidir si entre los recursos, accesibles a un agente consumidor, existe alguno que cumpla con las capacidades o características técnicas que el agente consumidor requiere. Cabe señalar que el sistema de reconocimiento de caras (el cual es una de las fuentes de información dinámica del sistema RAMS) ha sido implementado,

pero sus resultados aún no han sido incorporados a la propuesta actual del motor de inferencia.

Así, el motor de inferencia toma en cuenta la información que el agente productor proporciona al publicar un recurso en el sistema RAMS. En primer lugar, verifica que el agente productor esté dispuesto a compartir su recurso con el agente consumidor. Posteriormente, compara las características técnicas del recurso registrado con la petición del agente consumidor para determinar el grado de adecuación de dicho recurso. Cuatro grados de adecuación han sido definidos: a) excelente, b) muy bueno, c) bueno o d) posiblemente útil. De otra manera, el recurso se clasifica como inadecuado. A guisa de ilustración, una impresora sería clasificada como:

- **Excelente:** si la marca y el modelo de la impresora, que aparecen en la petición del agente consumidor, corresponden con la marca y el modelo de alguna impresora registrada. En este caso, algún agente productor publicó una impresora justamente como la que requiere el agente consumidor.
- **Muy buena:** si la tecnología de impresión, resolución, velocidad, colores y espacio de impresión (e.g., ambos lados) de la impresora, que ha sido publicada en el sistema RAMS, responde a las necesidades del agente consumidor. Aún cuando no haya correspondencia con el modelo y la marca de la impresora que el agente consumidor requiere, la mayoría de sus características técnicas satisface los requerimientos de dicho agente.
- **Buena:** si la tecnología de impresión, la resolución y los colores de la impresora registrada concuerdan con los que el agente consumidor necesita. Aún cuando la impresora no cuente con todas las características técnicas que se especificaron en la petición, el agente consumidor podría estar parcialmente satisfecho con un dispositivo que cumple sólo algunos de sus requerimientos.
- **Posiblemente útil:** si los colores que maneja la impresora registrada coinciden con los que el agente consumidor requiere. Aunque este dispositivo no satisface las necesidades de dicho agente, podría ser eventualmente útil. Para ilustrar este caso, suponga que un estudiante necesita imprimir urgentemente un diagrama de clases para analizarlo con su asesor. El estudiante realiza la petición correspondiente al sistema RAMS, el cual le informa que sólo existe una impresora calificada como posiblemente útil para satisfacer su petición. Como el estudiante necesita imprimir inmediatamente su diagrama, decide utilizar dicha impresora, aún cuando no sea la más adecuada.

Es posible que no existan recursos publicados en el sistema RAMS que cubran todos los requerimientos de un agente consumidor. Gracias a la clasificación precedente el sistema RAMS puede ofrecer información sobre los recursos que satisfacen incluso parcialmente las necesidades de dicho agente. Así, un agente consumidor siempre estará consciente de que entre menos positiva sea la evaluación de un recurso, él podría obtener menor satisfacción si decide utilizarlo.

El motor de inferencia recibe como entrada: a) la petición del agente consumidor en forma de un documento XML y b) la identificación de dicho agente. Una vez que el motor de inferencia cuenta con esta información, obtiene el tipo y el subtipo de recurso que requiere el agente consumidor.

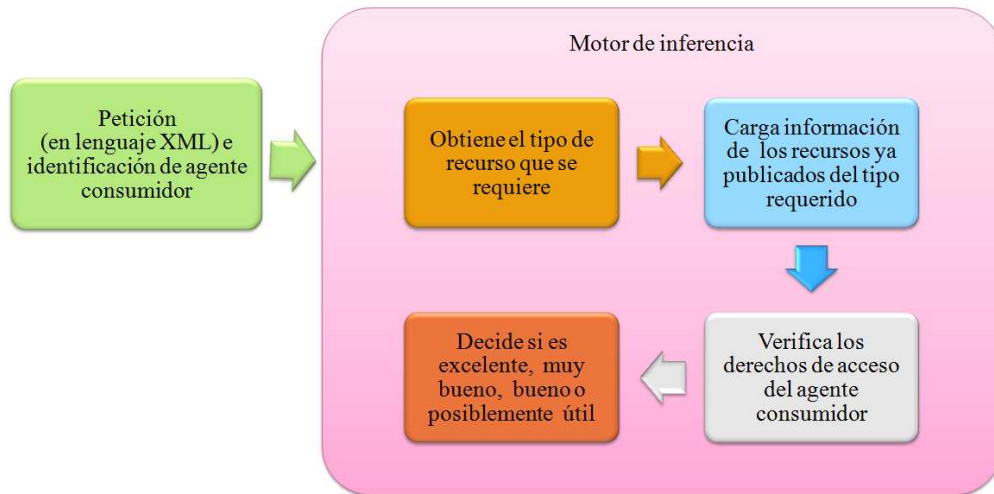


Figura 4.7: Procesos que se realizan en el motor de inferencia

El proceso de selección de los recursos adecuados, en cuanto a capacidades o características técnicas, inicia cuando el motor de inferencia obtiene el tipo y el subtipo del recurso solicitado (ver Figura 4.7). Esta información se encuentra especificada en el documento XML que el motor de inferencia recibió. Posteriormente, se carga la información de todos los recursos de dicho subtipo que han sido publicados en el sistema RAMS (e.g., si se requiere un escáner, entonces se cargará la información de todos los escáneres registrados en el sistema RAMS). De esta información, se descartan los recursos sobre los que el agente consumidor no tiene permisos. Después, el motor de inferencia compara la descripción técnica de los recursos restantes con las necesidades descritas en la petición, con el objetivo de asignar a cada uno de los recursos la etiqueta **excelente**, **muy bueno**, **bueno** o **posiblemente útil**. Finalmente, la información referente al conjunto de recursos adecuados en diferentes grados, se envía al sistema de notificación, cuya implementación no ha sido concretada.

### 4.3 Sistema de reconocimiento de caras

Como se mencionó anteriormente (cf. sección 3.2), un recurso humano es un colaborador que decide interactuar con sus colegas por medio del sistema RAMS. La característica de movilidad de este tipo de recurso pone en evidencia la necesidad de contar con un mecanismo que capture implícitamente su ubicación, con la finalidad de:

1. ofrecer un ambiente ubicuo en donde los recursos humanos reciban información personalizada de acuerdo a su ubicación;

2. proporcionar información sobre la ubicación de algún recurso humano a los colaboradores interesados.

El mecanismo propuesto en este trabajo de investigación es un sistema de reconocimiento de caras que identifica a los colaboradores, en lugar de los recursos físicos que portan. Los recursos físicos utilizados usualmente para localizar a las personas (e.g., *PDA* y *Smartphone*) son fácilmente transferibles y olvidables. Por esta razón, si se asume que un recurso humano siempre se encuentra en el mismo lugar que su dispositivo móvil, esta suposición podría provocar que los colaboradores interesados en la ubicación de dicho recurso humano reciban información errónea.

Un sistema de reconocimiento de caras evita los problemas de transferencia y olvido de dispositivos móviles ya que un colaborador, que interactúa con sus colegas por medio del sistema RAMS, no necesita llevar consigo ningún dispositivo físico para ser localizado. El sistema RAMS conocerá su ubicación, capturando y procesando imágenes de su cara con cámaras distribuidas en el espacio de colaboración donde esté implementado. El sistema de reconocimiento de caras consta de dos fases:

1. **Fase de aprendizaje:** consiste en entrenar al sistema para que reconozca a los colaboradores que se requiere identificar;
2. **Fase de prueba o funcionamiento:** se encarga de hacer predicciones sobre la identidad de los colaboradores.

El sistema de reconocimiento de caras es una adaptación de un sistema de reconocimiento de nebulosas planetarias, que fue realizado por el equipo del Dr. Olague en colaboración con el Instituto de Astronomía de la UNAM. Esta adaptación fue posible gracias al trabajo realizado en conjunto con el Dr. Olague del Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE) y con el Dr. Decouchant del Centre National de Recherche Scientifique (CNRS) de Francia.

### 4.3.1 Biblioteca *OpenCV*

La identificación de una persona requiere resolver varios problemas. El primero de ellos se refiere a cómo discernir una cara humana del resto de los objetos captados por una cámara. Para solucionar este problema, se requiere un algoritmo que evalúe permanentemente los objetos que están frente a una cámara, de manera que esta última capture una imagen sólo cuando detecte una cara humana.

*OpenCV* (*Open Source Computer Vision*) es una biblioteca construida por Intel, que está compuesta de funciones programadas en varios lenguajes (e.g., C++, python y Visual Basic). *OpenCV* está orientada principalmente al campo de visión por computadora en tiempo real.

Para construir el sistema de reconocimiento, se utilizó una herramienta de detección de caras, programada en lenguaje C++, que proporciona la biblioteca *OpenCV*. Esta herramienta es un detector de objetos que utiliza el principio de características *Haar-like*



[Viola y Jones, 2001], las cuales codifican la existencia de contrastes en regiones específicas de una imagen (no necesariamente una cara humana). Un conjunto de estas características *Haar-like* puede ser utilizado para codificar los contrastes y las relaciones espaciales que existen en una cara humana.

El detector de objetos de *OpenCV* fue propuesto inicialmente por [Viola y Jones, 2001] y mejorado por [Lienhart y Maydt, 2002]. Este detector utiliza un clasificador que fue entrenado con algunos cientos de ejemplos positivos y negativos. Los ejemplos positivos son imágenes del objeto que se desea detectar (i.e., una cara humana), mientras que los ejemplos negativos son imágenes arbitrarias.

Para encontrar un objeto determinado (en este caso, una cara humana) después de la fase de aprendizaje, el clasificador es aplicado a una región de interés. El resultado que arroja este clasificador es 1 si la región de interés corresponde al objeto que se está buscando o 0 en otro caso. Cuando se busca un objeto en una imagen grande, una ventana de búsqueda se mueve a través de la imagen y verifica cada posición, mediante el clasificador, para decidir si se trata de un objeto de interés. Este clasificador puede ser redimensionado fácilmente, a fin de encontrar objetos de interés de diferentes tamaños. Así, este procedimiento de escaneo resulta más eficiente que la redimensión de toda la imagen para ajustarla al tamaño de las imágenes de entrenamiento. Para encontrar un objeto de tamaño desconocido en una imagen, el procedimiento de escaneo en busca de caras humanas debe realizarse repetidas veces, utilizando ventanas de búsqueda de diferentes tamaños.

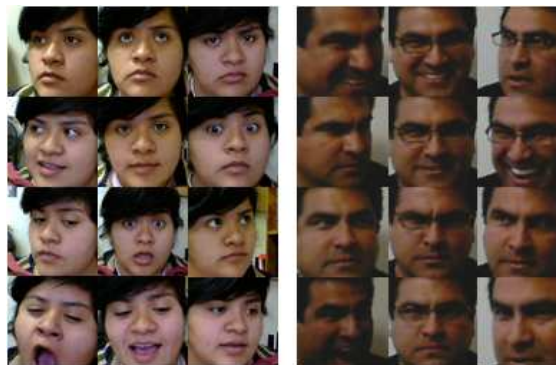


Figura 4.8: Construcción del conjunto de entrenamiento

La herramienta de detección de caras que proporciona *OpenCV* es utilizada tanto en la fase de aprendizaje como en la fase de prueba. En la fase de aprendizaje, *OpenCV* permite construir un conjunto de entrenamiento, el cual está compuesto aproximadamente por 30 imágenes diferentes de cada colaborador. Para construir dicho conjunto de entrenamiento (ver Figura 4.8), se toman fotografías de las personas que se requiere identificar: 1) en diferentes posiciones (e.g., de frente, volteando un poco a la izquierda y viendo hacia arriba), 2) haciendo gesticulaciones diversas y 3) utilizando accesorios. Entre más posiciones, gesticulaciones y cambios en la apariencia de un colaborador se capturen al crear el conjunto de entrenamiento, la probabilidad de que el sensor de reconocimiento cometa

errores cuando hace predicciones para identificar a un colaborador será menor.

En la fase de prueba, la herramienta de detección de caras de *OpenCV* se utiliza en todo momento para evaluar los objetos que la cámara capta. De esta manera, la cámara capturará una imagen siempre que se detecte una cara humana. Particularmente, esta herramienta de detección ofrece las siguientes funcionalidades:

- a) **Reconocimiento de caras de diferentes tamaños** (ver Figura 4.9): el tamaño de una cara depende de la distancia entre el colaborador y la cámara. En consecuencia, cuando se detecta una cara, la cámara toma una fotografía, de la cual primero se extrae la cara y después se escala para obtener finalmente una imagen de 245 x 245 píxeles;

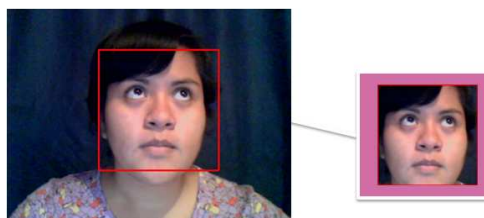


Figura 4.9: Detección de una cara humana

- b) **Localización simultánea de varias caras humanas** (ver Figura 4.10): esta funcionalidad resulta bastante útil en la fase de prueba, ya que si varios colaboradores se encuentran frente a la cámara, esta herramienta es capaz de detectar simultáneamente la cara de cada uno de ellos.

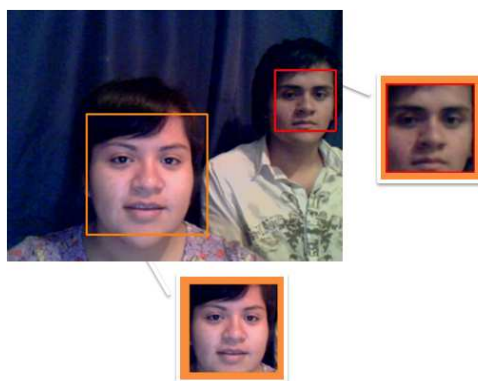


Figura 4.10: Detección simultánea de varias caras humanas

### 4.3.2 Análisis de Componentes Principales

*OpenCV* ayuda a resolver el problema de detección de caras humanas. El siguiente problema a resolver consiste en encontrar un método para obtener las características que

permitan diferenciar a un colaborador de otro. Para resolver este problema, se utilizó el método de Análisis de Componentes Principales (PCA *Principle Components Analysis*), también llamado método de *Eigenfaces* [Turk y Pentland, 1991].

Este método también es utilizado en las fases de aprendizaje y de prueba. Particularmente, en la fase de aprendizaje, este método analiza el conjunto de entrenamiento que contiene las imágenes de los colaboradores que se desea reconocer, con el fin de obtener las características principales que permiten distinguir una persona de otra. En la fase de prueba, este método se utiliza para obtener las características principales de una imagen recién capturada.

El método de *Eigenfaces* ha sido catalogado como un método basado en apariencias, cuya meta principal es crear representaciones de baja dimensión, a partir de las imágenes de las caras de las personas que se desean reconocer. Estas representaciones se obtienen al suprimir información irrelevante [Bolme et al., 2003] y al descomponer la estructura facial en componentes conocidos como *Eigenfaces*. Es importante destacar que, de acuerdo con este método, dichas características no corresponden necesariamente a los ojos, a la nariz y a las orejas, sino que capturan los puntos de la imagen que causan una variación significativa entre las caras del conjunto de entrenamiento. Estas características ayudarán a diferenciar un colaborador del resto.

En contraste con el método de *Eigenfaces*, los métodos basados en características geométricas intentan distinguir caras, comparando las propiedades y las relaciones espaciales entre elementos faciales como ojos, boca, nariz y mentón. El éxito de estos métodos depende en gran medida de la forma de extraer las características y del proceso de medición.

El método de *Eigenfaces* es considerado como el primer sistema genuino de reconocimiento de caras, ya que la mayoría de los trabajos precedentes estaban basados en características geométricas. Este método tiene buenos resultados cuando se utilizan imágenes que no se ven afectadas por cambios en la iluminación. Por esta razón, se recomienda construir el conjunto de entrenamiento en un ambiente similar al que existirá cuando el sistema se ponga en marcha, a fin de evitar que las variaciones de luz alteren los resultados del análisis.

La Figura 4.11 muestra las operaciones y los procesos que se llevan a cabo para obtener las características principales que diferencian un colaborador de otro. En primer lugar, se cargan las imágenes del conjunto de entrenamiento (que originalmente fueron tomadas a colores) y enseguida se cambian a blanco y negro para realizar su tratamiento. Este cambio se hace con la intención de trabajar solamente con un canal, en vez de trabajar con tres canales como en el caso de las imágenes a color. Posteriormente, se calcula el promedio, la covarianza y la desviación promedio de estas imágenes. Todas estas operaciones permiten obtener las *Eigenfaces* y los *Eigenvectores*, los cuales son necesarios para calcular los pesos. Para obtener estos tres últimos valores, se realizan tres procesos que constan de una serie de operaciones. En la Figura 4.11 se representan dichos procesos como cajas negras, ya que sólo se utilizaron para completar el método de *Eigenfaces*, sin modificar su implementación.

De hecho, los pesos constituyen la información más importante que este análisis ar-

roja. Estos pesos están formados por vectores que contienen una etiqueta y un conjunto de características, donde la etiqueta identifica al colaborador al que pertenecen dichas características. Cabe señalar que pueden existir varios vectores asociados a un mismo colaborador.

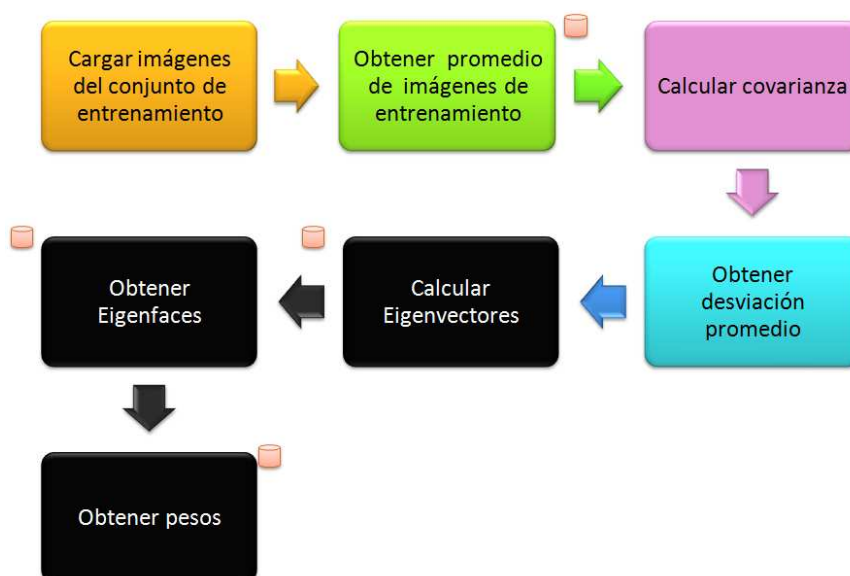


Figura 4.11: Análisis de Componentes Principales

### 4.3.3 *Support Vector Machines*

Por medio de la herramienta de detección de objetos, que provee *OpenCV*, es posible crear un conjunto de entrenamiento formado por las imágenes de las caras de las personas que serán identificadas. Por su parte, el método de Análisis de Componentes Principales permite extraer las características que diferencian un colaborador de otro. Para completar el proceso de identificación de personas, se utiliza una biblioteca llamada LIBSVM [Chang y Lin, 2001]. En la fase de aprendizaje, esta biblioteca permite crear un modelo de clasificación de las características obtenidas mediante el método de *Eigenfaces*. En la fase de prueba, este modelo hace posible que se realicen predicciones sobre la identidad de un colaborador.

SVM (*Support Vector Machines*) fue desarrollado por [Cortes y Vapnik, 1995], con el objetivo de realizar clasificaciones binarias. Este algoritmo recibe como entrada un conjunto de entrenamiento compuesto por parejas formadas de etiquetas e instancias. Las etiquetas representan clases (i.e., personas), mientras que las instancias simbolizan sus características. El objetivo de SVM es encontrar un hiperplano que ayude a separar dos clases mediante un margen máximo. Este hiperplano se conoce como modelo de clasificación.

La Figura 4.12 muestra dos clases separadas por un margen. Los puntos que caen sobre él se llaman vectores de soporte (*support vectors*), en tanto que la línea que se encuentra

a la mitad de dicho margen es el hiperplano de separación óptimo. Originalmente SVM sólo puede resolver problemas de clasificación binaria, pero LIBSVM utiliza una técnica de comparación “uno contra uno” para realizar una clasificación multiclase.

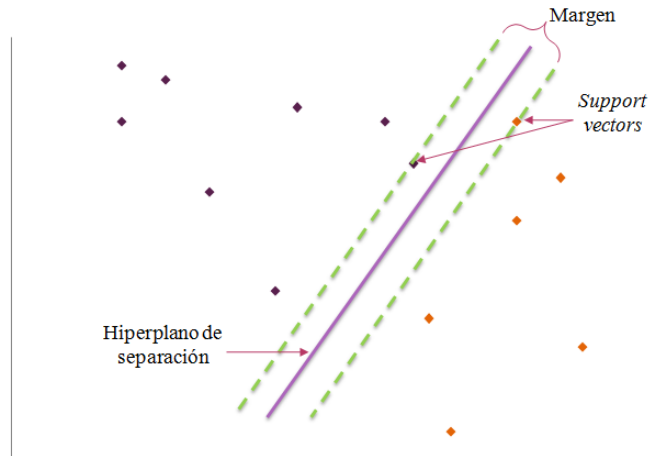


Figura 4.12: Hiperplano construido por SVM

El modelo de clasificación que ha sido implementado en el sistema de reconocimiento de caras es multiclase, ya que se quiere identificar a un grupo de colaboradores. Este modelo se construye en la fase de aprendizaje, a partir de los pesos que se obtienen del análisis de componentes principales. En la fase de prueba, el modelo de clasificación recibe los datos obtenidos del análisis de características principales de una imagen recién tomada, con el fin de realizar una predicción sobre la identidad del colaborador al que pertenece dicha imagen.

#### 4.3.4 Experimento

Recapitulando, el sistema de reconocimiento de caras consta de dos fases, una de aprendizaje y una de prueba o funcionamiento. La primera se realiza sólo una vez, mientras que la segunda se efectúa cada vez que la cámara capta una cara humana. La fase de aprendizaje comprende las siguientes actividades:

1. por medio de la herramienta de detección de caras humanas de *OpecCV*, se crea un conjunto de entrenamiento conformado por las imágenes de los colaboradores que se requiere identificar;
2. se analizan las imágenes del conjunto de entrenamiento mediante el método de *Eigenfaces* para obtener las características principales de cada colaborador;
3. mediante la biblioteca LIBSVM se construye un modelo de clasificación a partir de las características obtenidas en el análisis anterior.



Figura 4.13: Colaboradores que reconoce el sistema para este experimento

Por su parte, la fase de prueba o funcionamiento comprende las siguientes actividades:

1. se captura una imagen reciente del colaborador que se desea identificar por medio de la herramienta de detección de caras de *OpenCV*;
2. se analiza la imagen mediante el método de *Eigenfaces* para obtener sus características principales;
3. se hace una predicción del colaborador a quien corresponde dicha imagen. Para obtener esta predicción, el modelo de clasificación creado por la biblioteca LIBSVM evalúa las características obtenidas en el análisis anterior.

Con el fin de realizar un experimento, el sistema de reconocimiento de caras fue entrenado para reconocer a cuatro colaboradores (ver Figura 4.13). Cada uno de ellos representa una clase diferente.

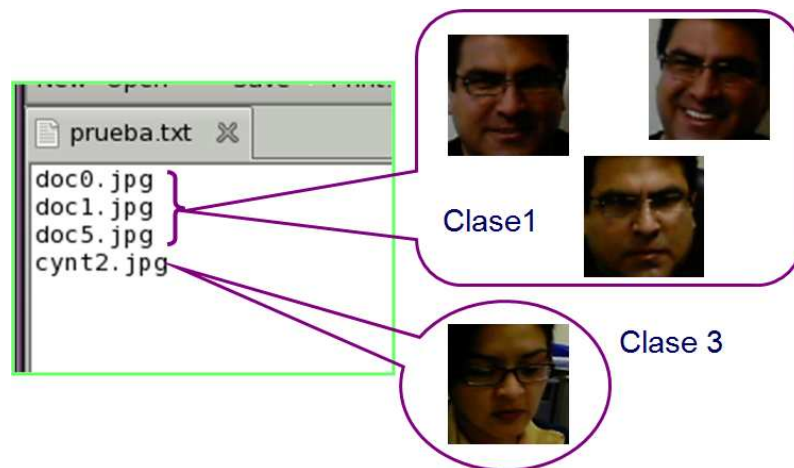
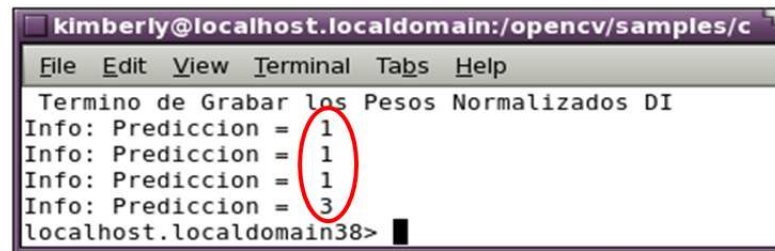


Figura 4.14: Imágenes de prueba

Posteriormente, el sistema se probó utilizando nuevas imágenes de dos colaboradores registrados. Como se muestra en la Figura 4.14, se analizaron tres imágenes de la clase 1 y una imagen de la clase 3. El resultado que arrojó el sistema de reconocimiento de

caras es correcto. Como se puede ver en la Figura 4.15, este resultado indica que las tres primeras imágenes corresponden a la clase 1, mientras que la última imagen pertenece a la clase 3.



```
kimberly@localhost.localdomain:/opencv/samples/c
File Edit View Terminal Tabs Help
Termino de Grabar los Pesos Normalizados DI
Info: Prediccion = 1
Info: Prediccion = 1
Info: Prediccion = 1
Info: Prediccion = 3
localhost.localdomain38>
```

Figura 4.15: Resultados que arroja el sistema de reconocimiento de caras





# Capítulo 5

## Conclusiones y trabajo futuro

El objetivo de este trabajo de tesis fue el diseño de un sistema que facilite la administración de recursos compartidos en un ambiente cooperativo y ubicuo (sección 5.1). En este capítulo se retoma el planteamiento del problema, el cual se refiere a la falta de soporte informático en organizaciones (que cuentan con múltiples recursos públicos y privados) donde resulta indispensable el uso compartido de recursos cuya disponibilidad puede cambiar de acuerdo a diversos parámetros (e.g., permisos y restricciones de uso). Enseguida, se presentan las conclusiones del sistema que ha sido diseñado para ofrecer una solución a este problema (sección 5.2). Finalmente, se proponen algunas mejoras y extensiones futuras de este sistema (sección 5.3).

### 5.1 Recapitulación del planteamiento del problema

Este trabajo de tesis está inmerso en dos campos de investigación: en el cómputo ubicuo y en el Trabajo Cooperativo Asistido por Computadora (TCAC). El objetivo del cómputo ubicuo es incrementar el uso de sistemas computacionales a través del entorno físico al hacerlos disponibles, pero también invisibles al usuario. Desafortunadamente, la mayoría de los trabajos de investigación, que abordan la ubicuidad, forman parte del dominio de los sistemas mono-usuarios. Por parte del TCAC, esta tesis toma la perspectiva computacional (i.e., desarrollo de sistemas colaborativos) de este campo, con el fin de diseñar e implementar un sistema que administre la percepción de información de conciencia de grupo. Esta información se refiere al estado (i.e., percepción, localización y ubicación) de los colaboradores y de sus recursos compartidos. Un aspecto importante que debe tomarse en cuenta cuando se obtiene información sobre el estado de un colaborador es la privacidad, ya que se debe cuidar la forma y la cantidad de información que se publica. Es posible que un sistema colaborativo que ha sido diseñado de manera inadecuada ofrezca: 1) demasiada información, lo que podría provocar problemas sociales o 2) información escasa, lo que podría ocasionar problemas de contactos inapropiados o pérdida de oportunidades.

El ámbito de estudio de esta tesis es el entorno de una organización, cuyos recursos humanos son colaboradores potenciales que pueden estar localizados en varios edificios.

Dentro de esta organización, existen múltiples recursos heterogéneos, los cuales pueden estar distribuidos en lugares públicos o privados. En un escenario real, resulta imposible que una persona disponga en su oficina de cada uno de los recursos físicos existentes en la organización. Adicionalmente, desde el punto de vista técnico, no es factible duplicar todos los recursos informáticos de los colaboradores en cada uno de los dispositivos de almacenamiento que existen en la organización. Por esta razón, se vuelve evidente la necesidad de compartir dichos recursos entre los colaboradores. Sin embargo, para soportar el uso compartido de los recursos inmersos en un entorno con carácter ubicuo, se requiere administrar la disponibilidad de los colaboradores y de sus recursos.

## 5.2 Conclusiones

En una organización donde existen múltiples recursos heterogéneos, resulta evidente la necesidad de contar con una herramienta que administre el uso compartido de dichos recursos. En respuesta a esta necesidad, se ha propuesto el sistema RAMS, el cual contempla tres tipos de recursos: físicos (e.g., impresora, escáner y sala de juntas), informáticos (e.g., aplicaciones y multimedia) y humanos (i.e., una persona). La arquitectura de *software* del sistema RAMS está compuesta por cuatro subsistemas esenciales: 1) un *broker*, 2) un motor de inferencia, 3) un conjunto de sensores y 4) un sistema de notificación. La interacción entre los colaboradores y el sistema RAMS está basada en el modelo de publicación/subscripción, en el cual participan agentes productores que publican recursos que desean compartir con sus colegas y agentes consumidores que buscan recursos que necesitan utilizar. El *broker* contiene un módulo de publicación y un módulo de subscripción, además de un filtro de información, el cual clasifica los recursos publicados por tópico y las peticiones de recursos por contenido.

Por su parte, el motor de inferencia se encarga de decidir cuál es el recurso que mejor satisface las necesidades de un agente consumidor. Esta decisión se toma mediante la evaluación de información estática y dinámica. La información estática se refiere a la información que proporcionan los agentes productores cuando publican sus recursos. Esta información incluye la descripción de las capacidades o características técnicas de los recursos, sus restricciones de uso y los permisos asignados a los agentes consumidores. Por su parte, la información dinámica se refiere al estado (i.e., disponibilidad, presencia y localización) de los recursos. Esta información se obtiene del conjunto de sensores y de las aplicaciones de usuario.

El conjunto de sensores está compuesto por un sistema de reconocimiento de caras y un sistema de localización del recurso más cercano. El primer sistema permite localizar y conocer la presencia de un colaborador dentro de la organización. En tanto que el sistema de localización del recurso más cercano es capaz de decidir cuál es el recurso que se encuentra menos alejado del colaborador que lo solicita. Las aplicaciones de usuario permiten que un colaborador cambie explícitamente su estado y el de sus recursos, con el objetivo de evitar interrupciones. Una vez que el motor de inferencia toma una decisión, el sistema de notificación se encarga de informarla al agente consumidor. Adicionalmente,

el agente consumidor puede activar filtros de información, los cuales le permitirán impedir la recepción de notificaciones que no son de su interés.

El uso de información dinámica permite conocer el estado real de los colaboradores y de sus recursos compartidos. Se ha propuesto un sistema basado en sensores, ya que los sistemas que conforman este conjunto estarán en continuo funcionamiento para proveer la información de estado que se requiere. Cabe destacar que el sistema de reconocimiento de caras permite conocer de forma implícita la presencia y la ubicación de los colaboradores, siempre que ellos lo autoricen. Por su parte, las aplicaciones de usuario posibilitan que un colaborador cambie su estado en cualquier momento. De esta forma, se pueden ofrecer un sistema que automatice algunas tareas, pero que al mismo tiempo permita mantener la privacidad que los colaboradores desean.

### 5.3 Trabajo futuro

La contribución principal de esta tesis es el diseño y la implementación (parcial) de un sistema colaborativo con carácter ubicuo, el cual permite la administración del uso compartido de recursos. Este diseño ha permitido descubrir múltiples extensiones que enriquecerían el sistema RAMS, las cuales han sido clasificadas en: a) implementación de los subsistemas pendientes, b) extensión de los subsistemas implementados y finalmente c) desarrollo de aplicaciones del entorno de colaboración.

Los subsistemas del sistema RAMS que aún no se han implementado son:

- **sistema de localización del recurso más cercano:** este sistema permitirá determinar, a partir del conjunto de recursos adecuados para satisfacer una petición, cuál es el que se encuentra más cerca del agente consumidor que realiza la petición.
- **sistema de notificación:** este sistema es responsable de entregar la respuesta que el motor de inferencia proporciona sobre el recurso que mejor satisface una petición. En estos momentos, la implementación del sistema RAMS sólo funciona de forma local.

Dentro de las extensiones que se pueden realizar a los componentes del sistema RAMS que han sido implementados, se encuentra la implantación de estos componentes en un entorno distribuido, como se ha propuesto en la sección 3.8, ya que por ahora funcionan de manera local.

Adicionalmente, en la implementación actual, los módulos de publicación y de suscripción solicitan información específica del subtipo de recurso que un agente desea compartir o buscar. Esta forma de descripción podría ser mejorada permitiendo que los agentes describan los recursos por medio de un conjunto de palabras, de manera que no se sientan limitados por las opciones restringidas que el sistema RAMS les permite elegir. Aunque esta forma de descripción sería más flexible, también podría ocasionar problemas al motor de inferencia, ya que las búsquedas de recursos con este tipo de descripciones son muy sensibles al conjunto de palabras que se emplean. Así, el algoritmo de decisión que utiliza

el motor de inferencia debería modificarse, de forma que estas descripciones basadas en conjuntos de palabras incrementen la eficiencia en la búsqueda de recursos, e.g., Ruta et al. [Ruta et al., 2008] proponen el uso de una ontología para la descripción de recursos.

En cuanto al algoritmo del motor de inferencia, se propone desarrollar un método basado en los conceptos de abducción y contracción [Colucci et al., 2003]. Resultaría interesante realizar un análisis de este método y de algunos otros (e.g., el que proponen Sycara et al. [Sycara et al., 2002]), ya que podrían ofrecer una forma de mejorar la manera en que el motor de inferencia discrimina entre los recursos que cumplen los requerimientos de un agente consumidor y los que no lo hacen.

El sistema de reconocimiento de caras puede hacerse más robusto, de forma que sea capaz de identificar a un colaborador que se encuentre en cualquier posición (i.e., no necesariamente de frente). Esta extensión se puede lograr mediante la creación de varios modelos de clasificación, en lugar de un único modelo de clasificación con el que se cuenta en la implementación actual. Por su parte, el sistema de localización del recurso más cercano podría mejorarse agregando mecanismos que ayuden a localizar el dispositivo (PDA) que un colaborador porta. Esta forma de localización sería útil en los lugares abiertos donde no existen cámaras (e.g., jardines o explanadas). Así, el sistema RAMS sería capaz de proporcionar las mismas funcionalidades que en un espacio cerrado.

Adicionalmente, se propone el desarrollo de las siguientes aplicaciones que conformarán el entorno de colaboración ubicua:

- **aplicaciones de usuario:** por medio de estas aplicaciones, un colaborador podrá cambiar de forma explícita la información que conforma su estado (i.e., presencia, localización y disponibilidad) con el fin de permitir o restringir la interacción con sus colegas o el uso compartido de sus recursos.
- **entorno de mensajería instantánea:** esta aplicación apoyará la comunicación remota entre colegas, la cual podrá llevarse a cabo cuando alguno de los colaboradores que desean interactuar no se encuentren presentes en la organización, pero hayan iniciado una sesión remota mediante el sistema RAMS, e.g., entre el CICESE y el CINVESTAV.
- **interfaces gráficas de usuario:** actualmente, el sistema RAMS no cuenta con ninguna interfaz gráfica amigable para el usuario. Resultaría conveniente crear interfaces de usuario, que permitan realizar los procesos de publicación y suscripción por medio de herramientas que puedan ser utilizadas de manera intuitiva.
- **mapa para mostrar la localización de los colaboradores y de sus recursos:** se propone un mapa que permita visualizar la ubicación de los colaboradores y de los recursos físicos compartidos. De esta forma, cuando un agente consumidor realice una petición de un recurso físico, podrá visualizar el camino que debe recorrer para llegar al recurso que requiere.

## Publicación del autor

[Garcia et al., 2008] K. Garcia, S. Mendoza, G. Olague, D. Decouchant y J. Rodriguez, *Shared Resource Availability within Ubiquitous Collaboration Environments*, In Proceedings of the 14th Collaboration Researchers' International Workshop on Groupware (CRIWG '08), LNCS Springer, pp. 25-40, Nebraska (EEUU), Septiembre 2008.



# Referencias

- [Abel, 1990] M. Abel, Experiences in an exploratory distributed organization, *In Intellectual Teamwork: Social Foundations of Cooperative Work*, Lawrence Erlbaum Associates, New Jersey (EEUU), pp. 489-510, 1990.
- [Bardram y Christensen, 2007] J. Bardram y H. Christensen, *Pervasive Computing Support for Hospitals: an Overview of the Activity-Based Computing Project*, In IEEE Pervasive Computing, 6(1):44-51, Enero 2007.
- [Berard y Coutaz, 1997] F. Berard y J. Coutaz, *Awareness As an Extended Notion of Observability*, In CHI'97 Workshop on Awareness in Collaborative Systems, Conference on Human Factors in Computing Systems (SIGCHI'97), ACM Press, 29(4), Atlanta GA (EEUU), Octubre 1997.
- [Bolme et al., 2003] D. Bolme, R. Beveridge, M. Teixeira y B. Draper, *The CSU Face Identification Evaluation System: Its Purpose, Features and Structure*, In Proceedings of International Conference on Vision Systems, Springer-Verlag, Graz (Austria), Abril 2003.
- [Chang y Lin, 2001] Chang C-C y Lin C-J, LIBSVM: A Library for Support Vector Machines, 2001.
- [Colucci et al., 2003] S. Colucci, T. Di Noia, E. Di Sciascio, F. Donini y M. Morgiello, *Concept Abduction and Contraction in Description Logics*, In Proceedings of the 16th International Workshop on Description Logics (DL'03), 2003.
- [Cortes y Vapnik, 1995] C. Cortes y V. Vapnik, Journal of Support Vector Networks, Machine Learning, 20(3):273-297, Septiembre 1995.
- [Coutaz et al., 1997] J. Coutaz, J. Crowley y F. Bérard, *Eigen-Space Coding as a Means to Support Privacy in Computer Mediated Communication*, INTERACT '97: Proceedings of the IFIP TC13 Interantional Conference on Human-Computer Interaction, pp. 532-538, Londres (Inglaterra), 1997.
- [Dahlberg y Sanneblad, 2000] P. Dahlberg y J. Sanneblad, *The Use of Bluetooth enabled PDAs: some preliminary use experiences*, In Proceedings of IRIS 23 Laboratorium for Interaction Technology, 2000.

- [Dourish y Bellotti, 1992] P. Dourish y V. Bellotti, *Awareness and Coordination in Shared Work Spaces*, In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'92), 1992.
- [Dourish y Bly, 1992] P. Dourish y S. Bly, *Portholes: Supporting Awareness in a Distributed Work Group*, In Proceedings of Conference on Human Factors in Computing System (CHI'92), ACM Press, pp. 541-547, Monterrey CA (EEUU), 1992.
- [DOM Level 2, 2000] *DOM Level 2 Traversal and Range Specification*, a W3C Recommendation, 13 Noviembre 2000.
- [Keith, 2000] E. W. Keith, *Core Jini*, 2da Edición, Prentice Hall, EEUU, Diciembre 2000.
- [Ellis et al., 1991] C.A. Ellis, S.J. Gibbs y G.L. Rein, *Groupware: Some Issues and Experiences*, Communications of the ACM, 34(1):38-58, 1991.
- [Eugster et al., 2003] P. Eugster, P. Felber R. Guerraoui y A. Kermarrec, *The many faces of publish/subscribe*, ACM Computing Survey, 25(2):114-131, 2003.
- [Greenberg y Johnson, 1997] S. Greenberg y B. Johnson, *Studying Awareness in Contact Facilitation*, In CHI'97 Workshop on Awareness in Collaborative Systems, Conference on Human Factors in Computing Systems (SIGCHI'97), ACM Press, 29(4), Atlanta GA (EEUU), Octubre 1997.
- [Guttman et al., 1999] E. Guttman, C. Perkins, J. Veizades, y M. Day, *Service Location Protocol Version 2*, IETF RFC 2608, Junio 1999.
- [Hernandez et al., 2007] B. Hernandez, G. Olague, R. Hammoud, L. Trujillo y E. Romero, *Visual Learning of Texture Descriptors for Facial Expression Recognition in Thermal Imagery*, Computer Vision and Image Understanding, 106(2-3):258-269, 2007.
- [Hodes et al., 2002] T.D. Hodes, S.E. Czerwinski, B. Y. Zhao, A.D. Joseph y R. H. Katz, *An Architecture for Secure Wide-Area Service Discovery*, Wireless Networks, 8(2/3):213-230, Marzo 2002.
- [Lienhart y Maydt, 2002] R. Lienhart y J. Maydt, *An Extended Set of Haar-like Features for Rapid Object Detection*, In IEEE ICIP, Septiembre 2002.
- [Markarian et al., 2006] A. Markarian, J. Favela, M. Tentori y L.A. Castro, *Seamless Interaction Among Heterogeneous Devices in Support for Co-located Collaboration*, In Proceedings of the 12th International Workshop on Groupware: Design, Implementation, and Use (CRIWG 2006), LNCS 4154, Springer, pp. 389-404, Medina del Campo (España), Septiembre 2006.
- [McDaniel, 1996] S.E. McDaniel, *Providing Awareness Information to Support Transitions in Remote Computer-Mediated Collaboration*, In Proceedings of the Conference Companion on Human Factors in Computing Systems: Common Ground (CHI'96), ACM Press, pp. 57-58, Vancouver BC (Canada), Abril 1996.



- [Miller, 2003] J.R. Miller, *The Remote application controller*, Compute Graphics, 27(4):605-615, 2003.
- [Muehl et al., 2007] G. Muehl, L. Fiege y P. Pietzuch, *Distributed Event-Based Systems*, Springer-Verlag, Alemania, 2007.
- [Ruta et al., 2008] M. Ruta, T. Di Noia, E. Di Sciascio, F. Scioscia, *Abduction and Contraction for Semantic-based Mobile Dating in P2P Environments*, In proceedings of International Conference on Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM, pp. 626-632, 2008.
- [Sycara et al., 2002] K. Sycara, S. Widoff, M. Klusch y J. Lu, *LARKS: Dynamic Match-making Among Heterogeneous Software Agents in Cyberspace*, Journal of Autonomous Agents and Multi-Agent Systems, 5(2):173-293, 2002.
- [Turk y Pentland, 1991] M. Turk y A. Pentland, *Eigenfaces for Recognition*, Journal of Cognitive Neuroscience, 3(1):71-86, 1991.
- [Viola y Jones, 2001] P. Viola y M. Jones, *Rapid Object Detection Using Boosted Cascade of Simple Features*, In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'01), IEEE, Hawaii (EEUU), Diciembre 2001.
- [Weiser, 1993] M. Weiser, *Some Computer Science Problems in Ubiquitous Computing*, Communications of the ACM, 36(7):75-84, Julio 1993.
- [Whittaker et al., 1994] S. Whittaker, D. Frohlich y O. Daly-Jones, *Informal workplace communication: What is it like and how might we support it?*, In Proceedings of ACM Conference on Human Factors in Computing Systems, 1994.
- [Zhu et al., 2005] F. Zhu, M. Mutka y L. Ni, *Service Discovery in Pervasive Computing Environments*, In IEEE Pervasive Computing, 4(4):80-90, Octubre-Diciembre, 2005.