



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

PREDESYS

Realizado por
JOSÉ ANTONIO JIMÉNEZ CARMONA
(NIF)

Dirigido por
PABLO NEIRA AYUSO

Departamento
LENGUAJES Y SISTEMAS INFORMÁTICOS

Sevilla, junio 2011

Índice general

1. Acerca del proyecto	9
1.1. Licencias	9
2. Introducción	11
2.1. Mi proyecto: Predesys	11
2.2. Objetivos generales	12
2.3. Trabajos relacionados	12
3. Planificación	21
4. Coste	23
4.1. El modelo COCOMO	23
4.2. Coste del proyecto	24
5. Elicitación de requisitos	27
5.1. Objetivos del sistema	27
5.2. Requisitos de Información	28
5.3. Reglas de negocio	30
5.4. Requisitos funcionales	32
5.4.1. Actores del sistema	32
5.4.2. Diagramas de casos de uso del sistema	32
5.4.3. Casos de uso del sistema	40
5.5. Requisitos no funcionales	61
6. Arquitectura	63
6.1. Sistemas distribuidos	63
6.2. Arquitectura Cliente-Servidor	63
6.3. Arquitectura de Predesys	64
7. Implementación	67
7.1. Subsistema de gestión de usuarios	67
7.2. Subsistema de gestión de servicios	67
7.3. Subsistema de gestión de errores	67
8. Pruebas	69
9. Distribución	71
9.1. Cómo generar el <i>paquete tarball</i> de un componente	72
9.2. Cómo generar el <i>paquete Debian</i> de un componente	72
9.3. Requisitos del Servidor	72

9.4. Requisitos de la API de Servicios	73
9.5. Requisitos del Radar Bluetooth	73
9.6. Requisitos del Radar LAN	73
9.7. Requisitos de People Presence	73
10.Instalación	75
10.1. Servidor	75
10.2. API de Servicios	76
10.3. Radar Bluetooth	77
10.4. Radar LAN	78
10.5. Servicios de Predesys	79
11.Uso y configuración	81
11.1. Manejo del Servidor	81
12.Cómo crear un servicio de Predesys	87

Índice de figuras

2.1.	Logotipo de Predesys	11
2.2.	Esquema de Predesys	12
2.3.	Dispositivo emisor de mensajes de BluKit	13
2.4.	Pantalla de la aplicación web para móviles de Geo Remind Me	14
2.5.	Pantalla de la aplicación móvil nativa de Geo Remind Me	14
2.6.	Pantalla de la aplicación web de Geo Remind Me	15
2.7.	Esquema de Power Warning	16
2.8.	Pantalla de la aplicación de configuración e BlueHoc	17
5.1.	Diagrama de subsistemas	33
5.2.	Diagrama de casos de uso del Subsistema de Gestión de Usuarios (1) . . .	34
5.3.	Diagrama de casos de uso del Subsistema de Gestión de Usuarios (2) . . .	35
5.4.	Diagrama de casos de uso del Subsistema de Gestión de Servicios (1) . . .	36
5.5.	Diagrama de casos de uso del Subsistema de Gestión de Servicios (2) . . .	37
5.6.	Diagrama de casos de uso del Subsistema de Gestión de Servicios (3) . . .	38
5.7.	Diagrama de casos de uso del Subsistema de Gestión de Servicios (4) . . .	39
5.8.	Diagrama de casos de uso del Subsistema de Gestión de Servicios (5) . . .	40
5.9.	Diagrama de casos de uso del Subsistema de Gestión de Sistema	40

Índice de cuadros

2.1. Tabla comparativa de trabajos relacionados	19
---	----

Capítulo 1

Acerca del proyecto

El autor de este proyecto es José Antonio Jiménez Carmona, alumno de Ingeniería Técnica en Informática de Gestión en la Escuela Técnica Superior de Ingeniería Informática de la Universidad de Sevilla. Este documento conforma el proyecto Fin de Carrera del autor y tiene como tutor a Pablo Neira Ayuso, profesor del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla.

Datos de contacto:

- José Antonio Jiménez Carmona: jacarmona@us.es
- Pablo Neira Ayuso: pneira@us.es

1.1. Licencias

Este apartado presenta las licencias establecidas para el proyecto en sí y para este mismo documento.

▪ Licencia de este documento

Este documento se rige por la licencia *Reconocimiento-CompartirIgual 3.0* de Creative Commons[1]. Esta licencia exige que en cualquier explotación de la obra hará falta reconocer la autoría y permite la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

▪ Licencia del proyecto

El proyecto Predesys se rige por la licencia *GNU Lesser General Public License 3.0*[2].

Capítulo 2

Introducción

Hoy en día, Internet está presente de lleno en nuestras vidas y en muchas ocasiones necesitamos acceder a él para obtener información concreta sobre diversos asuntos. Además, en muchas ocasiones, esta información que buscamos está relacionada con algún entorno físico en el que podemos encontrarnos. El problema está en que para hacer uso de algún servicio de información de ese tipo, normalmente necesitamos un ordenador y un software concreto, algo que no solemos llevar encima cuando salimos de casa.

2.1. Mi proyecto: Predesys



Figura 2.1: Logotipo de Predesys

Predesys es una solución a la creación de servicios de información basados en la detección de dispositivos inalámbricos y de otros tipos en un determinado lugar, ofreciendo un sistema extensible por medio de plugins para el cual el usuario no requiere ningún hardware o software especial, necesitando tan sólo su teléfono móvil o algún otro dispositivo de uso común.

Concretamente, se trata de un sistema contenido en un servidor central y un conjunto de servidores secundarios asociados con el objetivo de detectar la presencia de diferentes dispositivos (principalmente, inalámbricos, como teléfonos móviles y ordenadores portátiles) que se encuentren cerca físicamente de dichos servidores secundarios para ofrecer una serie de servicios de información a los usuarios de esos dispositivos y a terceros. Está diseñado para los sistemas operativos Linux, es software libre y está escrito en el lenguaje de programación Python[3].



Figura 2.2: Esquema de Predesys

2.2. Objetivos generales

Los objetivos principales del proyecto son:

1. Ofrecer un sistema capaz de detectar un dispositivo inalámbrico concreto y ofrecer un cierto servicio a dicho dispositivo.
2. Ofrecer una arquitectura extensible basada en plugins de servicios para el sistema.

2.3. Trabajos relacionados

A continuación, veremos algunos trabajos relacionados con la temática en la que se mueve Predesys, junto con una tabla comparativa de todos estos trabajos, que aparece al final. Los medios que he utilizado en la búsqueda de estos trabajos han sido Internet y el buscador Google.

BluKit

BluKit es un sistema de mensajería a través de Bluetooth que permite enviar diferentes mensajes a teléfonos móviles que se encuentren en alguna zona concreta. Cada unidad BluKit puede conectarse a una red de mensajería llamada BluNet, que permite enviar

mensajes a otras zonas en las que se encuentren otras unidades BluKit.

- Tipo de trabajo: Aplicación software y servicio de información
- Coste (de unidad BluKit): 39,95 euros (sin IVA)
- Coste (de pertenencia a BluNet): Bajo presupuesto
- Licencia: Propietaria
- Tecnologías implicadas: Bluetooth
- Autores: BluLabs (info@blulabs.es)
- Sitio web: [4]



Figura 2.3: Dispositivo emisor de mensajes de BluKit

Geo Remind Me

GeobRemind Me es una aplicación muy ligera que permite añadir alertas especiales a *smartphones*. Estas alertas avisarán al usuario cuando esté cerca de una localización en la que tenga alguna tarea pendiente.

- Tipo de trabajo: Aplicación software para dispositivos móviles, aplicación web
- Coste: Gratuito
- Licencias: GPLv3 y AGPLv3
- Tecnologías implicadas: Android, iPhone, GPS, HTML 5
- Autores: Equipo de Geo Remind Me
- Sitio web: [5]

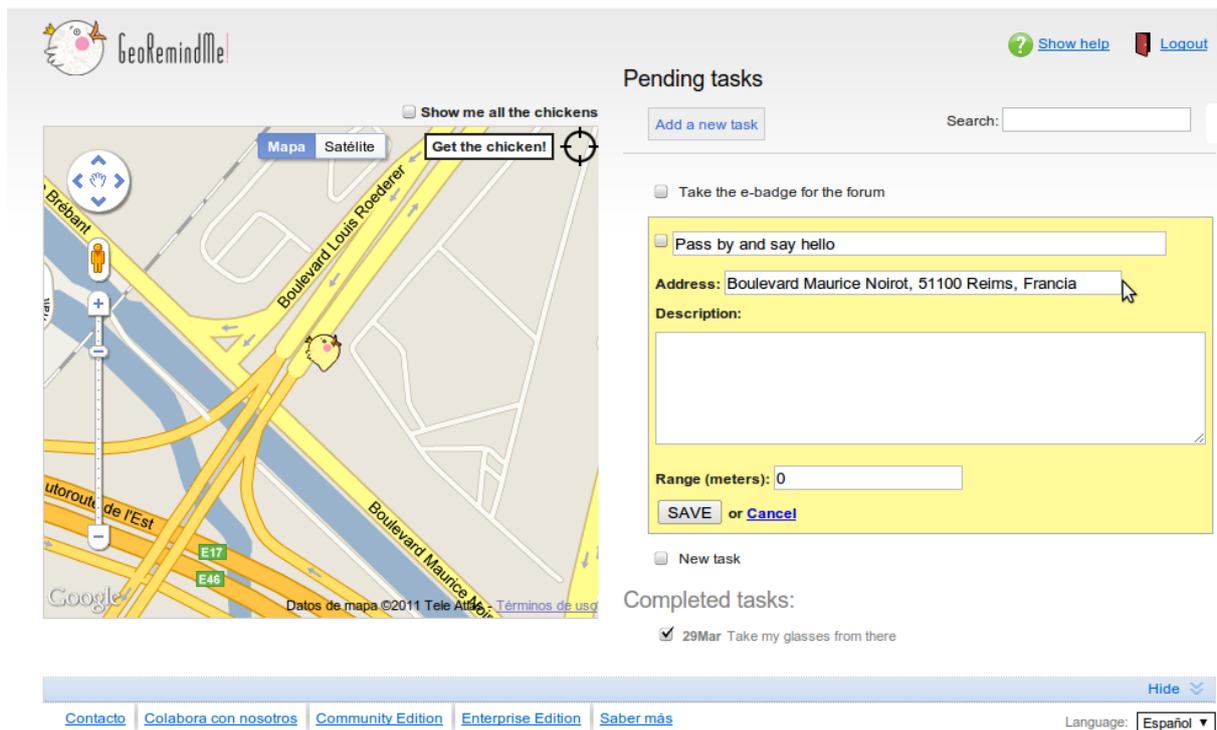


Figura 2.4: Pantalla de la aplicación web para móviles de Geo Remind Me



Figura 2.5: Pantalla de la aplicación móvil nativa de Geo Remind Me



Figura 2.6: Pantalla de la aplicación web de Geo Remind Me

PowerWarning

PowerWarning es un detector de presencia de personas en una determinada sala. Consiste en utilizar un teléfono móvil que detecta cuándo unas puertas han sido abiertas y envía un SMS a otro teléfono para avisar a alguien de la intrusión.

- Tipo de trabajo: Software para dispositivos móviles
- Coste: Gratuito
- Licencia: Apache License 2.0
- Tecnologías implicadas: Simbian, SMS, Python
- Autores: Luca Cassioli (cassioli@libero.it)
- Sitio web: [6]

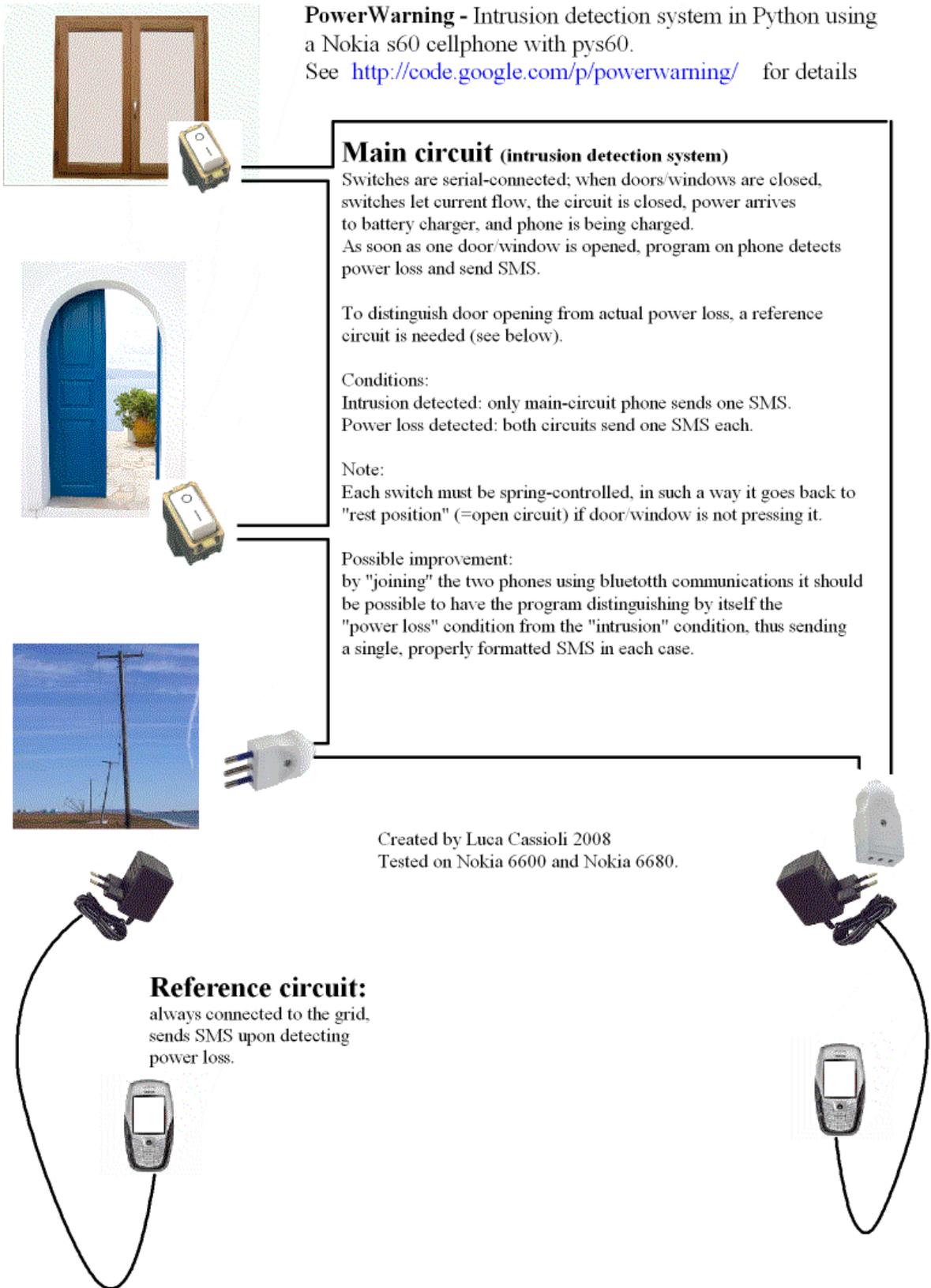


Figura 2.7: Esquema de Power Warning

BlueHoc

BlueHoc es un simulador de protocolos de comunicación realizado por IBM. Permite que las aplicaciones software puedan enviar información a través de Bluetooth creyendo que la están enviando por otro protocolo, como TCP/IP.

- Tipo de trabajo: Software de comunicaciones
- Coste: Gratuito
- Licencia: IBM Public License
- Tecnologías implicadas: Bluetooth, TCP/IP, C++
- Autor: Apurva Kumar (kapurva@in.ibm.com)
- Sitio web: [7]

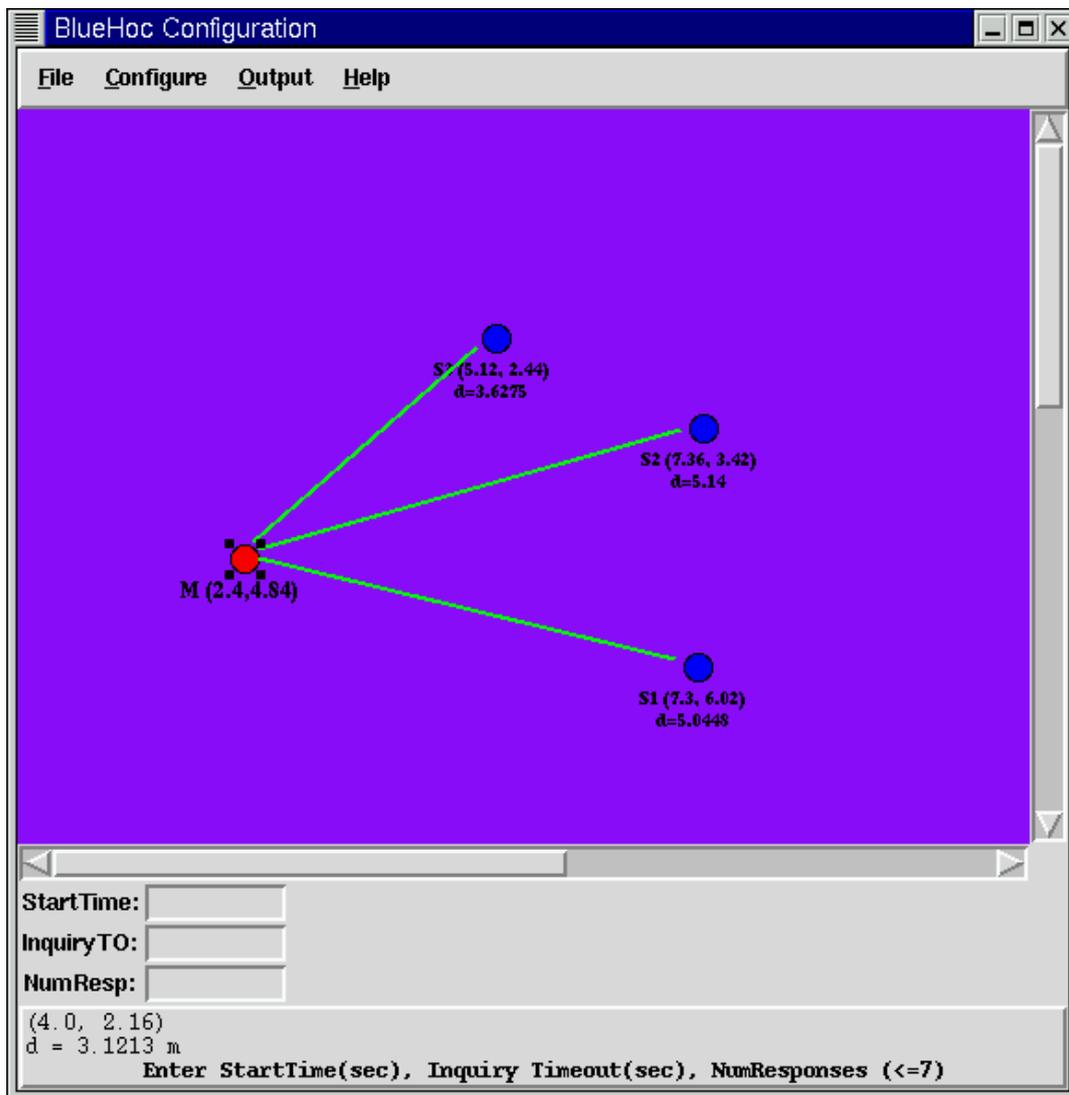


Figura 2.8: Pantalla de la aplicación de configuración e BlueHoc

PySerial

PySerial es una biblioteca del lenguaje de programación Python que proporciona soporte para conexiones en serie (RS-232) a través de una variedad de dispositivos diferentes:

puertos serie antiguos, adaptadores Bluetooth, puertos de infrarrojos... También soporta puertos serie remotos a través de la RFC 2217 (desde la versión 2.5).

- Tipo de trabajo: Biblioteca software
- Coste: Gratuito
- Licencia: Python Software Foundation License
- Tecnologías implicadas: Bluetooth, IR, puerto serie
- Autor: Chris Liechti (cliechti@gmx.net)
- Sitio web: [8]

RFIDIOt

RFIDIOt es una biblioteca del lenguaje de programación Python para detectar dispositivos RFID basado en PySerial.

- Tipo de trabajo: Biblioteca software
- Coste: Gratuito
- Licencia: GPL 2.0
- Tecnologías implicadas: RFID, Python
- Autor: Adam Laurie (adam@algroup.co.uk)
- Sitio web: [9]

Java Socket Interface for Infrared Communication

Java Socket Interface for Infrared Communication es una API para el lenguaje de programación Java hecha por Hewlett Packard para comunicar un ordenador con sistema operativo Linux con otro dispositivo por medio de rayos infrarrojos.

- Tipo de trabajo: Biblioteca software
- Coste: Gratuito
- Licencia: LGPL 2.1
- Tecnologías implicadas: IR, Linux, Java
- Autores: A. Venkataramani (avramani@india.hp.com) y Jean Tourrilhes (jt@hpl.hp.com)
- Sitio web: [10]

Tabla comparativa de trabajos

A continuación, se presenta una tabla comparativa del proyecto y de los trabajos relacionados anteriormente, donde se refleja los puntos fuertes de cada uno.

Proyecto	Sistema completo	Extensible	Gratuito	Libre uso	Gest. usuarios	Ind. dispositivos	Multi det.	Gest. errores
BluKit	✓	✗	✗	✗	✓	✓	✗	✓
GRM	✓	✓	✓	✓	✓	✗	✗	✓
PW	✗	✗	✓	✓	✗	✗	✗	✗
BlueHoc	✗	✗	✓	✓	✗	✓	✗	✗
PySerial	✗	✗	✓	✓	✗	✓	✓	✗
RFIDIOT	✗	✗	✓	✓	✗	✗	✗	✗
JSIIC	✗	✗	✓	✓	✗	✗	✗	✗
Predesys	✓	✓	✓	✓	✓	✓	✓	✓

Cuadro 2.1: Tabla comparativa de trabajos relacionados

- *Ind. dispositivos* = Independencia de los dispositivos usuario.
- *Multi det.* = Multideteción. Detección por varios protocolos.

Capítulo 3

Planificación

A continuación, se exponen los tiempos estimados para la realización de cada parte del proyecto y los tiempos reales que han resultado al final, junto con el error relativo. Los tiempos están expresados en horas.

Tarea	Estimación (h)	Real (h)	Error relativo
Investigación	54	64	19 %
Planificación	12	15	25 %
Diseño de la Arquitectura	40	48	20 %
Elicitación y Análisis de Requisitos	35	40	14 %
Implementación	140	168	20 %
Pruebas	28	31	11 %
Total	309	366	18 %

Podemos concluir diciendo que el error relativo total es aceptable, pues es tan sólo del 18 %.

Capítulo 4

Coste

En este capítulo veremos lo que teóricamente cuesta el proyecto, en términos de tiempo, dinero y personal. Mediremos este coste basándonos en el modelo COCOMO, muy utilizado en el ámbito de la ingeniería de software. Antes de exponer el cálculo del coste del proyecto, introduciremos brevemente el modelo COCOMO[11].

4.1. El modelo COCOMO

El Modelo Constructivo de Costes (o COCOMO, por su acrónimo del inglés COConstructive COst MOdel) es un modelo matemático de base empírica utilizado para estimación de costes de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: Básico, Intermedio y Detallado.

Este modelo fue desarrollado por Barry W. Boehm a finales de los años 70 y comienzos de los 80, exponiéndolo detalladamente en su libro "Software Engineering Economics" (Prentice-Hall, 1981). Pertenece a la categoría de modelos de subestimaciones basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el "tamaño" del proyecto, en líneas de código principalmente.

Las ecuaciones que se utilizan en los tres modelos son:

- $E = a(Kl)b * m(X)$, en persona-mes
- $Tdev = c(E)d$, en meses
- $P = E/Tdev$, en personas

donde:

- E es el esfuerzo requerido por el proyecto, en persona-mes
- $Tdev$ es el tiempo requerido por el proyecto, en meses
- P es el número de personas requerido por el proyecto
- a, b, c y d son constantes con valores definidos en una tabla, según cada submodelo
- Kl es la cantidad de líneas de código, en miles.
- $m(X)$ Es un multiplicador que depende de 15 atributos.

A su vez, cada submodelo también se divide en modos que representan el tipo de proyecto, y puede ser:

- **Modo Orgánico:** un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles (medio).
- **Modo Semilibre o Semiencajado:** corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- **Modo Rígido o Empotrado:** el proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.

Modelo Básico de COCOMO

El modelo Básico de COCOMO se utiliza para obtener una primera aproximación rápida del esfuerzo, y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes:

Modo	a	b	c	d
Orgánico	2,40	1,05	2,50	0,38
Semilibre	3,00	1,12	2,50	0,35
Rígido	3,60	1,20	2,50	0,32

Estos valores son para las fórmulas:

- Personas por mes necesarias para llevar adelante el proyecto: $MM = a * (Kl^b)$
- Tiempo de desarrollo del proyecto (en meses): $TDEV = c * (MM^d)$
- Personas necesarias para realizar el proyecto: $CosteH = MM/TDEV$
- Coste total (mensual) del proyecto: $CosteM = CosteH * Salario\ medio\ entre\ los\ programadores\ y\ analistas$

Se puede observar que a medida que aumenta la complejidad del proyecto (modo), las constantes aumentan de 2.4 a 3.6, que corresponde a un incremento del esfuerzo del personal. Hay que utilizar con mucho cuidado el modelo básico puesto que se obvian muchas características del entorno.

4.2. Coste del proyecto

Para estimar el coste del proyecto Predesys, se ha empleado el modelo Básico de COCOMO en el modo Orgánico, por ser un proyecto con menos de 10000 líneas de código y unipersonal.

- **Personas por mes necesarias:**
 $MM = a * (Kl^b) = 2,4 * (7,2^{1,05}) = 19 \text{ personas/mes}$
- **Tiempo de desarrollo del proyecto (en meses):**
 $TDEV = c * (MM^d) = 2,5 * (19^{0,38}) = 7,65 \approx 8 \text{ meses}$

- **Personas necesarias para realizar el proyecto:**

$$CosteH = MM/TDEV = 19/7,65 = 2.48 \approx \mathbf{3 \text{ personas}}$$

- **Coste total (mensual) del proyecto:**

$$CosteM = CosteH * \text{Salario medio entre los programadores y analistas} = 3 * 1000 \\ = \mathbf{3000 \text{ euros}}$$

- **Coste final del proyecto:**

$$CosteF = CosteM * TDEV = 3000 * 8 = \mathbf{24000 \text{ euros}}$$

Capítulo 5

Elicitación de requisitos

Este capítulo presenta los requisitos que deberá cumplir el sistema. Comenzaremos exponiendo los objetivos del sistema, para después ver las operaciones del mismo, los requisitos y restricciones de información y los casos de uso, finalizando con los requisitos no funcionales. Daremos así una definición global del proyecto.

5.1. Objetivos del sistema

El sistema debe cumplir los siguientes objetivos y suboperaciones:

Subsistema de Gestión de Usuarios

El sistema debe soportar operaciones de creación, modificación y eliminación de usuarios, así como organizarlos por grupos.

- **OBJ-0001: Gestión de grupos de usuarios**

El sistema debe ser capaz de crear/editar/borrar grupos de usuarios. Dichas operaciones se realizarán sobre la base de datos del Servidor.

- **OBJ-0002: Gestión de usuarios**

El sistema debe ser capaz de crear/editar/borrar usuarios y asociar/desasociar a un usuario concreto un dispositivo determinado. Dichas operaciones se realizarán sobre la base de datos del Servidor.

Subsistema de Gestión de Servicios

El sistema debe soportar operaciones de creación, modificación y eliminación de servicios, así como organizarlos por tipos.

- **OBJ-0003: Gestión de tipos de servicios**

El sistema debe ser capaz de crear/editar/borrar tipos de servicios. Dichas operaciones se realizarán sobre la base de datos del Servidor.

- **OBJ-0004: Gestión de servicios**

El sistema debe ser capaz de crear/editar/borrar servicios. A cada servicio se le asociarán, opcionalmente, 1 ó más grupo de usuarios como únicos usuarios autorizados para ejecutar dicho servicio. Dichas operaciones se realizarán sobre la base de datos del Servidor.

- **OBJ-0005: Gestión de tipos de Radares**

El sistema debe soportar distintos tipos de Radares, pudiendo recibir identificadores de dispositivos detectados de todos ellos.

- **OBJ-0006: Gestión de Radares**

El sistema debe soportar la recepción de identificadores de dispositivos detectados de varios Radares simultáneamente. Cada uno de los Radares tendrá asociada una cuenta de usuario del sistema para identificarse con el Servidor.

- **OBJ-0007: Gestión de registro de identificadores de dispositivos**

El sistema registrará en la base de datos del Servidor todos los identificadores de dispositivos detectados que se envíen desde los Radares, hasta cierto límite de registros a partir del cual los últimos serán eliminados.

- **OBJ-0008: Gestión de envío de mensajes a dispositivos**

El sistema, a través de alguno de los Radares que tenga asociados, deberá ser capaz de enviar un mensaje de texto a un dispositivo determinado.

Subsistema de Gestión de Errores

El sistema deberá controlar la aparición de errores de ejecución.

- **OBJ-0009: Gestión de registro de errores**

El sistema guardará en un archivo de texto fácil de localizar todas las excepciones que ocurran en el Servidor.

5.2. Requisitos de Información

A continuación, se presentan los requisitos de información del sistema, es decir, qué información debe almacenar el sistema y cómo debe hacerlo.

IRQ-0001: Grupos de usuarios

El sistema deberá almacenar la información correspondiente a los grupos de usuarios, en concreto:

- Identificador del grupo de usuarios
- Datos del grupo de usuarios (nombre)

IRQ-0002: Usuarios

El sistema deberá almacenar la información correspondiente a los usuarios, en concreto:

- Identificador del usuario
- Grupo de usuarios al que pertenece
- *Habilitado* (será un parámetro binario que indique si el usuario puede acceder al sistema).
- Contraseña
- Datos del usuario (nombre y dirección de correo electrónico)
- Dispositivos asociados con el usuario

IRQ-0003: Tipos de servicio

El sistema deberá almacenar la información correspondiente a los servicios, en concreto:

- Identificador del tipo de servicio
- Datos del tipo de servicio (nombre)

IRQ-0004: Servicios

El sistema deberá almacenar la información correspondiente a los servicios, en concreto:

- Identificador del servicio
- Tipo de servicio
- *Autenticado* (será un parámetro binario que indique si el servicio requiere autenticación de usuario para ejecutarse).
- *Habilitado* (será un parámetro binario que indique si el servicio está disponible).
- Datos del servicio (nombre, versión y autor)

IRQ-0005: Tipos de dispositivo

El sistema deberá almacenar información sobre los tipos de dispositivo que puede detectar a través de los radares o a los que puede asociar un usuario.

- Identificador del tipo de dispositivo
- Datos del tipo de dispositivo (nombre)

IRQ-0006: Dispositivos

El sistema deberá almacenar información sobre los dispositivos detectados a través de los radares y los dispositivos asociados a cada usuario.

- Identificador del dispositivo
- Tipo de dispositivo

IRQ-0007: Tipos de Radar

El sistema deberá almacenar información sobre los tipos de dispositivo Radar que soporte.

- Identificador del tipo Radar
- Datos del Radar (nombre)

IRQ-0008: Radares

El sistema deberá almacenar información sobre los distintos dispositivos Radares que integren el sistema.

- Identificador del Radar
- Tipo de Radar
- Datos del Radar (nombre)

IRQ-0009: Historial de dispositivos detectados

El sistema deberá almacenar un historial de los dispositivos detectados a través de los Radares, con información sobre cada detección.

- Dispositivo detectado
- Radar que hizo la detección

5.3. Reglas de negocio

Vamos a ver ahora las reglas de negocio del sistema, es decir, las restricciones que debe tener la información almacenada por el sistema.

CRQ-0001: Unicidad de los identificadores de los grupos de usuarios

- **Asociado a:** *IRQ-0001*
- **Descripción:** El identificador de un grupo de usuarios no puede ser igual al de otro grupo.
- **Motivo:** Garantizar la identificación y distinción entre grupos de usuarios.

CRQ-0002: Unicidad de los identificadores de los usuarios

- **Asociado a:** *IRQ-0002*
- **Descripción:** El identificador de un usuario no puede ser igual al de otro usuario.
- **Motivo:** Garantizar la identificación y distinción entre usuarios.

CRQ-0003: Unicidad de los identificadores de los tipos de servicio

- **Asociado a:** *IRQ-0003*
- **Descripción:** El identificador de un tipo de servicio no puede ser igual al de otro tipo.
- **Motivo:** Garantizar la identificación y distinción entre tipos de servicio.

CRQ-0004: Unicidad de los identificadores de los servicios

- **Asociado a:** *IRQ-0004*
- **Descripción:** El identificador de un servicio no puede ser igual al de otro servicio.
- **Motivo:** Garantizar la identificación y distinción entre servicios.

CRQ-0005: Unicidad de los identificadores de los tipos de dispositivo

- **Asociado a:** *IRQ-0005*
- **Descripción:** El identificador de un tipo de dispositivo no puede ser igual al de otro tipo.
- **Motivo:** Garantizar la identificación y distinción entre tipos de dispositivo.

CRQ-0006: Unicidad de los identificadores de los dispositivos

- **Asociado a:** *IRQ-0006*
- **Descripción:** El identificador de un dispositivo no puede ser igual al de otro dispositivo.
- **Motivo:** Garantizar la identificación y distinción entre dispositivos.

CRQ-0007: Asociación de dispositivos a usuarios

- **Asociado a:** *IRQ-0002* y *IRQ-0006*
- **Descripción:** Un mismo usuario puede tener asociados 1 ó más dispositivos y 1 mismo dispositivo puede estar asociado a 1 ó más usuarios.
- **Motivo:** Garantizar la flexibilidad de las asociaciones de dispositivos.

CRQ-0008: Grupo de usuarios correcto

- **Asociado a:** *IRQ-0001* y *IRQ-0002*
- **Descripción:** El grupo de usuarios asociado con un usuario debe ser un grupo existente o ninguno.
- **Motivo:** Garantizar una correcta integración de la información.

CRQ-0009: Tipos de servicio asociados con los servicios

- **Asociado a:** *IRQ-0003* y *IRQ-0004*
- **Descripción:** Un servicio siempre tiene que tener asociado un tipo de servicio existente.
- **Motivo:** Garantizar una correcta integración de la información.

5.4. Requisitos funcionales

A continuación, se exponen los requisitos funcionales del sistema. Según Wikipedia[12], un requisito funcional define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica.

5.4.1. Actores del sistema

Según Wikipedia[13] un actor del sistema es toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también incluye a todos los sistemas externos, además de entidades abstractas, como el tiempo.

Los actores del sistema han de ser los siguientes:

- **ACT-0001: Usuario Normal:** Este actor representa a los usuarios que hacen uso de los servicios del sistema, siendo detectados por el mismo. No tienen privilegios especiales ni pueden configurar el sistema. A priori, serán mayoría frente a los administradores.
- **ACT-0002: Usuario Registrado:** Este actor representa a los usuarios normales que estén registrados en el sistema. Pueden identificarse via Internet con su usuario y contraseña para ejecutar ciertos servicios no automáticos que así lo requieran. No tienen privilegios especiales ni pueden configurar el sistema. A priori, serán mayoría frente a los administradores.
- **ACT-0003: Administrador:** Este actor representa al usuario registrado encargado de gestionar todo el sistema, teniendo privilegios especiales.
- **ACT-0004: Radar:** Este actor no representa a una persona, pero sí a un elemento clave del sistema: el Radar, el cual es el encargado de enviar cada cierto tiempo al Servidor los identificadores de los dispositivos que ha detectado. Este actor es en realidad un tipo de usuario registrado.

5.4.2. Diagramas de casos de uso del sistema

Veamos a continuación los diagramas de los casos de uso del sistema. Según Wikipedia[13], un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema.

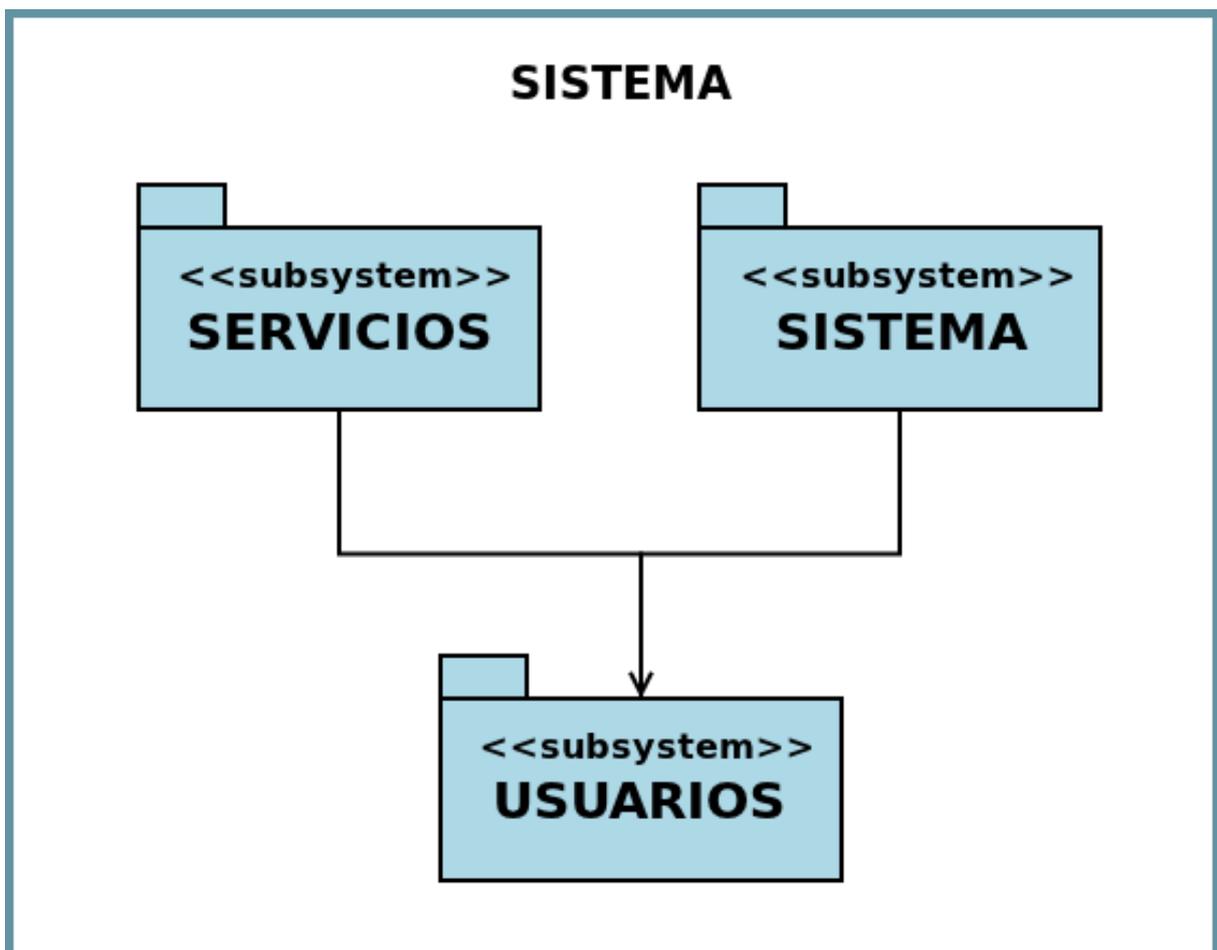


Figura 5.1: Diagrama de subsistemas

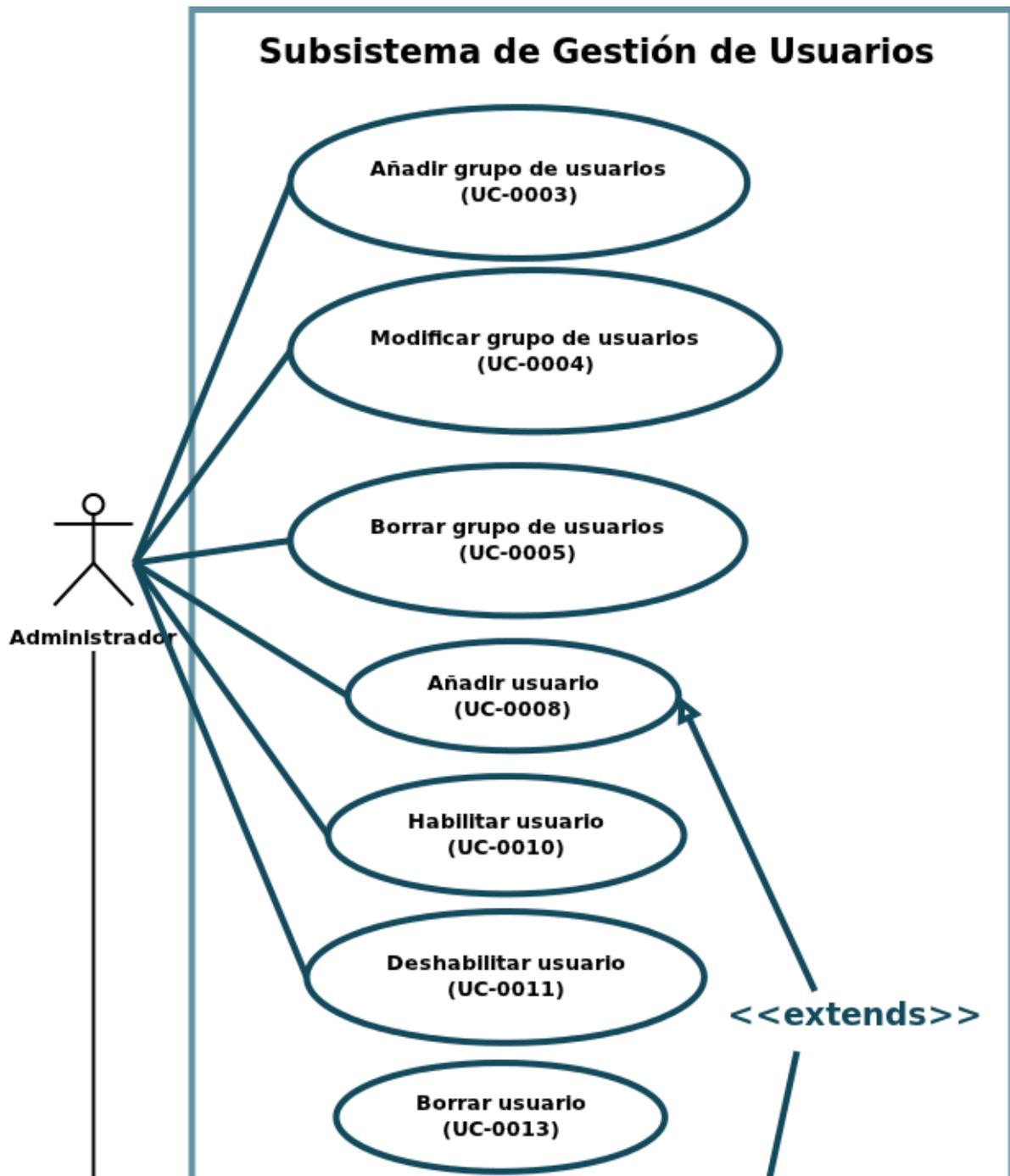


Figura 5.2: Diagrama de casos de uso del Subsistema de Gestión de Usuarios (1)

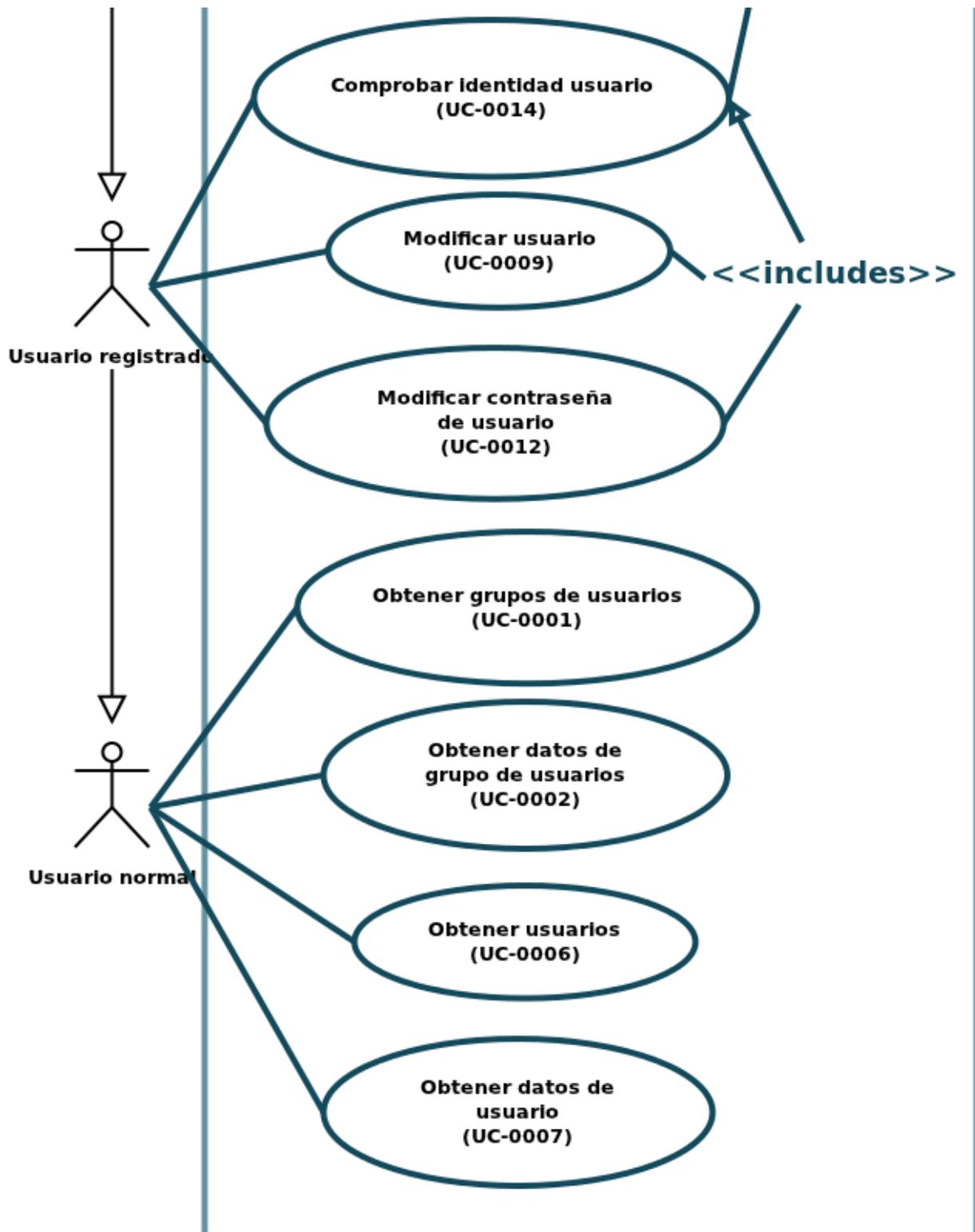


Figura 5.3: Diagrama de casos de uso del Subsistema de Gestión de Usuarios (2)

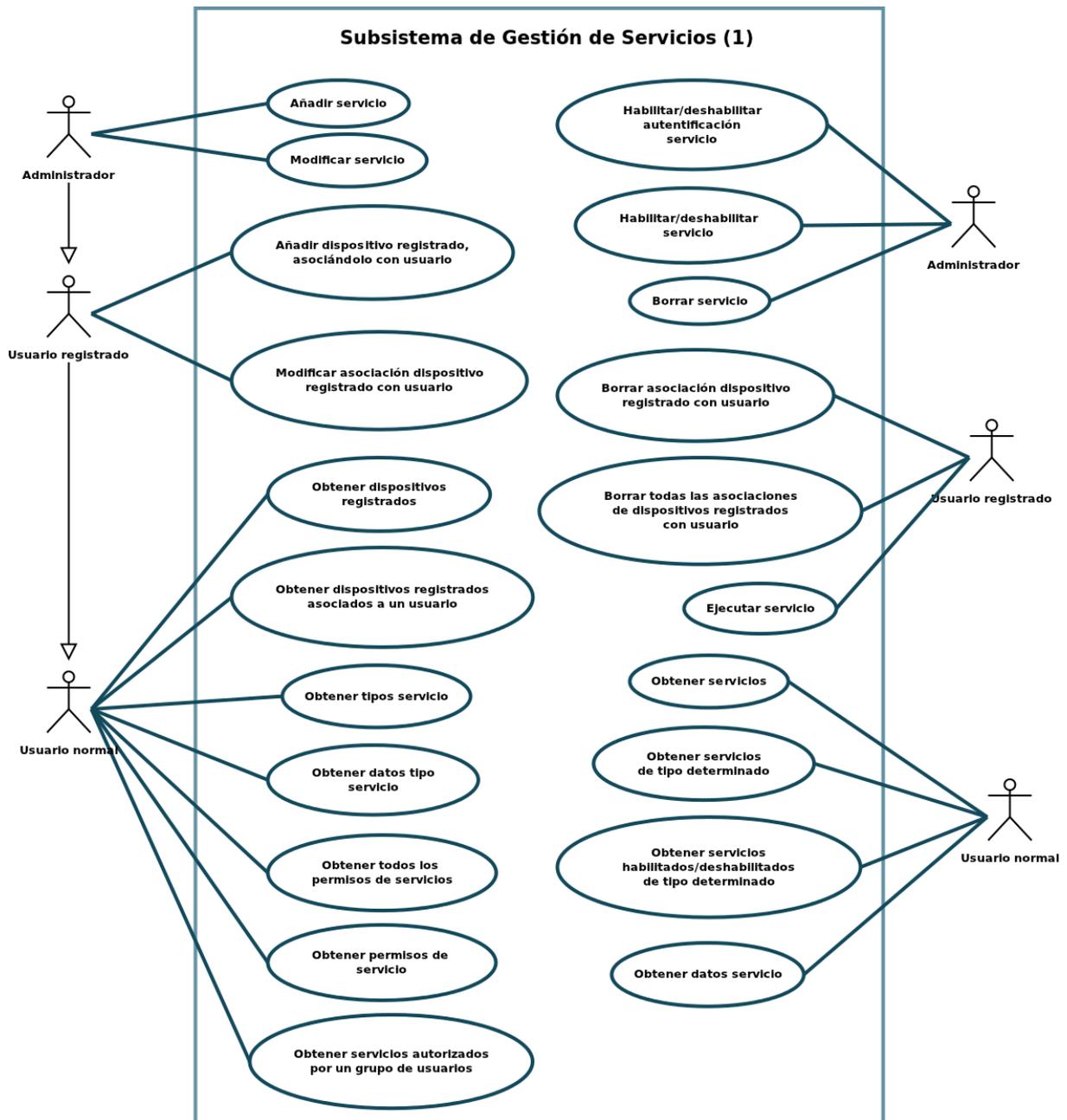


Figura 5.4: Diagrama de casos de uso del Subsistema de Gestión de Servicios (1)

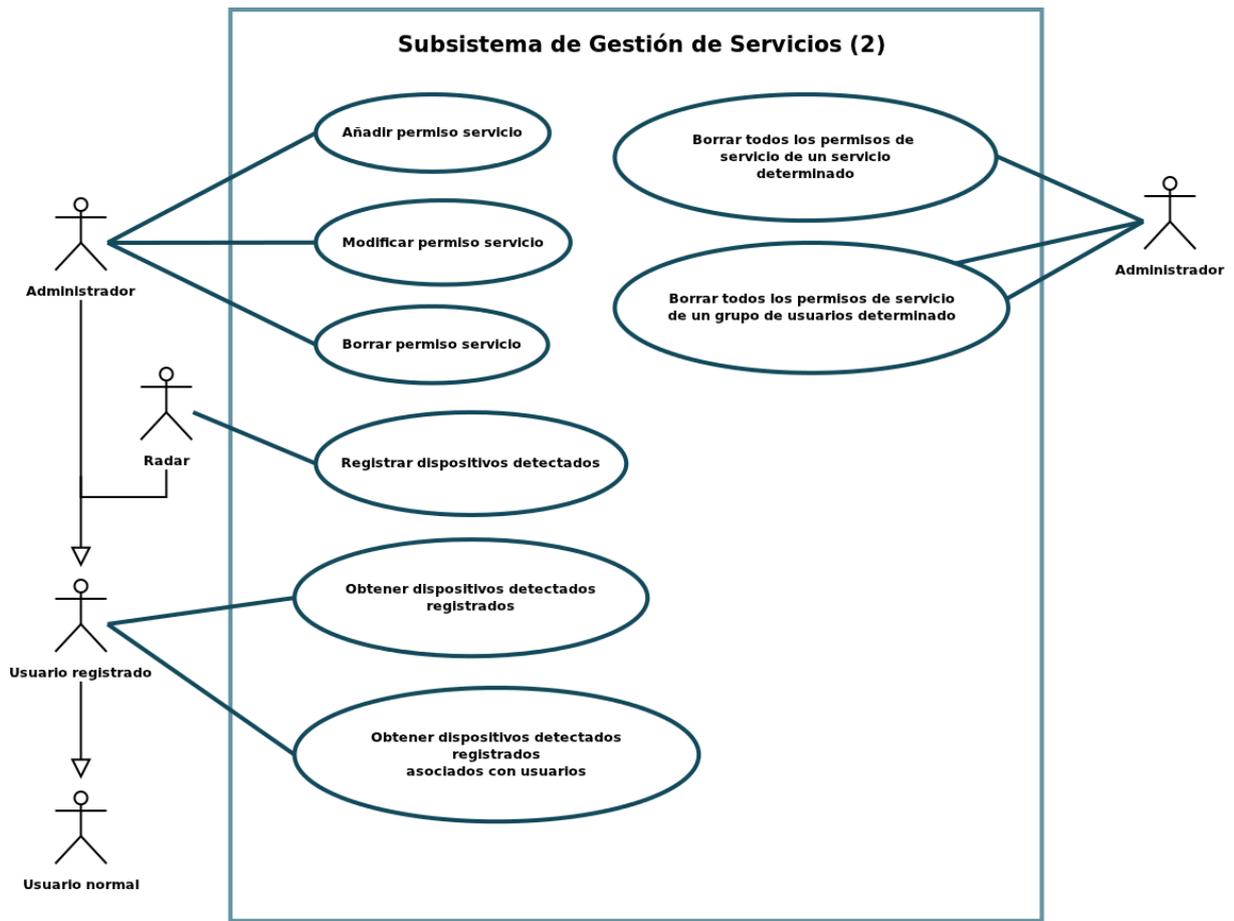


Figura 5.5: Diagrama de casos de uso del Subsistema de Gestión de Servicios (2)

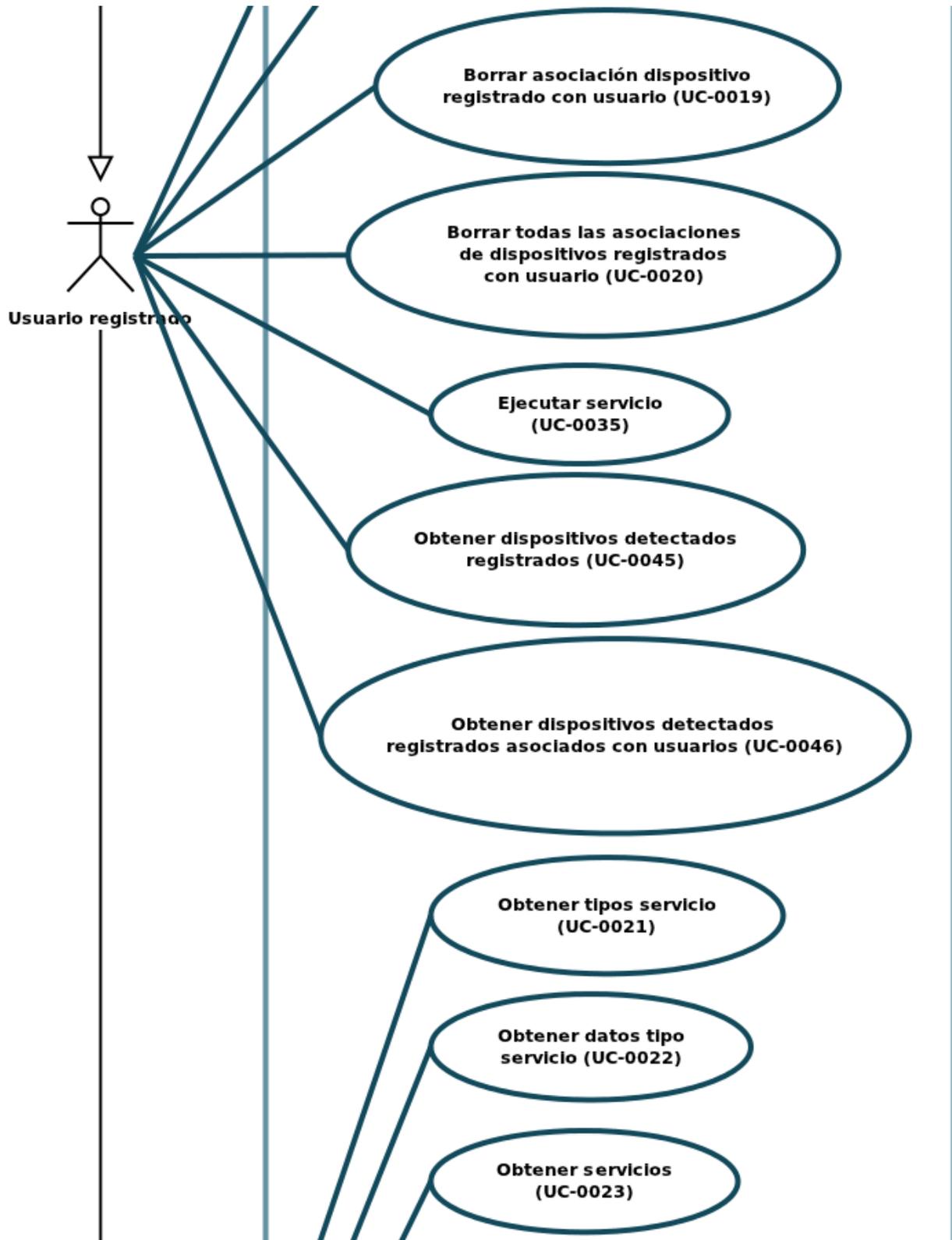


Figura 5.6: Diagrama de casos de uso del Subsistema de Gestión de Servicios (3)

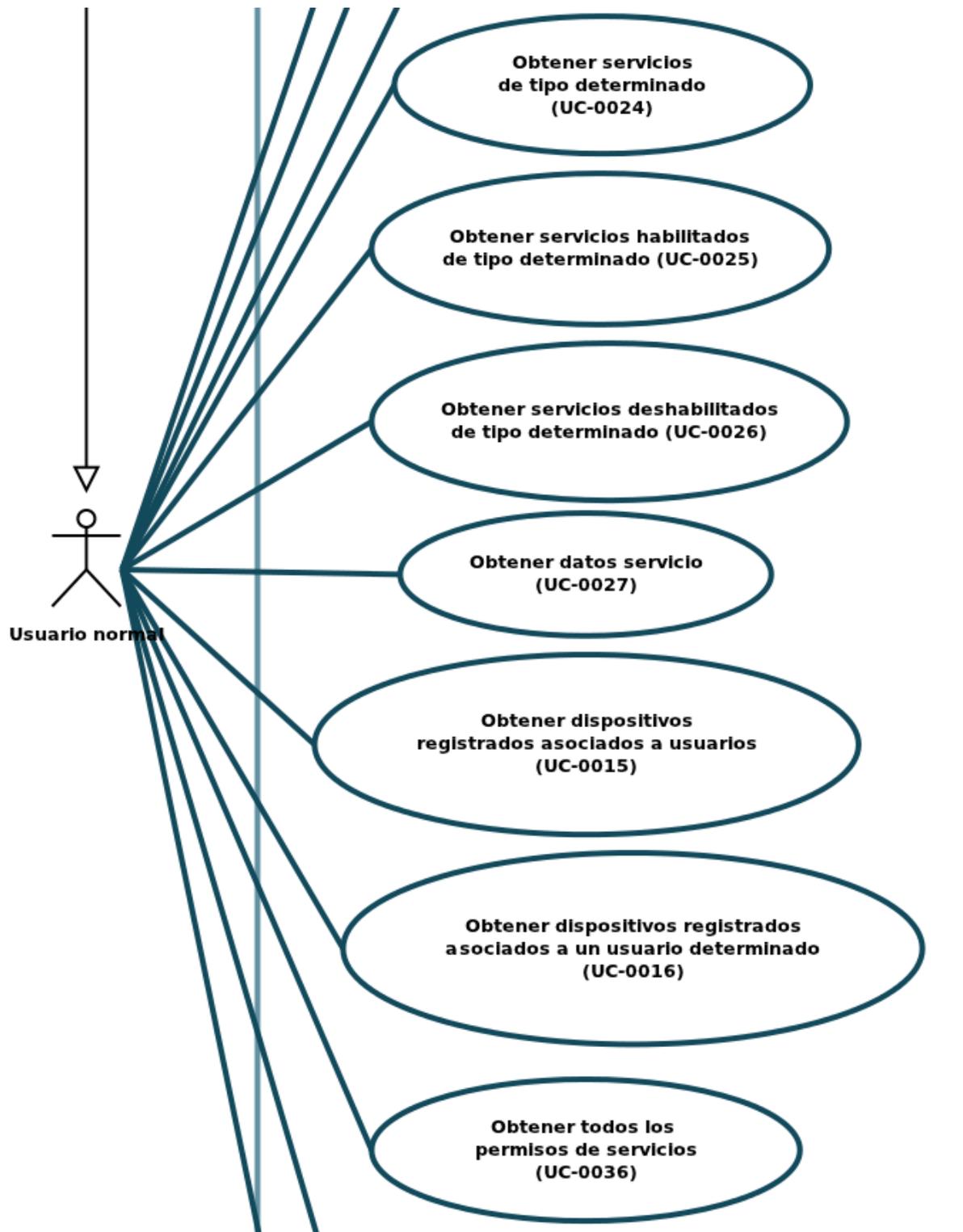


Figura 5.7: Diagrama de casos de uso del Subsistema de Gestión de Servicios (4)

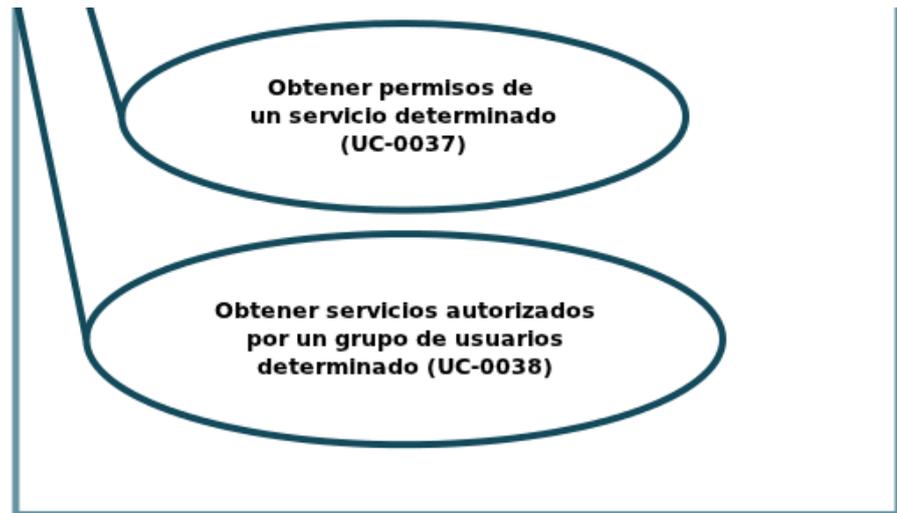


Figura 5.8: Diagrama de casos de uso del Subsistema de Gestión de Servicios (5)

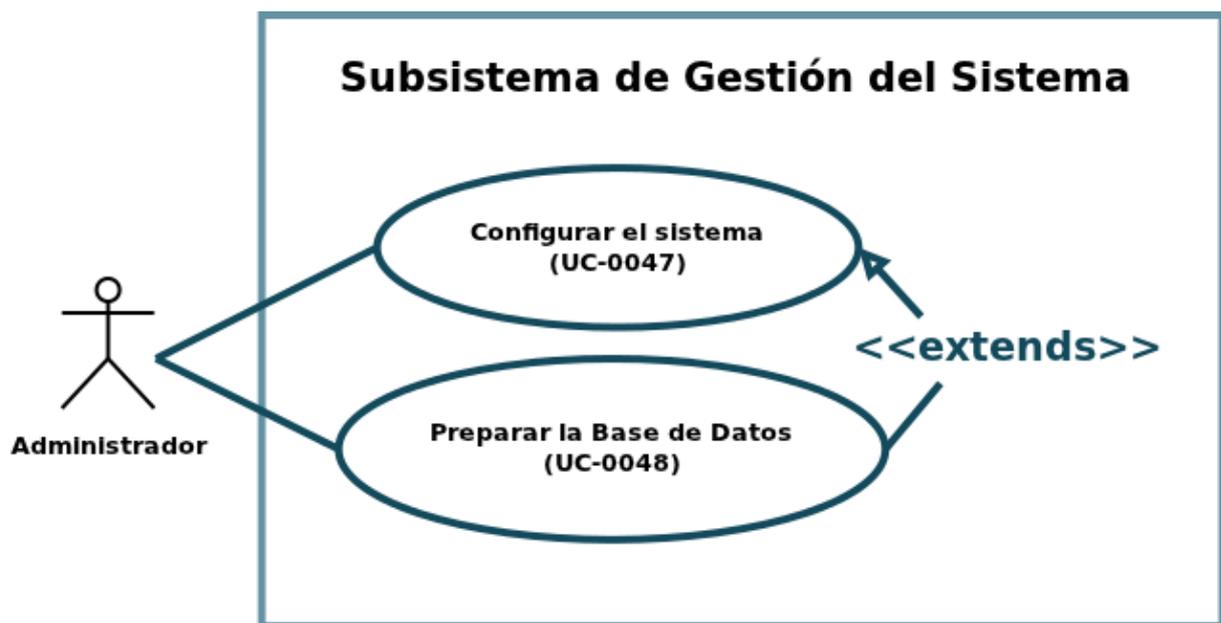


Figura 5.9: Diagrama de casos de uso del Subsistema de Gestión de Sistema

5.4.3. Casos de uso del sistema

Ya hemos visto los diagramas de los casos de uso del sistema. Ahora, vamos a definirlos en detalle.

Subsistema de Gestión de Usuarios

UC-0001	Obtener todos los grupos de usuarios
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida obtener un listado de todos los grupos de usuarios.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema el listado de grupos de usuarios. 2. El sistema muestra el listado de grupos de usuarios.

UC-0002	Obtener los datos sobre un grupo de usuarios
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida obtener todos los datos de un grupo de usuarios determinado.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema los datos de un grupo de usuarios determinado, proporcionando al sistema el identificador de dicho grupo. 2. El sistema muestra los datos del grupo de usuarios.

UC-0003	Añadir un grupo de usuarios
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida añadir un grupo de usuarios.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). En el sistema no debe estar registrado ningún grupo de usuarios con el mismo identificador que el del grupo nuevo que se pretende añadir.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo añadir un grupo de usuarios, proporcionando al sistema los datos del nuevo grupo. 2. El sistema añade el nuevo grupo de usuarios y, en caso de error, lo notifica al administrador.
Postcondición	El nuevo grupo de usuarios queda registrado en el sistema.

UC-0004	Modificar un grupo de usuarios
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida modificar un grupo de usuarios.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). En el sistema no debe estar registrado ningún grupo de usuarios con el mismo identificador que el nuevo identificador que se pretende establecer para el grupo de usuario.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo modificar un grupo de usuarios, proporcionando al sistema su identificador y sus nuevos datos. 2. El sistema modifica el grupo de usuarios y, en caso de error, lo notifica al administrador.
Postcondición	El grupo de usuarios queda modificado en el sistema con los nuevos datos.

UC-0005	Borrar un grupo de usuarios
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida borrar un grupo de usuarios.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El usuario que se pretende borrar debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo borrar un grupo de usuarios, proporcionando al sistema el identificador de dicho grupo. 2. El sistema borra el grupo de usuarios y, en caso de error, lo notifica al administrador.
Postcondición	El grupo de usuarios queda borrado del sistema.

UC-0006	Obtener todos los usuarios
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida obtener un listado de todos los usuarios.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema el listado de usuarios. 2. El sistema muestra el listado de usuarios.

UC-0007	Obtener los datos de un usuario
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario decida obtener todos los datos de un usuario determinado.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema los datos de un usuario determinado, proporcionando al sistema el identificador de dicho usuario. 2. El sistema muestra los datos del usuario.

UC-0008	Añadir un usuario
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida añadir un usuario.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). No debe haber ningún usuario registrado con el mismo identificador que el del usuario que se pretende añadir.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo añadir un usuario, proporcionando al sistema los datos del nuevo usuario. 2. El sistema añade el nuevo usuario y, en caso de error, lo notifica al administrador.
Postcondición	El nuevo usuario queda registrado en el sistema.

UC-0009	Modificar un usuario
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida modificar un usuario.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El usuario que se pretende modificar debe estar registrado en el sistema. En el sistema no debe estar registrado ningún usuario con el mismo identificador que el nuevo identificador que se pretende establecer para el usuario.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo modificar un usuario, proporcionando al sistema su identificador y sus nuevos datos (excepto su contraseña). 2. El sistema modifica el usuario (sin cambiar su contraseña) y, en caso de error, lo notifica al administrador.
Postcondición	El usuario queda modificado en el sistema con los nuevos datos (su contraseña no queda cambiada).

UC-0010	Habilitar un usuario
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida habilitar un usuario.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El usuario que se pretende habilitar debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo habilitar un usuario, proporcionando al sistema el identificador de dicho usuario. 2. El sistema habilita el usuario (permitiendo así que pueda ejecutar los servicios para los que esté autorizado) y, en caso de error, lo notifica al administrador.
Postcondición	El usuario queda habilitado en el sistema y puede ejecutar los servicios para los que esté autorizado.

UC-0011	Deshabilitar un usuario
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida deshabilitar un usuario.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El usuario que se pretende deshabilitar debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo deshabilitar un usuario, proporcionando al sistema el identificador de dicho usuario. 2. El sistema deshabilita el usuario (impidiendo así que pueda ejecutar servicios) y, en caso de error, lo notifica al administrador.
Postcondición	El usuario queda deshabilitado en el sistema y no puede ejecutar ningún servicio.

UC-0012	Modificar la contraseña de un usuario
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida modificar la contraseña de un usuario.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El usuario cuya contraseña se pretende modificar debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo modificar la contraseña de un usuario, proporcionando al sistema el identificador de dicho usuario y su nueva contraseña. 2. El sistema modifica la contraseña del usuario y, en caso de error, lo notifica al administrador.
Postcondición	La contraseña del usuario queda modificada.

UC-0013	Borrar un usuario
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida borrar un usuario.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El usuario que se pretende borrar debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo borrar un usuario, proporcionando al sistema el identificador de dicho usuario. 2. El sistema borra el usuario y, en caso de error, lo notifica al administrador.
Postcondición	El usuario queda borrado del sistema.

UC-0014	Comprobar la identidad de un usuario determinado
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida comprobar la identidad de un usuario (comprobar si un par usuario-contraseña es válido).
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El usuario debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo comprobar la identidad de un usuario, proporcionando su identificador y su contraseña. 2. El sistema notifica al administrador si la contraseña es correcta.

Subsistema de Gestión de Servicios

UC-0015	Obtener todos los dispositivos registrados asociados a usuarios
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida obtener un listado de todos dispositivos registrados asociados a usuarios.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema el listado de dispositivos registrados asociados a usuarios. 2. El sistema muestra el listado de dispositivos.

UC-0016	Obtener todos los dispositivos registrados asociados a un usuario determinado
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida obtener un listado de todos los dispositivos registrados asociados a un usuario determinado.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El usuario cuyos dispositivos se pretenden obtener debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema el listado de dispositivos registrados asociados a un usuario determinado, proporcionando al sistema el identificador del usuario. 2. El sistema muestra el listado de dispositivos asociados al usuario.

UC-0017	Añadir un nuevo dispositivo asociándolo a un usuario
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida añadir un nuevo dispositivo y asociarlo con un usuario determinado.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El usuario con el que se pretende asociar un nuevo dispositivo debe estar registrado en el sistema. El dispositivo a añadir no debe estar registrado previamente en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo añadir un nuevo dispositivo, asociándolo con un usuario determinado, proporcionando al sistema el identificador del usuario y los datos del nuevo dispositivo. 2. El sistema añade el nuevo dispositivo, asociándolo con el usuario y, en caso de error, lo notifica al administrador.
Postcondición	El nuevo dispositivo registrado en el sistema.

UC-0018	Modificar la asociación de un dispositivo con un usuario
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida modificar la asociación de un dispositivo concreto con un usuario determinado cambiando en dicha asociación el dispositivo actual por otro.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El usuario con el que se pretende asociar el nuevo dispositivo debe estar registrado en el sistema. El dispositivo a reemplazar de la asociación y el nuevo dispositivo deben estar registrados en el sistema. La asociación a modificar no debe existir previamente en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo modificar una asociación de un dispositivo concreto con un usuario determinado, proporcionando al sistema el identificador del usuario, el identificador del dispositivo a reemplazar y el identificador del nuevo dispositivo. 2. El sistema modifica la asociación del dispositivo con el usuario y, en caso de error, lo notifica al administrador.
Postcondición	La asociación queda modificada en el sistema.

UC-0019	Borrar una asociación de un dispositivo con un usuario
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida borrar una asociación de un dispositivo concreto con un usuario determinado.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El usuario y el dispositivo cuya asociación se pretende borrar deben estar registrados en el sistema. La asociación que se pretende borrar debe estar registrada previamente en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo borrar una asociación de un dispositivo concreto con un usuario determinado, proporcionando al sistema el identificador del usuario y el identificador del dispositivo. 2. El sistema borra la asociación y, en caso de error, lo notifica al administrador.
Postcondición	La asociación queda borrada del sistema y si el dispositivo de dicha asociación ya no está asociado a ningún otro usuario, éste queda borrado del sistema.

UC-0020	Borrar todas las asociaciones de dispositivos con un usuario
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida borrar todas las asociaciones de dispositivos con un usuario determinado.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El usuario del cual se pretenden borrar todas sus asociaciones con dispositivos debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo borrar todas las asociaciones de dispositivos con un usuario determinado, proporcionando al sistema el identificador del usuario. 2. El sistema borra todas las asociaciones de dispositivos con el usuario.
Postcondición	Las asociaciones quedan borradas del sistema y si los dispositivos de dichas asociaciones ya no están asociados a ningún otro usuario, éstos quedan borrados del sistema.

UC-0021	Obtener todos los tipos de servicios
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida obtener un listado de todos los tipos de servicio.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema el listado de tipos de servicio. 2. El sistema muestra el listado de tipos de servicio.

UC-0022	Obtener los datos de un tipo de servicio
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario decida obtener todos los datos de un tipo de servicio.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema los datos de un tipo de servicio, proporcionando al sistema el identificador de dicho tipo de servicio. 2. El sistema muestra los datos del tipo de servicio.

UC-0023	Obtener todos los servicios
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida obtener un listado de todos los servicios.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema el listado de servicios. 2. El sistema muestra el listado de servicios.

UC-0024	Obtener todos los servicios de un tipo determinado
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida obtener un listado de todos los servicios de un tipo determinado.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema el listado de servicios de un tipo determinado, proporcionando el identificador de dicho tipo de servicios. 2. El sistema muestra el listado de servicios del tipo especificado.

UC-0025	Obtener todos los servicios habilitados de un tipo determinado
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida obtener un listado de todos los servicios de un tipo determinado que estén habilitados.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema el listado de servicios de un tipo determinado que estén habilitados, proporcionando el identificador de dicho tipo de servicios. 2. El sistema muestra el listado de servicios habilitados del tipo especificado.

UC-0026	Obtener todos los servicios deshabilitados de un tipo determinado
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida obtener un listado de todos los servicios de un tipo determinado que estén deshabilitados.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema el listado de servicios de un tipo determinado que estén deshabilitados, proporcionando el identificador de dicho tipo de servicios. 2. El sistema muestra el listado de servicios deshabilitados del tipo especificado.

UC-0027	Obtener todos los datos de un servicio
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario decida obtener todos los datos de un servicio.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema los datos de un servicio determinado, proporcionando al sistema el identificador de dicho servicio. 2. El sistema muestra los datos del servicio.

UC-0028	Añadir un servicio
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida añadir un servicio.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). En el sistema no debe estar registrado ningún servicio con el mismo identificador que el del servicio nuevo que se pretende añadir.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo añadir un servicio, proporcionando al sistema los datos del nuevo servicio. 2. El sistema añade el nuevo servicio y, en caso de error, lo notifica al administrador.
Postcondición	El nuevo servicio queda registrado en el sistema.

UC-0029	Modificar un servicio
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida modificar un servicio.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El servicio que se pretende modificar debe estar registrado en el sistema. En el sistema no debe estar registrado ningún servicio con el mismo identificador que el nuevo identificador que se pretende establecer para el servicio.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo modificar un servicio, proporcionando su identificador y sus nuevos datos (excepto su contraseña). 2. El sistema modifica el servicio y, en caso de error, lo notifica al administrador.
Postcondición	El servicio queda modificado en el sistema con los nuevos datos.

UC-0030	Habilitar el requerimiento de autenticación en un servicio
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida habilitar la autenticación en un servicio.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El servicio cuya autenticación se pretende habilitar debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo habilitar la autenticación en un servicio determinado, proporcionando al sistema el identificador de dicho servicio. 2. El sistema habilita la autenticación en dicho servicio (haciendo que, para ejecutar el servicio, el usuario deba autenticarse) y, en caso de error, lo notifica al administrador.
Postcondición	La autenticación en el servicio queda habilitada.

UC-0031	Deshabilitar el requerimiento de autenticación en un servicio
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida deshabilitar la autenticación en un servicio.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El servicio cuya autenticación se pretende deshabilitar debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo deshabilitar la autenticación en un servicio determinado, proporcionando al sistema el identificador de dicho servicio. 2. El sistema deshabilita la autenticación en dicho servicio (haciendo que, para ejecutar el servicio, el usuario no deba autenticarse) y, en caso de error, lo notifica al administrador.
Postcondición	La autenticación en el servicio queda deshabilitada.

UC-0032	Habilitar un servicio
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida habilitar un servicio.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El servicio que se pretende habilitar debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo habilitar un servicio determinado, proporcionando al sistema el identificador de dicho servicio. 2. El sistema habilita dicho servicio (haciendo que sea ejecutable por los usuarios) y, en caso de error, lo notifica al administrador.
Postcondición	El servicio queda habilitado.

UC-0033	Deshabilitar un servicio
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida deshabilitar un servicio.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El servicio que se pretende deshabilitar debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo deshabilitar un servicio determinado, proporcionando al sistema el identificador de dicho servicio. 2. El sistema deshabilita dicho servicio (haciendo que no sea ejecutable por ningún usuario) y, en caso de error, lo notifica al administrador.
Postcondición	El servicio queda deshabilitado.

UC-0034	Borrar un servicio
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida borrar un servicio.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El servicio que se pretende borrar debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo borrar un servicio, proporcionando al sistema el identificador de dicho servicio. 2. El sistema borra el servicio y, en caso de error, lo notifica al administrador.
Postcondición	El servicio queda borrado del sistema.

UC-0035	Ejecutar un servicio
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida ejecutar un servicio.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El servicio que se pretende ejecutar debe estar registrado en el sistema y el usuario debe estar autorizado a ejecutarlo.
Secuencia	<ol style="list-style-type: none"> 1. El usuario del sistema solicita al mismo ejecutar un servicio, proporcionando al sistema el identificador de dicho servicio. 2. Si el servicio requiere autenticación del usuario: <ol style="list-style-type: none"> a) El sistema solicita al usuario su identificador y su contraseña. b) El usuario proporciona al sistema su identificador y su contraseña. 3. El sistema ejecuta el servicio, devolviendo su resultado al usuario si procede y, en caso de error, lo notifica al usuario.
Postcondición	El servicio ha sido ejecutado.

UC-0036	Obtener todos los permisos de todos los servicios
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida obtener un listado de todos los permisos (grupos de usuarios autorizados para la ejecución) de todos los servicios.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema el listado de permisos de los servicios. 2. El sistema muestra el listado de permisos.

UC-0037	Obtener todos permisos de un servicio determinado
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida obtener un listado de todos los permisos (grupos de usuarios autorizados para la ejecución) de un servicio.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El servicio debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema el listado de permisos de un servicio determinado, proporcionando el identificador del servicio. 2. El sistema muestra el listado de permisos.

UC-0038	Obtener todos los servicios cuyos permisos contengan (al menos, 1 de ellos) a un grupo de usuarios determinado
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida obtener un listado de todos los servicios cuyos permisos (al menos, 1 de ellos) contengan a un grupo de usuarios determinado.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El grupo de usuarios debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema el listado de servicios cuyos permisos contengan a un grupo de usuarios determinado, proporcionando el identificador del grupo de usuarios. 2. El sistema muestra el listado de servicios.

UC-0039	Añadir un permiso de servicio
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida añadir un permiso de servicio (par servicio-grupo de usuarios).
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). En el sistema no debe estar registrado ningún permiso de servicio (par servicio-grupo de usuarios) idéntico al permiso nuevo que se pretende añadir. El servicio y el grupo de usuarios asociados al permiso que se pretende añadir deben estar registrados en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo añadir un permiso de servicio, proporcionando al sistema los datos del nuevo permiso (identificador del servicio e identificador del grupo de usuarios). 2. El sistema añade el nuevo permiso y, en caso de error, lo notifica al administrador.
Postcondición	El nuevo permiso de servicio queda registrado en el sistema.

UC-0040	Modificar un permiso de servicio
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida modificar un permiso de servicio (par servicio-grupo de usuarios).
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El permiso de servicio (par servicio-grupo de usuarios) que se pretende modificar debe estar registrado en el sistema. El servicio y el grupo de usuarios nuevos asociados al permiso deben estar registrados en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo modificar un permiso de servicio, proporcionando al sistema los datos actuales del permiso y los nuevos datos del permiso (los datos son el identificador del servicio y el identificador del grupo de usuarios). 2. El sistema modifica el permiso y, en caso de error, lo notifica al administrador.
Postcondición	El permiso de servicio queda modificado en el sistema.

UC-0041	Borrar un permiso de servicio
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida borrar un permiso de servicio (par servicio-grupo de usuarios).
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El permiso (par servicio- grupo de usuarios) que se pretende borrar debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo borrar un permiso de servicio, proporcionando al sistema los datos del permiso (el identificador del servicio y el identificador del grupo de usuarios). 2. El sistema borra el permiso y, en caso de error, lo notifica al administrador.
Postcondición	El permiso de servicio queda borrado del sistema.

UC-0042	Borrar todos los permisos de servicio de un servicio determinado
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida borrar todos los permisos de servicio (par servicio-grupo de usuarios) de un servicio determinado.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El servicio cuyos permisos se pretende borrar debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo borrar todos los permisos de un servicio, proporcionando al sistema el identificador del servicio. 2. El sistema borra todos los permisos del servicio y, en caso de error, lo notifica al administrador.
Postcondición	Todos los permisos de servicio del servicio quedan borrados del sistema.

UC-0043	Borrar todos los permisos de servicio de un grupo de usuarios determinado
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida borrar todos los permisos de servicio (por servicio-grupo de usuarios) de un grupo de usuarios determinado.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>). El grupo de usuarios cuyos permisos se pretende borrar debe estar registrado en el sistema.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo borrar todos los permisos de un grupo de usuarios, proporcionando al sistema el identificador del grupo. 2. El sistema borra todos los permisos del grupo y, en caso de error, lo notifica al administrador.
Postcondición	Todos los permisos de servicio del grupo de usuarios quedan borrados del sistema.

UC-0044	Registrar dispositivos detectados por el sistema
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario especial autorizado del sistema decida registrar los dispositivos detectados por el sistema.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario especial del sistema solicita al mismo registrar los dispositivos detectados por el sistema proporcionándole el identificador y el identificador del tipo de dispositivo de cada uno de los dispositivos. 2. El sistema registra los dispositivos.
Postcondición	Los dispositivos detectados por el sistema quedan registrados.

UC-0045	Obtener todos los dispositivos detectados por el sistema
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida obtener un listado de todos los dispositivos detectados por el sistema.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema el listado de dispositivos detectados por el sistema. 2. El sistema muestra el listado de dispositivos.

UC-0046	Obtener todos los dispositivos detectados por el sistema que estén asociados con, al menos, 1 usuario del sistema
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario del sistema decida obtener un listado de todos los dispositivos detectados por el sistema que estén asociados con, al menos, 1 usuario del sistema.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>) y debe tener la base de datos instalada (ver caso de uso <i>Preparar la base de datos del sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El usuario solicita al sistema el listado de dispositivos detectados por el sistema que estén asociados a usuarios del sistema. 2. El sistema muestra el listado de dispositivos.

Subsistema de Gestión de Sistema

UC-0047	Configurar el sistema
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida configurar el mismo.
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo configurarlo. 2. El sistema solicita al administrador los valores de cada uno de los parámetros de la configuración. 3. El administrador proporciona los valores de los parámetros de la configuración. 4. El sistema graba la configuración con los datos que ha proporcionado el administrador.
Postcondición	El sistema queda configurado.

UC-0048	Preparar la base de datos del sistema
Versión	1.0 (15/03/2010)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador del sistema decida preparar la base de datos del sistema.
Precondición	El sistema debe estar configurado (ver caso de uso <i>Configurar el sistema</i>).
Secuencia	<ol style="list-style-type: none"> 1. El administrador del sistema solicita al mismo preparar la base de datos. 2. El sistema instala todas las tablas, restricciones y datos iniciales de la base de datos del sistema en su estado inicial.
Postcondición	La base de datos del sistema queda preparada y el sistema ya puede manejar datos.

5.5. Requisitos no funcionales

Establezcamos ahora los requisitos no funcionales que debe cumplir el sistema. Según Wikipedia[14], un requisito no funcional es, en la ingeniería de sistemas y la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que ni describen información a guardar, ni funciones a realizar.

NFR-0001: Software libre

El sistema debe ser licenciado como software libre, permitiendo su integración en sistemas propietarios, por lo que su licencia debe ser LGPL.

NFR-0002: Bibliotecas software libres

Todas las bibliotecas que use el sistema deben ser software libre.

NFR-0003: Sistema para Linux

El sistema debe estar diseñado para sistemas operativos Linux.

NFR-0004: Acceso al sistema

El sistema debe ser accesible por los usuarios desde cualquier plataforma hardware/software.

NFR-0005: Facilidad de uso y configuración

El sistema debe ser gestionable y configurable por medio de una interfaz de usuario que sea fácil de utilizar.

NFR-0006: Acceso remoto para el administrador

El sistema debe ser accesible por el administrador de forma remota por medio de una conexión segura, para gestionarlo.

NFR-0007: Sistema de plugins de servicios

El sistema deberá soportar un sistema de plugins para los servicios ofrecidos, en el que cada servicio es implementado por un plugin.

NFR-0008: Seguridad para identificación de usuarios

El sistema debe implementar algún mecanismo de seguridad para verificar la identidad de cada usuario registrado, por medio de un nombre de usuario y una contraseña.

NFR-0009: Cifrado de contraseñas de usuarios

Todas las contraseñas de usuarios que almacene el sistema deben estar cifradas por su suma de comprobación MD5.

NFR-0010: Cifrado de información

El sistema debe permitir el cifrado de la información que circule por el mismo.

NFR-0011: Legibilidad de errores registrados

Los errores ocurridos en el sistema que se registren deben estar guardados en un formato legible fácilmente para el administrador.

Capítulo 6

Arquitectura

6.1. Sistemas distribuidos

Un sistema distribuido se define como una colección de computadoras separadas físicamente y conectadas entre sí por una red de comunicaciones distribuida; cada máquina posee sus componentes de hardware y software que el usuario percibe como un solo sistema (no necesita saber qué cosas están en qué máquinas). El usuario accede a los recursos remotos (RPC) de la misma manera en que accede a recursos locales, o un grupo de computadores que usan un software para conseguir un objetivo en común.

Los sistemas distribuidos deben ser muy confiables, ya que si un componente del sistema se descompone otro componente debe de ser capaz de reemplazarlo, esto se denomina Tolerancia a Fallos.

El tamaño de un sistema distribuido puede ser muy variado, ya sean decenas de hosts (red de área local), centenas de hosts (red de área metropolitana), y miles o millones de hosts (Internet); esto se denomina escalabilidad.

La computación distribuida ha sido diseñada para resolver problemas demasiado grandes para cualquier supercomputadora y mainframe, mientras se mantiene la flexibilidad de trabajar en múltiples problemas más pequeños. Por lo tanto, la computación en grid es naturalmente un entorno multiusuario; por ello, las técnicas de autorización segura son esenciales antes de permitir que los recursos informáticos sean controlados por usuarios remotos.

Fuente: Wikipedia, <http://es.wikipedia.org>

6.2. Arquitectura Cliente-Servidor

Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la

centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

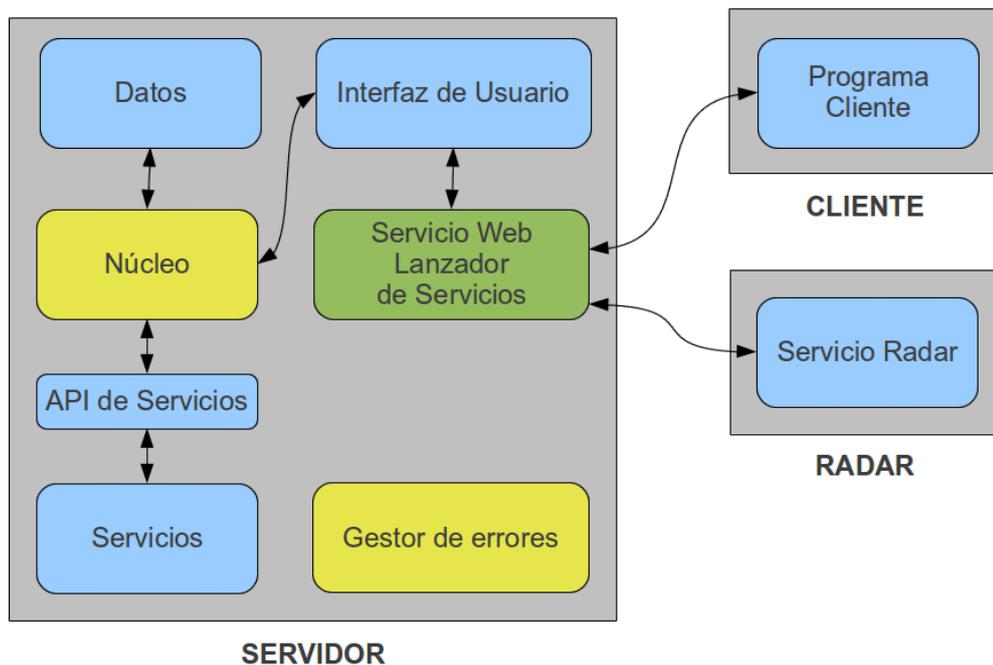
La arquitectura cliente-servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico.

La red cliente-servidor es aquella red de comunicaciones en la que todos los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta y que los pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de sólo lectura y los que, por el contrario, pueden ser modificados, etc. Este tipo de red puede utilizarse conjuntamente en caso de que se este utilizando en una red mixta.

Fuente: Wikipedia, <http://es.wikipedia.org>

6.3. Arquitectura de Predesys

A continuación, se expone una introducción a la arquitectura del proyecto:



Como puede verse en la figura, el soporte físico para PredeSYS consta de un Servidor (un PC), 1 ó más Radares (PCs) y 1 ó más Clientes (PCs o dispositivos móviles).

PredeSYS es un sistema que detecta dispositivos (bluetooth, Ethernet, WiFi...) por medio del servicio Radar y ejecuta tareas automáticas cuando se detectan dichos dispositivos y otras tareas que son ejecutables por los usuarios mediante el Servicio Web de Acceso a Servicios. Estas tareas se definen por medio de servicios, que son plugins que tiene instalados el Servidor.

Globalmente, hay 2 tipos de servicios, los servicios llamables y los automáticos. Los servicios automáticos sólo los ejecuta el servicio Radar, que lo hace cada cierto tiempo y los llamables los ejecuta el usuario (el programa Cliente) cuando quiera. Todos los servicios se ejecutan por medio del Servicio Web de Acceso a Servicios y pueden requerir autenticación por medio de un nombre de usuario y una contraseña (en el caso de los servicios automáticos se requiere autenticación ya que sólo una máquina Radar debería poder ejecutarlos).

Los servicios automáticos pueden, además de realizar alguna tarea, al finalizar ésta, enviar una orden al servicio Radar para que envíe algún mensaje con cualquier información a los dispositivos detectados (según el tipo de servicio Radar usado, a través de Internet, de bluetooth, etc).

El Servidor contiene toda la lógica de los servicios y los datos de los mismos. Todo el control del Servidor lo ejerce el Núcleo, que es el componente principal, y es el que ejecuta

los servicios (que son scripts que realizan una determinada tarea). Los Servicios pueden acceder a los Datos pero no pueden modificarlos (si requirieran guardar datos, deberían gestionar su propia base de datos). El Núcleo es el intermediario entre los Servicios y los Datos, ya que debe controlar y asegurar el correcto funcionamiento del sistema y limitar los datos a los que pueda acceder cada parte del sistema. La forma en la que los Servicios acceden a los Datos y a cualquier funcionalidad del sistema es a través de la API de Servicios, que provee varias funciones limitadas que acceden al Núcleo.

Los Radares pueden ser 1 ó más, todas con conectividad por TCP/IP con el Servidor. Tienen tan sólo 1 función muy básica aunque muy importante. Esta función es detectar periódicamente la presencia de dispositivos que estén físicamente cerca de ellos y enviar sus direcciones MAC al Servidor, a través de uno de los Servicios (en este caso, un servicio especial de acceso restringido a los Radares) para ejecutar a su vez todos los servicios automáticos que estén instalados en el Servidor y, en el caso de que alguno lo especifique, enviar mensajes a dichos dispositivos.

El Gestor de Errores se encarga de registrar todos los errores que ocurran en el Servidor, para que éstos sean localizables fácilmente.

Capítulo 7

Implementación

- 7.1. Subsistema de gestión de usuarios
- 7.2. Subsistema de gestión de servicios
- 7.3. Subsistema de gestión de errores

Capítulo 8

Pruebas

Para probar el proyecto se han tenido en mente la realización de pruebas unitarias para cada una de las funciones y métodos del sistema, especialmente del Núcleo del Servidor, que es la parte más crítica y por tanto, la que requiere una mayor fiabilidad.

También han tenido especial relevancia pruebas de rendimiento del sistema y de la detección de dispositivos.

Capítulo 9

Distribución

Los componentes principales de Predesys (Servidor, API de Servicios, Radar Bluetooth y Radar LAN) así como los servicios de Predesys (Presence People) se distribuyen en 2 archivos de formatos distintos por cada uno de ellos. Los formatos son el formato de *paquete tarball* (con extensión *.tar.gz*) y formato de *paquete Debian* (con extensión *.deb*).

El *paquete tarball* es un archivo comprimido que contiene el programa junto a un script de instalación. Tiene la ventaja de que puede instalarse en cualquier sistema Linux pero tiene la desventaja de no instalar las dependencias que no estén instaladas en el sistema, es decir, el software necesario para hacer funcionar al programa.

El *paquete Debian* es un archivo que contiene el programa listo para instalar en cualquier sistema operativo Linux tipo Debian (por ejemplo, el propio Debian o Ubuntu). Tiene la ventaja de que es muy fácil de instalar y comprueba todas las dependencias (descargándolas e instalándolas automáticamente si el sistema no las tuviera instaladas) pero tiene la desventaja de que, en principio, sólo puede instalarse en los sistemas operativos Linux de tipo Debian.

Para instalar de una sólo vez todas las dependencias del proyecto (programas y bibliotecas que el proyecto requiere), puede ejecutar (como administrador), desde el directorio raíz del repositorio del código fuente del proyecto, el script llamado *install-dependencies.sh*. También, puede generar todos paquetes de todos los componentes de Predesys ejecutando, desde el directorio raíz del repositorio del código fuente del proyecto, el script llamado *build.sh*, aunque más adelante (en este mismo capítulo) se explica como generar cada paquete individualmente.

Para generar cualquiera de los paquetes, es necesario tener los siguientes paquetes de software instalados en el sistema operativo (los nombres son los establecidos en los sistemas Linux tipo Debian, en otros sistemas pueden ser distintos):

- tar (versión igual o superior a la 1.23)
- build-essential (versión igual o superior a la 11.5)
- devscripts (versión igual o superior a la 2.10.67)
- lintian (versión igual o superior a la 2.4.3)
- fakeroot (versión igual o superior a la 1.14.4)
- debhelper (versión igual o superior a la 8.0.0)
- dh-make (versión igual o superior a la 0.55)

Para instalar cualquiera de los *paquetes tarball*, es necesario tener el siguiente paquete de software instalado:

- make (versión igual o superior a la 3.81)

Para instalar cualquiera de los *paquetes Debian*, es necesario tener el siguiente paquete de software instalado:

- dpkg (versión igual o superior a la 1.15.8)

9.1. Cómo generar el *paquete tarball* de un componente

Para generar el *paquete tarball* de alguno de los componentes de Predesys, partiendo del directorio raíz del código fuente del repositorio del proyecto, navegue hasta el directorio `<nombre-del-componente>/scripts` y ejecute el siguiente script:

```
generate-tarball.sh
```

Y, en el directorio generado *output*, aparecerá el archivo *tar.gz* resultante, el cual estará listo para distribuirse e instalarse.

9.2. Cómo generar el *paquete Debian* de un componente

Para generar el *paquete Debian* de alguno de los componentes de Predesys, partiendo del directorio raíz del código fuente del repositorio del proyecto, navegue hasta el directorio `<nombre-del-componente>/scripts` y ejecute el siguiente script:

```
generate-deb.sh
```

Y, en el directorio generado *output*, aparecerá el archivo *.deb* resultante, el cual estará listo para distribuirse e instalarse.

9.3. Requisitos del Servidor

El Servidor requiere tener los siguientes paquetes de software instalados en el sistema para poder ejecutarse:

- python (versión igual o superior a la 2.6.6)
- python-libxml2 (versión igual o superior a la 2.7.7)
- python-sqlalchemy (versión igual o superior a la 0.6.3)
- python-mysqldb (versión igual o superior a la 1.2.2)
- sysv-rc (versión igual o superior a la 2.87)
- mysql-server (versión igual o superior 5.1.49) - sólo si la base de datos del Servidor estará en la misma máquina que el Servidor.

9.4. Requisitos de la API de Servicios

La API de Servicios requiere tener los siguientes paquetes de software instalados en el sistema para poder ejecutarse:

- predesys-server (versión igual o superior a la 0.1.0) - Se trata del Servidor de Pre-desys
- python (versión igual o superior a la 2.6.6)
- python-setuptools (versión igual o superior a la 0.6.14)

9.5. Requisitos del Radar Bluetooth

Para poder ejecutarse, el Radar Bluetooth requiere que la máquina donde esté instalado disponga de un adaptador bluetooth (hardware) y tenga los siguientes paquetes de software instalados:

- python (versión igual o superior a la 2.6.6)
- python-libxml2 (versión igual o superior a la 2.7.7)
- bluez (versión igual o superior a la 4.66)
- python-bluez (versión igual o superior a la 0.18)
- python-lightblue (versión igual o superior a la 0.3.2)
- sysv-rc (versión igual o superior a la 2.87)

9.6. Requisitos del Radar LAN

El Radar LAN requiere tener los siguientes paquetes de software instalados en el sistema para poder ejecutarse:

- python (versión igual o superior a la 2.6.6)
- python-libxml2 (versión igual o superior a la 2.7.7)
- arp-scan (versión igual o superior a la 1.6)
- sysv-rc (versión igual o superior a la 2.87)

9.7. Requisitos de People Presence

El servicio de Predesys "People Presence" requiere tener los siguientes paquetes de software instalados en el sistema para poder ejecutarse:

- predesys-server (versión igual o superior a la 0.1.0)
- python (versión igual o superior a la 2.6.6)

Capítulo 10

Instalación

Todos los componentes de Predesys se pueden instalar cada uno bien mediante el archivo en formato *paquete tarball* o bien mediante el archivo en formato *paquete Debian*. El Servidor, la API de Servicios y todos los Servicios deben ser instalados en la misma máquina; los distintos Radares (ya sean Radar Bluetooth o Radar LAN) puede instalarse en la misma máquina o en otras distintas.

Después de instalar el Servidor y los diferentes Radares, es necesario configurarlos mediante sus interfaces de usuario o modificando sus archivos de configuración.

10.1. Servidor

Instalación mediante el archivo de distribución *tarball*

Descomprima el archivo *.tar.gz* en cualquier directorio y desde el directorio extraído del archivo comprimido (que contiene el programa), ejecute como administrador la siguiente orden:

```
make install
```

El Servidor quedará instalado. Su Servicio Web de Acceso a Servicios quedará instalado como un servicio del sistema (que se arranca en cada inicio del sistema) y estará ejecutándose.

Si quisiera desinstalarlo, debería ejecutar como administrador la siguiente orden también desde el directorio extraído del archivo comprimido:

```
make uninstall
```

Instalación mediante el archivo de distribución *paquete Debian*

Desde el directorio donde esté el archivo *.deb*, ejecute como administrador la siguiente orden:

```
dpkg -i [NombreArchivo]
```

Donde [NombreArchivo] es el nombre del archivo *.deb*. El Servidor quedará instalado. Su Servicio Web de Acceso a Servicios quedará instalado como un servicio del sistema (que se arranca en cada inicio del sistema) y estará ejecutándose.

Si quisiera desinstalarlo, debería ejecutar como administrador la siguiente orden (da igual el directorio donde se encuentre):

```
apt-get purge predesys-server
```

Consideraciones una vez instalado

Una vez instalado el Servidor, éste debe configurarse para establecer los parámetros de la base de datos a utilizar (ver capítulo *Uso y Configuración*). Después de hacer esto y reiniciar el Servidor, éste estará listo.

Para parar, iniciar de nuevo o reiniciar el servicio web, ejecute como administrador la siguiente orden (da igual el directorio donde se encuentre):

```
service predesys-server-webservice [opción]
```

Donde [opción] debe ser `stop`, `start` o `restart`, respectivamente.

10.2. API de Servicios

Instalación mediante el archivo de distribución *tarball*

Descomprima el archivo *.tar.gz* en cualquier directorio y desde el directorio extraído del archivo comprimido (que contiene el programa), ejecute como administrador la siguiente orden:

```
make install
```

La API de Servicios quedará instalada. El módulo Python instalado se llama *predesys*.

Si quisiera desinstalarlo, debería ejecutar como administrador la siguiente orden también desde el directorio extraído del archivo comprimido:

```
make uninstall
```

Instalación mediante el archivo de distribución *paquete Debian*

Desde el directorio donde esté el archivo *.deb*, ejecute como administrador la siguiente orden:

```
dpkg -i [NombreArchivo]
```

Donde [NombreArchivo] es el nombre del archivo *.deb*. La API de Servicios quedará instalada. El módulo Python instalado se llama *predesys*.

Si quisiera desinstalarlo, debería ejecutar como administrador la siguiente orden (da igual el directorio donde se encuentre):

```
apt-get purge predesys-service-api
```

10.3. Radar Bluetooth

Instalación mediante el archivo de distribución *tarball*

Descomprima el archivo *.tar.gz* en cualquier directorio y desde el directorio extraído del archivo comprimido (que contiene el programa), ejecute como administrador la siguiente orden:

```
make install
```

El Radar quedará instalado. Este componente quedará instalado como un servicio del sistema (que se arranca en cada inicio del sistema) y estará ejecutándose.

Si quisiera desinstalarlo, debería ejecutar como administrador la siguiente orden también desde el directorio extraído del archivo comprimido:

```
make uninstall
```

Instalación mediante el archivo de distribución *paquete Debian*

Desde el directorio donde esté el archivo *.deb*, ejecute como administrador la siguiente orden:

```
dpkg -i [NombreArchivo]
```

Donde [NombreArchivo] es el nombre del archivo *.deb*. El Radar quedará instalado. Este componente quedará instalado como un servicio del sistema (que se arranca en cada inicio del sistema) y estará ejecutándose.

Si quisiera desinstalarlo, debería ejecutar como administrador la siguiente orden (da igual el directorio donde se encuentre):

```
apt-get purge predesys-bluetooth-radar
```

Consideraciones una vez instalado

Una vez instalado el Radar Bluetooth, éste debe configurarse para establecer los parámetros para la conexión con el Servidor (ver capítulo *Uso y Configuración*). Después de hacer esto y reiniciar el Radar, éste estará listo.

Para parar, iniciar de nuevo o reiniciar el servicio, ejecute como administrador la siguiente orden (da igual el directorio donde se encuentre):

```
service predesys-bluetooth-radar [opción]
```

Donde [opción] debe ser `stop`, `start` o `restart`, respectivamente.

10.4. Radar LAN

Instalación mediante el archivo de distribución *tarball*

Descomprima el archivo *.tar.gz* en cualquier directorio y desde el directorio extraído del archivo comprimido (que contiene el programa), ejecute como administrador la siguiente orden:

```
make install
```

El Radar quedará instalado. Este componente quedará instalado como un servicio del sistema (que se arranca en cada inicio del sistema) y estará ejecutándose.

Si quisiera desinstalarlo, debería ejecutar como administrador la siguiente orden también desde el directorio extraído del archivo comprimido:

```
make uninstall
```

Instalación mediante el archivo de distribución *paquete Debian*

Desde el directorio donde esté el archivo *.deb*, ejecute como administrador la siguiente orden:

```
dpkg -i [NombreArchivo]
```

Donde [NombreArchivo] es el nombre del archivo *.deb*. El Radar quedará instalado. Este componente quedará instalado como un servicio del sistema (que se arranca en cada inicio del sistema) y estará ejecutándose.

Si quisiera desinstalarlo, debería ejecutar como administrador la siguiente orden (da igual el directorio donde se encuentre):

```
apt-get purge predesys-lan-radar
```

Consideraciones una vez instalado

Una vez instalado el Radar LAN, éste debe configurarse para establecer los parámetros para la conexión con el Servidor (ver capítulo *Uso y Configuración*). Después de hacer esto y reiniciar el Radar, éste estará listo.

Para parar, iniciar de nuevo o reiniciar el servicio, ejecute como administrador la siguiente orden (da igual el directorio donde se encuentre):

```
service predesys-lan-radar [opción]
```

Donde [opción] debe ser `stop`, `start` o `restart`, respectivamente.

10.5. Servicios de Predesys

Instalación mediante el archivo de distribución *tarball*

Descomprima el archivo *.tar.gz* en cualquier directorio y desde el directorio extraído del archivo comprimido (que contiene el programa), ejecute como administrador la siguiente orden:

```
make install
```

El servicio de Predesys quedará instalado.

Si quisiera desinstalarlo, debería ejecutar como administrador la siguiente orden también desde el directorio extraído del archivo comprimido:

```
make uninstall
```

Instalación mediante el archivo de distribución *paquete Debian*

Desde el directorio donde esté el archivo *.deb*, ejecute como administrador la siguiente orden:

```
dpkg -i [NombreArchivo]
```

Donde [NombreArchivo] es el nombre del archivo *.deb*. El servicio quedará instalado.

Si quisiera desinstalarlo, debería ejecutar como administrador la siguiente orden (da igual el directorio donde se encuentre):

```
apt-get purge <nombre-paquete>
```


Capítulo 11

Uso y configuración

A continuación, se presentan las órdenes de interfaz de línea de comandos que se pueden ejecutar para manejar el proyecto.

11.1. Manejo del Servidor

Grupos de usuarios

- Listar todos los grupos de usuarios:
`predesys-server usergroups list`
- Obtener los datos de un grupo de usuarios:
`predesys-server usergroups data [user-group-id]`
- Añadir un nuevo grupo de usuarios (ejecutar como administrador):
Directamente: `predesys-server usergroups add -id [id] -n [name]`
Interactivamente: `predesys-server usergroups add`
- Modificar un grupo de usuarios (ejecutar como administrador):
Directamente: `predesys-server usergroups mod -oid [old-id] -nid [new-id] -n [name]`
Interactivamente: `predesys-server usergroups mod`
- Borrar un grupo de usuarios (ejecutar como administrador):
Directamente, forzado: `predesys-server usergroups del -f [id]`
Interactivamente, confirmación necesaria: `predesys-server usergroups del [id]`

Usuarios

- Listar todos los usuarios:
`predesys-server users list`
- Obtener los datos de un usuario:
`predesys-server users data [user-id]`

- Añadir un nuevo usuario (ejecutar como administrador):
Directamente: `predesys-server users add -id [id] -gid [user-group-id] -e [enabled] -n [name] -em [email-address] -mac [mac-addresses]`

Donde:

-e -> 0 = deshabilitado, 1 = habilitado

-n, -em -> Opcionales

-mac -> Direcciones MAC separadas por comas. Ejemplo: `-mac "11:11:11:11:11:11, 22:22:22:22:22:22, 33:33:33:33:33:33"`

Interactivamente: `predesys-server users add`

- Modificar la contraseña de un usuario (ejecutar como administrador):
Directamente: `predesys-server users change -user [user-id] -passwd [password]`
Interactivamente: `predesys-server users change`

- Modificar un usuario (ejecutar como administrador):
Directamente:
`predesys-server users mod -oid [old-id] -nid [new-id] -gid [user-group-id] -n [name] -em [email-address] -mac [mac-addresses]`

Donde:

-e -> 0 = deshabilitado, 1 = habilitado

-n, -em -> Opcionales

-mac -> Direcciones MAC separadas por comas. Ejemplo: `-mac "11:11:11:11:11:11, 22:22:22:22:22:22, 33:33:33:33:33:33"`

Interactivamente:

`predesys-server users mod`

- Habilitar/deshabilitar un usuario (ejecutar como administrador):
`predesys-server users enable -id [id] -e [enabled]`

Donde:

-e -> 0 = deshabilitado, 1 = habilitado

- Borrar un usuario (ejecutar como administrador):
Directamente, forzado: `predesys-server users del -f [id]`
Interactivamente, confirmación necesaria: `predesys-server users del [id]`

Direcciones MAC

- Listar todas las direcciones MAC:
`predesys-server addresses list`

- Listar todas las direcciones MAC de un usuario:
`predesys-server addresses list data -uid [user-id]`
- Listar todos los usuarios de una dirección MAC:
`predesys-server addresses list data -a [address]`
- Añadir una nueva dirección MAC (ejecutar como administrador):
Directamente: `predesys-server addresses add -a [address] -uid [user-id]`
Interactivamente: `predesys-server addresses add`
- Modificar una dirección MAC (ejecutar como administrador):
Directamente: `predesys-server addresses mod -oa [old-address] -oid [old-user-id] -na [new-address] -nuid [new-user-id]`
Interactivamente: `predesys-server addresses mod`
- Borrar una dirección MAC (ejecutar como administrador):
Directamente, forzado: `predesys-server addresses del -f [address]`
Interactivamente, confirmación necesaria: `predesys-server addresses del [address]`

Tipos de servicio

- Listar todos los tipos de servicio:
`predesys-server servicetypes list`
- Obtener los datos de un tipo de servicio:
`predesys-server servicetypes data [service-type-id]`

Servicios

- Listar todos los servicios:
`predesys-server services list`
- Listar todos los servicios de un tipo determinado:
`predesys-server services list -tid [service-type-id]`
- Listar todos los servicios habilitados de un tipo determinado:
`predesys-server services list -e -tid [service-type-id]`
- Obtener los datos de un servicio:
`predesys-server services data [service-id]`
- Añadir un nuevo servicio (ejecutar como administrador):
Directamente:
`predesys-server services add -id [id] -v [version] -tid [service-type-id] -a [authenticated] -e [enabled] -p [path] -n [name] -au [author]`

Donde:

-a -> 0 = sin autenticación, 1 = con autenticación

-e -> 0 = deshabilitado, 1 = habilitado

-v, -n, -au -> Opcionales

Interactivamente:

```
predesys-server services add
```

- Modificar un servicio (ejecutar como administrador):

Directamente: `predesys-server services mod -oid [old-id] -nid [new-id] -v [version] -tid [service-type-id] -p [path] -n [name] -au [author]`

Donde:

-v, -n, -au -> Opcionales

Interactivamente: `predesys-server services mod`

- Habilitar/deshabilitar la autenticación de un servicio (ejecutar como administrador):

```
predesys-server services auth -id [service-id] -a [authenticated]
```

Donde:

-a -> 0 = sin autenticación, 1 = con autenticación

- Habilitar/deshabilitar un servicio (ejecutar como administrador):

```
predesys-server services enable -id [service-id] -e [enabled]
```

Donde:

-e -> 0 = deshabilitado, 1 = habilitado

- Borrar un servicio (ejecutar como administrador):

Directamente, forzado: `predesys-server services del -f -id [service-id]`

Interactivamente, confirmación necesaria: `predesys-server services del -id [service-id]`

- Ejecutar un servicio que no requiere autenticación:

```
predesys-server services run -id [service-id] -args 'arg1, arg2...'
```

- Ejecutar un servicio que requiere autenticación:

```
predesys-server services run -id [service-id] -user [user-id] -passwd  
-args 'arg1, arg2...'
```

Donde:

-passwd -> Opcional. Si no se introduce su valor, el sistema lo pedirá al usuario.

Permisos de servicio

- Listar todos los permisos de servicio:

```
predesys-server serviceperms list
```

- Obtener los datos de un permiso de servicio:

```
predesys-server serviceperms data -sid [service-id] -gid [user-group-id]
```

- Añadir un permiso de servicio (ejecutar como administrador):

Directamente:

```
predesys-server serviceperms add -sid [service-id] -gid [user-group-id]
```

Interactivamente:

```
predesys-server serviceperms add
```

- Modificar un permiso de servicio (ejecutar como administrador):

Directamente:

```
predesys-server serviceperms mod -osid [old-service-id] -ogid [old-user-group-id]  
-nsid [new-service-id] -ngid [new-user-group-id]
```

Interactivamente:

```
predesys-server serviceperms mod
```

- Borrar un permiso de servicio: (ejecutar como administrador)

Directamente, forzado:

```
predesys-server serviceperms del -f -sid [service-id] -gid [user-group-id]
```

Interactivamente, confirmación necesaria:

```
predesys-server serviceperms del -sid [service-id] -gid [user-group-id]
```

- Borrar todos los permisos de servicio de un servicio determinado (ejecutar como administrador):

Directamente, forzado:

```
predesys-server serviceperms del -f -sid [service-id]
```

Interactivamente, confirmación necesaria:

```
predesys-server serviceperms del -sid [service-id]
```

- Borrar todos los permisos de servicio de un grupo de usuarios determinado (ejecutar como administrador):

Directamente, forzado:

```
predesys-server serviceperms del -f -gid [user-group-id]
```

Interactivamente, confirmación necesaria:

```
predesys-server serviceperms del -gid [user-group-id]
```


Capítulo 12

Cómo crear un servicio de Predesys

Un servicio de Predesys no es más que un script de Python que debe implementar una función con unos parámetros de entrada y tipo de salida concretos. Esta función varía en función del tipo de servicio que quiera implementar (ver sección *Arquitectura de Predesys* del capítulo *Arquitectura*). Veamos cómo tiene que ser esta función en cada caso:

- Para servicios automáticos y servicios de usuario sin autenticación:

```
def run(args)
```

Donde *args* es una lista (de cadenas de texto) de los identificadores de los dispositivos detectados.

- Para servicios de usuario con autenticación:

```
def run(userId, args)
```

Donde *userId* es el identificador del usuario que ha solicitado ejecutar el servicio (una cadena de texto) y *args* es una lista (de cadenas de texto) de los identificadores de los dispositivos detectados.

Todos los servicios tienen acceso a la API de Servicios de Predesys (un módulo de Python), que ofrece un acceso limitado al Núcleo del Servidor. Con esta API, se pueden obtener los usuarios registrados en el sistema, formatear un mensaje para que se envíe a través de alguno de los Radares a alguno de los dispositivos detectados, obtener los dispositivos que están registrados en el sistema...

Una vez terminado el servicio, para añadirlo al Servidor, tan sólo hay que ejecutar como administrador la orden diseñada a tal efecto (ver sección *Manejo del Servidor* del capítulo *Uso y Configuración*).

Bibliografía

- [1] C. Commons, “Licencia reconocimiento-compartirigual 3.0.” <http://creativecommons.org/licenses/by-sa/3.0/es/legalcode.es>.
- [2] F. S. Foundation, “Gnu lesser general public license 3.0.” <http://www.gnu.org/licenses/lgpl-3.0-standalone.html>.
- [3] P. S. Foundation, “Python programming language.” <http://www.python.org>.
- [4] BluLabs, “Blukit.” <http://www.blulabs.es>.
- [5] G. R. Me, “Geo remind me.” <http://www.georemindme.com>.
- [6] L. Cassioli, “Powerwarning.” <http://code.google.com/p/powerwarning>.
- [7] A. Kumar, “Bluehoc.” <http://bluehoc.sourceforge.net>.
- [8] C. Liechti, “Pyserial.” <http://pyserial.sourceforge.net>.
- [9] A. Laurie, “Rfidiot.” <http://http://rfidiot.org>.
- [10] A. Venkataramani and J. Tourrilhes, “Java socket interface for infrared communication.” http://www.hpl.hp.com/personal/Jean_Tourrilhes/IrDA/Java.html.
- [11] Wikipedia, “Cocomo.” <http://es.wikipedia.org/wiki/COCOMO>.
- [12] Wikipedia, “Requisito funcional.” http://es.wikipedia.org/wiki/Requisito_funcional.
- [13] Wikipedia, “Caso de uso.” http://es.wikipedia.org/wiki/Caso_de_uso.
- [14] Wikipedia, “Requisito no funcional.” http://es.wikipedia.org/wiki/Requisito_no_funcional.