

PREDESYS

Documentación del proyecto PredeSys (versión 1)

Cubre las siguientes versiones de los componentes de PredeSys:

Bluetooth Proxy: 0.1.0

José Antonio Jiménez Carmona (josantjim@gmail.com)

Tutor Proyecto Fin de Carrera:

Pablo Neira Ayuso

Universidad de Sevilla

Dpto. de Lenguajes y Sistemas Informáticos

13 de diciembre de 2010

Índice general

1. Licencia	5
1.1. Licencia de este documento	5
1.2. Licencia del proyecto	5
2. Introducción	9
2.1. El porqué	9
2.2. De qué se trata	9
2.3. Objetivos generales	9
2.4. Trabajos relacionados	9
3. Planificación	11
3.1. Estimación	11
3.2. Tiempo empleado	12
4. Elicitación de requisitos	13
4.1. Objetivos específicos	13
4.2. Casos de uso	13
5. Arquitectura	15
5.1. Sistemas distribuidos	15
5.2. Arquitecturas Cliente-Servidor	15
5.3. Arquitectura de Predesys	15
6. Implementación	17
7. Distribución	19
8. Instalación	21
8.1. Proxy Bluetooth	21

Capítulo 1

Licencia

1.1. Licencia de este documento

Este documento se rige por la licencia *Reconocimiento-CompartirIgual 3.0* de Creative Commons. 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.

Puede consultar el contenido de la licencia en la siguiente dirección web:

<http://creativecommons.org/licenses/by-sa/3.0/es/legalcode.es>

1.2. Licencia del proyecto

El proyecto Predesys se rige por la licencia *GNU Lesser General Public License 3.0*. El contenido completo de la licencia (en inglés) se expone a continuación:

GNU LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, “this License” refers to version 3 of the GNU Lesser General Public License, and the “GNU GPL” refers to version 3 of the GNU General Public License.

“The Library” refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An “Application” is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A “Combined Work” is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the “Linked Version”.

The “Minimal Corresponding Source” for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The “Corresponding Application Code” for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
- d) Do one of the following:
 - 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
 - 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
- e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy’s public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

Capítulo 2

Introducción

2.1. El porqué

La creación de este proyecto viene motivado por, además de la obligación de realizar un proyecto de fin de carrera en mi titulación, mis inquietudes sobre algunas tecnologías de comunicación (como el bluetooth o la web), mis inquietudes sobre cómo integrar distintas plataformas informáticas (PCs, dispositivos móviles, microcontroladores, etc) y la conveniencia de tener un sistema de localización de personal en algunas organizaciones (como pueden ser las distintas asociaciones de estudiantes de la Escuela Técnica Superior de Ingeniería Informática de la Universidad de Sevilla o cualquier empresa).

2.2. De qué se trata

PredeSYS es un sistema software contenido en un servidor central y en un conjunto de servidores secundarios con el objetivo de detectar la presencia de dispositivos bluetooth (principalmente, teléfonos móviles) previamente registrados que se encuentren cerca físicamente de dichos servidores secundarios para ofrecer una serie de servicios de información a los usuarios de dichos dispositivos y a terceros. El sistema está pensado para implantarse en oficinas u otros tipos de instalaciones físicas. Está diseñado para los sistemas operativos Linux y es software libre (su licencia es LGPL 3.0).

2.3. Objetivos generales

Los objetivos principales del proyecto son:

1. Disponer de una o más máquinas capaz de detectar a un dispositivo bluetooth concreto y realizar alguna tarea (servicio) asociada a dicha detección.
2. Disponer de un sistema de plugins para que terceros puedan programar tareas (servicios) adicionales para el sistema.

2.4. Trabajos relacionados

Capítulo 3

Planificación

3.1. Estimación

El proyecto consta de varios componentes bien diferenciados (ver sección *Arquitectura* para más información). La estimación del tiempo empleado necesario para realizar cada una de las partes del proyecto es la siguiente:

- **Diseñar la arquitectura del proyecto a un nivel esquemático:**

Requiere definir qué componentes necesita el proyecto y cómo se comunicarán entre ellos.

Tiempo estimado: 2 semanas

- **Completar la implementación de la primera versión del proxy bluetooth:**

Esto implica investigar sobre la tecnología bluetooth, investigar sobre el protocolo HTTP e investigar sobre la distribución de aplicaciones para Linux.

Tiempo estimado: 1 mes

- **Completar la implementación de la primera versión del servidor:**

Esto implica el diseño de cómo debe accederse a los servicios que ofrezca el servidor, el diseño del sistema de plugins para que terceros puedan programar servicios adicionales, y el sistema de autenticación para determinados servicios que requieran la identificación del usuario.

Tiempo estimado: 1 mes

- **Completar la implementación de la primera versión del radar bluetooth:**

Esta parte del proyecto sólo requiere los conocimientos de la primera parte y de cómo esté hecho el servidor, por lo que el tiempo para realizarla no debería ser mucho.

Tiempo estimado: 1 semana

- **Completar la implementación de la primera versión de las bibliotecas de cliente para la plataforma Android de TCP/IP y de bluetooth:**

Para realizar esta parte, hay que aprender a desarrollar aplicaciones para el sistema operativo Android y el uso de alguna biblioteca para manejar el dispositivo bluetooth de los teléfonos móviles que tengan dicho sistema operativo.

Tiempo estimado: 1 mes

- **Completar la implementación de la primera versión de las bibliotecas de cliente para PC de TCP/IP y de bluetooth:**

Requiere investigar sobre cómo diseñar e integrar bibliotecas software en Linux.

Tiempo estimado: 2 semanas

3.2. Tiempo empleado

- **Diseñar la arquitectura del proyecto a un nivel esquemático:**

Tiempo empleado: 2 semanas

El tiempo empleado ha sido el previsto.

- **Completar la implementación de la primera versión del proxy bluetooth:**

Tiempo empleado: 2 meses

Se ha doblado el tiempo estimado inicialmente, debido a problemas surgidos en la investigación sobre la distribución de aplicaciones para Linux. Estos problemas han sido mi desconocimiento casi total de cómo debe organizarse el código fuente del proyecto y cómo debe distribuirse el código fuente a desarrolladores y el programa final a los usuarios. Para la distribución del programa a los usuarios, ha sido necesario investigar acerca de cómo se crean los *paquetes Debian* (archivos que contienen un programa listo para instalar), para lo cual, la documentación existente en Internet es bastante confusa.

Capítulo 4

Elicitación de requisitos

4.1. Objetivos específicos

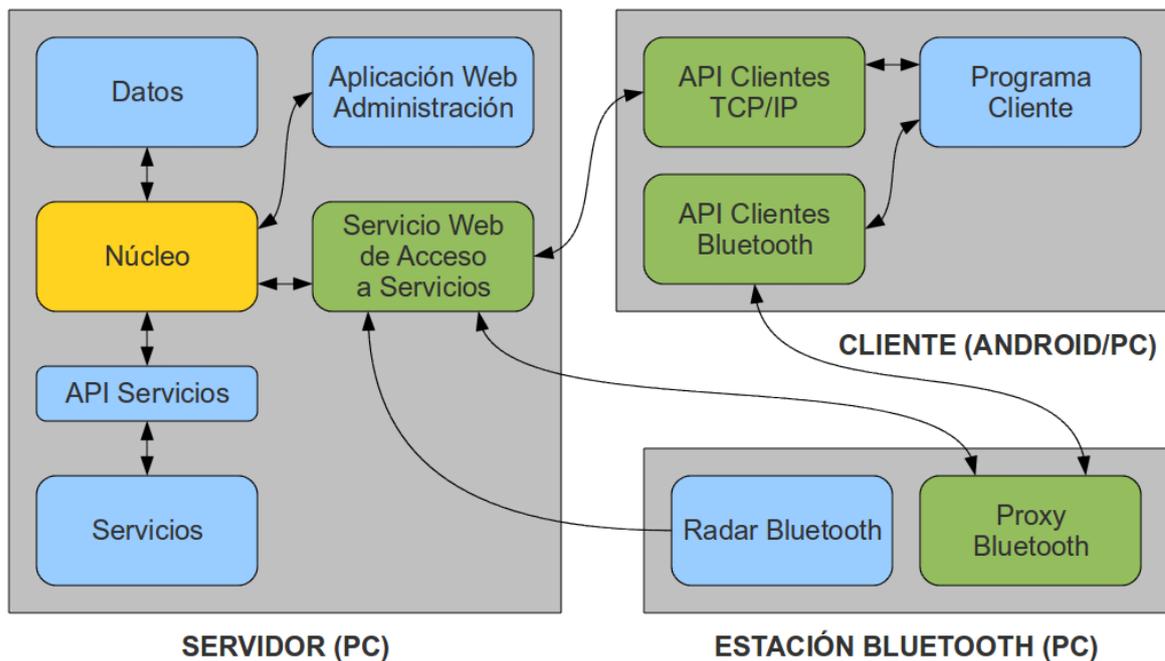
4.2. Casos de uso

Capítulo 5

Arquitectura

- 5.1. Sistemas distribuidos
- 5.2. Arquitecturas Cliente-Servidor
- 5.3. 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 Estaciones Bluetooth (PCs) y 1 ó más Clientes (PCs o dispositivos móviles con el sistema operativo Android).

El Servidor contiene toda la lógica de los servicios y los datos de los mismos. Ofrece los servicios a través del Servicio Web de Acceso a Servicios, que no es más que un servicio web al que

se hace una petición *post* desde el Programa Cliente para obtener la información deseada por el usuario. El Servidor contiene, entre otros componentes, una aplicación web accesible desde el exterior para que un Administrador configure el sistema como quiera (la Aplicación Web de Administración). 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 acceden a los Datos y los modifican. El Núcleo es el intermediario entre los Servicios y los Datos y cualesquiera otros dos componentes del Servidor, ya que debe controlar y asegurar el correcto funcionamiento del sistema y limitar los datos a los que pueda acceder cada parte del sistema.

El cliente hará las peticiones *post* a través de la API de Clientes TCP/IP (requiere conexión a Internet) o bien a través de la API de Clientes Bluetooth (requiere conectividad bluetooth). Si el Programa Cliente usa la API de Clientes Bluetooth, enviará las peticiones a una de las Estaciones Bluetooth a través de una conexión bluetooth que a su vez las enviará al Servicio Web de Acceso a Servicios a través de una conexión por TCP/IP (que es lo que hace la API de Clientes TCP/IP directamente). La API de Clientes Bluetooth, a priori, hace lo mismo que la API de Clientes TCP/IP pero es lo que hace interesante al proyecto, ya que permite al usuario acceder a los servicios sin conexión a Internet (eso sí, el usuario debe estar cerca físicamente de una de las estaciones bluetooth, ya que el alcance de las señales bluetooth es muy limitado).

Las estaciones bluetooth pueden ser 1 ó más, todas con conectividad por TCP/IP con el servidor. Tienen tan sólo 2 funciones muy básicas aunque muy importantes. La primera, “rebotar” peticiones bluetooth de los Clientes hacia el Servidor, como se ha explicado en el párrafo anterior y, la segunda, detectar periódicamente la presencia de dispositivos bluetooth que estén físicamente cerca de la Estación Bluetooth y enviar sus direcciones MAC al Servidor, a través de uno de los Servicios (en este caso, un Servicio especial de acceso limitado a las Estaciones Bluetooth). Ésta última función tiene la finalidad de ejecutar ciertas tareas en el servidor cuando se encuentre cerca de la estación alguno de los usuarios, generalmente para ofrecerle algún servicio concreto asociado a su presencia física (es decir, a la presencia física de su dispositivo bluetooth, como su teléfono móvil).

Capítulo 6

Implementación

Capítulo 7

Distribución

Capítulo 8

Instalación

8.1. Proxy Bluetooth

El Proxy Bluetooth se distribuye en 2 archivos de formatos distintos: formato *tarball* (con extensión *.tar.gz*) y formato de *paquete Debian* (con extensión *.deb*).

El *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 (en un entorno gráfico como GNOME bastaría con unos pocos de clicks del ratón) 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.

Dependencias

El Proxy Bluetooth requiere tener los siguientes paquetes de software instalados en el sistema (los nombres son los establecidos en los sistemas Linux tipo Debian, en otros sistemas pueden ser distintos) para poder ejecutarse:

- python (versión igual o superior a la 2.5)
- python-libxml2 (versión igual o superior a la 2.6)
- python-bluez (versión igual o superior a la 0.15)
- sysv-rc (versión igual o superior a la 2.86)

En el caso de que quiera instalar el Proxy Bluetooth desde el archivo *tarball*, necesitará, para poder instalarlo, el siguiente paquete:

- make (versión igual o superior a la 3.81)

Instalación mediante el *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 Proxy Bluetooth 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

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

service predesys-bluetooth-proxy [opción]

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

Instalación mediante el 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 Proxy Bluetooth 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-proxy

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

service predesys-bluetooth-proxy [opción]

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