

Gestión de la red XEiLL

Alumno: Daniel Morillas Muñoz

Titulación: 2º ciclo de Ingeniería en Informática

Consultor: Víctor Carceler Hontoria

Fecha: 19 de junio del 2006

1. RESUMEN

La XEiLL (Xarxa Educativa i Lliure) se trata de una red inalámbrica, de libre acceso, basada en tecnología WiFi, que permite a cualquier persona, que se encuentre dentro de su área de cobertura, poder usarla sin ningún coste.

En la actualidad, la gestión y mantenimiento de todos los datos relacionados con los puntos de acceso de la red (software instalado, hardware, enlaces y VPNs entre nodos, IPs, etc.) se lleva de una forma más o menos manual.

El objetivo de este proyecto es el de diseñar una base de datos que permita almacenar todos los datos actuales, y los de los futuros centros que formen parte de la red, así como el desarrollo de una aplicación que facilite, de una forma fácil e intuitiva, llevar una gestión eficiente de toda esta información.

1.1. Estructura de la Memoria

La memoria está dividida en diferentes secciones o apartados, los cuales, a su vez, están subdivididos en capítulos. Las secciones en las que se organiza esta memoria son:

- **Introducción.** Breve introducción a los motivos del desarrollo de la aplicación, así como los objetivos de la misma, el planning de trabajo y tiempos realizados para su implementación.
- **Estudio del entorno tecnológico.** En este capítulo se introduce al lector en el entorno teórico en el que se haya el desarrollo del actual proyecto. Se listarán las definiciones de los diferentes conceptos importantes a conocer, poniendo mayor énfasis en el estudio de la tecnología Wi-Fi.
- **Análisis y Diseño.** Estudio del análisis de requisitos y diseño de la aplicación. Los capítulos en los que se divide esta sección son: definición de los requisitos del sistema, definición de la persistencia de datos y definición de la interfaz gráfica, a través de los cuales se explican estos conceptos y decisiones.
- **La aplicación.** Explicación de la implementación realizada.
- **Conclusiones.** Evaluación del resultado de la aplicación, de los objetivos conseguidos y enumeración de las posibles líneas de ampliación.

2. ÍNDICE

1.	RESUMEN	2
1.1.	Estructura de la Memoria	2
2.	ÍNDICE.....	3
3.	ÍNDICE DE IMÁGENES	5
4.	INTRODUCCIÓN	6
4.1.	Descripción del proyecto	6
4.2.	Objetivos del proyecto.....	7
4.3.	Análisis de riesgos	7
4.4.	Planning de trabajo	7
4.4.1.	Desglose de tareas	7
4.4.2.	Calendario de trabajo	9
5.	ESTUDIO DEL ENTORNO TECNOLÓGICO.....	10
5.1.	Introducción al Wi-Fi	10
5.2.	Otras definiciones	10
6.	ANÁLISIS Y DISEÑO	12
6.1.	Definición de los requisitos del sistema.....	12
6.1.1.	Requisitos no funcionales.....	12
6.1.2.	Requisitos funcionales	12
6.1.3.	Entradas y salidas del sistema	14
6.1.4.	Gestión de errores del sistema	14
6.2.	Definición de la persistencia de datos.....	15
6.2.1.	Sistema gestor base de datos	15
6.2.2.	Sistema de persistencia.....	16
6.2.3.	Diagrama entidad relación.....	17
6.3.	Definición de la interfaz gráfica	24
6.3.1.	Tecnología escogida.....	24
6.3.2.	Interfaz gráfica.....	26
7.	LA APLICACIÓN.....	31
7.1.	Arquitectura	31
7.1.1.	Presentación	32
7.1.2.	Modelo.....	32
7.1.3.	Persistencia.....	32
7.1.4.	Base de datos.....	35
7.2.	Aplicación	35
7.2.1.	Identificación de usuario	35

Memoria del Proyecto

7.2.2.	Barra de herramientas	36
7.2.3.	Área de navegación	38
7.2.4.	Área de trabajo	39
7.3.	Área de trabajo de la aplicación	39
7.3.1.	Registro de incidencias	39
7.3.2.	Gestión de usuarios y perfiles	41
7.3.3.	Gestión de un centro	44
8.	CONCLUSIONES.....	60
8.1.	Objetivos.....	60
8.2.	Líneas futuras de trabajo	61
9.	BIBLIOGRAFÍA.....	62
9.1.	Direcciones Web	62

3. ÍNDICE DE IMÁGENES

Imagen 1: Planning de trabajo	9
Imagen 2: Diagrama E-R	18
Imagen 3: Ejemplo diseño 1	27
Imagen 4: Ejemplo diseño 2	28
Imagen 5: Ejemplo diseño 3	29
Imagen 6: Ejemplo diseño 4	30
Imagen 7: Identificación usuario.....	35
Imagen 8: Barra de herramientas de centros	37
Imagen 9: Barra de herramientas de usuarios	37
Imagen 10: Área de navegación.....	38
Imagen 11: Registro de incidencias	41
Imagen 12: Listado usuarios y perfiles.....	42
Imagen 13: Gestión usuarios	43
Imagen 14: Gestión perfiles.....	44
Imagen 15: Información de un centro	45
Imagen 16: Listado de puntos de acceso	47
Imagen 17: Punto de acceso.....	48
Imagen 18: Enlaces WDS.....	49
Imagen 19: Listado servidores	50
Imagen 20: Servidor	51
Imagen 21: Servicio.....	52
Imagen 22: Adaptador de red	53
Imagen 23: VPN	53
Imagen 24: Nodo	55
Imagen 25: Rango IP	56
Imagen 26: DNS	56
Imagen 27: Encaminamiento	56
Imagen 28: Monitorización.....	57
Imagen 29: Monitorización "NO RESPONDE".....	59
Imagen 30: Monitorización "RESPONDE".....	59
Imagen 31: Gráfica monitorización.....	59

4. INTRODUCCIÓN

4.1. Descripción del proyecto

XEiLL se trata de una red educativa de libre acceso basada en tecnología WiFi que permite que cualquier persona, con conectividad, pueda acceder sin necesidad de altas o cuotas, siendo el único requisito estar dentro del área de cobertura de alguno de los puntos de acceso.

Cada uno de los nodos que conforman esta red está formado por diferentes centros educativos, de cualquier nivel, los cuales cuentan con la siguiente infraestructura:

- LAN propia independiente de la red XEiLL.
- Conexión a Internet.
- Personal de mantenimiento que usa los recursos tecnológicos para fines didácticos.

En cada uno de estos centros educativos, cuando se desea convertir en un nodo de la red, es necesario instalar el siguiente *hardware*:

1. Un pequeño servidor: un PC con GNU/Linux y dos adaptadores de red.
2. Uno o varios puntos de acceso.

Dicho servidor actúa como pasarela entre la XEiLL y la LAN del centro, a través de un adaptador de red en la LAN y otro en una VLAN, en la que están todos los puntos de acceso del centro.

Esta pasarela presta diferentes servicios de red básicos para la XEiLL:

1. DHCP para los clientes.
2. DNS (es responsable del subdominio *centroX.xeill.net*)
3. Proxy HTTP/FTP (los clientes de la XEiLL pueden acceder a Internet a través del Proxy)
4. Túneles (VPNs) a otros centros.
5. Encaminamiento para el tráfico de la XEiLL (OSPF)
6. Jabber (mensajería instantánea)
7. Web del centro.
8. Servicio de streaming de audio o PBX de VoIP
9. etc.

Llegados a este punto, a pesar que el número de centros que participan actualmente en la XEiLL no es demasiado elevado (10 centros por el momento), se convierte en una tarea difícil e incómoda el poder llevar a cabo el mantenimiento de toda la información de los nodos, hardware instalado en cada uno de ellos y monitorización de la red. Esto fuerza la necesidad de tener

una base de datos centralizada donde guardar toda la información y una aplicación que nos permita poder gestionar toda esta información (dar de alta nueva información, actualizarla o, simplemente, consultarla) y llevar una monitorización básica de la red.

4.2. Objetivos del proyecto

Una vez finalizado el proyecto se espera haber desarrollado una aplicación web que permita gestionar:

1. Todos los aspectos de la instalación de cada nodo (tanto hardware – modelo y características de los equipos–, como software) y su configuración.
2. Llevar un registro de todas las actuaciones/incidencias.
3. Monitorización básica del estado de la red.

A grandes rasgos, la aplicación web de gestión de la información deberá gestionar:

1. Todos los puntos de acceso que tiene cada centro (dónde están instalados, marca/modelo, usuario/contraseña de administración, etc.)
2. Características del servidor/servidores de cada centro (marca/modelo, versión de SO, etc.)
3. Parámetros de configuración del nodo (subred de la XEiLL, subdominios registrados, etc.)
4. Información diversa del centro (dirección, teléfono, persona de contacto, etc.)

4.3. Análisis de riesgos

Los riesgos con los que nos podemos encontrar en el desarrollo de este proyecto son los principales que se pueden dar en el desarrollo de cualquier proyecto:

- Mala definición de los requisitos de la aplicación.
- No realizar un buen diseño de la base de datos y poder olvidarnos de la definición de alguna tabla o dato.
- Hacer una mala planificación del proyecto, pudiendo provocar retrasos posteriores.

4.4. Planning de trabajo

4.4.1. Desglose de tareas

A continuación se pasará a detallar un listado del desglose de las diferentes tareas en las que se divide este proyecto. Para cada tarea se especificará: los objetivos a alcanzar en dicha tarea y la duración estimada en días.

Plan de proyecto

Objetivos: Hacer una primera aproximación del alcance y objetivos del proyecto para poder realizar un desglose de tareas y planificación inicial de las misma.

Estimación: 4 días

Toma de requisitos

Objetivos: Poder realizar un listado con todos los requisitos (tanto funcionales como tecnológicos) y necesidades que deberá cumplir la aplicación, para así tener claro lo que hay que realizar.

Estimación: 5 días

Diseño de la estructura de la base de datos

Objetivos: Definir el diagrama de Entidad-Relación de la base de datos (tablas, campos, relaciones, etc.) que usará la aplicación.

Estimación: 5 días.

Diseño de la interfaz gráfica

Objetivos: Maquetación de la interfaz gráfica que deberá tener la aplicación final: definición de diferentes pantallas, navegación, estilo de la página (fuentes, colores, etc.)

Estimación: 10 días.

Implementación de la aplicación web

Objetivos: Una vez está claro cuál es la base de datos a usar y la interfaz gráfica, se deberá implementar la aplicación a partir de la cual se accederá a la base de datos (inserción, modificación, consulta de datos)

Estimación: 20 días.

Estudio de herramientas de monitorización

Objetivos: Estudio de diferentes soluciones para implementar la herramienta de monitorización de la red XEILL.

Estimación: 5 días.

Implementación monitorización básica

Objetivos: Una vez decidida la solución a usar se deberá implementar la herramienta que monitorice el funcionamiento correcto de la red.

Estimación: 5 días.

Documentación

Objetivos: Realización de la documentación final de la memoria del proyecto.

Estimación: 10 días.

4.4.2. Calendario de trabajo

Teniendo en cuenta el desglose de tareas anterior, la estimación de cada una de ellas y que el proyecto se comienza el lunes 06/03/2006 el calendario de trabajo del proyecto es el que se puede observar a continuación.

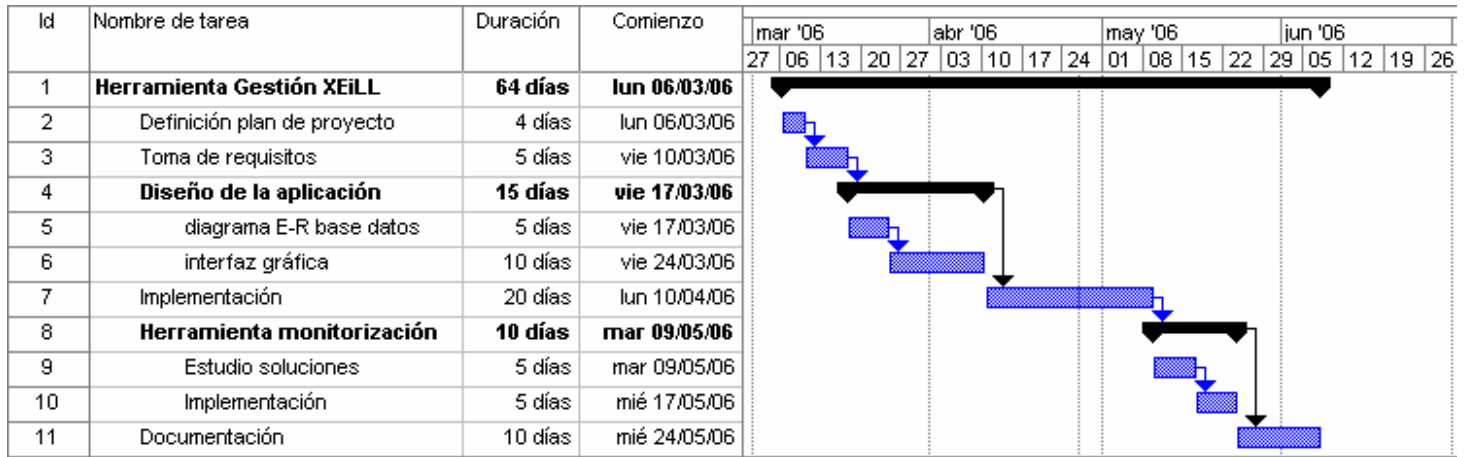


Imagen 1: Planning de trabajo

5. ESTUDIO DEL ENTORNO TECNOLÓGICO

5.1. Introducción al Wi-Fi

Wi-Fi (Wireless Fidelity) es el nombre común por el que se conoce a un conjunto de estándares en el que se basan todos los productos de redes inalámbricas de área local o WLAN (Wireless Local Area Network), basados en las especificaciones IEEE 802.11.

En sus inicios, los estándares Wi-Fi se crearon para ser usados en los accesos a redes locales inalámbricas, pero con el paso del tiempo, el avance de las tecnologías y la aparición de nuevas necesidades han provocado que, en la actualidad, sea muy frecuente que también se utilice para el acceso a Internet.

Actualmente Wi-Fi es una marca de la *Wi-Fi Alliance*, que es la organización que se encarga de la prueba y certificación que todos los equipos fabricados bajo esta marca cumplen los estándares IEEE 802.11x.

En los inicios de la definición del estándar Wi-Fi, éste estaba pensado para dispositivos inalámbricos con una velocidad de 1 y 2 Mbps y que operaban en la banda de frecuencias de 2,4 Ghz. Como cualquier otra tecnología, con el paso del tiempo y las necesidades provocó que se fuesen creando nuevos estándares dentro de la familia 802.11. Algunos de los estándares actuales son:

- **802.11a:** soporta velocidades de hasta 54 Mbps, utilizando la banda de frecuencia de 5 Ghz. Se trata de un protocolo orientado a la transmisión de paquetes que no soporta QoS (*Quality of Service*).
- **802.11b:** soporta velocidades de hasta 11 Mbps, utilizando la banda de frecuencia de 2,4 Ghz.
- **802.11g:** Es una extensión del anterior, siendo la base de la mayoría de las redes locales inalámbricas de hoy día. Alcanza velocidades de hasta 54 Mbps en la banda de frecuencia de 2,4 Ghz. Utiliza la tecnología OFDM (*Orthogonal Frequency-Division Multiplexing*) que también se utiliza en el protocolo 802.16 (WiMAX).
- **802.11i:** Es uno de los últimos estándares Wi-Fi aprobados. El objetivo principal de este estándar es el de dotar de protocolos más seguros en la comunicaciones de redes WLAN, uno de los problemas más importantes. Algunas de las mejoras más importantes que ofrece este estándar son: Acceso Wi-Fi protegido (WPA) y *Robust Security Network* (RSN).

Finalmente también comentar que como complemento al Wi-Fi, apareció una nueva tecnología, el WiMax, que es el estándar usado para el desarrollo de redes inalámbricas de gran alcance.

5.2. Otras definiciones

- **Punto de acceso:** Es el dispositivo que conecta los diferentes dispositivos inalámbricos o estaciones Wi-Fi formando la red

inalámbrica. También se puede usar para ejercer funciones de puente entre una red Ethernet y una Wi-Fi.

- **SSID:** Identificador del área de servicio del punto de acceso.
- **VPN:** *Virtual Private Network*. Es una conexión privada entre dos o más servidores. Es la extensión de una red local a través de una red pública, como Internet.
- **WDS:** *Wireless Distribution System*. Se trata de un sistema de enlazar dos puntos de acceso de forma inalámbrica. Un punto de acceso puede tener varios enlaces WDS y en cada enlace participan únicamente dos puntos de acceso. Los puntos de acceso de un mismo enlace tienen el mismo canal y SSID.
- **WiMAX:** Nombre que se conoce al conjunto de estándares, basados en las especificaciones IEEE 802.16, para el desarrollo de productos para redes inalámbricas de gran alcance. Este estándar surgió de la necesidad de cubrir zonas en las que no se podían cubrir con los métodos clásicos (cable de cobre, fibra óptica,...) y que, hasta el momento se hacían con métodos nada recomendables (infraestructuras propietarias, tecnologías 802.11), ya fuese por coste o por seguridad.

6. ANÁLISIS Y DISEÑO

6.1. Definición de los requisitos del sistema

6.1.1. Requisitos no funcionales

Por requisitos no funcionales entendemos todas aquellas características que debe cumplir la aplicación y que no son dependientes del funcionamiento de la misma (p.e.: tecnología a usar, hardware necesario, etc.)

A continuación se pasan a listar los requisitos no funcionales de nuestro proyecto. Cada requisito va codificado con un código por si en algún otro momento debemos hacernos referencia a él. El formato del código será RNF-XXX (donde XXX es un número único y secuencial identificativo de este requisito):

[RNF-001] Deben usarse tecnologías y herramientas de software libre.

[RNF-002] Debe ser una herramienta que pueda consultarse a través de la web (html, jsp, applet, etc.)

[RNF-003] Todos los datos gestionados por esta aplicación deben guardarse en un dispositivo de almacenamiento no volátil, de fácil acceso y estándar.

6.1.2. Requisitos funcionales

Al contrario que los anteriores, los requisitos funcionales son todas aquellas características que debe cumplir la aplicación y que sí son dependientes del funcionamiento de la misma (p.e.: si el acceso a la aplicación se hace protegido o no, datos que se debe gestionar, etc.)

Al igual que los no funcionales, los funcionales los codificaremos de forma similar, siendo, en este caso, el código RF-XXX:

[RF-001] Debe tratarse de una herramienta fácil de usar, con una interfaz amigable.

[RF-002] La aplicación debe estar protegida por nombre usuario/contraseña.

[RF-003] Para cada nodo WiFi (centro educativo) se debe especificar la siguiente información:

- Información varia:
 - Nombre del centro.
 - Dirección del centro.
 - Teléfono.
 - Persona de contacto.
 - E-mail de contacto.
 - Información del router de acceso a Internet: IPs interna y externa y redirecciones de puerto.
 - DNS asignado al centro.

- Puntos de acceso:
 - Dónde están instalados.
 - Marca / modelo.
 - Estándar (a/b/g/WiMax)
 - Canal.
 - Ssid.
 - Dirección MAC en LAN y WLAN.
 - IP administración.
 - Usuario/contraseña administrador.
 - Enlaces WDS con otros puntos de acceso.
- Servidor(-es):
 - Marca / modelo.
 - Versión SO.
 - Usuario / contraseña (usuario sin privilegios)
 - Contraseña de administrador.
 - Adaptadores de red: identificador (ethx, tapx), IP, MAC y tipo de adaptador (físico o virtual).
 - VPN con otros centros.
 - Servicios funcionando: nombre, puerto y protocolo (TCP o UDP)
- Configuración del nodo:
 - Subdominio en la XEiLL
 - Bloques IPs asignados (por DHCP a los clientes o a los servidores del centro).
 - Nombres DNS registrados: nombre, IP de internet e IP de la XEiLL.
 - Información de encaminamiento: IP y área de cada uno de los nodos de los 2 extremos.

[RF-004] Se debe permitir una monitorización básica de la red.

[RF-005] Se debe llevar un registro de actuaciones/incidencias. Pudiendo añadir, el usuario, un comentario a cada registro.

6.1.3. Entradas y salidas del sistema

Entradas

Las entradas de este sistema serán los campos y botones de los formularios de entrada de datos que tenga la aplicación. Los primeros para introducir los datos y los segundos para ejecutar acciones (añadir, actualizar, borrar)

Salidas

Este sistema sólo tendrá un tipo de salida que será la pantalla del ordenador por donde se mostrarán todos los mensajes de interacción del usuario (avisos y errores) y los datos de salida.

6.1.4. Gestión de errores del sistema

Los principales errores que se pueden producir en la aplicación son errores de entrada de datos (formato de datos incorrecto, faltan datos por especificar, etc.) o de base de datos.

Una vez se produzca cualquiera de estos errores lo que se hará es mostrar, mediante una ventana de diálogo, un mensaje de error al usuario, explicándole el error y las acciones que debe realizar (si es el caso).

6.2. Definición de la persistencia de datos

Dado que la aplicación del proyecto es para la gestión de datos, se necesitará de un sistema de persistencia el cual utilizaremos para guardar todos los datos de la aplicación y así, posteriormente, poderlos recuperar para su correcta manipulación (inserción, eliminación, modificación, listado, etc.)

A continuación se pasará a analizar los elementos más importantes del sistema de persistencia escogido:

- **Sistema gestor de base de datos:** aplicación escogida para la gestión del almacenaje persistente de los datos. Deberá de tratarse de una aplicación estándar y necesaria para el volumen de nuestra aplicación.
- **Sistema de persistencia:** Sistema o tecnología usada para la implementación de la persistencia y el acceso al Sistema Gestor de base de datos (SGBD)
- **Diagrama Entidad-Relación:** finalmente se mostrará el diagrama Entidad-Relación de las entidades de la base de datos y se comentarán brevemente las más importantes.

6.2.1. Sistema gestor base de datos

El Sistema gestor de base de datos (también conocido como SGBD) es la aplicación encargada de llevar la gestión y manipulación del almacenaje de datos que necesita la aplicación final. Para hacer independiente de la aplicación final, el soporte físico en el que se guardan los datos y el acceso a ellos, suelen proveer de una serie de lenguajes para facilitar dicha tarea: un lenguaje de definición de datos, un lenguaje de manipulación de datos y, finalmente, un lenguaje de consulta de datos.

Algunos de los objetivos principales de los SGBD son la persistencia, consistencia, seguridad e integridad de los datos. En la actualidad los SGBD más importantes (tanto libres como comerciales) implementan en lenguaje estándar SQL para la definición, manipulación y consulta de los datos de la base de datos.

En el caso de nuestra aplicación gráfica, la necesidad de un SGBD viene dada por el hecho de tener que guardar toda la información de cada centro. La correcta gestión de los centros nos obliga a tener que guardar en algún dispositivo físico persistente todos los datos, de forma que, en posteriores ejecuciones de la aplicación podamos volver a recuperar los datos.

El hecho de que la información deba ser compartida por diferentes usuarios, ubicados en zonas dispersas también nos implica que no sólo necesitemos asegurar la recuperabilidad de los datos sino también el acceso centralizado a los mismos. Esto hace también la necesidad de tener un SGBD que permita el acceso remoto a los datos de la aplicación.

MySQL es uno de los SGBD sencillos, libres y de código abierto más importantes en la actualidad, con un gran soporte de toda la comunidad de Internet. A parte de tratarse de un sistema muy respaldado por la comunidad de programadores y de tratarse de software libre, también cumple los requisitos imprescindibles tanto de un SGBD, como de las necesidades de nuestra aplicación: acceso remoto, seguridad, integridad de los datos, etc.

Otros de los aspectos importantes que implementa MySQL es el hecho que usa el lenguaje estándar SQL como lenguaje de manipulación y acceso a los datos. Este aspecto permite la opción de migrar a otro SGBD diferente (que soporte el estándar SQL) sin que la aplicación lo notase, ni tuviese que sufrir cambios.

Todos estos aspectos y el hecho que nuestra aplicación no tenga la necesidad de guardar, por el momento, grandes volúmenes de información, ni de soportar una alta concurrencia en el acceso simultáneo de usuarios, se cree que con el Sistema Gestor de Base de Datos MySQL hay suficiente para el desarrollo de la misma. En todo caso, el hecho de tratarse de un producto que cumple con los estándares de los SGBD nos permitiría poder cambiar a otro SGBD estándar sin se considerase oportuno.

6.2.2. Sistema de persistencia

Los sistemas de persistencia son sistemas o *frameworks* encargados de hacer completamente transparente la aplicación final del SGBD que se está usando. En el caso de Java permiten que la aplicación pueda acceder a los datos de un SGBD sin necesidad de conocer el que se está usando y de llevar toda la gestión del driver JDBC (Java DataBase Connectivity: API Java que permite el acceso a un SGBD ofreciendo métodos para: iniciar/cerrar la conexión, lanzar consultar a la base de datos, gestionar transacciones, etc.)

Algunos de los Sistemas de persistencias más importantes actualmente, en el mundo Java son:

- **EJB: Enterprise Java Beans.** Se trata de objetos Java (que forman parte del estándar Sun conocido como J2EE) que residen en un servidor de aplicaciones y a los que se puede acceder a ellos de forma remota. De entre los diferentes tipos de EJB existentes, están los conocidos como *Entity EJBs* que son los encargados de la encapsulación del acceso a los datos.
- **JDO: Java Data Objects.** El último estándar Sun en el acceso a datos persistentes. Se trata de una tecnología que usa POJOs (*Plain Old Java Objects*: nombre por el que se conocen a las clases normales de Java, que no son de ningún tipo especial: EJB, Bean, etc.). El acceso a los datos viene definido en fichero XML que son los encargados de configurar tanto el acceso al SGBD (servidor, base de datos, etc.), como el mapeo entre el objeto java y las diferentes entidades de la base de datos.

- Hibernate: Se trata de una herramienta libre y de código abierto que permite el acceso a datos de una forma sencilla como los JDO (a través de POJOS y ficheros XML). A diferencia de JDO, no se trata de un estándar Sun, sino de un producto desarrollado por una empresa independiente.

Debido a la poca complejidad de la aplicación, de la base de datos y de la poca concurrencia que se espera en esta aplicación, se ha decidido usar, como motor de persistencia, la herramienta *Hibernate*. Después de evaluar diferentes tecnologías (EJB, JDO y la misma Hibernate) se ha decidido usar ésta por las siguientes razones:

1. En comparación con los EJB, se trata de una capa de persistencia ligera, muy indicada para aplicaciones como la nuestra, con poca carga de datos, accesos y no tener una gran necesidad de un control de la concurrencia.
2. Muy indicada para aplicaciones sencillas, dada su facilidad de uso, configuración y gran eficiencia.
3. Su gran secreto reside en el hecho de trabajar con POJOS (*Plain Old Java Objects*) como representación de las entidades en la base de datos, y ficheros XML para realizar el mapeo Java-BBDD. Su uso es tan sencillo como el de usar los conocidos *Beans* de Java.
4. A pesar de no tratarse de una tecnología estándar, como son los JDO de Sun, si que se trata de un estándar *de facto* por el hecho de haberse extendido su uso de forma muy rápida por toda la comunidad Java, contando con su respaldo y soporte.
5. La gran ventaja, respecto los JDO, de tratarse de una tecnología y no de un estándar es que no hay múltiples implementaciones, sólo hay una y, por lo tanto, es más fácil de obtener un buen mantenimiento y buena documentación del desarrollador de esta aplicación.
6. Existen multitud de herramientas y plug-ins (desarrollados tanto por la comunidad Java, como por el propio desarrollador de la herramienta) que facilitan y hacen más rápido el desarrollo con *Hibernate*.
7. Trabaja con multitud de Sistemas Gestores de Bases de Datos.
8. Cumple con uno de los requisitos más importantes que eran necesarios para el desarrollo de este proyecto: se trata de software libre.

Finalmente, comentar que se tuvieron inicialmente en cuenta otras tecnologías (como Castor u OJB de Jakarta), pero se descartaron dada su poca repercusión dentro de la comunidad Java.

6.2.3. Diagrama entidad relación

A continuación se muestra una imagen del diagrama Entidad-Relación de la base de datos.

Memoria del Proyecto

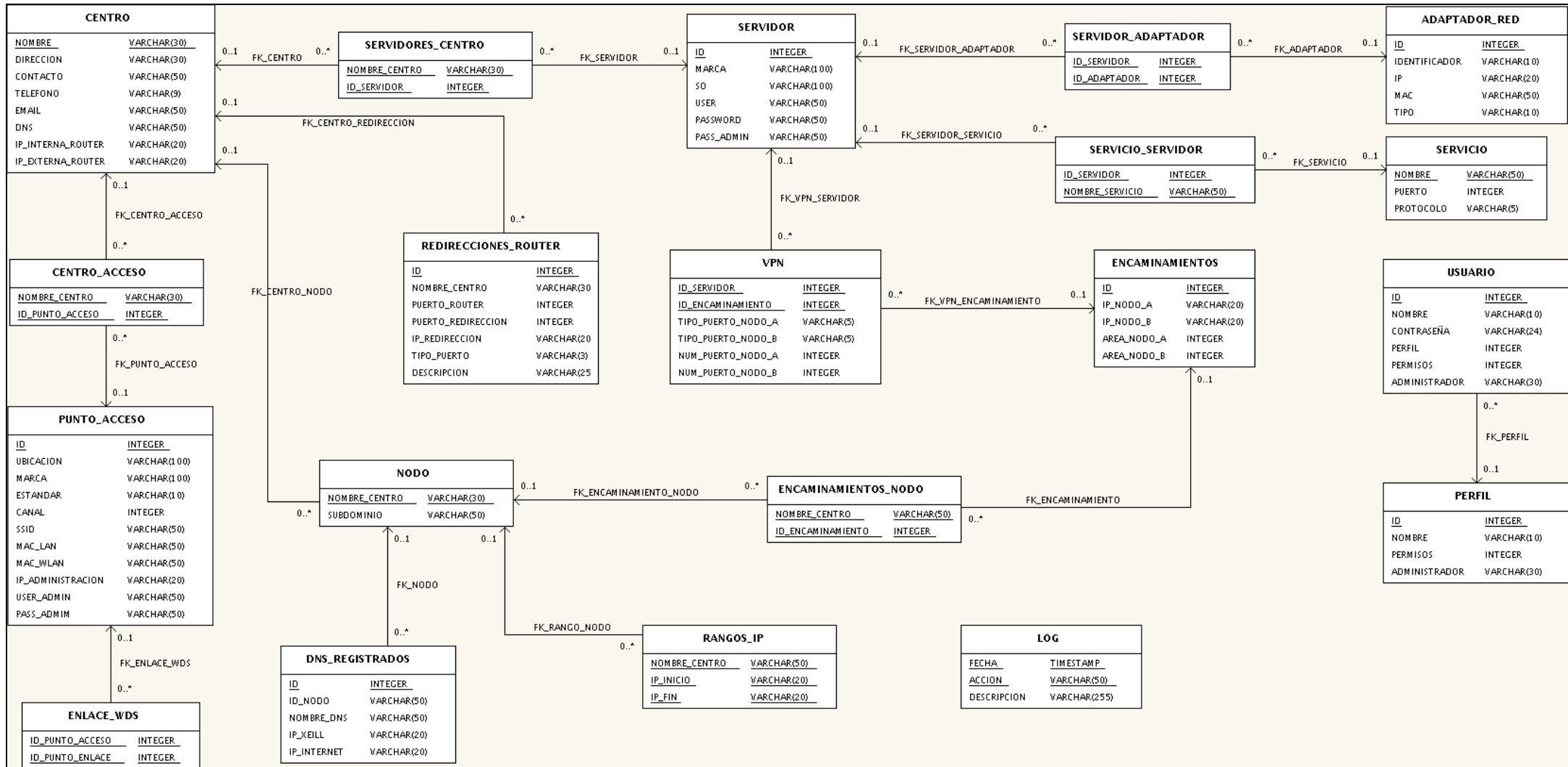


Imagen 2: Diagrama E-R

A continuación se pasan a listar las entidades más importantes de este diagrama. En rojo están indicadas las claves primarias de cada tabla:

Tabla CENTRO:

Tabla que guarda toda la información de un centro que funciona como nodo de la red XEiLL.

A continuación se detallan los campos de esta tabla:

Nombre	Tipo de dato	Descripción
<i>Nombre</i>	<i>VARCHAR(30)</i>	<i>Nombre del centro.</i>
Direccion	VARCHAR(30)	Dirección postal del centro.
Contacto	VARCHAR(50)	Persona de contacto del centro en cuestión.
Telefono	VARCHAR(9)	Teléfono de contacto del centro.
Email	VARCHAR(50)	Dirección de correo electrónico de contacto del centro.
DNS	VARCHAR(50)	DNS del centro.
Ip_interna_router	VARCHAR(20)	IP interna del router del centro
Ip_externa_router	VARCHAR(20)	IP externa del router del centro

Tabla REDIRECCIONES_ROUTER:

Posibles redirecciones de puertos entre la interfaz externa del router de un centro y cualquier otra máquina. Por ejemplo: el puerto 80 de la interfaz externa del router está redirigido al puerto 80 de la máquina 192.168.0.22.

A continuación se detallan los campos de esta tabla:

Nombre	Tipo de dato	Descripción
<i>Id</i>	<i>INTEGER</i>	<i>Identificador interno del registro.</i>
Nombre_centro	VARCHAR(30)	Nombre del centro al cual pertenece la redirección de puerto del registro actual. <i>Foreign key</i> a la tabla CENTRO.
Puerto_router	INTEGER	Puerto del router que está redireccionado.
Puerto_redireccion	INTEGER	Puerto al que se redirecciona
Ip_redireccion	VARCHAR(20)	IP de la máquina a la que se hace la redirección.
Tipo_puerto	VARCHAR(3)	Tipo de puerto usado en la redirección de routers: TCP o UDP.

Descripción	VARCHAR(255)	Breve comentario sobre el redireccionamiento.
-------------	--------------	---

Tabla PUNTO_ACCESO:

Punto de acceso WiFi instalado en un centro.

A continuación se detallan los campos de esta tabla:

Nombre	Tipo de dato	Descripción
Id	INTEGER	Identificador único de un punto de acceso.
ubicacion	VARCHAR(100)	Lugar del centro donde está ubicado el punto de acceso.
marca	VARCHAR(100)	Marca y modelo del hardware usado en el punto de acceso.
estandar	VARCHAR(10)	Estándar WiFi que sigue este punto de acceso (a, b, g, WiMax,...)
canal	INTEGER	Canal sobre el que trabaja este punto de acceso.
ssid	VARCHAR(50)	Identificador del área de servicio del punto de acceso.
mac_lan	VARCHAR(50)	Dirección MAC de la red LAN de este punto de acceso.
mac_wlan	VARCHAR(50)	Dirección MAC de la conexión WLAN de este punto de acceso.
ip_administracion	VARCHAR(20)	Dirección IP de administración del punto de acceso.
user_admin	VARCHAR(50)	Nombre de usuario de administración del punto de acceso.
pass_admin	VARCHAR(50)	Contraseña de administración del punto de acceso.

Tabla SERVIDOR:

Información de un servidor instalado en un centro.

A continuación se detallan los campos de esta tabla:

Nombre	Tipo de dato	Descripción
Id	INTEGER	Identificador único del servidor.
Marca	VARCHAR(100)	Marca y modelo del servidor.
So	VARCHAR(100)	Sistema Operativo y versión que hay instalado.
User	VARCHAR(50)	Nombre del usuario sin privilegios que hay dado de alta.

Password	VARCHAR(50)	Contraseña del usuario sin privilegios que hay dado de alta.
Pass_admin	VARCHAR(50)	Contraseña del usuario administrador.

Tabla ADAPTADOR_RED:

Adaptadores de red instalados en un servidor.

A continuación se detallan los campos de esta tabla:

Nombre	Tipo de dato	Descripción
id	VARCHAR(30)	Identificador interno del registro.
identificador	VARCHAR(10)	Identificador del adaptador de red (ipX, ethX,...)
Ip	VARCHAR(20)	Dirección IP del adaptador de red.
Mac	VARCHAR(50)	Dirección MAC del adaptador de red.
Tipo	VARCHAR(10)	Tipo del adaptador de red: físico o virtual.

Tabla SERVICIO:

Información necesaria de un servicio que corre en el servidor.

A continuación se detallan los campos de esta tabla:

Nombre	Tipo de dato	Descripción
Nombre	VARCHAR(50)	Nombre del servicio.
Puerto	INTEGER	Puerto en el que está escuchando el servicio.
Protocolo	VARCHAR(5)	Protocolo de comunicación usado por el servicio (TCP o UDP)

Tabla ENCAMINAMIENTOS:

Tabla con los encaminamientos entre dos nodos.

A continuación se detallan los campos de esta tabla:

Nombre	Tipo de dato	Descripción
Id	INTEGER	Identificador del encaminamiento.
Ip_nodo_a	VARCHAR(20)	IP virtual del nodo A del encaminamiento del nodo del centro.
Ip_nodo_b	VARCHAR(20)	IP virtual del nodo B del encaminamiento del nodo del

		centro.
Area_nodo_a	INTEGER	Área del nodo A del encaminamiento del nodo del centro.
Area_nodo_b	INTEGER	Área del nodo B del encaminamiento del nodo del centro.

Tabla VPN:

Tabla con los diferentes túneles que hay entre dos servidores. Una VPN no deja de ser un encaminamiento al cual también se le indica el tipo y número de puerto de los dos extremos conectados.

A continuación se detallan los campos de esta tabla:

Nombre	Tipo de dato	Descripción
Id_servidor	INTEGER	Identificador del servidor en el cual está asignada esta VPN. <i>Foreign key</i> a la tabla SERVIDOR.
id_encaminamiento	INTEGER	Identificador del encaminamiento de la VPN. <i>Foreign key</i> a la tabla ENCAMINAMIENTOS.
Tipo_puerto_nodo_a	VARCHAR(5)	Tipo del puerto (TCP o UDP) del extremo A de la VPN.
Tipo_puerto_nodo_b	VARCHAR(5)	Tipo del puerto (TCP o UDP) del extremo B de la VPN.
Num_puerto_nodo_a	INTEGER	Número de puerto del extremo A de la VPN.
Num_puerto_nodo_b	INTEGER	Número de puerto del extremo B de la VPN.

Tabla NODO:

Información del nodo WiFi que hay instalado en un centro educativo.

A continuación se detallan los campos de esta tabla:

Nombre	Tipo de dato	Descripción
Nombre_centro	VARCHAR(30)	Nombre del centro al cual pertenece el nodo. <i>Foreign key</i> a la tabla CENTRO.
subdominio	VARCHAR(50)	Subdominio en la XEILL del nodo.

Tabla DNS_REGISTRADOS:

DNS registrados en un nodo.

A continuación se detallan los campos de esta tabla:

Nombre	Tipo de dato	Descripción
id	INTEGER	<i>Identificador interno del registro.</i>
id_nodo	VARCHAR(50)	<i>Foreign key a la tabla NODO.</i>
Nombre_dns	VARCHAR(50)	nombre del DNS (p.e.: jabber.centrox.xeill.net)
Ip_xeill	VARCHAR(20)	IP de la XEILL a la que está asignado este DNS.
Ip_internet	VARCHAR(20)	IP de internet a la que está asignado este DNS.

Tabla RANGOS_IP:

Tabla que guarda una lista con diferentes rangos de IPs.

A continuación se detallan los campos de esta tabla:

Nombre	Tipo de dato	Descripción
nombre_centro	VARCHAR(50)	<i>Nombre del centro donde está el nodo al cual se le asigna el rango IP. Foreign key a la tabla NODO.</i>
Ip_inicio	VARCHAR(20)	<i>IP de inicio del rango.</i>
Ip_fin	VARCHAR(20)	<i>IP de fin del rango.</i>

Tabla LOG:

Tabla que guarda todo el registro de incidencias/acciones realizadas en la base de datos.

A continuación se detallan los campos de esta tabla:

Nombre	Tipo de dato	Descripción
Fecha	TIMESTAMP	<i>Fecha en la que ha ocurrido la incidencia.</i>
Accion	VARCHAR(50)	<i>Acción realizada (inserción, modificación, eliminación, etc.)</i>
Descripcion	VARCHAR(255)	Descripción añadida por el usuario de la aplicación sobre este registro.

Tabla USUARIO:

Tabla que guarda la información de todos los usuarios dados de alta en este sistema y que pueden acceder a él.

A continuación se detallan los campos de esta tabla:

Nombre	Tipo de dato	Descripción
Id	INTEGER	Identificador interno del registro.
Nombre	VARCHAR(10)	Nombre de usuario.
Contraseña	VARCHAR(24)	Contraseña del usuario, guardada de forma cifrada con MD5.
Perfil	INTEGER	Perfil de usuario, en el caso de tener este usuario uno asignado. <i>Foreign key</i> a la tabla PERFIL.
Permisos	INTEGER	Máscara con los permisos de acceso de este usuario.
Administrador	BOOLEAN	Indica si este usuario tiene acceso de administrador (permisos para gestionar los usuarios y el registro de incidencias)

Tabla PERFIL:

Tabla que guarda la información de todos los perfiles de usuario dados de alta en este sistema y que pueden ser asignados a un usuario.

A continuación se detallan los campos de esta tabla:

Nombre	Tipo de dato	Descripción
Id	INTEGER	Identificador interno del registro.
Nombre	VARCHAR(10)	Nombre del perfil.
Permisos	INTEGER	Máscara con los permisos de acceso de este perfil.
Administrador	BOOLEAN	Indica si este perfil tiene acceso de administrador (permisos para gestionar los usuarios y el registro de incidencias)

6.3. Definición de la interfaz gráfica

6.3.1. Tecnología escogida

Una de las condiciones importantes a la hora de diseñar la interfaz gráfica de la aplicación es que ésta sea accesible por cualquier usuario, sea cual sea su

ubicación y el lugar desde el cual accede. Además, el usuario, desde la aplicación que desarrollemos, debe poder acceder remotamente a toda la información de la aplicación, pudiéndola compartir con todos los usuarios de la misma.

Partimos con dos opciones que satisfagan los dos puntos anteriores:

1. Aplicación *standalone*, que cada usuario se instale en su propia máquina y que acceda remotamente al servidor de base de datos.
2. Aplicación web que se pueda acceder a través de un explorador de Internet, de forma que no haga falta, por parte de ningún usuario la instalación de ninguna aplicación.

Finalmente, podría también interesar una solución que cumpliera las dos características: una aplicación web que permita compartir datos con otros usuarios y una aplicación *standalone* para algún usuario que quiera, de forma local, gestionar sus propios datos de centros educativos.

Por la flexibilidad que permite y el acceso a datos compartidos con otros usuarios (que es el principal objetivo de la aplicación) se ha decidido, finalmente por una aplicación a la cual se pueda acceder a través de un navegador de Internet.

Una vez llegados a este punto, hay diferentes tecnologías que satisfacen nuestras necesidades, pero las principales son:

1. PHP: lenguaje de programación usado para la creación de aplicaciones web dinámicas.
2. J2EE: estándar de Sun, basado en el lenguaje de programación Java y que incorpora diferentes especificaciones entre las que destacan: JSP (lenguaje semejante a PHP pero usando el lenguaje de programación Java), Servlets (aplicaciones que se ejecutan en un servidor y a la que se acceden remotamente) y EJBs.
3. Java Applet: Aplicación gráfica (también conocida como cliente rico) que usa el lenguaje de programación Java y que se ejecuta en el contexto de un navegador web.

Todas las tecnologías tienen sus ventajas y desventajas, pero finalmente se ha decidido usar los Java Applets por la potencia gráfica que nos aporta: en el desarrollo de interfaces gráficas nos permite hacer más cosas o las mismas que PHP o JSP, pero con menos esfuerzo.

Algunas ventajas que nos aporta un Java Applet respecto las otras dos tecnologías son:

1. Cuando se trata de interfaces gráficas, es posible igualar o superar los resultados, invirtiendo menos esfuerzo.
2. Al tratarse de una aplicación gráfica en Java, se puede desarrollar fácilmente una aplicación que pueda servir tanto como Applet, como aplicación *standalone*.
3. Una vez está cargado el Applet en el navegador del usuario, el uso de la aplicación se hace como una aplicación cliente residente en el PC del usuario, siendo las únicas conexiones remotas los accesos a datos.

Como he comentado también tiene algunas desventajas, como son:

1. Al tenerse que cargar toda la aplicación cliente en el navegador, la primera conexión es más lenta que en un entorno web como PHP o JSP.
2. Se requiere que el navegador que se use para acceder tenga instalado un plug-in de Java en su ordenador para poder ejecutar la aplicación.

Finalmente conviene comentar que existen herramientas libres que permiten crear una capa web por encima de una aplicación gráfica realizada con Java, residiendo la aplicación gráfica en un servidor y usando el usuario una interfaz web, como si fuese PHP o JSP (un ejemplo de este tipo de herramientas sería *WebOnSwing*¹). Esta solución nos abriría 3 posibles opciones a ofrecer a los usuarios de la herramienta:

1. Acceso a la aplicación gráfica a través de un cliente rico ejecutado en un navegador de Internet: Java Applet.
2. Acceso a la aplicación gráfica a través de un cliente rico ejecutado desde el propio ordenador del usuario: aplicación Java *standalone*.
3. Acceso a la aplicación gráfica a través de un cliente liviano ejecutado desde un navegador de Internet: *framework* que crea una capa HTML por encima de un cliente rico (como por ejemplo, el ya comentado *WebOnSwing*).

6.3.2. Interfaz gráfica

Hablando del diseño de la interfaz gráfica, propiamente dicha, uno de los requisitos principales que se ha tenido en cuenta a la hora de diseñar la interfaz gráfica de la aplicación, es que se trate de un diseño atractivo y que a la vez resulte intuitivo y fácil de aprender por parte del usuario de la misma, sin que necesite de una gran información y documentación para saberla usar.

En la siguiente figura se puede observa una imagen de la aplicación:

¹ Breve descripción en la referencia bibliográfica nº 5

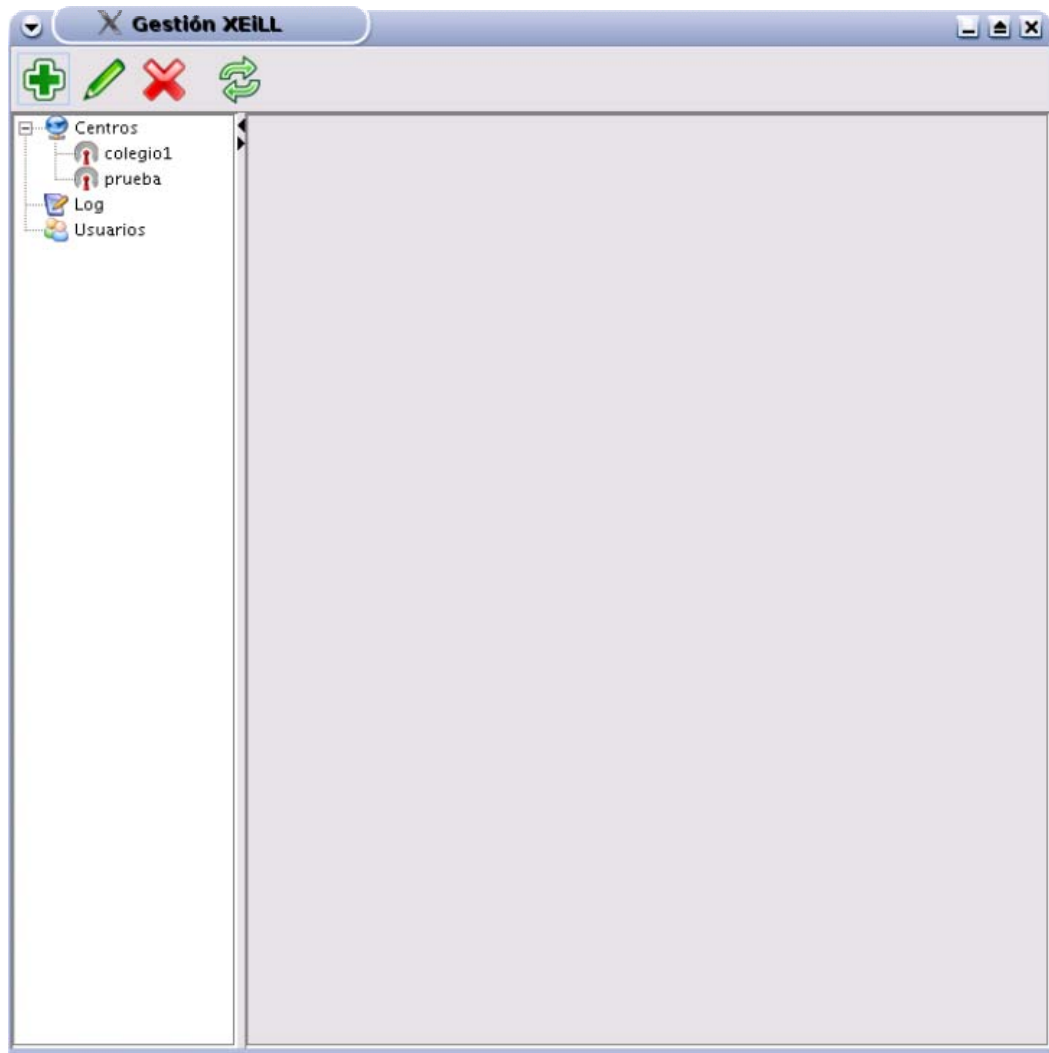


Imagen 3: Ejemplo diseño 1

En esta imagen se puede observar que la aplicación esta dividida en 3 secciones:

- ***Barra de herramientas:*** Se encuentra en la parte superior de la ventana. En esta barra se encuentran las 4 opciones que permiten trabajar con los datos: crear nuevo centro, editar el centro seleccionado, borrar centro y refrescar la lista de centros.
- ***Área de navegación:*** Se muestra un árbol con las diferentes entidades con las que se puede trabajar. La entidad *centros* contiene una lista con el nombre de todos los centros que están dados de alta en la base de datos. La entidad *log* permite ver una lista de registro de incidencias que se ha ido generando. Finalmente la opción de *usuarios* es la que permite acceder a la pantalla de gestión de los usuarios y perfiles de la aplicación.

- Área de trabajo: Como el nombre indica será el área donde se mostrará el contenido de la información que se esté tratando en cada momento.

En la siguiente imagen se puede ver un ejemplo de cómo se visualizaría el registro de incidencias de la aplicación:

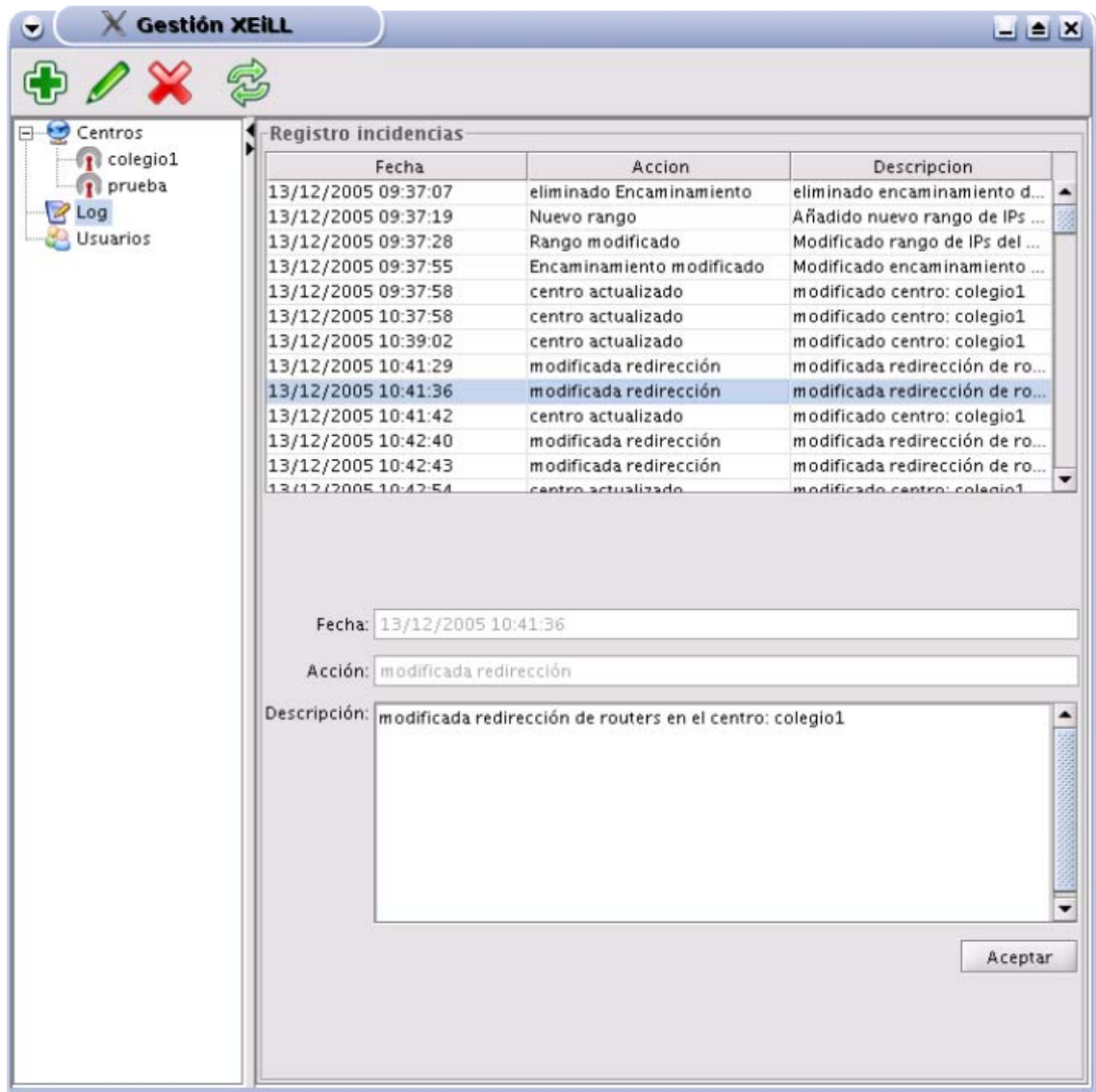


Imagen 4: Ejemplo diseño 2

A continuación, se puede observar un ejemplo de la pantalla para gestionar los centros:

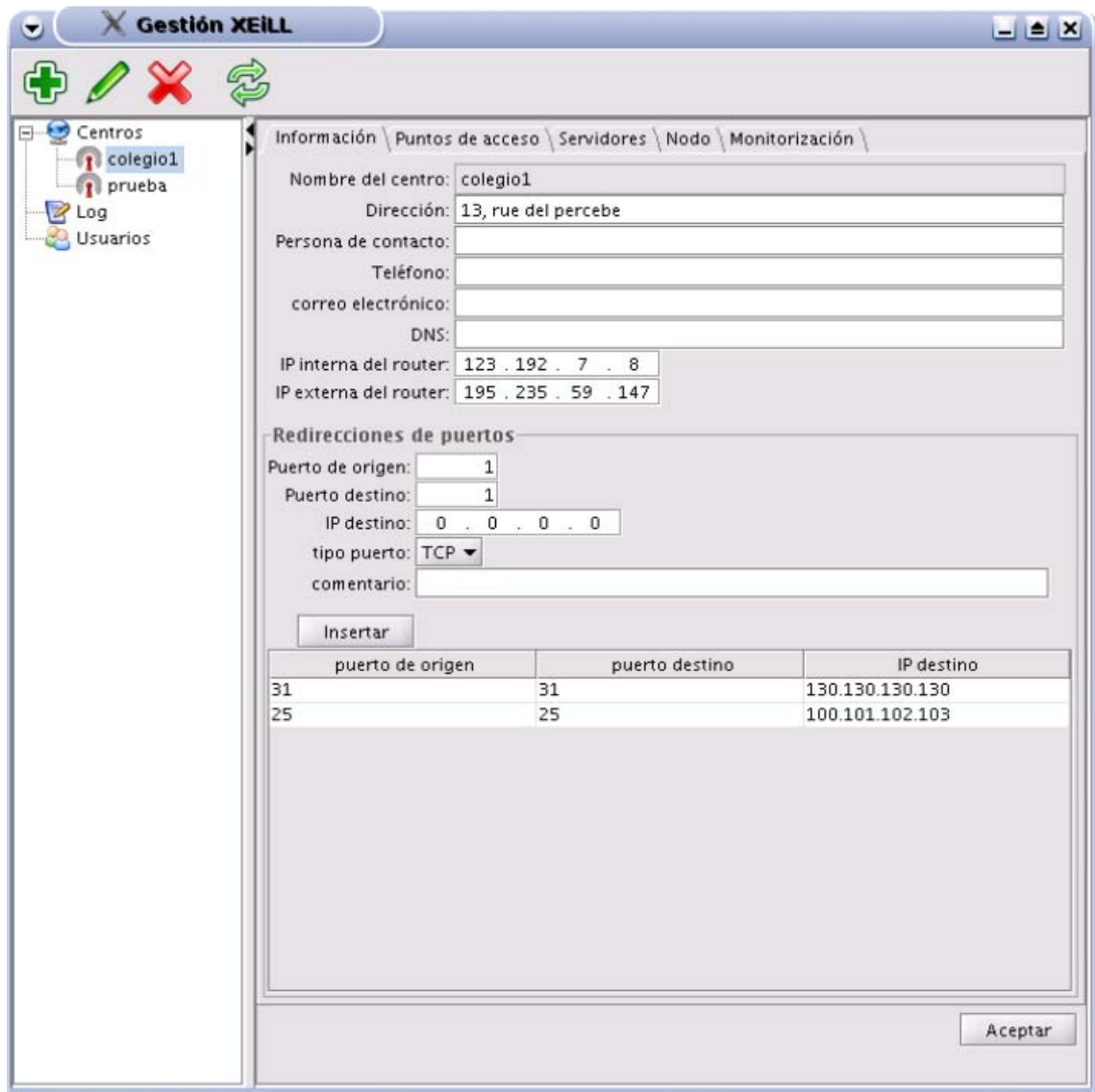


Imagen 5: Ejemplo diseño 3

Finalmente, se puede observar una pantalla de lo que sería la herramienta de gestión de usuarios y perfiles:

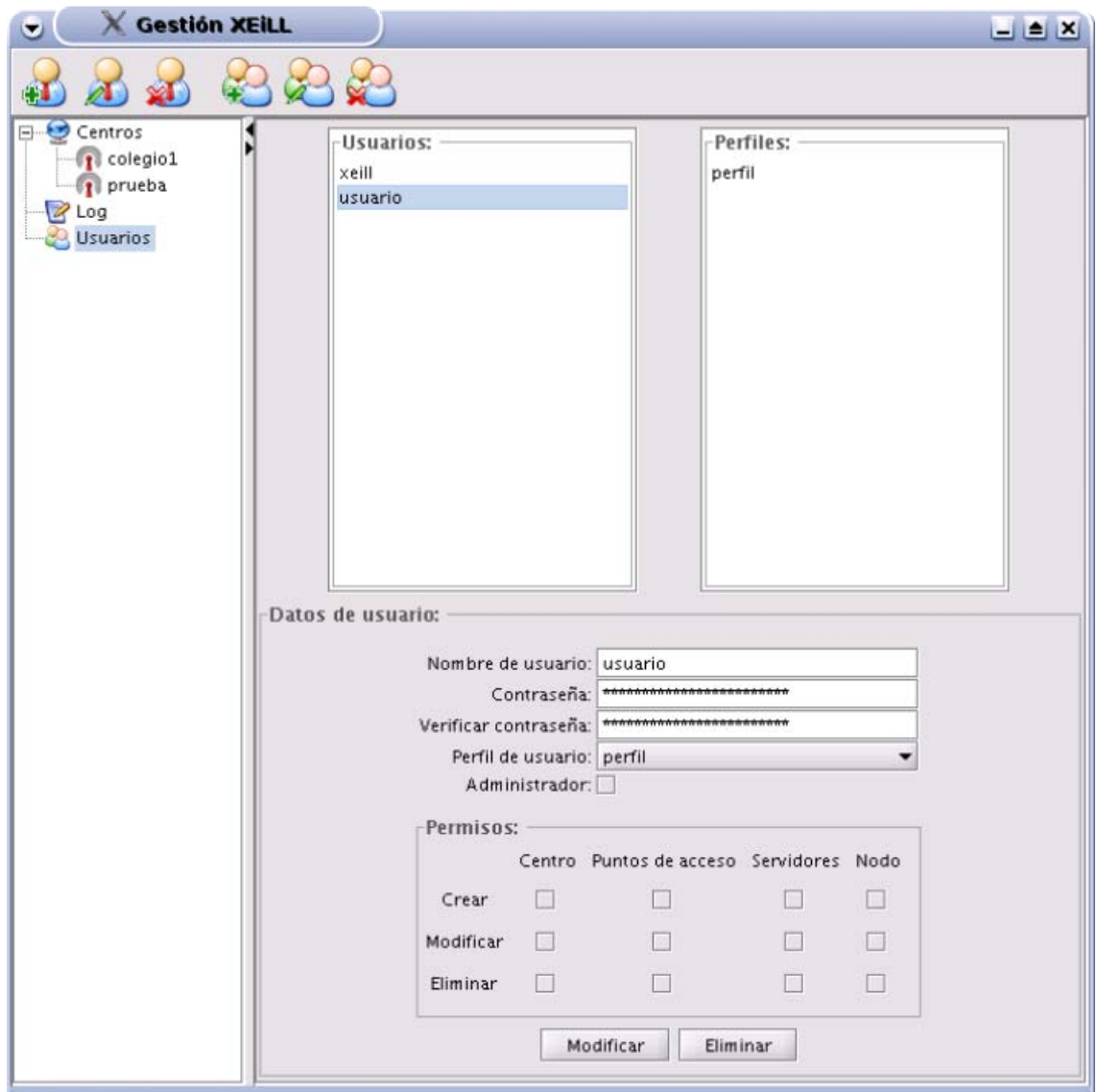


Imagen 6: Ejemplo diseño 4

7. LA APLICACIÓN

Una vez definidos los requisitos (funcionales y no funcionales) de la aplicación, el sistema de persistencia a utilizar y la interfaz gráfica escogida (tanto diseño, como tecnología), en este apartado se pasará a definir y explicar la implementación final realizada, basándonos en la información de los apartados anteriores.

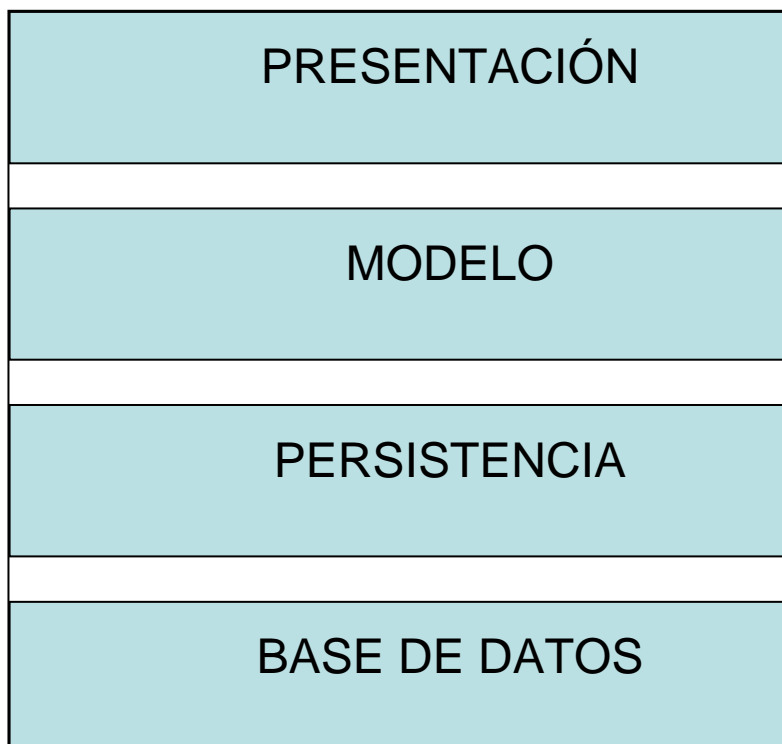
Esta explicación estará dividida en dos secciones diferentes:

1. Arquitectura: Se explicará la arquitectura o división de módulos realizada para la implementación de la aplicación.
2. Aplicación: La aplicación en sí, su funcionamiento, sus diferentes pantallas y objetivos, etc.

7.1. Arquitectura

La aplicación se ha intentado modular en diferentes capas, según las funciones que realiza cada capa. En algunos casos, la interrelación entre dos capas llega a ser tanta que puede llegar a no quedar muy claro el límite entre ambas, donde empieza una y termina la anterior.

En el siguiente esquema se puede ver una imagen de esta arquitectura:



7.1.1. Presentación

Es la capa en la que está implementada la interfaz gráfica de la aplicación. Es la encargada de la gestión de la interacción con el usuario, recibiendo los eventos de las acciones de éste sobre la aplicación y de mostrarle toda la información y avisos al mismo.

A nivel tecnológico, está implementada con el paquete Swing de la API de Java, API para la representación gráfica de interfaces de usuario avanzadas.

Para poder mostrar toda la información al usuario y recoger la información introducida, deberá interactuar con la siguiente capa, la del “Modelo”.

7.1.2. Modelo

Es la capa en la que está implementado el modelo de datos de la aplicación. En el caso de la nuestra, el modelo de datos es, básicamente, las diferentes entidades de datos y alguna clase de utilidad.

En la mayoría de casos, de las clases de esta capa, cada una de ellas representa una entidad de la base de datos, en la cual están guardados (mediante atributos de instancia) los datos de la misma y se pueden acceder a ellos (mediante métodos de instancia)

Se trata de clases Java estándares (conocidas como POJO) y que necesitan interactuar con la capa de “Persistencia” para poder cargar/guardar los datos de la base de datos.

Todas estas clases están ubicadas dentro del paquete *xeill.model*. Como se ha comentado anteriormente, este paquete contiene todas las clases que representan una tabla de la base de datos y la clase de utilidad *Encrypt*. Esta clase proporciona dos métodos utilizados para el cifrado de las contraseñas de usuarios:

- md5Encrypt: método que cifra una contraseña, pasada como parámetro, mediante el método de calcular su módulo MD5.
- Sh1Encrypt: Este método, actualmente no se usa, pero lo que hace es cifrar la contraseña pasada como parámetro aplicándole el algoritmo de cifrado SHA-1.

7.1.3. Persistencia

Esta es la capa encargada de interactuar con la base de datos. Hace transparente a la aplicación tanto el acceso a la base de datos como el SGBD usado, dando la posibilidad de cambiar de sistema (dentro de los estándares) sin que la aplicación se vea afectada.

Los detalles de esta capa ya han sido explicados en el apartado “Sistema de Persistencia”

Remarcar de las características de esta capa, la posibilidad de configurar el acceso a la base de datos. Mediante un fichero XML de configuración, que proporciona Hibernate, podremos configurar el acceso a la base de datos ya sea porque cambiamos el servidor, porque cambiamos el SGBD utilizado, etc. El fichero que se utiliza para esto es **hibernate.cfg.xml** y de todos los parámetros de configuración, los que más nos interesan a nosotros son:

```
<property name="hibernate.connection.url"> jdbc:mysql://[HOST]:[PORT]/[DB] </property>  
<property name="hibernate.connection.driver_class">[DRIVER JDBC]</property>  
<property name="hibernate.connection.username">[USERNAME]</property>  
<property name="hibernate.connection.password">[PASSWORD]</property>
```

En negrita se encuentran la información que se debe especificar:

1. El primero es la URL JDBC de conexión a la base de datos. La del ejemplo es la necesaria para MySQL, pero cambia dependiendo de la base de datos a usar.
2. El segundo parámetro es el driver JDBC necesario para realizar la conexión a la base de datos. En el caso de MySQL es: *com.mysql.jdbc.Driver*
3. El tercer parámetro es el nombre de usuario a usar para conectarse a la base de datos.
4. El cuarto y último parámetro es la contraseña de conexión a la base de datos.

Hibernate, para la implementación de la capa de persistencia, trabaja con la tecnología conocida como O/RM (Object-Relational mapping) que se trata de una técnica de programación que lo que hace es enlazar una base de datos a conceptos de un lenguaje orientado a objetos (p.e.: una tabla es una clase, cada registro de la tabla una instancia u objeto y cada campo un atributo de la clase).

Esta técnica de programación permite que, por ejemplo, una consulta SQL termine construyendo un objeto Java (en el caso de trabajar con el lenguaje orientado a objetos de Java) o que tengamos toda la base de datos modelada en una arquitectura de objetos.

Evidentemente para poder llevar a cabo el O/RM se necesita indicarle a la herramienta que lo implementa (en nuestro caso Hibernate) cuál es el mapeo que hay que realizar entre las clases de Java y la base de datos. En el caso de Hibernate esto se hace mediante diferentes ficheros XML de configuración, en los cuales se indican las diferentes relaciones entre una tabla de la base de datos y la correspondiente clase Java que la modela y representa.

En este fichero no sólo se define la relación tabla-clase, sino también otros conceptos como relaciones campo de tabla-atributo de clase, estructura dónde guardar una relación que no sea uno-a-uno, etc.

A continuación se muestra un ejemplo de un fichero XML de mapeo de una clase java con una tabla de nuestra base de datos:

```
<hibernate-mapping package="xeill.persistence">
  <class name="xeill.model.Usuario" table="USUARIO">
    <id name="Id" type="java.lang.Integer" unsaved-value="null">
      <column name="id" not-null="true" unique="true" sql-type="INTEGER"/>
      <generator class="increment"/>
    </id>

    <property name="nombre" type="java.lang.String">
      <column name="nombre" length="10" not-null="true" unique="true" sql-
type="VARCHAR"/>
    </property>

    <property name="contraseña" type="java.lang.String">
      <column name="contraseña" length="16" sql-type="VARCHAR"/>
    </property>

    <many-to-one name="perfil" class="xeill.model.Perfil" column="perfil" unique="true"
cascade="none"/>

    <property name="permisos" type="java.lang.Integer">
      <column name="permisos" sql-type="INTEGER"/>
    </property>

    <property name="administrador" type="java.lang.Boolean">
      <column name="administrador" sql-type="BOOLEAN"/>
    </property>
  </class>
</hibernate-mapping>
```

La etiqueta **class** es la que se utiliza para realizar el mapeo clase-tabla. En el atributo *name* se debe indicar el nombre de la clase (especificando el nombre completo: package + nombre de clase) con la que estamos tratando y en el atributo *table* cuál es la tabla de la base de datos a la que está mapeando.

Dentro de este nodo, encontramos los nodos **property** que son los usados para indicar la relación entre cada uno de los campos de la base de datos y los atributos de la clase. Con los atributos, del nodo, *name* y *type*, se indica el nombre del atributo de clase y el tipo de dato que es, respectivamente. Este nodo, tiene el nodo hijo **column** que es el que se usa para indicar el nombre del campo de la tabla y su tipo de datos al que mapea el atributo que se está tratando.

Finalmente, otra de los nodos importantes, que se pueden observar en este ejemplo, es el nodo **many-to-one**, que es uno de los varios que tiene Hibernate

para especificar las *foreign keys* de la base de datos, que en lenguaje java se traducen como referencias a otras clases (ya sean mediante una instancia de una clase para una relación uno-a-uno o mediante lista de objetos para el resto)

7.1.4. Base de datos

Finalmente está esta última capa que es la encargada de la gestión del almacenamiento físico de los datos que usa la aplicación. Facilita una serie de lenguajes para la gestión y recuperación de dichos datos.

Los detalles de esta capa ya han sido explicados en el apartado “Sistema gestor de base de datos”.

7.2. Aplicación

En el apartado anterior de “Definición de la interfaz gráfica”, ya ha sido definida y explicada, a grandes rasgos, la interfaz gráfica usada y las diferentes pantallas que se han creado, por lo que en este apartado se pasará a entrar en más detalle en algunos de los elementos que la forman.

7.2.1. Identificación de usuario

El acceso a la aplicación, tal como definen los requisitos, debe estar controlado y restringido por contraseña.

Esto implica que, al iniciar la aplicación, lo primero que se hace es pedir al usuario su nombre y contraseña con el cual quiere identificarse. Una vez introducidos estos datos y validados se mostrará la pantalla principal de la aplicación, para que el usuario pueda gestionar la información correctamente, según los permisos del usuario con el que se ha conectado.

La imagen siguiente muestra la pantalla de identificación mostrada al usuario para que introduzca su identificación.

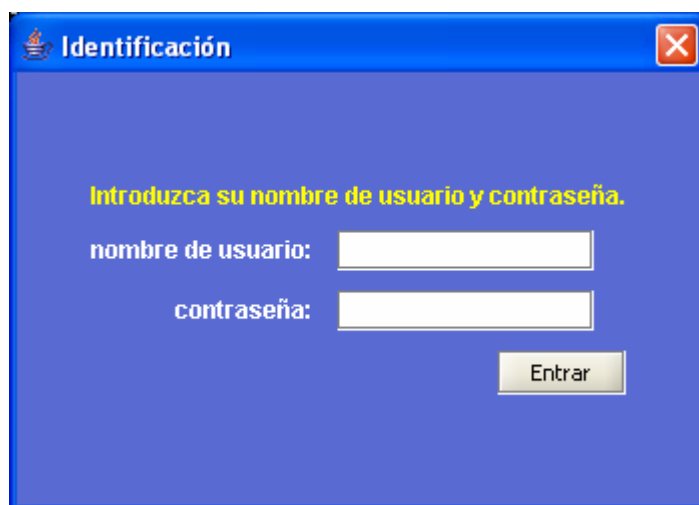


Imagen 7: Identificación usuario

Lo único que se debe realizar es la introducción de los datos y pulsar al botón “Entrar”. Una vez realizado esto, las acciones realizadas son:

1. Se cifra la contraseña introducida (las contraseñas son guardadas de forma cifrada en la base de datos)
2. Se consulta en la base de datos que existe una entrada con ese nombre de usuario y contraseña.
3. Si existe, se le da acceso al uso de la aplicación, según los permisos creados para el usuario.
4. Si no existe, se muestra, al usuario, el mensaje de error: “El usuario/contraseña es incorrecto”.

7.2.2. Barra de herramientas

La barra de herramientas es el elemento situado en la parte superior de la ventana principal, formada por una serie de botones con iconos, los cuales sirven para ejecutar las funciones más utilizadas del menú principal de cualquier aplicación (p.e.: abrir fichero, salvar fichero, salir,...)

Dado que en el caso de nuestra aplicación las acciones a realizar son reducidas y no existe ningún menú, la barra de herramientas se ha usado como menú, en el que aparecen todas las opciones disponibles a realizar, de forma que para el usuario sea muy fácil y accesible poder acceder a todas y cada una de ellas de forma rápida.

En nuestra aplicación disponemos de dos barras de herramientas diferentes que aparecerán dependiendo de las acciones que se puedan realizar. Las barras de herramientas de las que disponemos son:

1. Gestión de información: Es la barra de herramientas que nos permite gestionar los centros de la aplicación: nuevo centro, editar centro, borrar centro y recargar la lista de centros.
2. Gestión de usuarios y perfiles: barra de herramientas que permite crear, editar, borrar usuarios y perfiles.

Otra de las características de las barra de herramientas son los conocidos como *tooltips*, que son etiquetas de ayuda que aparecen al colocar el cursor del ratón encima de un icono. Cada una de estas etiquetas muestra una breve ayuda de la acción que realiza dicho icono al ser pulsado.

Gestión de la información

Es la barra de herramientas que aparece cuando se está en cualquier parte de la aplicación que no sea la gestión de usuarios. Con esta barra se podrá gestionar la lista de centros y es como muestra la imagen siguiente:



Imagen 8: Barra de herramientas de centros

La acción que realiza cada uno de los botones de la barra de herramientas, en el mismo orden, son:

- Crea un nuevo centro: Muestra el formulario de información de un centro (será explicado más adelante) con todos los campos vacíos.
- Edita un centro: Edita el centro seleccionado, mostrando el formulario de información de un centro, con toda la información del mismo editable para poder ser modificada.
- Borra un centro: Borra, de la base de datos, el centro seleccionado, pidiendo confirmación al usuario.
- Refresca la lista de centros: Refresca la lista de centros, de la base de datos.

Gestión de usuarios

Es la barra de herramientas que aparece cuando accedemos a la sección de Gestión de usuarios (pulsando dos veces con el ratón sobre la opción de “Usuarios”) y que permite gestionar la información de usuarios y perfiles de la aplicación. En la siguiente imagen se puede observar un ejemplo:



Imagen 9: Barra de herramientas de usuarios

Las acciones que se pueden realizar con ella son:

- Nuevo usuario: muestra el formulario de información de usuario, con todos los campos vacíos para poder crear un nuevo usuario.
- Edita un usuario: edita el usuario seleccionado, mostrando el mismo formulario pero relleno con toda su información.
- Borra un usuario: borra, de la base de datos, el usuario seleccionado, pidiendo confirmación al usuario.
- Nuevo perfil: hace lo mismo que el de “Nuevo usuario” pero con el formulario de información de perfil.

- Edita un perfil: igual que “Edita un usuario” pero con la información y el formulario de gestión de un perfil.
- Borra un perfil: borra, de la base de datos, el perfil seleccionado, pidiendo antes confirmación al usuario.

7.2.3. Área de navegación

El área de navegación está ubicada en la parte izquierda y está formada por un árbol con las diferentes entidades por las que se puede navegar y acceder y, en definitiva, trabajar con ellas.

Para poder acceder a la información de cada una de las entidades del árbol, bastará con hacer un “doble clic” de ratón sobre la misma, provocando que aparezca, en la parte derecha de la aplicación la información necesaria.

A continuación se muestra una imagen, en la que se puede observar dicha área:

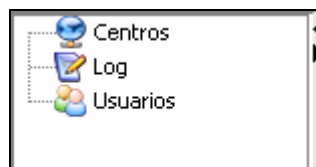


Imagen 10: Área de navegación

Centros

Bajo este nodo aparecerá la lista con el nombre de todos los centros que están dados de alta en la base de datos. Esta lista puede ser recargada pulsando el icono de recarga, de la barra de herramientas correspondiente.

Para poder editar toda la información de un centro de la lista, habrá que hacer alguna de las dos acciones siguientes:

1. Teniendo seleccionado un nodo/centro de la lista, pulsar el icono de editar, de la barra de herramientas.
2. Hacer “doble clic” sobre el centro que se desea editar.

Una vez realizada alguna de estas dos acciones se mostrará, en la parte derecha de la ventana, el formulario de datos de un centro con toda la información del centro seleccionado. Esto también provocará que la barra de herramientas sea cambiada a la de gestión de centros.

Log

Este es el nodo que permitirá el acceso al registro de incidencias de la aplicación.

Pulsando dos veces sobre el mismo mostrará, en la parte derecha de la ventana, la lista de incidencias de la aplicación y un formulario para poder editar cada una de las mismas.

De la misma manera que con los centros, la doble pulsación de ratón también provocará que la barra de herramientas sea cambiada a la de gestión de centros.

Usuarios

Este es el último de los nodos del árbol y es el que permite el acceso a la pantalla de gestión de usuarios y perfiles de la aplicación.

Al igual que con los demás nodos, este acceso se realizará con un “doble clic” de ratón, lo cual también provocará que la barra de herramientas se cambie a la de gestión de usuarios.

7.2.4. Área de trabajo

Está situada en la parte derecha de la ventana principal de la aplicación y es el área donde aparece toda la información necesaria para trabajar. Pueden aparecer 3 tipos de formularios de datos, según con qué tipo de información estemos trabajando en cada momento:

- Registro de incidencias: lista con las incidencias registradas en el sistema y un formulario para poder editar los datos de una incidencia en particular.
- Gestión de usuarios y perfiles: lista de todos los usuarios y perfiles dados de alta en la aplicación, así como un formulario para poder editar los datos de uno de ellos en particular.
- Gestión de centros: formulario para poder gestionar toda la información relacionada con un centro seleccionado.

Dado que es el apartado más extenso y que más información tiene se explicará su funcionamiento en un apartado diferente.

7.3. Área de trabajo de la aplicación

En la sección anterior ya se ha explicado cuál es el objetivo e información de este parte de la aplicación, así que aquí se pasará directamente a explicar cada uno de las pantallas en las que se divide.

7.3.1. Registro de incidencias

Como ya se ha explicado en otras ocasiones, esta es la pantalla en la que se podrá visualizar la lista de incidencias registradas en el sistema. Por incidencia se entiende algunas de las acciones que puede realizar el usuario con la aplicación: dar de alta un centro, modificar sus datos, dar de alta un servidor de un centro, etc.

Como se puede observar en la imagen de más abajo, esta pantalla está dividida en dos secciones: la lista de incidencias y un formulario de datos.

En la lista de incidencias es donde se podrá observar una tabla con la totalidad de incidencias que se han registrado en el sistema. Cada línea de la tabla equivale a una incidencia registrada y proporciona la siguiente información:

1. Fecha y hora en la que se produjo la incidencia.
2. Acción que se realizó.
3. Breve descripción sobre la incidencia. Esta información es editable permitiendo al usuario poder ampliar la descripción sobre la incidencia.

Haciendo una doble pulsación de ratón sobre una incidencia (sobre una línea de la tabla) se mostrará, en el formulario inferior toda la información de la incidencia. De esta información sólo estará editable y modificable la descripción, dando así la posibilidad de modificarla y guardarla.

Fecha	Accion	Descripcion
31/03/2006 08:34:10	nuevo centro	insertado centro: centro
31/03/2006 08:35:30	centro eliminado	eliminado centro: centro
31/03/2006 08:44:02	nuevo centro	insertado centro: centro (prueba)

Fecha:

Acción:

Descripción:

Aceptar

Imagen 11: Registro de incidencias

7.3.2. Gestión de usuarios y perfiles

Pantalla desde la que se podrá llevar toda la gestión de los usuarios y perfiles de la aplicación: dar de alta, borrar y modificar sus datos. A esta pantalla, a la que sólo se podrá llegar pulsando sobre la opción de “Usuarios” del árbol del área de navegación, sólo tienen acceso aquellos usuarios con permisos de administrador.

La pantalla de gestión de usuarios y perfiles está dividida en 3 secciones:

1. Listado de todos los usuarios y perfiles creados.

2. Formulario para la gestión de datos de un usuario.
3. Formulario para la gestión de datos de un perfil.

Nada más acceder a esta pantalla nos aparecerán dos listas con los usuarios y perfiles de la aplicación. Estas dos listas son como las que se pueden observar en la siguiente imagen:

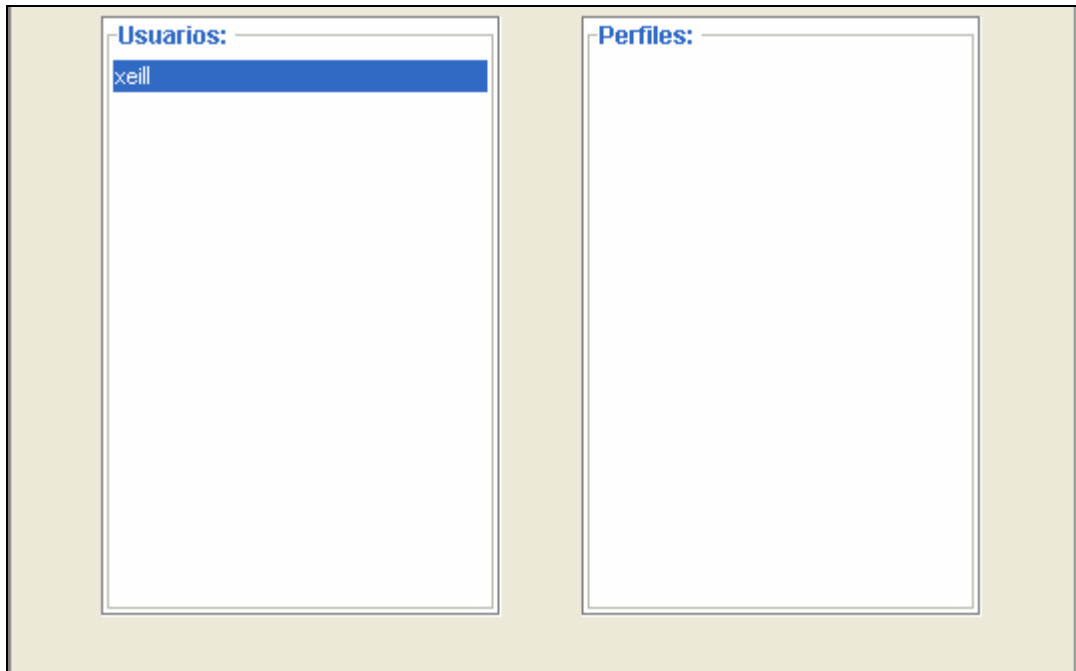


Imagen 12: Listado usuarios y perfiles

En la lista de la izquierda, como ya indica el título de la misma, aparece un listado con todos los usuarios creados y en la de la derecha de todos los perfiles.

Seleccionando alguno de los elementos de alguna de las dos listas, haciendo doble clic sobre ellos, mostrará el formulario de datos correspondiente con los datos del usuario o perfil seleccionado.

En el caso de seleccionar la opción de editar los datos de un usuario o de dar de alta uno, el formulario que aparece es el de la imagen de más abajo. La información que se debe especificar de un usuario es:

- Nombre de usuario.
- Contraseña de acceso.
- Repetir la contraseña para comprobar que se ha introducido correctamente.
- Perfil de usuario que se quiere le asociar, de entre la lista de perfiles creados. En el caso de seleccionar un perfil el usuario heredará sus

permisos y al conectarse no se hará caso de los permisos asociados al usuario. En el caso de no asociar ningún perfil (opción "Ninguno") los permisos que se usarán al identificarse serán los asociados al usuario.

- Si el usuario es administrador o no. Un usuario administrador es el único que tiene acceso al registro de incidencias y a la gestión de usuarios.
- Los permisos del usuario. Hay tres tipos diferentes de permisos, asociados a cuatro tipos de información. Los permisos existentes son los de: crear, modificar y eliminar. Los tipos de información son: Centro, Puntos de acceso, Servidores y Nodo. Una combinación de estos dos datos define el tipo de gestión que puede hacer el usuario con la información.

Datos de usuario:

Nombre de usuario:

Contraseña:

Verificar contraseña:

Perfil de usuario: Ninguno ▼

Administrador:

Permisos:

	Centro	Puntos de acceso	Servidores	Nodo
Crear	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modificar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eliminar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Imagen 13: Gestión usuarios

La contraseña especificada para el usuario, en la base de datos, se guarda de forma cifrada, por cuestiones de seguridad.

En el caso de seleccionar la opción de editar los datos de un perfil o de dar de alta uno, el formulario que aparece es el de la imagen de más abajo. La información que se debe especificar de un perfil es:

- Nombre del perfil.
- Si el perfil es administrador o no. Su uso es el mismo que para el usuario.
- Los permisos del perfil. Su uso es el mismo que para el usuario.

	Centro	Puntos de acceso	Servidores	Nodo
Crear	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modificar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eliminar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Imagen 14: Gestión perfiles

7.3.3. Gestión de un centro

Esta pantalla es la más importante y completa de todas y es la que permite gestionar toda la información de los diferentes centros que forman la red XEiLL. La pantalla es como se muestra en la imagen de más abajo.

Como se puede observar, esta pantalla está dividida en 5 pestañas diferentes que permiten realizar la gestión y acciones sobre diferentes grupos de información de un centro:

- Información: Permite gestionar la información general de un centro.
- Puntos de acceso: desde esta pestaña se puede llevar la gestión de toda la información de los diferentes puntos de acceso del centro.
- Servidores: pestaña desde la que se puede gestionar la información de los diferentes servidores que hay instalados en el centro.
- Nodo: desde aquí se puede llevar la gestión de la información del nodo WiFi del centro.
- Monitorización: esta pestaña contiene una pequeña aplicación que permite realizar una sencilla monitorización del centro en cuestión. Mediante esta aplicación se podrá comprobar si el nodo del centro está funcionando y los diferentes tiempos de respuesta que tienen sus servicios.

A continuación se pasará a explicar más en detalle cada una de las diferentes pestañas de esta pantalla.

El botón de Aceptar, situado en la parte inferior del formulario es el que permite indicar a la aplicación que se quieren guardar los datos en la base de datos (ya sean modificados o de creación de un nuevo centro). Hay que resaltar que hasta que no se pulse este botón no se guardará ningún dato en la base de datos. Cualquier modificación que se haga y se guarde de cualquiera de los formularios, se guardará en memoria para ser trasladados a la base de datos en el momento que se pulse este botón.

Información \ Puntos de acceso \ Servidores \ Nodo \ Monitorización \

Nombre del centro: colegio1

Dirección: 13, rue del percebe

Persona de contacto:

Teléfono:

correo electrónico:

DNS:

IP interna del router: 123 . 192 . 7 . 8

IP externa del router: 195 . 235 . 59 . 147

Redirecciones de puertos

Puerto de origen: 1

Puerto destino: 1

IP destino: 0 . 0 . 0 . 0

tipo puerto: TCP ▼

comentario:

Insertar

puerto de origen	puerto destino	IP destino
25	25	100.101.102.103
31	31	130.130.130.130

Aceptar

Imagen 15: Información de un centro

Información general

Esta es la pestaña que permite gestionar la información general de un centro, y es la que se puede observar en la imagen anterior.

La información general de un centro la forman los siguientes datos:

- Nombre del centro.
- Dirección del centro.
- Contacto del centro: persona, teléfono y correo electrónico.
- DNS del centro.
- IP interna del router del centro: es la IP asignada a ese router dentro de la intranet del centro.
- IP externa del router del centro: es la IP asignada a ese router en internet y es la que se usa para conectarse a los servidores del centro y usar sus servicios.
- Redirecciones de puertos: son redirecciones entre puertos de la interfaz externa del router (la que es accesible desde internet) a puertos de otras máquinas de la intranet.

La información referente a las redirecciones de puertos necesita de otro formulario, ya que gestiona más información. Estos datos son:

- Puerto de origen: puerto del router, origen de la redirección, accesible desde internet.
- Puerto de destino: puerto al que está redirigido el acceso.
- IP de destino: servidor donde se encuentra habilitado el puerto de destino.
- Tipo de puerto: protocolo de conexión habilitado en ese puerto: TCP o UDP.
- Comentario: breve comentario sobre el uso de la redirección.

También aparece una tabla con una lista de todas las redirecciones que hay dadas de alta para este centro.

Para dar de alta una redirección lo único que hay que hacer, cuando el formulario está vacío, es introducir los datos necesarios de la redirección y pulsar el botón "Insertar".

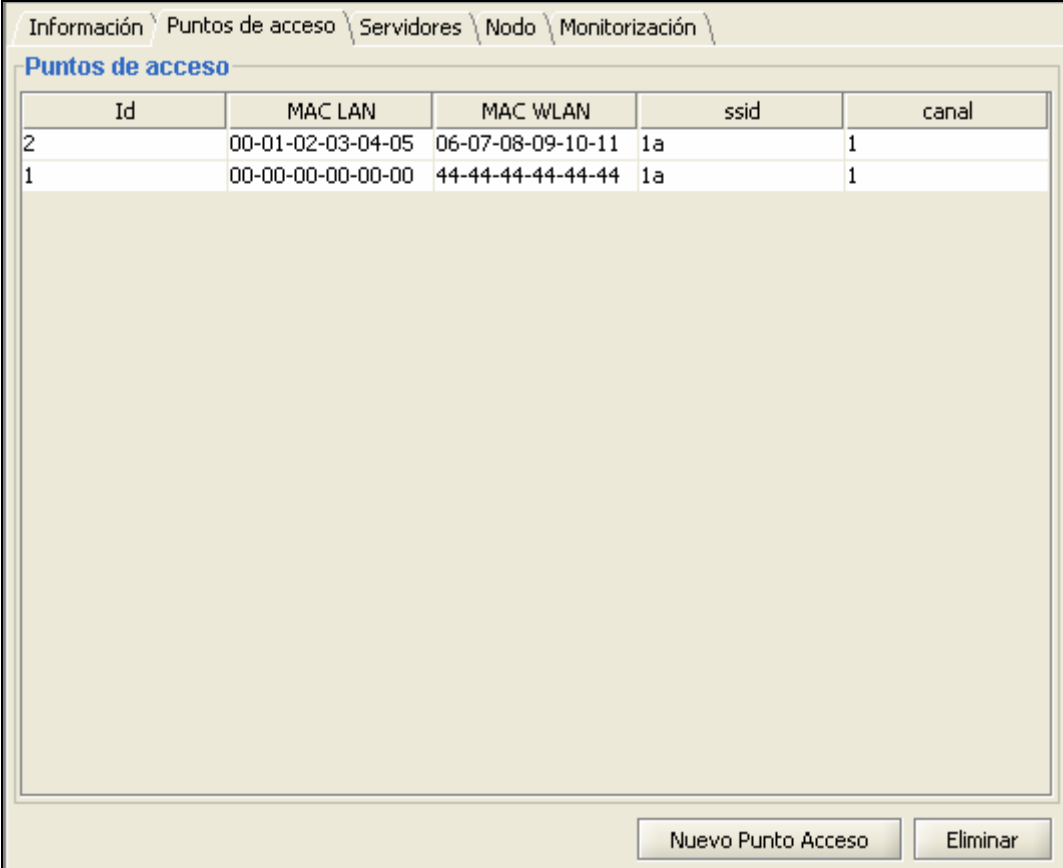
En el caso de querer modificar una redirección ya creada lo que habrá que hacer es pulsar dos veces sobre la fila de la tabla correspondiente a la redirección a editar. En ese momento aparecerán dos botones: "Modificar" y

“Eliminar” que sirven, respectivamente, para actualizar la información, con los nuevos datos introducidos, o eliminarla definitivamente de ese centro.

Puntos de acceso

Desde esta pestaña será desde la que se podrá gestionar la información de los diferentes puntos de acceso que haya instalados en el centro correspondiente.

La primera pantalla que aparecerá, será la lista completa de todos los puntos de acceso que ya han sido creados para este centro. Dicha pantalla será como la de la siguiente imagen:



Id	MAC LAN	MAC WLAN	ssid	canal
2	00-01-02-03-04-05	06-07-08-09-10-11	1a	1
1	00-00-00-00-00-00	44-44-44-44-44-44	1a	1

Imagen 16: Listado de puntos de acceso

Como se puede observar, aparece la lista de puntos de acceso y dos botones. Desde esta pantalla se podrán realizar 3 acciones diferentes:

1. Dar de alta un nuevo punto de acceso. Esta acción se realizará pulsando el botón “Nuevo Punto de Acceso” que cambiará la pantalla mostrando el formulario de entrada de datos de un punto de acceso.
2. Editar los datos de un punto de acceso. Se realizará pulsando dos veces sobre la fila correspondiente al punto de acceso que se quiere editar. El formulario mostrado será el mismo que el de crear un nuevo punto de acceso, pero con la información a editar.

3. Eliminar un punto de acceso. Finalmente está la opción de eliminar un punto de acceso que se realizará con el botón de “Eliminar”. Al pulsar este botón se pedirá confirmación al usuario y, en caso afirmativo, eliminará el punto de acceso que esté seleccionado en ese momento, en la tabla.

En la imagen siguiente se muestra la pantalla usada tanto para la creación como la edición de un punto de acceso:

Información > Puntos de acceso > Servidores > Nodo > Monitorización

Ubicación de instalación:

Marca/Modelo:

Estándar: a ▼

Canal:

ssid:

MAC en la LAN:

MAC en la WLAN:

IP de administración:

Nombre de usuario de administración:

Contraseña de administración:

Enlaces WDS

Id	MAC LAN	MAC WLAN
----	---------	----------

Imagen 17: Punto de acceso

La información que se puede gestionar con este formulario es:

- Ubicación del punto de acceso: breve descripción de la ubicación física, en el centro, donde se encuentra el punto de acceso correspondiente.
- Marca/Modelo: marca y modelo del dispositivo usado como punto de acceso.
- Estándar WiFi que usa el punto de acceso.

- SSID del punto de acceso.
- Dirección MAC del dispositivo dentro de la red LAN en la que se encuentra.
- Dirección MAC del dispositivo dentro de la red WLAN en la que se encuentra.
- IP de administración: dirección IP donde se encuentra alojada la aplicación de gestión del dispositivo.
- Nombre de usuario de administración necesario para poder acceder a la aplicación de gestión.
- Contraseña del usuario de administración.
- Lista de enlaces WDS asociados a dicho punto de acceso.

La lista de enlaces del punto de acceso viene mostrada en la tabla situada en la parte inferior del formulario. Para poder llevar una gestión de los enlaces se han creado los botones: “Añadir WDS” y “Eliminar WDS”.

Como ya se ha definido anteriormente, un WDS es un enlace entre dos puntos de acceso que tienen el mismo canal y ssid. Esto significa que si, en la aplicación, queremos dar de alta un nuevo enlace WDS en el punto de acceso que estemos gestionando en ese momento, se deberá pulsar el botón “Añadir WDS”. Este botón lo que hará es mostrar una ventana de diálogo (la mostrada en la imagen de más abajo) que mostrará todos aquellos puntos de acceso, de todos los centros creados en la aplicación, que compartan canal y ssid con el actual. Cada vez que se realice esta operación en la lista no aparecen los ya seleccionados como enlaces WDS.

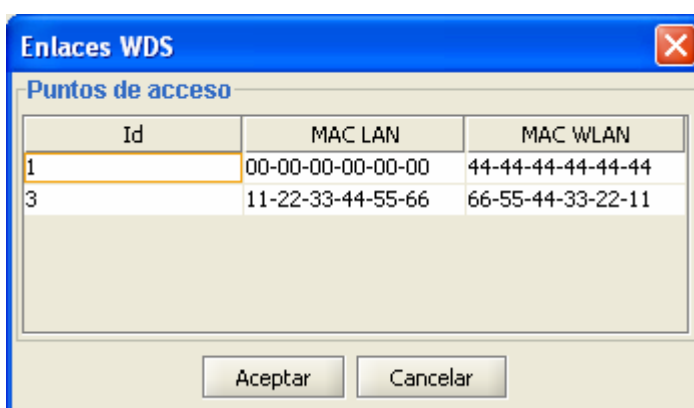


Imagen 18: Enlaces WDS

Una vez seleccionados de la lista, los puntos de acceso a añadir en la lista de enlaces WDS y pulsado el botón de aceptar, la ventana de cerrará y se podrá observar que en la lista de enlaces WDS ya aparecen los nuevos seleccionados.

Finalmente, con el botón de “Eliminar WDS” se eliminarán los enlaces que estén seleccionados en la tabla.

Por último, los botones situados en la parte inferior de la pantalla, son los que permiten indicar qué se desea hacer con la información introducida en el formulario:

- Guardar punto de acceso: guardará todos los datos introducidos y volverá a la pantalla inicial, con la lista de puntos de acceso.
- Cancelar: volverá a la pantalla anterior de la lista de puntos de acceso, sin haber guardado ningún cambio realizado.

Servidores

De manera similar a la pestaña de Puntos de acceso, en la pestaña de servidores, la primera pantalla que se verá será la lista completa de servidores que han sido dados de alta para este centro. Dicha pantalla será como la de la siguiente imagen:

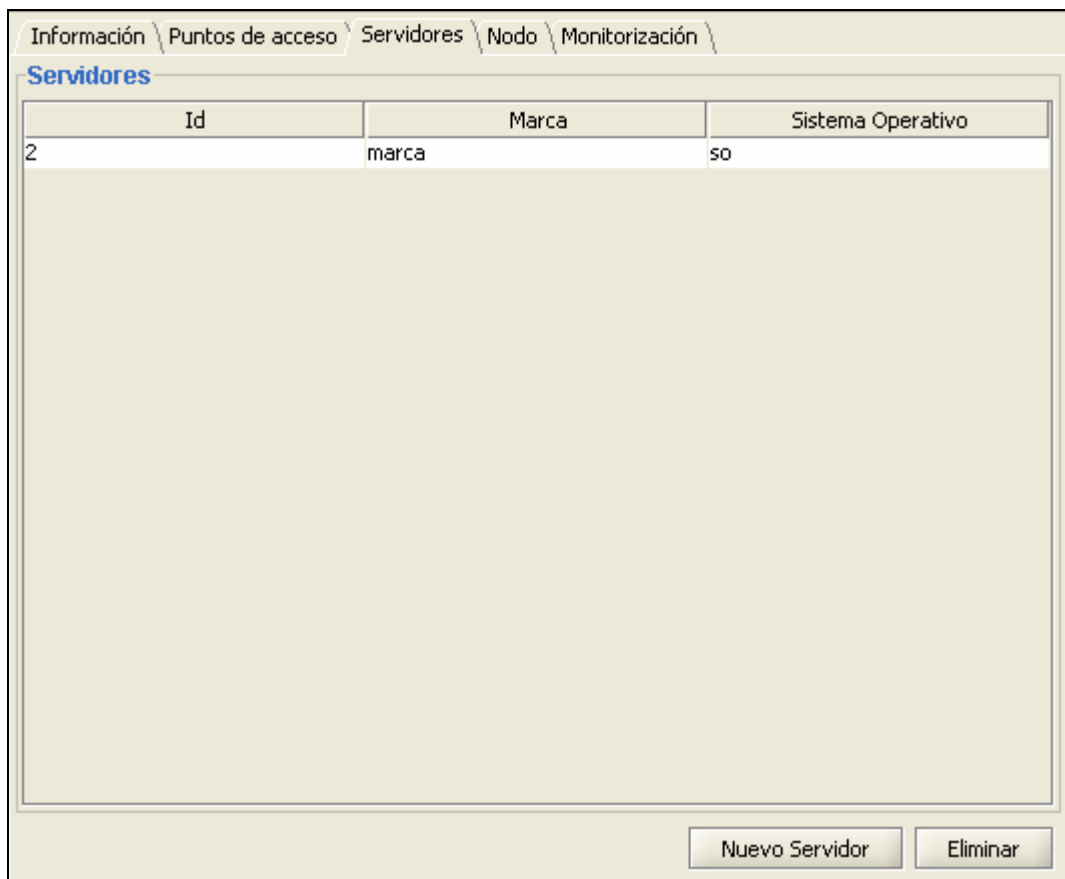


Imagen 19: Listado servidores

El funcionamiento de dicha pantalla también es semejante que con los puntos de acceso, pudiendo realizar las acciones de: crear nuevo servidor, editar los datos de un servidor y eliminar servidor.

En el caso de editar los datos de un servidor ya creado (haciendo doble clic sobre la fila correspondiente) o querer crear un nuevo servidor (pulsar el botón de “Nuevo Servidor”) aparecerá la pantalla de la siguiente imagen, desde la que se podrá gestionar toda la información:

Información \ Puntos de acceso \ Servidores \ Nodo \ Monitorización \

Marca/Modelo:

Versión de SO:

Nombre de usuario sin privilegios:

Password de usuario sin privilegios:

Password de administrador:

Servicios

nombre	puerto	protocolo
uoc	80	TCP
servicio	7	UDP

Adaptadores de red

ID	IP	dirección MAC	tipo
tapx	10.11.12.13	mac	fisico

VPNs

núm. puerto nodo A	tipo puerto nodo A	núm. puerto nodo B	tipo puerto nodo B
10	TCP	20	UDP

Imagen 20: Servidor

La información que puede ser gestionada desde este formulario es:

- Marca y modelo del servidor.
- Versión del sistema operativo instalado.
- Nombre del usuario sin privilegios que hay creado en el sistema del servidor.
- Contraseña del usuario sin privilegios.
- Contraseña del administrador del servidor.

- Lista de los servicios que están ejecutándose en el servidor.
- Adaptadores de red instalados.
- VPNs con otros servidores.

Los servicios instalados en un servidor, al tener cada uno de ellos información adicional, son tratados en un formulario aparte, mostrado en una ventana de diálogo. En la pantalla principal de servidores se muestra una lista de todos los servicios dados de alta en el servidor que se está gestionando. Desde esta pantalla principal, podemos hacer 2 acciones con los servicios:

1. Dar de alta un servicio nuevo, mediante el botón “Nuevo servicio”
2. Editar los datos de un servicio, pulsando dos veces, en la tabla, sobre la fila del servicio a editar.

En cualquiera de los dos casos, la ventana de diálogo que se muestra para la edición de los datos es la que se puede observar en la siguiente imagen:

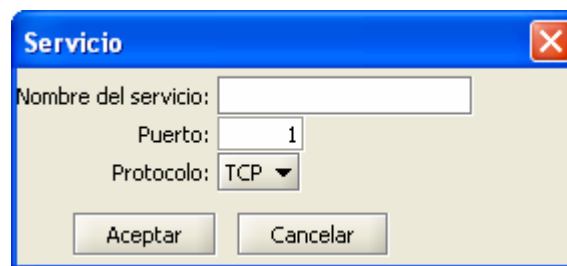
A screenshot of a Windows-style dialog box titled "Servicio". The dialog box has a blue title bar with a close button (X) in the top right corner. The main area is light gray and contains three input fields: "Nombre del servicio:" with an empty text box, "Puerto:" with a text box containing the number "1", and "Protocolo:" with a dropdown menu showing "TCP". At the bottom of the dialog box, there are two buttons: "Aceptar" and "Cancelar".

Imagen 21: Servicio

Desde este formulario se puede proporcionar los siguientes datos de un servicio:

- Nombre del servicio.
- Puerto del servidor en el que está escuchando este servicio.
- Protocolo sobre el que trabaja (TCP o UDP)

En el caso de estar editando un servicio ya existente, a parte del botón de “Aceptar”, también aparecerá el botón de “Eliminar” que permitirá borrar el servicio en cuestión.

En el caso de los adaptadores de red, es idéntico al de los servicios. La ventana de diálogo que se mostraría en ese caso es la de la siguiente imagen:



Imagen 22: Adaptador de red

Los datos que se facilitan son:

- Identificador del adaptador de red: tapx o ethx
- Dirección IP del adaptador de red.
- Dirección MAC del mismo.
- Tipo de adaptador de red usado: físico o virtual.

Finalmente, con la gestión de las VPNs del servidor pasa exactamente igual que con los dos casos anteriores, siendo, en este caso, la ventana de diálogo como la que se muestra en la imagen siguiente:

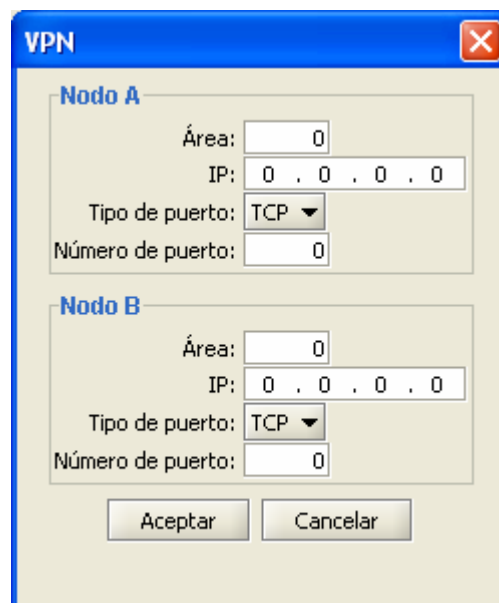


Imagen 23: VPN

En este caso habrá que facilitar los datos referentes a los dos extremos de la red VPN, siendo estos datos, para cada extremo (nodo), los siguientes:

- Área en la que se encuentra el nodo.
- IP del nodo.

- Tipo de puerto sobre el que está montado el extremo de la VPN: TCP o UDP.
- Puerto en el que está escuchando el nodo.

Nodo

Esta es la última pestaña relacionada con la gestión de la información de un centro. La pantalla que aparece al pulsar sobre esta pestaña es la de la imagen que hay más abajo, y la información que permite controlar es:

- Subdominio de la XEiLL del nodo WiFi del centro en cuestión.
- Listado de los rangos IP que acepta el nodo.
- Listado con los DNS que hay registrados en este nodo.
- Listado de encaminamientos creados en el nodo.

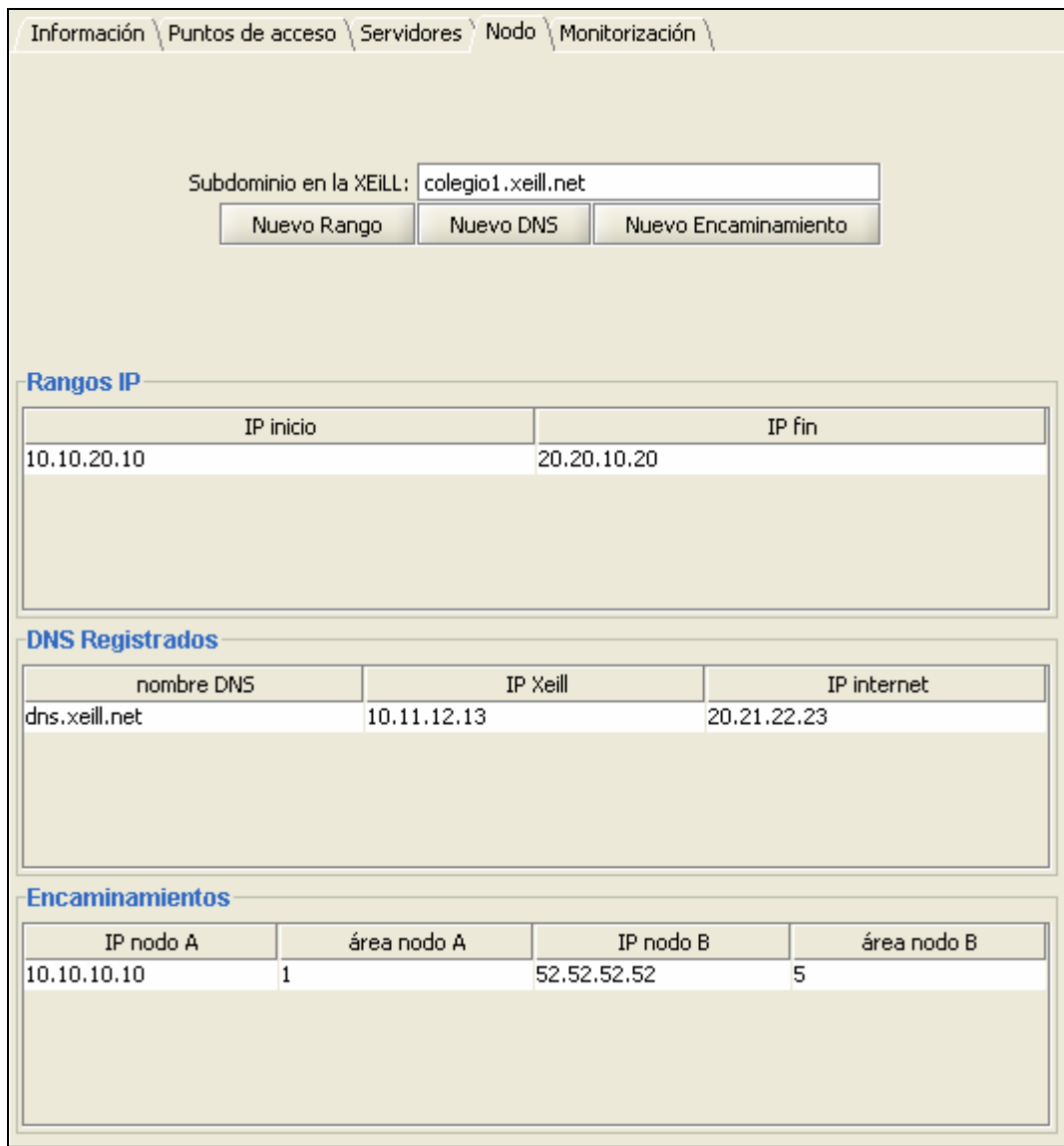


Imagen 24: Nodo

En el caso de la gestión de los rangos, DNS y encaminamientos funciona de forma semejante a como se hace con los servicios, adaptadores de red y VPNs en la pestaña del Servidores. En este caso, como se puede observar en la imagen anterior, tenemos las 3 listas correspondientes: Rangos IP, DNS Registrados y Encaminamientos. Para editar los datos de una entidad sólo hace falta pulsar dos veces sobre la fila en cuestión de la tabla correspondiente. En el caso de querer crear una nueva entidad sólo hará falta pulsar sobre uno de los 3 botones: “Nuevo Rango”, “Nuevo DNS” o “Nuevo encaminamiento”, según el caso.

En cualquiera de los dos casos (editar o crear entidad nueva) se mostrarán sendas ventanas de diálogo que están explicadas a continuación.

La siguiente imagen es la correspondiente a la ventana de diálogo de edición de los datos de un Rango IP, de la lista de los mismos:

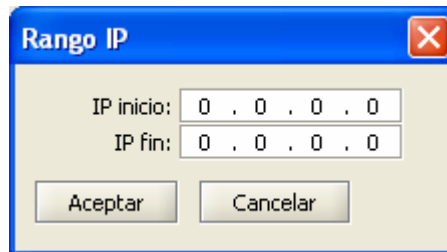


Imagen 25: Rango IP

Para dar de alta, en la lista, un rango nuevo, sólo hace falta proporcionar dos datos, que son la IP de inicio del rango y su IP de fin.

Con el registro de DNS ocurre algo similar, siendo la siguiente imagen la correspondiente a la ventana de diálogo de gestión de datos de un DNS del nodo:

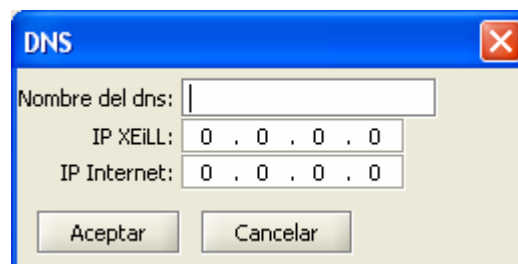


Imagen 26: DNS

La información a facilitar es:

- Nombre del DNS.
- Dirección IP dentro de la red XEiLL.
- Dirección IP pública en Internet.

Finalmente está la ventana de diálogo de un Encaminamiento, que es la que se muestra en la siguiente imagen:

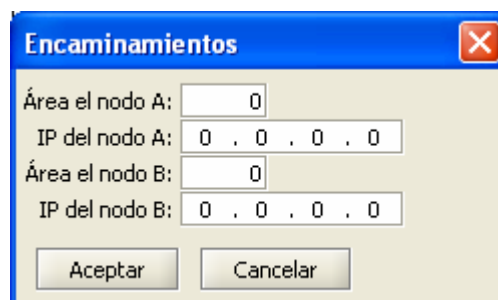


Imagen 27: Encaminamiento

La información a proporcionar es la correspondiente a cada uno de los dos nodos del encaminamiento y no es más que el área del nodo y la IP del mismo.

Monitorización

Finalmente, esta es la última pestaña que forma el conjunto de herramientas para poder llevar a cabo la gestión de un centro de la red XEiLL.

En esta pestaña podremos encontrar una sencilla herramienta de monitorización que permite controlar el correcto funcionamiento de la conexión de un centro. Comprobando la conectividad de algunos de los diferentes servicios que ofrezca un centro, podremos comprobar el correcto funcionamiento del mismo.

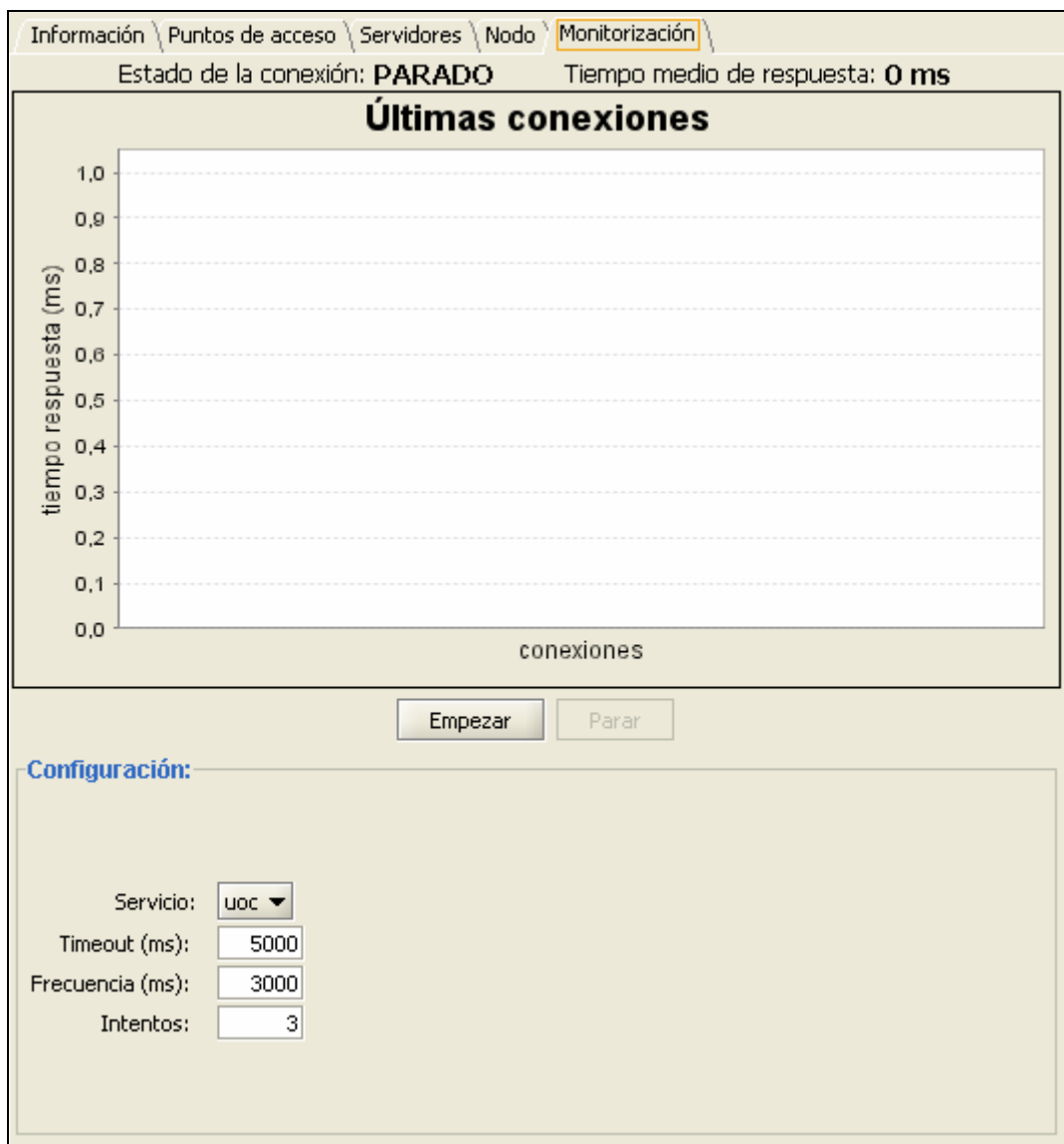


Imagen 28: Monitorización

En la anterior imagen se puede observar la pantalla mostrada en esta pestaña, pudiendo diferenciar las diferentes secciones en las que se divide:

- En la parte superior de la misma hay una línea con la información sobre el funcionamiento del servicio, informando tanto del estado de la conexión, como de su tiempo medio de respuesta.
- A continuación se muestra una gráfica con el historial de las últimas 5 conexiones y el tiempo de respuesta, del servicio, en cada una de ellas.
- Finalmente, encontramos un bloque de configuración, el cual permite parametrizar la monitorización de un servicio. Estos datos son:
 - Una lista de todos los servicios TCP que tiene el centro (entre la totalidad de los servidores instalados). El servicio que haya en cada momento seleccionado es sobre el cual se realizará la monitorización del estado de conexión.
 - El timeout o espera máxima de tiempo de respuesta, expresado en milisegundos. Si, al intentar hacer la conexión al servicio, pasan la cantidad de milisegundos introducidos en este campo, se dará por errónea, suponiendo que el servicio está parado.
 - La frecuencia, expresada en milisegundos, con la que se hará cada una de las conexiones. Es el tiempo que se esperará entre que ha finalizado una conexión con el servicio y se realiza la siguiente.
 - Número de intentos de conexiones fallidas seguidas, a partir de las cuales se da por supuesto que el servicio no responde. Un número de intentos fallidos inferior a esta cantidad indicará que, aunque el servicio no está respondiendo, todavía no se dará por supuesto que sea porque el servicio no está en marcha.

Entre la gráfica del historial de conexiones y la configuración, encontramos dos botones que son los que se usarán para poner en marcha este servicio o pararlo. Son los botones de “Empezar” y “Parar”. El primero sólo estará habilitado cuando no se ha puesto en marcha la monitorización de un servicio y, por lo tanto, estando deshabilitado cuando se está llevando éste a cabo. El segundo botón es a la inversa, permanecerá deshabilitado cuando no esté en marcha la monitorización, habilitándose en el momento que comience.

Una vez que se pulsa el botón de “Empezar”, y con un intervalo igual al número introducido en el campo de configuración “Frecuencia”, se irán realizando conexiones al servicio.

A medida que se van realizando estas conexiones la parte superior de la pantalla va actualizando los valores de “Estado de la conexión” y “Tiempo medio de respuesta”. El segundo dato, a pesar de que en la gráfica sólo aparecen las últimas 5 conexiones, informa del tiempo medio de respuesta de todas las conexiones realizadas desde el momento en el que se comenzó la monitorización del servicio, inicializándose a 0 en el momento que se para la monitorización en curso y se comienza una nueva.

El dato de “Estado de la conexión” se irá actualizando dependiendo el número de conexiones erróneas que se realicen. Cuando nos encontramos en el caso de que se han realizado un número de conexiones erróneas igual o superior al indicado en “Intentos”, la etiqueta de este dato cambiará a “NO RESPONDE”, tal y como se muestra en la siguiente imagen:

Estado de la conexión: **NO RESPONDE**

Imagen 29: Monitorización "NO RESPONDE"

En el momento que se realiza con éxito una conexión al servicio, estemos en el estado que estemos, esta etiqueta cambiará a “RESPONDE”, tal y como se muestra en la siguiente imagen:

Estado de la conexión: **RESPONDE**

Imagen 30: Monitorización "RESPONDE"

A medida que se vayan realizando todas conexiones, la gráfica se irá actualizando con el estado de la última 5 conexiones que se han realizado. En la siguiente imagen se puede observar un ejemplo de una ejecución en la que las últimas 5 conexiones de ese momento habían tenido unos tiempos de respuesta de 125, 110, 94, 109 y 109 milisegundo cada una:

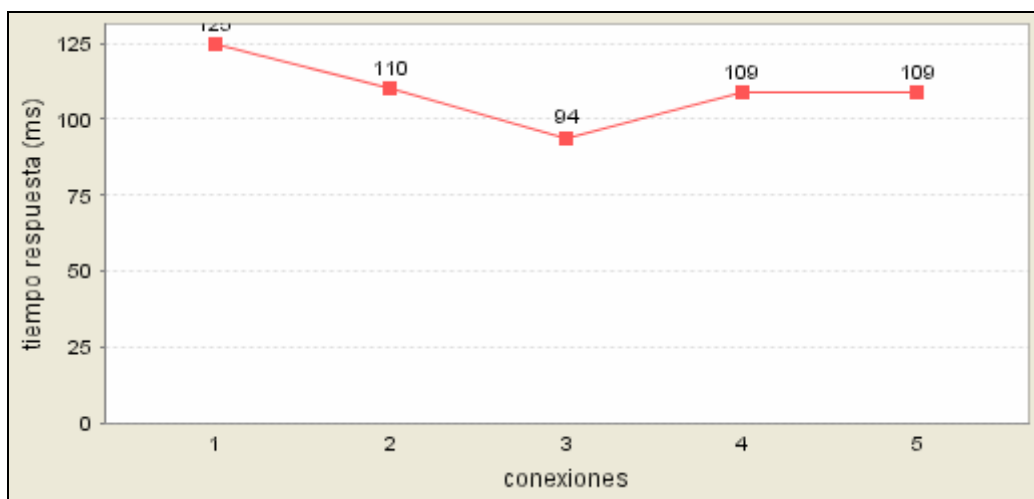


Imagen 31: Gráfica monitorización

8. CONCLUSIONES

8.1. Objetivos

En el capítulo de introducción y más concretamente en el apartado 4.2 del mismo, se explicaron y marcaron cuáles eran los objetivos que justificaban la realización de este proyecto y que se marcaban como aspectos básicos a cumplir a la finalización del proyecto.

En un capítulo posterior, el capítulo del Análisis y Diseño, y más concretamente los apartados 6.1.1 y 6.1.2, también definíamos cuáles eran los requisitos funcionales y no funcionales que debíamos tener en cuenta a la hora de comenzar a diseñar e implementar la aplicación.

Ahora, una vez finalizado el proyecto, es el momento de revisar si todos aquellos objetivos que nos habíamos marcado los hemos cumplido y si todos los requisitos que habíamos definido los hemos respetado.

Respecto a los objetivos se comentó que tenía que tratarse de una herramienta con una interfaz gráfica fácil e intuitiva, que permitiese almacenar en un dispositivo centralizado todos los datos de los centros que forman la XEiLL, pudiendo gestionar toda la información correspondiente a cada centro. Además también debía llevar un registro de todas las incidencias de la aplicación e incluir una herramienta para llevar una monitorización básica del estado de la conexión de un centro.

Como se ha podido observar durante el desarrollo del presente documento, los objetivos marcados han sido cumplidos en su totalidad, resultando en:

1. Tenemos una herramienta gráfica, implementada en Java, con pocas órdenes y acciones a realizar estando representadas todas mediante botones o acciones de ratón.
2. Todos los datos a gestionar están almacenados en una base de datos centralizada de forma que siempre se puedan recuperar y cualquier usuario pueda acceder a ellos.
3. Se realiza un registro de las incidencias más importantes que ocurren en la aplicación, dando la opción, a un usuario con perfil de administrador, de poder editarlas y modificar la descripción de cada una de ellas.
4. Cada centro tiene la opción de comprobar el estado de sus servicios, realizando de esta forma una monitorización básica de cada uno de ellos.

Respecto a los requisitos del proyecto, también podemos decir, al igual que con los objetivos que han sido cumplidos en su totalidad, teniendo de esta forma:

1. Toda la tecnología y herramientas que se han usado para el desarrollo del proyecto se ha tratado de *software libre*.

2. El acceso a la aplicación está protegido mediante nombre de usuario y contraseña, pidiendo la identificación del mismo al inicio.

8.2. Líneas futuras de trabajo

Como cualquier desarrollo de un proyecto, a la finalización del mismo e independientemente del grado de consecución de los objetivos marcados en el inicio, siempre nos encontramos con posibles ampliaciones o líneas futuras de trabajo que permitan mejorar la aplicación desarrollada. A continuación se detalla una lista de algunas de estas posibles ampliaciones que se tienen contempladas:

- **Mejorar la herramienta de monitorización.** Actualmente la herramienta de monitorización implementada se trata de una herramienta básica que lo único que hace es abrir una conexión con un servicio y contar el tiempo que tarda éste en responder. Una posible mejora sería la de estudiar posibles soluciones más complejas que permitan llevar una monitorización más detallada y completa que la implementada. Un ejemplo sería el de instalar en el servidor central la herramienta de monitorización de redes WiFi *JFFNMS* y que la aplicación se comunicase con esta herramienta y usase sus datos para generar estadísticas y gráficas de resultados.
- **Ubicación geográfica de los centros.** Otra de las posibles mejoras de la aplicación sería poder ubicar geográficamente los centros en un mapa. Mediante una herramienta de mapas (como por ejemplo *GoogleMaps*) y la dirección postal de un centro, se trataría de mostrar una imagen de un mapa situando la posición geográfica de ese centro.
- **Aplicación web.** Existen diversas herramientas libres o *frameworks* que permiten “traducir” una aplicación gráfica escrita en Java en un conjunto de páginas web de forma que lo que tengamos finalmente sea una aplicación web (mejor rendimiento de carga) pero sin perder toda la potencia de la aplicación gráfica. Un ejemplo de este tipo de aplicaciones es *WebOnSwing*. La solución pasaría por usar esta herramienta (u otra similar) que permitiese poder usar la aplicación como si se tratase de una aplicación web, sin necesidad de tener un plug-in de Java instalado.

9. BIBLIOGRAFÍA

9.1. Direcciones Web

(1) <http://www.hibernate.org/>

Página del sistema de persistencia usado en la aplicación. Desde esta página es posible desde buscar información o documentación, hasta entrar en una comunidad de desarrolladores, para poder entrar en contacto con otras personas que usan la herramienta, pasando, evidentemente, por la posibilidad de descargar las librerías necesarias para poder trabajar con Hibernate.

(2) <http://www.mysql.com/>

Esta es la página del SGBD usado en la aplicación desarrollada. De forma similar que con Hibernate, desde esta página web se pueden hacer cosas como: descargar el software necesario, buscar y consultar documentación o acceder a diferentes cursos y certificaciones que se realizan sobre MySQL.

(3) <http://www.jgoodies.com/>

JGoodies son un conjunto de librerías que facilitan el desarrollo de interfaces gráficas. Entre este conjunto de librerías están las llamadas *Looks*, las cuales son para cambiar el aspecto de la interfaz gráfica de la aplicación.

(4) <http://www.jfree.org/jfreechart/>

JFreeChart es una librería libre que provee una API para la generación de todo tipo de gráficas. Estas librerías han sido usadas para la generación de la gráfica de la aplicación de monitorización.

(5) <http://webonswing.sourceforge.net>

Como se ha comentado durante el desarrollo de la memoria, WebOnSwing es una de las herramientas existentes que permiten la creación de una aplicación web a partir de una serie de clases Java. Transforma una aplicación gráfica de escritorio en una aplicación web.

(6) <http://www.jffnms.org/>

Página web de la aplicación JFFNMS, aplicación comentada como posible alternativa para desarrollar una monitorización de la red más completa que la que se ha desarrollado en este proyecto.

(7) <http://www.icon-king.com/>

Página web de los iconos usados en la aplicación (los de las barras de herramientas y de los nodos del árbol). En concreto los que se han usado son los *Nuvola* (<http://icon-king.com/?p=15>)