

# Arquitectura para Laboratorios Remotos Físicos y Virtuales

Grosclaude, Eduardo - Bertogna, Leandro M. - Del Castillo, Rodolfo - López Luro, Francisco  
- Zanellato, Claudio

Universidad Nacional del Comahue, Departamento de Informática y Estadística

Neuquén, Argentina

{oso, mlbertog, rolo, flopez, czanella}@uncoma.edu.ar

## Resumen

*Los entornos de capacitación y operación a distancia presentan problemáticas complejas debido a la dificultad del acceso y la diversidad en los tipos de recursos involucrados. En este trabajo se describe un marco de trabajo para la definición y utilización remota de laboratorios físicos y virtuales, así como también detalles de implementación en aspectos como la interacción cliente-recurso, seguridad o disponibilidad. Este marco comprende desde la creación de plantillas de trabajo de laboratorios por parte del docente, hasta la adaptación de clientes a sus correspondientes entornos para poder realizar sus labores. Hasta el momento se ha definido y desarrollado un prototipo sobre laboratorios con dispositivos de interconexión de redes de computadoras, otorgándonos una valiosa experiencia. Esto ha llevado al diseño e implementación de nuevos aspectos de la herramienta dando como resultado la extensión a nuevos conjuntos de laboratorios.*

**Palabras claves:** Laboratorio Remoto, Educación a Distancia, Laboratorio Virtual.

**Workshop:** Tecnología Informática Aplicada en Educación (WTIAE)

## 1. Introducción

Las tecnologías de comunicaciones y redes posibilitan ya soluciones operativas para que organizaciones, grupos, comunidades y sociedades desarrollen actividades que en el pasado requerían la presencia física y la interacción directa entre los participantes. Este trabajo cooperativo (comunicarse, capacitarse, enseñar, compartir) va desarrollándose cada vez más a medida que los servicios multimedia interactivos mejoran, las comunicaciones se hacen más veloces, y las TIs amplían su alcance.

Las tecnologías cuyo objetivo es soportar actividades de colaboración entre individuos o grupos se identifican genéricamente con el término *groupware*. Dentro de ellas, la denominada Trabajo Cooperativo Soportado por Computadoras, o CSCW (*Computer Supported Cooperative Work*), define espacios de trabajo [1] y esquemas de interacción (diferenciados en sincrónicos y asincrónicos [2]).

Las interacciones del tipo asincrónico son aquellas que no requieren la presencia física simultánea de las personas para interactuar. Las TIs para interacciones asincrónicas han madurado sensiblemente, habiendo aparecido, durante los últimos años, ejemplos notablemente exitosos de *groupware* asincrónico. *LOTUS Notes* es considerado probablemente el producto comercial más representativo en esta área. Muchas aplicaciones de Internet también pueden ser consideradas en cierto modo como tecnologías *groupware*.

La interacción sincrónica hace referencia al intercambio verbal o visual de mensajes, tal como el que se da en conferencias, conversaciones y otras actividades en las que varios participantes colaboran y están físicamente presentes. El teléfono es un muy buen ejemplo de una tecnología antigua que soporta una forma simple pero muy efectiva de interacción sincrónica. Durante mucho tiempo las computadoras y la Internet han tenido formas primitivas de servicios interactivos orientados a caracteres. Hoy, servicios de voz y video de baja calidad sobre IP son también una práctica común con aplicaciones populares como *Yahoo Messenger*, *ICQ* o *MSN Messenger*.

En aquellos casos especiales de interacción remota donde se requiere una determinada calidad de telepresencia, existen demandas más rigurosas de ancho de banda y confiabilidad de comunicaciones para lograr una interacción apropiada [2]. Para realizar experimentos a gran escala se requiere entonces una colaboración muy compleja entre las organizaciones participantes. Además, la disponibilidad de una infraestructura de gran ancho de banda está permitiendo la realización de estos experimentos, proveyendo una mejor comprensión del rol que jugará la interacción sincrónica en los servicios de comunicación del futuro.

A pesar de que las aplicaciones de colaboración remota no han abandonado completamente la fase de experimentación, hay muchos dominios donde estas parecen tener las mismas características que la presencia física. Esto podría reducir el traslado y/o movimiento de personas con beneficios sustanciales para la productividad total de una organización.

En la mayoría de las aplicaciones CSCW el proceso de colaboración se realiza a través de estaciones de trabajo o computadoras personales, donde una aplicación multimedial distribuida permite que los usuarios compartan elementos o medios proveyendo una vista coherente de los mismos. Se denominan sitios a los lugares físicos en los que se hallan las computadoras o estaciones de trabajo, y donde los participantes se unen a un determinado evento distribuido. Un evento distribuido consiste en una secuencia de interacciones realizadas a través de la aplicación entre varios sitios.

Las aplicaciones CSCW poseen específicamente tres partes conceptuales:

- 1) Telepresencia: Lograr la sensación de presencia de los participantes remotos a través de audio, video, o cualquier otro medio de representación.
- 2) Espacio de trabajo compartido: Un espacio compartido que permita a los usuarios lograr una vista común y comprensión de los objetos o ideas, involucrados en la colaboración.
- 3) Control de interacción: Este es el medio por el que se logra la colaboración ordenada entre los participantes remotos. Esto se consigue a través de la funcionalidad de administración del ambiente que tiene control total de los elementos que la aplicación necesita utilizar.

El proyecto “Prácticas Remotas sobre Laboratorios Físicos y Virtuales” [3] plantea una primera aproximación a la realización de un modelo de CSCW mediante la creación de un *framework* que permita definir y hacer uso de laboratorios remotos, cuyo soporte pueda ser tanto equipamiento físico como virtual. El objetivo de este trabajo es presentar las experiencias logradas en la implementación de un primer prototipo de laboratorio remoto, actualmente en funcionamiento en un laboratorio de redes de computadoras, y describir la evolución en su arquitectura así como también aspectos de implementación en desarrollo. En la sección siguiente mencionaremos algunas implementaciones destacadas a nivel mundial. En la sección tercera se describirá la última adaptación de la arquitectura en la que estamos trabajando, y en la cuarta analizaremos los detalles de implementación analizados desde el punto de vista de los distintos actores participantes en la aplicación, finalizando en las dos últimas secciones con las conclusiones obtenidas y los puntos que aún resta desarrollar.

## **2. Avances en laboratorios Virtuales**

Varios trabajos publicados, revisan el problema de la implementación de laboratorios remotos. En [4] se define como laboratorio virtual al acceso remoto por internet pudiendo interactuar a distancia y permitiendo realizar tareas que normalmente se desarrollarían en forma local. Su objetivo es ofrecer laboratorios reales cuyo proceso físico puede ser controlado en forma

remota. Los usuarios pueden acceder a consultar documentación, realizar simulaciones o experimentos en plantas reales. Otra publicación [5] describe un laboratorio remoto de ingeniería, con instrumental con restricciones de tiempo real. Se plantean soluciones puntuales, creando una herramienta virtual para realizar experimentos en el laboratorio. En un tercer escenario, descrito en [6], se comparte el acceso de personas a laboratorios reales de física en forma remota, pero sólo se puede realizar una sesión por vez; y estas participan de la misma a través de videoconferencia. En esta clase de entornos restringidos, los alumnos se limitan a ingresar la información solicitada y los dispositivos responden en forma preestructurada. Este tipo de entorno pone límites a la experimentación y creatividad del alumno, además de hacer engorrosa la generación de nuevos experimentos.

El enfoque que planteamos tiene puntos en común con [7] y [8], donde se establecen tres etapas: la primera, de **desarrollo** del entorno, donde se define una arquitectura flexible; la segunda, de **generación de los recursos** (que pueden ser virtuales, remotos, o una combinación de ambos), con su respectivo modelo de interacción; y la última, basada en la segunda etapa, planear y desarrollar un **conjunto de experimentos** coordinado por un agente que asegura el cumplimiento de un conjunto de políticas. En otros trabajos basados en aspectos similares ([9][10]), se encuentra como ejemplo el acceso remoto para la instalación y configuración de sistemas operativos Linux y redes de computadoras. En estos proyectos, el desarrollo está centrado en el alumno y se busca una capacidad de fácil configuración, aunque no contemplan el uso de herramientas de uso libre y sus entornos administrativos resultan poco dinámicos.

Nuestra propuesta busca mantener algunas de las ideas básicas, como el dinamismo al momento de la creación y configuración de recursos, y una complejidad aceptable para la gestión de los laboratorios, todo esto en un marco que se pueda aplicar fácilmente a una amplia gama de laboratorios.

### 3. Arquitectura e Implementación

En la arquitectura del sistema que se describe en este trabajo confluyen distintos actores (Figura 1). A continuación describimos brevemente cada uno de los integrantes y su función dentro del *framework*.

Tenemos como actores principales a los alumnos situados en los equipos clientes, que constan de una computadora con acceso a Internet y un navegador web con soporte para Java. El alumno tiene acceso a los contenidos teóricos vía web, y mediante un *applet* Java puede interactuar con los distintos recursos de forma remota. Luego tenemos un administrador cuya interfaz se hace disponible como un plugin para la plataforma Eclipse [11]. Usando este administrador, el docente genera la plantilla del laboratorio y la exporta en un formato preestablecido (XML) ubicándolo en el servidor de laboratorio. Como interfaz de gestión (aún en diseño) se dispone de una interfaz web, que permite la gestión del perfil del alumno, la consulta de una bitácora de las prácticas realizadas, para seguimiento individual del estudiante, y la revisión de la configuración y uso de los recursos que intervienen en el laboratorio.

En la Figura 1 los servidores se encuentran separados por su rol, con el fin de clarificar el esquema, aunque posiblemente se encuentren en un solo equipo dependiendo de la disponibilidad en la implementación. Los roles de los distintos servidores son:

- *Servidor de LABS*: Este componente en la arquitectura funciona como un portal web para los distintos tipos de clientes, alumnos, docentes y administradores, permitiendo que estos

inicien una sesión segura, accedan a contenidos teóricos varios y puedan iniciar, crear o gestionar las prácticas en algún laboratorio previamente seleccionado.

- *Servidor de Recursos (SR)*: El rol de este actor es el de proveer acceso a los recursos físicos y virtuales. El diagrama de secuencia en la Figura 2 muestra la serie de eventos que ocurren antes de que el alumno pueda comenzar efectivamente a interactuar con los recursos del laboratorio. Puede haber múltiples instancias de este componente dentro de la arquitectura, permitiendo de esta manera la distribución geográfica de los recursos.
- *Servidor de Aplicación de Políticas*: Este servidor es el encargado de aplicar las diferentes políticas sobre los SR para permitir que cada cliente acceda a los distintos recursos del laboratorio.
- *Servidor LDAP*: Este servidor actúa como base de datos de autenticación para alumnos, docentes y administradores para acceder al portal y por lo tanto a los laboratorios.

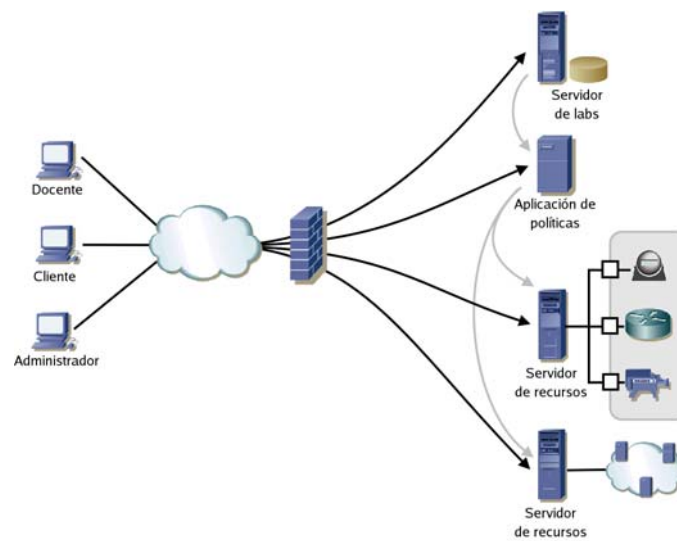


Figura 1. Arquitectura del Laboratorio remoto

Por último, se tienen los recursos propiamente dichos. Estos pueden variar ampliamente en su naturaleza. Se contemplan dispositivos bien conocidos como routers y PCs para armar laboratorios de redes, así como cámaras de video u otros tipos de hardware que puedan ser conectados a una PC y produzcan información que pueda ser procesada y enviada por la red. La Figura 1 muestra cómo uno o varios recursos pueden ser administrados por un representante o SR.

Hasta aquí hemos descripto cada uno de los actores, y cómo éstos interactúan en el tiempo, con el diagrama de secuencias. A continuación describiremos en forma general los *modos de comunicación* posibles entre el cliente y un determinado recurso, una vez establecida la comunicación entre ambos.

- *Comunicación bidireccional interactiva*: en esta categoría tenemos los intérpretes de comandos interactivos gráficos y de texto, como Virtual Network Computing (VNC), shells remotos a consolas Linux, routers, etc.
- *Ejecución remota de comandos sobre recursos*: mediante este modo, el cliente puede ejecutar operaciones sobre los recursos remotos, como por ejemplo apagar un dispositivo, mover un brazo mecánico que da soporte a una cámara web, o cualquier otra operación que se pueda implementar entre una PC y un dispositivo de hardware. Para este mecanismo genérico de comunicación, el Cliente utiliza un protocolo muy simple que consta de una terna (**Recurso, Operación, Parámetros**) como mensaje y recibe como respuesta de manera asincrónica una dupla (**Código, Mensaje**) que identifica el resultado de la ejecución

remota de la operación solicitada. Se pueden recibir varios mensajes de respuesta ante una solicitud de operación remota, como “operación iniciada”, “operación finalizada”, etc.

- *Recepción de flujos de datos*: este tipo de comunicación permite la recepción de flujos de datos genéricos, que pueden ser mostrados en el cliente cuando este último disponga de los filtros (*plugins*) adecuados. Esto será posible gracias a las características de extensibilidad del cliente descritas más adelante. Como ejemplos podemos mencionar la visualización de una cámara web o la lectura de texto simple generado por un sensor conectado a algún recurso.
- *Envío de flujos de datos*: Así como se provee la lectura de datos, también es necesario que el cliente pueda generar un flujo de datos hacia el Servidor de Recursos. La interpretación de este flujo es posible gracias a que se conoce de antemano la naturaleza de los datos enviados (por ejemplo, envío de voz o video desde el cliente hacia el servidor durante una sesión interactiva con un docente). También es un escenario posible el envío de archivos hacia un recurso. Estos archivos pueden representar la configuración de un router, o el programa que será embebido en un Robot para una prueba de programación remota.

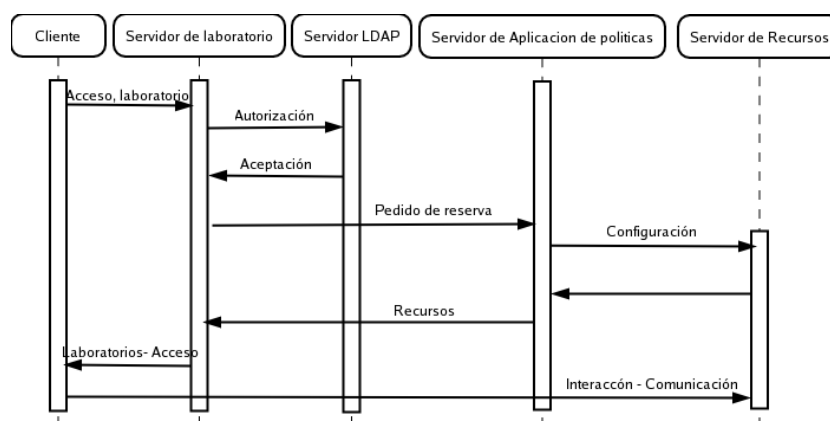


Figura 2. Diagrama de Secuencias

Para la implementación de todos estos modos de comunicación se utilizan protocolos estándar, y este enfoque permite desacoplar las implementaciones del cliente y el manejo del recurso en sí mismo.

No existe un límite explícito para los tipos de Laboratorio que se proyecta poner en funcionamiento mediante este *framework*. Aunque el prototipo inicial del proyecto [3] se basó en dispositivos de redes tales como routers y PCs interconectados, las aplicaciones que pueden implementarse mediante los *modos de comunicación* descritos anteriormente son completamente variables.

#### 4. Aspectos de Implementación

Hasta el momento, si bien muchos de los elementos propuestos se encuentran en diseño, otros ya se encuentran implementados y en uso, otorgando una valiosa experiencia en la conformación y dinámica de las prácticas en forma remota para redes de computadoras.

Se desarrolló un prototipo para un laboratorio con dispositivos de interconexión de redes y computadoras que funcionan como servidores UNIX en su versión Linux, para realizar prácticas de configuración y prueba. Durante estas prácticas se experimentó con diversos escenarios que simulan el funcionamiento de redes locales, redes de larga distancia e interconexión de las mismas. Típicamente, se establecen las operaciones que están permitidas en cada recurso, se establecen determinados valores característicos para los dispositivos (como direcciones IP o máscaras de

subred) y se dejan otros sin configurar para que sean modificados por los alumnos durante la realización de la práctica.

Para el funcionamiento del sistema existe un servidor que provee el acceso a los laboratorios, sus descripciones y enlaces a los distintos recursos, además de controlar el acceso de los usuarios. Otro servidor actúa como representante y está conectado a los recursos reales por medio de puertos seriales (dispositivos y computadoras). Ambos servidores funcionan sobre un sistema operativo Linux.

Las prácticas de los alumnos son llevadas a cabo usando los servicios de un navegador web estándar con soporte para Java. Mediante el acceso a la página web del servidor de laboratorios, y luego de la autenticación del usuario en el servidor (utilizando el protocolo LDAP), se muestra una lista de las prácticas y materiales disponibles para este alumno. Una vez elegido el laboratorio, se delega el control a una página HTML como interfaz del laboratorio, mostrando todos los recursos disponibles y las consignas de la práctica.

La Figura 3 muestra cómo el usuario puede seleccionar con el ratón sobre el gráfico el recurso al que desea acceder, ingresando al mismo a través de una consola SSH implementada en Java [12].

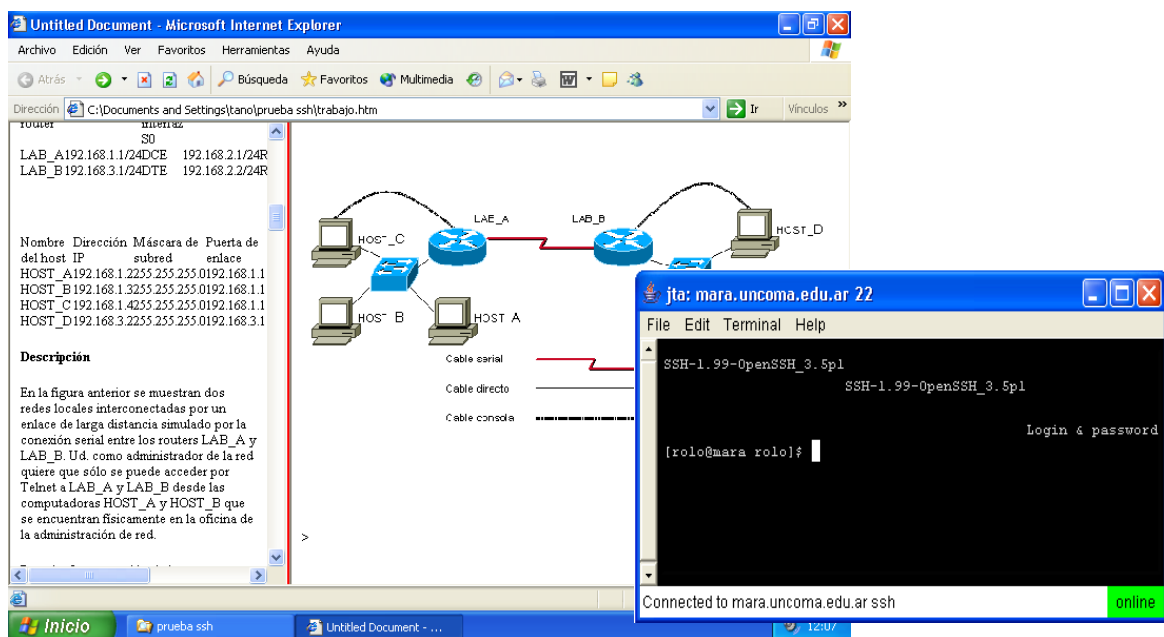


Figura 3. Prototipo

#### 4.1 Modelo de uso del Alumno

Si bien actualmente, como se mencionó en el punto anterior, la interfaz de control consiste únicamente en una página HTML, se advirtió falta de flexibilidad de esta solución para contemplar las distintas modalidades de comunicación a los diversos recursos. Este diseño está en vías de ser reemplazado por un *applet* Java que permitirá descargar el código necesario en tiempo de ejecución, para adaptarse de esa manera a los cambios en tiempo real.

Una meta de diseño del *applet* Java que permite el acceso a los recursos es que pueda ser extendido, tanto en cuanto a su funcionalidad interna como respecto a la interfaz gráfica que usa el alumno. Un ejemplo ayudará a aclarar estos dos conceptos. La implementación actual del cliente hace uso de una aplicación que implementa el protocolo *secure shell* (SSH) como medio de comunicación para enviar los *comandos remotos* a un Servidor de Recursos. Esta última lleva a cabo convenientemente

la comunicación con el recurso en sí mismo, pero existen clases abstractas en el diseño del *applet* que permiten agregar otros mecanismos de comunicación, como RMI o Web Services, en la implementación del cliente. Por otro lado, el que implementa un recurso podrá proveer una interfaz gráfica a medida, que hiciera más confortables el control y visualización. Para ello tendrá que respetar una interfaz de programación del cliente (sin conocer sus detalles de codificación) e implementar en Java su interfaz gráfica que se comunicará directamente con el cliente, quien a su dialogará con el Recurso. Estas dos técnicas se basan en la teoría de Application Frameworks como por ejemplo [13].

## 4.2 Modelo de uso del Docente

El módulo **RN-Editor** (Figura 4), conforma uno de los componentes de la arquitectura definida en este proyecto. Su objetivo principal es la creación en forma visual de un laboratorio, el cual será utilizado en una etapa posterior por los alumnos para operar los dispositivos físicos y virtuales en él representados.

Al igual que los demás módulos, fue diseñado bajo el modelo de software de código abierto, y fue desarrollado utilizando PDE (Plug-in Development Environment) y JDT (Java Development Tools) como un plug-in para Eclipse[11][14], basándose en el framework GEF [15] para aplicaciones de edición gráfica.

Su función principal es la creación de topologías de red de comunicaciones, pero su uso no se limita a este área sino que puede extenderse a un amplio espectro de dominios de aplicación tales como: diseños de jerarquías para departamentos de Recursos Humanos, diseño de gráficos para Matemática, diseños de circuitos lógicos para el área de Ingeniería, etc. Para lograr esto, permite incorporar nuevos recursos definidos por el usuario a través de archivos XML.

Básicamente, con el RN-Editor se puede:

- Crear diagramas de redes de recursos a través de un asistente provisto (que se incorpora a los asistentes por defecto a la plataforma Eclipse)
- Editar archivos de diagramas.
- Agregar recursos a un diagrama tanto predefinidos como definidos por el usuario a través de una paleta de recursos disponibles, como se observa en la figura 4.
- Conectar los recursos a través de una herramienta de conexión.
- Editar las propiedades de cada recurso o conexión a través de la Vista de Propiedades.
- Exportar los contenidos de un diagrama a una imagen JPG y a un archivo XML para ser utilizado por otros componentes.

Como todos los *plugins* que pueden ser enlazados a Eclipse, RN-Editor cuenta con un *manifiesto* que establece las características básicas de su comportamiento, sus dependencias y cómo extiende a algunas componentes de Eclipse. La clase básica es RNEditorPlugin, que define una única instancia (a través del patrón Singleton) que mantendrá la información de estado del *plugin*. El resto de las clases son utilizadas de acuerdo a las extensiones determinadas en el manifiesto; de éstas la más importante es RNEditor, aquella que encapsula la funcionalidad inicial de la herramienta.

Las clases están subdivididas en cinco paquetes: un paquete **editor**, que contiene las clases que representan al editor en sí mismo (entre ellas **RNEditor**), un paquete **model**, que incluye los objetos propios del modelo, un paquete **parts** con las clases implementadoras de los controladores MVC, y finalmente los paquetes **commands** y **actions** que incluyen las clases con los comandos y acciones utilizados por el usuario para interactuar con la herramienta.

El *plugin* permite la generación de una imagen JPG que equivale a la vista del diagrama en el editor gráfico, y un archivo XML donde se listan todos los recursos y conexiones creadas, así como las propiedades de cada uno.

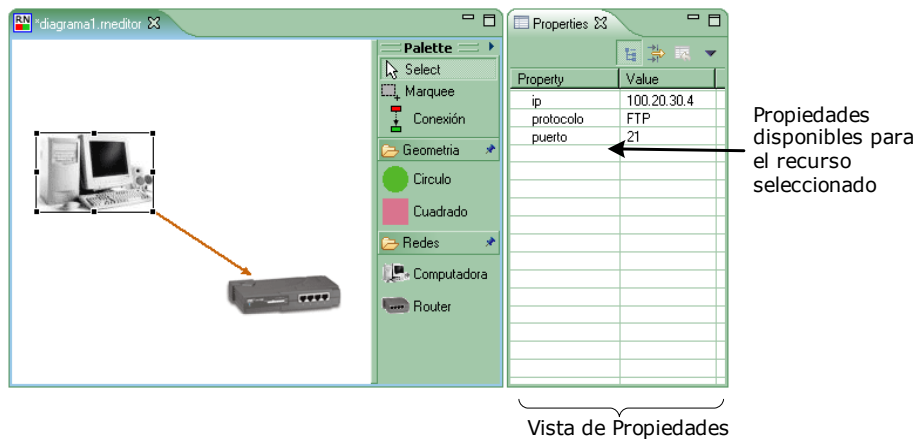


Figura 4. Editor de Recursos

### 4.3 Seguridad

La seguridad es un tema crítico en cualquier aplicación remota, especialmente cuando entre los distintos actores de la aplicación se intercambia información confidencial del sistema. Para esto se han relevado y diseñado soluciones para varios aspectos relacionados con el desarrollo del laboratorio. A continuación describiremos cada uno de ellos.

El primer aspecto es la autenticación. En este trabajo intentamos acceder a múltiples recursos por medio de un único proceso de ingreso, más conocido como *single sign on* (SSO). El objetivo de esta arquitectura es transferir la funcionalidad y complejidad de los componentes, en este caso los recursos de laboratorio, a un solo servicio de autenticación en el dominio. El ejemplo más concreto de este sistema de autenticación es *kerberos*[16]. Este protocolo usa fuertes algoritmos de encriptación para que el cliente pueda probar su identidad al servidor, y viceversa, a través de una conexión de red insegura. Además, después de que el cliente prueba su identidad, puede encriptar toda su comunicación para asegurar privacidad e integridad de datos. Con el fin de cumplir con algunos de los objetivos de esta tecnología, usamos para la autenticación y la administración de cuentas de usuario un repositorio basado en un esquema de directorio LDAP. Cada vez que los usuarios inician su sesión de trabajo se validan a través de este servicio, así como al momento del acceso a las propiedades de cada recurso. En estos momentos se está implementando un control de acceso RBAC (*Role Based Access Control*) para limitar cada una de las operaciones posibles dependiendo el rol o el grado del estudiante sobre los recursos en cada laboratorio.

El segundo aspecto es la encriptación y uso de canales seguros. Como se mencionó en el punto anterior, la modalidad SSO además de realizar una autenticación centralizada tiene como objetivo establecer canales seguros para asegurar privacidad e integridad de los datos transmitidos. En nuestro trabajo hay dos procesos que constituyen puntos críticos para la seguridad. El primero es cuando los alumnos ingresan al sistema a través de Internet; el segundo, cuando éstos comienzan a realizar las prácticas. Para la protección del primer proceso utilizamos un servidor Apache seguro con la modalidad SSL, o *Secure Sockets Layer*. SSL trabaja usando una clave privada para encriptar los datos transferidos a través de la conexión HTTP. Actualmente, la conexión segura se establece con el servidor Apache y el alumno ingresa su identificación y contraseña. Ésta viaja a través de un canal seguro al servidor LDAP, donde se verifica y valida esta información y se extraen los datos relevantes del perfil del alumno. Se encuentra en diseño la incorporación de una unidad certificadora para emitir certificados a los alumnos así como también a los servidores. Para el



segundo proceso, donde el usuario comienza a realizar las prácticas, existen varios modos de comunicación cliente-recurso; y las metodologías de seguridad a ser aplicadas varían según los requerimientos del caso, como uso de canales seguros SSH, RMI-SSL o canales seguros en Web Services. Estas herramientas encriptan todo el tráfico, incluyendo las contraseñas, para eliminar varios tipos de ataque al nivel de red, además de implementar túneles y algunos servicios de autenticación.

Por último se recurre también al control de acceso entre redes con cortafuegos o *firewall*. Éste se usa básicamente para filtrado de paquetes de acuerdo a políticas de uso predefinidas (como vetar el que los servidores de laboratorio inicien conexiones a redes en el exterior). Para el equipo del cortafuegos se utiliza una máquina servidor con sistema operativo Linux que incorpora por defecto como cortafuegos para IPV4 en el núcleo del sistema operativo *netfilter*.

#### **4.4 Alta Disponibilidad**

La arquitectura del Laboratorio puede dar cabida a una amplia variedad de actividades de aprendizaje, con diferentes recursos y modalidades de uso. En general se requerirá alta interactividad y sincronización de operación, cualquiera sea el conjunto de recursos utilizado durante las prácticas. El Laboratorio, insertado en la Internet, con servicio *best-effort*, tiene una usabilidad suficientemente comprometida por los retardos y la falta de confiabilidad que experimentamos frecuentemente como usuarios de la red. Será necesario considerar, junto con la arquitectura del Laboratorio, de qué forma se posibilitan las mejores condiciones en que los usuarios puedan acceder a este servicio.

Todo sistema complejo enfrenta problemas al tener que garantizar la disponibilidad y escalabilidad de sus recursos. En un ambiente como éste, destinado a soportar procesos educativos, la calidad de la experiencia de uso cobra una dimensión nueva debido a lo subjetivo del proceso de aprendizaje[17].

A mayor ubicuidad de la solución, potencialmente mayor ámbito donde será compartida, y mayores restricciones de asignación de recursos. El resultado es que un usuario que ha obtenido un recurso deberá hacer uso del mismo en su franja de tiempo habilitada. La no disponibilidad del recurso en el momento preciso equivaldrá a frustrar la misión del ambiente, con altos costos tanto en el desarrollo práctico de un curriculum como en la experiencia subjetiva del estudiante. Esta es una diferencia significativa con otras plataformas colaborativas construidas con material más estático, donde la asincronicidad de la operación es mayor, donde pueden intervenir instancias de aceleración y replicación de contenidos (como niveles de cache) y donde, en otras palabras, la disponibilidad juega un rol menos crítico.

Por todos estos motivos se requiere contemplar los aspectos de disponibilidad en la especificación de esta arquitectura. Llamamos Alta Disponibilidad a un nivel de disponibilidad implicado por un diseño que debe alcanzar o superar los requerimientos de operación para los cuales se implementa ese sistema [18]. De hecho, Alta Disponibilidad es una actitud de diseño. Diseñar para Alta Disponibilidad implica construir un supersistema robusto, que contenga a aquel del cual se esperan fallos en diversos modos.

El diseño de Alta Disponibilidad opera según el patrón básico de identificar los puntos únicos de falla, y eliminarlos introduciendo sistemáticamente instancias redundantes de recursos. Para cada recurso redundante, se dispone una instancia activa y una en rol de vigilancia o *standby*. Un componente usual de las soluciones de Alta Disponibilidad es un servicio de *failover* o acción ante fallas que capture las condiciones de falla del elemento activo, y señalice esta condición al elemento

en *standby*, que pasará a tomar el rol de activo hasta que la condición desaparezca o sea solucionada.

Un análisis de disponibilidad del sistema conformado por el Laboratorio permite distinguir tres zonas clave, de lo interior a lo exterior:

- Los recursos propiamente dichos, cuya utilización es viabilizada por el Laboratorio;
- La red interna, que comunica los elementos del Laboratorio entre sí y a uno o más dispositivos de frontera; y
- Estos dispositivos de frontera junto con su conexión a Internet, que hacen disponible a todo el conjunto.

Cada instancia del Laboratorio en funcionamiento está acoplada a un conjunto de instancias de recursos que salvo excepciones son, en principio, únicos e irremplazables. El análisis de puntos únicos de falla del sistema, con el objetivo de eliminarlos por replicación de recursos, termina necesariamente en este obstáculo.

En cambio, en la red interna se tiene una cantidad de opciones. De los componentes de la red interna, se distinguen los elementos de conectividad (switches, routers, cableado) y los servidores de acceso al Laboratorio. La disponibilidad de los primeros se ve frecuentemente apoyada por herramientas estándar conocidas, como agregación de links en modo primario/*standby*, protocolos inteligentes de ruteo de nivel 3 y protocolo STP de creación de árboles de expansión sobre vínculos redundantes. Los elementos de acceso al conjunto de recursos quedan amparados por la posibilidad de replicación y funcionamiento en modo cluster:

- Servicios de autenticación, inherentemente redundantes gracias a las características distribuidas del protocolo LDAP.
- Servidores de contenido como descripciones de laboratorios, páginas de ejercicios, diagramas, consignas, ayuda, en modo redundante.
- Puntos de aplicación de políticas, en modo redundante.

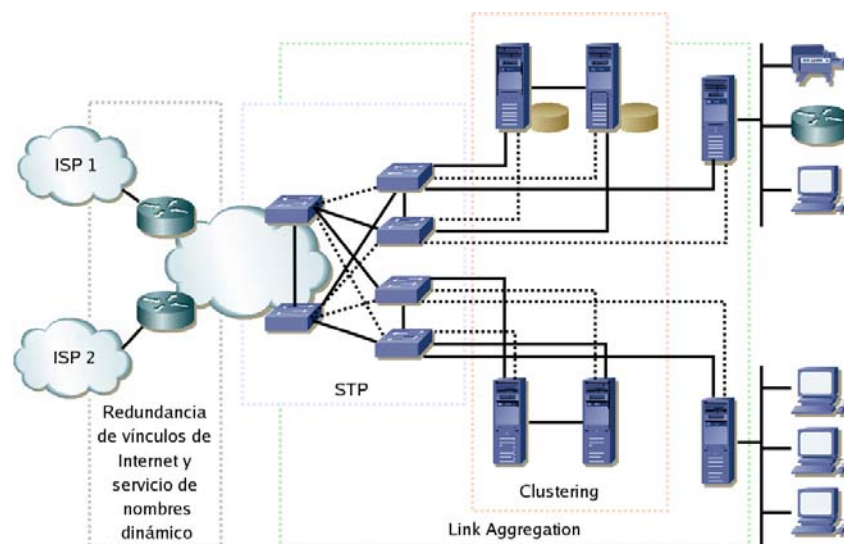


Figura 5. Alta Disponibilidad

Bajo esta visión, el componente crítico para la disponibilidad que queda es el más externo, o punto de acceso al servicio del Laboratorio desde Internet. Para este componente, los riesgos de disponibilidad se centran en los relativos a la prestación del servicio de conexión a Internet del Laboratorio. Si se pretende asegurar la disponibilidad del sistema, se impone considerar doble suscripción a estos servicios, en forma concurrente con esquemas de DNS dinámico en *failover*.

Una línea de acción para el proyecto, consistente con estas premisas, es el estudio de herramientas de Alta Disponibilidad, que van desde la implementación de clusters resilientes hasta herramientas de monitoreo de acuerdos de nivel de servicio (SLAs). La Figura 5 muestra un resumen de los puntos de aplicabilidad del diseño de Alta Disponibilidad en el contexto del Laboratorio, y las herramientas pertinentes en cada zona.

## 5. Trabajo Futuro

Actualmente nos encontramos trabajando en un proyecto de incorporación de Robótica educativa en la escuela media [19]. Identificamos en el Laboratorio de Robótica un espacio en el que se logra despertar en los alumnos un deseo por el descubrimiento y la invención como pocos otros ámbitos escolares permiten. La robótica no sólo puede desarrollarse como espacio exclusivo para el área informática, sino también posibilitar la concreción de proyectos educativos interdisciplinarios en los que intervienen diversas áreas. En colaboración con Ciencias Naturales en la elaboración y obtención de mediciones útiles para sus estudios; con Física en el trabajo con mecanismos de palancas, neumostática; con Matemática en el uso de variables, proposiciones lógicas y conceptos de Geometría; con Informática como introducción a la programación; etc.

Considerando que las instituciones deben compartir los recursos a los que acceden con fines educativos; que experiencias tan enriquecedoras como las que ofrecen los laboratorios de robótica no deben quedar circunscriptas a las instituciones que logran acceder al recurso físico y que los avances de las tecnologías de comunicaciones facilitan el intercambio de información, se abre la posibilidad de compartir solidariamente los recursos mediante la creación y desarrollo de laboratorios remotos físicos y virtuales.

Proponemos que el acceso remoto a recursos ubicados físicamente en otra institución sea soportado por una plataforma on-line para proyectos educativos colaborativos permitiendo al docente resignificar el uso que los adolescentes hacen de Internet, convirtiendo lo que se concibe como un elemento de entretenimiento y distracción en una poderosa herramienta de estudio y construcción colectiva de conocimiento. La incorporación de estos laboratorios implica necesariamente replantear la práctica docente, con rediseño continuo de las experiencias a desarrollar y de los modelos organizativos que potencien la colaboración y cooperación entre: docente-docente, docente-alumno y alumno-alumno.

## 6. Conclusiones

Como se ha planteado en este trabajo, el problema de la creación y funcionamiento de laboratorios remotos supone una gran complejidad tanto a nivel tecnológico como didáctico. Nuestro desafío es encontrar una propuesta lo suficientemente general para abarcar distintos tipos de problemáticas brindando una solución satisfactoria a cada uno de estos aspectos.

El caso práctico en el que estamos trabajando aporta dos aspectos fundamentales. El primero consiste en poder aplicar y probar conceptos del *framework* bajo desarrollo, y el restante es poder contar con una herramienta útil para la puesta en producción de cursos que puedan formar integralmente a los alumnos que deseen capacitarse. Si bien es prematuro enunciar conclusiones definitivas, se pueden enumerar algunos de los problemas con los que nos encontramos. En particular resultan evidentes la dificultad de adaptar las aplicaciones a este tipo de entornos y la limitada infraestructura de comunicaciones con garantías en la calidad del servicio. Nos aguarda aún una tarea de investigación hacia la caracterización de las soluciones disponibles para estos problemas. Por otro lado, aparecen igualmente evidentes la conveniencia de la configuración única y centralizada del acceso a los laboratorios remotos, optimizada y consistente, y la flexibilidad con la cual pueden aprovechar los alumnos esta forma de trabajo.

El trabajo aún está en sus inicios, dirigiéndose actualmente nuestra atención a la práctica realizada por los alumnos a fin de obtener realimentación para orientar los siguientes esfuerzos. Estos se aplicarán a las implementaciones para las áreas de administración y creación dinámica de laboratorios. En un futuro cercano éstas se encontrarán concluidas, así como las extensiones a otros entornos de capacitación y operación.

## Referencias

- [1] Wilson P. –“Computer Supported Cooperative Work. Computer Networks and ISDN Systems” Proceedings of the 18th Annual International Conference on Information Technologies and Programming, pp. 9 - 22, Institute of Mathematics, Bulgarian Academy of Science,1993.
- [2] Quemada J., De Miguel T., Azcorra A., Pavón S. - “ISABEL: A CSCW Application for the Distribution of Events” – Departamento de Ingeniería de Sistemas Telemáticos, Workshop on Multimedia Networks and Systems. 1996
- [3] Bertogna, Leandro; Del Castillo, Rodolfo; López Luro, Francisco; Zanellato, Claudio R. – “Practicas Remotas sobre Laboratorios físicos y virtuales”, Workshop de Investigadores en Ciencias de la Computación, 2004.
- [4] Di Stefano, A.; Fazzino F.; Lo Bello, L.; Mirabella, O. – “VirtualLab: A Java application for distance learning”. International Conference on Emerging Technologies and Factory Automation, 1997
- [5] Overstreet, J. W. and A. Tzes. “An Internet Based Real-Time Control Engineering Laboratory”, IEEE Control System Magazine 1999.
- [6] C. C. Ko, B. M. Chen, J. Chen, Y. Zhuang, and K. C. Tan, “Development of a web-based laboratory for control experiments on a coupled tank apparatus”, IEEE Transactions in Education, Vol. 44, No. 1, pp. 76-86, 2001.
- [7] Y. Lee, W. Ma, D.H.C. Du, and J. Schnepf, "Creating a Virtual Network Laboratory," University of Minnesota Supercomputing Institute Research Report UMSI 99/59, 1999.
- [8] Bing Duan, Keck-Voon Ling and Habib Mir M. Hosseini, "Developing a Framework for Online Laboratory Learning Objects ", Proceedings of WSEAS ICECS and WSEAS E-Activities 2003, Vol, 5, pp. 467-157, Sentosa, Singapore,2003.
- [9] A. Berqia, A. Diop, J. Harms, "A Virtual Laboratory for Practical exercises", in International Conference on Engineering Education ICEE 2002, Manchester, GB, August 19-20,2002.
- [10] Coelho, Paulo and Raúl Oliveira, "Arquitectura do laboratório remoto e-Cassiopeia". In proceedings of CRC'2001, Covilhã, November, 2001.
- [11] Eclipse Platform - Technical Overview. “[www.eclipse.org/whitepapers/eclipse-overview.pdf](http://www.eclipse.org/whitepapers/eclipse-overview.pdf)”. 2001
- [12] The Java(tm) Telnet Application/Applet - <http://www.mud.de/se/jta/>
- [13] Eckstein, J.: Empowering Framework Users. In: Fayad, M.E., Schmidt, D.C., Johnson, R.E. (eds.): Building Application Frameworks: Object-Oriented Foundations of Framework Design, John Wiley & Sons, New York, (1999) 505-521.
- [14] Gamma, E., and Kent, B., Contributing to Eclipse: Principles, Patterns, and Plug-Ins, Addison-Wesley, 2004
- [15] William Moore, David Dean, Anna Gerber, Gunnar Wagenknecht and Philippe Vanderheyden, Eclipse Development using the Graphical Editing Framework and the Eclipse Modeling Framework, IBM RedBooks, 2004, ISBN 0738453161
- [16] S.P. Miller, B.C. Bneuman, J.I Schilles and J.H. Saltzer. Kerberos Authentication and Autorization System. In Project Athena Technical Plan, chapter E.2.1. MIT. 1987.
- [17] Grosclaude Eduardo, "Soporte de Sistemas para Procesos Colaborativos", Workshop de Investigadores en Ciencias de la Computación, Río Cuarto, 2005.
- [18] Evan Marcus, Hal Stern, "Blueprints for High Availability", Second Edition, Wiley, 2003.
- [19] Sánchez Laura, Rodríguez Jorge, Narváez Rafael. “Hacia un laboratorio escolar de robótica remoto”, Workshop de Investigadores en Ciencias de la Computación, Río Cuarto, 2005.