

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO INFORMÁTICO

Título del proyecto:

ULABGRID2: “SISTEMAS GRID Y PEER-TO-PEER
AUTOAJUSTABLES. REDES SOCIALES Y MECANISMOS
DESCENTRALIZADOS PARA LA GESTIÓN DE RECURSOS”

José Andrés del Campo Martínez

Luis Manuel Díaz de Cerio

Óscar Ardaiz Villanueva

Pamplona, 30 de Junio de 2010

Índice

1. Introducción	3
1.1 Origen del proyecto ULabGrid2	4
1.2 Conceptos relacionados	5
2. Estado del arte	9
2.1 Framework ULabGrid	10
2.2 Otros trabajos relacionados	13
3. Objeto del proyecto	16
3.1 Objeto general del proyecto	17
3.2 ULabGrid2 integrado con el CMS MOODLE	19
3.2 ULabGrid2 integrado con la red social FaceBook	21
4. Desarrollo	23
4.1 ULabGrid2 integrado con el CMS MOODLE	24
4.1.1 Framework ULabGrid2 + MOODLE	24
4.1.2 Prototipo ULabGrid 2 + MOODLE	25
4.1.3 Tecnologías utilizadas	32
4.1.4 Implementación	33
4.1.5 Problemas encontrados	40
4.1.6 Etapas y plazos de ejecución	42
4.2 ULabGrid2 integrado con la red social FaceBook	44
4.2.1 Framework ULabGrid2 + FaceBook	44
4.2.2 Prototipo ULabGrid 2 + FaceBook	45
4.2.3 Tecnologías utilizadas	53
4.2.4 Implementación	55
4.2.5 Problemas encontrados	64
4.2.6 Etapas y plazos de ejecución	67
5. Análisis de resultados	69
5.1 ULabGrid2 integrado con el CMS MOODLE	70
5.2 ULabGrid2 integrado con la red social FaceBook	77
6. Conclusiones	81
7. Líneas futuras	84
8. Publicaciones	87
9. Bibliografía	89

1. Introducción

En esta sección se indica brevemente el origen y propósito de este proyecto. Además, se introducen algunos conceptos a los cuales se hará referencia durante el desarrollo de esta memoria.

1.1 Origen del proyecto ULabGrid2

Las instituciones educativas han estado durante años centradas en el profesor: los profesores proveen el material educativo, imparten las lecciones, asisten a los estudiantes, realizan exámenes, etc. De este modo, el soporte por ordenador del aprendizaje ha sido construido siguiendo este modelo: soporte para las actividades del profesorado con Sistemas de Administración de Cursos (Course Management System, CMS) que proveen a los estudiantes acceso al material didáctico, acceso a tests automatizados y acceso a evaluaciones. Sin embargo, están surgiendo nuevos paradigmas educativos: el modelo de aprendizaje Peer-to-Peer promueve que los peers sean estudiantes y profesores al mismo tiempo “aprendiendo con y de cada uno” [1]. El paradigma educativo Peer-to-Peer puede tener lugar en diferentes contextos de aprendizaje y diferentes circunstancias: Aprendizaje de adultos y niños, formal e informal, a distancia o presencial [2]. El aprendizaje está cambiando “desde cursos presenciales a una conversación global” soportada por Internet y las herramientas Web 2.0 [3]. El mayor conductor hacia este cambio es la naturaleza descentralizada de Internet. Las instituciones y sistemas educativos centralizados ya no son necesarios. Internet permite a las personas conocerse de manera descentralizada mediante el aprendizaje Peer-to-Peer. Los sistemas Web 2.0 permiten a las personas conectarse entre sí mediante funcionalidades de redes sociales. El uso de redes sociales para el soporte del aprendizaje ha sido denominado como “conectivismo” [4]. Esta teoría propone que el aprendizaje en red tiene muchas ventajas sobre el aprendizaje en grupos cerrados.

El aprendizaje Peer-to-Peer debería ser soportado por sistemas de información de diferentes formas. Dado que el aprendizaje Peer-to-Peer es una actividad distribuida, en este trabajo enfocamos la exploración de sistemas de soporte distribuido, específicamente sistemas Peer-to-Peer y Grid. Las redes Grid y Peer-to-Peer suponen dos conceptos que podrían ser considerados diferentes a primera vista. No existen definiciones exactas que nos permitan comparar sus similitudes y diferencias de una forma no ambigua. Sin embargo, la idea general tiende a separar ambos mundos teniendo en cuenta la infraestructura utilizada para permitir acceso supercomputadores y conjuntos de datos (Grids), y aquellos que permiten comunidades ad hoc de clientes finales para anunciar y acceder a ficheros de los ordenadores compartidos (P2P). Sin embargo, nuestra contribución en este trabajo incluye una aproximación a la convergencia Peer-to-Peer y Grid en el área del aprendizaje, donde permitimos que grupos distribuidos se beneficien de los mecanismos subyacentes P2P y Grid.

En este proyecto se pretende dar continuidad al trabajo comenzado con el proyecto ULabGrid, emprendido por el grupo de investigación “Tecnologías y Aplicaciones Informáticas Para Entornos Colaborativos” de la universidad Pública de Navarra, mediante el cual se planteó una solución Grid a el trabajo colaborativo en la enseñanza, más en particular, en las clases prácticas o de laboratorio.

Es importante destacar que mediante ULabGrid2 se pretende extender las funcionalidades de ULabGrid a un ámbito más general, no por ello desechando el trabajo desarrollado con relación a la enseñanza, pero sí permitiendo un mayor aprovechamiento de las posibilidades que brindan los sistemas Grid y P2P, añadiendo además las ventajas aportadas por el uso de redes sociales y gestores de contenidos como se explicará a lo largo de esta memoria.

Este mayor aprovechamiento al que hago referencia, puede ser considerado como una aportación a la emergente e-Ciencia cuya necesidad se fundamenta en la creciente exigencia por parte de los científicos de más recursos de procesamiento y almacenamiento de datos, así como de nuevas formas de trabajo colaborativo que conduzcan a la sociedad del conocimiento. [5]

Además, la tecnología derivada del grid abre un enorme abanico de posibilidades para el desarrollo de aplicaciones en muchos sectores. Por ejemplo: desarrollo científico y tecnológico, educación, sanidad, y administración pública. [6]

1.2 Conceptos relacionados

A continuación se introducirán algunos conceptos que serán tratados durante el desarrollo de la memoria como son: Grid, P2P, red social, gestor de contenidos, e-Learning, CSCL (Aprendizaje soportado por ordenador) y trabajo colaborativo (groupware).

Grid

Los Grids surgieron dentro de las comunidades científicas que reconocieron los beneficios de usar una infraestructura que conecta y comparte recursos que están localizados geográficamente y/o organizados de forma dispersa. Las motivaciones iniciales para el crecimiento de las tecnologías grid fueron las necesidades, que estas comunidades tuvieron, de acceder a recursos remotos así como a grandes bases de datos y supercomputadores para el propósito de realizar simulaciones a gran escala y análisis de datos.

Tras el paso de unos pocos años, las tecnologías grid han madurado hasta un punto en el cual están siendo adoptadas por muchas otras comunidades con diferentes necesidades y requerimientos.

Los recursos que hoy se comparten en un grid pueden ser desde ordenadores (PC, supercomputadores, PDA, portátiles, móviles, etc), hasta software u otro tipo de información.

Nosotros creemos que desde el momento en el que los grid forman parte de la infraestructura de recursos de Internet, deberían permitir el uso de nuevas aplicaciones y actividades de aprendizaje en el área de la educación.

Características:

- Capacidad de balanceo de sistemas
- Alta disponibilidad
- Reducción de costes

Los sistemas grid, también presentan hoy en día algunas dificultades debido a su naturaleza:

- Recursos heterogéneos
- Gestión de recursos, descubrimiento, asignación, reserva ...
- Necesidad de desarrollo de aplicaciones orientadas a su funcionamiento en un Grid o “gridificadas”

En definitiva, nos encontramos ante un paradigma de computación distribuida altamente versátil, escalable y que permite combinar la potencia de muchos equipos para lograr una capacidad global prácticamente ilimitada. Sus principales inconvenientes provienen de la dificultad para sincronizar los procesos de todos estos equipos, monitorizando recursos, asignando cargas de trabajo y estableciendo políticas de seguridad informática fiables. [6]

P2P

Una red peer-to-peer (P2P) o red de pares, es una red de computadoras en la que todos o algunos aspectos de ésta funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es decir, actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red.

Características:

- Escalabilidad
- Robustez
- Descentralización
- Anonimato

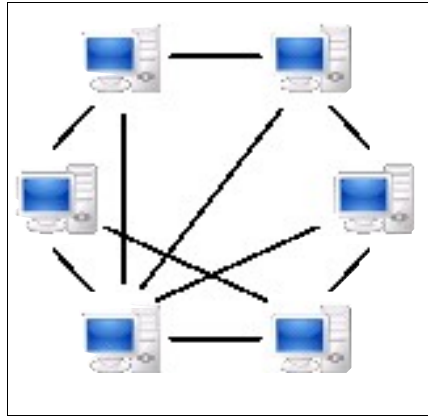


Figura 1. Ejemplo de una red P2P

Las infraestructuras grid y Peer-to-peer (P2P) tienen mucho en común y especialmente la idea básica de compartición de recursos. Entre las características diferentes podemos ver la P2P como más anónima y generalizada en ordenadores de usuarios de Internet, mientras que las grids nacen de una estructura de nodos más controlada y jerarquizada en centros científicos

Red Social (SW)

Un servicio de red social se centra en la construcción y la verificación de las redes sociales online para las comunidades de personas que comparten intereses y actividades, o que están interesados en explorar los intereses y las actividades de otros, y que requiere el uso de software.

En general, los servicios de redes sociales permiten a los usuarios crear un perfil para ellos mismos, y se pueden dividir en dos grandes categorías: la creación de redes sociales internas (ISN) y la creación de redes sociales externas (ESN) como, por ejemplo, MySpace, Weckee, Bebo y Facebook. [6] [7]

Se puede afirmar que las redes sociales en Internet están muy relacionadas con el desarrollo de la Web 2.0 y las comunidades virtuales.

Gestor de contenidos (CMS, Content Management System)

Software que brinda un framework para facilitar la gestión de contenidos (creación, administración, publicación, etc.), generalmente utilizando tecnologías web. Estos sistemas suelen caracterizarse por ofrecer una interfaz sencilla que sirva para administrar la base de datos que almacena los contenidos gestionados en el sitio. Los gestores de contenidos suelen administrar también la gestión de usuarios, brindando la posibilidad de crear varios perfiles que asignen diferentes permisos a distintos tipos de usuarios, por ejemplo, puede existir un usuario administrador con capacidad de gestionar todas las características del CMS, usuarios con permisos de edición de determinadas secciones del sistema y usuarios por defecto sin capacidad de edición. Además, este tipo de sistemas suelen permitir la colaboración de varios usuarios en el mismo trabajo y la interacción mediante herramientas de comunicación como chats o foros de discusión.

En este proyecto se ha hecho uso de un gestor de contenidos orientado a la enseñanza, este tipo de gestores de contenidos son llamados Sistemas de Gestión del Aprendizaje (LMS, Learning Management System) y se caracterizan por dar soporte a e-Learning y organizar los contenidos en

cursos. Las principales funciones del LMS son: gestionar usuarios, recursos así como materiales y actividades de formación, administrar el acceso, controlar y hacer seguimiento del proceso de aprendizaje, realizar evaluaciones, generar informes, gestionar servicios de comunicación como foros de discusión, videoconferencias, entre otros. [6]

e-Learning

e-Learning hace referencia a la educación apoyada por medios electrónicos, no sólo Internet, si no también cualquier tipo de soporte multimedia como puede ser un CD-ROM, aunque puede afirmarse que Internet es la herramienta principal para hacer posible este modelo educativo, normalmente asociado a la educación a distancia, aunque también puede ser considerado como un gran apoyo para la mejora de la educación presencial ya que hace posible una gestión real del conocimiento: intercambio de ideas, opiniones, prácticas, experiencias, etc.

Las plataformas más ampliamente utilizadas para proporcionar un servicio de e-Learning son los CMS o LMS.

Se puede considerar que existen varios tipos de servicios e-Learning, menciono a continuación los más importantes:

- Aprendizaje basado en ordenador: Se refiere al uso de computadoras como elemento clave en entornos educativos.
- Entrenamiento basado en ordenador: Hace referencia a actividades de aprendizaje a realizar mediante computadores, que suelen permitir evaluaciones automáticas, de forma que el alumno puede recibir un feed-back instantáneo además de completar dichas actividades siguiendo el ritmo que se adapte a su situación.
- Aprendizaje colaborativo soportado por ordenador (CSCL, Computer-Supported Collaborative Learning): supone una de las técnicas más innovadoras de enseñanza/aprendizaje y como es lógico, esta basada en la utilización de las nuevas tecnologías para alcanzar una educación que cambie el concepto de “enseñanza directa” desde el profesor al alumno, permitiendo un nuevo paradigma educativo que permita el aprendizaje desde la colaboración. Estas nuevas técnicas, en las que el estudiante adopta una actitud más activa frente a su pasividad en el aprendizaje tradicional, es también conocido como e-Learning 2.0 y se abordará en más detalle a continuación. [6][8]

Aprendizaje colaborativo soportado por computador (CSCL, Computer-Supported Collaborative Learning)

El Aprendizaje Colaborativo Apoyado por Computador (CSCL) es un área emergente de las ciencias del aprendizaje referente a estudiar como las personas pueden aprender de manera conjunta con la ayuda de los computadores. [9]

Este tipo de aprendizaje presenta, a simple vista, ventajas tales como: una mayor interacción, compartición de ideas, incremento de la motivación de los estudiantes creando una actitud más positiva hacia el aprendizaje, desarrollo de habilidades sociales, etc.

Existen numerosas herramientas, algunas de las cuales ya han sido mencionadas, que ayudan o pueden servir de apoyo al CSCL, por ejemplo: Wikis, Blogs, LMS (CMS), vídeo-conferencias, chats, whiteboards, campus virtuales, etc. [8]

Otro término utilizado para referirse a este tipo de aprendizaje es el de Aprendizaje Colaborativo Virtual.

Trabajo Colaborativo (groupware)

Este término hace referencia a todo software destinado a permitir que diversos usuarios trabajen de forma concurrente sobre un mismo entorno.

Este software se puede dividir en tres categorías:

- Herramientas de colaboración-comunicación: Facilitan la compartición de información (colaboración asíncrona). Ejemplos: correo electrónico, publicaciones web.
- Herramientas de conferencia: Facilitan la compartición de información de forma interactiva (colaboración síncrona). Ejemplos: conferencias audio o vídeo, chats.
- Herramientas de gestión colaborativa: Facilitan las actividades en grupo. Ejemplo: calendarios electrónicos, sistemas de gestión de proyectos.

2. Estado del Arte

En esta sección se detallará el estado del arte referente al proyecto ULabGrid2. Cabe destacar, que el punto de partida del proyecto ULabGrid2 fue otro proyecto de investigación, ULabGrid [10], que evidentemente supone el punto más importante de esta sección. Además también se describen otros trabajos relacionados, ya sea por ser aplicaciones que permiten la colaboración en grupos, enfocadas al aprendizaje o que utilizan infraestructuras Grid.

2.1 Framework ULabGrid

Las herramientas colaborativas basadas en la infraestructura web/internet como el correo electrónico, los grupos de discusión, las conferencias vídeo/audio y los campus virtuales han sido propuestas e implementadas en muchos escenarios de aprendizaje a distancia. Muchas de estos sistemas están enfocados hacia herramientas individuales que tienden a no ser reutilizables (habilidad para reutilizar sistemas de computación con otras aplicaciones sin necesidad de realizar modificaciones en el propio sistema) a pesar del hecho de que suponen una carga significativa de esfuerzo duplicado. Por ejemplo, un gran número de sistemas (Exploratum [11], JSPICE [12]) están basados en scripts que necesitan ser modificados para añadir nuevas aplicaciones al sistema. Otros diseños son más flexibles, por ejemplo, el prototipo MOL [13] emplea interfaces web estáticas que pueden ser adaptadas para herramientas individuales. Sin embargo, estas interfaces web estáticas no son adecuadas para ninguna herramienta. Las soluciones enfocadas hacia temas individuales son generalmente reutilizables, pero las tareas de adaptación a un entorno de producción y la integración en infraestructuras computacionales completas no son triviales. Hasta lo que conocemos, PUNCH (Purdue University Network Computing Hubs) es el primer, y hasta la fecha el único, sistema computacional basado en web que está diseñado para soportar herramientas arbitrarias y es utilizado en de forma regular en un entorno real [14][15], pero el sistema PUNCH no opera sobre un Grid.

Los Grids, como una infraestructura de manejo de recursos en Internet, deberían permitir nuevas aplicaciones y actividades de aprendizaje. El grupo de investigación que me ha dado la oportunidad de participar en el desarrollo del proyecto ULabGrid2, ya propuso como punto de partida el proyecto ULabGrid [10] como un nuevo framework que permite a los educadores diseñar “laboratorios colaborativos a distancia para estudiantes” basados en la utilización de infraestructuras Grid. A continuación se describe el prototipo que fue desarrollado presentando los resultados de dicho proyecto.

El proyecto ULabGrid consistió en la implementación y evaluación de una aplicación educacional de computación intensiva en una infraestructura grid [16]. Además, esta aplicación fue utilizada en una clase de laboratorio de aero-navegación con las consiguientes respuestas e interacciones en tiempo real.

En este estudio, diseñamos e implementamos una aplicación de aprendizaje colaborativo a distancia (CSCL) que utiliza una infraestructura grid para aprovechar las ventajas de sus recursos. Los usuarios hacia los que está enfocada la aplicación, son estudiantes aún no graduados. El sistema permitirá a los estudiantes ejecutar las herramientas necesarias para desarrollar las clases prácticas vía internet y proveerá servicios de sistema operativo para los recursos en red (provee archivos transparentes al usuario, funciones de administración de procesos y recursos, manejando la seguridad y control de acceso a través de muchos dominios administrativos, y administrando el estado de la información). Los usuarios serán capaces de ejecutar herramientas remotamente desde cualquier lugar y en cualquier momento, utilizando sus propios ordenadores.

Arquitectura del Framework ULabGrid

La infraestructura ULabGrid fue desarrollada utilizando el estándar de facto provisto por Globus Toolkit 2 (GT2) [17]. El pilar de la administración de recursos conocido como Globus Resource Allocation Manager (GRAM) provee soporte para: Asignación de recursos, delegación de trabajo y administración de progreso y estado de trabajos. El pilar de servicios de información conocido como Index Service provee soporte para descubrimiento. La información provista por

estos servicios con la consideración del tiempo de ejecución estimado y los retrasos de cola serán utilizados para programar los ejercicios de laboratorio sobre los recursos disponibles.

Los servicios de administración de datos del Grid FTP serán utilizados para administrar la transferencia de datos entre nodos, mientras que el Grid Security Infraestructure (GSI) ayudará a garantizar el acceso seguro a los recursos.

La infraestructura que se está desarrollando consiste esencialmente en cuatro componentes principales:

- Usuario: usuario final, básicamente el cliente del sistema.
- Colección de recursos: Colección de nodos de cómputo/almacenamiento que componen el Grid.
- SW Administrador de sesión: Parte encargada de administrar cada cliente en curso (incluyendo la configuración y las peticiones de cambios de estado de sesión por parte del cliente). Consiste en un portal web mediante el cual se permite comenzar los procesos de simulación en máquinas remotas, monitorizaciones de ejecuciones y devolución de resultados a los estudiantes.
- SW Administrador de grid: Administra los recursos del grid (incluyendo recursos de descubrimiento, transferencia de ficheros, estado de los trabajos, etc.) Este componente debe proveer una manera eficiente de organizar todos los recursos conectados y será responsable de la elección del mejor recurso disponible dadas las demandas estáticas (requerimientos de usuario) y los datos dinámicos (carga de la red, carga de la máquina, etc.).

Ambos componentes han sido diseñados para soportar herramientas gráficas interactivas (como el simulador de redes NS-NAM). La naturaleza de este tipo de herramientas (interactividad, rápido tiempo de respuesta, interfaces gráficas...) dirige el diseño de ambos componentes. Desde el punto de vista de el SW administrador de recursos, necesitamos saber constantemente el estado de los recursos (carga de usuarios, carga de las máquinas...) para decidir que recurso (máquina) es el mejor para correr una nueva simulación. Desde el punto de vista del SW administrador de sesión necesitamos un sistema remoto que permita visualizar un entorno computacional “de escritorio” no solamente en la propia máquina en la que está corriendo, si no desde cualquier lugar con acceso a Internet donde esté ubicada la máquina del usuario.

- Como se muestra en la Figura 2, el trabajo funcional de los SW administradores de sesión y de grid se describe como sigue:

1. El usuario se conecta al servidor en el cual está corriendo el SW administrador de sesión, realizando una petición de nueva conexión y dando información sobre sí mismo como el nombre de la aplicación que quiere ejecutar, identificador del grupo al que pertenece (si pertenece a alguno) y el identificador del terminal desde el que está trabajando.
2. El SW administrador de sesión, después de la autenticación del usuario, solicita al SW administrador de grid los recursos computacionales solicitados y le pasa el identificador de usuario.
3. Antes de comenzar la ejecución remota, el SW administrador de grid transfiere todos los archivos del usuario vía servicios Globus GSI-FTP a la máquina remota seleccionada. Con el nombre de la aplicación y el nombre de la máquina remota el SW administrador de grid lanza un proceso de servidor VNC y la aplicación en la máquina remota utilizando para ello los servicios ofrecidos por GRAM.
4. El usuario inicia el cliente VNC en su propia máquina para comenzar la sesión en la máquina remota.
5. Durante la sesión, el usuario puede realizar peticiones al SW administrador de sesión

para

- Hacerle saber sobre otros usuarios del mismo grupo, los cuales están trabajando con la aplicación.
- Poder ser únicamente un estudiante pasivo (sin capacidad de control sobre la aplicación)
- Poder ser un estudiante activo (ser capaz de tomar el control de la aplicación y colaborar)

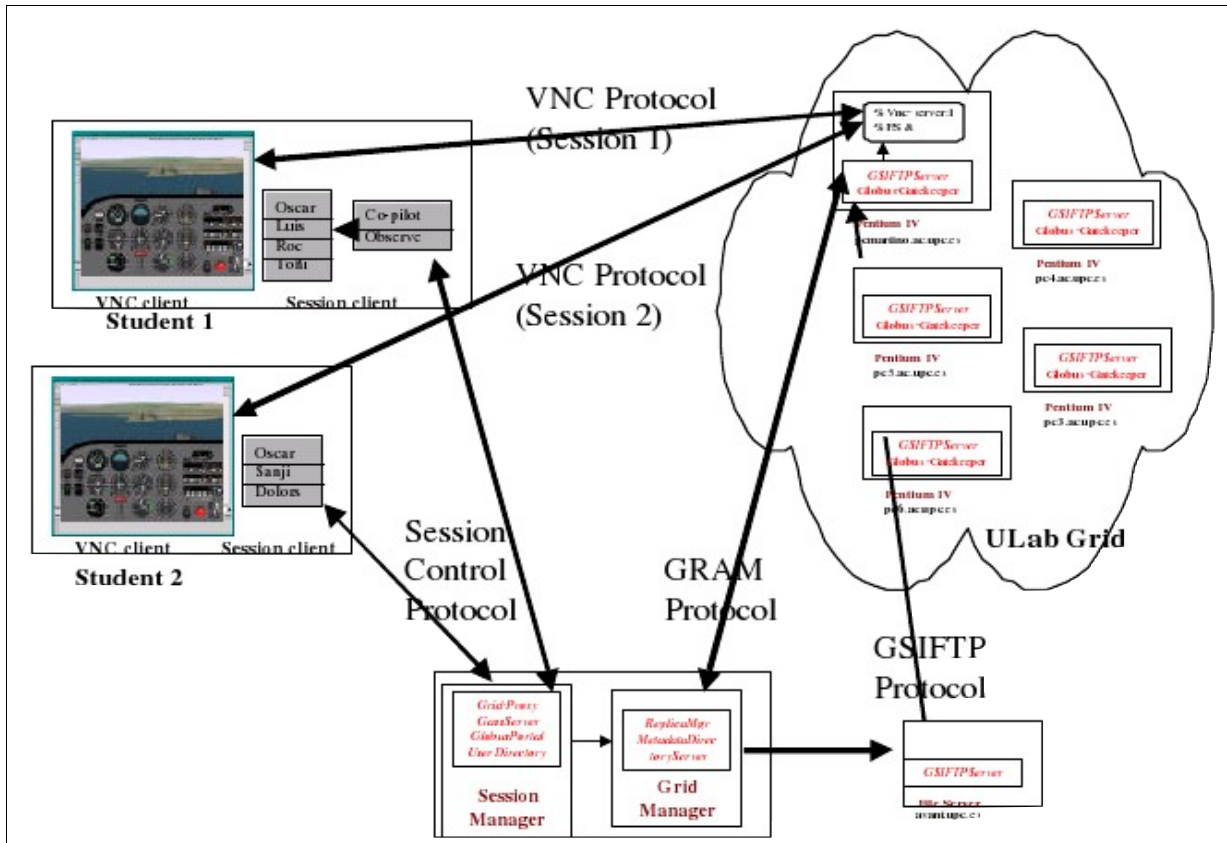


Figura 2. Arquitectura del Framework ULabGrid

En esta sección se ha descrito la arquitectura ULabGrid y la aplicación de colaboración para aprendizaje a distancia que el grupo de investigación implementó sobre ella. Hemos mostrado que para cursos prácticos que tienen requerimientos especiales como gran potencia computacional, acceso a instrumentos y datos muy especializados y colaboración entre usuarios de diferentes instituciones, una infraestructura grid puede ser considerada como una solución viable.

Los resultados preliminares con el software de simulación de vuelos mostraron la necesidad de un estudio futuro en el área de algoritmos de manejo de recursos para aplicaciones interactivas y computacionalmente cambiantes que corren en un grid. También se comenzó a construir un simulador grid utilizando el entorno de desarrollo de aplicaciones J-Sim [18] que también fue utilizado para investigar diferentes estrategias de manejo de recursos.

2.2 Otros trabajos relacionados

Comenzaremos esta sección enumerando aplicaciones orientadas a la colaboración que también utilicen infraestructuras Grid u otras infraestructuras que permitan el acceso remoto a recursos compartidos:

PUNCH

Hasta donde conocemos, PUNCH (Purdue University Network Computing Hubs) es el primero y, hasta la fecha, el único sistema tecnológico basado en web que está diseñado para dar acceso remoto a recursos compartidos para soportar la ejecución arbitraria de herramientas y que es utilizada regularmente en un entorno universitario [14][15]. Sin embargo, el sistema PUNCH ha sido construido sobre una herramienta no estática y no provee acceso interactivo a aplicaciones.

Telescopio World-Wide

Según nuestro conocimiento, hay muy pocas iniciativas que intenten utilizar las tecnologías grid para propósitos educativos. Hay algunos proyectos que implementan un grid para permitir a los estudiantes el acceso a instrumentos remotos y conjuntos de datos reales. El Telescopio World-Wide ha sido diseñado [19] con un objetivo en mente. Es una herramienta educativa que permite a los estudiantes de todo el mundo acceder a los datos almacenados por un telescopio de estudio astronómico, aunque no ha sido implementado utilizando tecnologías grid.

DataGrid

DataGrid [20], un grid que provee alta energía a físicos con PetaBytes de datos extraídos del Gran Colisionador de Hadrones, siendo dicho que permite a los estudiantes acceder al mismo conjunto de datos y herramientas que los científicos.

También existen otras herramientas orientadas a la colaboración que no utilizan infraestructuras Grid, pero que también están relacionadas con este proyecto, a continuación detallo algunas de estas herramienta genéricas:

Grupos de discusión (foro)

En Internet, un foro, también conocido como foro de mensajes, foro de opinión o foro de discusión, es una aplicación web que da soporte a discusiones u opiniones en línea [6].

Listas de correo

Las listas de correo electrónico son un uso especial del correo electrónico que permite la distribución masiva de información entre múltiples usuarios de Internet a la misma vez. En una lista de correo se escribe un correo a la dirección de la lista (ejemplo: silet@correo.org) y le llega masivamente a todas las personas inscritas en la lista, dependiendo de como esté configurada la lista

de correo el receptor podrá o no tener la posibilidad de enviar correos [6].

Campus Virtuales

Un campus virtual se refiere a un servicio ofrecido en línea por un colegio o universidad mediante el cual se permite que el trabajo académico sea completado, parcial o completamente, en online. A menudo contando con la asistencia de profesores u otro personal docente. [8]

Videoconferencias

Videoconferencia es la comunicación simultánea bidireccional de audio y vídeo, permitiendo mantener reuniones con grupos de personas situadas en lugares alejados entre sí. Adicionalmente, pueden ofrecerse facilidades telemáticas o de otro tipo como el intercambio de informaciones gráficas, imágenes fijadas, transmisión de ficheros desde el pc, etc. [6].

A continuación detallo algunas aplicaciones comerciales orientadas hacia el trabajo colaborativo:

Sironta

Sironta es una herramienta P2P de software colaborativo para el intercambio, creación y edición de documentos que requieren trabajo en grupo [6] [21].

eGroupWare

eGroupWare es una solución de trabajo en grupo vía web, de código abierto. Incluye un calendario, una libreta de direcciones, un gestor de contactos, un cliente de correo electrónico IMAP, un InfoLog, funciones de CRM, un gestor de proyectos, un gestor de recursos, un gestor de ficheros, una plantilla de tiempos, un wiki, una base de conocimiento y un motor de flujos de trabajo. [6] [22].

BSCW (*Basic Support for Cooperative Work, o Be Smart- Cooperate Worldwide*)

BSCW es una herramienta informática de tipo software colaborativo o groupware cuyo objetivo es facilitar el trabajo en grupo mediante el uso únicamente de un navegador web y de una conexión a Internet. BSCW permite, entre otras cosas, almacenar y organizar la información, compartirla con otros usuarios, disponer de agendas comunes, mantener debates, etc. Proporciona, en definitiva, una serie de utilidades de apoyo a la colaboración o a la cooperación [6][23].

También existen otros como OpenGroupWare [24], Kolab [25], TalkAndWrite [26], Synergeia [27], Open-Xchange [28], etc.

Finalmente, otro tipo de herramientas relacionadas con este proyecto son las herramientas orientadas al entorno educativo:

MOODLE

Moodle [29] se define como una plataforma e-learning social: "La filosofía de Moodle incluye una enfoque constructivo y social de la educación, enfatizando que los educadores (y no sólo profesores) pueden contribuir a la experiencia educativa de muchas maneras. Las características de Moodle reflejan esto en varios aspectos, tales como hacer posible que los estudiantes hagan comentarios sobre las entradas de una base de datos (o incluso contribuyen a las entradas ellos mismos), o trabajen de manera colaborativa en una wiki " [30].

Elgg

La plataforma de aprendizaje personal Elgg [31], ha sido construida reutilizando muchas plataformas Web 2.0 como blogs, wikis y el sistema de redes sociales para apoyar el aprendizaje en un entorno de red.

LionShare

Los sistemas peer-to-peer son los sistema más descentralizados que utilizan recursos localizados en el borde de la red: equipo del usuario final. Los sistemas peer-to-peer soportan las diversas las aplicaciones de intercambio de archivos para computación distribuida de alto rendimiento, sin embargo, hay muchos aspectos tecnológicos involucrados en dicho sistema. Hay unos cuantos trabajos que tratar de utilizar estas tecnologías para el aprendizaje colaborativo. Halm et. al. [32] discute que al modelo técnico Peer-to-peer se ajusta mejor al paradigma del "diseño del aprendizaje". Su propio grupo ha desarrollado una aplicación peer-to-peer para compartir archivos en uso universitario, "LionShare" [33].

Edutella

Edutella [34] es otra aplicación de un sistema Peer-to-peer para compartir información destino a usos educativos.

The QuarkNet/Grid Collaborative Learning e-Lab

En el The QuarkNet/Grid Collaborative Learning e-Lab [35] los estudiantes dieron datos experimentales para su análisis y acceso por cualquier otro estudiante para su propio análisis.

PlanetLab

La plataforma distribuida para el despliegue de servicios PlanetLab se ha utilizado en un numerosos cursos para brindar a los estudiantes acceso una amplia cobertura de servicios [36].

3. Objeto del Proyecto

En esta sección se enumeran y detallan los principales objetivos del proyecto ULabGrid2. Se comienza por los objetivos generales para pasar a un posterior análisis de objetivos específicos de las dos direcciones del proyecto: su integración con el CMS MOODLE por una parte y con la red social FaceBook por otra.

3.1 Objeto general del proyecto

El objeto de este proyecto es extender las funcionalidades del proyecto ULabGrid explicado en la sección anterior, de manera que se le añadan funcionalidades como la integración con gestores de contenidos y/o redes sociales.

De este modo el proyecto inicial ULabGrid ha dado lugar a dos proyectos que partiendo de una misma base y teniendo un objeto común (trabajo colaborativo), también tienen objetos específicos diferentes destinados a ofrecer funcionalidades distintas como se explicará a continuación.

Así pues, se detallará el objeto de, por una parte, ULabGrid2 integrado con el gestor de contenidos MOODLE, y por otra parte, ULabGrid2 integrado con la red social FaceBook [7].

Escenarios objetivo

A continuación presentamos tres escenarios que se pretenden alcanzar con este proyecto inherentes tanto al aprendizaje Peer-to-Peer (CSCL) que proclama, como se ha explicado anteriormente, que los peers son estudiantes y profesores al mismo tiempo “aprendiendo con y de cada uno”, como al trabajo colaborativo fuera del aprendizaje (groupware).

a) Comunicación directa entre peers

Muchas actividades de aprendizaje o colaboración requieren que los peers se comuniquen con los otros peers, síncrona o asíncronamente, para pedir información, dar consejos, discutir nuevos temas, etc. También es requerida una comunicación dentro del grupo cuando el mensaje no está dirigido a un peer específico, pero varios peers pueden responder. El soporte por ordenador de la comunicación directa peer-to-peer tiene muchas formas: mensajes de texto, streaming de audio y vídeo, etc.

b) Descubrimiento de peers y conexión con otros peers

En un sistema no trivial, con más de unos pocos usuarios, es requerido algún mecanismo de descubrimiento y de selección de los peers a los que conectarse. También el criterio de descubrimiento y selección varía dependiendo de la aplicación. En un escenario de aprendizaje Peer-to-Peer o de colaboración, el descubrimiento basado en red parece ser adecuado ya que también conlleva una reputación del peer [37].

c) Compartir acceso a sesiones de aplicaciones

Las sesiones pueden ser compartidas con diferentes propósitos: Tareas prácticas y/o ejercicios de simulación basados en ordenador, solicitar sesiones de aplicación para ser ejecutadas e interfaces gráficas de usuario para interactuar y visualizar resultados de simulación, etc. Mediante el acceso de peers a sesiones de aplicación de otros peers, los primeros pueden observar el progreso de los segundos en la realización del ejercicio, e incluso ayudarles tomando el control de la interfaz de aplicación. Este escenario requiere un sistema que permita acceso concurrente de múltiples usuarios a la aplicación. Este requerimiento implica un interfaz multiusuario donde las salidas gráficas son mostradas en bastantes pantallas y se comparte las entradas de control (normalmente ratón y teclado).

Objetivos inherentes a la utilización de infraestructuras grid

De acuerdo a la definición, “Un grid es una infraestructura hardware y software a gran escala, distribuida geográficamente y compuesta por redes de recursos heterogéneas, compartida por múltiples organizaciones que están coordinadas para proveer soporte de cómputo transparente, fiable, omnipresente y consistente a un amplio rango de aplicaciones” [38]. A continuación se detallan los objetivos en cuanto a funcionalidad, que se pretenden alcanzar utilizando para ello una infraestructura grid, como es ULabGrid2.

- Instalación y mantenimiento de software

Sobre una infraestructura grid, el software será accesible por los usuarios sin necesidad de perder tiempo descargando, instalando y configurando sus propias máquinas.

- Licencias

A menudo, el software no es de libre distribución y se necesitan licencias para su utilización. Una infraestructura grid podría permitir el acceso a software privativo sin necesidad de comprar una licencia por parte del usuario, y sin incurrir en ninguna acción ilegal, ya que el software al cual se accede puede ser ejecutado con licencia en otro ordenador diferente al del usuario.

- Falta de recursos hardware

A veces, el software solo puede ser ejecutado en máquinas específicas o muy potentes de manera que el usuario no puede utilizarlo en su casa. En un grid estas máquinas específicas pertenecerían al propio grid y el estudiante únicamente necesitaría acceder a el remotamente.

- El Hardware y Software distribuidos geográficamente a gran escala pueden ser compartidos por múltiples organizaciones administrativas

Utilizando una infraestructura grid, los usuarios podrían colaborar con otros usuarios que estén trabajando con las mismas herramientas en diferentes lugares.

- Fácil acceso

ULabGrid2 permite el acceso al software instalado computadores remotos mediante un navegador web o una aplicación de escritorio.

3.2 ULabGrid2 integrado con el CMS MOODLE

El Aprendizaje Colaborativo Soportado por Ordenador (CSCL) se ha beneficiado de las tecnologías Web y de Internet en general, creando herramientas y aplicaciones que fomentan el método educativo de “aprendizaje por discusión”.

Las tecnologías Grid son tecnologías de compartición de recursos que permiten a los estudiantes (y profesores) acceder a los recursos de muchas formas diferentes. Un método educativo que debería tomar ventaja de estas tecnologías es el conocido como “aprender haciendo” [39]. Con este método los estudiantes aprenden mientras y después del proceso de realización de un proyecto, una composición, una herramienta, un experimento, un artefacto, etc. La puesta en práctica este método tiene un requisito fundamental: Acceso a un número determinado de recursos, y este acceso puede ser provisto por un grid. Un grid puede proveer acceso a conjuntos de datos, recursos computacionales e instrumentaciones remotas. Estos recursos pueden ser compartidos y solicitados bajo demanda por el periodo de duración de una actividad de “aprender haciendo”.

Como ya hemos comentado, las aplicaciones educativas pueden aprovechar las ventajas de las tecnologías grid. Primero, se podría permitir el acceso de estudiantes a instrumentaciones remotas y datos para experimentos de laboratorio utilizando instrumentaciones y datos reales. Segundo, se puede proveer acceso a herramientas y aplicaciones con requerimientos computacionales no abordables por muchos centros. Tercero, se puede permitir la colaboración entre estudiantes de diferentes instituciones. Cuarto, se puede habilitar la creación de cursos virtuales y clases entre diferentes organizaciones.

Los estudiantes pueden acceder a instrumentaciones remotas únicamente disponibles en laboratorios científicos, empresas orientadas a la investigación y centros de desarrollo debido a su alto coste. De este modo, los estudiantes pueden utilizar las últimas tecnologías para adquirir habilidades demandadas por las compañías, y practicar con las técnicas más actuales. Los estudiantes pueden también hacer uso de conjuntos de datos reales obtenidos por científicos desde sus últimos experimentos, y pueden reproducir experimentos y comparar sus conclusiones con las conclusiones científicas.

Algunos tipos de aplicaciones colaborativas que realizan un uso intensivo de recursos son: aplicaciones interactivas que requieren un gran número de recursos computacionales en tiempo real, simulaciones interactivas, mundos virtuales interactivos y experimentos interactivos. Las aplicaciones no interactivas pueden requerir un gran capacidad de almacenamiento para datos históricos temporales como los eventos durante cualquier periodo de tiempo. Algunos ejemplos de aplicaciones educativas interactivas con un alto uso de recursos computacionales son los simuladores visuales, simuladores de sistemas complejos (como los simuladores de redes), aplicaciones de vídeo, aplicaciones de realidad virtual, etc.

Finalmente, un grid permite compartir recursos entre diferentes escuelas, de modo que permite a los profesores preparar actividades de aprendizaje aprovechando las ventajas de un gran número de recursos de aprendizaje disponibles de bastantes instituciones. Estos recursos compartidos podrían llevarnos a una disposición formal entre instituciones, de diseñar y desarrollar cursos virtuales inter-escolares.

A continuación se detallan algunos objetivos específicos del proyecto ULabGrid2 integrado con el CMS MOODLE.

- Compartir información

Los peers aprendices deben ser capaces de acceder a recursos de información de otros peers

como por ejemplo, tareas asignadas, notas de clase, resultados de experimentos de laboratorio, etc. Los peers utilizarán tal información para comparar con sus propios resultados y discutir las posibles diferencias. Este escenario requiere un sistema que permita a los peers compartir sus archivos de datos con los demás.

– Reducir costes en el cambio de contenidos (hardware y software)

Actualizar el software educativo de un laboratorio (actualizar versiones, nuevas aplicaciones, etc.) incluye un alto coste para el estudiante. Las versiones nuevas son más caras que las antiguas y muchas veces requieren más recursos computacionales. Los cambios y actualizaciones de el software son una vez más gestionados por el administrador liberando al estudiante de realizar esta tarea desde casa.

– Conseguir resultados homogéneos

Otro problema resuelto si el software del laboratorio reside en el grid es el hecho de que se garantiza que el resultado de los ejercicios realizados por todos los estudiantes proviene de un entorno único. Esto permitirá realizar fácilmente la tarea de detección de errores, tanto para estudiantes como para profesores.

– Computación transparente

Los estudiantes no tienen que perder tiempo averiguando como utilizar un software en particular y podrían concentrar todos sus esfuerzos en el proceso de aprendizaje. Por ejemplo, ellos podrían programar en lenguaje C independientemente de la herramienta de programación. Esto mejorará la curva de aprendizaje del estudiante.

– Otras ventajas inherentes al aprendizaje a distancia ofrecido por ULabGrid:

a) Los estudiantes son capaces de familiarizarse con el entorno de ejecución de la herramienta antes de empezar sus prácticas, consultando manuales o los ejemplos provistos por el propio sistema

b) los estudiantes son capaces de realizar los ejercicios desde casa sin necesidad de la presencia física en los centros educativos.

3.3 ULabGrid2 integrado con la red social FaceBook

Mientras que ULabGrid2 integrado con el CMS MOODLE da lugar a un software orientado hacia CSCL, la integración con la red social FaceBook [7] da lugar a un software más orientado hacia el trabajo colaborativo fuera del ámbito universitario (groupware), además de brindar la posibilidad de compartir sesiones de aplicación no orientadas hacia el trabajo colaborativo como se explicará en las líneas siguientes.

A continuación se detallan algunos objetivos específicos de ULabGrid2 integrado con FaceBook:

- Mayor alcance

Uno de los objetivos específicos principales es compartir sesiones de aplicación con un público más amplio ya que las características de las redes sociales permiten tener un alcance mayor que utilizando un gestor de contenidos.

- Mayor difusión

La integración de la red social FaceBook con ULabGrid2 permite que sea más fácil hacer conocer y utilizar la aplicación por muchas más personas, principalmente a partir de las conexiones entre amigos, que además proporcionan una difusión de la aplicación más confiable por parte de los nuevos usuarios. Además, debido a esta característica de FaceBook se puede implementar la funcionalidad de diferenciar si una determinada sesión a la que queremos acceder, está siendo lanzada desde el ordenador de un usuario conocido o de un usuario desconocido.

- Descentralización total

Al contrario que en el caso anterior, en el que se necesita de un servidor (o varios) que alberguen al gestor de contenidos que se integra con ULabGrid2, la red social FaceBook permite alcanzar el objetivo de conseguir un sistema completamente descentralizado, un sistema completamente P2P. Esto evita tener un punto central de fallo.

- Más posibilidades de encontrar nuevos peers con intereses comunes

Del mismo modo que la naturaleza de las redes sociales permiten una mejor difusión de las aplicaciones que las utilizan, también permiten a sus usuarios encontrar otros usuarios o peers con los mismos intereses, que trabajan en la misma materia, etc.

- Más posibilidades de encontrar nuevas aplicaciones

ULabGrid2 integrado con FaceBook permite a sus usuarios encontrar nuevas aplicaciones, de una forma sencilla, que antes les eran desconocidas. Al contrario que en ULabGrid2 integrado con MOODLE mediante el cual los usuarios acceden a cursos virtuales sabiendo que aplicación van a utilizar, en el caso de ULabGrid2 integrado con FaceBook podrán encontrar en cada conexión aplicaciones compartidas por los usuarios en cada momento, por lo que tiene un comportamiento más aleatorio, aunque manteniendo el acceso a Grids encargados de servir aplicaciones de una forma más estable.

– Acceso sesiones individuales

Otro objetivo específico de ULabGrid2 integrado con FaceBook es lograr un software P2P de compartición de sesiones “uno a uno”. Es decir, un software que permita a los usuarios el acceso individual a sesiones de aplicación, además del acceso concurrente típico de un sistema colaborativo. De este modo, al contrario que en el caso anterior, el usuario puede acceder individualmente a sesiones de aplicación sin la necesidad de pertenecer a un grupo de usuarios.

– Servir sesiones propias

Mientras en el caso anterior los usuarios siempre acceden a las sesiones servidas desde un grid compuesto por servidores lanzados por centros educativos, administrativos, etc., en este caso los propios usuarios son los que comparten sesiones de aplicación lanzadas desde su propio ordenador.

4. Desarrollo

En esta sección se explican las fases de análisis, diseño e implementación de los prototipos desarrollados, incluyendo información referente a la arquitectura del sistema, tecnologías utilizadas, funcionalidad, interfaces de usuario, problemas encontrados y fases de ejecución del proyecto.

4.1 ULabGrid2 integrado con el CMS MOODLE

4.1.1 Framework ULabGrid2 + MOODLE

El framework a construir se basa en lo expuesto en la sección 3, objeto del proyecto. Varios de esos escenarios podían ser apoyados por las diferentes tecnologías de distribución; sin embargo, había un escenario que sólo era provisto por el framework ULabGrid [40], ULabGrid proporciona las siguientes funcionalidades para varios de los escenarios anteriores:

a) Compartir sesiones de aplicación utilizando un sistema de pantalla compartida: el sistema Virtual Network Computing VNC [41] permite compartir el escritorio remoto entre varios usuarios. ULabGrid permite a los usuarios acceder a diferentes sesiones de aplicaciones compartidas que se ejecutan bajo demanda en un “pool” dinámico de recursos Grid.

b) Compartir información utilizando un sistema de archivos compartidos: en las aplicaciones de ULabGrid se disponía de datos de cualquiera de las sesiones de las aplicaciones que se ejecutan en el grupo de recursos grid.

Para soportar el resto de escenarios de aprendizaje peer-to-peer hemos ampliado el framework en la nueva versión ULabGrid2. Las funcionalidades que esta nueva versión del framework incorpora son:

c) Comunicación directa peer-to-peer , Inicialmente mensajes de texto.

d) Descubrimiento de peers por medio de mecanismos basados en red.

El marco resultante, ULabGrid2 + MOODLE, se muestra en la Figura 3. Los estudiantes (peers) forman una red conectando pares entre sí. Los peers son conectados directamente por comunicación o intercambio de archivos, e indirectamente a través de una recursos del grid orientados al intercambio de sesiones de aplicación. Un mecanismo del gestor de contenidos permite a los peers descubrirse entre sí a través de sus conexiones directas, con quién me comunico, o a través de conexiones indirectas, con quién comparto sesiones de aplicación.

En el gráfico podemos observar varios elementos:

- Usuarios o peers
- Recursos (VNC + App)
- Gestor de contenidos MOODLE con el módulo ULabGrid2 integrado
- Servidor NFS
- Grid

y varios tipos de relaciones entre ellos definidas por el color de las líneas:

- Rojas: Conexiones con MOODLE (login, acceso al curso, acceso al módulo, etc.)
- Azules: Son de dos tipos. Entre usuarios, ya sea por pertenencia al mismo grupo o por comunicación directa mediante chat. El otro tipo se refiere a las conexiones entre el servidor que aloja al gestor de contenidos y módulo, el servidor NFS y los recursos del Grid (servidores VNC y aplicaciones servidas)
- Verdes: Conexiones a las diferentes sesiones de aplicación mediante VNC

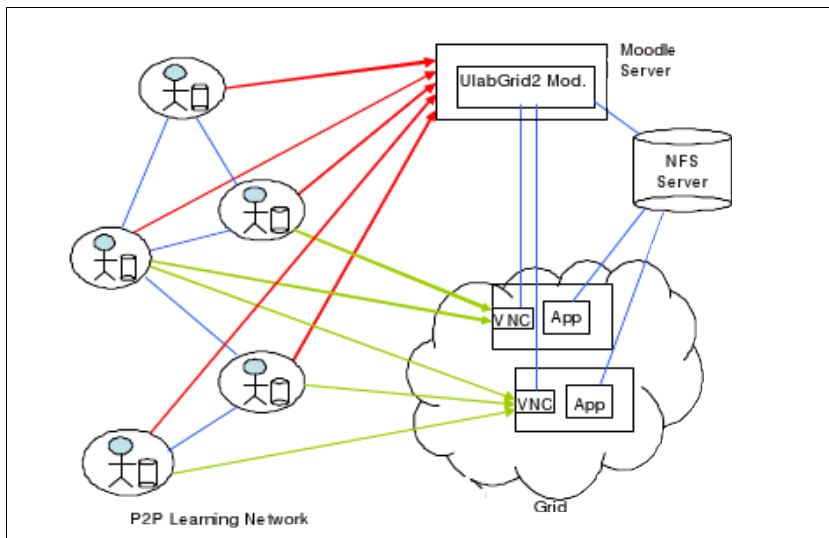


Figura 3. Framework ULabGrid2 + MOODLE

4.1.2 Prototipo ULabGrid2 + MOODLE

MOODLE es un gestor de contenidos que se organiza en cursos en los cuales se pueden añadir actividades que permiten la utilización de sus diferentes módulos (chat, foro, test, etc.).

Hemos implementado el prototipo de ULabGrid2 como un módulo que se encarga de permitir el acceso de los usuarios a las aplicaciones servidas en el Grid de tal forma que cada curso tendrá asignada una aplicación. Es decir, el módulo que hemos desarrollado dará acceso a una aplicación u otra dependiendo del curso en el que se haya añadido.

Los usuarios estarán organizados en grupos para poder trabajar de forma colaborativa. Así, cuando un usuario accede a un curso en el que se ha añadido una actividad que enlaza con el módulo ULabGrid2, al pulsar sobre esta actividad, el sistema comprueba si existe algún usuario de su mismo grupo conectado, es decir, comprueba si su grupo ya tiene asignados recursos dentro del Grid. Si hay algún compañero conectado, el usuario que acaba de acceder es conectado, utilizando para ello VNC (cuya función original es el acceso a escritorios remotos), a la misma sesión de aplicación para que puedan compartir su control. Si no hay compañeros utilizando ULabGrid2, se buscan recursos libres dentro del Grid y son asignados al grupo del usuario que se acaba de conectar para comenzar a utilizar la aplicación que corresponda. Pero además, el sistema también permite acceder a los recursos del Grid asignados a otro grupo, por ejemplo, para visualizar la sesión de aplicación de otro grupo si éste la comparte.

La disponibilidad de los recursos del Grid varía con el tiempo según las interacciones de los usuarios, es decir, que un mismo grupo puede ser conectado a recursos diferentes dentro del Grid en ocasiones diferentes. Sin embargo, los ficheros que han sido generados desde la propia aplicación en una máquina concreta del Grid, deben estar disponibles para el grupo aunque este se conecte a otra máquina. Por este motivo, cada máquina dentro del Grid dispone de directorios que enlazan a los directorios remotos servidos desde el servidor NFS. De esta manera, aunque se acceda a una aplicación determinada en máquinas diferentes, los ficheros generados pueden ser guardados en el

mismo directorio de forma transparente al usuario.

Pero no sólo se accede al servidor NFS desde las máquinas del Grid, si no que el servidor web en el que está alojado MOODLE también tiene permisos para acceder a los directorio remotos con el objetivo de poder gestionar estos directorios desde el propio módulo ULabGrid2, desde el cual cada grupo podrá manejar su propio directorio de grupo y compartir ficheros con el resto de grupos del mismo curso utilizando para ello un directorio común.

Además, cabe destacar que el sistema también permite la comunicación directa entre los usuarios por medio de un chat, añadido a ULabGrid2 a partir del módulo chat ya disponible para MOODLE.

A continuación se detalla con más detenimiento, y con la utilización de imágenes de diferentes interfaces del sistema, los escenarios expuestos en el punto anterior.

Compartir sesiones de aplicación en ULabGrid2

Los alumnos (Peers) del mismo curso ejecutan aplicaciones en equipos remotos a través de un módulo de Moodle llamado módulo ULabGrid2 en la figura 3. Los equipos remotos tienen instalada la la licencia de las aplicaciones y un servidor VNC. El módulo de Moodle corre la aplicación y un cliente VNC que se conecta al servidor para acceder a la sesión.

La figura 4 muestra la interfaz de vista por un usuario. El usuario accede a la sesión a través del cliente VNC ejecutado en el navegador Web. Las aplicaciones a distancia tienen una interfaz gráfica de usuario que permite a éste realizar la actividad arrastrando o haciendo doble clic en los iconos e introduciendo mensajes de texto.

Los peers en el mismo grupo de estudiantes comparten la misma sesión de forma automática. Inicialmente sólo un estudiante del grupo tiene el control de la sesión (ratón, teclado, etc) y los demás están en modo de solo visualización. El resto de compañeros del grupo pueden solicitar el control en cualquier momento y el compañero que está controlando la sesión puede ceder el control o rechazar la petición. Los peers que controlan la interfaz de sesiones son mostrados con una estrella.

Cada grupo de estudiantes decide si sus sesiones se hacen públicas o no a otros grupos. La Figura 5 muestra el menú de control de un usuario, a la izquierda se encuentra el panel "Mi ULG sesión" que muestra mi grupo y el botón "compartir control". Si la sesión se hace pública cualquier otro grupo del mismo curso puede tener acceso a ella, pero sólo en modo de solo visualización de modo que no se les permite tomar o pedir al control. Hay 3 posibles estados para compartir: Sesión no compartida, sesión compartida y sesión libre compartida. En el estado sesión compartida un estudiante puede ver las sesiones de los demás, pero para ello debe de haber compartido sus sesiones con los demás. El estado de sesión compartida libre representa que una sesión de usuario que puede ser vista por cualquier persona sin tener que compartir sus sesiones. Este modo de funcionamiento puede ser utilizado por los usuarios que necesitan ayuda y están dispuestos a mostrar su trabajo libremente a cualquier persona. A la izquierda de la Figura 5 se muestran los iconos del panel de control con los iconos que representan estos estados y los botones para cambiar a otros estados.

Compartir datos entre peers

Los estudiantes (peers) no sólo comparten sesiones, sino también datos y archivos de programa. Los peers de un mismo grupo comparten el mismo directorio privado y todos los archivos del directorio están a disposición de cualquier peer perteneciente al grupo. Los peers de un

grupo no tienen permitido acceder al directorio privado de otros grupos. Los archivos se pueden hacer públicos o no en cualquier momento con un simple clic. Cuando un peer quiere que un archivo se haga público, éste es copiado en un directorio común del sistema que es accesible por todos los grupos. El propietario del grupo del permisos de administración completos sobre el archivo, de este modo siempre tiene la opción de borrar el archivo del directorio común, mientras que los otros grupos sólo tienen acceso de lectura. Los otros también grupos pueden copiar archivos del directorio común a su directorio privado con el fin de tener pleno acceso.

Las sesiones de aplicación en equipos remotos tienen pleno acceso a los directorios privados correspondientes a través de NFS pero no al directorio común.

Comunicación directa entre peers en UlabGrid2

La comunicación entre compañeros es una característica muy importante del sistema, no sólo dentro de un grupo de peers, si no entre compañeros de diferentes grupos. En cualquier ventana de sesión de aplicación, como en la figura 4, hay una lista de pares conectados disponible, cada usuario puede ver que usuarios de su grupo están activos (en amarillo), que usuarios de otros grupos están conectados (en verde), y que usuarios sólo participarán en la sesión de chat (en rojo). Todos los usuarios pueden escribir en la zona de chat para solicitar o responder preguntas con respecto a la sesión de la aplicación que se visualiza, también se mantiene un registro de los accesos.

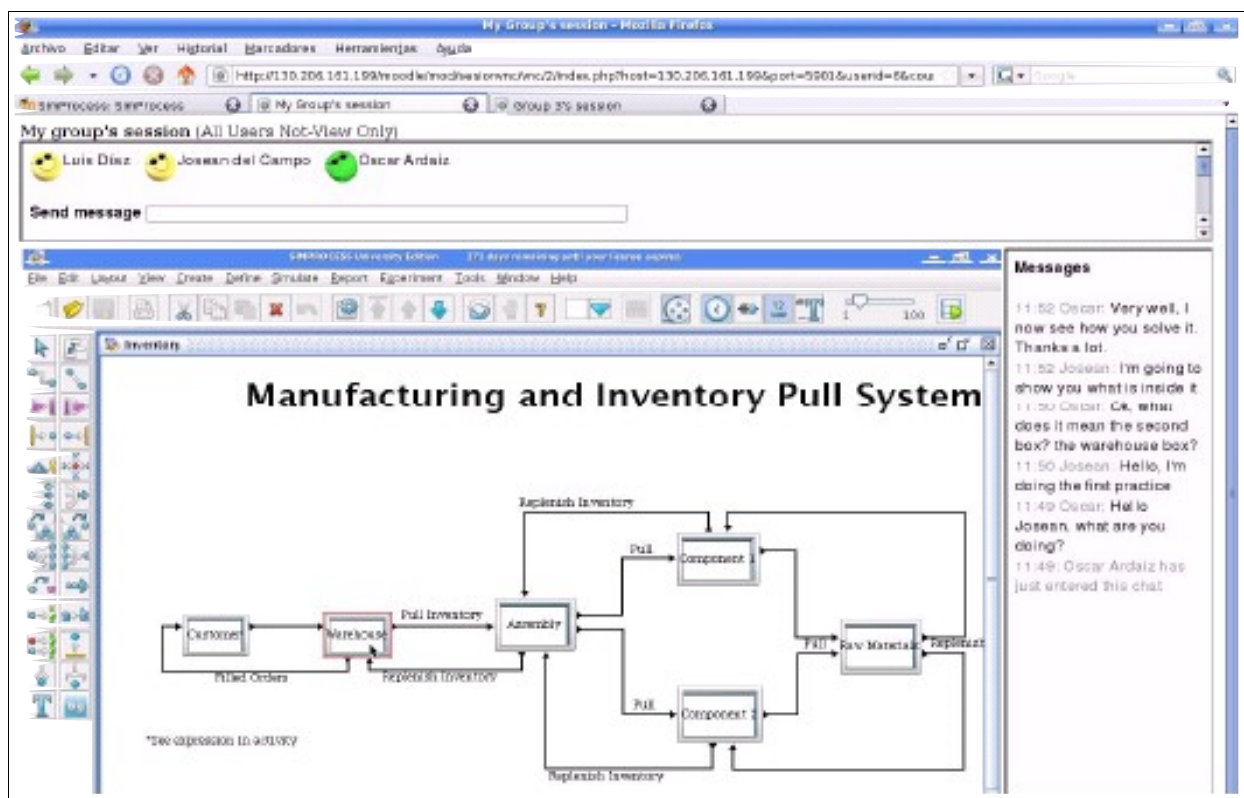


Figura 4. Interfaz Web: sesión de aplicación compartida y chat

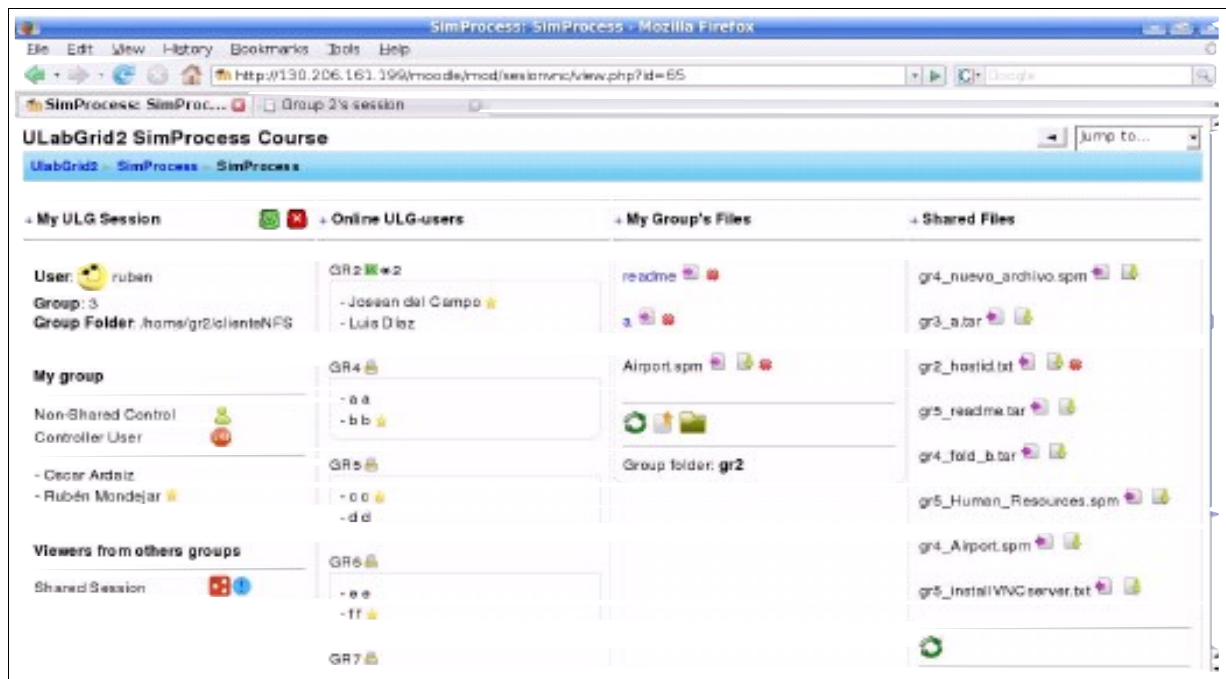


Figura 5. Panel de control de un usuario del “Grupo 3”: Este grupo sabe que nadie está visualizando sus sesión. (En la segunda columna “Online ULG-users” se puede observar que el grupo 2 está compartiendo su sesión y que tiene dos “viewers”)

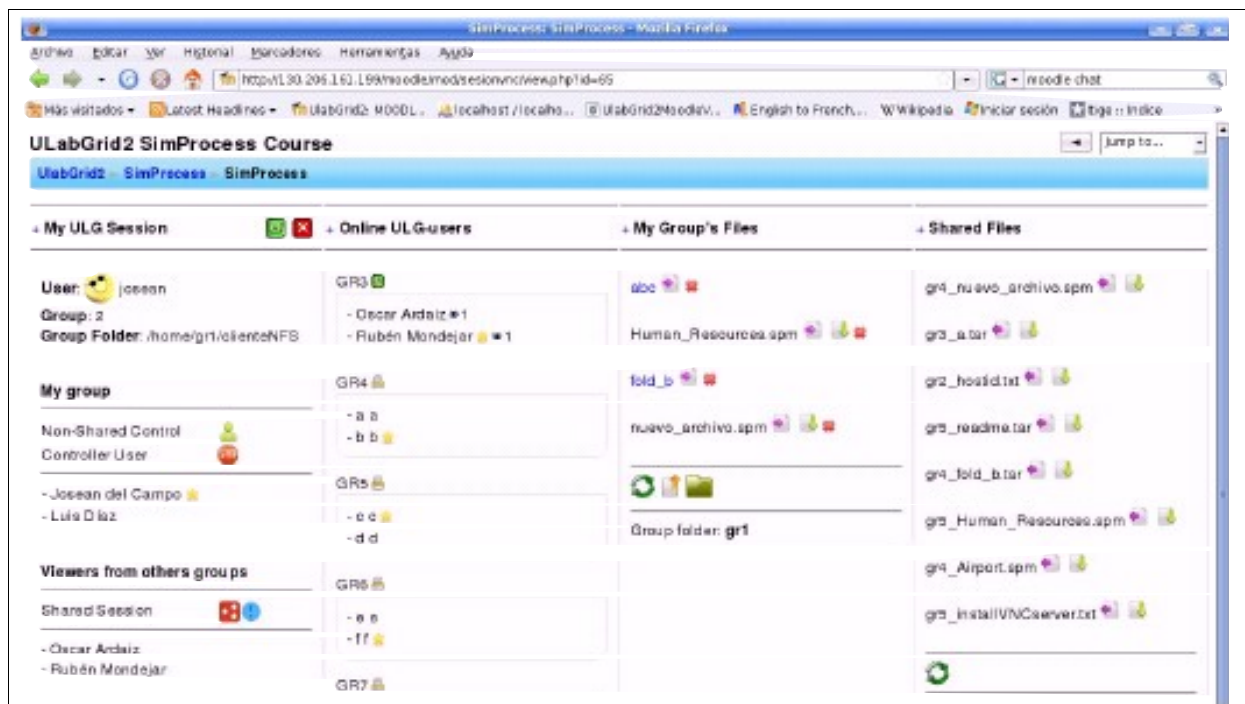
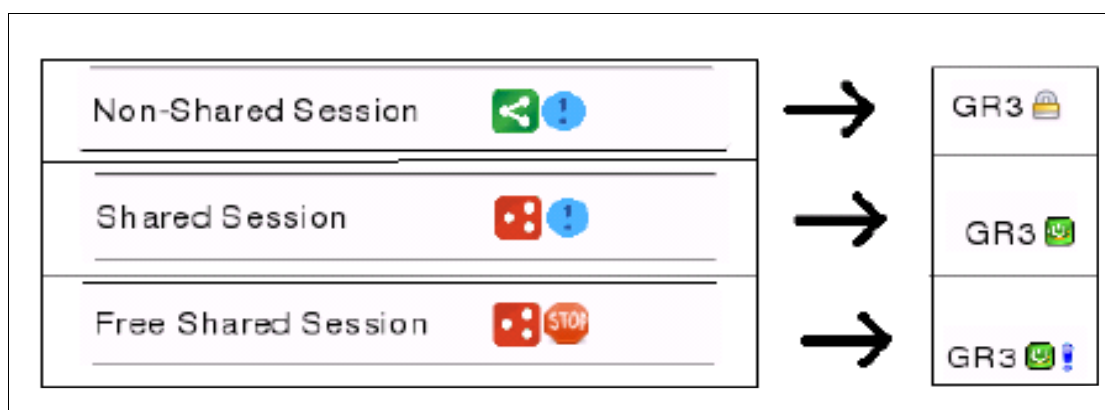


Figura 6. Panel de control de un usuario del “Grupo 2”: este grupo sabe que hay dos usuarios externos visualizando su sesión, es mostrado en la esquina de abajo a la izquierda (También puede observar en la segunda columna “Online ULG-users” que el grupo 3 está compartiendo su sesión, que no tiene “viewers”, y que cada miembro del grupo 3 está observando una sesión de otro grupo)



*Figura 7. Opciones de estado en el Panel de control de usuario (izquierda),
Como ven estos estados los otros usuarios (derecha)*

Descubrimiento de peers en ULabGrid2

El descubrimiento de peers en ULabGrid2 es facilitado por los indicadores de estado. Estos indicadores tienen la misión de informar a la comunidad sobre las acciones de los compañeros. Hemos incluido información en tiempo real sobre el inicio o cierre de sesión o cierre por parte de los compañeros y sesiones y archivos compartidos.

Los peers son capaces de saber quién está conectado en el curso al mismo tiempo. En las figuras 5 y 6 la segunda columna "Online ULG-users" muestra todos los grupos accediendo al mismo curso en el que nos encontramos, y que miembros de cada grupo están conectados. El usuario puede ver el estado de cada grupo: Sesión no compartida, sesión compartida o sesión libre compartida, a la derecha de la figura 7.

Cada grupo puede ver quién está accediendo a sus sesiones en el apartado "viewers from other groups". En la figura 5, el grupo no tiene ningún usuario visualizando su sesión de aplicación, en la figura 6 el grupo tiene dos usuarios visualizando su sesión de aplicación.

La columna "Online ULG-users" también proporciona información acerca de cuántos usuarios de otros grupos están visualizando sesiones que no son propiedad de su propio grupo, esto es indicado por el icono del ojo y el número que aparecen a la derecha de un grupo (en la figura 5 se ven dos usuarios del grupo 2, en la figura 6 ningún usuario está viendo al grupo 3). Y a la derecha de cada usuario, otro icono de ojo y el número indica a cuántos otros grupos está visualizando el usuario (en la figura 5 ningún usuario de otro grupo está visualizando otros grupos, en la figura 6 ambos usuarios del grupo 3 están visualizando un grupo).

El directorio común contiene los archivos comunes compartidos e información sobre los propietarios de los archivos. Esta información es útil para descubrir otros peers con intereses comunes o que trabajan en temas relacionados.

A continuación se detalla el procedimiento de configuración del módulo y de acceso.

Configuración del módulo

La página de configuración permite tener un control centralizado sobre los diferentes elementos que se interconectan en ULabGrid2 y únicamente el administrador del sitio MOODLE puede acceder a ella.

Desde esta página se pueden configurar los siguientes parámetros del sistema, necesarios para el correcto funcionamiento de ULabGrid2:

- Conexiones SSH a las máquinas: Es necesario especificar la ubicación de la clave RSA dentro del servidor web, ya que esta clave es utilizada para verificar que quien intenta arrancar/parar los servidores VNC alojados en el Grid es el propio servidor web y no otra máquina externa al sistema. Cada servidor VNC tiene almacenada la misma clave RSA de manera que cuando el servidor web intenta acceder a un servidor VNC las claves de cada una de las dos máquinas son comparadas.

SSH CONNECTIONS	
Path RSA key:	<input type="text" value="/home/josean/cron/josean-rsync-key"/>

Figura 8. Clave RSA para las conexiones SSH

- Aplicaciones servidas: Es necesario especificar que aplicaciones están instaladas en los servidores VNC, su ubicación y el número de grupos MOODLE que la podrán utilizar.

SERVED APPLICATIONS			
app	path	nusers	actions
<input type="text" value="SimProcess"/>	<input type="text" value="/bin/SIMPROCESS"/>	<input type="text" value="2"/>	Save / Delete
<input type="text" value="Eclipse"/>	<input type="text" value="/bin/eclipse"/>	<input type="text" value="10"/>	Save / Delete

Figura 9. Aplicaciones servidas

- Servidores VNC: Es necesario especificar la IP de cada servidor VNC alojado en el Grid, que aplicación/es sirve cada uno y a través de que puerto.

VNC SERVERS			
app	ip	port	actions
<input type="text" value="Eclipse"/>	<input type="text" value="172.16.29.27"/>	<input type="text" value="5901"/>	Save / Delete
<input type="text" value="Eclipse"/>	<input type="text" value="172.16.29.28"/>	<input type="text" value="5901"/>	Save / Delete
<input type="text" value="SimProcess"/>	<input type="text" value="130.206.164.33"/>	<input type="text" value="5902"/>	Save / Delete
<input type="text" value="Eclipse"/>	<input type="text" value="172.16.29.28"/>	<input type="text" value="5903"/>	Save / Delete
<input type="text" value="SimProcess"/>	<input type="text" value="130.206.164.33"/>	<input type="text" value="5903"/>	Save / Delete
<input type="text" value="Seleccionar App"/>	<input type="text"/>	<input type="text"/>	New

Figura 10. Configuración de los servidores VNC

- Servidor NFS: Es necesario especificar su IP, la ruta de los directorios servidos y si son compartidos por todos los grupos de un mismo curso o son independientes de cada grupo.

NFS SERVER

IP NFS:

app	local_folder	served_folder	common	actions
SimProcess	clienteNFS	shared	<input type="checkbox"/>	Save / Delete
Eclipse	clienteNFS	shared	<input type="checkbox"/>	Save / Delete
SimProcess	commonDir	/home/josean/SimProce	<input type="checkbox"/>	Save / Delete
Eclipse	commonDir	/home/josean/Repast	<input type="checkbox"/>	Save / Delete

User owner of common directories:

Figura 11. Configuración del servidor NFS

- Usuario Linux: Es necesario especificar el prefijo de los usuarios Linux propietarios de las cuentas de los equipos que lanzarán a través de VNC las aplicaciones y asignarlos a los grupos MOODLE existentes. Por ejemplo, si el prefijo especificado es 'gr' y se han definido 3 grupos MOODLE para utilizar una determinada aplicación, cada servidor VNC que sirva esa aplicación tendrá los usuarios gr1, gr2 y gr3 y cada grupo MOODLE accede a estas máquinas siempre con el mismo usuario Linux con el fin de diferenciar su directorio personal.

SESSIONVNC GROUPS

Prefix linux users:

Asignar grupos

SimProcess

group	user
19	2
2	1

Eclipse

group	user
10	5
9	4
8	3

Figura 12. Asociación de usuarios linux con grupos moodle

La gestión de la configuración implica el alta, borrado, modificación y listado de los datos que ha sido detallados y deberá estar definida correctamente antes de ser salvada, en caso contrario, el módulo no tendrá un funcionamiento correcto.

Acceso al módulo

Para acceder al módulo, este deberá ser instalado como tal y añadido a uno o varios cursos disponibles en el sitio MOODLE.

Cuando un usuario trate de acceder al módulo, el sistema comprobará si el usuario pertenece a un grupo con permisos para acceder a sesionvnc en el curso actual.

En caso afirmativo, buscará la IP de un servidor VNC disponible y comprobará su conectividad. Si la comprobación es positiva, lanzará una sesión VNC con la aplicación solicitada y en caso contrario, seguirá buscando servidores VNC cuya IP sea alcanzable.

Una vez asignada una IP con conectividad a Internet se sirve al cliente el applet de Java que permite la conexión a través de TightVNC para que el cliente pueda utilizar el SW requerido (en nuestro caso SimProcess o Eclipse + Repast Symphony).

4.1.3 Tecnologías utilizadas

En este proyecto se dispone de un Grid de servidores VNC de modo que si falla alguno de ellos se proporcionarán los mecanismos necesarios para que automáticamente se busque otro servidor que pueda ser utilizado, haciendo uso de SSH y SCP para lograr este objetivo.

Además también serán necesarios otros servidores como Apache para mostrar el interfaz del sistema al usuario vía web y NFS para dotar a los usuarios de capacidad de almacenamiento remoto y distribuido.

El acceso se realiza mediante navegador web, y las tecnologías web utilizadas son: HTML, JavaScript, AJAX (DHTML) y un applet de Java en el lado de cliente y PHP + MySQL en el lado del servidor.

En la primera fase del proyecto se hicieron las siguientes elecciones:

- Elección de Sistema Operativo: Kubuntu 8.04

Las otras opciones que se probaron (también distribuciones de Linux) fueron, Fedora 9 y Kubuntu 6.

Entre Kubuntu y Fedora se eligió Kubuntu ya que deriva de Ubuntu y Ubuntu de Debian mientras que Fedora deriva de Red Hat, siendo Debian la distribución más popular de Linux, y por consiguiente la más documentada.

- Elección de servidor web: Apache 2

Entre Microsoft Windows Server y servidor Apache, para trabajar sobre un sistema operativo Linux, elegimos el servidor Apache, por compatibilidad y por utilizar GPL.

- Elección de Gestor de contenidos para red social: MOODLE

El gestor de contenidos es una parte fundamental del sistema ya que se encarga de mantener información sobre el estado de conexión de los usuarios, los recursos del Grid disponibles o asignados a cada grupo, las aplicaciones que serán servidas desde el Grid, los directorios remotos utilizados, las salas de chat y sus participantes, el estado de las sesiones de cada grupo (compartida, no compartida, etc.) además de alojar las páginas web a las que los usuarios acceden para conectarse al Grid e interactuar con el resto de usuarios.

Ante el problema de como crear las redes sociales nos vimos ante la cuestión de que gestor elegir: MOODLE, Joomla, utilizar FaceBook...

La elección final fue MOODLE por ser un gestor de contenidos orientado al aprendizaje muy extendido, y que además permite un desarrollo modular de componentes.

- Elección de servidor VNC: TightVNC

Las primeras pruebas de escritorio remoto las hicimos con el escritorio remoto de Kubuntu 8.04 para después realizar pruebas con Xtightvnc y Tightvnc, comprobando que tiene un mejor funcionamiento el segundo, por tanto Tightvnc fue el VNC server elegido, que además permite ser servido a través del interfaz web mediante un applet Java de código abierto, que podremos modificar según sea conveniente.

- Elección de sistema de almacenamiento: NFS

Básicamente, las opciones más importantes que teníamos eran NFS y Samba y la elección de NFS fue debida a su buen funcionamiento bajo Linux.

- Elección de aplicaciones a compartir: Eclipse + Repast Symphony y SimProcess

Las aplicaciones elegidas son instaladas en las máquinas del Grid para ser servidas mediante VNC a los usuarios que acceden a ULabGrid2 a través de MOODLE.

SimProcess es un software de simulación y modelado que facilita el análisis del comportamiento de los diferentes procesos de un sistema o negocio (Business Process Simulation) elegido para compartir porque es el SW que actualmente se utiliza en la asignatura Sistemas Informáticos (Dpto: de Automática y Computación; Área: Arquitectura y Tecnología de Computadores) de la UPNA.

Eclipse, es un entorno de desarrollo integrado (IDE) en código libre que permite desarrollar proyectos software generalmente en el lenguaje Java pero al que se le puede añadir plugins para desarrollar otro tipo de proyectos. Se eligió esta aplicación y el plugin de Repast Symphony fue elegido por ser un potente SW de simulación bajo GPL.

También se realizaron pruebas con la aplicación de simulación de biomedicina Taverna 1.7.

Tras la elección del SW a utilizar se pasó a la instalación de este en los dos ordenadores disponibles:

- Utilizando dos máquinas (Kubuntu 8.04)
- Instalación de Apache 2 + MOODLE en máquina 1
- Instalación de Tight VNC en máquina 1 y máquina 2
- Instalación de NFS server en máquina 2
- Instalación de aplicaciones Eclipse + Repast Symphony y SimProcess en máquinas 1 y 2

4.1.4 Implementación

Árbol de directorios

Se detalla a continuación el árbol de directorios completo que compone el módulo desarrollado y alojado en el servidor Apache, explicando el propósito de cada script utilizado.

Como se puede comprobar, existen varios tipos de scripts como PHP (.php), HTML (.html), Shell Scripts (.sh), JavaScript (.js) y XML (.xml), además de ejecutables (.jar) y clases (.class) Java.

También existe un directorio encargado de alojar las imágenes utilizadas en el desarrollo del módulo por considerarse poco relevantes en el mismo.

- **sesionvnc**: directorio principal
 - **chat**: directorio de los componentes del chat
 - **actPart.php**: actualiza los participantes en las sesiones chat
 - **blank.php**: inicializa un elemento div que forma parte de las ventanas de chat
 - **blank2.php**: ídem
 - **chat.php**: genera la página web para chats sin visor VNC
 - **index.php**: gestiona cada sesión chat
 - **lib.php**: almacena las funciones necesarias para el módulo chat
 - **messages.php**: genera el elemento web con los mensajes de una sesión chat
 - **participants.php**: genera el elemento web con los participantes de una sesión chat
 - **sendMessage.php**: introduce los mensajes a enviar en la base de datos
 - **db**: directorio referente a la base de datos
 - **install.xml**: contiene información referente a las nuevas tablas que son creadas en la base de datos al instalar el módulo sesionvnc.
 - **images**: directorio para almacenar las imágenes utilizadas
 - **ask.png, back.png, etc.**
 - **lang**: directorio para funciones de idiomas
 - **en_utf8**: subdirectorio
 - **sesionvnc.php**: contiene información necesaria para la instalación del módulo
 - **vnc**: directorio para almacenar el applet java del visor VNC
 - **1..12**: 12 directorios replicados que contienen la misma información (se necesitan directorios replicados para evitar problemas al servir varias instancias del applet de TightVNC)
 - **index.php**: genera la web de acceso a sesiones propias. Muestra el applet y los componentes del chat.
 - **index2.php**: genera la web de acceso a sesiones ajenas. Muestra el applet y los componentes del chat.
 - ***.class**: clases javas de TightVNC viewer.
 - **VncViewer.jar**: Applet java TightVNC viewer.
 - **actualizaShareSession.php**: modifica el atributo “shared” de una sesión y muestra quién está visualizando mis sesiones
 - **actualizaVNCmygroup.php**: permite mostrar los usuarios conectados de un grupo y modificar el atributo “control” de éste.
 - **actualizaVNCusers.php**: permite mostrar los usuarios conectados grupos diferentes a los del usuario actual.
 - **asignaGrupos.php**: asocia grupos Moodle y usuarios Linux. Además, crea directorios para clientes NFS en máquinas remotas y en la máquina local (servidor Apache)
 - **borrar_dir.php**: elimina un directorio de “My Group Files”
 - **borrar_file.php**: elimina un fichero de “My Group Files” o “Shared Files”
 - **cambiarControl.php**: comprueba si es necesario realizar cambios de control en un grupo y permite realizarlos
 - **checkConnected.php**: comprueba si un determinado usuario está conectado y utilizando un determinado curso
 - **checkControl.php**: comprueba el atributo “control” para una sesión definida por una dirección IP y un puerto

- **checkSession.php**: comprueba si los atributos que hacen referencia a curso, usuario, grupo, IP y puerto son correctos
- **compartir_dir.php**: comprime un directorio y copia el archivo resultante en el directorio especificado
- **compartir_dir.sh**: shell script encargado de llevar a cabo la tarea ordenada por `compartir_dir.php`
- **compartir_file.php**: copia un fichero de un directorio a otro
- **config.html**: genera la web de configuración del módulo `sesionvnc`
- **conseguirControl.php**: cambia el atributo “connected” del actual usuario y curso al valor 4
- **creaxstartup.sh**: genera archivos de configuración de servidor VNC
- **deleteApp.php**: elimina una aplicación que se está sirviendo
- **deletedir.sh**: shell script encargado de eliminar un directorio
- **deleteNFS.php**: elimina un registro de la tabla `sesionvnc_nfs`
- **deleteVNCdesktop.php**: elimina un registro de la tabla `sesionvnc_desktops`
- **disconnectvnc.php**: cambia el estado de usuarios que se desconectan, de servidores que dejan de estar ocupados y lanza `pararvnc.sh` parando el servidor VNC.
- **download.php**: permite descargar un fichero
- **editApp.php**: permite modificar los datos introducidos de aplicaciones a lanzar mediante VNC
- **editNFS.php**: permite modificar datos referentes al servidor NFS
- **editVNCdesktop.php**: permite modificar datos referentes a los servidores VNC
- **eliminarEnBD.php**: elimina datos de un usuario que se desconecta de una sesión de otro usuario
- **estoyEnBD.php**: comprueba si un usuario está conectado a una determinada sesión
- **funciones.js**: almacena funciones javascript que son llamadas desde archivos PHP
- **index.php**: lista todas las instancias del módulo `sesionvnc` en un curso determinado
- **infoVNCuser.php**: muestra información sobre un usuario determinado
- **iniciavnc.sh**: copia archivo de configuración al servidor VNC y lo lanza
- **insertarEnBDchat.php**: añade un nuevo usuario a la tabla de una sesión de chat en la base de datos
- **insertarEnBD.php**: añade un nuevo usuario a la tabla referente a una sesión VNC en la base de datos
- **lib.php**: librerías necesarias para el funcionamiento del módulo
- **logBorrar.php**: añade una nueva entrada en el archivo “eventos.log” cuando un usuario elimina un fichero o directorio
- **logCompartir.php**: añade una nueva entrada en el archivo “eventos.log” cuando un usuario transfiere un fichero o directorio entre “My Group Files” y “Sahed Files” o viceversa.
- **logDisconnectChatOnly.php**: añade una nueva entrada en “eventos.log” cuando un usuario se desconecta de una sesión de chat (sin TigthVNC viewer)
- **logDisconnectMySession.php**: añade una nueva entrada en “eventos.log” cuando un usuario se desconecta de su propia sesión
- **logDisconnectOtherSession.php**: añade una nueva entrada en “eventos.log” cuando un usuario se desconecta una sesión ajena

- **logMkdir.php**: añade una nueva entrada en “eventos.log” cuando un usuario crea un nuevo directorio
- **logMySession.php**: añade una nueva entrada en “eventos.log” cuando un usuario se conecta a su propia sesión
- **logOtherSession.php**: añade una nueva entrada en “eventos.log” cuando un usuario se conecta a una sesión ajena
- **mkdir.php**: crea un directorio en “My Group Files”
- **nameVNCuser.php**: muestra información referente al usuario en la página principal
- **negarControl.php**: cambia el valor del atributo “connected” de un determinado usuario de 2 a 5
- **newApp.php**: inserta un nuevo registro en la tabla “sesionvnc_app” que representa una nueva aplicación a compartir
- **newNFS.php**: inserta un nuevo registro en la tabla “sesionvnc_nfs” que representa el servidor NFS utilizado
- **newVNCdesktop.php**: inserta un nuevo registro en la tabla “sesionvnc_desktops” que almacena los servidores VNC utilizados en el sistema
- **noPideControl.php**: cambia el valor del atributo “connected” de un determinado usuario de 2, 3 o 4 a 1
- **paravnc.sh**: shell script encargado de lanzar la orden de sistema operativo que para el servidor VNC que ha sido lanzado
- **pasarControl.php**: cambia el valor del atributo “connected” del usuario actual de 4 a 1 y el del compañero del usuario actual de 2 a 4
- **pideControl.php**: cambia el valor del atributo “connected” del usuario actual de 1 a 2
- **read_dir.php**: permite mostrar vía web, los ficheros contenidos en un directorio
- **shared_control.php**: modifica el atributo “control” en la tabla “sesionvnc_desktops”
- **shared_session.php**: modifica el atributo “shared” en la tabla “sesionvnc_desktops”
- **updatedir.sh**: shell script encargado de eliminar el contenido de un directorio y crearlo de nuevo
- **upload.php**: permite cargar un fichero en “My Group Files”
- **version.php**: define la versión del módulo
- **view.php**: genera la web que supone la interfaz principal de la aplicación
- **xtartup**: contiene la estructura básica de un archivo de configuración para servidor VNC. Es utilizado para crear cada archivo específico según la aplicación a compartir

Base de datos

Moodle es un sistema de gestión de cursos desarrollado en PHP que opera sobre una base de datos de un amplio tamaño. La versión de Moodle que hemos utilizado cuenta con 185 tablas en la base de datos a las que hemos añadido 8 nuevas tablas utilizadas por el módulo desarrollado que se detallan a continuación.

Nota: En todas las tablas mostradas a continuación, el atributo “id” se incrementa automáticamente en una unidad cada vez que se inserta un nuevo registro en la tabla, y supone además la clave primaria para cada una de estas tablas.

mdl_sesionvnc

Esta tabla almacena un identificador (id), el curso (course, referencia a la tabla mdl_course) al que se asocia cada instancia del módulo sesionvnc y finalmente su nombre (name).

Campo	Tipo
id	bigint(10)
course	bigint(10)
name	varchar(255)

Figura 13. Tabla mdl_sesionvnc

mdl_sesionvnc_app

Esta tabla almacena los datos referentes a cada aplicación que se comparte mediante el módulo: su identificador (id), el nombre de la aplicación (app), la ruta donde se almacena en el servidor VNC (path) y el número de usuarios que la utilizarán (nusers).

Campo	Tipo
id	bigint(10)
app	varchar(255)
path	varchar(255)
nusers	bigint(10)

Figura 14. Tabla mdl_sesionvnc_app

mdl_sesionvnc_chat

Esta tabla almacena un identificador (id) y asocia un elemento chat (chatid, hace referencia a la tabla mdl_chat) a cada grupo de usuarios (groupid, hace referencia a la tabla mdl_groups).

Campo	Tipo
id	bigint(10)
groupid	tinyint(2)
chatid	tinyint(2)

Figura 15. Tabla mdl_sesionvnc_chat

mdl_sesionvnc_desktops

Esta tabla representa los servidores VNC utilizados a través del módulo. Para ello almacena un identificador (id), la aplicación lanzada (app, hace referencia a mdl_sesionvnc_app), su dirección IP (ip), su puerto (port), el grupo que lo está utilizando si lo está utilizando alguno (groupid), información sobre como se está compartiendo (shared) y que usuario tiene el control (control).

Campo	Tipo
id	bigint(10)
app	varchar(255)
ip	varchar(255)
port	smallint(4)
groupid	tinyint(2)
shared	tinyint(1)
control	tinyint(1)

Figura 16. Tabla mdl_sesionvnc_desktops

mdl_sesionvnc_groups

Esta tabla almacena información sobre los usuarios Linux (userid) asociados a cada grupo de usuarios moodle (groupid, hace referencia a la tabla mdl_groups), además de un identificador (id)

Campo	Tipo
id	bigint(10)
groupid	tinyint(2)
userid	tinyint(2)

Figura 17. Tabla mdl_sesionvnc_groups

mdl_sesionvnc_nfs

Almacena información referente a los directorios utilizados para la utilización del servicio NFS. Así, se almacena un identificador (id), la aplicación a la que está destinada (app, hace referencia a mdl_sesionvnc_app), el nombre de los directorios en máquina local (local_folder), el nombre de los directorios servidos (served_folder) y si estos son compartidos por los todos los grupos de un curso o individuales a cada grupo (common, 0 indica individual y 1 compartido)

Campo	Tipo
id	bigint(10)
app	varchar(255)
local_folder	varchar(255)
served_folder	varchar(255)
common	tinyint(1)

Figura 18. Tabla mdl_sesionvnc_nfs

mdl_sesionvnc_shared_session

Esta tabla se utiliza para almacenar la información referente a qué usuarios (userid, hace referencia mdl_user) están utilizando la sesión de cada grupo (groupid, hace referencia a mdl_groups), además de almacenar un identificador (id).

Campo	Tipo
id	bigint(10)
groupid	tinyint(2)
userid	tinyint(2)

Figura 19. Tabla mdl_sesionvnc_shared_session

mdl_sesionvnc_user_lastaccess

Esta tabla almacena información sobre los últimos accesos de cada usuario a cada instancia del módulo sesionvnc de cada curso. Para ello se almacena un identificador único (id), de usuario (userid, hace referencia a mdl_user), de curso (courseid, hace referencia a mdl_course), la hora de acceso (timeaccess) y si está o no conectado en el momento actual (connected).

Campo	Tipo
id	bigint(10)
userid	bigint(10)
courseid	bigint(10)
timeaccess	bigint(10)
connected	tinyint(1)

Figura 20. Tabla mdl_sesionvnc_user_lastaccess

4.1.5 Problemas encontrados

Detalle a continuación problemas y dificultades más importantes encontrados a lo largo del desarrollo del proyecto y las soluciones que se tomaron para solucionarlos.

– Aprendizaje de programación de módulos en MOODLE

La utilización de MOODLE ha aportado muchas ventajas al proyecto, pero también ha implicado la obligación del aprendizaje de la sintaxis que deben seguir los archivos de un módulo, las librerías propias a utilizar y el significado de las diferentes tablas de la base de datos. Para ello fue de una gran utilidad, la ayuda de la activa comunidad de desarrolladores de MOODLE que obtuve a partir de sus diferentes foros.

– Windows no sirve sesiones VNC en paralelo.

El grid encargado de servir las sesiones VNC accedidas desde los cursos del CMS MOODLE que incluyen el módulo referente a ULabGrid2 está compuesto por computadores, cada uno de los cuales puede servir una o varias sesiones VNC al mismo tiempo. Estos computadores no pueden servir varias sesiones VNC al mismo tiempo si su sistema operativo es Windows, ya que en este caso, los accesos mediante escritorio remoto siempre tienen el acceso a la sesión del usuario actual, impidiendo de esta forma el acceso de diferentes usuarios al mismo tiempo. Esta fue una de las razones por las que las conexiones a Grid deben ser realizadas a computadores Linux, ya que en este caso, la sesión servida no es necesariamente la misma que la que es visualizada por el usuario de la computadora. Los ordenadores bajo Linux soportan múltiples sesiones X11 lo que permite a su vez, alojar varias conexiones VNC en la misma IP, pero utilizando diferentes puertos.

– Las contraseñas SSH impedían la automatización completa de ULabGrid2

El funcionamiento de ULabGrid2 con MOODLE requiere que algunos recursos alojados en los computadores que forman el grid sean gestionados desde el servidor Apache que aloja MOODLE. Por ejemplo, desde el módulo ULabGrid2 de MOODLE se crean, editan y eliminan directorios de los computadores del grid. Para ello, instalamos servidores SSH en las máquinas que alojan los servidores VNC, pero al realizar las llamadas desde el código de MOODLE no se conseguía el resultado esperado sin interacción humana debido a la necesidad de introducir contraseñas manualmente. Lo mismo ocurría al intentar lanzar o parar servidores VNC en las máquinas, ya que estos no están corriendo continuamente, si no que funcionan bajo demanda. Este problema se hizo extensible al uso de SCP, ya que está basado en SSH, utilizado por ejemplo, para sobrescribir los archivos de configuración de los servidores VNC o para la transferencia de ficheros entre diferentes equipos. Estos problemas se solucionaron utilizando claves RSA que nos permitieron automatizar por completo este tipo de proceso.

– Vulnerabilidad SSH

Tras la implantación del primer prototipo de ULabGrid2 en los dos ordenadores disponibles en ese momento, uno de ellos sufrió un ataque aprovechando una vulnerabilidad en la configuración del servidor SSH. Tras el ataque, dicho ordenador se dedicaba a atacar otras máquinas intentando también, acceder mediante SSH. La solución a esta vulnerabilidad fue aumentar las medidas de seguridad.

Estos son los cambios que se realizaron:

1. Se dejó de utilizar el puerto de SSH por defecto (22)

2. Se restringió el acceso SSH solo a los usuarios autorizados (excluyendo al usuario root)
3. Se restringió el acceso SSH mediante contraseña permitiendo únicamente el acceso con clave RSA
4. Se permitieron las conexiones SSH a cada servidor VNC únicamente desde el servidor Apache.

Tras estos cambios no se volvieron a sufrir este tipo de ataques.

– Cada servidor VNC solo tiene un archivo de configuración

Utilizando un mismo usuario Linux se pueden lanzar varias instancias de un servidor VNC, pero esto presenta un problema, únicamente un archivo de configuración (/home/\$USER/.vnc/xstartup) gestiona que aplicación se servirá a través del servidor. Sin embargo, la naturaleza del proyecto implica lanzar aplicaciones diferentes y la solución a este problema consistió en reconfigurar el archivo inmediatamente antes de lanzar cada instancia de cada servidor VNC. De esta manera, si lanzamos dos servidores y queremos compartir dos aplicaciones A y B, se configura el archivo xstartup para lanzar la aplicación A, se lanza el servidor número uno, se reconfigura el archivo xstartup para lanzar la aplicación B y se lanza el servidor número dos. De este modo, aunque el archivo de configuración haya quedado configurado de tal manera que la aplicación a compartir es la B, el servidor número uno seguirá sirviendo la aplicación A ya que ésta era la que constaba en dicho archivo en el momento del lanzamiento del servidor.

– Conectividad

Desde la pantalla de configuración de ULabGrid2 + MOODLE el administrador indica que servidores VNC constan en el grid, detallando para ello la dirección IP y el puerto de cada uno de ellos. En un principio, cuando un usuario era el primero de su grupo que accedía al sistema se le asignaba el primer recurso libre, de manera que si el servidor asignado no funcionaba se mostraba un mensaje de error en el applet del visor VNC y el grupo no podía acceder a otro servidor que funcionase de manera correcta. Esto se solucionó realizando comprobaciones, utilizando para ello la herramienta Ping, de si la IP y puerto al que el sistema se intentaba conectar era accesible a través de Internet, así, cuando un recurso libre no era accesible se pasaba a la comprobación de la conectividad el siguiente servidor que todavía no estaba siendo utilizado hasta encontrar uno de ellos que permitiese su acceso al grupo. En caso de no encontrar ninguno se mostraba un error por pantalla indicando que la inexistencia de recursos disponibles y evitando el acceso al visor VNC (applet de TightVNC).

– Sistemas de directorios y permisos

La creación del sistema de directorios ha supuesto varios problemas, ya que al principio se comenzó utilizando un usuario Linux por cada aplicación (Eclipse + Repast Symphony y Simprocess) en cada servidor VNC con el objetivo de tener archivos de configuración diferentes, cada uno configurado para lanzar únicamente la aplicación que corresponde sin lanzar el escritorio remoto completo. Tras esto, nos dimos cuenta de que era necesario crear un usuario Linux en cada servidor VNC y en el servidor Apache, por cada grupo de MOODLE registrado en el sistema. Esto es necesario para que cada grupo MOODLE tenga su propio directorio y éste no sea accesible para el resto de grupos. Además, también fue necesario crear grupos Linux incluyendo a cada usuario Linux que se ha comentado anteriormente (apartado “Configuración del módulo”) y al propietario de los directorios comunes a todos los grupos. Otro aspecto a tener en cuenta es que los identificadores UID y GID de cada usuario y grupo deben coincidir en cada máquina para que el sistema NFS funcione correctamente, algo que

ocasionó bastantes problemas al principio de su utilización.

- Restricciones de seguridad del applet Java (VNC viewer) para acceder a sesiones remotas
Un applet de Java es un programa en código Java que es integrado dentro de una página web cargada en el ordenador del cliente que accede a ella. Estos applets tienen fuertes restricciones de seguridad. No pueden acceder fuera de su espacio de almacenamiento; por ejemplo, los applets normalmente se almacenan en un servidor remoto y se ejecutan en el puesto cliente, en este caso no podrán acceder al espacio de almacenamiento (disco) del cliente. Por lo mismo razón un applet local no puede acceder a una base de datos remota. Nuestro caso se asemeja más al último ejemplo, ya que los applets se encuentran alojados en el servidor Apache, se cargan en el ordenador del cliente y se encargan de acceder a el servidor VNC correspondiente dentro del grid, que normalmente va a tener una dirección IP diferente del servidor Apache (No siempre era así en el primer prototipo, ya que realizábamos la instalación de un servidor VNC en la misma máquina en la que estaba instalado el servidor Apache).

La primera solución que se intentó consistía en crear un archivo .policy añadiendo el permiso deseado y acceder al applet desde un appletviewer indicando la utilización del archivo .policy creado. De esta forma se consiguió acceder desde el viewer pero seguía dando problemas el acceso desde el navegador web.

La solución final para este problema consistió en firmar el applet para crear un “trusted applet”, de manera que cuando un cliente lo carga, es mostrado un mensaje con los datos de la firma digital del applet. Una vez el cliente lo acepta, ya tiene permisos para trabajar correctamente.

Ejemplo de error:

“Error: access denied (java.net.SocketPermission 130.206.161.199:5901 connect, resolve)”.

- Mismo applet cargado en diferentes ventanas

Si hay más de una pestaña con el applet del visor VNC, por ejemplo al visualizar la sesión propia y una de un compañero, aparece un problema inherente al hecho de que el código del applet accedido era el mismo para todas las ventanas, provocando que al cerrar una de ellas se perdiese la conexión con el applet en las demás. La solución para este problema consistió en crear réplicas del applet en diferentes carpetas del árbol de directorios del servidor VNC. De este modo, cerrar una ventana con conexión a un determinado applet no ocasiona problemas con otras ventanas de sesión, ya que estas estarán utilizando applets diferentes.

4.1.6 Etapas y plazos de ejecución

En este contexto el ciclo de vida de una aplicación está compuesto por una fase de despliegue y otra fase de funcionamiento donde tendrá que adaptarse continuamente a condiciones ambientales cambiantes y competir por el uso de los recursos. En la primera fase es fundamental que el middleware ayude a tratar el problema de la gestión (buscar IPs de servidores VNC a utilizar, seleccionar una y asignarla) de recursos, en el segundo que proporcione servicios de comunicación (SSH y SCP), almacenamiento (NFS), ejecución de tareas (VNC), distribución de contenidos (Apache + MOODLE) y publicación-suscripción, entre otros servicios.

AÑO 2008

Mayo - Julio

- Análisis
- Elección de Sistema Operativo
- Elección de servidor web
- Elección de Gestor de contenidos para red social
- Elección de servidor VNC
- Elección de sistema de almacenamiento
- Elección de aplicaciones a compartir
- Diseño del sistema

Agosto-diciembre

- Desarrollo de la primera versión del prototipo
- Instalación del SW elegido en las máquinas disponibles.
- Implementación de módulo para MOODLE: sesionvnc

AÑO 2009

Enero-Febrero

- Implantación del prototipo en dos máquinas.
- Evaluación de la primera versión del prototipo para obtener las primeras conclusiones
- Definir los cambios a realizar en el proyecto

Marzo-Mayo

- Implementación de los cambios en la versión definitiva del prototipo
- Implantación del prototipo sobre un Grid real
- Comprobación de su funcionamiento sobre un entorno real (utilización por alumnos)
- Analizar y evaluar los datos tras el uso del prototipo UlabGrid2 por parte de los alumnos.
- Obtención de conclusiones
- Evaluación externa y difusión de los resultados del proyecto

4.2 ULabGrid2 integrado con la red social FaceBook

4.2.1 Framework ULabGrid2 + FaceBook

Al igual que en el framework explicado anteriormente, correspondiente a la integración de ULabGrid2 con MOODLE, esta nueva versión se basa en lo expuesto en la sección 3, objeto del proyecto. ULabGrid proporciona las siguientes funcionalidades para varios de los escenarios que han sido explicados:

a) Compartir sesiones de aplicación utilizando un sistema de pantalla compartida: el sistema Virtual Network Computing VNC [42] permite compartir el escritorio remoto entre varios usuarios. ULabGrid permite a los usuarios acceder a diferentes sesiones de aplicaciones compartidas que se ejecutan bajo demanda en un “pool” dinámico de recursos Grid.

Además, en esta nueva versión del framework, estos recursos también son ejecutados y servidos en las propias máquinas de los usuarios o peers, no únicamente en el “pool” como sucedía en la versión anterior. Esta es la principal diferencia y ventaja que presenta este nuevo framework. Nótese que los componentes referentes a las aplicaciones servidas (VNC + App en el gráfico) ya no aparecen únicamente en el grid, si no que también aparecen asociados a cada usuario.

b) Compartir información: en esta nueva versión no se comparten archivos de la misma manera que en la anterior, mediante NFS, si no que se comparte la información de las sesiones que el propio usuario sirve, utilizando para ello la utilidad Data Store de FaceBook [42], que permite almacenar esta información de una manera descentralizada.

A continuación se enumeran otras funcionalidades soportadas por el framework:

c) Comunicación directa peer-to-peer: Es esta versión del framework aprovechamos las conexiones ya existentes entre los usuarios por medio de las redes sociales, en este caso FaceBook.

d) Descubrimiento de peers por medio de mecanismos basados en red: se utiliza para ello la información almacenada en el Data Store de FaceBook que sirve para controlar que usuarios de la aplicación están conectados en cada momento y que aplicaciones están sirviendo, indicando su IP:puerto. Además se detalla otros tipos de información como la relacionada con que grupos de trabajo se forman entre los peers o de que tipo es cada sesión.

El marco resultante es ULabGrid2 + FaceBook y se muestra en la figura 21. Los usuarios (peers) forman una red conectando pares entre sí. Los peers son conectados directamente, como se ha explicado, por sus asociaciones mediante redes sociales, en este caso FaceBook, e indirectamente a través de una red de recursos, del grid y propios, orientados al intercambio de sesiones de aplicación.

En el gráfico podemos observar varios elementos:

- Usuarios o peers
- Recursos (VNC + App)
- Red social FaceBook + Data Store
- Grid

y varios tipos de relaciones entre ellos definidas por el color de las líneas:

- Rojas: Conexiones con FaceBook (login, información sobre amigos, recursos disponibles, etc.)
- Azules: Conexiones referentes a la red social (Amigos FaceBook)
- Verdes: Conexiones a las diferentes sesiones de aplicación mediante VNC

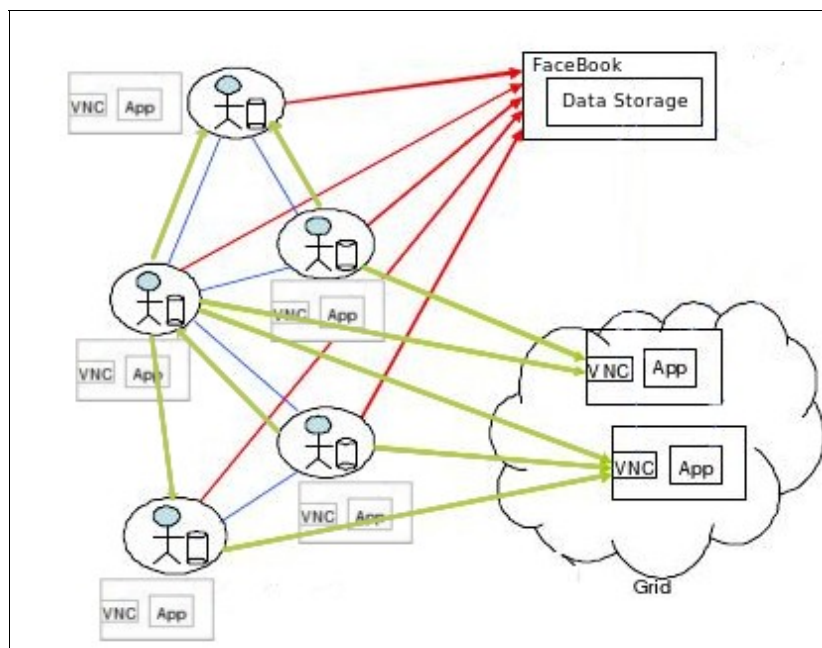


Figura 21. Framework ULabGrid2 + FaceBook

4.2.2 Prototipo ULabGrid2 + FaceBook

FaceBook es una red social que permite el desarrollo de aplicaciones que hagan uso de ella [43] [44]. Así, estas aplicaciones pueden ser orientadas a web o a escritorio. En este caso la aplicación ULabGrid2 que hace uso de FaceBook es una aplicación de escritorio que utiliza la red social para varias funciones que serán explicadas a continuación.

El acceso a las aplicaciones será mucho más libre que en el caso anterior, ya que en este nuevo prototipo, los usuarios no están restringidos por la aplicación compartida en el curso en el que están alistados, si no que a través del Data Store de FaceBook pueden acceder a un rango de recursos mucho menos restringido.

Los usuarios de la aplicación no serán, por defecto, organizados en grupos como en el prototipo anterior, si no que serán usuarios más independientes, aunque seguirán diferenciando al resto de usuarios en dos grandes grupos. Si antes se podían diferenciar a los usuarios entre pertenecientes o no a mi grupo, ahora se pueden diferenciar entre amigos o desconocidos, evidentemente, basándonos en su relación a través de FaceBook. Asimismo, los usuarios de la aplicación podrán crear grupos propios seleccionando los participantes entre sus amigos y poder trabajar así, de forma colaborativa.

Cuando un usuario accede a ULabGrid2, se lanzan en su máquina tantos servidores VNC como sean necesarios para servir las aplicaciones que desea compartir con el resto de la comunidad de usuarios. Al lanzar estas aplicaciones se indica si serán utilizadas por otros usuarios de forma individual, de una manera en la que esos recursos son asignados al primer usuario que accede a ellos, o por el contrario serán utilizadas por otros usuarios de manera colaborativa. En este último caso, como se ha mencionado, el usuario que lanza la sesión es el encargado de crear el grupo de usuarios, amigos suyos en FaceBook, que serán los únicos usuarios con permisos para acceder a dichas sesiones.

Además de las sesiones lanzadas por los usuarios, también existirán sesiones lanzadas en Grid como el utilizado en el prototipo anterior, ULabGrid2 + MOODLE. No se han definido diferentes tipos de sesiones en el Grid. Resumiendo, contamos con tres tipos de sesiones que posteriormente se explicarán en más detalle: individuales, colaborativas y de grid.

La disponibilidad de los recursos varía con el tiempo según el comportamiento de los usuarios, es decir, que esta disponibilidad depende de que los usuarios lancen o no la aplicación ULabGrid2. De este modo, los recursos disponibles en el framework de la aplicación cambian constantemente.

A continuación se detalla con más detenimiento, y con la utilización de imágenes de diferentes interfaces del sistema, los escenarios expuestos en el punto anterior.

Compartir sesiones de aplicación en ULabGrid

Los usuarios (Peers) de la aplicación ejecutan sesiones de aplicación en equipos, remotos o propios, utilizando para ello el visor VNC TightVNC. Los equipos que actúan como servidores de sesiones tienen instalada la licencia de las aplicaciones servidas y un servidor VNC.

La figura 22 muestra la interfaz de vista por un usuario. El usuario accede a la sesión a través del cliente VNC integrado en la aplicación ULabGrid2. Las aplicaciones servidas a distancia, al igual que las de escritorio, tienen una interfaz gráfica de usuario que permite a éste utilizarlas empleando para ello los mecanismos básicos de entrada como son el teclado y el ratón.

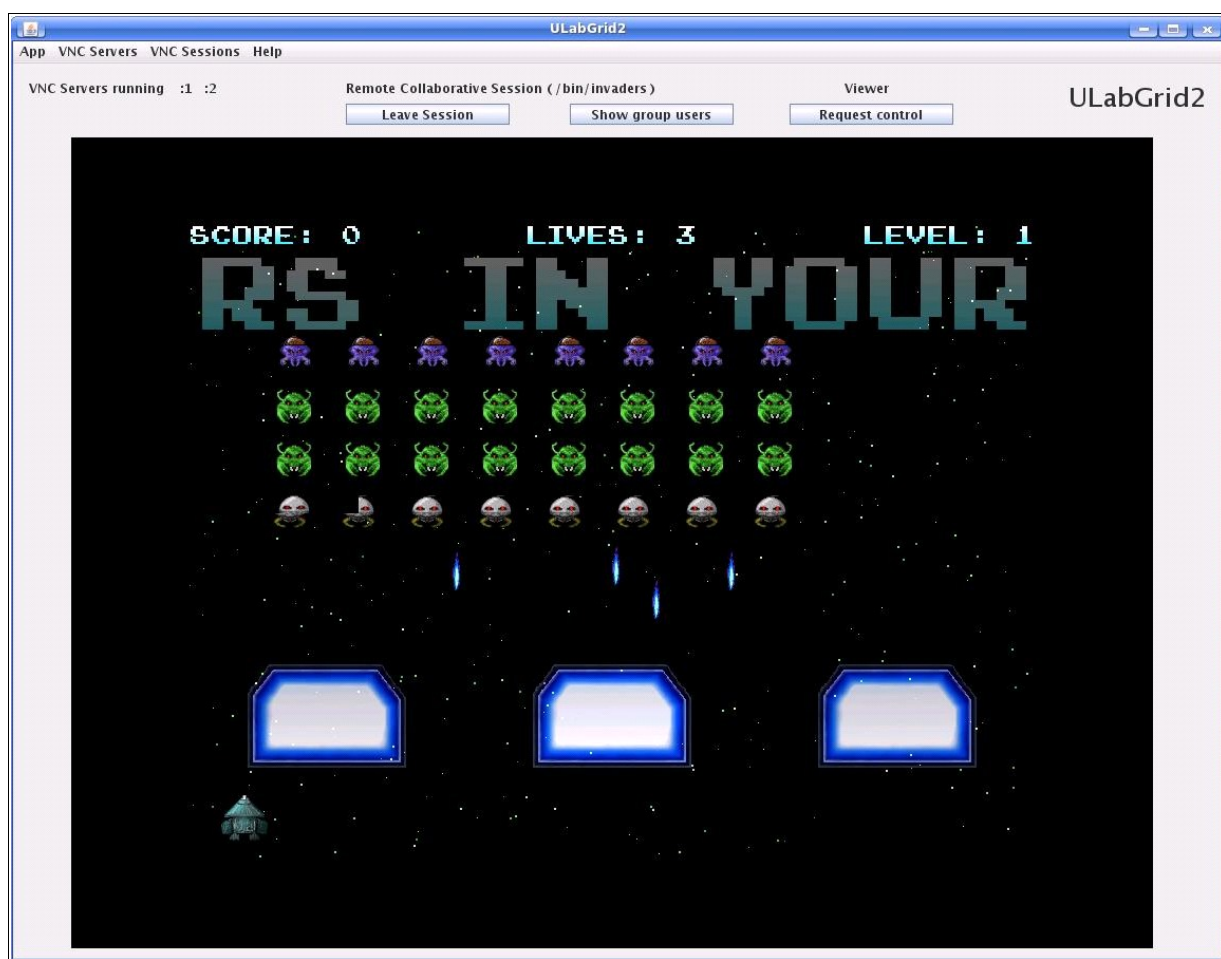


Figura 22. Interfaz gráfica de ULabGrid2 + FaceBook. Sesión Colaborativa

Cada usuario tiene la posibilidad de acceder a varios tipos de sesiones, las cuales presentan algunas diferencias. La diferencia más importante entre estas sesiones de aplicación es su tipo, ya que pueden ser individuales (accesibles por un único usuario) o colaborativas (accesibles por un grupo de usuarios al mismo tiempo).

A continuación se detallan todas las posibilidades que brindan los diferentes tipos de sesiones mostrando los indicativos que aparecen en la parte superior en cada una de ellas:

Sesiones locales: Sesiones lanzadas en la máquina del usuario

- Individuales: Se accede en modo “View only” (no tiene sentido utilizar a través de VNC a una sesión individual servida en tu propia máquina) para que el usuario pueda observar como se está utilizando una determinada aplicación que está siendo ejecutada en su máquina.



Figura 23. Indicador de sesión local individual

- Colaborativas: Accesible por el grupo de usuarios que ha configurado el propio usuario que la ha lanzado. Este grupo de usuarios debe ser formados por amigos FaceBook. El control se comparte entre los miembros del grupo. Inicialmente sólo un usuario del grupo tiene el control de la sesión (ratón, teclado, etc) y los demás están en modo de solo visualización. El resto de compañeros del grupo pueden solicitar el control en cualquier momento y el compañero que está controlando la sesión puede ceder el control o rechazar la petición.



Figura 24. Indicador de sesión local colaborativa

Sesiones remotas: sesiones lanzadas en máquinas de otros usuarios de la aplicación

- Individuales: Accesibles en modo controlador por un único usuario al mismo tiempo. Cuando un usuario accede a este tipo de sesión, ésta es marcada como bloqueada para el resto de usuarios. El usuario que accede tiene el control absoluto sobre la sesión.

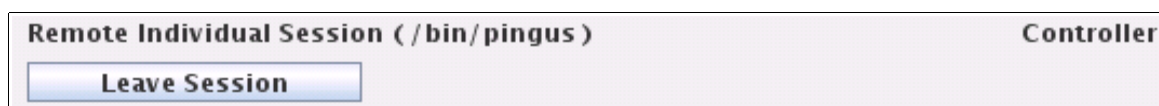


Figura 25. Indicador de sesión remota individual

- Colaborativas: Funcionan del mismo modo que las sesiones colaborativas locales. La única diferencia radica en que en este caso no es el usuario el que ha elegido los miembros del grupo, si no que ha debido ser elegido. Estas sesiones no son visibles para

usuarios no seleccionados en el grupo.



Figura 26. Indicador de sesión remota colaborativa

- Sesiones Grid: Son sesiones que corren en un grid, son lanzadas sin necesidad de asociarlas a un usuario FaceBook como en el caso de las sesiones explicadas anteriormente.

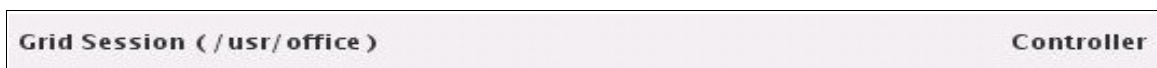


Figura 27. Indicador de sesión grid

Compartir información

Si en la versión de ULabGrid2 integrada con FaceBook los diferentes grupos podían intercambiar información en forma de archivos físicos, utilizando para ello NFS, y el acceso a sesiones se producía de forma remota contra un grid, en este caso la información que comparten los usuarios son las propias sesiones que lanzan en sus máquinas, como se ha adelantado en el apartado anterior.

Para compartir esta información se hace uso del Data Store de FaceBook, una base de datos distribuida que utiliza un lenguaje de consulta propio, FQL, en lugar del clásico SQL.

Esta base de datos distribuida permite la creación de tipos de objetos, objetos y asociaciones y ha sido utilizada para crear el esquema que permite compartir información entre los usuarios de ULabGrid2. Este esquema está basado en el modelo mostrado en la figura 28.

Como se puede observar hay tres tipos de objetos:

- connection: Encargado de enumerar todos las sesiones (objetos ulabgriduser) disponibles almacenando sus identificadores. FQL no permite utilizar las claves foráneas como se utilizan en SQL, por lo que la técnica utilizada consiste en crear una nueva propiedad en el objeto connection que almacene el identificador de cada nuevo objeto ulabgriduser. El objeto connection consta de una propiedad “inicial” vacía necesaria para el correcto funcionamiento del programa.

- ulabgriduser: Se almacenará un objeto ulabgriduser por cada sesión que se esté compartiendo, sea del tipo que sea.

Por cada una de ellas se almacena la siguiente información:

- ip: Dirección IP del servidor VNC que sirve la sesión
- app: Ruta de la aplicación servida
- userid: Identificador FaceBook del usuario que ha servido la sesión
- port: Puerto del servidor VNC mediante el cual se sirve la sesión
- estado: Estado de la sesión

- n: Sesión no bloqueada (individual)
 - s: Sesión bloqueada (individual)
 - c: Sesión colaborativa
 - g: Sesión grid
 - idgrupo: Identifica un objeto group. Se utiliza cuando la sesión es colaborativa (ulabgriduser.estado="c") para poder acceder al objeto group que define que usuarios forman el grupo que puede acceder a la sesión. Si el objeto ulabgriduser no hace referencia a una sesión colaborativa, este campo se deja vacío
 - group: Objeto encargado de almacenar en una cadena de texto los usuarios que conforman un grupo y su estado. La cadena es formada separando con espacios en blanco letras e identificadores de usuarios. Cada letra indica el estado del usuario al que hace referencia el siguiente identificador y pueden darse las siguientes opciones:
 - c: "Controller". Usuario controlador
 - v: "Viewer". Usuario que únicamente puede observar
 - n: "Not connected". Usuario no conectado
 - w: "Asking for the control". Viewer pidiendo el control de la sesión
- Ejemplo de cadena group.usuarios: "c 12334 v 12345 n 12312 v 12134 w 14321"

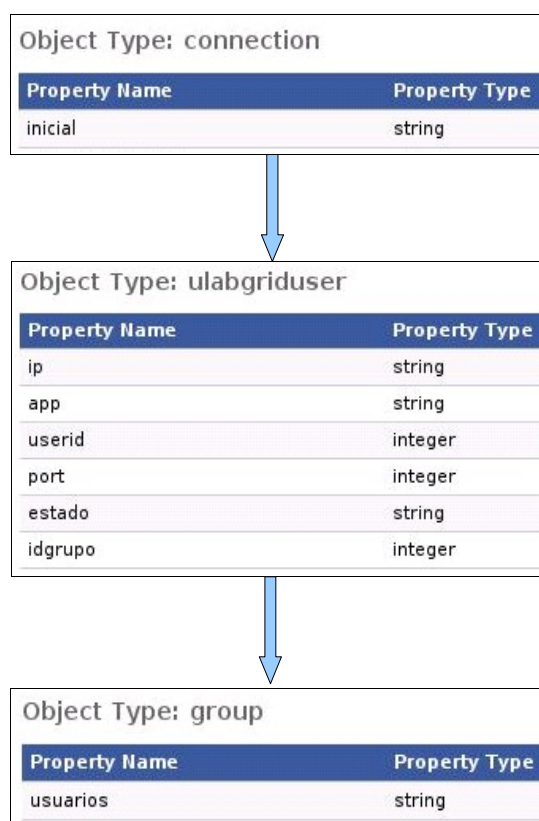


Figura 28. Tipos de objetos almacenados en el Data Store de FaceBook

Comunicación directa entre peers

Si bien no se ha implementado un chat como en el anterior prototipo, las características de FaceBook permiten la interacción entre los usuarios. Ésta es especialmente sencilla en el caso de grupos colaborativos, cuando más necesaria resulta, ya que la propia red social brinda la posibilidad de utilizar un chat entre los usuarios y dentro de un grupo de colaboración todos los usuarios son necesariamente amigos del usuario que lanzado la sesión.

Descubrimiento entre peers

La naturaleza descentralizada del framework en el que está basado este prototipo implica una mayor complejidad en la tarea de descubrimiento de peers. Esta funcionalidad ha sido implementada mediante la utilización del Data Store de FaceBook (DSFB) [42], mencionado anteriormente, para almacenar la información necesaria mediante la cual los usuarios son capaces de conocer en tiempo real que otros usuarios están conectados.

Mediante el campo `userid` del objeto `ulabgriduser` del DSFB, se tiene constancia de qué usuarios están compartiendo sesiones. Esta información aparece reflejada en el menú de aplicación “Servidores VNC” que se descompone en los siguientes submenús:

VNC Servers

- Local Servers (Servidores locales, sesiones lanzadas por el propio usuario)
 - Se muestran tantos menús como sesiones haya lanzado el usuario
 - Sesiones individuales (modo View Only) o colaborativas

- Friend Servers (Servidores lanzados por amigos FaceBook que están utilizando la aplicación ULabGrid2 en ese momento)
 - Aparecen tantos menús como amigos FaceBook estén conectados mediante la aplicación ULabGrid2
 - Por cada amigo aparece un menús por cada sesión que haya compartido
 - Si la sesión es individual aparecerá en el caso de que no esté ocupada por otro usuario.
 - Si la sesión es compartida aparecerá si el usuario forma parte del grupo colaborativo configurado para esa sesión.

- Unknown Servers (Servidores lanzados por usuarios desconocidos que están utilizando la aplicación ULabGrid2 en ese momento)
 - Aparece un menú por cada usuario desconocido que esté utilizando ULabGrid2
 - Por cada uno de ellos aparecen las sesiones individuales que esté compartiendo
 - No se puede formar parte de un grupo colaborativo configurado por un usuario desconocido, ya que este grupo siempre está formado por amigos FaceBook de éste

- Grid Servers (Servidores lanzados sin asociar a usuarios)
 - Aparece un nuevo menú por cada sesión Grid almacenada en el Data Store de FaceBook

Además del descubrimiento de los peers referentes a usuarios que están compartiendo sesiones, reflejado en los submenús “Friend Servers” y “Unknown Servers” que acaban de ser explicados, también queda constancia una vez se ha accedido a una sesión colaborativa.

Al acceder a una sesión colaborativa tenemos la posibilidad de comprobar que usuarios pertenecientes al grupo colaborativo están conectados pulsando el botón “Show group users”, mediante el cual aparece una pantalla como la mostrada en la figura 29. En esta pantalla aparecen los usuarios del grupo, marcando el usuario propio en rojo, junto con su estado que puede ser “Controller”, “Not Connected”, “Viewer” o “Asking for the control”.



Figura 29. Interfaz gráfica de mensaje con los miembros de un grupo colaborativo
* Los nombres y apellidos han sido ocultados

A continuación se detalla el procedimiento de configuración del módulo y de acceso.

Configuración

Al tratarse de una aplicación descentralizada no existe una herramienta de configuración centralizada como en el prototipo anterior.

En este caso la configuración de los recursos a servir se realiza desde la propia aplicación lanzada en la máquina de cada usuario, con la excepción de los recursos grid. En este último caso, la inserción de la información referente a estos recursos en el DSFB se realiza mediante un software desarrollado para ese propósito que es independiente al prototipo que se está explicando. Su funcionamiento se basa en introducir por pantalla los datos necesarios para que estos recursos sean accesibles por los usuarios, esta información consiste en la ruta de la aplicación a compartir y la pareja dirección IP, puerto necesaria para la realización de las conexiones VNC.

También se dispone de una aplicación para devolver a su estado inicial al objeto connection del DSFB, encargado de almacenar la información referente a qué sesiones están siendo servidas en cada momento.

Cuando un usuario lanza la aplicación ULabGrid2 puede acceder a la configuración de las aplicaciones que serán servidas pulsando “App → Configuration”, mediante el cual aparece una pantalla como la mostrada en la siguiente figura.

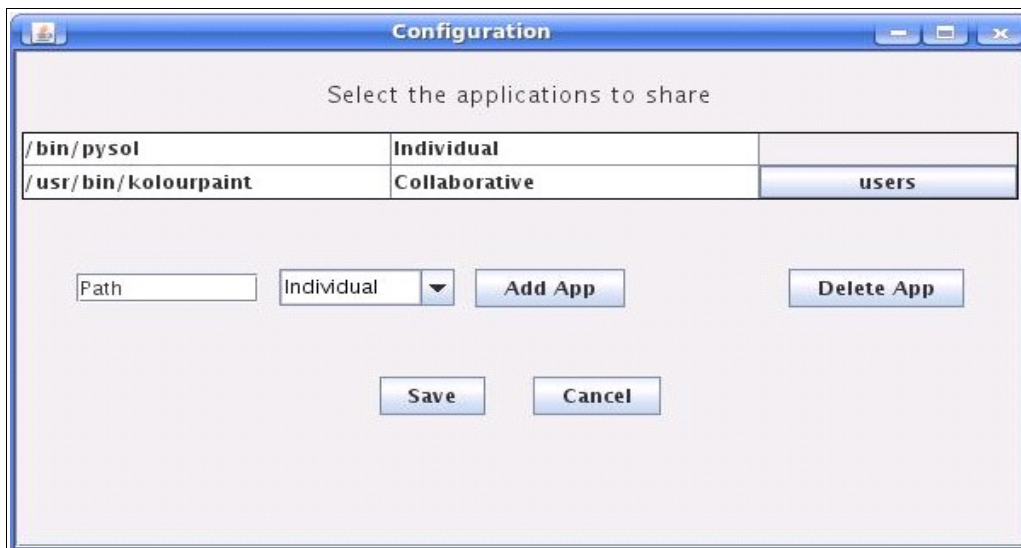


Figura 30. Interfaz grfica del dilogo de configuraci3n

La configuraci3n que cada usuario selecciona es almacenada en un archivo de configuraci3n e indica que aplicaciones comparte, que a su vez indica cuantos servidores VNC se han de lanzar, y si stas sern compartidas como sesiones individuales o compartidas. En este ltimo caso tambin es configurable el grupo colaborativo que tendr acceso a la sesi3n y que ser formado por amigos FaceBook del usuario que la lanza.

Para configurar cada grupo colaborativo es necesario pulsar el bot3n “users” que aparece a la derecha de cada aplicaci3n se ha seleccionado como colaborativa, apareciendo de este modo, una pantalla de selecci3n de usuarios como la mostrada en la siguiente figura.



Figura 31. Interfaz grfica del dialogo de selecci3n de usuarios para grupo colaborativo

* Los nombres y apellidos han sido ocultados

Nota: La configuraci3n de las sesiones a lanzar es nicamente posible bajo el sistema operativo Linux. Utilizando Windows nicamente se puede utilizar la aplicaci3n como cliente ya que no se pueden servir varios servidores VNC al mismo tiempo, por lo tanto tambin no aparece la opci3n de configuraci3n. El programa no funciona bajo otros sistemas operativos.

Acceso a ULabGrid2

Al lanzar la aplicación se hace una llamada a FaceBook mediante la cual se consigue un token que junto con la clave de la aplicación FaceBook es necesario para formar la URL a la que se necesita acceder para hacer login en la aplicación.

Una vez se ha lanzado la URL en un navegador web y el usuario ha accedido y confirmado su acceso en la aplicación, se realiza una nueva llamada a FaceBook para conseguir una clave de sesión y una clave secreta temporal necesarias para las posteriores llamadas a FaceBook, como por el ejemplo para la gestión del DSFB. Si el usuario no ha hecho login la ejecución de la aplicación no logrará pasar de este punto.

Una vez se ha conseguido una sesión, la aplicación descarga la información necesaria referente a los amigos FaceBook del usuario.

Después, se realiza la configuración de aplicaciones explicada en el apartado anterior. Cuando ya han sido configuradas las aplicaciones que van a servirse es necesario pulsar el menú “App → Start” para poder utilizar ULabGrid2. Al pulsar este menú, se recorre el archivo de configuración lanzando un servidor VNC por cada sesión que se desea compartir e insertando la información referente a cada una de estas sesiones en el DSFB mediante la creación de nuevas propiedades en el objeto connection, nuevos objetos ulabgriduser, nuevos objetos group y nuevas asociaciones entre el objeto connection y los nuevos objetos ulabgriduser por un lado, y los nuevos objetos group con los objetos ulabgriduser a los que hacen referencia por otro.

Una vez ha terminado este proceso, dos hilos se encargan de informar a la aplicación de que sesiones están siendo compartidas en cada momento y de los cambios de estado de los usuarios pertenecientes a grupos colaborativos en los que el usuario participa.

A partir de este punto, el usuario puede acceder a sesiones como la mostrada en la figura 22.

4.2.3 Tecnologías utilizadas

Aunque, del mismo modo que con la integración de ULabGrid2 con MOODLE, también se dispone de un Grid de servidores VNC en el caso de la integración con FaceBook, en este último caso dicho Grid no se presenta como el elemento principal a la hora de servir recursos (sesiones), si no que son los propios usuarios que lanzan la aplicación desde Linux los mayores proveedores de estos recursos.

La participación del Grid de servidores VNC en este caso, no implica la utilización de SSH y/o SCP ya que dichos servidores no son controlados desde un servidor central como se hacía con MOODLE, si no que como se ha explicado, el software desarrollado únicamente permite la inserción en el DSFB de los datos referentes a las sesiones servidas desde el Grid. La gestión de los propios servidores ha de realizarse de forma independiente para asegurar su disponibilidad y la configuración de las aplicaciones que se desean servir.

El acceso al sistema se realiza esta vez desde una aplicación de escritorio desarrollada, como se detallará a continuación, en el lenguaje de programación orientado a objetos Java, y haciendo uso del mismo Applet VNC utilizado en la versión de MOODLE, TightVNC. Además es importante destacar la utilización del API de FaceBook además del ya mencionado Data Store.

En la primera fase del primer proyecto se hicieron algunas elecciones que han sido extensibles

a este nuevo proyecto.

– Elección de Sistema Operativo durante el desarrollo: Kubuntu 8.04

Se decidió trabajar con un Sistema Operativo Linux por la naturaleza GPL del proyecto y por la facilidad para utilizar servidores VNC de código abierto.

Como se ha explicado, las otras opciones que se probaron (también distribuciones de Linux) fueron, Fedora 9 y Kubuntu 6.

Entre Kubuntu y Fedora se eligió Kubuntu ya que deriva de Ubuntu y Ubuntu de Debian mientras que Fedora deriva de Red Hat, siendo Debian la distribución más popular de Linux, y por consiguiente la más documentada.

– Elección de Sistemas Operativos para las pruebas (ejecución): Linux y Windows

La aplicación desarrollada soporta ser ejecutada en cualquier máquina con cualquier sistema operativo de la familia Linux. La funcionalidad de la aplicación será completa siempre que dicho sistema operativo tenga instalado el servidor VNC TightVNC, necesario para compartir sesiones de aplicación.

La popularidad del sistema operativo Windows nos ha llevado a la decisión de implementar ULabGrid2 + FaceBook de manera que pueda ser lanzado bajo dicho sistema, en cualquiera de sus versiones, aunque de una forma en la que la aplicación se comporta únicamente como cliente ya que la naturaleza de Windows impide servir diferentes sesiones VNC al mismo tiempo e independientes del escritorio del usuario de la propia máquina.

La aplicación no funciona bajo sistemas operativos diferentes de Linux y Windows.

– Elección de red social: FaceBook

La utilización de una red social software es una parte fundamental del sistema ya que es la responsable de la interacción entre usuarios, el descubrimiento de nuevos peers y la difusión de la aplicación entre otras cuestiones.

La elección de FaceBook se debió principalmente a su popularidad y su naturaleza, la cual permite la creación de aplicaciones, tanto Web como de escritorio, que hacen uso de la propia red. Además esta tarea es facilitada por los APIs de FaceBook disponibles para diferentes lenguajes.

– Base de datos distribuida: Data Store de FaceBook

Se encarga de mantener información sobre el estado de conexión de los usuarios, los recursos disponibles, las aplicaciones que serán servidas desde el Grid o las máquinas de los usuarios, los grupos colaborativos que son formados, el estado de las sesiones de cada grupo (tipo de sesión, control de sesiones colaborativas por parte de los usuarios), etc.

La elección del DSFB como base de datos distribuida se debe a la elección de FaceBook como red social. La otra opción que se barajó fue utilizar una base de datos en un servidor propio lo que implicaría volver a tener un servidor centralizado que se convertiría en un punto crítico del sistema, razón por la que desechamos esta opción.

En referencia a la utilización del DSFB en el desarrollo del proyecto, cabe destacar la existencia de la funcionalidad DataStoreAdmin orientada a la gestión de los objetos, tipos de objetos y asociaciones que toman parte en los modelos definidos en dicho almacén de datos.

– Elección de servidor VNC: TightVNC

La elección del servidor VNC para el primer prototipo se hizo extensible para este último proyecto.

Como ya se comentó para el primer proyecto, las primeras pruebas de escritorio remoto las hicimos con el escritorio remoto de Kubuntu 8.04 para después realizar pruebas con Xtightvnc y Tightvnc, comprobando que tiene un mejor funcionamiento el segundo, por tanto Tightvnc fue el VNC server elegido, que además permite ser servido a través del interfaz web mediante un applet Java de código abierto, que podremos modificar según sea conveniente.

– Elección de aplicaciones a compartir: Eclipse + Repast Symphony, SimProcess, Pingus, PySol, Open-Invaders.

Las aplicaciones servidas en las pruebas de la aplicación fueron las ya elegidas para el primer prototipo: Eclipse + Repast Symphony y SimProcess.

Además se compartieron otras aplicaciones con la única premisa de que fueran de código abierto y se eligieron tres juegos: Pingus (<http://pingus.seul.org/welcome.html>), PySol (<http://www.pysol.org/>) y Open-Invaders (<http://www.jamyskis.net/invaders.php>).

4.2.4 Implementación

Diagrama de clases

A continuación se muestra el diagrama de clases en el que está basado el desarrollo del proyecto.

Nótese, que en el primer diagrama se muestra una visión global del proyecto, y para ello se han simplificado las clases omitiendo sus atributos y métodos, que a su vez serán explicados en más detalle en un posterior análisis individual de cada una de estas clases.

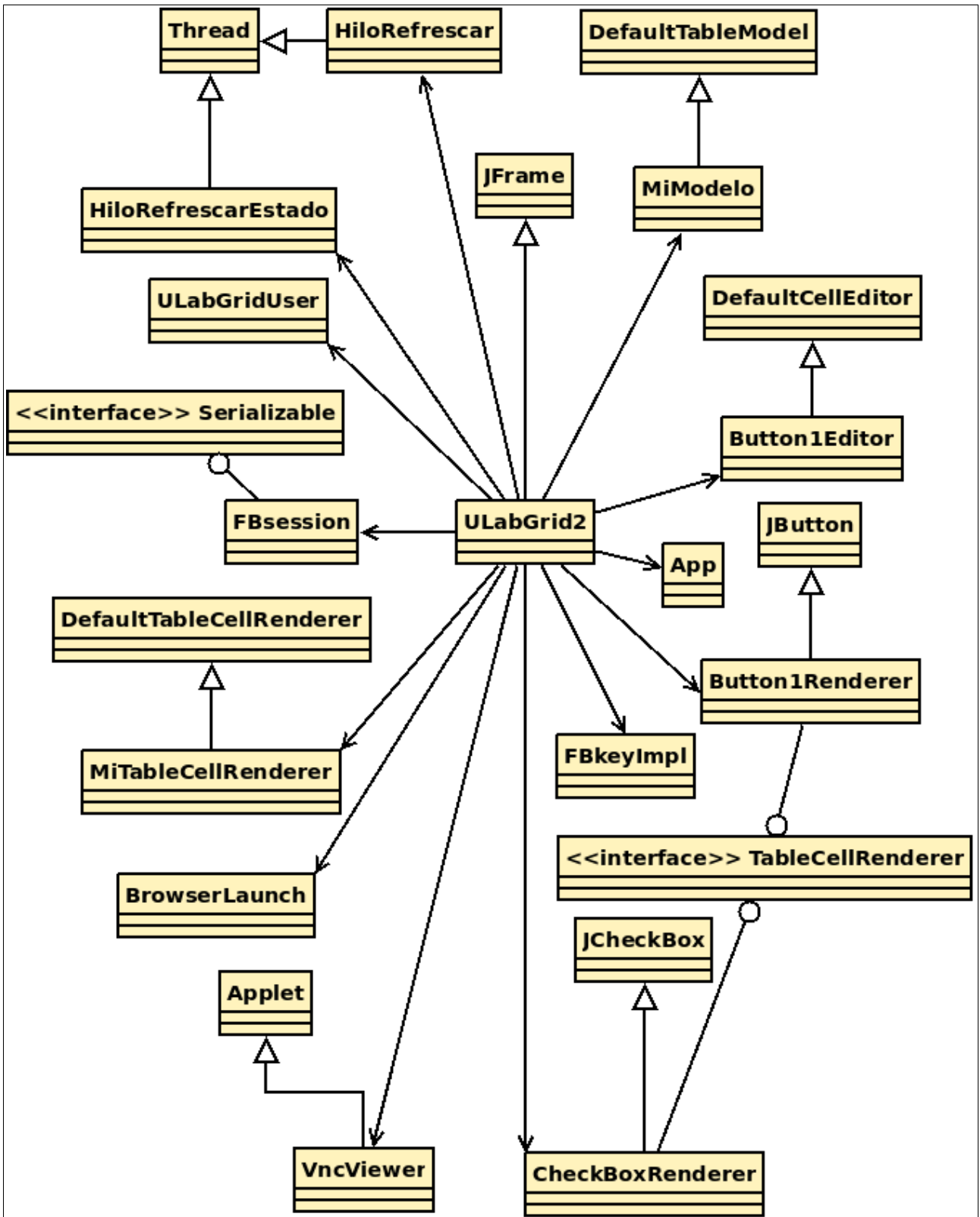


Figura 32. Diagrama de clases

Como se ha mencionado, a continuación se detallan las clases desarrolladas por separado:

ULabGrid2

Es la clase principal de la aplicación y como se puede comprobar en el diagrama anterior, es la clase encargada de utilizar el resto de clases desarrolladas.

Entre otros métodos, consta del método main, por lo tanto es la clase ejecutable. Extiende de la clase JFrame ya que muestra el entorno gráfico al ser ejecutada.

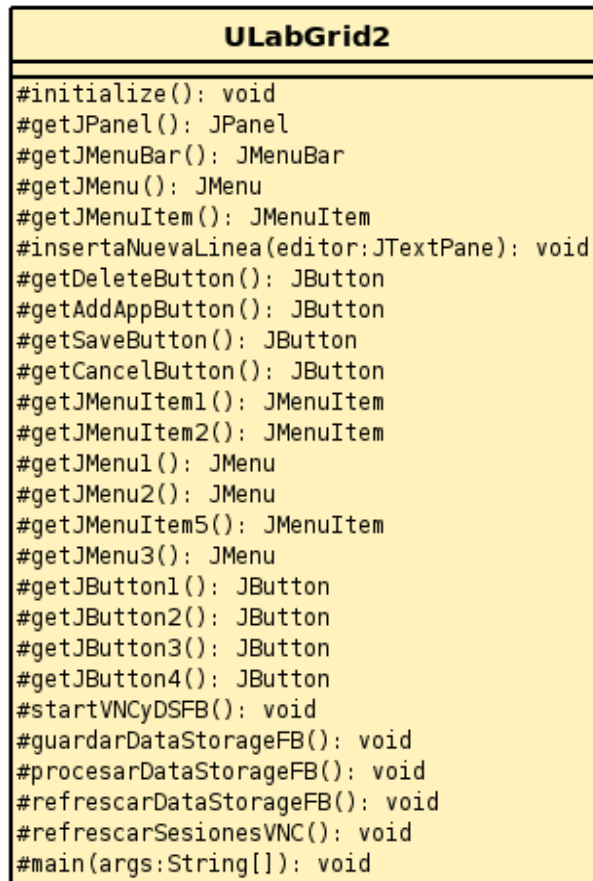


Figura 33. Clase ULabGrid2

Nota: El gran número de atributos de la clase hace imposible mostrarlos todos en la figura por lo que los detallo a continuación:

```
protected long serialVersionUID
protected Runtime r
protected Process p
protected String API_KEY
protected long fidConnection, fidUlabgriduser, fidGroup
protected Vector fidUlabgridusers
protected FBkeyImpl fb
protected int
protected int numApps
protected String xStartup
protected String rutaConf
protected String sCadena
protected String contenidoXStartup
protected FileWriter textOut
protected BufferedReader bf
```

```

protected Vector rutas
protected String servidoresLanzados
protected boolean appletLanzado
protected boolean startPushed
protected HashMap hashMap
protected List friends
protected HashMap friends2
protected boolean bloqueaSesion
protected ULabGridUser sesionBloqueadaULGuser
protected String estado
protected Vector fidUsers
protected Vector fidUsersBackUp
protected String nombreJMenu
protected String token, url, userIdObjeto, ipObjeto, appObjeto
protected String portObjeto, estadoObjeto, idGrupoObjeto
protected FBsession sesionFB
protected long userId
protected int flagSesion
protected HashMap sesionesCompartidasULGuser
protected HashMap sesionesCompartidasHashMap
protected ULabGridUser sesionCompartidaActual
protected ULabGridUser sesionCompartidaAuxiliar
protected HashMap idGrupoApp
protected boolean esSesionCompartida
protected ULabGridUser sesionAuxiliar
protected DefaultTableModel dm
protected JTable table
protected FileWriter fichero
protected boolean esSesionGrid
protected JPanel jPanel
protected JLabel jLabel
protected JLabel jLabel1
protected JLabel jLabel2
protected JLabel jLabel3
protected JMenuBar jJMenuBar
protected JMenu jMenu
protected JMenuItem jMenuItem
protected JMenuItem jMenuItem1
protected JMenuItem jMenuItem2
protected JMenu jMenu1
protected JMenu jMenu7
protected JMenu jMenu2
protected JMenuItem jMenuItem5
protected JDialog dialogo
protected JMenu jMenu3
protected JMenu jMenu4
protected JMenu jMenu5
protected JMenu jMenu6
protected JMenuItem jMenuItem7
protected JButton jButton1
protected JButton jButton2
protected JButton jButton3
protected JButton jButton4
protected JTextField rutaApp
protected JComboBox tipoApp
protected JFrame v = null
protected VncViewer applet1
protected HiloRefrescar hr
protected HiloRefrescarEstado hre

```

FbkeyImpl

Esta clase es la encargada de realizar las llamadas al Data Store de FaceBook que en un principio se realizaron mediante RMI. Nótese que incluye en sus atributos la clave secreta de la aplicación.

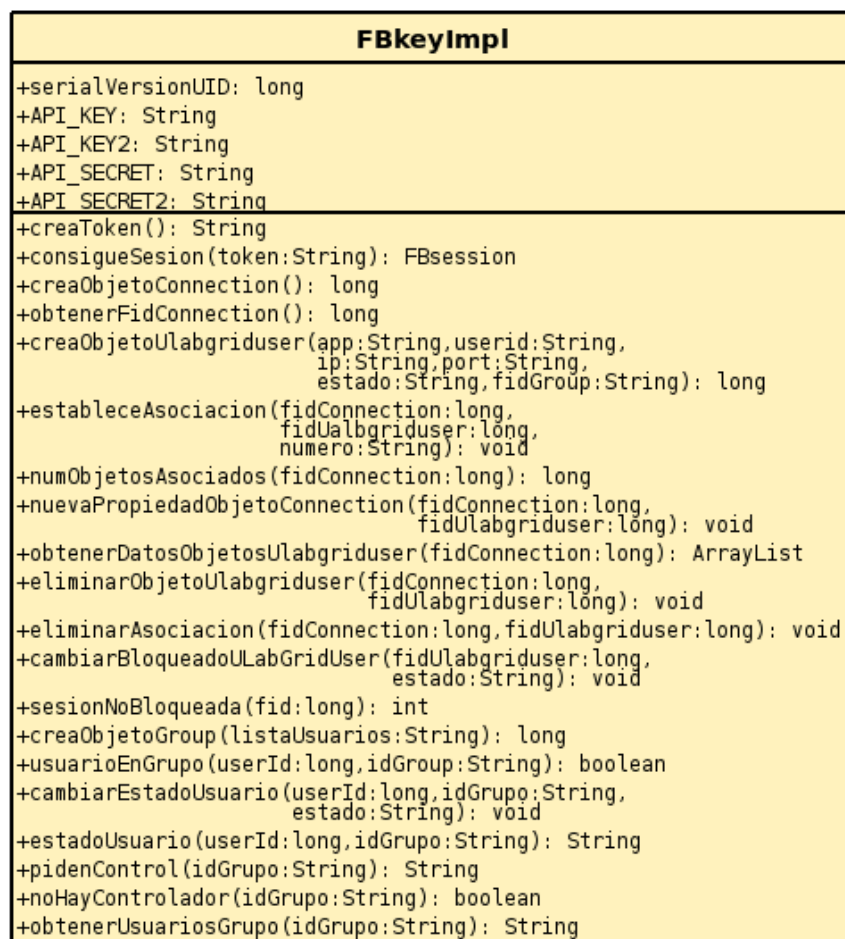


Figura 34. Clase FBkeyImpl

App

Esta clase representa las aplicaciones compartidas mediante ULabGrid2.

Se utiliza como una estructura encargada de almacenar por cada aplicación, su ruta, y si se ha configurado o no como una aplicación colaborativa.

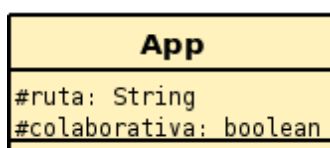


Figura 35. Clase App

ULabGridUser

Esta clase representa un usuario de la aplicación y la aplicación que comparte. Es decir, si un usuario comparte dos aplicaciones se crearán dos instancias de este objeto. Al igual que en la clase anterior, es utilizada como estructura para almacenar datos, en este caso, los referentes a cada sesión compartida por los usuarios.

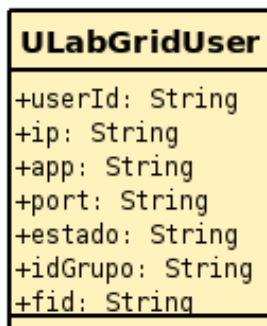


Figura 36. Clase ULabGridUser

FBsession

Esta clase representa una sesión de FaceBook. Al acceder a FaceBook se siguen una serie de pasos explicados anteriormente mediante los cuales se consigue una clave de sesión y una clave secreta temporal que son las claves almacenadas en los atributos session y secret de esta clase.

Los métodos creaMap se utilizan en determinadas llamadas a los métodos de la clase FbkeyImpl que necesitan la creación de estructuras Map para ejecutarse correctamente. Nótese que implementa el interfaz Serializable.

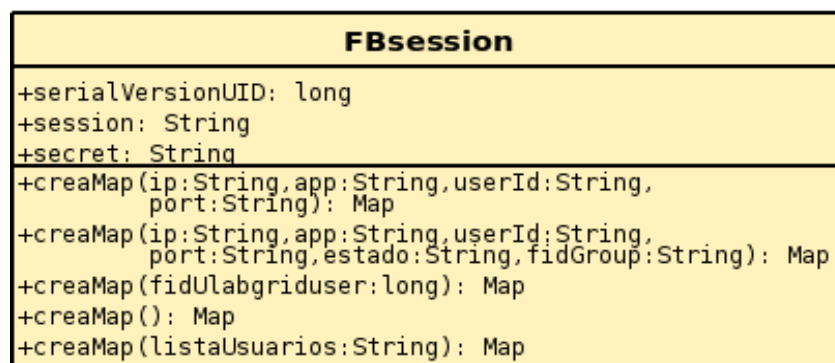


Figura 37. Clase FBsession

HiloRefrescarEstado

Esta clase que extiende de Thread, se ejecuta continuamente en background mientras el usuario utiliza la aplicación. Su función es la de alertar al usuario sobre posibles cambios en el control, es decir, si el usuario tiene el control de una aplicación colaborativa y un compañero solicita dicho control, este hilo será el encargado de notificar ese estado y el programa lanzará un mensaje de alerta.

HiloRefrescarEstado
+continuar: boolean
+estado: String
+userW: String
+num: int
+detenElHilo(): void
+reanudarElHilo(): void
+run(): void

Figura 38. Clase HiloRefrescarEstado

HiloRefrescar

Esta clase, al igual que la anterior, hereda de la clase Thread y por lo tanto se ejecuta en background mientras el usuario utiliza la aplicación. Es la encargada de detectar cambios en el Data Store de FaceBook de manera que los menús son actualizados en tiempo real añadiendo las nuevas aplicaciones compartidas por los usuarios y eliminando las de usuarios que se desconecten.

HiloRefrescar
+continuar: boolean
+detenElHilo(): void
+run(): void

Figura 39. Clase HiloRefrescar

BrowserLaunch

Como se ha explicado, para utilizar ULabGrid2 es necesario hacer login en FaceBook y para ello se lanza un navegador web con la URL adecuada. Aunque el programa utiliza la clase Desktop, esta es de origen reciente y por lo tanto hay JREs que no la soportan. En tal caso se utiliza ésta clase para lanzarlo.

BrowserLaunch
+openURL(url:String): void

Figura 40. Clase BrowserLaunch

Button1Editor

Esta clase se encarga de cambiar el evento que se ejecuta al pulsar sobre una celda de una JTable (por defecto se edita) para conseguir ejecutar el comportamiento de un JButton.

Nótese que la clase extiende de DefaultCellEditor.

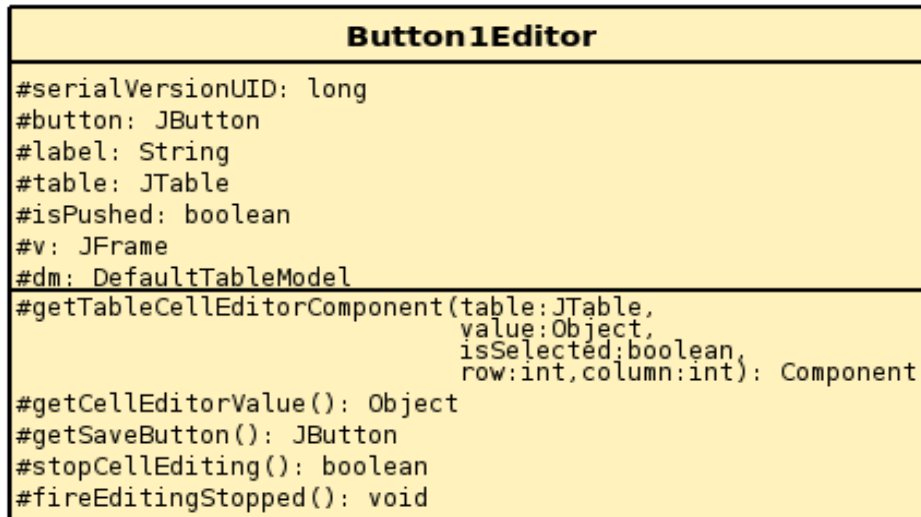


Figura 41. Clase Button1Editor

Button1Renderer

Esta clase se encarga de cambiar la manera que se muestran los datos de una celda (por defecto toString()) para conseguir mostrar jButtons.

Nótese que la clase extiende de la clase JButton e implementa el interfaz TableCellRenderer.

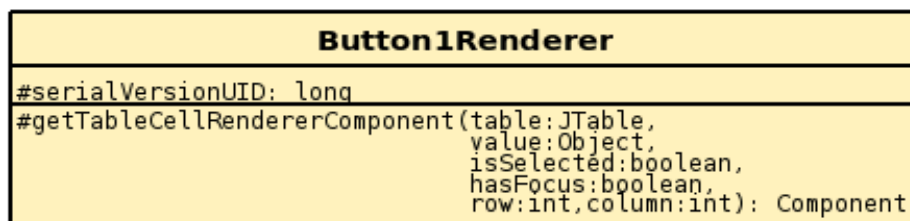


Figura 42. Clase Button1Rederer

CheckBoxRenderer

Esta clase se encarga de cambiar la manera en la que se muestran los datos de una celda (por defecto toString()) para conseguir mostrar JCheckBoxs.

Nótese que extiende de la clase JCheckBox e implementa el interfaz TableCellRenderer.

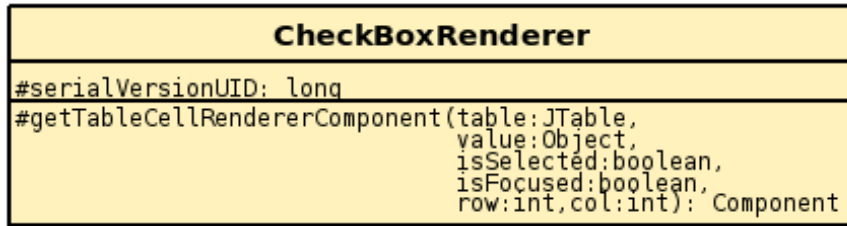


Figura 43. Clase CheckBoxRenderer

MiModelo

Esta clase extiende de DefaultTableModel y se utiliza para manejar la JTable insertada en la ventana que aparece al pulsar App -> Configuration.

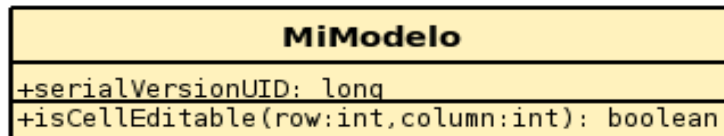


Figura 44. Clase MiModelo

MiTableCellRenderer

Esta clase se encarga de customizar el rederizado de las celdas de JTable que contienen String.
 Modifica el color de fondo de la celda (blanco)
 Modifica el aspecto del texto de las celdas (Arial, negrita, tamaño 12)
 Extiende de la clase DefaultTableCellRenderer

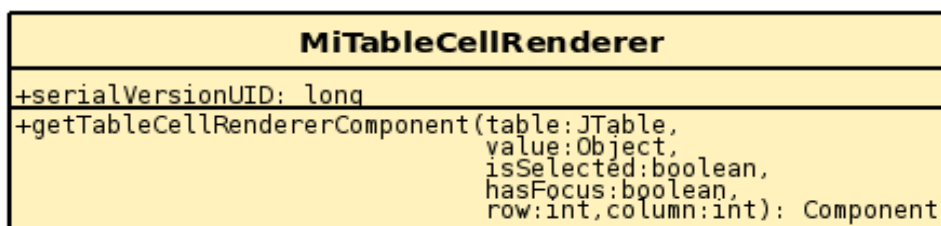


Figura 45. Clase MiTableCellRenderer

VncViewer

Esta clase que extiende de la clase applet, es la clase ejecutable del paquete TightVNC y por tanto la clase que es instanciada desde ULabGrid2, pero ésta a su vez es asociada con otras clases del mismo paquete que no aparecen en el diagrama de clases mostrado anteriormente.

Estas clases son:

- AuthPanel
- ButtonPanel
- CapabilityInfo
- CapsContainer
- ClipboardFrame
- DesCipher
- HTTPConnectSocket
- HTTPConnectSocketFactory
- InStream
- MemInStream
- OptionsFrame
- RecordingFrame
- ReLoginPanel
- RfbProto
- SessionRecorder
- SocketFactory
- VncCanvas
- VncCanvas2
- ZlibInStream

4.2.5 Problemas encontrados

Detalle a continuación problemas y dificultades más importantes encontrados a lo largo del desarrollo del proyecto y las soluciones que se tomaron para solucionarlos.

- Aprendizaje de programación aplicaciones FaceBook

La utilización de FaceBook ha aportado muchas ventajas al proyecto, pero también ha implicado la obligación del aprendizaje de el procedimiento que debe seguir una aplicación que se conecta a FaceBook, las librerías propias a utilizar y los API de desarrollo. Para ello fue de una gran utilidad, la ayuda de la activa comunidad de desarrolladores de FaceBook que obtuve a partir de sus diferentes foros, además de la extensa documentación existente.

- Windows no sirve sesiones VNC en paralelo (problema común al prototipo anterior)

El grid y las máquinas de usuarios encargados de servir las sesiones VNC accedidas desde la aplicación desarrollada pueden servir una o varias sesiones VNC al mismo tiempo. Estos computadores no pueden servir varias sesiones VNC al mismo tiempo si su sistema operativo es Windows, ya que en este caso, los accesos mediante escritorio remoto siempre tienen el acceso a la sesión del usuario actual, impidiendo de esta forma el acceso de diferentes usuarios al mismo tiempo. Esta fue una de las razones por las que las conexiones a Grid deben ser realizadas a computadores Linux, ya que en este caso, la sesión servida no es necesariamente la misma que la que es visualizada por el usuario de la computadora. Los ordenadores bajo Linux soportan múltiples sesiones X11 lo que permite a su vez, alojar varias conexiones VNC en la misma IP, pero utilizando diferentes puertos.

– Cada servidor VNC solo tiene un archivo de configuración (problema común al prototipo anterior)

Utilizando un mismo usuario Linux se pueden lanzar varias instancias de un servidor VNC, pero esto presenta un problema, únicamente un archivo de configuración (/home/\$USER/.vnc/xstartup) gestiona que aplicación se servirá a través del servidor. Sin embargo, la naturaleza del proyecto implica lanzar aplicaciones diferentes y la solución a este problema consistió en reconfigurar el archivo inmediatamente antes de lanzar cada instancia de cada servidor VNC. De esta manera, si lanzamos dos servidores y queremos compartir dos aplicaciones A y B, se configura el archivo xstartup para lanzar la aplicación A, se lanza el servidor número uno, se reconfigura el archivo xstartup para lanzar la aplicación B y se lanza el servidor número dos. De este modo, aunque el archivo de configuración haya quedado configurado de tal manera que la aplicación a compartir es la B, el servidor número uno seguirá sirviendo la aplicación A ya que ésta era la que constaba en dicho archivo en el momento del lanzamiento del servidor.

– Autenticación del usuario de la aplicación de escritorio en FaceBook

El proceso de autenticación para aplicaciones de escritorio sigue el siguiente esquema:

1. Crear token de autenticación
2. Formar URL con el token y la clave de aplicación.
3. Acceder a la URL con un navegador web para que el usuario haga login en el sistema
4. Nueva llamada a FaceBook para conseguir una clave de sesión y una clave secreta temporal.
5. Creación del cliente de FaceBook y utilización del mismo.

Al crear una aplicación en FaceBook se facilitan dos claves necesarias para las posteriores llamadas de la aplicación, estas son: clave de aplicación y clave secreta. Desde FaceBook se recomienda no incluir las claves secretas en el código de aplicaciones de escritorio para evitar posibles usos malintencionados de la aplicación desarrollada. Así, la forma correcta de conseguir la autenticación es almacenar la clave secreta de aplicación en un servidor, evitando así que la clave aparezca en el código del usuario final, que la utiliza para generar las claves secretas temporales que son pasadas a las aplicaciones de escritorio que lo soliciten.

La obtención de la clave secreta temporal supuso varios problemas ya que un primer lugar se generó desde un servidor web, pero luego no fue sencillo conseguir utilizar estos datos generados mediante PHP por la aplicación desarrollada en Java.

Finalmente se consiguió generar la clave por medio de un servidor RMI (Java) destinado a la invocación remota de métodos, de modo que la comunicación entre la aplicación de escritorio y el servidor fue mucho más sencilla.

Una vez solucionado el problema de generación de claves secretas surgió un nuevo problema con las mismas que nos hizo replantear de nuevo este proceso, ya que la utilización de determinados métodos necesarios para el desarrollo del programa no es posible utilizando claves secretas temporales. De este modo, el servidor RMI que contaba con la clave secreta se comenzó a utilizar, además de generador de claves secretas temporales, como medio para ejecutar los métodos que necesitaban la clave secreta de aplicación para funcionar. Es decir, la solución consistió en invocar de manera remota los métodos que necesitaban la clave secreta de aplicación con el objeto de no almacenar esta clave en el código que se facilita al usuario. Se necesitó ejecutar de esta forma todos los métodos que permiten la gestión del DSFB.

Nota: El proceso de autenticación para aplicaciones de escritorio descrito arriba ha sido modificado recientemente (ya no es necesario crear el token de autenticación ni realizar la llamada destinada a conseguir una sesión) con la inclusión del conjunto de APIs “Connect” (http://wiki.developers.facebook.com/index.php/Facebook_Connect), aunque el proceso que

se ha descrito anteriormente sigue siendo válido.

Nota2: Se desarrolló una nueva versión del prototipo en la cual se eliminó la parte del servidor RMI para poder demostrar que es posible crear una aplicación totalmente P2P, aún a costa de realizar el desarrollo sin cumplir las recomendaciones de FaceBook, esta es la razón por la que el servidor RMI no aparece en el Framework explicado al comienzo de esta sección.

– Invocación de métodos del Data Store de FaceBook sólo disponibles para aplicaciones web
Se ha comentado en el punto anterior que se necesitó ejecutar los métodos de gestión del DSFB de forma remota ya que es necesario contar con la clave secreta de aplicación para poder utilizar algunos de estos métodos.

Volvieron a surgir problemas con estos métodos, ya que independientemente de la clave utilizada, hay métodos que no se permite lanzar desde aplicaciones de escritorio y que son, por tanto, únicamente utilizables por aplicaciones web.

Estos métodos son los destinados a la definición de objetos, acceso a objetos, definición de asociaciones de objetos y acceso a asociaciones de objetos, siendo los métodos destinados a la definición de preferencias de usuario los únicos accesibles por aplicaciones de escritorio.

La solución consistió en definir otra nueva aplicación en FaceBook de manera que pudimos contar con una aplicación de cada tipo: Por un lado, una aplicación de escritorio como hasta ese momento, que funcionaba con una clave secreta temporal y se utilizaba para conseguir el identificador de usuario y hacer login. Y por otro lado, una aplicación web para, utilizando sus claves de aplicación y secreta, poder acceder a todos los métodos del DSFB.

– Falta de métodos referentes al Data Store de FaceBook en el API de Java

En el diseño de la aplicación se planeó seguir los siguientes pasos en la creación de los objetos del DSFB referentes a cada sesión compartida por los usuarios y para la obtención de la información sobre qué otras sesiones hay disponibles:

1. Crear una instancia del objeto connection (si todavía no existe) y almacenar su identificador (método `Data.createObject`)
2. Crear varias instancias del objeto `ulabgriduser`, una por cada sesión a compartir (método `Data.createObject`)
3. Definir la asociación: `connection` → `ulabgriduser` (método `Data.defineAssociation`)
4. Crear asociaciones entre el objeto `connection` y los objetos `ulabgriduser` (método `Data.setAssociation`)
5. Obtener los identificadores (o directamente datos) de los objetos `ulabgriduser` asociados, a partir del Id del objeto `connection` (`Data.getAssociatedObjects`).

El problema surgió porque este último método, `Data.getAssociatedObjects`, no está disponible en el API de FaceBook en Java, por lo que fue necesario improvisar una solución que resolviese la tarea que era destinada a tal método. Era estrictamente necesario poder acceder a los objetos `ulabgriduser` asociados al objeto `connection` para poder conocer que otras sesiones de aplicación estaban siendo compartidas, y sin poder utilizar el método de acceso a los objetos `ulabgriduser` por medio de sus asociaciones con el objeto `connection`, la solución consistió en crear una nueva propiedad en el objeto `connection` que almacenase el identificador del nuevo objeto `ulabgriduser` asociado. De éste modo, si el un determinado objeto `ulabgriduser` tuviese el identificador, por ejemplo, 1234, se crearía una nueva propiedad en el objeto `connection` llamada “u1234”, de tal forma que recorriendo las propiedades del objeto `connection` se consigue acceder a los objetos `ulabgriduser` asociados.

- Integración del Applet VncViwer en el código de la aplicación

El applet utilizado funcionaba como una aplicación stand-alone y su integración en ULabGrid2 supuso algunos problemas.

Tras conseguir ubicarlo en el JFrame que almacena los elementos gráficos de la aplicación ULabGrid, la ejecución producía excepciones en el código del applet, ya que este estaba programado para recibir algunos parámetros al ser lanzado, como por ejemplo los parámetros referentes a puerto y host. La solución consistió en modificar el código original para evitar la petición de estos parámetros.

También se modificó el código del applet para evitar la petición de contraseña inherente a cada acceso a escritorio remoto. Esta contraseña ya no es necesaria al considerarse que la inserción de la contraseña de acceso a FaceBook es suficiente para la utilización de la aplicación.

Otros problemas surgieron al intentar recargar el applet, por ejemplo al cambiar de una sesión a otra. Para la realización de estos cambios es necesario cambiar algunos parámetros del applet (IP:puerto) pero no basta únicamente con esto. El proceso seguido para cambiar de una sesión a otra es el descrito a continuación:

1. Parar el applet (applet.stop())
2. Destruir el applet (liberar recursos, applet.destroy())
3. Establecer parámetros
4. Inicializar applet (applet.init())
5. Lanzar applet (applet.start())

- Configurar sesiones propias

Cuando un usuario lanza servidores VNC para compartir sesiones es necesario indicar que par IP:puerto corresponde a cada una de ellas.

Apareció un nuevo problema con referencia a este aspecto, ya que al comprobar desde el código Java qué IP tiene la máquina en la que se lanza la aplicación, siempre indicaba 127.0.0.1 por lo que la sesión únicamente funcionaba en localhost, siendo inaccesible por el resto de usuarios.

No se trataba de un problema en el código, si no una mala configuración de la máquina en la cual el código estaba siendo desarrollado y testeado. Bastó con cambiar en el archivo de Linux /etc/hosts la dirección 127.0.0.1 que aparecía a la izquierda del nombre del equipo por la dirección IP pública de la máquina.

4.2.6 Etapas y plazos de ejecución

AÑO 2009

Junio-Julio

- Análisis
- Elección de Sistema Operativo
- Elección de servidor VNC

- Elección de Sistema de Red Social
- Elección de aplicaciones a compartir
- Diseño del sistema

Agosto-diciembre

- Desarrollo de la primera versión del prototipo
- Instalación del SW elegido en las máquinas disponibles.
- Implementación de la primera versión del prototipo

AÑO 2010

Enero-Febrero

- Implantación del prototipo en dos máquinas.
- Evaluación de la primera versión del prototipo para obtener las primeras conclusiones
- Definir los cambios a realizar en el proyecto

Marzo-Mayo

- Implementación de los cambios en la versión definitiva del prototipo
- Implantación del prototipo sobre un Grid real
- Comprobación de su funcionamiento sobre un entorno real (utilización por alumnos)
- Analizar y evaluar los datos tras el uso del prototipo UlabGrid2 por parte de los alumnos.
- Obtención de conclusiones
- Evaluación externa y difusión de los resultados del proyecto

5. Análisis de Resultados

En esta sección se explican los procedimientos seguidos en pruebas reales de los prototipos desarrollados, se muestran los resultados y el análisis de los mismos.

5.1 ULabGrid2 integrado con el CMS MOODLE

Finalmente, describimos un estudio experimental mediante el cual planeamos demostrar la habilidad del framework para soportar actividades de aprendizaje peer-to-peer y evaluamos las experiencias de aprendizaje de los usuarios.

Para validar nuestro framework y prototipo desarrollamos un experimento con varios estudiantes. Programamos una actividad de aprendizaje para estudiantes de Ingeniería Informática. En esta carrera, los estudiantes hacen uso de un Entorno de Desarrollo Integrado (IDE). Esta actividad es normalmente efectuada por el profesor de las clases de prácticas. En contraste, en el experimento los estudiantes partieron de unos datos iniciales que les proporcionamos junto con las instrucciones y los objetivos de la tarea de aprendizaje, y ellos tuvieron que realizar la tarea por ellos mismos y con la ayuda de los demás compañeros (Peers). El experimento constó de los pasos expuestos a continuación:

- 1) Los profesores dan a los estudiantes acceso al sistema ULabGrid2 proporcionándoles un nombre de usuario y contraseña; al mismo tiempo, los profesores forman grupos pequeños, de 2 o 3 estudiantes, para realizar las actividades. Los estudiantes entran en el sistema y seleccionan el curso de Moodle que implementa el módulo con el prototipo. Todos los estudiantes realizarán las mismas tareas, aunque el sistema puede soportar que cada grupo de estudiantes realice una tarea diferente. En este momento, los estudiantes pueden observar con que compañero/s comparten grupo y que otros grupos han accedido al mismo curso.

- 2) Los estudiantes instancian una sesión remota de aplicación desde el pool de un grid, solo se instanciará una sesión por grupo. Los estudiantes acceden a las sesiones a través del cliente VNC ejecutado en su navegador Web. Los estudiantes realizarán la tarea en la sesión remota de aplicación. Los estudiantes de cada grupo colaboran pasándose de uno a otro el control de la aplicación y conversando a través del sistema de chat.

- 3) En cualquier momento, los estudiantes pueden estar interesados en acceder a sesiones en las que otros estudiantes estén trabajando en ese mismo momento. Para que cada estudiante pueda ver las sesiones de otros estudiantes debe compartir su sesión con los demás. Un estudiante debería acceder a las sesiones de los otros para comprobar como están realizando la tarea y obtener ideas de como proceder, además pueden mantener conversaciones a través del chat para realizar preguntas sobre la sesión de aplicación actual.

- 4) En cualquier momento, los estudiantes pueden acceder a las “habitaciones chat” de otros grupos sin necesidad de compartir su propia sesión para pedir acceso a su sesión o realizar alguna pregunta sin visualizar la sesión.






- 5) En cualquier momento, los estudiantes pueden proporcionar al resto de grupos acceso libre a su sesión; los otros grupos recibirán el aviso en su propia interfaz y podrán entrar a la sesión para proporcionar ayuda, etc.







- 6) Cuando la tarea de aprendizaje está completada, cada grupo tiene que publicar sus resultados en un directorio compartido. Los otros estudiantes pueden acceder a los resultados de otros grupos para compararlos con los suyos.

A continuación se muestran las instrucciones entregadas a los alumnos al inicio del experimento y los cuestionarios que tuvieron que resolver con los resultados obtenidos.

Ejercicio 1 (20 minutos):

Esta actividad la realizáis en grupos de 2 o 3 personas, podeis:

- (por defecto) una persona controla el ratón y las otras observan y chatean (Non-shared control) : estados **a y b**.
- En este modo de funcionamiento un usuario controla el ratón y la sesión (Controller user) y puede dejar el control pinchado ; mientras que otros usuario pueden pedir el control .
- Compartir el manejo del ratón simultáneamente (Shared Control) estados **c y d**. Para pasar al estado compartido se pincha en  **a o b**.
- En este modo de funcionamiento el uso de ratón es compartido, pero un usuario tiene el control de la sesión para des-compartirla pinchando en , Los otros usuarios pueden pedir el control de la sesión pinchando .

a) Non-Shared Control Controller user	 	Solo una persona controla el raton (puede compartir el control) El usuario es el controlador (puede pedir dejar el control)
b) Non-Shared Control Non-Controller user		Solo una persona controla el raton. El usuario no es el controlador (puede pedir tomar el control)
c) Shared Control Controller user	 	Control de ratón compartido (puede pedir pasar a control no compartido) El usuario controla la sesión (puede pedir dejar el control)
d) Shared Control Non-Controller user		Control de ratón compartido El usuario no controla (puede pedir el control)

Los pasos de la actividad son :

- Acceder al curso : Eclipse
- Logeate con tu username: tu nombre de pila, y password : primera letra de nombre y primera letra de apellido
- Accede a tu sesión : My ULG Session [o]
- Realiza la actividad con la aplicación Eclipse.
 - Abre el proyecto Java : Ejercicio 1.
 - Corrige el código de *ButtonImageExample.java* para que salga una ventana con 2 botones diferentes:













- Comprueba que funciona ejecutando el proyecto con Run->Java Application.
- Publica tu solución desde el interface de control My ULG : MyFiles -> SharedFiles
 - Sal de tu sesión : My ULG Session [x]
 - Realiza el cuestionario 1.

Ejercicio 2 (20 minutos):

Esta actividad la realizáis en grupos de 1 persona, pero podéis colaborar con otros, podéis:

1. Entrar en la sesión de otros solo para chatear: pedir ayuda, dar ayuda, ...(sin compartir el acceso gráfico a la aplicación)
2. Permitir acceso libre a vuestra sesión para que os ayuden (Sesión compartida libre : cualquier puede acceder a vuestra sesión)
3. Acceder a las sesiones de otras personas reciprocas: para ello tenéis que permitir el acceso a vuestra sesión a personas que hayan permitido acceso.



Para cambiar de una opción a otro se utiliza el panel de control:



1. Non-Shared Session  	Sesión no compartida (se puede pasar a los estados compartido reciproca y compartida libre)	GR4 
2. Shared Session  	Sesión compartida reciproca (se puede pasar a los estados no compartido y compartida libre)	GR4 
3. Free Shared Session  	Sesión compartida libre (se puede pasar a los estados no compartido y compartida reciproca)	GR3  

La tercera columna indica el estado de compartición del grupo correspondiente. En ese estado además se puede ver el número de usuarios que están observando esa sesión:

GR2   2

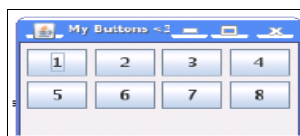
Y también se puede saber el número de sesiones que esta viendo los usuarios de cada grupo:

GR3  

- Rubén Mondejar   1

Los pasos de la actividad son:

1. Acceder al curso: Eclipse
2. Logéate con tu username: *tu nombre de pila*, y password: *primera letra de nombre y primera letra de apellido*.
3. Accede a tu sesión : My ULG Session [o]
4. Realiza la actividad con la aplicación Eclipse.
 - a. Abre el proyecto Java: Ejercicio 2.
 - b. Corrige el código de **ButtonsExample.java** para que salga una ventana con 8 botones como la siguiente:



- c. Comprueba que funciona ejecutando el proyecto con Run → Java Application
5. Publica tu solución desde el interface de control My ULG : MyFiles -> SharedFiles
 6. Sal de tu sesión : My ULG Session [x]
 7. Realiza el cuestionario 2

A continuación se muestran los resultados obtenidos en los cuestionarios, se mostrarán la media obtenida para las cuestiones cuantitativas (1-7) y todas las respuestas para las cuestiones cualitativas

Preguntas a responder		Respuestas
		1: Mínima calificación 7: Máxima calificación
En el primer ejercicio con la herramienta ULabGrid2:		
1	Comprendiste el ejercicio que tenias que realizar	4,75
2	Consideras que tenias los conocimientos previos necesarios para realizar ese ejercicio (calificación máxima = había realizado ejercicios similares)	4,5
3	Resolvisteis el ejercicio satisfactoriamente	6,5
4	Tu aportación en la resolución del ejercicio fue importante	4,5
5	Consideras que has aprendido a resolver ese tipo de problema	3,5
6	¿Qué dificultades has encontrado en la realización de este ejercicio?	<ul style="list-style-type: none"> - Nulos conocimientos de Java, delegación del control al compañero (2) - No recordar conceptos de Java (2) - Falta de conocimientos para utilizar la aplicación Eclipse (2) - Pocas indicaciones previas. Mejor si se hubiera practicado con la aplicación antes de la realización de los ejercicios - El delay del chat entorpecía la comunicación con los compañeros. - El chat no es el canal de comunicación más adecuado. Mejor si fuese audio.
7	Consideras que habéis adoptado una estrategia de grupo (coordinación, reparto de tareas...) adecuada para resolver el ejercicio.	4,63
8	¿Por qué?	<ul style="list-style-type: none"> - El compañero tenía más conocimientos y controló el sistema (2) - Quien tenía el control realizaba la tarea y el resto indicábamos posibles soluciones vía chat (2) - No lo hemos contemplado. - El ejercicio no era tan complejo para requerir este tipo de coordinación.

		<ul style="list-style-type: none"> - Al compartir entre todos el control, cada uno puede aportar todo lo que sabe. - Sí, la coordinación es mejor que el control concurrente.
9	Consideras que poder visualizar la sesión de manera concurrente ha sido útil.	4,88
10	Consideras que manejar el ratón de manera compartida es útil.	2,88
11	Consideras que el mecanismo para pedir y coger el control de la sesión es correcto.	4,25
12	Qué utilidades de ULabGrid2 han facilitado la realización de este ejercicio?	<ul style="list-style-type: none"> - El chat (4) - El escritorio compartido (4) - El repositorio común - NS/NC
13	¿Qué crees que es necesario mejorar en ULabGrid2 para poder realizar ejercicios en grupo de manera simultánea pero no co-localizados?	<ul style="list-style-type: none"> - Un sistema team-speak (2) - Disminuir el delay en el chat (2) - La interfaz no es demasiado clara - La pantalla de trabajo se reduce bastante. - Tener dos pantallas a la vez sería interesante. - Que los mecanismos para cambiar de dueño del ratón estuvieran en la misma pantalla que la aplicación que se comparte para que no fuese necesario salirse a otra pantalla. - Poder escribir cada uno lo suyo sin tener que esperar a que el que está escribiendo acabe. De esta forma, mientras uno escribe, el resto no hacen nada, por lo que pierdes recursos - Mayor agilidad en el chat y que asocie a cada usuario un color para llevar mejor la conversación - Indicadores de quién tiene el control en cada momento serían bastante deseables. - NS/NC

En el segundo ejercicio con la herramienta ULabGrid2:		
14	Comprendiste el ejercicio que tenias que realizar	5,75
15	Consideras que tenias los conocimientos previos necesarios para realizar ese ejercicio (calificación máxima = había realizado ejercicios similares)	5,25
16	Resolvisteis el ejercicio satisfactoriamente	4,25
17	Tu aportación en la resolución del ejercicio fue importante	4,25
18	Consideras que has aprendido a resolver ese tipo de problema	4,38
19	¿Qué dificultades has encontrado en la realización de este ejercicio?	<ul style="list-style-type: none"> - Errores en la herramienta (3) - Desconocimiento Java (3) - Dificultades técnicas - Interfaz poco claro - NS/NC
20	Consideras que habéis adoptado una estrategia de grupo (coordinación, reparto de tareas...) adecuada para resolver el ejercicio.	2,43
21	¿Por qué?	<ul style="list-style-type: none"> - No hubo colaboración (3) - No vi claro como colaborar (2) - Pedir ayuda por chat - Ofrecer ayuda por chat - NS/NC
22	¿A cuantas personas tuviste que pedir ayuda para resolver el ejercicio?	1,5
23	¿A cuantas personas ayudaste a resolver el ejercicio?	1
24	Consideras que poder pedir ayuda a otros compañeros con la opción "free-sharing" es útil	4,38
25	Consideras que poder acceder a la actividad de otros compañeros solo si compartes tu actividad es útil	4,86
26	Consideras que saber cuantas personas estaban viendo una sesión indica si es una sesión importante	3,86
27	¿Qué utilidades de ULabGrid2 han facilitado la realización de este ejercicio?	<ul style="list-style-type: none"> - Visualizar sesiones de otros compañeros (4) - El chat (3)

		<ul style="list-style-type: none"> - Repositorio Común - Ninguna - NS/NC
28	¿Qué crees que es necesario mejorar en ULabGrid2 para poder realizar ejercicios de manera simultánea con personas no co-localizadas?	<ul style="list-style-type: none"> - Eliminación de fallos - Mejora de la velocidad de ejecución - Aviso cuando te hablan por chat - Ver estado del resto de compañeros sin entrar en su sesión - Más aprendizaje de la herramienta - Mejora interfaz - 2 pantallas a la vez - NS/NC

5.2 ULabGrid2 integrado con la red social FaceBook

Finalmente, describimos un estudio experimental mediante el cual planeamos demostrar la habilidad del framework para soportar los escenarios para los que ha sido diseñado.

Para validar nuestro framework y prototipo desarrollamos un experimento con varios estudiantes. Programamos una actividad para estudiantes de Ingeniería Informática que siguió el siguiente proceso:

- Visualización de un vídeo explicativo del uso de la aplicación ULabGrid2 + FaceBook.
 - Este vídeo fue apoyado por las explicaciones pertinentes del profesor encargado de dirigir el experimento.
- A partir de este punto, los alumnos comenzaron a utilizar la aplicación y a interactuar entre ellos sin ninguna otra explicación.

El experimento se efectuó sobre un escenario compuesto por:

- 4 máquinas Linux con servidores VNC y aplicación ULabGrid2 instalados
- 2 máquinas Windows con la aplicación ULabGrid2 instalada.
- No se utilizaron nodos grid en el experimento.

Respecto a las aplicaciones configuradas en cada máquina para ser servidas cabe destacar:

- En cada máquina Linux se sirvieron dos aplicaciones, una en sesión individual y otra en sesión colaborativa.
 - Recordemos que desde Windows, la aplicación no sirve sesiones
 - Se dejó lanzado un ordenador en Linux sirviendo dos sesiones desde ULabGrid2, habiéndose lanzado con un usuario FaceBook desconocido para los alumnos que tomaron parte en el experimento.
 - Las aplicaciones servidas desde las máquinas Linux del laboratorio donde tuvo lugar el experimento fueron configuradas para servir las siguientes aplicaciones:
 - kolourpaint (Versión Linux del programa de tratamiento gráfico MS Paint)
 - pysol (Solitario de cartas en código abierto)
 - SimProcess (Simulador utilizado en la carrera de Ingeniería Informática)
 - Eclipse (Entorno de desarrollo integrado de uso muy extendido)

Estas aplicaciones fueron configuradas de forma que ninguna pareja app:tipo_sesión se dió más de una vez

- En el ordenador de usuario desconocido se sirvió la app Invaders (Versión en código abierto del mítico vídeo juego)

A continuación se muestran los cuestionarios que tuvieron que resolver los alumnos con los resultados obtenidos.

A continuación se muestran los resultados obtenidos en los cuestionarios, se mostrarán la media obtenida para las cuestiones cuantitativas (1-7) y todas las respuestas para las cuestiones cualitativas

Preguntas a responder		Respuestas
Sobre la primera sesión:		
1	¿Cuántos amigos tuyos tenía acceso a ULabGrid2?	2,16
2	¿Cuántas aplicaciones podías compartir?	0 (Windows) 2,25 (Linux)
3	¿Con cuantos amigos compartiste tus aplicaciones? ¿Por qué?	<ul style="list-style-type: none"> - Con ninguno porque usaba Windows (2) - Con uno porque era el número de amigos que tenía - Con dos porque era el número de amigos que tenía (3)
4	¿Cuántas aplicaciones podías acceder a través de ULabGrid2?	5
5	¿Qué aplicaciones usaste? ¿Por qué?	<p>Aplicaciones:</p> <ul style="list-style-type: none"> - Eclipse (6) - Kolourpaint (3) - Invaders (4) - Solitario (5) - SimProcess (5) <p>Razón:</p> <ul style="list-style-type: none"> - Eran todas las que aparecían (3) - Quería probar el funcionamiento de la aplicación (2) - NS/NC
6	¿Usaste alguna aplicación compartida? ¿Por qué?	<ul style="list-style-type: none"> - Eclipse (3). Para probar el funcionamiento de la aplicación - No (3). No vi la opción para acceder a aplicaciones compartidas
7	¿Con quien usaste una aplicación simultáneamente? ¿Cómo compartisteis el manejo de esa aplicación compartida?	<ul style="list-style-type: none"> - No la usé (3) - Con dos amigos (2). Petición de control y respuesta

		<ul style="list-style-type: none"> - Con dos amigos y personas que no eran mis amigos.
8	¿Invitarías a tus amigos a utilizar ULabGrid2? ¿Por qué?	<ul style="list-style-type: none"> - No. Hace falta más ancho de banda para utilizar este tipo de aplicaciones - No. Este tipo de aplicaciones pueden acarrear problemas de seguridad - Posiblemente, aunque temo que se le encuentren usos maliciosos. - Si, aunque dudo que la utilizaran (2) - Dependería de si encontrase una aplicación en la que realmente le encontrase una ventaja el utilizarlo.

Sobre la segunda sesión:		
9	Antes de comenzar la segunda sesión ¿has enviado alguna invitación de amistad por Facebook? ¿Por qué?	<ul style="list-style-type: none"> - Si, porque poder dar permiso de acceso a mis aplicaciones a personas que no eran mis amigos y poder ver más aplicaciones (4) - No. Ya tenía agregados a todos mis amigos previamente. - No lo haría porque probablemente las personas con las que quisiera compartir algo probablemente serían ya amigos míos.
10	¿Has añadido alguna aplicación para que sea accedido a través de ULabGrid 2? ¿Cual?	<ul style="list-style-type: none"> - No (3) - Sí (3) <p>Aplicaciones</p> <ul style="list-style-type: none"> - Kaffeine - /usr/bin/xeyes - Konsole
11	¿Has dado permiso a nuevos amigos para usar tus aplicaciones?	<ul style="list-style-type: none"> - No. Usaba Windows (2) - Sí (3) - No lo haría porque probablemente las personas con las que quisiera compartir algo probablemente serían ya amigos míos.
12	¿Has usado alguna aplicación nueva de UlabGrid2? ¿Cuál?	<ul style="list-style-type: none"> - Sí (6) <p>Aplicaciones</p>

		<ul style="list-style-type: none"> - SimProcess (1) - Xeyes (2) - Kaffeine (1) - Konsole (2)
Sobre el experimento:		
13	Puedes identificar algún escenario de uso de ULabGrid2 (en empresas, la universidad, etc.) donde consideres que sería muy útil su uso.	<ul style="list-style-type: none"> - Para sesiones magistrales donde el profesor hace algo y el resto ve su pantalla. - Parece útil para asistencia técnica remota (arreglar fallos, hacer actualizaciones, etc) (2) - En un aula para emplear colaborativamente algún programa con el que realizar un proceso de aprendizaje o ideas (2) - Para asistencia/tutoriales del uso de algún programa. - En la empresa no sería válido, por temas de confidencialidad.
14	¿Cuáles son, en tu opinión, las mejores características de ULabGrid2?	<ul style="list-style-type: none"> - Compartición sencilla de aplicaciones (3) - Integración con los amigos que ya tiene el usuario, funciona ligero - Trabajo colaborativo - NS/NC
15	¿Qué crees que es necesario mejorar en ULabGrid2?	<ul style="list-style-type: none"> - Virtualización de aplicaciones (o chroot) - Más flexibilidad en el tipo de aplicaciones a servir. - Se puede mejorar el uso de aplicaciones compartida. Por ejemplo, permitir que todos los usuarios puedan usar la aplicación usando una política de round-robin o similar. - Otras formas de seleccionar gente con la que compartir, como otras redes sociales o correos electrónicos (2) - La facilidad para seleccionar a quién se le permite el acceso o no. - Posibles problemas con la apertura de puertos, NAT, retardos, etc. - Usabilidad en Windows

6. Conclusiones

En esta sección se van a exponer las conclusiones obtenidas una vez terminado el proyecto.

En este proyecto hemos presentado, mediante ULabGrid2, nuestra aportación a la convergencia de las diferentes tecnologías de Internet con el fin de operar en un entorno colaborativo.

De este modo, ha quedado demostrado que los paradigmas descentralizados que Internet nos brinda, como son los sistemas P2P y Grid, junto con las herramientas Web 2.0, como son los sistemas de gestión de contenidos y las redes sociales, y con el protocolo VNC, pueden converger hacia un nuevo modelo en el que la interacción entre los usuarios es mayor facilitando así su colaboración.

Esta colaboración puede ser enfocada hacia un entorno educativo, como así fue orientado el proyecto inicial, ULabGrid, que dio lugar al desarrollo del proyecto presentado en esta memoria. En una educación cada vez menos centrada en la figura del profesor, herramientas como la que presentamos con el primer prototipo hacen más sencilla la transición hacia un aprendizaje P2P en el que los alumnos puedan aprender los unos de los otros apoyados por el aprendizaje colaborativo soportado por ordenador (CSCL) o e-Learning 2.0.

Del mismo modo, también hemos demostrado con el desarrollo de un segundo prototipo que la convergencia de estas tecnologías y herramientas Web 2.0 pueden aprovecharse en entornos no educativos, como puede ser el mundo empresarial, aportando múltiples ventajas como una gran escalabilidad y aportando una gran facilidad en la colaboración.

Con la realización de este proyecto se han cumplido varios objetivos:

- Integración del proyecto ULabGrid con gestores de contenidos y redes sociales
- Implantación en la herramienta de las funcionalidades necesarias para alcanzar varios escenarios objetivos como son la comunicación directa entre peers, el descubrimiento de peers y conexión con otros peers, y finalmente, el acceso compartido a sesiones de aplicación.
 - Aprovechamiento de las características de infraestructuras Grid: compartición de licencias y acceso a recursos tanto hardware como software.
 - Objetivos alcanzados inherentes a la integración de ULabGrid con el gestor de contenidos MOODLE: Compartir información, resultados homogéneos, computación transparente, familiarización con la herramienta a utilizar, aprendizaje a distancia...
 - Objetivos alcanzados inherentes a la integración de ULabGrid con la red social software FaceBook: Mayor alcance, mayor difusión, descentralización total, más posibilidades de encontrar nuevos peers con intereses comunes, más posibilidades de encontrar nuevas aplicaciones, acceso a sesiones individuales, servir sesiones propias.

Con el análisis de los resultados obtenidos en los experimentos realizados para comprobar el funcionamiento de los prototipos desarrollados se pueden extraer varias conclusiones:

- ULabGrid2 + MOODLE:
 - En un grupo de alumnos conocimientos medios sobre el ejercicio expuesto se obtuvo una solución al problema casi perfecta gracias a la eficacia del trabajo colaborativo en un ámbito educativo.
 - El análisis de resultados en el primer ejercicio indica que la colaboración entre los miembros de cada grupo se realizó de manera sencilla.
 - Los alumnos tuvieron una opinión muy favorable sobre la visualización concurrente de sesiones y los mecanismos implementados para los cambios de control. No tuvieron tal opinión sobre el control concurrente del puntero del ratón.
 - Herramientas de interacción como chats sirven de gran ayuda a la hora de utilizar este

tipo de herramientas.

- El segundo ejercicio fue realizado organizando los estudiantes de manera individual, esto provocó unos resultados mucho peores de los obtenidos cuando los estudiantes se organizaron en grupos, aunque la colaboración posible entre los usuarios independientemente de su organización, permitió que los resultados siguiesen siendo aceptables. Incluso los alumnos consideraron que su aprendizaje fue mayor.
 - Los alumnos consideraron muy útil la opción de compartir sesiones libremente con el resto de usuarios.
- ULabGrid + FaceBook
- Los usuarios compartieron las aplicaciones disponibles con todos sus amigos disponibles.
 - Se obtuvieron algunas opiniones de usuarios reacios a invitar a sus amigos a utilizar la aplicación por varios motivos: es necesario más ancho de banda, temor por posibles fallos de seguridad en este tipo de aplicaciones, etc.
 - Los usuarios identificaron varios posibles usos de la aplicación: apoyo a sesiones magistrales, asistencia técnica remota, aprendizaje colaborativo, etc.
 - La mayoría de los usuarios que utilizó la aplicación coincidió en que la mayor ventaja de ésta es la posibilidad de compartir aplicaciones de forma sencilla.

7. Líneas Futuras

En esta sección se van a mencionar brevemente posibles mejoras que se pueden aplicar al proyecto.

Integración de ULabGrid2 + MOODLE + FaceBook

En referencia a las posibles mejoras aplicables al proyecto que hemos desarrollado, destaca por encima de todas, la integración completa de los dos frameworks presentados, ULabGrid2 + MOODLE y ULabGrid2 + FaceBook, para formar un nuevo framework que podría ser denominado ULabGrid3, y que permitiría el trabajo conjunto entre los usuarios de ambas aplicaciones, pudiendo aprovechar las ventajas que brindan los dos prototipos desarrollados en una única aplicación.

En la siguiente figura se muestra un diagrama con que muestra una posible arquitectura del sistema.

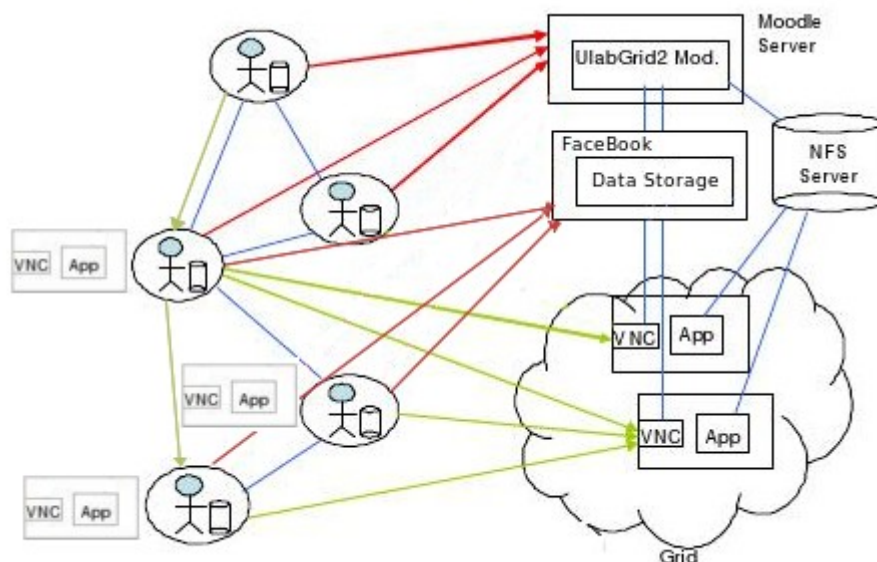


Figura 46. Posible framework de ULabGrid3

Hay algunos elementos que facilitan la integración de ambos frameworks, como es que en los dos sistemas se realizan conexiones a grid y lo más importante, el acceso a las sesiones de aplicación es completamente igual tanto para el prototipo que utiliza MOODLE como para el que utiliza FaceBook, es decir, los dos utilizan el protocolo VNC para el acceso a estas sesiones.

Posiblemente, sería necesario incluir algún elemento más de control, o una capa de middleware, que sea capaz de gestionar las sesiones que han sido iniciadas desde MOODLE o FaceBook. Recordemos que en el primer caso esta información se almacena en un base de datos del propio servidor que aloja a MOODLE y en el segundo, la información es almacenada utilizando para ello el Data Store de FaceBook.

También, sería necesario definir otros aspectos, como por ejemplo:

- Almacenamiento: hasta ahora se ha utilizado NFS y sería necesario decidir si debería ser accesible también por los usuarios de FaceBook ya que, aparentemente, lo más lógico sería contar con un repositorio común para todos los usuarios.
- Login: Definición de si es necesario hacer login en FaceBook y MOODLE o bastaría con acceder a uno de ellos, contando en ese caso con diferentes funcionalidades.
- Qué usuarios serían capaces de servir sesiones, hasta ahora sólo podían realizarlo los usuarios de ULabGrid2 + FaceBook que utilizan Linux, pero podría ser una funcionalidad extensible a los usuarios que acceden mediante MOODLE.

En definitiva, en esta sección únicamente se pretende lanzar la idea de la posible integración

de los frameworks desarrollados, pero sería necesario un análisis mucho más exhaustivo para llevarla a cabo.

Grabar sesiones VNC

El applet utilizado para visualizar las sesiones VNC, TightVNC, posee la funcionalidad de grabar las sesiones a las que se accede.

Esto sería de gran utilidad, sobre todo en un entorno educativo, ya que permitiría a los profesores comprobar cual ha sido el procedimiento seguido por los alumnos para la realización de los ejercicios planteados, en lugar de poder analizar únicamente los resultados finales.

Mejorar seguridad

Aunque la seguridad de la aplicación ha quedado fuera del alcance de este proyecto, sería conveniente analizar los posibles fallos para desarrollar las mejoras necesarias.

Una importante mejora que puede realizarse consiste en utilizar túneles SSL en todas las conexiones VNC.

También se podría incluir contraseñas de acceso a VNC, para determinadas sesiones a definir por el usuario que las lanza.

Mejorar sobrecarga del servidor

En el experimento para la comprobación del primer prototipo desarrollado se pudo comprobar que con el aumento de los usuarios el servidor se sobrecargaba rápidamente.

Esto se debe a la utilización de un gran número de objetos AJAX, encargados de realizar comunicaciones asíncronas con el servidor. Sería necesario replantear el funcionamiento de esta aplicación para realizar una utilización más eficiente de los objetos AJAX mencionados

Extensión a otros Sistemas Operativos

Aunque la versión de ULabGrid2 que utiliza MOODLE es accesible desde cualquier sistema operativo ya que está completamente basado en tecnologías web, hemos explicado que la versión de ULabGrid2 que utiliza FaceBook solo se puede ejecutar desde Linux y desde Windows, en este último caso contando con menos funcionalidades.

Sería interesante poder extender el uso de esta herramienta sobre MAC/OS y otros Sistemas Operativos.

Extensión a otros sistemas Web 2.0

Aunque se han utilizado el gestor de contenidos MOODLE y la red social FaceBook para la gestión de los usuarios en este proyecto, las posibilidades que Internet nos ofrece son innumerables.

Con el objetivo de conseguir mayor difusión se podría integrar el sistema con otras herramientas, como correos electrónicos, y especialmente sería interesante ampliar el rango de usuarios capaces de utilizar la aplicación a otras redes sociales.

8. Publicaciones

En esta sección se referencian las publicaciones generadas a raíz del desarrollo de este proyecto.

Durante el desarrollo de este proyecto fueron aceptadas dos publicaciones, ambas con relación a la integración de ULabGrid2 con MOODLE, que son adjuntadas con esta memoria.

A continuación se expone las referencias de ambas publicaciones.

– **Sharing Application Sessions for Peer-to-Peer Learning**

Autores: Oscar Ardaiz, Luis Manuel Diaz de Cerio, Jose Andres Del Campo (Public University of Navarra, Pamplona, Spain), Rubén Mondejar (Rovira y Virgili University, Tarragona, Spain)

Congreso: Interacción 2009. X Congreso Internacional de Interacción Persona-Ordenador. Barcelona, 7, 8 y 9 de Septiembre de 2009.

– **ULabGrid2 Project: P2P and Grid support for Peer-to-Peer Learning**

Autores: Oscar Ardaiz, Luis Manuel Diaz de Cerio, Jose Andres Del Campo (Public University of Navarra, Pamplona, Spain), Rubén Mondejar (Rovira y Virgili University, Tarragona, Spain)

Congreso: CSE 2009. 12th IEEE International Conference on Computational Science and Engineering (CSE 2009). 7th IEEE International Conference on Embedded and Ubiquitous Computing (EUC 2009). 2009 IEEE International Conference on Privacy, Security, Risk and Trust (PASSAT 2009). 2009 IEEE International Conference on Social Computing (SocialCom 2009). Vancouver, Canada, 29-31 August 2009

9. Bibliografía

En esta sección se referencian los artículos, publicaciones y páginas web utilizadas a la hora de recoger información.

- [1] D. Boud, R. Cohen, J.Sampson, Peer learning in higher education: learning from & with each other. Kogan Page. London. 2001.
- [2] Peer-to-peer learning <http://p2pfoundation.net/Category:Education> 2007.
- [3] N. White and J. Kapma “From workplace courses to global conversations” in *Leren in organisaties 8-6/7*, Rotterdam, jun 2008.
- [4] G. Siemens, “Connectivism: A Learning Theory for the Digital Age,” *elearnspace*, December 2004, <http://www.elearnspace.org/Articles/connectivism.htm>.
- [5] Red Española de e-Ciencia. Grid y Supercomputación al Servicio de la Comunidad Científica Española. <http://www.e-ciencia.es/>
- [6] Wikipedia en castellano, <http://es.wikipedia.org>
- [7] Facebook, <http://www.facebook.com>
- [8] Wikipeda en inglés, <http://en.wikipedia.org>
- [9] Stahl, G., Koschmann, T., & Suthers, D. (2006). Computer-supported collaborative learning: An historical perspective.
- [10] Ardaiz, O., Artigas, P., Diaz de Cerio, L., Freitag, F., Messeger, R., Navarro, L., Royo, D., Sanjeevan, K., D.ULabGrid, an infrastructure to develop distant laboratories for undergrad students over a Grid. In *Proceedings of the First European Across Grids Conference*, 2003.
- [11] Adasiewicz, C.: Exploratorium: User friendly science and engineering. *NCSA Access* 9, 2, 10–11
- [12] Souder, D., Herrington, M., Garg, R. P., and DeRyke, D.: JSPICE: A component- based distributed Java front-end for SPICE. In *Proceedings of the 1998 Workshop on Java for High-Performance Network Computing* (1998)
- [13] Reinefeld, A., Baraglia, R., Decker, T., Gehring, J., Laforenza, D., Ramme, F., Romke, T., and Simon, J.: The MOL project: An open, extensible metacomputer. In *Proceedings of the 1997 IEEE Heterogeneous Computing Workshop (HCW97)* (1997), pp. 17–31
- [14] Kapadia N.H., Figueiredo R.J., Fortes J.A.B.: PUNCH: Web Portal for Running Tools. *IEEE Micro*. May -June 2000.
- [15] Kapadia, N. H., Fortes, J. A. B., Lundstrom, M. S., Royo D.: PUNCH: A Computing Portal for the Virtual University. Forthcoming in a special issue on Virtual Universities and Engineering Education.
- [16] Foster I., Kesselman C., Nick J., Tuecke S.: Grid Services for Distributed System Integration. *Computer*, 35(6), 2002

- [17] Globus Toolkit v2, www.globus.org/toolkit, 2003.
- [18] Hung-ying Tyan, "J-sim a component-based, compositional simulation environment", www.j-sim.org, PhD Dissertation Ohio State University 2003.
- [19] Szalay, A., Gray, J.: The World-Wide Telescope. Volume 293, Number 5537, pp. 2037-2040. Issue of 14 Sep 2001.
- [20] Data Grid Project. <http://eu-datagrid.web.cern.ch/eu-datagrid>. 2003.
- [21] "Sironta", <http://www.sironta.com>
- [22] "eGroupWare", <http://www.egroupware.org>
- [23] "Be Smart- Cooperate Worldwide", <http://public.bscw.de>
- [24] "OpenGroupWare", <http://www.opengroupware.org/>
- [25] "Kolab", <http://www.kolab.org/>
- [26] "TalkAndWrite", <http://www.talkandwrite.com/>
- [27] "Synergiea", <http://bscl.fit.fraunhofer.de/>
- [28] "Open-Xchange", <http://www.open-xchange.com>
- [29] J. Cole. Using Moodle: Teaching with the Popular Open Source Course Management System. O'Reilly Community Press, Cambridge, 2005.
- [30] Moodle Pedagogical Approach <http://en.wikipedia.org/wiki/Moodle>, feb. 2009.
- [31] Campbell, A.P., Ammann, R., and Dieu, B. (2005). Elgg--A personal learning landscape. TESL-EJ, 9(2), <http://tesl-ej.org/ej34/m1.html>
- [32] M. Halm, B. Olivier, U. Farooq., C. Hoadley. "Collaboration in Learning Design Using Springer-Verlag. Berlin. 2005.
- [33] M Halm, A Valentine, LionShare: an academic Peer-to-peer community. <http://lionshare.its.psu.edu/> 2007.
- [34] Mcgreal, M. Nilsson, M. Nilsson, M. Nilsson, M. Nilsson The Edutella Peer-to-peer Network - Supporting Democratic E-learning and Communities of Practice Online Education Using Learning Objects, Routledge-Falmer, 2003.
- [35] M. Bardeen, E. Gilbert, T. Jordan, P. Nepywoda, E. Quigg, M. Wilde, and Y. Zhao. "The QuarkNet/Grid Collaborative Learning e-Lab". Future Generation Computer Systems 22 Elsevier, 2006, 700-708.

[36] Planetlab Courseware. <http://www.planet-lab.org/courseware> retrieved feb. 2009.

[37] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, “ The Eigentrust algorithm for reputation management in Peer-to-peer networks” In Proceedings of the 12th international Conference WWW '03. ACM, New York, NY, 2003, 640-651.

[38] Bote, M., Dimitriadis, Y., Gómez-Sánchez, E.: Grid uses and characteristics: a grid definition. In Proceedings of the First European Across Grids Conference, 2003.

[39] Kolb, D. A. Experiential Learning - Experience as the Source of Learning and Development, Prentice- Hall, New Jersey. 1984.

[40] O. Ardaiz, L. Diaz de Cerio, A. Gallardo, R. Messeguer, K. Sanjeevan, “UlabGrid Framework for Computationally Intensive Remote and Collaborative Learning Laboratories”, IEEE International Symposium on Cluster Computing and the Grid, 2004.

[41] Richardson T., Stord-Fraser T., Wood K.R., Hopper A.: Virtual network computing. IEEE Internet Computing, 2(1):33-38, January-February 1998.

[42] Data Store API, Documentación.
[http://wiki.developers.facebook.com/index.php/Data Store API documentation](http://wiki.developers.facebook.com/index.php/Data_Store_API_documentation)

[43] Facebook Developers, <http://www.facebook.com/developers>

[44] Facebook Developers, Documentación. <http://developers.facebook.com/docs/>

ULabGrid2

PFC Ingeniería Informática
UPNA 30 Junio 2010

Alumno: José Andrés del Campo

Tutores:
Luis María Díaz de Cerio
Óscar Ardaiz

ÍNDICE

- 1 – Introducción
- 2 – Estado del arte
- 3 – Objeto del proyecto
- 4 – Desarrollo
- 5 – Análisis de resultados
- 6 – Conclusiones
- 7 – Líneas Futuras
- 8 – Publicaciones

INTRODUCCIÓN

Origen del proyecto

- Aprendizaje P2P
- Internet y Web 2.0
- Conectivismo
- P2P y Grid
- ULabGrid → solución Grid a el trabajo colaborativo en la enseñanza
- ULabGrid2 → ámbito más general

INTRODUCCIÓN

Conceptos relacionados

- Red Social
- Gestor de contenidos (CMS)
- e-Learning
- CSCL
- Groupware

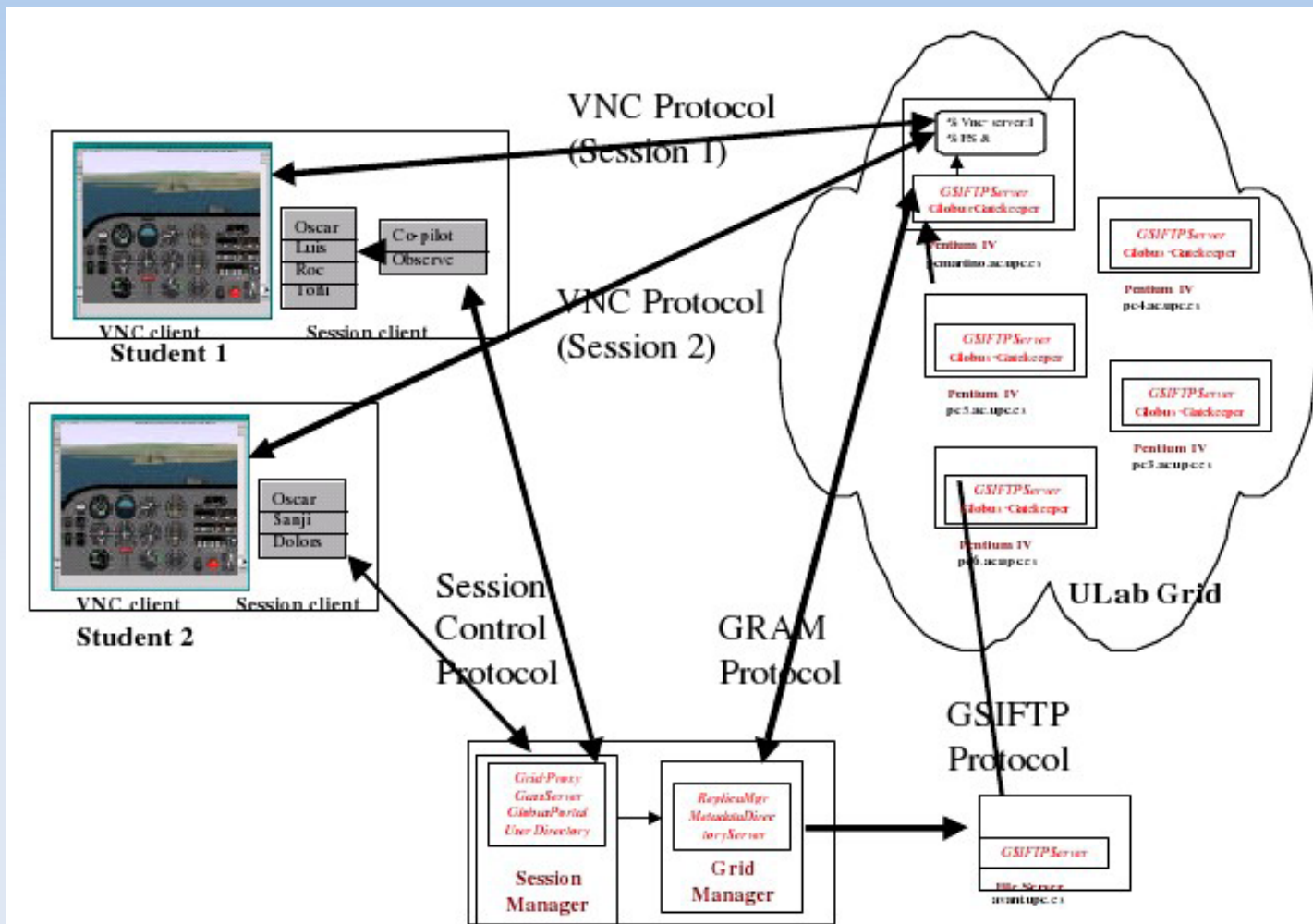
ESTADO DEL ARTE

ULabGrid

- Implementación y evaluación de App CSCL
- Globus Toolkit 2
- Componentes
 - Cliente usuario final
 - Colección de recursos
 - SW administrador de sesión
 - SW administrador de grid

ESTADO DEL ARTE

ULabGrid



ESTADO DEL ARTE

Otros trabajos relacionados

- PUNCH
- DataGrid
- Campus Virtuales
- Sirona
- MOODLE
- etc...

OBJETO DEL PROYECTO

Objeto General del proyecto

- Extender funcionalidades de ULabGrid
- Integración
 - Gestores de contenidos → MOODLE
 - Redes sociales → FaceBook
- Escenarios objetivos (CSCL / Groupware)
 - Comunicación directa entre peers
 - Descubrimiento de peers y conexión
 - Compartir acceso a sesiones de aplicaciones

OBJETO DEL PROYECTO

Objeto General del proyecto

- Grid / P2P
 - Instalación y mantenimiento de SW
 - Licencias
 - Recursos hardware
 - SW y HW compartido por múltiples organizaciones administrativas
 - Fácil acceso

OBJETO DEL PROYECTO

ULabGrid integrado con el CMS MOODLE

- Compartir información
- Reducir costes en cambios de contenidos (HW y SW)
- Conseguir resultados homogéneos
- Computación transparente
- Familiarización con la herramienta a utilizar
- Educación a distancia

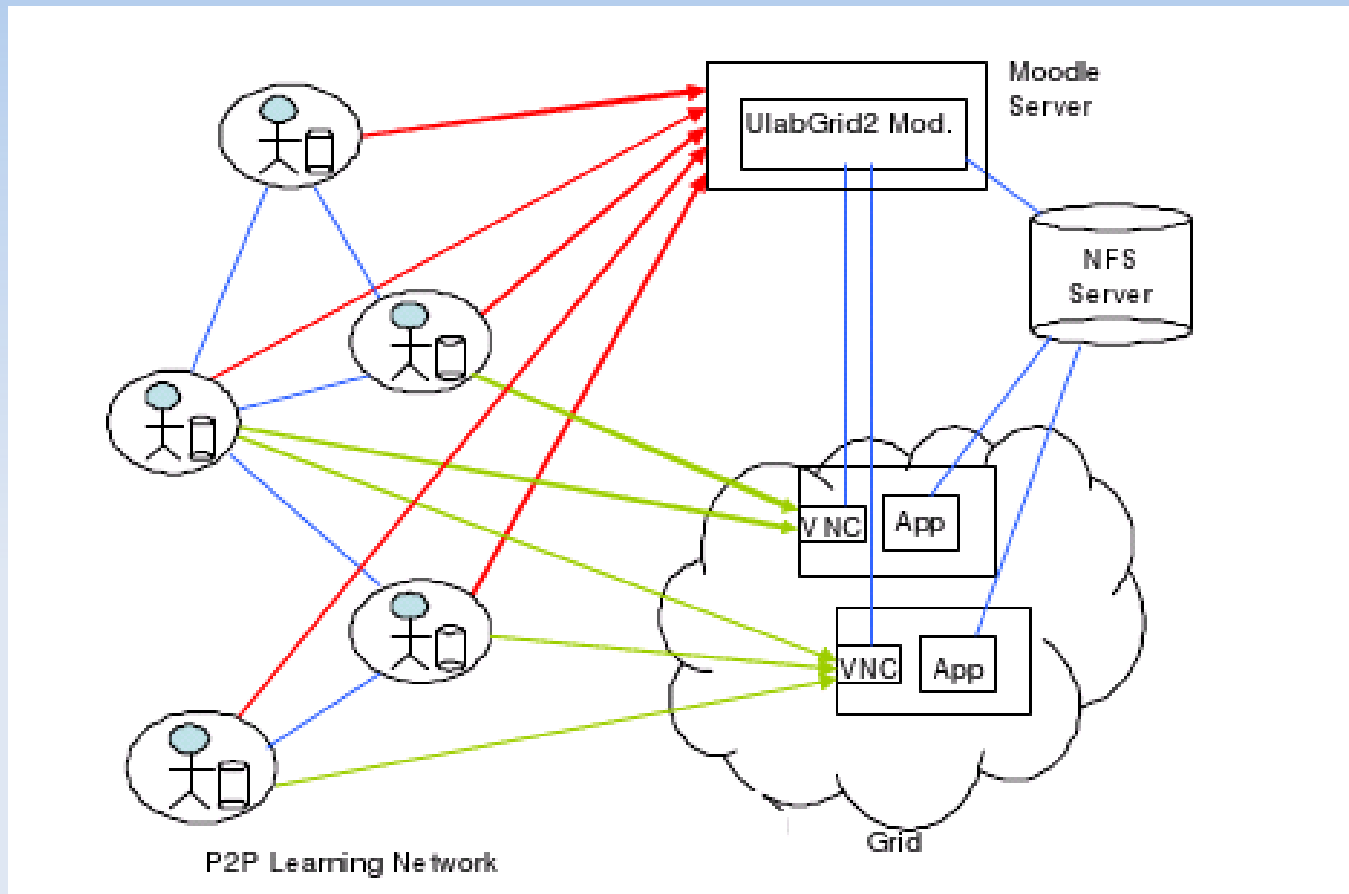
OBJETO DEL PROYECTO

ULabGrid2 integrado con la red social FaceBook

- Mayor alcance
- Mayor difusión
- Descentralización total
- Más posibilidades de encontrar nuevos peers con intereses comunes
- Más posibilidades de encontrar nuevas aplicaciones
- Acceso a sesiones individuales
- Servir sesiones propias

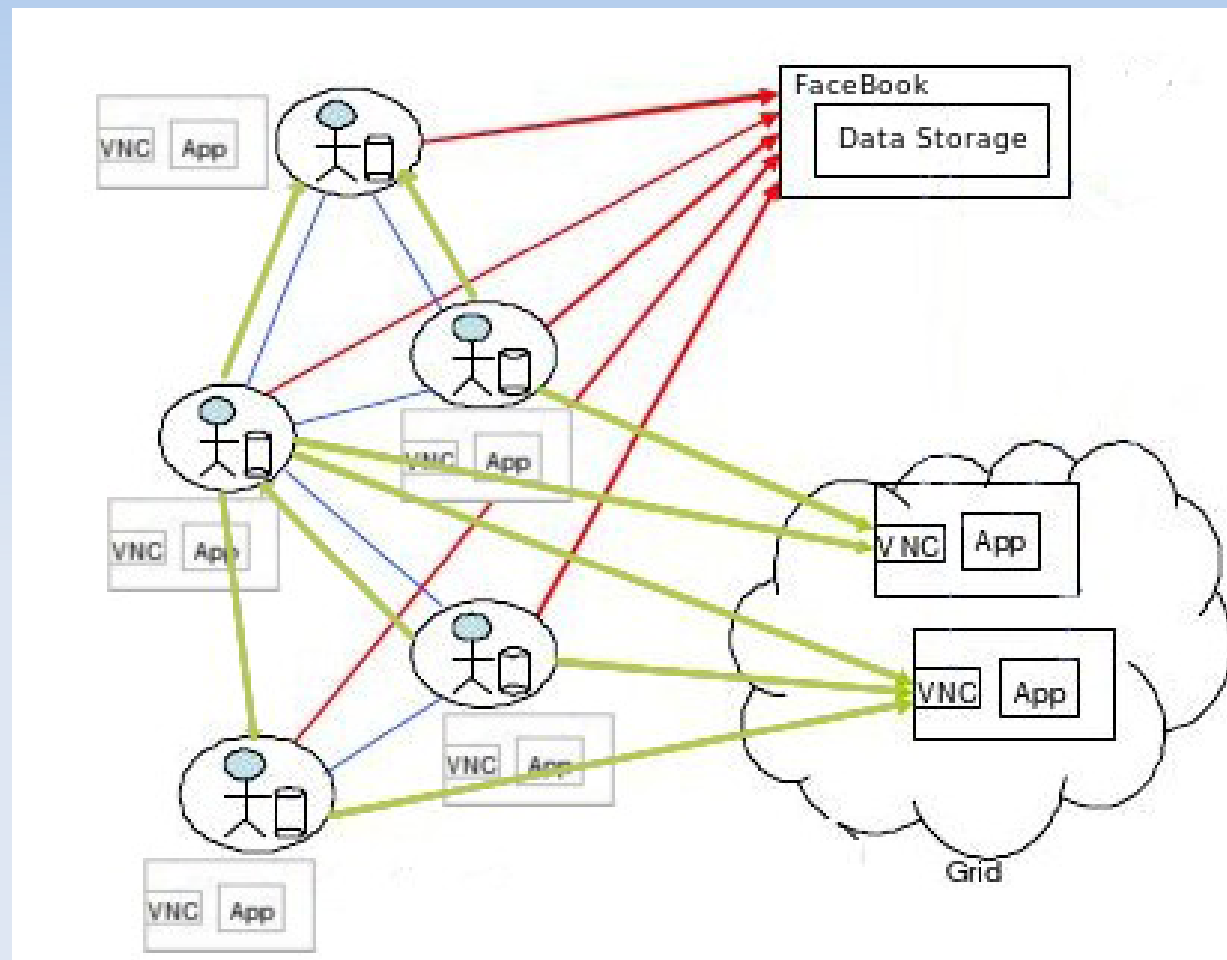
DESARROLLO

ULabGrid2 + MOODLE



DESARROLLO

ULabGrid2 + FaceBook



DESARROLLO

ULabGrid2 + FaceBook

- Modelo en el Data Store de FaceBook

Object Type: connection

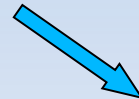
Property Name	Property Type
inicial	string

Object Type: ulabgriduser

Property Name	Property Type
ip	string
app	string
userid	integer
port	integer
estado	string
idgrupo	integer

Object Type: group

Property Name	Property Type
usuarios	string



DESARROLLO

Problemas encontrados

- Windows no sirve sesiones VNC en paralelo
- Aprendizaje de programación
 - Módulos en MOODLE
 - Aplicaciones en FaceBook
- Un único archivo de configuración por cada servidor VNC
- Restricciones seguridad del Applet
- DSFB

ANÁLISIS DE RESULTADOS

Experimento ULabGrid2 + MOODLE

- Ejercicio 1 → grupos
 - Estrategias de grupo
- Ejercicio 2 → individuales
 - Comunicación entre diferentes grupos
- En ambos ejercicios se consideró
 - Muy útil
 - Acceso compartido a sesiones
 - Chat
 - Menos útil
 - Repositorio común

ANÁLISIS DE RESULTADOS

Experimento ULabGrid2 + FaceBook

- Ejercicio 1 → Sin crear nuevos amigos
 - Fácil manejo de la aplicación en sesiones individuales
 - Más difícil en sesiones compartidas
- Ejercicio 2 → Creando nuevos amigos
 - Compartir nuevas aplicaciones
- Experimento
 - Reticencia a invitar a sus amigos a usar la aplicación
 - Es sencillo compartir aplicaciones

CONCLUSIONES

- P2P + Grid + Web 2.0 + VNC → entorno colaborativo
 - Educación (CSCL)
 - Empresa (Groupware)
- Objetivos cumplidos
 - Integración de ULabGrid
 - Gestor de contenidos
 - Red social
 - Escenarios objetivo alcanzados
 - Objetivos específicos alcanzados

CONCLUSIONES

Experimento ULabGrid2 + MOODLE

- Conocimientos medios → buen resultado
- Colaboración sencilla
- Opinión favorable
 - Visualización concurrente de sesiones
 - Mecanismos de cambio de control
 - Chat
 - Compartir sesiones del tipo “Free-shared”
- Opinión desfavorable
 - Control concurrente de los dispositivos de entrada

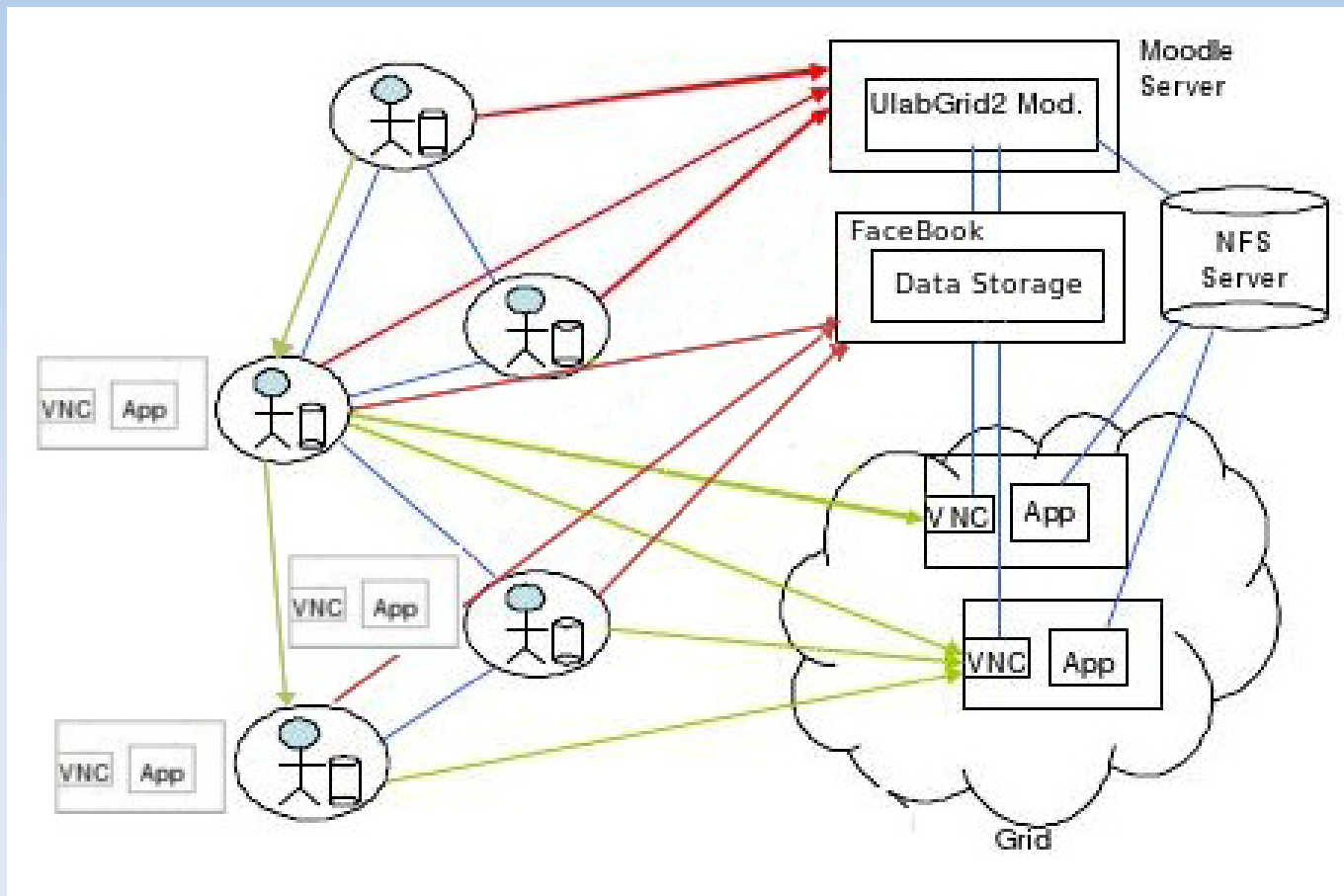
CONCLUSIONES

Experimento ULabGrid2 + FaceBook

- Los usuarios compartieron las aplicaciones con todos sus amigos
- Reticencia a invitar a amigos a utilizar la aplicación
- Los usuarios identificaron varios posibles usos
 - Apoyo a sesiones magistrales
 - Asistencia técnica remota
 - Aprendizaje colaborativo
- Permite compartir aplicaciones de forma sencilla

LÍNEAS FUTURAS

¿ULabGrid3?



LÍNEAS FUTURAS

Otras posibles mejoras

- Grabar sesiones VNC
- Mejorar seguridad
- Mejorar sobrecarga del servidor
- Extensión a otros Sistemas Operativos
- Extensión a otros sistemas Web 2.0

PUBLICACIONES

- **Sharing Application Sessions for Peer-to-Peer Learning.**
Autores: Oscar Ardaiz, Luis Manuel Diaz de Cerio, Jose Andres Del Campo (Public University of Navarra, Pamplona, Spain), Rubén Mondejar (Rovira y Virgili University, Tarragona, Spain) Congreso: Interacción 2009. X Congreso Internacional de Interacción Persona-Ordenador. Barcelona. 7, 8 y 9 de Septiembre de 2009.
- **ULabGrid2 Project: P2P and Grid support for Peer-to-Peer Learning.**
Autores: Oscar Ardaiz, Luis Manuel Diaz de Cerio, Jose Andres Del Campo (Public University of Navarra, Pamplona, Spain), Rubén Mondejar (Rovira y Virgili University, Tarragona, Spain) Congreso: 2009 IEEE International Conference on Social Computing (SocialCom 2009). Vancouver, Canada. 29-31 August 2009

ULabGrid2

FIN