Trabajo práctico de investigación

El trabajo práctico de investigación tiene varios objetivos:

- Fomentar en el alumno la investigación sobre aplicaciones y temas relacionados con GNU/Linux y software libre.
- Aprovechar los recursos existentes en internet y la gran cantidad de información disponible, siendo el alumno capaz de procesarla, interpretarla y comunicarla.
- Compartir con el resto de la clase el trabajo realizado por cada alumno. Dar una mirada al extenso mundo del software libre en diversos temas a elección de los propios alumnos.
- Evaluar el interés y las capacidades de los grupos de trabajo.

Grupos
El trabajo podrá ser individual o en grupos de dos y tres personas. Para el caso de los grupos, se debe dejar bien establecido qué subtemas estarán a cargo de cada uno de los participantes. Todos los participantes deberán tener un tema en particular para investigar y para presentar.

Presentación
El docente destinará días y horarios (dentro del horario de clases) para la presentación de cada trabajo frente a la clase. El tiempo de la presentación será entre 15 y 25 minutos, dependiendo del tipo de trabajo y la cantidad de participantes del grupo.
El docente intentará poner a disposición de los alumnos un proyector y una PC con GNU/Linux instalado, o en su defecto alguna distribución GNU/Linux live-cd suministrada por el grupo. Quien lo desee podrá traer su propio equipo.

La monografía
Deberá ser un trabajo original. No se permite la inclusión de documentación extensa de programas de software ni tutoriales o material disponible en Internet. Deberá poseer como mínimo las siguientes secciones: Carátula, índice y contenidos, introducción, desarrollo, conclusiones o resumen final y referencias.
Para el caso de grupos, cada integrante será responsable de una monografía con el subtema que le corresponda.

Metodología de trabajo
El trabajo se realizará fuera del horario de clases, pero se podrán realizar consultas al docente durante las mismas o por email. Se deberán ir presentando los avances realizados en el siguiente orden:

- Temario general, temas incluidos, alcance del trabajo y objetivos.
- Indice y contenidos de la monografía. Esquema de la presentación.
- Monografía y presentación en estado avanzado.
- Organización de la presentación: Temas incluidos y tiempos.
- Presentación y exposición del trabajo realizado.
- Entrega de material final.

El temario y el alcance de los temas debe ser aceptado por el profesor.

Abanico de temas
El docente brindará una gran variedad de temas para seleccionar, sin ambargo los alumnos podrán proponer otros tópicos, aceptados, o no, según criterio del profesor. La orientación, el alcance y la profundidad de cada tema se acordará previamente con el profesor. Todos los grupos o personas deberán tener temas diferentes (válido para los dos turnos del taller). Por lo tanto, cuando un grupo confirma con el docente el tema seleccionado, ese tema queda inhabilitado para ser realizado por otro grupo.
Es muy amplia la variedad de temas posibles: Programas para ingeniería, aspectos del kernel Linux, librerias para programación, lenguajes de programación, servidores, administración, seguridad, etc.

Licencia
El trabajo de poseer en forma explícita una licencia acorde al software utilizado (dentro de la monografía y dentro de la presentación). Se recomienda la licencia GNU FDL o la Creative Commons en alguna de sus variantes.

Formato de los archivos
La monografía y la presentación deberán realizarse en formato **ISO 26300 - OASIS (*Open Document Format for Office Applications) http://es.wikipedia.org/wiki/OpenDocument***
ADVERTENCIA: Trabajar en otros formatos y luego realizar la conversión de formatos puede llevar a resultados inesperados en la visualización del contenido. Podrán bajarse puntos de evaluación del trabajo si se detecta esta situación.

Material entregado
Para la aprobación final del trabajo se deberá entregar:

- Impresión reducida de la presentación realizada (diapositivas).
- Monografía impresa sobre el tema desarrollado.
- Un CDROM con todos los archivos involucrados y anexos, en formatos abiertos. La presentación y la monografía deberán estar además en formato pdf.

Evaluación
El trabajo práctico de investigación constará y será evaluado según los siguientes items:

- Investigación del tema seleccionado y procesamiento de la información obtenida.
- Monografía.
- Presentación oral frente a la clase, asistida con presentación en proyector y PC.
- Documentación entregada en CD (formatos OpenDocument y pdf correctos).

En el caso de un grupo, se tomará en cuenta la performance global del mismo, pero la nota será evaluada en forma individual para cada uno de los alumnos, según la monografía y la exposición en clase. La nota promedia con los parciales.

Listado de temas sugeridos
Estos son algunas ideas. Se privilegian en la evaluación herramientas y software asociados y aplicados a las ingenierías. El tema se debe acordar y negociar con el encargado de trabajos prácticos.

1. Investigación sobre licencias de sofware propietario, open source y free software. Casos reales. Modelo de negocios adoptados por las empresas de software libre u open source. Legales.
2. Desarrollo del kernel. Versiones 2.4 y 2.6. Configuración del kernel. Plataformas soportadas. Estructura. Funcionamiento, etc.
3. Utilización de la librerías para facilitar la labor del programador. Ejemplo: libc, gtk, fltk, ncurses, turbovision, aalib, readline, slang, etc, etc. Selección de alguna en particular, características, tutorial, ejemplos.
4. Scripts avanzados en BASH . Herramientas auxiliares, ejemplos, etc.
5. Lenguajes de programación mas utilizados. C/C++, Perl, Python, Bash, Tcl/tk, etc. Ejemplos básicos de uso y demostración.
6. Herramientas para ingenieria. Programas de cálculo, visualización y análisis de datos, simulación, etc. Presentación y ejemplos de uso.
7. Seguridad, hardening, acceso remoto, características del ssh y telnet, etc.
8. Administradores de ventanas, de sesión y escitorios. Recursos consumidos, dependencias, utilización básica, capacidades, ventajas y desventajas de cada uno, historia del desarrollo, etc.
9. Servidor web. Características, instalación y configuración, modulos, etc.
10. Impresión. Distintos sistemas de impresión. Arquitectura cliente-servidor. Filtros, ghostscript, etc.
11. Herramientas para administración de paquetes: Instalación, búsqueda, actualización, obtención de información, descompresión, etc. Cada distribución es un tema distinto. Las posibilidades: Debian, Fedora, Mandriva y Gentoo.
12. Armando una distribución live-cd a medida. Herramientas y consideraciones. Ejemplo práctico.
13. Juegos open source y free software. Origen, historia y desarrolladores, lenguaje utilizado, recursos necesarios. Aspectos de la instalación. Demostración.
14. Alternativas de bases de datos. Diferencias, ventajas y desventajas de cada una. Licencia, origen e historia. Lenguajes de consulta, particularidades de c/u y demostración básica.
15. Herramientas para desarrollo de software. IDEs, debuggers, parches, cvs, etc.
16. Aplicaciones de consola. Como hacer de todo desde la consola. Ejemplos y demostración.
17. Reproductores multimedia. Patentes de software e inconvenientes asociados a los MP3, MPEG4, DVD, DIVX, Realplayer, codecs, Macrovision, etc. Soluciones con Lame, mplayer, y otros. Rippers, burn tools, etc. Aspectos legales. Demostración.
18. Programación básica en lenguaje C en sistemas GNU/Linux. Utilización de dispositivos, ejemplos de e/s, etc.
19. Herramientas de consola para convesrión de formatos. Formatos de audio: mp3, wav, ogg. Formatos de imágen: jpg, png, xpm, gif, etc. Otros: postscript, eps, pdf, doc, etc. Investigar otros y opciones programas para X.
20. Administración, configuración y mantenimiento de un sistema GNU/Linux con muchos usuarios
21. Recursos y herramientas en internet para programadores de software libre. Sitios mas populares, importantes y centrales. Que hay detrás de cada uno.