

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 10



TÍTULO:

**PERSONALIZACIÓN DE NOVA PARA COMPETENCIAS
DE PROGRAMACIÓN EN LA EDUCACIÓN CUBANA**

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

Autores

Ana María Núñez Cuadrado.
Mariannis Berovides Hernández.

Tutor

Ing. Santiago Almira Ramírez.
Lic. José Jorge Lorenzo Vila

Consultante

Msc. Tomás López Jiménez.

Ciudad de la Habana, Julio de 2008

“Año 50 de la Revolución”

"La conciencia propia y el orgullo de la independencia garantizan el buen ejercicio de la libertad."

José Martí

DECLARACIÓN DE AUTORÍA

Por este medio declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ___ días del mes de ____ de 2007.

Ana Maria Nuñez Cuadrado

Mariannis Berovides Hernández

Lic. Santiago Almira Ramírez.

Lic. José Jorge Lorenzo Vila.

AGRADECIMIENTOS

A nuestros familiares y amigos.

A Dovier por ser el artífice del tema, impulsarnos y aportarnos su vasta experiencia.

A nuestro profesor y amigo: Tomás López Jiménez.

A nuestros “gurús”: Micha, Anielkis y Migue, por ser la mejor fuente de conocimientos sobre Nova, gracias por su paciencia.

A todos los del laboratorio Nova: Los “pelús”, los que ya no lo son, y a los que nunca lo fueron.

Al “Pillo” por su esfuerzo como diseñador.

A Marilú y Anmaris por hacer de un simple documento una tesis.

A los diseñadores, nuestros compañeros de aula, a nuestros profes y tutores.

Mariannis y Ana.M.

DEDICATORIA

A mi mamá y mi papá por ser lo mejores padres del mundo, por el amor y el apoyo que siempre me han brindado. A mi hermanito querido.

A mi tía Luly por su paciencia y ayuda, a Diana, a Pepe, por estar ahí cada vez que los necesité. A mi abuelita Juana, por el amor y la dedicación de todos estos años. A Chichi. A Mima y a tía Fella, por el cariño que siempre me dieron. A tía Yaima, a Camilo, a Niño y a Kamilita, por el apoyo incondicional que siempre me han dado. A Osval, a Isabel y a Anabel. A todos mis familiares que de una forma u otra me han permitido llegar hasta aquí. A Dayi y a Mari por tener la paciencia de aguantarme todos estos años. A Mauro. A los pelus, por marcar una etapa de mi vida. A mis compañeros de aula y del proyecto por el apoyo de todos estos años.

Ana

A Dios que me dio todas estas personas a quien dedicarle este sueño:

A mi familia por ser la mejor:

Mi Mamá, mi Papá, mi hermana y amiga Anmarita.

Mis únicas Abuelitas Tere y Pancha, El clan de la Vigía, por todo su amor y apoyo.

Mi abuelo Tirso que no está, pero sin dudas estuviese muy orgulloso.

A los que tengo lejos, y llevo muy presente.

A mi novio Pablo, por motivarme y apoyarme en ser cada día mejor para forjar una familia desde los cimientos más sólidos del amor y el conocimiento.

A Yandry por enseñarme el valor de las competencias de programación.

A todos los del laboratorio 7.

A los amigos de todos estos años.

A Ana por acompañarme hasta esta última tarea Universitaria.

M@rianitta.

RESUMEN

Este trabajo describe el desarrollo de la personalización de una **distribución libre**, que satisface las necesidades de un sistema para competencias de programación que cumpla con las normas internacionales.

Se persigue brindar un entorno cubano capaz de solucionar las necesidades en esa esfera de la enseñanza y que a su vez facilite el proceso de preparación y competición de los estudiantes.

La **distribución** tomada como punto de partida para el desarrollo de la personalización es Nova, debido a las múltiples ventajas que ofrece y con el propósito de extender el trabajo realizado por el equipo de desarrollo residente en la Universidad de las Ciencias Informáticas hacia esa esfera.

Como resultado final se obtiene una imagen **cliente** y una imagen **servidor**. La primera contiene herramientas que permiten programar en los lenguajes C, C++, Java y Pascal y la segunda, permite calificar los ejercicios a través de un jurado o sistema de control.

Palabras Claves: Personalización, Nova, Competencias de Programación.

AGRADECIMIENTOS.....	I
DEDICATORIA.....	II
RESUMEN	III
INTRODUCCIÓN	1
CAPÍTULO 1.....	6
FUNDAMENTACIÓN TEÓRICA DEL TEMA	6
1.1 - ANÁLISIS DE LA EVOLUCIÓN DE LAS COMPETENCIAS INTERNACIONALES DE PROGRAMACIÓN.	6
1.1.1- <i>Evolución de las competencias de programación en la IOI. Caracterización de los componentes de software usados.</i>	6
1.1.2- <i>Evolución de las competencias de programación en la ACM-ICPC. Caracterización de los componentes de software usados.</i>	10
1.2 - EVOLUCIÓN DE LAS COMPETENCIAS NACIONALES DE PROGRAMACIÓN.	12
1.2.1- <i>Las Competencias Nacionales de Programación en la enseñanza Media Superior.</i>	12
1.2.2 - <i>Las Competencias Nacionales en la Enseñanza Superior.</i>	13
1.3 - NOVA BASE PARA LAS COMPETENCIAS DE PROGRAMACIÓN.	18
1.4 - LOS PROGRAMAS Y HERRAMIENTAS USADOS EN COMPETENCIAS QUE SON COMPATIBLES CON LA DISTRIBUCIÓN NOVA.....	20
1.4.1– <i>Navegadores.</i>	20
1.4.2 – <i>Compiladores.</i>	21
1.4.3 – <i>Depuradores.</i>	21
1.4.4- <i>Entornos de Desarrollo Integrado (IDEs).</i>	21
1.4.5- <i>Editores de Texto.</i>	22
1.4.6 – <i>Diccionario.</i>	22
1.4.7 – <i>Sistemas de Control</i>	22
1.5 - METODOLOGÍA PARA LA GESTIÓN Y CONSTRUCCIÓN DE LA PERSONALIZACIÓN DE LAS IMÁGENES.	24
1.5.1- <i>Metodologías Ágiles</i>	25
1.5.2- <i>Ingeniería de Liberaciones</i>	32
CAPÍTULO 2.....	34
PERSONALIZACIÓN DE LA IMAGEN CLIENTE Y SERVIDOR.....	34
2.1- CONSTRUCCIÓN DE LA IMAGEN BASE Y COMPILACIÓN DEL KERNEL.	34
2.2- PRIMER SPRINT. CONSTRUCCIÓN DEL SISTEMA Y COMPILACIÓN DE BINARIOS.....	34
2.2.1- <i>Plannig del Primer Sprint.</i>	34

2.2.2- <i>Development</i>	37
2.2.3- <i>Postgame</i>	49
2.3- SEGUNDO SPRINT. CONFIGURACIÓN DE LA IMAGEN.	49
2.3.1- <i>Plannig del Segundo Sprint</i>	49
2.3.2- <i>Development</i>	50
2.3.3- <i>Postgame</i>	53
2.4- TERCER SPRINT. OPTIMIZACIÓN DE LAS IMÁGENES.	53
2.4.1- <i>Plannig del Tercer Sprint</i>	54
2.4.2- <i>Development</i>	54
2.4.3- <i>Postgame</i>	55
CAPÍTULO 3	57
CANDIDATO DE LIBERACIÓN	57
3.1- REQUERIMIENTOS DE HARDWARE. INSTALACIÓN.	57
3.2- RELEASE DE NOVA PARA COMPETENCIAS DE PROGRAMACIÓN.	58
3.3- CARACTERÍSTICAS DE LA IMAGEN CLIENTE.	59
3.4 – CARACTERÍSTICAS DEL SISTEMA DE CONTROL DE LA IMAGEN SERVIDOR.	60
CONCLUSIONES	62
RECOMENDACIONES	63
REFERENCIAS BIBLIOGRÁFICAS	64
BIBLIOGRAFÍA	65
ANEXOS	68
GLOSARIO DE TÉRMINOS	76

ÍNDICE DE FIGURAS

FIGURA 1: ESTADO COMPARATIVO DEL COMPORTAMIENTO DE LA PARTICIPACIÓN EN COMPETENCIAS ENTRE LOS AÑOS 1997 Y 2005	11
FIGURA 2: DISTRIBUCIÓN DEL SISTEMA DE CONTROL PC2.....	23
FIGURA 3: PASOS DE LA INGENIERÍA DE PARA EL DESARROLLO DE LA PERSONALIZACIÓN	33
FIGURA 4: IMAGEN DE LA CONFIGURACIÓN DEL ARCHIVO FSTAB	48
FIGURA 5: JERARQUÍA Y CONFIGURACIÓN DE UN LABORATORIO CON LA PERSONALIZACIÓN DE NOVA PARA COMPETENCIAS DE PROGRAMACIÓN	58

ÍNDICE DE TABLAS

TABLA 1: COMPONENTES DE SOFTWARE USADOS EN IOI EN EL PERÍODO 1999-2007	8
TABLA 2: COMPONENTES DE SOFTWARE USADOS EN LA COPA VOID.....	14
TABLA 3: COMPONENTES DE SOFTWARE USADOS EN LA COPA ESTUDIANTIL POR INVITACIÓN	17
TABLA 4: PRODUCT BACKLOG DEL PRIMER SPRINT DE LA IMAGEN CLIENTE	35
TABLA 5: PRODUCT BACKLOG DEL PRIMER SPRINT DE LA IMAGEN DEL SERVIDOR	36
TABLA 6: PAQUETES NECESARIOS INSTALADOS EN LA IMAGEN CLIENTE	43
TABLA 7: RELACIÓN ENTRE LOS PAQUETES Y LOS DIRECTORIOS	46
TABLA 8: VERSIONES DE CADA UNA DE LAS HERRAMIENTAS INSTALADAS EN LA IMAGEN CLIENTE	46
TABLA 9: VERSIONES DE CADA UNA DE LAS HERRAMIENTAS INSTALADAS EN LA IMAGEN SERVIDOR	47
TABLA 10: PRODUCT BACKLOG DEL SEGUNDO SPRINT DE LA IMAGEN CLIENTE.....	50
TABLA 11: PRODUCT BACKLOG DEL TERCER SPRINT DE LA IMAGEN CLIENTE	54
TABLA 12: REPRESENTA LOS REQUERIMIENTOS MÍNIMOS Y RECOMENDADOS PARA LA IMAGEN CLIENTE	57
TABLA 13: REPRESENTA LOS REQUERIMIENTOS MÍNIMOS Y RECOMENDADOS PARA LA IMAGEN SERVIDOR	57

INTRODUCCIÓN

Hoy en el mundo se le ha conferido una gran importancia a las competencias de programación, que son eventos regidos por reglas y normas donde compiten estudiantes evaluándose las habilidades que poseen de programación en el desarrollo de algoritmos eficientes para resolver un problema en un plazo de tiempo, mediante el uso de herramientas.

Las competencias de programación se efectúan a nivel internacional, regional y nacional, pues enriquecen y potencian la mente humana a través del estudio y la práctica de las ciencias computacionales.

Las competencias fomentan la creatividad, el trabajo en equipo, y la innovación en la construcción de nuevos programas de software, y le permite a los estudiantes poner a prueba su capacidad para llevar a cabo proyectos en este campo.

En el ámbito internacional las competencias de programación más relevantes son: la Olimpiada Internacional de Informática (IOI) y la Competencia Internacional de Programación de la International Collegiate Programming Contest (ACM-ICPC), las que se toman como referente para el desarrollo de esta investigación.

La Olimpiada Internacional de Informática es una de las Olimpiadas Internacionales de Ciencias, que tiene como objetivo potenciar el aprendizaje de la informática en estudiantes entre doce y veinte años. Los problemas de esta competición están orientados a los algoritmos. No se requiere amplio conocimiento en cuanto al uso del Application Programming Interface (**API**).^{API}, pero sí habilidad en la resolución de problemas y en el diseño de algoritmos.

Por su parte, la Competencia Internacional de Programación de la International Collegiate Programming Contest (ACM-ICPC), está encaminada a medir las habilidades de los mejores programadores del mundo académico y universitario. Vale destacar que se ha convertido en la Competencia de Programación más antigua y prestigiosa del mundo, siendo el foco de atención de las compañías y universidades más importantes del orbe, para ofrecer oportunidades de trabajo, becas y premios económicos.

API- Interfaz de Programación de Aplicaciones: Conjunto de convenciones de programación que definen cómo se invoca un servicio desde un programa.

En el caso de Cuba, las competencias de programación han sido un tema de continuo perfeccionamiento desde la década de los 80. Estos eventos competitivos van dirigidos a la enseñanza media superior y superior.

El sistema de competencias, le permite a los estudiantes cubanos participar en los concursos de conocimientos a nivel de base, municipal, provincial y nacional. Luego de este último evento se conforma la preselección nacional, donde se imparte una preparación rigurosa durante varias semanas, con el objetivo de hacer una estricta selección para que represente al país en las Olimpiadas Internacionales.

En las competencias de programación, se requiere de varios parámetros, la organización del evento, las reglas de la competencia, el área y las herramientas donde se implementará la habilidad que compete. Resumiéndolo a la esfera de la informática se requiere de un hardware y software para competir en la habilidad de programar.

En la competencia juega un importante papel el entorno donde el competidor ejecuta o lleva a cabo la implementación de sus conocimientos, siendo esta otra de las denominaciones que se utilizan para definir el sistema operativo en el que se trabaja. Está compuesto fundamentalmente por aplicaciones como compiladores, editores de texto, navegadores, entornos de desarrollo, documentación y el jurado online.

En Cuba, en la competencia objeto de esta investigación, se ha utilizado en la mayoría de los casos Windows como sistema operativo, sin existir un entorno propiamente cubano para este tipo de evento.

En la Universidad de las Ciencias Informáticas (UCI) se han efectuado eventos como la Copa Void de Programación y la Competencia Estudiantil por Invitación de Programación (CEIP-UCI), que son competencias de lógica y algoritmos. En ellas, se han presentado algunas versiones de Nova, distribución cubana basada en la meta distribución **Gentoo** y creada por estudiantes de esa universidad, específicamente de la Facultad de Software Libre.

A la distribución Nova, se le han realizado diferentes personalizaciones de acuerdo a las necesidades, dentro de las que se encuentran las creadas para las competencias ya mencionadas. El hecho de que ninguna de estas versiones se conformó a partir de una investigación previa, y que dichas imágenes no trascienden las fronteras de la Universidad, se origina el **problema** de esta investigación: Se carece

de un entorno cubano que rebase los límites de la UCI, resultante de un proceso de investigación y desarrollo formalizado, para soportar competencias de programación en Cuba.

Siendo el **objeto de estudio** de este trabajo: las competencias de programación.

Se determinó como **objetivo general**: Configurar y personalizar los componentes de la distribución Nova, sistema operativo libre, cubano y altamente adaptable para competencias de programación.

La investigación determina como el **campo de acción** a las herramientas utilizadas en las competencias de programación.

Para el desarrollo de la investigación se plantea responder a las **preguntas científicas** siguientes:

- ¿Qué herramientas han sido utilizadas en los eventos internacionales de la IOI y la ACM-ICPC?
- ¿Qué herramientas se han usado en los eventos nacionales de programación?
- ¿Qué herramientas son compatibles con la Distribución Nova?
- ¿Qué metodología usar para la gestión de la personalización?
- ¿Cómo desarrollar la imagen cliente y servidor?
- ¿Cómo optimizar la personalización?

Para darle cumplimiento al objetivo de la investigación y resolver la problemática, se elaboran las siguientes **tareas**:

- Analizar la evolución de las competencias internacionales de programación de la IOI y la ACM-ICPC.
- Analizar la evolución de las competencias nacionales de programación.
- Analizar Nova Base para las Competencias de Programación.
- Identificar los programas y herramientas compatibles con la distribución Nova.
- Determinar los componentes de software esenciales para la personalización.

- Determinar la metodología a usar para la gestión de la personalización.
- Personalizar una imagen cliente y un servidor con los componentes y la metodología investigada.
- Optimizar la personalización.

Se desarrolla la investigación según los Métodos Teóricos y Empíricos.

Métodos Teóricos:

Histórico-Lógico, para estudiar la evolución y desarrollo de las herramientas utilizadas en las competencias nacionales e internacionales de programación.

Analítico – Sintético, para realizar un análisis de cada una de las herramientas utilizadas en los entornos de las competencias en el ámbito nacional e internacional. Luego se puede determinar el nexo que existe entre ellas, así como la compatibilidad con la distribución Nova.

Métodos Empíricos:

La entrevista, para conocer especificidades de las competencias de programación realizadas en Cuba, como están organizadas y que herramientas han sido utilizadas en ellas.

La encuesta, con el objetivo de obtener un estado de opinión de la personalización presentada en competencias y en base a ello, trabajar en próximas versiones.

El cuerpo de este trabajo está compuesto por la Introducción, Capítulos 1, 2 y 3, Conclusiones, Recomendaciones, Referencias bibliográficas, Glosario de términos y Anexos.

En el **Capítulo1: “Fundamentación Teórica del Tema”**, se realiza un análisis del estado del arte de las competencias de programación; además de un estudio preliminar de cada una de las herramientas utilizadas por la ACM-ICPC, la IOI, y las competencias nacionales incluyendo la Universidad de las Ciencias Informáticas. Se analiza también un grupo de metodologías con el objetivo de seleccionar la más adecuada para realizar la gestión de la personalización.

En el **Capítulo2: Personalización de la Imagen Cliente y Servidor**” se desarrolla, a partir de la estructura que propone la metodología seleccionada, la personalización de las imágenes cliente y servidor.

En el **Capítulo3: “Candidato de Liberación”** se presenta el producto y se explican sus características principales.

CAPÍTULO 1

FUNDAMENTACIÓN TEÓRICA DEL TEMA

En este capítulo se ofrece un estudio de la evolución de las competencias internacionales y nacionales de programación desde el punto de vista técnico. Se realiza un análisis de las herramientas que han sido utilizadas en estos eventos y su compatibilidad con la distribución Nova, se escoge además la metodología para proceder con el desarrollo de la personalización.

1.1 - Análisis de la evolución de las competencias internacionales de programación.

Las competencias internacionales que se toman como referencia para esta investigación, son la IOI y ACM-ICPC, cada una representa un nivel de enseñanza diferente, medio-superior y superior respectivamente. En primer lugar se estudia la cronología de diferentes ediciones de la IOI, enfocando el trabajo a la comisión técnica. Posteriormente se presenta un análisis de la ACM-ICPC, así como sus patrocinadores.

1.1.1- Evolución de las competencias de programación en la IOI. Caracterización de los componentes de software usados.

La idea de iniciar olimpiadas del saber en el campo de la informática fue propuesto en la XXIV Conferencia General de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO) en octubre de 1987, dando lugar a la primera edición de la Olimpiada Internacional de Informática en mayo de 1989 con la asistencia de la UNESCO, y desde ese momento se han realizado 18 ediciones de esta con los siguientes objetivos:

- Fomentar el intercambio de conocimientos y experiencias entre los estudiantes y docentes del área de la computación.
- Identificar a los jóvenes que demuestren mayor talento y aptitudes en este campo para orientarlos y apoyarlos en su formación.
- Promover el uso y aplicación de la informática como herramienta adecuada para la investigación y la resolución de problemas, como un elemento válido para el desarrollo humano.

- Promover un mejor conocimiento y comprensión de la ciencia y la tecnología.

Las primeras competencias efectuadas a partir del año 1989 tuvieron lugar en:

- IOI 1989: Pravetz, Bulgaria
- IOI 1990: Minsk, Bielorrusia
- IOI 1991: Atenas, Grecia
- IOI 1992: Bonn, Alemania
- IOI 1993: Mendoza, Argentina
- IOI 1994: Suecia
- IOI 1995: Holanda
- IOI 1996: Vezprem, Hungría
- IOI 1997: Ciudad del Cabo, Sudáfrica
- IOI 1998: Setúbal, Portugal

De esas primeras competencias no se han encontrado en el sitio oficial de las olimpiadas, características desde el punto de vista técnico. Por tanto, se toman como referencia para esta investigación las desarrolladas en el período 1999-2007, como se muestra en la Tabla1.1 que aparece a continuación.

Tabla 1: Componentes de software usados en IOI en el período 1999-2007

Ediciones (IOI)	Sistema Operativo	Lenguajes	Compilador	Debugger
Pekín, China (1999)	MS-DOS	Turbo Pascal v7.0 Turbo C v3.0	-	-
Pekín, China (2000)	MS-DOS	Turbo Pascal v7.0 Turbo C v3.0	-	-
Tampere, Finlandia (2001)	Windows 98, Debian	Pascal C y C++	Free Pascal gcc v2.95.2	ddd
Yong In, Corea (2002)	Windows XP, Debian	Pascal C y C++	Free Pascal v1.0.6 gcc v2.95	ddd
Kenosha-Wisconsin, Estados Unidos (2003)	Windows XP, Red Hat	Pascal C y C++	Free Pascal v1.0.10, gcc/g++ v 3.2.2	gdb v5.3, ddd v3.3.1
Atenas, Grecia (2004)	Windows XP, Red Hat	Pascal C y C++	Free Pascal v1.0.10, gcc /g++ v3.2.2 ,	gdb v5.3, ddd v3.3.1
Nowy Sacz, Polonia (2005)	Windows XP, Debian	Pascal C y C++	Free Pascal v1.0.10, gcc/ g++ v3.3	gdb v5.3, ddd v3.3.1
Mérida, México (2006)	GNU / Linux.	Pascal C y C++	Free Pascal v2,0 gcc/g + + v 3,3	gdb v5.3, ddd v3.3.1
Croacia, Zagreb. (2007)			Free Pascal v2,0 gcc/g + + v 3,3	

Como se puede observar, en las ediciones desarrolladas en Pekín (República Popular China), en los años 1999 y 2000, se utilizó como sistema operativo el MS-DOS y como lenguajes de programación

Turbo y Turbo Pascal, no se utilizaron ni compiladores, ni debugger, por estar integrados al sistema operativo.

En los años 2001 y 2002 en las ediciones de la IOI celebradas en Finlandia y Corea, respectivamente, se sustituye el MS-DOS por el Windows 98 y Debian (distribución de Linux), ajustándose al desarrollo de la tecnología. Se comienzan a utilizar como lenguajes de programación Pascal, C y C++, lo que trae aparejado la utilización de compiladores como Free Pascal, gcc y ddd como debugger.

En las competencias efectuadas en Estados Unidos (2003), Grecia (2004), Polonia (2005) México (2006), Croacia (2007) se utiliza como SO Windows XP y Red Hat, sustituyéndose este último en el año 2005 por Debian. Los lenguajes se mantienen y a los compiladores se le adiciona el g ++, en el caso de los debugger se le adiciona el gdb. En cada edición de las competencias de la IOI se utilizan versiones superiores a las anteriores, en correspondencia con el desarrollo de la informática.

Otro elemento que se incorpora a las competencias son los **STL documentation**, que se comienzan a usar oficialmente en las olimpiadas del año 2005, y se han ido incorporando consecutivamente en todas las demás como complemento para los que programan en los lenguajes C y C++.

Otro componente son los jurados online, estos sistemas se empiezan a discutir en 1999 como parte de los cambios que se requerían en las olimpiadas. Jyrki Nummenma, presidente del comité técnico, empezó a desarrollar los jurados y el cambio de sistema operativo, así en el 2001 se presentó el primer jurado online que era poco eficiente y trabajoso, pero fue algo novedoso que sustituyó el tedioso trabajo manual y el grabado en disquetes que podían llevar a la pérdida de las soluciones. A partir de ese año cada país desarrolla su propio jurado, elemento muy variable en cuanto a nombre, pero con una misma finalidad, aspecto que será tratado más adelante.

Cada año se especifican nuevos requerimientos de software para efectuar las competencias, en el año 2008 el entorno requerido es **Fedora Core 6** con **Kde** y **Gnome**. Como editores de texto el joe, vim, kate, kwrite, kdevelop, emacs, xemacs, rhide, xwpe y el lazarus. Los compiladores que se especifican son gcc >= 4.1, g++ >= 4.1, Free Pascal >= 2.0.4 con documentación y los depuradores gdb y ddd. Incluye además una documentación STL.

Como se puede observar en las competencias de la IOI se han utilizado los sistemas operativos MS-DOS, Windows y diferentes distribuciones de Linux. En la actualidad este último se ha generalizado,

por lo que se demuestra la necesidad de desarrollar habilidades en el manejo de dicho entorno por parte de los estudiantes cubanos.

1.1.2- Evolución de las competencias de programación en la ACM-ICPC. Caracterización de los componentes de software usados.

La Competición Internacional Universitaria ACM de Programación (ACM-ICPC) es una competencia de programación que se celebra anualmente, y aunque la forma organizacional dista mucho de la IOI tratada con anterioridad, en ambas prima el trabajo en equipo, el análisis de problemas y el desarrollo rápido de software.

La ACM-ICPC tiene sus raíces en una competencia celebrada en la universidad de Texas A & M en 1970 auspiciada por el Alpha de la UPE Computer Science Honor Society. Esta competición pasó a ser una competencia con varias rondas clasificatorias en 1977 y la final mundial se organizó con la colaboración de la ACM.

El concurso se transformó en una competencia de varios niveles con la primera final celebrada en el año 1977. Con sede en la Universidad de Baylor desde el decenio de 1980.

La idea de la competencia rápidamente adquirió popularidad en los Estados Unidos y Canadá, como una iniciativa innovadora para ayudar al desarrollo de los mejores alumnos en el emergente campo de la Informática, y se ha extendido en la actualidad a una red mundial de universidades que acogen las competiciones regionales previas a la Final Mundial.

El patrocinio de estas competencias ha sido asumido, históricamente, por distintas empresas, entre las que se destacan:

- La Corporación AT&T (American Telephone and Telegraph) corresponde a las iniciales del nombre antiguo de una de las compañías telefónicas más grande del mundo, el operador de televisión por cable más grande de los Estados Unidos, y hasta fue clasificado como un monopolio.
- Microsoft es una empresa privada multinacional estadounidense, fundada en 1975 dedicada al sector de las tecnologías de la información con sede en Redmond, Washington, Estados Unidos. Microsoft desarrolla, fabrica, licencia y produce software para equipos electrónicos.
- International Business Machines (IBM): Conocida coloquialmente como el Gigante Azul es una

empresa que fabrica y comercializa herramientas, programas y servicios relacionados con la informática.

IBM patrocina estos eventos desde el año 1997 siendo la empresa más importante en su desarrollo. Sus estrategias han influido positivamente pues la participación ha crecido a varias decenas de miles de los mejores estudiantes y profesores en las disciplinas de computación en 1756 universidades de 82 países en seis continentes. (1)

Estas competencias han tomado tal relevancia a nivel internacional que el número de equipos aumenta en un 10 y un 20% cada año, lo que deja claro que ACM-ICPC ha ido aumentando en número de participantes y países, siendo ahora la competencia internacional más importante que tiene lugar entre universitarios de todo el mundo.

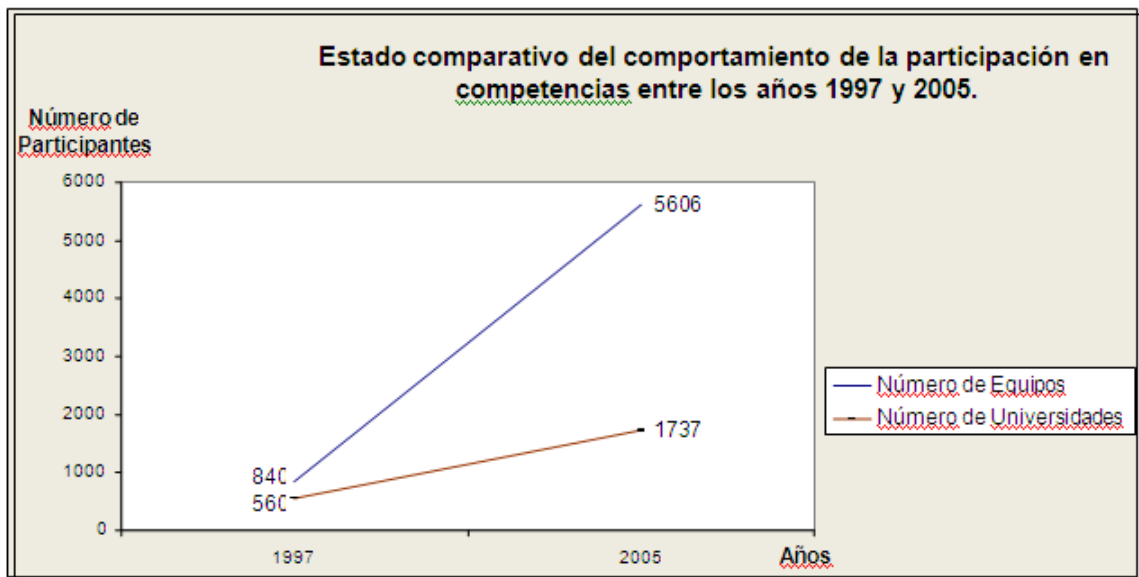


Figura 1: Estado comparativo del comportamiento de la participación en competencias entre los años 1997 y 2005

El hecho que IBM sea el patrocinador de la ACM-ICPC durante más de diez años implica que en cierta forma Microsoft ha abandonado el escenario de las competencias de programación. Por otra parte, IBM, sin dejar del todo las prácticas del **software propietario**, ha empezado a votar y a apoyar el **software libre**, lo que influye activamente en los entornos de sus competencias y en sus decisiones, las que han sido bien aceptadas por el mundo académico.

Las plataformas usadas por la ACM-ICPC sin dudas tienen en cuenta las controversias existentes entre los sistemas operativos pero a pesar de ello los intereses de su patrocinador influyen notablemente en su entorno.

En el año 2007, se utiliza como sistema operativo Fedora Core 6 Linux, incluyendo los entornos de escritorio Kde y Gnome. Los editores de texto incorporados son: vi/vim, gvim, emacs, gedit, kate. Los lenguajes empleados: Java v1.6, C/C++; cuyo compilador es gcc v4.1.2. El jurado usado es el PC2 Contest Control System v8.

En el presente año 2008 se incorporan a los requerimientos un IDE para Java, Eclipse v3.3.1 incluyendo el **plugins** CDT v4.0.1 para los lenguajes C/C++.

1.2 - Evolución de las competencias nacionales de programación.

En las competencias nacionales de programación al igual que las internacionales participan estudiantes del nivel medio-superior y superior, y son dirigidas, en el caso de Cuba, por el Ministerio de Educación (MINED) y el Ministerio de Educación Superior (MES) respectivamente. En cuanto a las competencias nacionales en la educación superior la investigación se enmarca en el área de la Universidad de las Ciencias Informáticas por ser una de las universidades que más eventos de programación realiza y cuentan con un excelente prestigio entre sus participantes cuyo número crece en cada edición.

1.2.1- Las Competencias Nacionales de Programación en la enseñanza Media Superior.

Las competencias de programación comienzan en la década de los años 80 como encuentros entre los Institutos Preuniversitarios de Ciencias Exactas (IPVCE) con la introducción de las mini computadoras en esos centros, aunque estos eventos se efectuaban ya en diferentes regiones del país en 1986, las competencias nacionales de forma organizada no tienen lugar hasta 1989 paralelamente a las olimpiadas internacionales.

Para participar en las olimpiadas internacionales se preparan estudiantes de centros de la enseñanza media o técnica. Estos, de acuerdo a los resultados que van teniendo en su preparación y los encuentros a nivel de base, municipales, provinciales y nacionales, pasan a formar parte de la Preselección Nacional y se les brinda una fuerte preparación en el Centro Nacional de Entrenamiento creado a tales efectos.

En las competencias nacionales celebradas en Cuba siempre se ha usado el sistema operativo Windows y mayoritariamente los lenguajes C/C++ y Pascal, dado que los compiladores de C y Pascal se adecuan mejor a las características de este evento por el tipo de problema que resuelven. Para ello los compiladores usados han sido Free Pascal y gcc.

No se usan servidores ni jurados online, así que los navegadores carecerían de sentido, por ello la calificación se procede de forma muy sencilla: se abre una ventana de MS-DOS y se validan los resultados, esto constituye un problema porque es totalmente diferente a lo reglado internacionalmente y puede provocar pérdidas de las soluciones.

En Cuba, en los centros de estudios se tienen computadoras de variadas arquitecturas de hardware, así como que no se tienen las herramientas para trabajar de acuerdo a lo estipulado internacionalmente, por ello sería conveniente un sistema que sea adaptable a esa diversidad.

1.2.2 - Las Competencias Nacionales en la Enseñanza Superior.

En la Universidad de las Ciencias Informáticas se efectúan competencias de programación, lógica y algoritmos, la mayoría organizadas por la Cátedra de Programación. Algunos de estos eventos son la Copa She-programer, Copa Grundy PC, Copa Troya, Copa UCI, Copa Pascal, Copa Estudiantil por Invitación y Copa Void. Estas dos últimas, son las más significativas para la investigación porque sus entornos sirven de punto de partida al desarrollo del trabajo.

1.2.2.1 - Evolución de la Copa Void de Programación (CVP).

La Copa Void es una competencia de programación, lógica y algoritmo dirigida a estudiantes de la enseñanza superior, que se efectúa en la UCI desde el 29 de marzo del año 2005. Surge como iniciativa de un grupo de estudiantes, de segundo año de la Facultad 10, inspirados en las competencias efectuadas por la ACM- ICPC.

En cada encuentro, los equipos que están compuestos por no más de tres estudiantes, deben realizar en un plazo de tiempo de cuatro horas, seis ejercicios de gran complejidad en idioma inglés, lo que contribuye de forma más amplia en la formación de los competidores y estudiantes en general.

Este evento competitivo ha sido de gran importancia para el intercambio de experiencias y de conocimientos entre estudiantes y profesores de la universidad, y ha potenciado el desarrollo intelectual de sus participantes.

Con este evento se ha logrado que en el país exista una competencia similar a las que se efectúan internacionalmente, esto favorece a que estudiantes que luego van a participar en competencias organizadas por la IOI o ACM-ICPC, lleguen más familiarizados con el entorno.

A continuación se presenta la Tabla 1.2 con todas las ediciones de este evento:

Tabla 2: Componentes de software usados en la Copa Void.

Edición	Sistema Operativo	Jurado Online	Lenguajes	Compiladores
1ra 2da	Windows XP	Jurado Online de Valladolid	C++ C# Java Pascal	g++ v4.0.0-8 java v5.0 gpc v030830 fpc v2.0.0-2
3ra	Windows XP	Jurado Online de Shepere Jurado Online de Valladolid	C++ C# Java Pascal	g++ v4.0.0-8 java v5.0 gpc v030830 fpc v2.0.0-2
4ta	Windows XP, Debian.	Jurado Online de Valladolid	C++ C# Java Pascal	g++ v4.0.0-8 java v5.0 gpc v030830 fpc v2.0.0-2
5ta	Nova	Jurado Online de Shepere	C/C++ Java Pascal Pike Scheme Common Lisp Haskell Ocaml Prolog Lua Nice Smalltalk Ada 95 Ruby Perl Bash Python	gcc v4.0.0-8 g++ v4.0.0-8 java v5.0 gpc v030830 fpc v2.0.0-2 pike v7.4.35 guile v1.6, gobi v0.9 gcl v2.6.5 clisp v2.33.2 ghc v6.2.1 ocaml v3.08.1 swipl v5.2.7 lua v5.0.2 nicec v0.9.6 gst v2.1.7 gnat v3.15 ruby v1.8.1

				perl v5.8.3 bash v2.05b-15 python v2.4
6ta, 7ma y 8va.	Nova	Jurado Online de Valladolid Jurado Online de Shepere	C/C++ Java Pascal Pike Scheme Common Lisp Haskell Ocaml Prolog Lua Nice Smalltalk Ada 95 Ruby Perl Bash Python	gcc v4.0.0-8 g++ v4.0.0-8 java v5.0 gpc v030830 fpc v2.0.0-2 pike v7.4.35 guile v1.6, gobi v0.9 gcl v2.6.5 clisp v2.33.2 ghc v6.2.1 ocaml v3.08.1 swipl v5.2.7 lua v5.0.2 nicec v0.9.6 gst v2.1.7 gnat v3.15 ruby v1.8.1 perl v5.8.3 bash v2.05b-15 python v2.4
9na, 10ma, 11na, 12ma, 13va	Nova	Programing Contest Control (PC ²) v 8.6.	C/C++ C#. Java Pascal	gcc/ g++ v4.0.0-8 gpc v030830 java v5.0 fpc v2.0.0-2
14va	Nova	Servidor UCI, PC ² v 8.6.	C/C++ C#. Java Pascal	gcc/ g++ v4.0.0-8 gpc v030830 java v5.0 fpc v2.0.0-2
15va	Nova	Servidor UCI	C/C++ C#. Java Pascal	gcc/ g++ v4.0.0-8 gpc v030830 java v5.0 fpc v2.0.0-2
16va	Nova	Servidor UCI	C/C++	g++ v4.0.0-8

			Java Pascal	fpc v2.0.0-2 gcc v4.0.0-8 java v5.0 gpc v030830
--	--	--	----------------	--

Como se observa en la Tabla 1.2, en la primera y segunda edición de la Copa Void de Programación, se utilizan los lenguajes C++, C#, Java, Pascal, así como los compiladores g++, java, gpc y fpc. El jurado online fue el motor automático de la Universidad de Valladolid, en España. El sistema operativo usado es Windows XP. La Tercera edición de este evento competitivo, en lo único que difiere de las anteriores es que como jurado de calificación se utilizó además el motor de calificación de Shepere, en Polonia. En la cuarta edición de esta copa, el único cambio significativo fue el uso del SO Debian.

Durante las primeras cuatro ediciones de este evento, se hace uso del Windows como Sistema Operativo y solamente cuatro lenguajes de programación (C++, C#, Java, Pascal.). Luego, debido a los buenos resultados obtenidos, la Universidad toma la decisión de oficializar la Copa Void como evento a ese nivel y establece que la misma debe realizarse en una plataforma libre y selecciona la distribución Nova para la competencia, además que determina la utilización de 18 lenguajes de programación, entre los más importantes están: Perl, Bash, Python, C++, C#, Java, Pascal.

Los compiladores mayormente utilizados en la competencia son: perl, bash, python, gcc, g++, java, gpc y fpc. Los jurados de calificación que se mantienen hasta la octava edición son el motor automático de la Universidad de Valladolid, en España y el motor de calificación de Shepere, en Polonia.

En la novena edición de la Copa, se comienza a utilizar como jurado el Programming Contest Control ó PC^{^2}, y nuevamente se reduce la cifra de lenguajes de programación a C, C++, Java, Pascal, C# y los compiladores a gcc, g++, java, gpc y fpc. Esta situación se mantiene en las siguientes convocatorias, donde el único cambio que se realiza es en el jurado online, que se comienza a utilizar el Servidor UCI, aplicación realizada en la Universidad de las Ciencias Informáticas y que ha tenido gran aceptación por parte de los competidores.

Este evento ha reportado numerosas ventajas a la Universidad de las Ciencias Informáticas y ha obtenido gran prestigio dentro y fuera de la misma, los estudiantes están más preparados y motivados a participar en eventos competitivos

1.2.2.2 - Evolución de la Competencia Estudiantil por Invitación de Programación.

La Competencia Estudiantil por Invitación de Programación es un evento que tiene por sede oficial la Universidad de las Ciencias Informáticas. El objetivo principal es propiciar un lugar para el intercambio de experiencia y conocimientos entre programadores de la enseñanza media y media-superior.

Es organizada por la Copa Void de Programación (CVP), la Cátedra de Programación Avanzada (CPAV) y algunas organizaciones estudiantiles como la FEU y UJC de la UCI.

La primera edición de este evento se efectúa el día 6 de junio y constituye el primer evento competitivo que se realiza en conjunto con otros centros educacionales del país, algunos son: La Universidad de la Habana (UH), el Instituto Superior Politécnico José Antonio Echeverría (ISPJAE), y equipos de la Preselección Nacional de Computación y Matemática (PSN). En la segunda edición participan además La Universidad Central “Marta Abreu” de las Villas (UCLV) y la Universidad de Matanzas “Camilo Cienfuegos” (UMCC). En la tabla que aparece a continuación se muestran las herramientas utilizadas en este evento.

Tabla 3: Componentes de software usados en la Copa Estudiantil por Invitación

Herramientas	1ra Convocatoria	2da Convocatoria
Sistema Operativo	Nova v0.9	Nova
Escritorios	Gnome v2.16.2	Gnome v2.16.2. Window Maker v0.92.0-r3.
Navegadores	Firefox v2.0.0.4	Firefox v2.0.0.4
Editores de Texto	gedit v2.16.2-r1. geany vi/vim v7.0.235.	vi/vim v7.0.235 gvim v7.0.235 emacs v21.4-r12 gedit v2.16.2-r1. mcedit v4.6.1-r3 joe v3.5 rhide v1.5.20050823
Lenguajes de Programación	C/C++ C# Java	C/C++ Java Pascal

	Pascal	
Compiladores	gcc v3.4.6 g++ v3.4.6 javac v1.4.2 fpc v2.0.2 mcs v1.1.13.6	gcc v4.1.1-r3 g++ v4.1.1-r3 javac v1.5.0.11 fpc v2.0.4
Entornos de Desarrollo Integrado (IDEs)	Eclipse v3.2.1-r2 CDT v3.1 fpc-ide v2.0.4	Eclipse v3.2.1-r2 CDT v3.1 fpc-ide v2.0.4
Depuradores	gdb v6.6-r2 ddd v3.3.11	gdb v6.6-r2 ddd v3.3.11
Sistema de Control	PC ² v8.6	PC ² v8.7

En las dos ediciones de este evento se utilizan las mismas herramientas, la única diferencia radica en las versiones. El SO utilizado es Nova, y el gestor de escritorio Gnome, además de WindowMaker en la segunda edición. El navegador fue Firefox. En ambas competencias se utilizan editores como gedit, geany, vi/vim, adicionándose para la segunda edición el gvim, emacs, mcedit, joe y el rhide. Los lenguajes de programación permitidos son C, C++, Java, Pascal y C#, este último solo es utilizado en la primera edición. Los IDEs empleados son Eclipse para Java, y el plugins CDT para C y C++. El sistema de control usado fue el PC ² y los debugger gdb y ddd.

1.3 - Nova Base para las competencias de programación.

La idea de crear una distribución netamente cubana se inicia por parte de dos estudiantes de la Universidad de las Ciencias Informáticas: Anielkis Herrera y Yoandy Martínez alrededor del año 2004. Inicialmente surge por la necesidad de tener una plataforma de desarrollo para las aplicaciones que se estaban realizando en la UCI en el marco del Software Libre. Es así como nace el proyecto Nova.

La distribución base para la creación de Nova es Gentoo porque permite compilar todo su software para obtener un sistema exactamente ajustado a la medida que se desea.

En sus inicios Nova era solo un proyecto productivo en la Facultad de Software Libre (FSWL) de la UCI que se dedicaba a la realización de una imagen que fuera estable, luego, debido al auge que cobra el Movimiento de Software Libre en todo el país y por la necesidad de migrar a un sistema operativo libre y cubano, se tomó la decisión de utilizar Nova para el proceso de migración.

De Nova se han realizado numerosas versiones en dependencia de las necesidades que se han ido presentando en la Universidad. Uno de los grandes logros de la distribución ha sido la creación del Live CD, que no es más que un sistema operativo que se ejecuta desde un disco sin necesidad de acceder al disco duro de la PC. Esto ofrece numerosas ventajas, pues permite efectuar reparaciones al sistema y salvadas de la información.

No menos importante es la creación del Live USB, herramienta similar al Live CD, pero que difiere de esta en que se ejecuta desde una memoria flash, además, el usuario tiene permisos de lectura y escritura, que le facilitan que pueda almacenar y configurar la información.

Se han creado numerosas versiones de imágenes como son la destinada a la docencia y los entornos de oficina. La primera ha sido utilizada por los estudiantes de la FSWL y está compuesta por un grupo de herramientas que son indispensables para el plan de asignaturas de esta. La segunda, es una imagen creada para usuarios que no se encuentran dentro del proceso de desarrollo del software, y está compuesta por un grupo de herramientas de oficina que solo requieren de conocimientos básicos para su utilización.

¿Por qué Nova?

Cuba es un país independiente que se ha trazado como meta lograr la soberanía tecnológica, para ello ha invertido capital y material humano desde inicios de la Revolución.

¿Por qué usar un sistema privativo que esclaviza y restringe, si existe la posibilidad de crear una distribución propia?

La elección de Linux como propuesta de sistema operativo para usar en las competencias de programación en la educación cubana, está relacionado con la necesidad que existe en Cuba de migrar al software libre, con la finalidad de alcanzar la independencia tecnológica, por lo que no se puede seguir pensando en la piratería como alternativa, es hora de realizar, de acuerdo a las necesidades, herramientas y sistemas operativos propios.

Es importante señalar que desde el año 2001 en las olimpiadas de informática a nivel mundial se ha usado Linux, no siendo así con Windows, que fue prohibido en la edición del 2006 en México, por las ventajas que reporta el primero, como ya se ha referenciado.

La Universidad de las Ciencias Informáticas cuenta con una distribución cubana que tiene un grupo de desarrollo que ofrece soporte de software y tiene en cuenta las necesidades de los usuarios para la construcción de sus aplicaciones.

La distribución Nova es un entorno ajustable a los programadores cubanos, manteniéndose actualizada respecto a sus herramientas.

1.4 - Los programas y herramientas usados en competencias que son compatibles con la distribución Nova.

El objetivo fundamental de este acápite es hacer una presentación de las herramientas que son compatibles con la distribución Nova, teniendo en cuenta las que se han utilizado en los eventos mencionados en epígrafes anteriores, así como otras que por su importancia, se deben incorporar en la personalización.

1.4.1– Navegadores.

Un navegador o explorador Web (en Inglés, navigator o browser), es una aplicación que permite visualizar documentos de hipertexto. En el mundo se utilizan infinidad de navegadores, de acuerdo a las necesidades y gusto del usuario. En las competencias de programación se utilizan para conectarse al sistema de control o jurado online.

La distribución Nova, soporta navegadores como Mozilla Seamonkey y Mozilla Firefox, Netscape Navigator, Epiphany y Opera, este último no es libre por lo que no se tiene en cuenta en esta investigación.

Mozilla es uno de los navegadores más populares y seguros, tanto así que muchos usuarios de Windows han optado por hacerlo su navegador predeterminado, además, en un estudio realizado recientemente por **Secunia**, es uno de los navegadores que más prestaciones y soluciones presenta (2), en capítulos posteriores se profundizará en la versión a utilizar en el presente trabajo.

1.4.2 – Compiladores.

Tanto los compiladores, como los debugger y los IDEs, están en dependencia de los lenguajes de programación que se utilicen durante la competencia. Estos son seleccionados por los organizadores del evento.

Un “compilador es un programa que recibe un programa fuente y devuelve un programa objeto” (3). Puede ser además una “aplicación informática que se usa para crear programas en un cierto lenguaje de programación. Convierte los programas creados en un lenguaje de programación al lenguaje interno del ordenador (código máquina). (4)

Durante las competencias de programación analizadas en epígrafes anteriores, se han utilizado numerosos compiladores, de ellos los más frecuentes son: gcc y g++ para C/C++, fpc y gpc para Pascal, mono para c#, java para Java, otros como perl, bash, python dirigidos a los lenguajes de igual nombre, son compatibles con Nova, solo que los bases de este trabajo son los compiladores de c, c++, Pascal y java, por ser los lenguajes de programación establecidos internacionalmente.

1.4.3 – Depuradores.

La necesidad de incorporar un depurador o debugger al sistema está dada por la razón de que muchos de los IDEs libres no los traen incluidos.

Un depurador es el: “... programa diseñado para ayudar a depurar otro programa al permitir al programador observar su ejecución instrucción por instrucción.” (5)

El depurador más usado en competencias nacionales e internacionales de programación es el gdb.

1.4.4- Entornos de Desarrollo Integrado (IDEs).

“Un entorno de desarrollo integrado (IDE) consiste básicamente en un software que previamente ha sido instalado en la máquina del cliente y cuyo principal objetivo es el desarrollo de otro software.” (6)

Un IDE está compuesto por un editor de texto, un compilador o intérprete, un depurador y herramientas de automatización; ofrece la posibilidad de un sistema de control de versiones y ayudan en la construcción de interfaces gráficas de usuarios.

Dentro de los IDEs más utilizados en competencias y que son compatibles con la distribución, se encuentran: Eclipse y Lazarus.

1.4.5- Editores de Texto.

Un editor de texto es un “Programa utilizado para crear y manipular archivos de textos (...) Estos programas se diferencian de los procesadores de textos porque no disponen de características elaboradas de formato e impresión (tales como alineado automático, negritas, fuentes tipográficas, cuerpos” (7)

Existen numerosos tipos de editores de texto, dentro de los compatibles con la distribución Nova se encuentran: joe, emacs, xemacs, gedit y vim/gvim, los cuales han sido utilizados en competencias mencionadas anteriormente y pueden ser incorporados a la personalización.

1.4.6 – Diccionario

En las competencias de programación, los diccionarios son herramientas que están orientadas a la traducción de palabras o frases de lenguas extranjeras, aunque también tratan los significados, usos de vocablos, y la definición de las diversas acepciones.

Se incluye esta herramienta debido a las facilidades que ofrece a los estudiantes en la traducción de problemas durante las competencias, favoreciendo el desarrollo del idioma Inglés. El diccionario que se incorpora a la personalización es el Stardict, al que se le debe instalar una librería especificándole el idioma que se va a utilizar.

1.4.7 – Sistemas de Control

En este acápite se analizan tres sistemas de control: PC2, Servidor UCI y Mooshak,. El primero por ser el estipulado por la ACM-ICPC, el segundo por ser desarrollado en la UCI y el tercero por ser uno de los más reconocidos internacionalmente.

1.4.7.1- PC^2

Es un sistema en tiempo real dinámico, diseñado para manejar y controlar competencias de programación. Incluye soporte para competencias simultáneas en múltiples sitios, operaciones en

plataformas homogéneas, incluyendo Windows y Linux mezclados en una sola competencia, y actualizaciones dinámicas en tiempo real de estados de las competencias.

Como se muestra en la siguiente imagen(Figura2), esta aplicación tiene que estar instalada tanto en las máquinas clientes como en el servidor por ser una aplicación de escritorio, además de ser una aplicación realizada en Java que puede consumir gran cantidad de recursos.

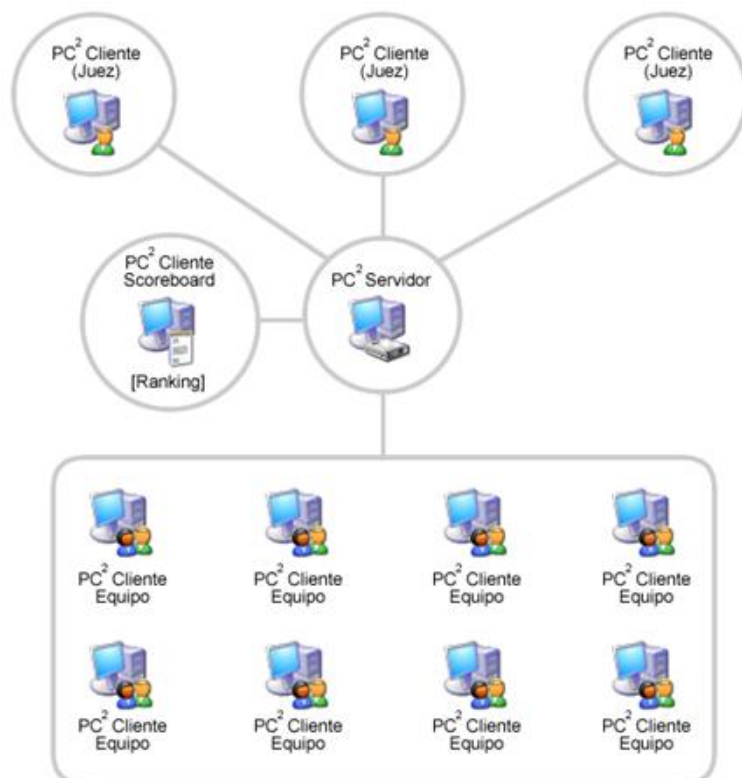


Figura 2: Distribución del sistema de control PC²

Por otra parte se tiene el planteamiento publicado en el sitio oficial de sistema:

" El uso de PC² para cualquier otro propósito que no sea en apoyo directo de la ACM ICPC, de la Dirección Regional de Concursos, concursos y locales de la intención de seleccionar los equipos para avanzar a la ACM ICPC Regional de Concursos, requiere el permiso expreso de la Asociación para la Maquinaria de Computación." (8)

Este sistema ha sido utilizado fundamentalmente por la ACM-ICPC y el uso de él, está prohibido para otras competencias, por lo que no es aplicable en Cuba.

1.4.7.2- Servidor UCI

EL servidor UCI, es un sistema desarrollado por Iván Olamendi, destacado profesor de la UCI, con un estilo semejante al de los jurados internacionales, destinado a ser utilizado en competencias, ya sean internas de un centro de estudios o entre varios de ellos, a través de la red nacional. Es una aplicación que se puede configurar y adaptar de acuerdo a las características de la competencia o institución en que va a ser utilizada, y está dirigida a ampliar la variedad y la naturaleza de los problemas soportados por la aplicación.

En las competencias se proponen problemas a solucionar, y se especifican los lenguajes que pueden utilizar. Esta aplicación permite soportar diferentes lenguajes (hasta el momento Perl, C++, Python, C, Pascal, Java, C#, PHP), siendo muy sencillo adicionarlos al sistema.

El Servidor UCI ha tenido gran aceptación, pero hace falta un alto requerimiento de hardware del cual no se dispone por la variedad de arquitecturas existentes en los centros de estudios del país, además que no cuenta con la madurez de otros sistemas reconocidos internacionalmente y de fácil acceso.

1.4.7.3-Mooshak

Mooshak es un sistema para la gestión de competencias de programación en la web que contiene las funciones básicas. Estas incluyen: juzgar automáticamente los programas presentados, responder preguntas sobre la aclaración de las descripciones del problema, la reevaluación de los programas, seguimiento de las impresiones, entre otras. Debido a que permite crear tanto competencias al estilo de la IOI como la ACM-ICPC, además de hacer preguntas y respuestas para las clases de preparación, se estudia con profundidad para ser el sistema de control en el primer release de Nova para las competencias de programación en Cuba.

1.5 - Metodología para la gestión y construcción de la personalización de las imágenes.

Una metodología es el conjunto de pasos o procedimientos a seguir para llevar a cabo un proyecto de cualquier índole o disciplina, tiene determinado objetivo, espera cierto resultado y brinda solución a un problema.

1.5.1- Metodologías Ágiles

En la actualidad se hace más necesario implantar metodologías de desarrollo de software que garanticen simplicidad y velocidad para crear sistemas operativos. Entregar un producto de calidad en tiempo y con el menor costo posible, es una de las principales metas que se proponen los equipos de desarrollo.

Las metodologías ágiles forman parte del movimiento de desarrollo ágil de software, se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito de un proyecto. Se le denomina ágil por la habilidad de responder de forma versátil al cambio para maximizar los beneficios, de forma que una metodología ágil es la que tiene como principios:

- Los individuos y sus interacciones son más importantes que los procesos y las herramientas.
- El software que funciona es más importante que la documentación exhaustiva.
- La colaboración con el cliente en lugar de la negociación de contratos.
- La respuesta delante del cambio en lugar de seguir un plan cerrado.

Sus elementos claves son:

- Poca documentación
- Simplicidad
- Análisis como una actividad constante
- Diseño evolutivo
- Integraciones
- Testeos diarios

En la siguiente tabla se puede observar una comparación entre una metodología tradicional y una ágil.

Tabla 1.4- Comparación entre Metodologías Ágiles y Metodologías Tradicionales.

Criterio	Metodologías Ágiles	Metodologías Tradicionales
Roles	Poseen pocos roles.	Poseen gran cantidad de roles.
Artefactos	Poseen pocos Artefactos.	Poseen gran cantidad de artefactos.
Arquitectura	Se define y mejora a lo	Se define al principio del

	largo del proyecto.	proyecto.
Aplicación	Deben aplicarse a proyectos pequeños de corta duración, con no más de 10 integrantes y que residan en el mismo sitio.	Puede aplicarse a proyectos de cualquier tamaño, pero son más eficientes en proyectos grandes y equipos dispersos.
Contrato	No tiene que existir	Sí existe un contrato prefijado.
Interacción con el Cliente	Constante, es parte del equipo de desarrolladores.	Sólo en reuniones oficiales.

Los modelos de gestión tradicionales no son efectivos para afrontar un reto que resulta fundamental, incorporar cambios con rapidez y en cualquier fase del proyecto. Se trata de evitar lo que tantas veces ha ocurrido: cuando el proyecto se encuentra en una etapa avanzada se descubre que no se va por el buen camino o, simplemente, el cliente decide introducir cambios sustanciales, y esos cambios obligan a tirar por la borda todo el trabajo realizado hasta entonces, e impide acabar en el plazo previsto. Dado que los cambios nunca van a dejar de existir, lo que se necesita es gestionar los proyectos de una forma más ágil.

Además de ello, las metodologías tradicionales no responden en buena medida al desarrollo de sistemas operativos, se requiere de una metodología que sea adaptable a iniciativas propias de desarrolladores, lo que es posible a través del trabajo con metodologías ágiles.

Dentro de la Metodologías ágiles más conocidas y aplicables se encuentran:

- XP – Extreme Programming
- Crystal Clear
- DSDM – Dynamic Systems Development Method
- Scrum

1.5.1.1- Extreme Programming

Extreme Programming conocida por XP es una metodología de desarrollo de software eficiente, de bajo riesgo y flexible. Se centra fundamentalmente en potenciar las relaciones interpersonales, promoviendo el trabajo en equipo y el aprendizaje de los desarrolladores. Es adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico.

Algunas de sus principales características son:

- Desarrollo iterativo e incremental.
- Pruebas unitarias continuas, frecuentemente repetidas y automáticas.
- Programación en parejas.
- Interacción frecuente del equipo de programación con el cliente o usuario.
- Corrección de todos los errores antes de añadir una nueva funcionalidad.
- Hacer entregas frecuentes.
- Refactorización del código.
- Propiedad del código compartido: promueve que todo el personal pueda corregir y extender cualquier parte del proyecto.
- Simplicidad en el código

Esta metodología está dirigida especialmente a equipos de desarrollo compuestos fundamentalmente por programadores, lo que no está incluido entre las tareas de la investigación, por lo que no es aplicable en el presente trabajo.

1.5.1.2- Crystal

Es una familia de metodologías que se centran fundamentalmente en las personas que componen el equipo y en producir la menor cantidad posible de artefactos. Sus principales puntos de estudio son el aspecto humano y el tamaño del equipo, así como la comunicación entre sus componentes.

Las Metodologías de “Crystal” más conocidas son la Crystal Clear y Crystal Orange. La primera está orientada de 3-8 personas, mientras que la segunda es de 25-50. Cada uno de estos métodos tiene asignado un color en dependencia del tamaño y la complejidad del proyecto.

Esta metodología está orientada fundamentalmente a equipos de desarrollo compuestos por más de tres integrantes por lo que no es factible en la presente investigación.

1.5.1.3- Dynamic Systems Development Method (DSDM).

Define el marco para desarrollar un proceso de producción de software. Reconoce que los proyectos son limitados por el tiempo, los recursos y los planes acorde a las necesidades de la empresa.

Algunas de sus principales características son:

- Es un proceso iterativo e incremental y
- El equipo de desarrollo y el usuario trabajan juntos.

El método empieza con un estudio de viabilidad y negocio. El estudio de viabilidad considera si DSDM es apropiado para el proyecto. El estudio de negocio es una serie de cortes para entender el área de negocio dónde tiene lugar el desarrollo. También propone arquitecturas de esbozos del sistema y un plan del proyecto.

El ciclo de vida contempla cinco fases:

- Estudio de viabilidad.
- Estudio del negocio.
- Modelado funcional.
- Diseño.
- Construcción e implementación.

Las tres últimas son iterativas, además de existir retroalimentación a todas las fases.

DSDM tiene principios subyacentes que incluyen una interacción activa del usuario, entregas frecuentes, equipos autorizados, pruebas a lo largo del ciclo. Igual que otros métodos ágiles, usan ciclos de plazos cortos de entre dos y seis semanas. Hay un énfasis en la alta calidad y adaptabilidad hacia requisitos cambiantes. (9)

Una metodología ágil por sí sola carece de sentido, por lo que es recomendable utilizar otra que la complemente. DSDM está pensada para usarse con XP y aunque pudiese adaptarse a otra, acarrearía una complejidad que dificultaría el trabajo.

1.5.1.4- Adaptive Software Development (ASD)

Se basa en la adaptación continua a circunstancias cambiantes. Dentro de sus principales características se encuentra la iteratividad, orientada a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje.

En la fase de Especulación se inicia el proyecto y se planifican las características del software, en la segunda se desarrollan las características, en la tercera fase se revisa su calidad y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

El trabajo de la metodología se enfoca directamente en fomentar las partes difíciles del desarrollo adaptable, fomentar la colaboración y el aprendizaje dentro del proyecto, ve la planificación como una paradoja en un ambiente adaptable, ya que los resultados son naturalmente imprevisibles. En la planificación tradicional, las desviaciones del plan son errores que deben corregirse. En un ambiente adaptable, en cambio, las desviaciones nos guían hacia la solución correcta. En este ambiente imprevisible se necesita que las personas colaboren de la mejor manera para tratar con la incertidumbre. (9)

Esta metodología es poco adaptable a la construcción de las imágenes pues está dirigida a la organización del equipo de desarrollo y no al proceso en sí.

1.5.1.5- Lean Development (LD) y (LSD) Lean Software Development

En LD los cambios se consideran riesgos, pero sí se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad. Su principal característica es introducir un mecanismo para implementar dichos cambios. Es la metodología menos divulgada.

Dado que LD es más una filosofía de administración que un proceso de desarrollo donde no es relevante el tamaño del equipo, la duración de las iteraciones, los roles o la naturaleza de sus etapas.

Últimamente LD ha evolucionado como Lean Software Development (LSD); LD se inspira en doce valores centrados en estrategias de gestión:

- Satisfacer al cliente es la máxima prioridad.
- Proporcionar siempre el mejor valor por la inversión.
- El éxito depende de la activa participación del cliente.
- Cada proyecto LD es un esfuerzo de equipo.
- Todo se puede cambiar.
- Soluciones de dominio, no puntos.
- Completar, no construir.
- Una solución al 80% hoy, en vez de una al 100% mañana.
- El minimalismo es esencial.
- La necesidad determina la tecnología.
- El crecimiento del producto es el incremento de sus prestaciones, no de su tamaño.
- Nunca empujes LD más allá de sus límites. (9)

LD y LSD son utilizadas generalmente como complemento de las metodologías XP y Scrum, no como una metodología a implementar en la personalización.

1.1.5.6- Scrum

Define un marco para la gestión de proyectos, cuyo objetivo es elevar al máximo la productividad de un equipo. Está especialmente indicado para proyectos con un rápido cambio de requisitos. No propone el uso de ninguna práctica de desarrollo en particular; sin embargo, es habitual emplearlo, y puede ser combinado con cualquiera de las metodologías mencionadas anteriormente.

Scrum por sí solo resuelve las necesidades de gestión porque:

- Libera creatividad e incrementa la productividad, tema imprescindible para incrementar las diversas personalizaciones que puedan surgir a partir de la primera versión.
- No es un proyecto por roles lo que lo hace más adaptable a un trabajo de dos autores.
- La flexibilidad para los requerimientos, indispensable ya que durante el desarrollo se descubren incompatibilidades de algunas herramientas con la distribución.
- El trabajo es efectuado por equipos auto-organizados y auto-dirigidos.

De forma general Scrum está especialmente indicado para proyectos con un rápido cambio de requisitos. El desarrollo de software se realiza mediante iteraciones, denominadas Sprints, generalmente con una duración de 30 días.

El resultado de cada sprint es un ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas se destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

Algunos de los conceptos que se manejan en Scrum:

- Product Backlog: Lista de las funcionalidades del producto ordenadas de mayor a menor importancia.
- Sprint Backlog: Tareas que se descomponen del Product Backlog.
- Sprint: Período que culmina con un ejecutable o sistemas con las funcionalidades del Sprint Backlog.

Esta metodología consta de 3 Fases:

1- Pregame¹

- Plannig

Se define el sistema a construir a través del Product Backlog, en la que se expresan los principales requerimientos y a partir de estos se estima el esfuerzo requerido.

2- Development

Es donde se desarrolla el Sprint y se intenta conseguir el objetivo dándole cumplimiento a los requerimientos planteados.

3- Postgame

Se han cumplido todas las funcionalidades del Sprint, se realizan pruebas y se revisan tareas pendientes o necesidades de mejoras. En caso de no cumplir los requerimientos se comienza una nueva iteración, de lo contrario se procede con la fase siguiente que propone la ingeniería de liberaciones.

Scrum se utiliza para la planificación y organización en el presente trabajo por su sencillez, además que las fases que plantea son ajustables al desarrollo de la personalización que se pretende crear. Utilizar Scrum por si solo carece de sentido, por ello se propone utilizarla con la Ingeniería de Liberaciones.

1.5.2- Ingeniería de Liberaciones

La ingeniería de liberaciones es similar a otras metodologías de desarrollo. Es un conjunto de buenas prácticas y su estructura está definida por cada equipo de desarrollo, de acuerdo a sus características y particularidades. Es la alternativa adoptada por el proyecto Nova de la Facultad 10 de la Universidad de las Ciencias Informáticas, para la realización de distintas personalizaciones.

La figura representa una serie de tareas o misiones para obtener una imagen, partiendo desde el la compilación del kernel y la construcción de la sistema base hasta la obtención del producto que se va a

¹ Los términos de la metodología Scrum se conservan en su idioma original.

liberar. Es necesario aclarar que dicha imagen ha sido modificada y adaptada a las necesidades de la personalización.



Figura 3: Pasos de la Ingeniería de para el desarrollo de la personalización

En este capítulo se ha hecho un estudio de los entornos usados en las competencias internacionales y nacionales, en lo concerniente a la IOI y a la ACM-ICPC. Su desarrollo y evolución demuestran la necesidad de migrar hacia sistemas libres, pues como ha sido expuesto en los últimos años Windows no es el entorno usado en las competencias de programación y queda atrás en este mundo académico, a diferencia de Linux que cada día desarrolla más distribuciones que brindan grandes prestaciones para estos eventos.

Por otra parte, el estudio de las competencias nacionales demuestra la necesidad realizar cambios para cumplir con las normas internacionales. Esto es posible a través de la implementación de este trabajo, que toma como base experiencias similares llevadas a cabo en la Universidad de las Ciencias Informáticas, demostrando que es posible estar en la vanguardia tecnológica para las competencias de programación.

Además del estudio de las herramientas, para desarrollar la personalización es necesario conocer cómo gestionar y construir imágenes, por ello se han seleccionado metodologías ágiles que faciliten el trabajo en el próximo capítulo.

CAPÍTULO 2

PERSONALIZACIÓN DE LA IMAGEN CLIENTE Y SERVIDOR

En este capítulo se desarrollan las imágenes a partir de la metodología Scrum, adoptando una estructura de acuerdo a sus fases; unido a dicha metodología se usa la ingeniería de liberaciones para construir la personalización, de manera que Scrum complementa cada uno de sus pasos.

2.1- Construcción de la Imagen Base y Compilación del Kernel.

El primer paso para conformar la personalización, es crear una imagen base a partir de un stage1 de Gentoo que contiene solo las herramientas necesarias para compilar el sistema: algunos comandos, librerías, compiladores, así como la jerarquía de archivos del sistema. Luego, a este se le incluyen el portage y el árbol del portage, conceptos que se tratan más adelante.

Luego se procede con la construcción del núcleo o kernel, que es el corazón del sistema, pues establece el enlace entre el hardware y el software. Para la compilación de este se utiliza la herramienta Genkernel, que lo detecta y configura automáticamente. La versión utilizada en la personalización es la 2.6.22-nova-r9.

Es importante mencionar que la construcción de la imagen base y el kernel fueron realizados por un grupo de desarrolladores especializados del proyecto Nova de la Facultad 10 de la Universidad de la Ciencias Informáticas.

2.2- Primer Sprint. Construcción del Sistema y Compilación de Binarios.

En este Sprint se comienza la construcción de las imágenes cliente y servidor a partir de la imagen base y el kernel. Luego se procede a compilar los binarios de las herramientas que se incluyen en la personalización.

2.2.1- Plannig del Primer Sprint.

Durante la fase de planeamiento del Primer Sprint se definen los requerimientos o funcionalidades de la personalización a través de las pilas del producto o product backlog.

2.2.1.1- Product Backlog.

Se definen dos listas de funcionalidades, la primera para la imagen cliente y la segunda para la imagen del servidor. Las columnas representan la forma de organizarse cada funcionalidad: Imp. corresponde a la importancia de cada una donde cinco es el mayor valor que puede tomar, Est. es el número de personas que se estima puedan dar cumplimiento a la tarea y T es el tiempo reflejado en días para culminar cada funcionalidad.

Tabla 4: Product Backlog del primer Sprint de la Imagen Cliente

Funcionalidad	Imp	Est	Cómo Probarlo	Notas	T (días)
Programar en C/C++, Java y Pascal.	5	2	Ir al menú, seleccionar el IDE o editor de texto del lenguaje en el que se desee programar.	Garantizar que los compiladores y los Debugger funcionen correctamente	5
Conectarse al Sistema de Control.	5	2	Ir al menú y ejecutar el navegador.	El navegador debe estar bien configurado.	1
La máquina tome un IP dinámicamente	5	2	Comprobar las conexiones de red.	El Protocolo dhcp funcione correctamente.	1
Conectarse a máquinas con Windows.	4	2	Alt.+F2, ejecutar conexión por samba a otra máquina.	Debe estar instalado y en el default el demonio samba	1
Montar Dispositivos de E/S automáticamente.	3	2	Conectar un dispositivo (CD, Flash-USB, disco floppy)	Comprobar que el System-volume-manager y el pmount estén instalados correctamente.	1

Tabla 5: Product Backlog del Primer Sprint de la Imagen del Servidor

Funcionalidad	Imp.	Est	Cómo Probarlo	Notas	T(días)
Especificar el puerto de espera de las peticiones.	5	2	En el archivo de apache.conf se especifica el puerto.	El estándar por defecto para HTTP es el 80	3
El jurado online funcione correctamente calificando problemas en los lenguajes C/C++, java y pascal.	5	2	Enviar un problema resuelto desde el cliente.	Que tenga los compiladores requeridos.	3
Permitir que se monten las particiones automáticamente.	4	1	Observar si muestra los íconos.	Configurar fstab	2
Garantizar un sistema con configuraciones básicas y ligero.	4	1	-	Instalar el sistema base solo con las librerías que requiere el Mooshak.	1
Cargar un sistema operativo seleccionado por el usuario.	4	1	Reiniciar la máquina	Configurar el grub.conf	1
Mostrar a quien dirigirse en caso de error 404 en el navegador del cliente.	3	1	-	Garantizar que esté configurado en el archivo de apache.conf.	1

2.2.2- Development.

Durante esta fase se explican los pasos a seguir durante el proceso de construcción de las imágenes cliente y servidor, así como los binarios de las herramientas que son compilados, de manera que queden garantizadas las funcionalidades determinadas en el primer sprint.

2.2.2.1- Development. Imagen Cliente y Servidor.

En primer lugar se necesita una partición en el disco duro, que no es más que una división que se le hace a este con un formato determinado, en este caso se utiliza el ext3 como sistema de archivos. El tamaño aproximado que debe tener es de 9GB. Para el particionamiento se utiliza la herramienta Gparted, que es el editor de particiones de Gnome y puede ser utilizado para crear, destruir, redimensionar, inspeccionar y copiar particiones o sistemas de archivos. Luego se toma la imagen base elaborada con anterioridad y se descomprime como **root** en la partición creada.

Es importante aclarar que la personalización es realizada desde un sistema Nova previamente instalado y totalmente funcional debido a que se utilizan muchas de sus configuraciones.

Configuración del resolv.conf

En el caso que se necesite utilizar la red para la instalación y configuración del sistema, se procede a la configuración del archivo `/etc/resolv.conf`; de lo contrario no es necesario.

En este archivo se procede a la configuración de los parámetros nameserver, para indicar las direcciones IP de los servidores DNS que deben ser utilizados.

Montar Sistemas de Archivos `/dev/` y `/proc/`

Es necesario montar los sistemas de archivos `/proc` y `/dev`. El primero es un sistema virtual de archivos que permite utilizar la información que proporciona el kernel, el segundo, es un directorio que contiene numerosos archivos que representan dispositivos de hardware. (Anexo III)

Configuración del make.conf

En este archivo, que se encuentra en el directorio `/etc/make.conf`, se hacen entre otras, las configuraciones de las variables que usarán el portage y el Overlay, para hacer todas las instalaciones y montar los sistemas de archivos en los directorios pertinentes. A continuación se explican algunas de

las variables configuradas.

```
CFLAGS="-O3 -march=i686 -pipe -mmmx -fno-gcse -fthread-jumps -fexpensive-optimizations -frerun-loop-opt"
```

```
CXXFlags = "${CFLAGS}"
```

CFLAGS y CXXFLAGS son variables de entorno usadas para decirle a la **colección de compiladores**, gcc, qué tipo de parámetros usar cuando compila código fuente. Las CFLAGS son para código escrito en C, mientras que CXXFLAGS son para código escrito en C++.

La primera y más importante opción es -march. Esta le dice al compilador qué código debía producirse para la arquitectura de su procesador, o sea, manda al compilador producir código específico para su CPU, tomando en cuenta todas sus capacidades, características y conjuntos de instrucciones. Es utilizada también para que los programas sean optimizados de acuerdo al tipo de procesador.

Otra de las opciones es -pipe. Esta hace que el proceso de compilación sea más rápido e indica al compilador que use tuberías en lugar de archivos temporales durante los diferentes estados de compilación.

El parámetro -mmmx activa los conjuntos de instrucciones MMX para arquitecturas x86-64. Es útil principalmente en multimedia, juegos y otras tareas intensivas de computación en punto flotante, aunque también contiene otros realces matemáticos. Esta instrucción se encuentra fundamentalmente en las más modernas CPUs.

```
MAKEOPTS= "-j2"
```

La variable MAKEOPTS representa el número de hilos y define cuántas compilaciones puede hacer simultáneamente la máquina, en este caso se tiene un solo procesador, por lo que se le da ese valor.

```
PORTDIR_OVERLAY="/usr/local/nova-overlay"
```

En esta variable, PORTDIR_OVERLAY, donde se define la ubicación del árbol de Portage, se agrega la dirección mostrada, si la carpeta nova-overlay no se encuentra en /usr/local debe ser creada.

En el caso de la imagen cliente se deben configurar los parámetros siguientes:

```
INPUT_DEVICES="keyboard mouse"
```

```
VIDEO_CARDS="nvidia"
```

La variable `FEATURES` define las acciones que toma el sistema Portage de forma predeterminada y modifica su comportamiento, por ejemplo con `ccache` se crea una cache de compilación, que acelera este proceso.

```
FEATURES="-ccache -distcc buildpkg"
```

Las `USE` son palabras claves que incorporan información de soporte y dependencias para un concepto en concreto. Si se define un determinado parámetro `USE`, Portage sabrá que el usuario desea soporte para la palabra clave escogida.

En el caso de la imagen cliente se configura de la siguiente manera:

```
USE="-qt3 -qt4 -arts -kde gtk gnome hal"
```

```
USE="$USE ppds lm_sensors pmunforce2"
```

Esta variable contiene opciones que controlan el comportamiento de la construcción de varios paquetes por ello se agregan o se deshabilitan diferentes `USE` del sistema que se necesitan.

El Portage.

Portage es un sistema de gestión de paquetes en forma de árbol que ofrece gran flexibilidad para instalar y mantener el software de un sistema.

Se procede a copiar el portage en una partición o carpeta para comenzar la instalación de paquetes. Este se monta en el directorio `/usr/portage` con `-bind`. Otra opción es con el comando `-sync`, pero lo que hace es sobrescribirlo, por lo que esa vía no es factible.

El portage se ha configurado a través del archivo `/etc/make.conf`

Construcción del Overlay

El siguiente paso es la construcción del Overlay, que es un sistema de mantenimiento e instalación de programas que trabaja paralelamente al portage. La diferencia con este radica en que el Overlay da la facilidad de instalar un programa que no tiene soporte en el portage, permitiendo a los usuarios copiar

el archivo que se desea instalar en el portage.

La configuración del Overlay se ha realizado en el `/etc/make.conf`.

Es importante para el Overlay tener instalado y configurado un sistema de control de versiones, en este caso se utiliza el subversion (svn) que ofrece las siguientes ventajas:

1. Soporte transaccional, las actualizaciones al repositorio se ejecutan o se cancelan como un todo.
2. No se necesitan comandos especiales para el manejo de directorios.
3. Soporte para versiones en archivos binarios.
4. Permite asignar **metadatos** y propiedades a los archivos bajo control de versiones. Estos metadatos son incluidos también dentro de dicho control
5. La integración con el servidor Apache permite que los repositorios puedan quedar accesibles por los protocolos http y https. (10)

Luego de tener el svn funcionando correctamente se procede a montar la partición de nova-Overlay del sistema. (Anexo III)

Montar /usr/src

A continuación se procede a montar la partición `/usr/src` que es donde se encuentran los fuentes del kernel, esto permite que a la hora de compilar paquetes, estos encuentren el código fuente necesario para efectuar la compilación.

Actualizar las variables del sistema.

Como se está construyendo la imagen desde un sistema ya existente, se procede a loguearse como **chroot** para trabajar directamente con el sistema en creación.

Luego se procede a actualizar las variables del sistema debido a que se está utilizando el chroot, para ello, el primer paso debe ser crear las variables de entorno y luego se cargan estas en memoria (Anexo III).

2.2.2.1.1- Development. Imagen Cliente.

Instalación de Xorg.

Xorg es la que proporciona una interfaz entre el hardware y el software gráfico. Para ello se configuran dos variables del archivo `/etc/make.conf`: `video_cards` e `input_devices`. La primera se usa para establecer los controladores de vídeo, su configuración normalmente se basa en el tipo y marca de la tarjeta que posea, en este caso se utiliza de tipo `nvidia`. La segunda es para determinar qué controladores han de ser creados para los dispositivos de entrada, en este caso el ratón y el teclado (`keyboard & mouse`).

A continuación de la configuración de estas dos variables se procede con la instalación de Xorg. (Anexo III)

Instalación de Gnome-light

Gnome es uno de los entornos de escritorio más utilizados en el mundo entero y una plataforma de desarrollo. Para una correcta configuración, es necesario revisar que estén agregados como parámetros a la variable `USE` del `/etc/make.conf` lo siguiente:

```
USE="-qt3 -qt4 -arts -kde gtk gnome hal"
```

Se le quitan los parámetros que le den soporte a `kde` puesto que en este caso no se van a utilizar: `-qt3 -qt4 -arts -kde`

Se le habilitan los parámetros por Gnome y **gtk**, así como el demonio `hal` que se explica más adelante.

Luego se procede a instalar Gnome-light que es un **entorno de escritorio** mínimo, o sea, solamente posee las aplicaciones necesarias para su funcionamiento.

Instalación de Demonios: samba, hald, y dhcp.

Un demonio es un programa que permanece en segundo plano ejecutándose continuamente para proporcionar diferentes funciones y ciertos servicios, pero sin la interacción del usuario.

Samba es un demonio que facilita que los participantes durante la competencia, copien o se conecten

por la red a máquinas que utilizan el sistema operativo Windows.

Hal (Hardware Abstraction Layer) es el demonio de abstracción del hardware, que funciona como una interfaz entre el software y el hardware del sistema, creando una plataforma consistente sobre la que es posible correr las aplicaciones, en otras palabras, el demonio hal permite que las aplicaciones de escritorio tengan acceso a información de hardware.

DHCP (Dynamic Host Configuration Protocol) es un protocolo que permite a los nodos de una red IP, obtener sus parámetros de configuración automáticamente. Este protocolo evita tener que configurar manualmente en cada computadora la dirección IP cada vez que se traslada a otra parte de la red, además le permite al administrador supervisar y distribuir de forma centralizada las direcciones IP necesarias, y automáticamente asignar y enviar un nuevo IP a la PC.

Instalar Gdm.

Gnome Display Manager (GDM), es el Administrador del entorno de escritorio Gnome, que gestiona la configuración de inicio de sesión para los usuarios en los sistemas.

Luego de proceder con la instalación de gdm se accede al archivo `/etc/conf.d/xdm` y se realiza el siguiente cambio:

```
Display Manager = "gdm"
```

El gestor de pantalla X (XDM) es una parte opcional del sistema X Window que se usa para el manejo de sesiones. El Sistema X Window o más comúnmente conocido por X11 o X, es el estándar de los entornos gráficos para Unix.

Adicionar al nivel de ejecución default xdm y dbus. Instalación del Gnome- volumen- manager.

Un nivel de ejecución o runlevel es un estado en el que el sistema está corriendo y contiene scripts que serán ejecutados al ingresar o salir del nivel de ejecución. Permite además que los scripts de los demonios se inicien o se detengan automáticamente. En Nova se puede definir qué demonios se inician y se detienen al entrar en cada runlevel.

Existen varios niveles de ejecución (cuantos uno desee), y cada uno de ellos proporciona

funcionalidades diferentes y un entorno de trabajo distinto. Default es un nivel de ejecución por defecto del sistema, donde se encuentran aquellos demonios cuyo arranque no compromete la estabilidad del mismo.

El objetivo de adicionar al nivel de ejecución default el xdm, es para que el gestor gráfico de Gnome inicie automáticamente al arrancar el sistema.

Luego se adiciona al nivel de ejecución default el dbus que es una vía de comunicaciones del sistema que Gnome usa extensamente.

Una vez concluido este paso, se procede a instalar Gnome-volume-manager que es un administrador de volúmenes para Gnome y es usado para montar dispositivos E/S automáticamente (entiéndase por dispositivos E/S por unidades de disquetes, CDRoms y memorias flash USB).

Luego a través de la instalación de pmount se puede habilitar el permiso a los usuarios para montar dispositivos de E/S.

Instalación de paquetes necesarios.

La instalación de cada uno de los paquetes que se presentan a continuación no es obligatoria, simplemente constituye un grupo de herramientas que debe tener la imagen por las ventajas que proporciona a los usuarios.

Los paquetes a instalar se representan en la siguiente tabla:

Tabla 6: Paquetes necesarios instalados en la imagen cliente

Paquete	Utilidad
Gamin	Es un servicio prestado por una librería que permite detectar cuando un archivo o directorio ha sido modificado
Gnome-system-tools	Herramientas de administración del entorno gráfico
Gnome-system-monitor	Monitor de procesos del sistema.
File-roller	Manejador de archivos de Gnome.
Evince	Visor de documentos pdf.

etcat: extractor de información del sistema portage

gensync: sincronizar y actualizar el árbol portage paralelo

revdep-rebuild: busca librerías o ejecutables rotos a causa de alguna actualización de algún paquete.

eclean es una herramienta para la limpieza de los fuentes y los binarios de Gentoo. Permite que los directorios de fuentes y binarios no crezcan infinitamente, manteniendo los archivos que aún puedan hacer falta.

equery es una utilidad muy flexible que permite obtener información sobre los paquetes, tal como los archivos que contiene, las USE con que se compiló, el md5 de cada uno de los archivos que pertenecen al paquete especificado.

Instalación de Abiword

Abiword es el procesador de texto oficial del entorno gráfico Gnome. Lo caracteriza la sencillez de su interfaz y los bajos requerimientos técnicos. Es además libre y **multiplataforma**.

Instalación de Network-Manager

Network Manager es una aplicación que permite al usuario configurar la red automáticamente. Es un demonio que se debe adicionar al nivel de ejecución default y él realiza las configuraciones de red cableada o inalámbrica.

2.2.2.1.2- Development. Imagen Servidor.

La característica más importante de la imagen servidor es que debe ser una imagen ligera, por ello solo se le instala el sistema de control y las dependencias de este.

Instalación de paquetes necesarios.

El desarrollo de la imagen servidor se inicia con el estudio del script `./install`, donde se determinan los directorios que necesita el Mooshak para colocar sus funcionalidades. En la siguiente tabla se muestran los directorios y los paquetes correspondientes a su instalación.

Tabla 7: Relación entre los paquetes y los directorios

Directorio	Paquete
/usr/bin/lp	net-print/cups
/usr/bin/time	sys-process/time
/usr/bin/crontab	sys-process/vixie-cron
/usr/bin/host	net-dns/bind-tools
/usr/bin/xsltproc	libsxsl

2.2.2.2- Compilación de Binarios

La compilación de los binarios es la instalación de las herramientas que debe tener la personalización para dar cumplimiento a las listas de funcionalidades expuestas con anterioridad.

2.2.2.2.1- Compilación de Binarios. Imagen Cliente.

A continuación se muestra una tabla con cada una de las herramientas que fueron seleccionadas en el capítulo anterior, así como las versiones instaladas en la imagen cliente. La elección de estas últimas fue realizada en base a los requerimientos de la IOI y la ACM-ICPC.

Tabla 8: Versiones de cada una de las herramientas instaladas en la imagen cliente

Tipo	Nombre	Versión
Navegador	Mozilla- Firefox	2.0.0.14
Compiladores	fpc	2.2.0
	gcc	4.1.2
	ddd	3.3.11
Editores de Texto	gedit	2.20.4-r1
	gim	7.1.266
	gvim	7.1.266
	joe	3.5
	xemacs	21.4.21-r1
	emacs	22

Diccionarios	Stardict	3.0.1
IDEs	Lazarus	0.9.24
	Eclipse	3.3.1.1
Debugger	gdb	6.7.1-r

2.2.2.2.2- Compilación de Binarios. Imagen Servidor.

Como se ha explicado con anterioridad en la imagen del servidor, su centro y razón de ser, es el sistema de control Mooshak, y a partir de las dependencias de instalación del mismo se determinan los paquetes que necesitan ser compilados. A continuación se muestra una tabla con las versiones instaladas en la imagen. La elección de estas últimas fue realizada en base a los requerimientos de la IOI y la ACM-ICPC. En el caso del sistema de control es la última versión disponible en su sitio oficial.

Es importante aclarar que el intérprete es necesario para que funcione el Mooshak, debido a que la instalación del este es un script que es interpretado por el tcl.

Tabla 9: Versiones de cada una de las herramientas instaladas en la imagen servidor

Tipo	Nombre	Versión
Sistema Control	Mooshak	1.4,1
Intérprete	Tcl	8.4.15
Servidor Web	Apache	2.2.8
Compiladores	Fpc	2.2.0
	Gcc	4.1.2
	Sun-Jdk	1.6.0.03
	Mono	1.2.5.1- r1

Es importante aclarar que el intérprete es necesario para que funcione el Mooshak, debido a que la instalación del este es un script que es interpretado por el tcl.

2.2.2.2- Algunas configuraciones.

Configuración del fstab

Una de las características de Linux es que permite utilizar varios tipos de sistemas de ficheros según su rapidez de acceso a datos o según la capacidad de recuperación. Para evitar la incomodidad de tener que montar dichos archivos manualmente cada vez que sea necesario su uso, es que se debe configurar el archivo fstab.

Este fichero se encuentra en el directorio /etc y posee las configuraciones de todos los sistemas de ficheros del sistema operativo, el lugar donde se encuentran y en que orden deben ser montados. Es de gran importancia su configuración porque permite montar automáticamente la partición al iniciar el sistema.

```
GNU nano 2.0.7                               File: /etc/fstab
# /etc/fstab: static file system information.
#
# noatime turns off atimes for increased performance (atimes normally aren't
# needed; notail increases performance of ReiserFS (at the expense of storage
# efficiency). It's safe to drop the noatime options if you want and to
# switch between notail / tail freely.
#
# The root filesystem should have a pass number of either 0 or 1.
# All other filesystems should have a pass number of 0 or greater than 1.
#
# See the manpage fstab(5) for more information.
#
# <fs>                <mountpoint>    <type>          <opts>          <dump/pass>
# NOTE: If your BOOT partition is ReiserFS, add the notail option to opts.
#/dev/BOOT            /boot            ext2             noauto,noatime  1 2
/dev/sda7              /                ext3             noatime         0 1
#/dev/SWAP            none             swap             sw              0 0
#/dev/cdrom            /mnt/cdrom       auto             noauto,ro       0 0
#/dev/fd0              /mnt/floppy      auto             noauto          0 0
#
# glibc 2.2 and above expects tmpfs to be mounted at /dev/shm for
# POSIX shared memory (shm open, shm unlink).
# (tmpfs is a dynamically expandable/shrinkable ramdisk, and will
# use almost no memory if not populated with files)
shm                   /dev/shm         tmpfs            nodev,nosuid,noexec 0 0
```

Figura 4: Imagen de la configuración del archivo fstab

Configuración del Grub.

GRUB es el Gran Gestor de Arranque Unificado que permite que el usuario a través de un menú pueda seleccionar un sistema operativo durante el proceso de arranque y pasar argumentos al kernel según una iniciación específica del mismo.

El archivo de configuración es `/boot/grub/grub.conf`, el que es utilizado para crear la lista de los sistemas operativos que pueden ser arrancados, así como otras configuraciones que el usuario necesite.

2.2.3- Postgame

Una vez concluida la fase de construcción de este sprint, se le realizan pruebas de aceptación a ambas imágenes por parte de los integrantes del proyecto Nova con conocimientos de las herramientas instaladas para demostrar el cumplimiento de las funcionalidades.

Durante estas pruebas se detectaron algunos errores en la imagen cliente con los compiladores gcc y cc, a la hora de programar en el lenguaje C++ por lo que se determina utilizar el compilador g++ en sustitución de los anteriores.

De forma general la imagen cliente tuvo buena aceptación por parte de los programadores que en ella realizaron las pruebas.

2.3- Segundo Sprint. Configuración de la Imagen.

Una vez concluida la etapa de construcción y compilación del primer Sprint y verificando que no quedan funcionalidades de las pilas del producto por garantizar, se procede inmediatamente a configurar la imagen con el objetivo de hacerla más agradable y cómoda para los usuarios. Este constituye otro paso de la ingeniería de liberaciones.

2.3.1- Plannig del Segundo Sprint.

Durante la fase de planeamiento del Segundo Sprint, se definen los requerimientos o funcionalidades de la personalización a través de las pilas del producto o product backlog, en esta iteración las funcionalidades están dirigidas a la configuración del producto.

2.3.1.1- Product Backlog.

Se define una lista de funcionalidades para dar cumplimiento a los parámetros de configuración que se necesitan en la personalización, que proporcionen una interfaz agradable y cómoda.

Tabla 10: Product Backlog del segundo Sprint de la Imagen Cliente

Funcionalidad	Imp.	Est	Cómo Probarlo	Notas	T(días)
Garantizar que el navegador se conecte al sistema de control.	5	2	Abrir el Firefox y que se conecte por si solo al sistema.	Configurar el apache y el navegador.	2
Garantizar que el usuario entre a la sesión competidor por defecto.	4	1	Reiniciar la PC y esperar unos minutos.	Configurar las opciones de accesibilidad.	1
Garantizar que los competidores trabajen en una sesión de usuarios y no en la de administrador.	4	1	El usuario accede a la sesión con el usuario competidor y la clave competidor.	La sesión competidor debe estar creada.	1

2.3.2- Development.

Durante esta fase se procede con la etapa de configuración de las imágenes cliente y servidor correspondiente a la personalización.

2.3.2.1 - Configuración de la Imagen.

El proceso de configuración de las imágenes está orientado fundamentalmente a los usuarios que trabajen con las imágenes, para que estos se sientan más cómodos, además de completar algunas configuraciones que permitan un correcto funcionamiento de los paquetes instalados.

2.3.2.1.1- Configuración de la Imagen Cliente.

Resolución de Pantalla

La resolución de pantalla es la cantidad de píxeles (o máxima resolución de imagen) que puede ser mostrada en la pantalla. Esta depende fundamentalmente de la calidad del monitor y de la tarjeta gráfica de la PC. Cuanto más alta es la resolución de pantalla mejor es la calidad de la imagen. En el caso de esta imagen se escogió una resolución de 1024x768 píxel.

Crear la sesión para competencias.

Se crea una sesión llamada competidor para el trabajo durante la competencia donde se habilitan los permisos de lectura de CD-ROM y disquetes, así como montar dispositivos de E/S automáticamente. Se niega el acceso a los privilegios de administrador (root) y se deshabilitan los íconos de configuración del sistema, de fondo de escritorio, tema y resolución de pantalla.

2.3.2.1.2- Configuración de la Imagen del Servidor

Mooshak.

Después de desempaquetar el contenido del archivo descargado del sitio oficial del Mooshak en Internet, se direcciona al descomprimido y para comenzar la instalación se ejecuta el script "install".

```
# tar xvzf mooshak.tgz
# cd Mooshak
# ./install
```

El script de instalación crea un usuario llamado 'Mooshak' con un directorio Public_html' con todos los datos. Aunque el usuario ya se ha creado, el sistema aún no lo reconoce como usuario, para ello se procede con el comando:

```
# user add -g users -s /sbin/no login mooshak
```

Varios Sub-directorios se crean automáticamente después de la instalación:

datos: Datos de todos las competencias

public_html: CGIs y html (páginas de ayuda)

Tcl: Paquetes utilizados por CGIs

plantillas: Plantillas HTML para CGIs

Binarios: Programas requeridos por Mooshak

contrib: otros archivos aportados por diferentes personas

Configurar Apache

La configuración de apache puede no admitir el directorio / Cgi-bin / que trae incorporado el Mooshak. Para evitar ello el fichero de configuración del apache debe traer las siguientes líneas:

```
<Directory /home/*/public_html/cgi-bin>
```

```
Options +ExecCGI -Includes -Indexes
```

```
SetHandler cgi-script
```

```
</Directory>
```

Si el archivo ya tiene incluidas estas líneas, se procede con la configuración de las opciones del apache, disponibles en el directorio /etc/conf.d/apache. Las opciones a incluir son USERDIR y SUEXEC.

Ya se ha mencionado que el instalador de Mooshak crea un usuario en el directorio: /home/Mooshak donde se guardan todos los datos del sistema de control, en caso que se quiera acceder a él sería <http://dirección.ip/~mooshak>. Para que apache reconozca este sitio en el home, se especifica la opción

UserDir que es la que indica el directorio que se añade como directorio personal de los usuarios para servir páginas cuando se realiza una petición del tipo ~usuario, en este caso ~mooshak.

La característica suexec provee a los usuarios de Apache la posibilidad de correr programas con interfaz común de acceso, **CGI** (Common Gateway Interface) y directivas interpretadas por el servidor, **SSI** (Server Side Includes) bajo ID's de usuario diferentes del ID de usuario del **servidor Web**, es uno de los elementos del sistema de control elegido. Respecto a los ID's más adelante se explica como crear estos usuarios.

2.3.3- Postgame

Durante la XVII Edición de la Copa Void efectuada por el comité organizador de la Facultad 10 de la Universidad de las Ciencias Informáticas el día 23 de febrero del año 2008 se presenta la imagen cliente creada y se aplica una encuesta a los participantes (ver Anexo II) con el objetivo de obtener información referente al estado de conformidad de los programadores al trabajar con la imagen cliente, y detectar dificultades en su funcionamiento. Se toma para ello una muestra de 48 personas.

El 67% de los competidores afirman que mientras trabajan con la imagen se sienten perdidos o incómodos, esto es debido a que el 71% de los participantes nunca antes habían trabajado con Linux. Solo el 29% tenía cierta experiencia en sistemas Linux, afirmó que estaba fácil el entorno.

El 19% de los encuestados presentaron problemas con la velocidad del eclipse, y dentro de ellos un 6% asegura que el sistema está muy lento.

Durante la competencia, 36 de los participantes utilizaron IDEs para realizar los ejercicios, mientras que los 15 restantes utilizaron editores de texto. La razón es que el 71% no conocía los editores de texto.

De todos los encuestados, el 58% aseguró que la imagen tenía una buena calidad y el 15% que estaba muy buena. Mientras que el 21% aseguraba que estaba regular y un 6% que estaba muy mala.

2.4- Tercer Sprint. Optimización de las imágenes.

La ingeniería de liberaciones propone hacer una serie de pruebas antes de liberar el producto, por tanto se parte de los resultados del sprint anterior, dado que la imagen está desarrollada en su

totalidad, se encamina a resolver las pruebas que se realizan hasta este momento que son de aceptación. Se ha decidido optimizar las imágenes, luego de las pruebas realizadas con anterioridad.

2.4.1- Plannig del Tercer Sprint.

En esta fase se elabora la lista de la pila del producto que responde a las necesidades recogidas de la encuesta realizada en la edición pasada de la Copa Void.

2.4.1.1- Product Backlog.

Tabla 11: Product Backlog del Tercer Sprint de la Imagen Cliente

Funcionalidad	Imp	Est	Cómo Probarlo	Notas	T(días)
Proporcionar un entorno agradable aplicando la propuesta de los diseñadores.	5	1	-	Modificar la estructura de los menús por lo que se necesita de la elaboración de nuevos iconos.	5
Optimización de la imagen cliente	5	1	-	Optimización del Eclipse	2
Actualizar algunas herramientas instaladas.	5	2	-	Añadir algunos editores de texto.	5
Proporcionar manuales de apoyo.	5	1	-	STL Documentation y Comandos Principales de Nova.	1

2.4.2- Development.

Con el objetivo de proporcionar una imagen más sencilla de usar para las personas que nunca se han enfrentado a una distribución de libre, se decide realizar una serie de configuraciones.

Para obtener una nueva estructura en el menú se instala la herramienta *alacarte* que es una aplicación para modificar menús de gnomo, esta es utilizada para adaptarlos a las necesidades y permite además ocultar, mostrar y renombrar los menús de Gnome.

Como se explicaba en el sprint anterior, en las pruebas realizadas en la Copa Void, se presentaron problemas con la velocidad del Eclipse, por lo que se decide optimizarlo. Para ello se configura el archivo `/usr/lib/eclipse-3.2/eclipse.ini` donde se le configuraron los parámetros de Xms con 40m y Xmx con 256. Estos parámetros son la memoria RAM mínima y máxima a utilizar.

Como resultado de la optimización se decide además configurar el archivo `/etc/inittab`, donde se deshabilitan las consolas en modo texto, dejando solo la `tty1`² y la `tty7`³. Esto favorece al ahorro de memoria RAM de la máquina.

Otro cambio que se le realiza a la imagen es la sustitución del **initramfs** del kernel de Gentoo por el de Nova, pues este último ha resuelto numerosos problemas que se han presentado, principalmente en el arranque del kernel en algunas arquitecturas de máquinas.

Para darle solución al problema de algunos competidores que se sintieron incómodos al trabajar con la imagen por no tener conocimientos de Linux o de los editores de texto e IDEs, se decide agregar a la imagen cliente, documentación de los principales comandos de Linux, así como un manual de usuario que explica cómo trabajar con la imagen.

2.4.3- Postgame

Para comprobar en qué medida la personalización y el entorno desarrollado para competencias de programación cumple con lo establecido internacionalmente, se realizan encuestas a estudiantes de la preselección nacional cubana, que han participado en diferentes ediciones de la IOI, así como a estudiantes de la Universidad de las Ciencias Informáticas que han participado en las ediciones de la Copa Void, para ello se realiza una encuesta tomando una muestra de 48 personas. (Anexo II).

El 98% de los encuestados consideran la imagen buena, el 2% restante la califica de regular. Durante la competencia se sintieron cómodos el 85% y el resto afirma que pudieron sentirse mejor. En cuanto a la velocidad del sistema no se presentaron dificultades, lo mismo sucedió con los competidores que realizaron la competencia con los IDE y editores de texto.

² Consola Virtual 1.

³ Consola correspondiente al entorno gráfico.

Como resultado de la imagen servidor se obtuvo que el 90% que conocía sobre los jurados online, están satisfechos con la imagen y no enfrentaron ningún problema, opinando además que Mooshak por su facilidad y sencilla interfaz favorece el aprendizaje del trabajo con el sistema de control. El administrador de la competencia opina que es el mejor jurado con el que se ha calificado hasta el momento.

Como se puede observar las imágenes cliente y servidor no presentaron problemas en su funcionamiento, permitiendo un buen desarrollo de la competencia. Por lo tanto se considera que no es necesaria la realización de un nuevo sprint por tener cumplidas todas las funcionalidades planteadas con anterioridad, y el producto queda listo para liberar.

Se ha personalizado la distribución Nova, dando cumplimiento a todas las funcionalidades determinadas en las pilas del producto de cada sprint, utilizando la Ingeniería de Liberaciones para desarrollar cada uno de estos. Una vez cumplidas las expectativas de los concursantes se procede a describir el candidato de liberación.

CAPÍTULO 3

CANDIDATO DE LIBERACIÓN

Se ha desarrollado una personalización que contiene las herramientas actuales que se han usado en las competencias internacionales de computación o informática, lo que proporciona a los estudiantes cubanos la posibilidad de familiarizarse y prepararse en un entorno, que, aunque en el mundo ha tomado gran avance, para ellos es totalmente novedoso y diferente. En este capítulo se exponen algunas de las características y se realiza la presentación del producto.

3.1- Requerimientos de Hardware. Instalación.

Los requerimientos de hardware se presentan en dos escalas: el mínimo y el recomendado, que no son más que las condiciones que debe tener el CPU para que la personalización corra correctamente.

Tabla 12: Representa los requerimientos mínimos y recomendados para la imagen cliente

Componente	Mínimo	Recomendado
RAM	128 MB	256 MB o superior
Procesador	Pentium 2	Pentium 4 o superior
Disco Duro	5 GB	10 GB

Tabla 13: Representa los requerimientos mínimos y recomendados para la imagen servidor

Componente	Mínimo	Recomendado
RAM	32 MB	Superior
Procesador	Pentium 2	Superior
Disco Duro	2 GB	Superior

Los requerimientos mínimos son 128 MB de RAM, y el recomendado 256 MB o superior. La misma está preparada para un procesador pentium 2 o superior, y el recomendado es un pentium 4 o superior.

En el caso de la imagen del servidor los requerimientos mínimos son de 32MB de RAM.

En la Universidad de las Ciencias Informáticas y en el Centro de Entrenamiento de la Preselección Nacional, se encuentran los tipos de arquitectura especificados.

3.2- Release de Nova para competencias de programación.

El release o producto liberado para las competencias de programación consta de una imagen cliente y un servidor.

La imagen cliente incluye las herramientas con las que el competidor interactúa. La imagen servidor se clasifica como web donde corre el jurado Mooshak con todas las prestaciones que se necesitan tanto en competencias de tipo ACM-ICPC como del estilo IOI.

La imagen cliente y la servidor actúan de forma concurrente en una misma competencia, y se instalan en computadoras diferentes, se enlazan a través de una conexión establecida del cliente hasta el servidor mediante el uso del navegador, como se observa en la figura 5.

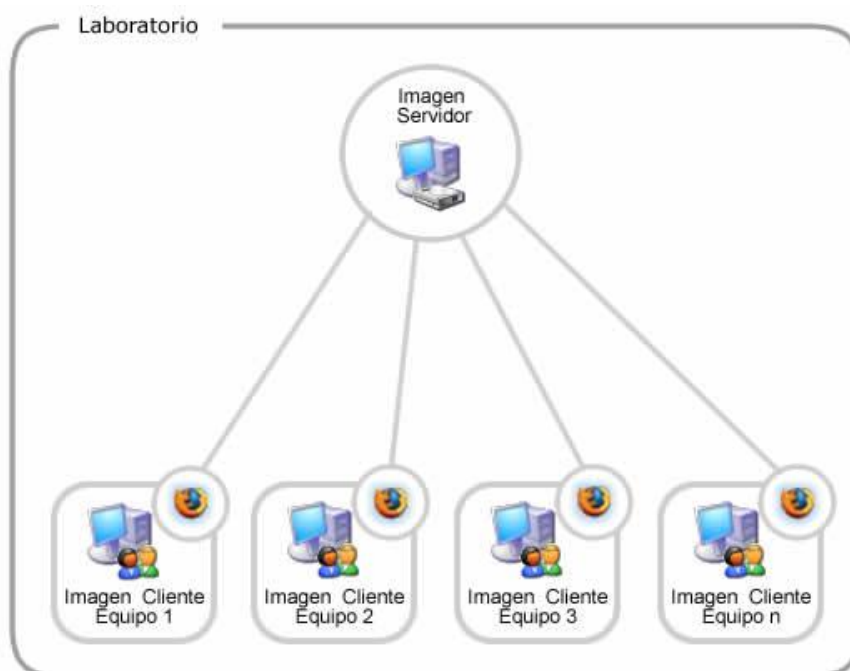


Figura 5: Jerarquía y configuración de un laboratorio con la personalización de Nova para competencias de programación

3.3- Características de la Imagen Cliente.

La imagen cliente tiene una sesión llamada “programación” a la que se accede con la contraseña “programación”. Una vez dentro de la imagen se pueden observar todas sus herramientas a través de tres menús:

- Aplicaciones.
- Lugares.
- Sistema.

Cada uno de esos menús contienen sus respectivos submenús, donde accediendo a ellos se pueden encontrar las herramientas instaladas en la misma.

Para el competidor, el menú Aplicaciones, es el de mayor importancia, pues el mismo contiene una serie de submenús para el trabajo de los programadores durante la competencia.

Dentro del menú Aplicaciones se encuentran:

- El menú Internet posee el navegador a través del cual los competidores se conectan al servidor. El diccionario puede ser utilizado como ayuda para la traducción inglés-español.
- El menú Programación, donde se encuentran los IDEs y editores de texto donde el competidor programa.
- Algunos editores de texto requieren de la consola para compilar, tal es el caso del Joe que es un editor de textos para terminales alfanuméricos. Para acceder al terminal o consola, se puede hacer a través del menú Accesorio, donde existen otras herramientas que servirán de apoyo al competidor como el Gestor de Archivos con el que se comprimen archivos, la calculadora científica y el mapa de caracteres que permite insertar símbolos.
- Otro de los menús es Documentación, donde se encuentran manuales que servirán de ayuda al programador en lenguajes como C y C++, además de algunos de los comandos principales para el trabajo con el Sistema Operativo Linux.

El menú Lugares, como su nombre lo indica, conlleva a los directorios más importantes del sistema: Carpeta personal, Escritorio, Equipo, Particiones de Red, así como el buscador de archivos y documentos recientes, y finalmente en el menú Sistema en el que se encuentran las opciones: ayuda acerca de Gnome, salir de la sesión y Apagar.

Además de los menús ya mencionados, la imagen posee herramientas en el panel superior para montar y desmontar dispositivos de entrada/salida como disquetes de 31/2, CD-ROM o Flash-USB. En este panel se puede acceder a la documentación.

3.4 – Características del Sistema de Control de la Imagen Servidor.

La forma de acceder a la imagen servidor es a través del navegador de un cliente, pues trabajando físicamente en la computadora donde se encuentra instalado, solo se puede acceder a los archivos en forma de consola y preparar la competencia es prácticamente imposible.

Cuando se inicia el sitio Mooshak, lo primero que aparece es una descripción general de este. Cuando el usuario selecciona la opción “Login”, aparece una nueva ventana donde decide que tipo de competencia desea realizar, así cómo en cuál interfaz entrar.

Las interfaces fundamentales son:

Administrador: es el que gestiona todo el sistema antes de la competencia.

Juez: gestiona todo el sistema durante la competencia y vela por su funcionamiento.

En la interfaz Administrador se crea la competencia con todos sus elementos: Grupos, Equipos, Usuarios, así como las propiedades de cada uno. Estos se organizan como directorios de manera que: competencia contiene a grupos, este a equipos y este a usuarios, con sus respectivos atributos.

Una vez creada la competencia, se determinan los lenguajes con los que se competirá; es aquí donde se especifica la extensión con la que el usuario debe subir la solución del problema, para que el sistema Mooshak sea capaz de reconocer automáticamente el tipo de lenguaje durante la competencia.

La creación de los problemas es otra de las opciones de esta interfaz. La misma se realiza a través de directorios donde se carga la descripción de estos en formato .doc y .pdf, y posteriormente estos se

convierten a página web. Conjuntamente con la descripción se definen los test del mismo, los que están compuestos por la solución del problema y los ficheros de entrada y salida.

En el momento de crear la competencia, es posible especificar algunos parámetros como el momento a partir del cual los competidores tendrán acceso al sitio y a los ejercicios.

Otra funcionalidad que ofrece el Mooshak es hacer competencias virtuales, o sea, estas se mantienen publicadas durante los días que se le especifiquen, el competidor puede acceder a ella y participar con todos los privilegios de un evento real. Estas políticas se le definen también en el momento de crear la competencia.

Otro parámetro que se determina es el modo de autenticación, el que puede ser por los usuarios creados o por autenticación LDAP. Para mantener ocultos los resultados del evento durante los últimos 60 min de competencias, el administrador del sitio, debe activar la opción Hide List.

La interfaz del Juez contiene las áreas de Submision y Question. La primera, indica el recibimiento de las soluciones del juez, lo que determina si el problema está finalizado o no, o si es necesario pasar por todos los test del problema antes de aceptarlo. La segunda opción permite a los competidores realizar preguntas al juez durante la competencia, para ello debe ser activada con anterioridad.

Los competidores pueden, en todo momento de la competencia, enviar las soluciones de sus ejercicios, es por eso que el juez debe especificar el número máximo de problemas pendientes.

En este capítulo se han mostrado los requerimientos de hardware para proceder con la instalación de la personalización creada. Se exponen además, las principales características que poseen las imágenes cliente y servidor que dan cumplimiento a los objetivos trazados en la presente investigación.

CONCLUSIONES

Se ofrecen los requerimientos y tendencias principales que caracterizan a los entornos internacionales de programación utilizados por los dos eventos de mayor nivel y prestigio académico mundial.

La personalización presentada cumple con esos requisitos, constituyendo así una contribución efectiva a una mejor preparación y desempeño de los estudiantes cubanos en las competencias de programación nacionales, regionales e internacionales.

Se ofrece también la opción de un sistema con los requerimientos del entorno de trabajo que deben cumplir las computadoras de los estudiantes que se preparan para los Competencias de Informática, especialmente las regionales e internacionales, en los que Linux es el Sistema Operativo Oficial.

Adicionalmente, se brinda una documentación muy necesaria para desarrollar personalizaciones de Nova, la que no existía antes del desarrollo de este proyecto.

Nova, distribución cubana desarrollada por estudiantes de la Universidad de las Ciencias Informáticas, extiende su personalización hasta la vanguardia y futuro de las Ciencias de la Computación, como modesta y comprometida contribución a la independencia tecnológica.

RECOMENDACIONES

Se recomienda:

Divulgar y distribuir esta personalización a estudiantes y profesores de la UCI y a sus correspondientes autoridades académicas, políticas y de masas para la generalización de su uso como entorno soporte de las competencias y concursos de programación en la universidad.

Realizar también acciones similares con los centros de educación de interés del MES y el MINED, así como con los JCCE, recomendándoles la generalización de su uso.

Incorporar en la formación de los organizadores de los eventos de Programación el estudio del presente trabajo e instalación de sistemas libres, especialmente Nova, utilizando como soporte su manual de usuario.

Continuar el desarrollo y aumento de las prestaciones del correspondiente servidor UCI para ser añadido en la personalización.

Asignar a un equipo de soporte técnico-profesional de la distribución NOVA, la misión de apoyar a los eventos de programación que se desarrollen en Cuba y actualizarlo continuamente.

Crear una herramienta que facilite la instalación de ambas imágenes, cliente y servidor, evitando el engorroso trabajo de compilación, actualización y posterior instalación de estas.

REFERENCIAS BIBLIOGRÁFICAS

1. **Domino, Lotus.** IBM. *IBM-Academic Initiative*. [En línea] <http://www-304.ibm.com/jct09002c/university/scholars/academicinitiative/>.
2. **Secunia.** Secunia_Stay Secure. *Secunia Highlights*. [En línea] Noviembre de 2007. <http://secunia.com/>.
3. **Mora, Fausto Loja.** *¿Qué es un compilador?* Abril de 2007.
4. **Gabanes, Nacho.** Arrakis. *Diccionario Informático*. [En línea] 2005. <http://usuarios.lycos.es/Resve/diccioninform.htm#Autor>.
5. Lycos. *Todo de la Informática*. [En línea] 2007. <http://usuarios.lycos.es/tutoinformatica/DEPURADOR.html>.
6. **Qualitrain Express.** Blog de Qualitrain. *Metodologías Ágiles de desarrollo de software*. [En línea] 2001. <http://www.qualitrain.com.mx/index.php/Procesos-de-software/Metodologias-agiles-de-desarrollo-primera-parte.html>.
7. **GL.** Lorenzo Servidor. *Diccionario*. [En línea] Enero de 2006.
8. **Macad.** ACM International Collegiate Programming Contest. *IBM Student Portal*. [En línea] <http://www-304.ibm.com/jct09002c/university/students/contests/acm/index.html>.
9. Desarrollo Web.com. *Arquitectura del Servidor Apache*. [En línea] Marzo de 2003. <http://www.desarrolloweb.com/articulos/1112.php>.
10. **Herrera, Anielikis y Rodriguez, Yoandy.** *Desarrollo Y Mantenimiento De Una Distribución De Linux*. Ciudad de la Habana : s.n., 2006.

BIBLIOGRAFÍA

William B. Poucher. (n.d.). *ACM-ICPC*. Retrieved Diciembre 2007, from <http://cm2prod.baylor.edu/public/static/staticPages.jsf?page=History&menu=community#main:articlesForm:articles:21:header>

ACM. (2007, Febrero). *ACM-ICPC*. Retrieved Diciembre 2007, from History: <http://icpc.baylor.edu/icpc/finals/frame.asp?page=About.htm>

ACM. (2007). *ACM-ICPC*. Retrieved Abril 2008, from World Finals Contest Image Installation Directions: <http://pc2.ecs.baylor.edu/InstallDirections.html>

Apache Software Foundation. (2007, Junio). Retrieved Marzo 2008, from HttpComponents: <http://www.apache.org/>

Committe Action. (2003, Julio 3). *ACM-ICPC*. Retrieved Octubre 2007, from Policies and Procedures: <http://cm2prod.baylor.edu/resources/pdf/ppgs.pdf>

Common UNIX Printing. (2006). Retrieved Febrero 2008, from Printer Drivers: <http://www.cups.org/>
Desarrollo Web.com. (2003, Marzo). Retrieved Febrero 2008, from Arquitectura del Servidor Apache: <http://www.desarrolloweb.com/articulos/1112.php>

Desarrollo Web.com. (2003).Domino, L. (n.d.). *IBM*. Retrieved Octubre 2007, from IBM-Academic Initiative: <http://www-304.ibm.com/jct09002c/university/scholars/academicinitiative/>

Flower, M. (2003). *ThoughtWorks*. Retrieved Enero 2008, from La Nueva Metodología: <http://www.programacionextrema.org/articulos/newMethodology.es.html>

Fowler, M. (n.d.). *Agile Spain*. Retrieved Enero 2008, from EsElCambioDelInterfazRefactorizar: <http://www.agile-spain.com/agilev2/>

Free Pascal. (2007). Retrieved Enero 2008, from Introduction: <http://www.freepascal.org/>

Gabanes, N. (2005). *Arrakis*. Retrieved Enero 2008, from Diccionario Informático: <http://usuarios.lycos.es/Resve/diccioninform.htm#Autor>

GL. (2006, Enero). *Lorenzo Servidor*. Retrieved Abril 2008, from Diccionario.

Goñi, A., & Herrera, A. (2007). *Proyecto Nova Base para crear un Sistema Operativo con Repercusiones Sociales*. Ciudad de la Habana.

Guia Arte Multimedia. (n.d.). *Desarrollo Web.com*. Retrieved Febrero 2008, from Instalar y Configurar Apache: <http://www.desarrolloweb.com/manuales/41/>

Hartmann, L. (2007, Diciembre). *IOI*. Retrieved Enero 2008, from History: <http://www.ioinformatics.org/history.shtml>

Herrera, A., & Rodriguez, Y. (2006). *Desarrollo Y Mantenimiento De Una Distribución De Linux*. Ciudad de la Habana.

Lycos. (2007). Retrieved Mayo 2008, from Todo de la Informática: <http://usuarios.lycos.es/tutoinformatica/DEPURADOR.html>

Macad. (n.d.). *ACM International Collegiate Programming Contest*. Retrieved from IBM Student Portal: <http://www-304.ibm.com/jct09002c/university/students/contests/acm/index.html>

Monahan, J. (n.d.). *ACM-ICPC*. Retrieved Octubre 2007, from Students from St. Petersburg University of IT, Mechanics and Optics Win: <http://www-03.ibm.com/press/us/en/pressrelease/23868.wss>

Mora, F. L. (2007, Abril). *¿Qué es un compilador?* Ecuador.

Mossberg, W. S. (2005). *CFI en Red*. Retrieved Enero 2008, from Netscape resucita, pero aún puede mejorar: <http://weblogs.cfired.org.ar/blog/archives/001284.php>

Olimpiadas Nacionales de Informática. (2007). Retrieved Enero 2008, from Mooshak: <http://www.dcc.fc.up.pt/oni/2007/index.cgi?page=faq>

Oramas, A. G. (2007). *Nova LNX como Plataforma de Desarrollo Personalizada*. Ciudad de la Habana.

Palacio, J. (2007). *Flexibilidad con Scrum*. Safe Creative.

Procaccino, D. (n.d.). *ACM-ICPC*. Retrieved Septiembre 2007, from Battle Brains: <http://www-03.ibm.com/press/us/en/pressrelease/22314.wss>

Qualitrain Express. (2001). *Blog de Qualitrain*. Retrieved Abril 2008, from Metodologías Ágiles de desarrollo de software: <http://www.qualitrain.com.mx/index.php/Procesos-de-software/Metodologias-agiles-de-desarrollo-primera-parte.html>

Reyes, E. G. (2007, Abril). *webDevStudio*. Retrieved Marzo 2008, from Integrated Development Environment: <http://webdevstudio.wordpress.com/descripcion/>

Ripoll, D. (2007). *Convocatoria Oficial II Competencia Estudiantil por Invitación de Programación (CEIP-UCI)*. Ciudad de la Habana.

Ripoll, D. (2006). *Copa Void de Programación. Resumen General de las ediciones I, II, III, IV, V, VI, VII, VIII, IX, X, XI, XII*. Ciudad de la Habana.

Ripoll, D. (2006). *Sistema PC² para el control de la competencias en la Copa Void de Programación*. Ciudad de la Habana.

Secunia. (2007, Noviembre). *Secunia_Stay Secure*. Retrieved Noviembre 2007, from Secunia Highlights: <http://secunia.com/>

Terán, J. (n.d.). *ACM*. Retrieved Septiembre 25, 2007, from Contest: <http://acm.uva.es/p/>

The 2000's ACM-ICPC Live Archive. (n.d.). Retrieved Septiembre 25, 2007, from <http://acmicpc-live-archive.uva.es/nuevoportal/contests/contest/main.php>

Van, D. (2007, Julio). *IBM*. Retrieved Diciembre 2007, from IBM & Linux: <http://www-03.ibm.com/linux/>

ANEXOS

ANEXO I. GUÍA DE LA ENTREVISTA REALIZADA AL COMPAÑERO FRANK HERNÁNDEZ GONZÁLEZ, ENTRENADOR PRINCIPAL DE LA PRESELECCIÓN NACIONAL DE COMPUTACIÓN.

Objetivo: Obtener información relacionada con las Competencias de Programación que se han desarrollado en la Enseñanza Media Superior en Cuba.

- ¿Qué es una competencia de programación?
- ¿Qué importancia le confiere a las competencias de programación?
- ¿Qué lenguajes de programación que se han usado en las competencias?
- ¿Cómo es el proceso de preparación para las competencias de la IOI?
- ¿Cuáles son las competencias nacionales que se han efectuado en Cuba?
- ¿Qué entornos y herramientas se han utilizado en ellas?
- ¿Siempre se ha usado Windows como Sistema operativo? En caso de no ser así, ¿a partir de que año usan Linux, con qué distribución y qué características?
- ¿Qué razones considera que provocaron o indujeron el cambio hacia Linux y al SWL en general?
- ¿Qué herramientas se han utilizado en las competencias nacionales: editores de texto, jurado, navegadores?
- ¿Qué herramientas Ud. considera que no deben faltar en la personalización que vamos a hacer para competencias de programación?

ANEXO II. GUÍA DE LA ENCUESTA REALIZADA EN LA XVII Y XVIII EDICIÓN DE LA COPA VOID.

Objetivo: Recoger información sobre la opinión de los usuarios de la imagen- cliente.

De forma general la imagen la consideras:*

- Muy buena.
- Buena.
- Regular.
- Mala.
- Muy mala.

Proponga mejoras:

Durante el trabajo con el sistema te sentiste:

- Perdido.
- Incómodo.
- Cómodo.
- Podría haberme sentido mejor.

Esto es debido a que:

- Sé trabajar en Linux.
- El entorno está fácil.
- No sé trabajar con Linux.
- Las herramientas están mal distribuidas.

Proponga mejoras:

La velocidad del sistema está:

- Muy buena.
- Buena.
- Regular.
- Lenta.
- Muy lenta.
- No sé.

Respecto a los editores de texto consideras:

- Está/n el/los que uso.
- Hay muchos que ni los conozco.
- Faltan los más importantes.
- Presentó problemas: _____.

Proponga algunos editores importantes no incluidos:

Respecto al IDE _____ usado por ti en esta imagen consideras:

- El IDE que usé no me dio problema.
- El IDE que usé me dio problema.
- No usé ningún IDE.

Liste los problemas que presentó con su IDE:

Si trabajó con el diccionario opine:

- Trabaja correctamente.
- No supe cómo usarlo.
- Tiene fallos.

Liste los fallos:

Otros criterios o sugerencias:

ANEXO III. TABLA DE ALGUNOS DE LOS COMANDO UTILIZADOS PARA REALIZAR LA PERSONALIZACIÓN.

Tabla I – Comandos utilizados para realizar la personalización.

Comando	Uso	Notas
tar -xvjpgf /home/ana/imagen_base.bz2 /media/disk	Descomprimir	x porque se está descomprimiendo. v (opcional) para ver que sucede durante todo el proceso. j porque se está descomprimiendo con la extensión bzip2. p para preservar los permisos. f para especificar que extraemos el archivo, no la entrada estándar.
mount -o bind /dev/ /media/disk/dev/	montar /dev	
mount -t proc none /media/disk/proc	montar /proc	
nano /etc/make.conf	acceder al archivo make.conf	
emerge subversión -pv	instalar subversion	
svn co http://svn.nova.prod.uci.cu/svn/nova-overlay	copiar el svn de la UCI para el de la maquina	
mount -o bind /usr/local/nova-overlay /media/disk/usr/local/nova-overlay	montar en nova-overlay	/usr/local/nova-overlay es la dirección del overlay del sistema en que se trabaja. /media/disk/usr/local/nova-overlay es la dirección del overlay del sistema que se construye.
mount -o bind /usr/src /media/disk/usr/src	montar /usr/src	
chroot /media/disk	conectarse	

	por chroot	
env-update	se crean las variables de entorno	
source /etc/profile	se cargan estas variables en memoria	
emerge -e system	actualizan los archivos de configuración	
etc-update		
emerge -ND world	actualizar todo los paquetes del sistema	D toma todas las dependencias de los paquetes N toma las nuevas uses que agregaste
emerge xorg-server -av	instalar xorg	
emerge gnome-ligth -av	instalar genome-ligth	
emerge samba	instalar samba	
emerge dhcpcd	instalar dhcp	
emerge gdm	instalar gdm	
rc-update add xdm default	adicionar xdm al default	
rc-update add dbus default	adicionar dbus al default	
emerge pmount	habilitar el permiso a cualquier usuario de montar cualquier dispositivo de E/S.	
emerge gentoolkit	instalar gentoolkit	

ANEXO IV: GUÍA DE LA ENCUESTA EN LA XVIII EDICIÓN DE LA COPA VOID.

Objetivo: Recoger información sobre la opinión de los usuarios de la imagen- servidora.

1. ¿Conoces lo que es un jurado online o sistema de control?

No.

Si.

2. Mi opinión general del sistema moohak es:

Muy buena

Buena

Prefiero otros como:

3. Los resultados de los problemas fueron devueltos:

Muy rápido.

Rápido.

Lento.

4. Al conectarme al servidor:

No tuve problemas.

Tuve problemas, ellos fueron:

5. ¿Crees que Mooshak facilita el proceso de enseñanza-aprendizaje para el trabajo con los jurados online?.

Muy de acuerdo.

De acuerdo.

Indiferente.

En desacuerdo.

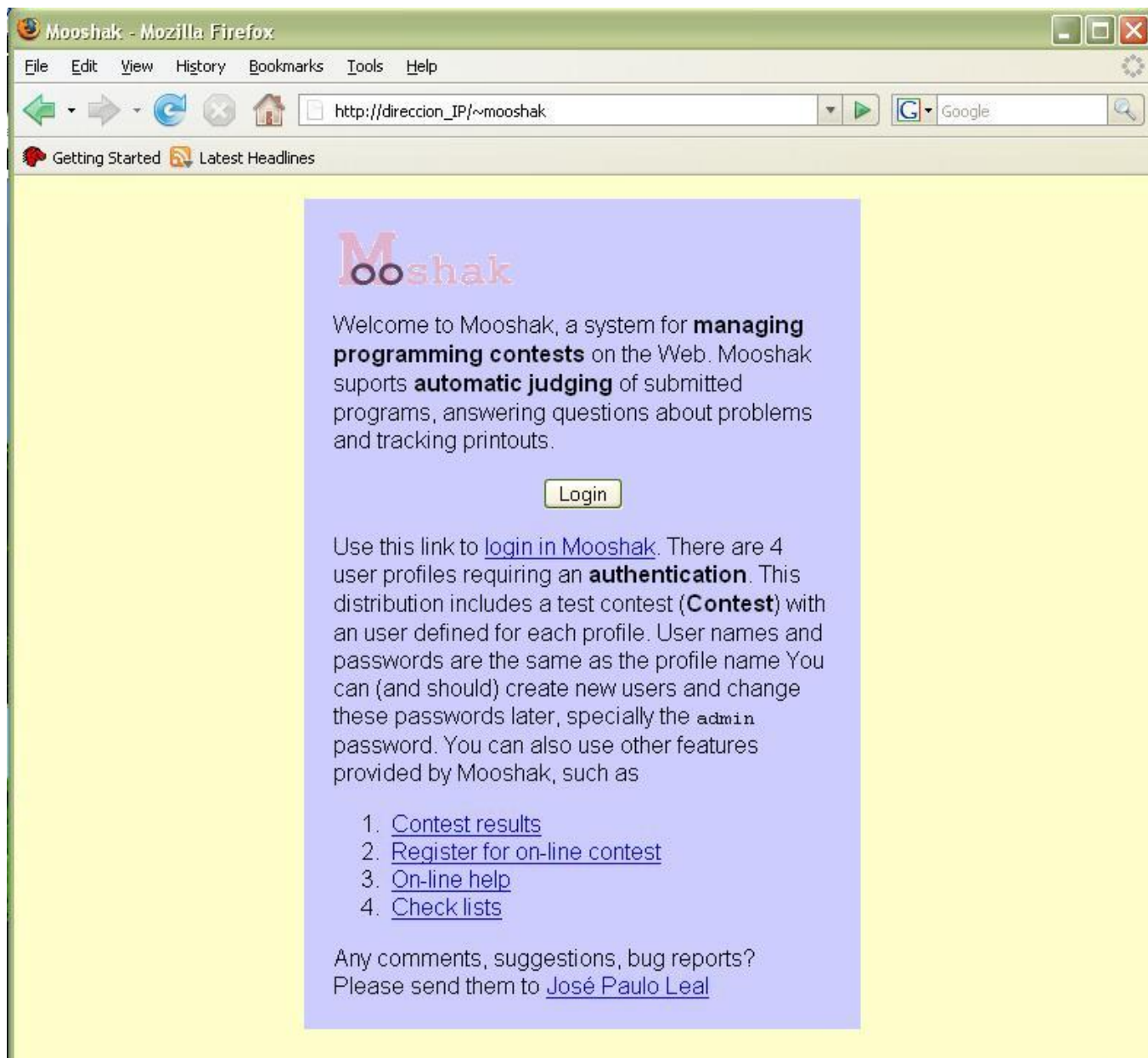
Muy en desacuerdo.

No sé.

ANEXO V



ANEXO VI



GLOSARIO DE TÉRMINOS

Cliente: Ordenador que accede a recursos y servicios brindados por el servidor..

Distribución Libre: Es una variante de sistema operativo Gnu/Linux, que incorpora determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones hogareñas, empresariales y para servidores. Pueden ser exclusivamente de software libre, o también incorporar aplicaciones, así como productos o paquetes bajo otras licencias más restrictivas.

Entorno de escritorio: Es un conjunto de software que ofrece al usuario de una computadora una interacción amigable y cómoda.

Fedora Core 6: Sistema operativo de propósito general y basado exclusivamente en software libre con el apoyo de la comunidad Linux.

Ficheros RPM: Son archivos que llevan incluidos todos los ficheros que componen un determinado programa. Dentro de él van los ficheros del programa a instalar, su descripción, a que directorios van a ir instalados, scripts de auto-configuración en algunos casos, etc.

Gnome: Es un entorno de escritorio para sistemas operativos de tipo Unix bajo tecnología X Window.. Nació como una alternativa a KDE

Gentoo: Es una distribución GNU/Linux que fue fundada por Daniel Robbins. El nombre Gentoo proviene del nombre en inglés del pingüino papúa, de allí proviene que su mascota sea un pingüino.⁴

Gtk: Es una biblioteca que posibilita la creación de interfaces gráficas de usuario (Graphical User Interfaces, GUI). Está disponible bajo licencia GPL. Utilizando dicha biblioteca pueden crearse programas de código abierto, tanto libres como comerciales.

Colección de Compiladores: Es un conjunto de compiladores creados por el proyecto GNU⁵

⁴ Tomado de la wiki http://es.wikipedia.org/wiki/Gentoo_Linux

⁵ Tomado de la wiki http://es.wikipedia.org/wiki/GNU_Compiler_Collection

CGI: Common Gateway Interface es un estándar para comunicar aplicaciones externas con los servidores Web. Un programa CGI es un ejecutable.

Initramfs: Es un pequeño sistema de archivos que el kernel o núcleo puede cargar en un disco RAM. Proporciona un entorno Linux mínimo que habilita la ejecución de programas antes de que se monte el sistema de archivos raíz..

Kde: K Desktop Environment o Entorno de Escritorio K, es un entorno de escritorio e infraestructura de desarrollo para sistemas Unix/Linux.

Metadatos: Es la información insertada en los archivos por el software de edición o creación de los mismos, contienen información acerca de la creación del archivo como: nombre de autor, autores anteriores, nombre de compañía, cantidad de veces que el documento fue modificado, fecha de creación, etc.

Multiplataforma: Es todo software, sistema operativo o lenguaje de programación que puede funcionar en diferentes sistemas operativos y/o ordenadores, pero el código fuente es el mismo.

Plugins: Es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica.

Root: También conocido por superusuario, es el nombre de la cuenta de usuario que posee todos los derechos de administración de un sistema operativo de tipo Unix.

Secunia: Organización de gran prestigio a nivel internacional en lo que se refiere a seguridad.

Servidor: Una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes.

Servidor web: Máquina que almacena y maneja los sitios web.

Software libre: Es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Suele estar disponible gratuitamente, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no se puede asociar el

término software libre con el de software gratuito, pues conservando su carácter de libre, puede ser distribuido comercialmente.

Software propietario: Se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o cuyo código fuente no está disponible o el acceso a éste se encuentra restringido. No ofrece garantías de tipo alguno, y no permite su uso a los a los usuarios en cualquier cantidad de terminales, a menos que se exprese y se pague concretamente.

SSI: Es una herramienta muy útil para el diseño de un sitio web. Es una opción básica para usar código embebido y ofrece un modo sencillo de obtener contenido dinámico.

STL documentation: Del inglés Standard Template Library, librerías para el lenguaje C++ que incluye estructuras de datos, algoritmos.