



Universidad de las Ciencias Informáticas

Facultad 7

Título:

Herramienta para el control y evaluación del tiempo de producción.

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autores: Yosmany Costa Ruiz

Reiniel Alvarez Pérez

Tutores: MsC. Eduardo Solís Céspedes

Ing. Ricardo Collada de la Rosa

Ciudad de La Habana, Junio 2010

“Año 52 de la Revolución”

Resumen

Este trabajo tiene como objetivo crear un mecanismo capaz de llevar a cabo una serie de reportes de lo que hacen los estudiantes en su tiempo de máquina en los laboratorios de producción. Teniendo como propósito elaborar una herramienta automatizada para el control y evaluación del tiempo de producción.

Para realizar dicha tarea se lleva a cabo el diseño e implementación de un sistema que permita controlar y establecer el uso de los procesos en una Computadora Personal (PC) con Sistema Operativo (SO) Linux basados en distribuciones Debian e informe a un servidor donde se procesen las estadísticas correspondientes. Para ello se investiga sobre las aplicaciones controladoras de procesos, así como las aplicaciones para el control del tiempo de producción, se analizan sus características y forma de funcionar y a la hora de diseñar e implementar el sistema, se usan diferentes lenguajes de programación y una base de datos.

Con la puesta en práctica de esta herramienta automatizada se podrá llevar un control más rígido, además de aumentar la eficiencia y disminuir la pérdida de tiempo en los laboratorios de producción, lo que permitirá al líder de proyecto conocer con datos específicos qué está fallando en el grupo de desarrollo, así como la preferencia de cada integrante de su equipo. Esto ayudará en gran medida a que las planificaciones del proyecto tengan éxito en su desempeño.

PALABRAS CLAVE

Control y evaluación, planificaciones, procesos y servicios, Sistema Operativo



Tabla de contenido

Resumen.....	I
Introducción	1
1. Capítulo 1. Fundamentación teórica	5
1.1. Análisis de otras soluciones existentes para el control del tiempo de producción.	5
1.1.1. Sistema para el control de la asistencia a la producción (SCAP).	5
1.1.2. Sniffer	6
1.2. Sistemas de monitoreo de procesos.....	7
1.2.1. Funcionamiento de los sistemas de monitoreo.....	8
1.2.2. Características de los sistemas de monitoreo	8
1.3. Comandos para monitorear procesos.....	9
1.4. Sistema de ventanas X Window System.....	11
1.4.1. Arquitectura de X Window	12
1.5. Modelo Cliente/Servidor.....	13
1.6. Herramientas, lenguajes y tecnologías a utilizar	15
1.6.1. Entorno integrado de desarrollo.....	15
1.6.2. Lenguajes de programación utilizados.	17
1.6.3. Socket.....	21
1.6.4. Bash.....	22
1.6.5. Joomla	22
1.6.6. Apache	23
1.6.7. MySQL.....	23
1.7. Metodología ágil a utilizar	24
2. Capítulo 2. Descripción y análisis de la solución propuesta	27
2.1. Propuesta del sistema a implementar.....	27



2.2. Modelo de Dominio.....	28
2.2.1 Diagrama de Modelo de Dominio.....	29
2.2.2. Descripción de los términos comunes.....	30
2.3. Reutilización de código	31
2.4. Por qué utilizar el sistema X Window System.....	31
2.5. Xlib y XCB	32
2.6. Estrategia de integración	33
3. Capítulo 3. Desarrollo ágil de la aplicación propuesta	35
3.1. Lista de Reserva del Producto	35
3.2. Requisitos No Funcionales	36
3.2.1. Software	36
3.2.2. Usabilidad.....	36
3.2.3. Fiabilidad.....	36
3.2.4. Rendimiento.....	37
3.3. Descripción de la arquitectura.....	37
3.4. Diseño de la base de datos	41
3.5. Diagrama de despliegue	41
3.6. Historias de usuarios.....	42
3.7. Tareas de Ingeniería	50
4. Capítulo 4. Validación de la solución propuesta	62
4.1. Casos de prueba de aceptación.....	62
Conclusiones.....	70
Recomendaciones	71
Referencias Bibliográficas.....	72

Introducción

En el mundo de hoy las tecnologías muestran un creciente desarrollo, ya que han conseguido un gran protagonismo en la comunidad científico-tecnológica y al mismo tiempo, la informática dentro de las ciencias que apoyan ese crecimiento ha aportado cambios significativos. El incremento vertiginoso de las producciones de software, ha impuesto a la competencia, productos con rapidez y con mayor calidad. Su impacto a nivel global se ve reflejado en todas las esferas de la sociedad. La gran mayoría de los trabajos, trámites y procesos que se realizan en la actualidad están vinculados a la informática, convirtiéndose en una nueva fuente económica para muchos países.

Es por eso por lo que las organizaciones mundiales, para el logro de sus objetivos, demandan la construcción de grandes y complejos sistemas informáticos que requieren de la combinación de diferentes componentes de hardware y software para alcanzar un funcionamiento acorde con dichas necesidades. Todo ello propicia que los avances tecnológicos centren sus objetivos principalmente en la producción de software.

En muchos países la industria de software se ha convertido en uno de los renglones de la economía que más aporta a su desarrollo e incluso ha llegado a ser un elemento clave en su situación política y social. Los ingresos por cuestiones de exportaciones de software han alcanzado cifras importantes, a tal estado, que sobrepasan las producciones de petróleo, diamantes y otros productos de gran valor en el mercado internacional.

Uno de los inminentes usos que tienen las tecnologías de la información y las comunicaciones (TIC), es que apuntan, entre otras cosas, a la automatización de diferentes procesos que intervienen en la industria del software. Cuba es uno de los países que ha despuntado, mostrando grandes perspectivas en este renglón. La informática en el país se va convirtiendo en una fuente de desarrollo económico-social, para esto se viene realizando un trabajo significativo en materia de informatización de la sociedad.

Es por eso por lo que en el año 2002, con el desarrollo del proceso de informatización de la sociedad cubana, surge la Universidad de las Ciencias Informáticas (UCI) con el

objetivo de ayudar e impulsar el desarrollo de la producción de software en el país. Esta Universidad, se ha convertido en la principal empresa del desarrollo tecnológico para Cuba. La misma cuenta con un modelo de formación centrado en la vinculación del estudio y la producción, para ello la mayoría de los estudiantes se encuentran vinculados a un proyecto productivo. Estos proyectos contribuyen además con los planes de recuperación económica y mejoras sociales del país, ya sea por concepto de exportación o reducción de las importaciones. La UCI ha aportado recursos que permiten un mayor beneficio de la sociedad.

Para el desarrollo de los proyectos se confía en la responsabilidad y compromiso de los estudiantes, sin embargo, en ocasiones sucede que se atrasan los cronogramas de desarrollo o la realización de una tarea. Según una encuesta que se aplicó a algunos estudiantes y profesores de la facultad 7, se pudo analizar que un factor que afecta seriamente el proceso productivo es que sus principales protagonistas, los estudiantes, se distraen y en ocasiones emplean mucho tiempo en entretenimientos como los chats, juegos de computadoras, los videos, películas, series de TV, entre otros.

Para los pocos profesores que existen en cada laboratorio de producción es muy difícil tener un control fiable del empleo del tiempo de máquina por parte de los estudiantes. La situación anterior conlleva a la necesidad de la creación de una herramienta de supervisión de tareas productivas para lograr un mejor cumplimiento de los cronogramas de trabajos. Esto evita que se violen políticas del reglamento para el uso de los laboratorios, que se logre una mejor superación de los estudiantes, que se proteja el equipamiento tecnológico y se logre una mayor concentración de las actividades a realizar y de los temas que están pendientes de solución.

Teniendo en cuenta los aspectos descritos con anterioridad se puede definir como **problema científico**: ¿Cómo controlar el aprovechamiento del tiempo de máquina en los laboratorios de producción?

Por lo que el **objeto de estudio** se basa en un sistema para el monitoreo, control y evaluación de tareas, delimitando así como el **campo de acción** el monitoreo de procesos y ventanas enfocadas en un ordenador con sistemas operativos GNU/Linux basados en distribuciones Debian. El **objetivo general** de la investigación es implementar

una aplicación Cliente/Servidor que permita supervisar los procesos para optimizar el uso de los ordenadores en los laboratorios de producción.

Con vista a darle cumplimiento al objetivo planteado se han definido las siguientes **tareas de investigación**:

1. Analizar el funcionamiento de los árboles de procesos en SO Linux basados en distribuciones Debian.
2. Determinar el lenguaje de programación, las herramientas y tecnologías para desarrollar el sistema.
3. Diseñar la herramienta en dos partes: una del cliente y otra del servidor.
4. Implementar la aplicación cliente que recoja el uso de los procesos y los envíe al servidor.
5. Implementar la aplicación servidora que reciba información de los clientes y realice gráficos estadísticos.
6. Validar el sistema en un ambiente de prueba.

El presente trabajo se encuentra estructurado por cuatro capítulos cuyo contenido se describe a continuación:

Capítulo 1: Fundamentación teórica: Se realiza un análisis sobre los diferentes sistemas de control del tiempo de producción, además de los diferentes sistemas de monitoreo, así como del funcionamiento del sistema de ventanas X Window System¹. Además, se describen las tecnologías, lenguajes y herramientas que dan solución al sistema propuesto.

Capítulo 2: Descripción y análisis de la solución propuesta: Se hace un análisis de la solución propuesta. Se especifica y justifica la utilización de herramientas ya existentes que formarán parte de la solución del problema a resolver, así como la estrategia de integración del software.

¹ El sistema X Window System (abreviado X o X11 para la versión 11) es un método independiente para proporcionar capacidades gráficas a un Sistema Operativo.

Capítulo 3: Desarrollo ágil de la solución propuesta: Se realiza el desarrollo ágil del software, utilizando como metodología de desarrollo a Scrum y Extreme Programming (XP). También se hace una descripción del funcionamiento del software a través de lista de reserva de productos, historias de usuarios y prototipos de interfaz de usuario. Además de explicar las tareas de ingeniería.

Capítulo 4: Validación de la solución propuesta: Se describen los casos de pruebas realizados a la aplicación para algunos casos de historias de usuarios, se muestran los resultados, se analizan y se presentan las funcionalidades alcanzadas hasta el período de desarrollo.

1. Capítulo 1. Fundamentación teórica

En este capítulo se analiza el funcionamiento de los sistemas de monitoreo de procesos, así como del control del tiempo de producción, se estudia la situación actual en el mundo y nacional acerca del desarrollo de este tipo de sistemas, se identifica el mecanismo de monitoreo más apropiado para realizar la actividad de control de procesos y se analiza el funcionamiento del sistema de ventanas X Window, así como, herramientas, tecnologías a utilizar y metodología de desarrollo a emplear.

1.1. Análisis de otras soluciones existentes para el control del tiempo de producción.

En Cuba, principalmente en la UCI, se han desarrollado herramientas o software con el objetivo de controlar de una forma u otra el tiempo de producción de los estudiantes, para tener un mejor aprovechamiento de su uso en los laboratorios de producción.

1.1.1. Sistema para el control de la asistencia a la producción (SCAP).

El SCAP es una aplicación web para llevar el control de las ausencias a las secciones de trabajo de la producción. Este sistema posee varios módulos dentro de los que se encuentra: el Sistema Matriz, el cual gestiona la estructura del área productiva: laboratorios, estaciones de trabajo, grupos productivos, estudiantes y profesores, gestiona el horario productivo por grupos de desarrollo, gestión del reloj de entrada y salida de las secciones de trabajo. También posee los procesos para el control de la asistencia: pases de lista, llegadas tardes, justificación, incidencias, presentes, ausentes, resúmenes, por ciento de asistencia de los estudiantes en un período determinado.

SCAP tiene como principales funcionalidades:

- 1- Gestión de la asistencia:
 - Estado del laboratorio.
 - Buscar estudiante.
 - Controlar la asistencia.
 - Buscar ausencias.
 - Resúmenes de ausencias.

- Estadísticas de ausencias del estudiante.
- 2- Configuración del sistema:
 - Crear laboratorios, grupos de producción.
 - Asignar puestos de trabajo a los grupos de producción.
 - Gestionar profesores.
 - Gestionar horario productivo.
 - Gestionar reloj de entrada y salida de las secciones de trabajo.
- 3- Accesos rápidos:
 - Estudiantes por máquinas.
 - Acceso por el teclado.
- 4- Cliente comunicador:
 - Marcar la entrada y salida de los estudiantes.
 - Brindar información del por ciento de asistencia de los estudiantes.
 - Controlar los programas instalados según la configuración por grupos de desarrollo.
 - Interrumpir la ejecución de programas prohibidos.
- 5- Buscador:
 - Mostrar información de la ubicación de un estudiante dentro del proyecto.
- 6- Sistema de mensajería:
 - Enviar y recibir mensajes instantáneos con los estudiantes que trabajan en el proyecto.

1.1.2. Sniffer

El Sniffer es un sistema o un conjunto de aplicaciones que permite monitorear los procesos en el Sistema Operativo Windows. Está formado por una aplicación cliente que se ejecuta de manera automática en los ordenadores clientes y realiza un monitoreo de todos los procesos que se encuentran activos, controlando de ellos el nombre, la hora de inicio, la hora de fin y el tiempo que estuvo activo. Dicha información, acompañada del nombre de usuario, el dominio y la dirección IP² del ordenador, son enviadas a un servidor. El servidor cuenta con una base de datos, en la cual se almacenan de forma

² Número que identifica un ordenador dentro de una red que utilice el protocolo de Internet.

eficiente todos los datos recolectados y luego son visualizados mediante reportes en una aplicación web. Esta aplicación web cuenta con tres niveles de acceso o roles:

- Usuario o cliente, que solo tiene acceso a su información.
- Líder o jefe de proyecto, que puede visualizar todo lo referente a los clientes que estén bajo su tutela.
- Administrador, el cual puede ver toda la información almacenada y modificarla.

Los sistemas descritos anteriormente solucionaban parcialmente el control del tiempo de producción, pero los mismos no constituían mecanismos totalmente fiables para el total control del mismo. Evaluando detalladamente los sistemas, se llegó a la conclusión de que SCAP aún presenta módulos que se encuentran en la fase de implementación, los cuales desempeñan un papel protagónico en la notificación de eventos relacionados con el control del tiempo de máquina de los estudiantes. Este sistema basa sus reportes en momentos determinados, o sea, el mismo no es capaz de llevar un control permanente del quehacer diario de un estudiante en su tiempo de máquina.

El Sniffer por su parte, posee un control fiable de los estudiantes, pero el mismo posee muy poca documentación y resultados, debido a que las pruebas realizadas a este se llevaron a cabo en un marco muy estrecho. Este software basaba su desarrollo en el SO Windows.

Luego de un estudio de las herramientas existentes para el control del tiempo de producción, se llegó a la conclusión de que ninguna representa una solución integral a la problemática existente. Las aplicaciones que controlan procesos en el Sistema Operativo GNU/Linux, no permiten conocer las características de las ventanas construidas por el servidor a petición de un programa cliente, ni tampoco notificar eventos, por lo que se decide llevar a cabo el desarrollo de este trabajo.

1.2. Sistemas de monitoreo de procesos

El monitoreo es un proceso continuo de recolección y análisis de datos cualitativos y cuantitativos. Permite realizar un conjunto de acciones periódicas y sistemáticas, de

vigilancia, observación y medición de los parámetros relevantes de un sistema. Estos obtienen resultados acerca de las condiciones en tiempo real de una computadora, pueden registrar, por ejemplo, las estadísticas de los recursos utilizados por el equipo.

Un proceso es una instancia de un programa de ejecución, a los cuales frecuentemente se les refiere como tareas. Este contexto puede englobar el de procesos hijos, que serían aquellos que se generaron del principal (proceso padre), los recursos del sistema que esté consumiendo, sus atributos de seguridad, etc. Cada proceso que se inicia es referenciado con un número de identificación único, que es siempre un entero positivo. (1)

1.2.1. Funcionamiento de los sistemas de monitoreo

Los sistemas de monitoreo realizan tareas de control constantemente. Se comunican con procesos que supervisan el SO, que contienen información sobre el ordenador y para capturar los datos, utilizan funciones de bajo nivel.

La monitorización de los sistemas puede realizarse a través de dos mecanismos, monitorización remota y agentes locales instalados. La primera de estas, se realiza desde un ordenador hacia uno o varios de estos. Para establecer una conexión entre ambas partes, utilizan un protocolo³ de red. Este tipo de monitoreo, puede representar de manera gráfica las acciones que se cometen en el ordenador que se está controlando.

El agente local, consiste en una aplicación que supervisa recursos del SO en el ordenador instalado. Su funcionamiento frente al monitoreo remoto básicamente es el mismo, la diferencia radica en que las herramientas locales no requieren de la conexión a una red, y por lo tanto, la implementación es menos compleja y riesgosa.

La actividad de monitoreo varía en cuanto a la necesidad de lo que se quiera controlar, puede ser, el estado de los servicios de red, procesos de un ordenador o parámetros físicos del hardware. Teniendo en cuenta el propósito para el que se controla, el sistema puede ser visible o no al usuario (objeto de vigilancia).

1.2.2. Características de los sistemas de monitoreo

El funcionamiento de los sistemas remoto, exige alta velocidad de conexión para obtener un rendimiento óptimo cuando la distancia y la cantidad de equipos son considerables.

³ Método estándar que permite la comunicación entre procesos (que potencialmente se ejecutan en diferentes equipos).

Son muy usados por la comodidad con que reflejan sus datos y generalmente están orientados al control de redes. Se recomienda su uso cuando se cuenta con un servicio de red fiable, ya que dependen totalmente de esta, por lo que no garantizan el control total de los recursos del sistema permanentemente. Su empleo puede ser muy costoso.

Las aplicaciones locales son ligeras, escasamente requieren recursos del SO y aseguran el control del mismo mientras es usado. Son independientes de la conexión a una red y facilitan integrarse a esta. Se sugiere su empleo, cuando es indispensable que un ordenador sea chequeado constantemente.

Este trabajo está centrado en la supervisión de procesos y manejo de ventanas que están siendo usadas, en especial la contenedora del foco, es decir, la que está activa en el momento dado. Se fundamenta en aplicaciones instaladas en cada ordenador, puesto que se requiere un constante monitoreo. Se independiza de la conexión remota, para evitar que las fallas en el servicio de red interrumpan el control de las actividades que se llevan a cabo en el equipo. Incluye además la notificación de eventos.

Un evento es cualquier suceso que activa un proceso en un sistema automatizado. Este término es muy empleado en las herramientas de supervisión, para advertir la aparición o desaparición de nuevos procesos. Puede ser empleado para distintos fines en la programación y está estrechamente relacionado con el término notificación⁴.

La notificación de eventos es un recurso que se utiliza en el monitoreo, para controlar las aplicaciones que se ejecutan.

1.3. Comandos para monitorear procesos

En los sistemas Unix se llama proceso a un programa en ejecución y al objeto abstracto que crea el SO para manejar el acceso a los recursos del sistema (memoria, Unidad Central de Procesamiento, dispositivos de Entrada y Salida (E/S)). Pueden coexistir varias instancias de un mismo programa ejecutando en forma simultánea, siendo cada una de ellas un proceso diferente.

Unix es un sistema multiproceso por tiempo compartido. A los ojos de un usuario en un momento dado hay múltiples programas en ejecución, cada uno de ellos avanzando en su tarea. Sin embargo, en una máquina con un solo procesador hay en cada instante solo un proceso en ejecución. Es el SO el que va rotando el uso del procesador a intervalos

⁴ Es el acto de comunicar una tramitación o petición.

breves (algunas decenas de milisegundos) entre los procesos definidos en el sistema, de forma que se crea la ilusión que todos avanzan simultáneamente.

Para cada proceso definido en el sistema, el kernel⁵ del Sistema Operativo Linux almacena y mantiene al día varios tipos de información sobre el proceso. Esta información quedaría ordenada de la siguiente forma:

- 1- Información general. Identificadores de proceso, usuario y grupo.
- 2- Ambiente (variables, directorio actual, etc.).
- 3- Información de los dispositivos de E/S.
- 4- Información de estado.

Los sistemas Unix cuentan con diferentes comandos que admiten visualizar los procesos en ejecución, estos son:

- ps: Imprime la lista de procesos en ejecución. Los campos de información más importantes desplegados por ps para cada proceso son: usuario, identificadores de proceso, uso de recursos reciente y acumulado, estado del proceso y comando invocado.
- top: Muestra los procesos usados en forma de tabla. Actualiza un reporte similar al generado por ps a intervalos periódicos. Normalmente, top ordena los procesos por uso de CPU en forma decreciente. Al comando top se le ha desarrollado versiones gráficas para entornos de escritorio, como gtop para Gnome⁶ y ktop para KDE⁷.
- pstree: Muestra los procesos en forma de árbol.
- w y uptime: Estos comandos reportan tres números que indican la carga promedio del sistema en un intervalo de 1, 5 y 15 minutos respectivamente. El parámetro reportado es el promedio de la cantidad de procesos listos para correr. Este valor es un indicador rápido de la actividad del sistema y suele vigilarse para detectar sobrecargas en el mismo, o registrarse periódicamente para un análisis posterior en caso de problemas.
- vmstat: Reporta varias estadísticas que mantiene el núcleo (kernel) sobre los procesos, la memoria y otros recursos del sistema. Alguna de la información reportada por vmstat es la siguiente:

⁵ Facilita a los programas el acceso seguro al hardware y administra los recursos del sistema.

⁶ Es un entorno de escritorio para sistemas operativos de tipo Unix bajo la tecnología X Window.

⁷ Acrónimo que significa "K Desktop Environment", uno de los GUI más importantes para sistemas UNIX.

- 1-Cantidad de procesos en diferentes estados.
- 2- Valores totales de memoria asignada a procesos y libre.
- 3- Estadísticas sobre paginado de memoria.
- 4- Uso de CPU, ejecutando en modo usuario y modo kernel.

Estos programas tienen un grupo de opciones que son utilizadas para mostrar datos como la carga de la Unidad Central de Procesamiento, el uso de la memoria RAM, el usuario y el identificador correspondiente a cada proceso, entre otros.

1.4. Sistema de ventanas X Window System

Los programas controladores, descritos anteriormente, permiten conocer el nombre de la aplicación que se ejecuta, pero no las características de la ventana que responde a un proceso.

Una ventana contiene los siguientes atributos: (2)

- Atributos asociados a la geometría de la ventana: posición, altura, ancho del borde y orden de apilamiento.
- Atributos no configurables (fijados en la creación): profundidad, clase (de entrada/salida o sólo de entrada), visual (tipo de funcionamiento del hardware gráfico).
- Atributos configurables: fondo, nombre de la ventana, borde, gravedad de bit, gravedad de ventana, contenido de la ventana mientras la visibilidad es parcial, total o ninguna, manejo de eventos, mapa de color y la posición del cursor.

El presente trabajo está enmarcado en supervisar ventanas y conocer las características de esta, como el nombre y la aplicación a la que pertenece. Para obtener esta información, es imprescindible estudiar el funcionamiento del programa que las crea, e interactuar con este. X Window es un potente y complejo sistema de ventanas, convertido en el sistema de ventana estándar de los sistemas Unix, y en especial de GNU/Linux. Es distribuido independiente de la arquitectura de la máquina, del SO utilizado y del tipo de hardware empleado por la computadora. Fue creado con el objetivo de facilitar el desarrollo de programas con interfaz gráfica de manera sencilla.

1.4.1. Arquitectura de X Window

El sistema de ventanas X Window está basado en la arquitectura cliente/servidor, donde el servidor se encuentra en cualquier computadora de la red, y el cliente en cualquier computadora conectada a esta. El servidor es un programa dedicado a suministrar servicios de display⁸ en una terminal gráfica, a petición del usuario. Además, controla la pantalla y maneja el teclado y el ratón (u otros dispositivos de entrada).

Igualmente, es responsable de la salida sobre la pantalla, el mapeado de colores, la carga de fuentes y el mapeado del teclado. Típicamente, los servidores se ejecutan en máquinas y terminales de trabajo de tipo gráfico y de alto rendimiento.

Los programas clientes que realizan peticiones al servidor son:

Aplicaciones: Programas que facilitan al usuario la realización de un determinado tipo de trabajo.

Gestor de ventanas: El gestor de ventanas, es un programa cliente encargado de gestionar la manera en que son presentadas las ventanas en pantalla. Cuenta con algunos privilegios que le permiten mostrar la interfaz; para lograrlo, establece una comunicación con el servidor X y este construye y mapea la ventana que es solicitada. La comunicación en este esquema, se realiza mediante cuatro mensajes:

- **Petición**: son mensajes de solicitud de servicio que envía el cliente al servidor, tales como: peticiones de creación, dimensionado o movimiento de una ventana y dibujo de primitivas gráficas.

Estos mensajes son atendidos en el mismo orden en que son solicitados.

- **Respuesta**: posible mensaje de respuesta, en caso de que el cliente necesite una información proveniente del servidor.
- **Evento**: es un mensaje enviado del servidor al cliente cuando ocurre un evento en el ordenador que constituye el servidor, como pulsación de teclado o botón del mouse⁹, movimiento de una ventana, etc.
- **Error**: Es enviado por el servidor al cliente en caso de error.

El intercambio de estos mensajes se realiza a través del protocolo X. Este se ejecuta sobre la conexión de red y permite que se efectúen solicitudes y respuestas entre cliente y

⁸ Un display puede definirse como una o más pantallas y sus dispositivos de entrada.

⁹ Dispositivo de entrada que rueda sobre una superficie plana (por lo general en el escritorio) y controla el puntero de una computadora.

servidor. Existen librerías que permiten desde la programación, interactuar con los eventos que se producen entre un cliente y el servidor, a través de funciones comprensibles por ambos.

1.5. Modelo Cliente/Servidor

En el presente trabajo se requiere de centralizar los reportes producidos por la aplicación local instalada. Para ello se hace imprescindible buscar una arquitectura, esta es un diseño estructural que gobierna la interconexión y la interacción de los componentes que se conectan. Actualmente existen numerosas arquitecturas de este tipo, entre las más aplicadas están:

1. Arquitectura Cliente-Cola-Cliente.
2. Arquitectura Redes de Pares (Peer to Peer (P2P)).
3. Arquitectura Multicapas.
4. Arquitectura Cliente /Servidor.

Las redes de pares se basan en la filosofía de compartir ficheros directamente de forma descentralizada y distribuida. Es muy utilizada cuando se trata de compartir datos, ya que todas las computadoras interconectadas gozan de la misma categoría y cada una brinda a la red, los recursos que su propio usuario le asigne. La arquitectura Cliente-Cola-Cliente, en la cual las redes P2P se basó inicialmente, habilita todos los nodos de la red para tratarlos como clientes simples. Esta arquitectura, al igual que la P2P no es factible para la solución de este trabajo, porque la información recopilada no tiene como propósito viajar hacia los demás ordenadores de la red para ser conocimiento de cualquier usuario. El modelo multicapa, a diferencia de la arquitectura cliente/servidor, posee en algunos casos tres tipos de nodos en la red.

- Clientes que interactúan con los usuarios finales.
- Servidores de aplicación que procesan los datos para los clientes.
- Servidores de la base de datos que almacenan los datos para los servidores de aplicación.

Puesto que en este trabajo no se procesan datos para los clientes, sino que solo se almacena la información, no se necesita este modelo. Se utiliza la arquitectura cliente/servidor, ya que la información coleccionada tiene que estar centralizada.

Cliente/Servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí, para intercambiar información, servicios o recursos.

El cliente es un programa que realiza peticiones e inicia un requerimiento de servicio.

Principales funciones del cliente:

- Generar consultas.
- Mantener y procesar el diálogo con el usuario.
- Manejar posibles errores.
- Permitir entrada de datos y validación.
- Gestionar ayuda.

El servidor es cualquier recurso de un cómputo conectado en una red dispuesto a proveer múltiples servicios a los clientes.

Principales funciones del servidor:

- Almacenar y organizar los datos.
- Actualizar los datos almacenados.
- Administrar recursos compartidos.
- Ejecutar toda la lógica para procesar una transacción.
- Procesar los elementos del servidor (datos, capacidad de CPU, almacenamiento en disco, capacidad de impresión, manejo de memoria y comunicación).
- Gestionar periféricos compartidos.
- Controlar el acceso concurrente a bases de datos compartidas.
- Permitir enlazar las comunicaciones con otras redes.

Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor.
- El cliente no depende de la ubicación física del servidor, del tipo de equipo físico en el que se encuentra, ni de su SO.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Esta arquitectura reparte la capacidad de proceso entre el cliente y el servidor. La asignación de responsabilidades y la organización de sus componentes constituyen

ventajas en un sistema. La división de estas dos partes, cliente y servidor, es conocido como separación de tipo lógico, debido a que el servidor no necesariamente se ejecuta en una sola máquina. Es recomendable que el servidor sea una máquina bastante potente, ya que necesita de más recursos para responder las solicitudes.

El servidor al iniciarse espera a que lleguen solicitudes de los clientes, por lo que adopta una posición pasiva en la comunicación. Al recibir una petición, la procesa y luego responde a los clientes. Tienen la capacidad de atender a varios clientes, aunque se puede restringir a un número limitado. Los clientes se mantienen activo, debido a que inician la comunicación con el servidor. Normalmente, pueden conectarse a varios servidores y permanecer en espera hasta que se responda su demanda.

1.6. Herramientas, lenguajes y tecnologías a utilizar

Para lograr un producto es importante el uso de tecnologías, lenguajes de programación y herramientas que ayudan a la solución del problema. Para solucionar el problema planteado se tiene en cuenta la necesidad de un desarrollo que controle ventanas en el Sistema Operativo GNU/Linux y el uso de herramientas libres, para construir un programa de código abierto.

1.6.1. Entorno integrado de desarrollo

Para implementar la aplicación, es ventajoso utilizar un Entorno Integrado de Desarrollo (IDE) que facilite la programación. Este es un conjunto de herramientas útiles para el programador. Los componentes más comunes de un IDE son: editor de texto, compilador, intérprete, depurador y los plugins¹⁰ que cada IDE contenga como característica especial. Para el desarrollo de este trabajo se utilizaron los siguientes:

Anjuta:

Es un (IDE) desarrollado por Naba Kumbar en 1999. Utilizado inicialmente para programar en C y C++ en sistemas GNU/Linux aunque actualmente admite lenguajes como C#, Perl, Python y Bash. Su principal objetivo es trabajar con GTK¹¹ y en el escritorio GNOME,

¹⁰ Un plug-in es un módulo de hardware o software que añade una característica o un servicio específico a un sistema.

¹¹ Biblioteca que contiene objetos y funciones para desarrollar interfaces gráficas de usuarios.

además ofrece un gran número de características avanzadas de programación. Anjuta es un software libre, liberado bajo la licencia GPL¹².

Anjuta incluye: (3)

- Administrador de proyectos.
- Asistentes.
- Plantillas.
- Depurador interactivo.
- Editor de texto; que verifica y resalta las sintaxis escritas.
- Extensión para Subversion (sistema de control de versiones).
- Intérprete de comandos propio.

A pesar de incluir abundantes utilidades y tener una amplia variedad de funcionalidades es un sistema bastante estable y muy ligero en el consumo de recursos. Por sus características es usado para la realización de script¹³ en el lenguaje Shell Script usando como intérprete a Bash.

Code::Blocks:

Es otro de los IDE empleado para la implementación de la herramienta que se propone, soporta los lenguajes de programación C y C++. Proporciona a los usuarios una completa interfaz de trabajo para escribir su código con facilidad, diseñado para ser muy extensible y totalmente configurable. Es una herramienta de software libre basada en la plataforma de interfaces gráficas WxWidgets, lo que permite correr libremente en diversos sistemas operativos.

Code::Blocks detecta los compiladores instalados en el sistema, facilitando elegir el más adecuado para compilar. Facilita el uso de una serie de librerías populares para desarrollar aplicaciones con interfaz gráfica para usuarios (GUI), como GTK+¹⁴, QT4, FLTK y wxWidgets, siendo estas las librerías más usadas en los sistemas libres.

¹² La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente su acrónimo del inglés GNU GPL, es una licencia creada por la Free Software Foundation, orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

¹³ Conjunto de instrucciones que permiten automatizar una tarea.

¹⁴ Es un conjunto de bibliotecas escritas en C para crear interfaces gráficas de usuario. Tiene una arquitectura orientada a objetos desarrollada en tres bibliotecas:

- GLib: Brinda varias herramientas para usar cuando se programa con GTK+.

Características generales: (4)

- Es open source¹⁵, con licencia GPLv3.
- Es multiplataforma, funciona en los sistemas operativos: GNU/Linux (Ubuntu y Debian), Windows 2000 / XP / Vista y Mac OS X 10.4+.
- Es extensible mediante plugins.
- Importa proyectos desarrollados en Dev-C++ y MSVC.
- Consta de espacios de trabajo para combinar múltiples proyectos.
- Facilita el completamiento de código.
- Vista de proyectos en forma de árbol.
- Desensamblado de código.
- Volcados de memoria.
- Múltiples hilos (threads).

Características de compilación: (5)

- Compila directamente o con archivos compuestos por un conjunto de dependencias y reglas.
- Proyectos con múltiples objetivos.
- Estadísticas y resumen de código.
- Soporte para compilación en paralelo.
- Dependencias entre proyectos dentro del espacio de trabajo.

1.6.2. Lenguajes de programación utilizados.

Lenguaje C

Una de las peculiaridades de C es su riqueza de operadores. Puede decirse que prácticamente dispone de un operador para cada una de las posibles operaciones en

-
- GDK: Abstrae las funciones de dibujo de ventanas de bajo nivel
 - GTK: Brinda un conjunto de widgets para usar cuando creas tus GUI.

¹⁵ Código abierto (open source en inglés) es un término que empezó a utilizarse en 1998 por algunos usuarios de la comunidad del software libre, usándolo como reemplazo al ambiguo nombre original, en inglés, del software libre (free software). El software de código abierto (OSS por sus siglas en inglés) es software para el que su código fuente está disponible públicamente.

código máquina. C es un lenguaje de programación de propósito general. Sus principales características son: (6)

- Programación estructurada.
- Economía de las expresiones.
- Abundancia en operadores y tipos de datos.
- Codificación en alto y bajo nivel simultáneamente.
- Reemplaza ventajosamente la programación en ensamblador.
- Utilización natural de las funciones primitivas del sistema.
- Producción de código objeto altamente optimizado.
- Punteros a funciones y variables estáticas, que permiten una forma rudimentaria de encapsulado y polimorfismo.
- Los argumentos de las funciones se transfieren por su valor.

Lenguaje C++

C++ es un lenguaje diseñado a mediados de los años 1980, por Bjarne Stroustrup. Es un súper conjunto de C que tiene como característica principal, el soporte para Programación Orientada a Objetos y de plantillas o programación genérica. Se creó para añadirle cualidades y características de las que carecía C. Los paradigmas que abarca C++ son:

- Programación estructurada.
- Programación genérica.
- Programación orientada a objetos.

Este lenguaje mantiene la potencia para programar a bajo nivel y se la han añadido elementos que le permiten también un estilo de programación con alto nivel de abstracción. (9)

Puesto que la información recopilada por la herramienta de control es enviada en un documento de manera organizada, se necesita integrar el sistema de supervisión a la generación de documentos a través del lenguaje de marcas extensible (XML).

Lenguaje PHP

Es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web, embebido en páginas HTML y ejecutado en el servidor; se incluye entre etiquetas especiales de comienzo y final que permiten entrar y salir del modo PHP. Este lenguaje soporta el trabajo con CGI, permite procesar la información de formularios y generar páginas con contenidos dinámicos.

PHP es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles, funciones. El resultado es normalmente una página HTML. Permite la posibilidad de ejecutarlo en diferentes tipos de servidores, entre ellos Apache.

La característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir una interfaz vía web para una base de datos es una tarea simple con PHP, permite además incorporar un gran número de extensiones, aumentando las potencialidades del lenguaje.

Lenguaje XML

XML son las siglas de Extensible Markup Language, es un metalenguaje que permite a los diseñadores crear sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones. Es una tecnología muy útil y valiosa para la representación y el intercambio de datos. XML es una versión de SGML (Structured Generalized Markup Language) y un conjunto de reglas que sirven para definir etiquetas semánticas y organizar un documento. Permite manejar datos complejos y proporcionar diferentes vistas. (10)

La estructura de un XML se compone por un conjunto ilimitado de etiquetas, cada una de estas debe abrirse y cerrarse respectivamente luego del contenido adicionado entre ambas. Este metalenguaje a diferencia del Lenguaje de Marcas de Hipertexto (HTML), separa el contenido de la presentación.

Los fundamentos del XML son muy sencillos y en principio, las únicas herramientas que se necesitan son:

- Un editor para escribir los documentos XML, por ejemplo el notepad en el Sistema Operativo Windows y el emacs en GNU/Linux.
- Un procesador o parser XML. Es además un estándar internacional que presenta una estructura jerárquica y su análisis sintáctico es fácil debido a las estrictas reglas que rigen la composición del documento.
- Es de vital importancia la aplicación de esta tecnología en el desarrollo de la implementación, ya que permite organizar la información que es enviada al servidor central.

Lenguaje Shell Script

Shell Script es un lenguaje escrito para el shell o intérprete de línea de comandos de un SO. Este posee gran importancia en la manipulación de archivos, ejecución de programas e impresión de textos.

Una característica clave de Shell Script es que la invocación de sus intérpretes son manejados como un núcleo de operaciones del sistema, logrando así una mayor facilidad de trabajo. Este lenguaje tiene gran utilidad en las primeras etapas del desarrollo de software, y suele ser un objeto de posterior conversión a una implementación subyacente diferente, con mayor frecuencia se convierten en Perl, Python o C.

El intérprete suele permitir el ocultamiento de las directivas dentro de secuencias de comandos, permitiendo la implementación sin fisuras en diferentes idiomas, sin impacto en los usuarios finales.

Este lenguaje posee gran cantidad de programas que sirven como intérpretes lográndose así una mayor facilidad a la hora de escribir el código de depuración en una secuencia de comandos, permitiendo que se vuelva a ejecutar la detección y corrección de errores.

Lenguaje JavaScript

Se trata de un lenguaje de programación del lado del cliente, utilizado para crear programas encargados de realizar acciones dentro del ámbito de una página web. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Con Javascript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso con que cuenta este lenguaje es el propio navegador.

Javascript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones y estructuras de datos complejas. Además, Javascript pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente; el programador se convierte en el verdadero dueño y controlador de cada evento que ocurre en la página cuando la está visualizando el cliente.

1.6.3. Socket

La comunicación entre la aplicación de supervisión y el servidor se realiza utilizando socket. Este es un método para la comunicación entre dos programas que corren sobre ordenadores distintos o en el mismo ordenador. Este se define como el punto final en una conexión. Se crean y se utilizan con un sistema de peticiones o de llamadas de función. Es usado para intercambiar flujos de datos. Un socket queda definido por una dirección, un protocolo y un número de puerto. (11)

Los sockets permiten implementar una arquitectura Cliente/Servidor. Uno de los programas debe estar ejecutado y en espera de que otro quiera conectarse a él, el primero es el cliente y el segundo el servidor.

Existen básicamente dos tipos de canales de comunicación o sockets, los orientados a conexión y los no orientados a conexión.

En el primer caso ambos programas deben conectarse entre ellos con un socket y hasta que no esté establecida correctamente la conexión, ninguno de los dos puede transmitir datos. Esta es la parte del Protocolo de Control de Transmisión (TCP) del Protocolo de Control de Transmisión/Protocolo de Internet (TCP/IP¹⁶), y garantiza que todos los datos van a llegar de un programa a otro correctamente. Se utiliza cuando la información a transmitir es importante, no se puede perder ningún dato y no importa que los programas se queden bloqueados esperando o transmitiendo datos. Si uno de los programas está atareado en otra cosa y no atiende la comunicación, el otro quedará bloqueado hasta que el primero lea o escriba los datos.

En el segundo caso, no es necesario que los programas se conecten. Cualquiera de ellos puede transmitir datos en cualquier momento, independientemente de que el otro programa esté escuchando o no. Es el llamado Protocolo de Datagrama de Usuario (UDP), garantiza que los datos que lleguen sean correctos, pero no que todos lleguen a su destino. Se utiliza cuando es muy importante que el programa no se quede bloqueado y no importa que se pierdan datos. Por esta razón se utiliza socket orientado a conexión.

¹⁶ Son las siglas de Protocolo de Control de Transmisión/Protocolo de Internet (en inglés *Transmission Control Protocol/Internet Protocol*), un sistema de protocolos que hacen posibles servicios Telnet, FTP, correo electrónico, y otros entre ordenadores que no pertenecen a la misma red.

1.6.4. Bash

Bash es un intérprete de comandos (shell) de tipo Unix escrito para el proyecto GNU. Su nombre es un acrónimo de Bourne-Again Shell. Es el shell por defecto en la mayoría de las distribuciones de GNU/Linux en la actualidad. Es el encargado de leer los caracteres tecleados por los usuarios, los interpreta y los ejecuta.

Para el desarrollo de este trabajo, se necesita conocer el valor de algunas variables del sistema que posee el intérprete de comandos, por ejemplo, el usuario que está trabajando en el sistema, además Bash facilita ejecutar comandos para crear y borrar copias de seguridad generada por la aplicación cliente, esto ocurre cuando el cliente no puede enviar la información al servidor porque el ordenador es reiniciado o apagado inesperadamente.

1.6.5. Joomla

Es uno de los mejores CMS de Código Abierto. Es usado por mucha gente para crear desde sitios simples hasta complejas aplicaciones corporativas. Es un sistema de administración de contenidos, y entre sus principales virtudes es que es fácil de instalar, simple de manejar y confiable, además que permite editar el contenido de un sitio web de manera sencilla. Es una aplicación de código abierto construida mayoritariamente en PHP bajo una licencia GPL. Este administrador de contenidos puede trabajar en Internet o intranets y requiere de una base de datos MySQL, así como, preferiblemente, de un servidor HTTP¹⁷ Apache.

Características:

- Mejorar el rendimiento web.
- Versiones imprimibles de páginas.
- Flash con noticias.
- Blogs.
- Foros.
- Encuestas.

¹⁷ (Siglas inglesas de *Hypertext Transfer Protocol*). Es el conjunto de reglas para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la World Wide Web.

- Calendarios.
- Búsqueda en el sitio web e internacionalización del lenguaje.
- Gráficos.

Su nombre es una pronunciación fonética para anglófonos de la palabra swahili *jumla*, que significa "todos juntos" o "como un todo".

1.6.6. Apache

Apache es un servidor de páginas web, un programa que permite acceder a páginas web alojadas en un ordenador. Es el más utilizado seguido de Microsoft Information Services. El servidor Apache es usado por múltiples razones como disponibilidad, facilidad de instalación, pocos recursos de hardware necesario, precio, disponibilidad del código fuente, entre otras. Es una herramienta multiplataforma, gratuita y libre.

Apache es altamente configurable y de diseño modular. Es muy sencillo ampliar sus capacidades. Posee muchos módulos y están disponibles para toda la comunidad. Soporta gran cantidad de lenguajes como Perl, PHP, Java y otros. Nos permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a la medida del administrador, de este modo podemos tener un mayor control sobre lo que sucede en el servidor.

1.6.7. MySQL

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la licencia GNU/GPL. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación, es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. Este aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multihilo. Soporta gran cantidad de tipos de datos para las columnas. Posee gran portabilidad entre sistemas, así como una alta capacidad de gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

1.7. Metodología ágil a utilizar

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayuda a la documentación para el desarrollo de programas. Estas van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando que personas deben participar en el desarrollo de las actividades, así como el papel que deben de tener. Detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

En la actualidad es imprescindible para obtener alta calidad en los resultados de un programa. Las metodologías clásicas tradicionales, no facilitan el desarrollo rápido de aplicaciones, pues requiere de una excesiva cantidad de documentación, además no se adaptan a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno donde los requisitos pueden variar. Las metodologías no ágiles, son aplicables cuando el equipo de desarrollo es numeroso, no así en las ágiles, que son usadas en equipos pequeños (se recomienda para menos de diez personas).

XP

La metodología XP empieza con cuatro valores: Comunicación, Retroalimentación, Simplicidad y Coraje. En esta plataforma XP construye un proceso de diseño evolutivo que se basa en reescribir un sistema simple en cada iteración. Todo el diseño se centra en la iteración actual y no se hace nada anticipadamente para necesidades futuras. El resultado es un proceso de diseño disciplinado, lo que es más, combina la disciplina con la adaptabilidad de una manera que indiscutiblemente la hace la más desarrollada de entre todas las metodologías adaptables.

XP más bien es una metodología encaminada para el desarrollo; consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Consta de 4 facetas principales, Planificación-Definición donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto; Desarrollo, es donde se realiza la implementación del sistema hasta que esté listo para ser entregado;

Entrega, puesta en marcha; y por último, Mantenimiento, donde se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso.

Las entregas son frecuentes, y existe una refactorización continua, lo que nos permite mejorar el diseño cada vez que se le añade una nueva funcionalidad.

Scrum

Define un marco para la gestión de proyectos que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints¹⁸, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para lograr coordinación e integración.

Scrum define métodos de gestión y control para complementar la aplicación de otros métodos ágiles como XP que, centrados en prácticas de tipo técnico, carecen de ellas. Esta metodología permite gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos.

En este capítulo se muestra de forma general los aspectos a tener en cuenta para el desarrollo de una herramienta de supervisión y notificación de eventos. Se describe el monitoreo remoto y a través de agentes locales, y se toma este último por las ventajas que posee y la manera en que se desea controlar las aplicaciones. Se realiza además un estudio del sistema de ventanas X Window System, y la forma de obtener resultados a partir de este. Para desarrollar la aplicación se analizan diferentes Entornos de Desarrollo Integrado (IDE) que participan en la implementación que se realiza y se toman, Anjuta

¹⁸ Se denomina “sprint” a cada iteración de desarrollo de Scrum recomendando realizarla en 30 días.

para la realización de script y Code::Block para la implementación de la herramienta propuesta, además de utilizar el CMS Joomla para el desarrollo de la aplicación web en el servidor.

2. Capítulo 2. Descripción y análisis de la solución propuesta

En este capítulo se hace un análisis de la solución propuesta. Se especifica y justifica la utilización de herramientas ya existentes que formarán parte de la solución del problema a resolver, así como la estrategia de integración al software.

2.1. Propuesta del sistema a implementar

Luego de un estudio de las herramientas existentes para el control de producción, se pudo comprobar que las herramientas actuales no representan una solución integral a la problemática existente. Las aplicaciones no permiten conocer las características de las ventanas construidas por el servidor a petición de un programa cliente.

Luego de analizada estas herramientas se decide comenzar el desarrollo de una herramienta para el control y evaluación del tiempo de producción. La aplicación funciona la mayor parte del tiempo en modo invisible, supervisando las ventanas enfocadas que el usuario utilice durante su estancia en el ordenador. Estos reportes son elaborados por la aplicación cliente, utilizando la tecnología XML y al producirse el cierre de sesión son enviados a una aplicación localizada en un servidor central.

Los reportes contienen el nombre del usuario al que pertenecen. La fecha en la que fue emitido y el número IP de la computadora donde se confeccionó. De las aplicaciones que registra se recoge la cantidad de rutas a las que accedió, su nombre y el tiempo total de ejecución (sumatoria del tiempo de todas las rutas accedidas por ella).

La herramienta cuenta con archivos de configuración que permiten definir las aplicaciones que se desean monitorear. Constantemente realiza una salva de lo que controla, lo que facilita que en caso de producirse un cierre forzoso del sistema, no se pierda la información.

El envío del documento XML hacia el servidor se realiza utilizando socket orientado a conexión, para asegurar que los datos que viajan desde un ordenador a otro lleguen completos, para esto el programa se queda bloqueado hasta que se realiza la operación.

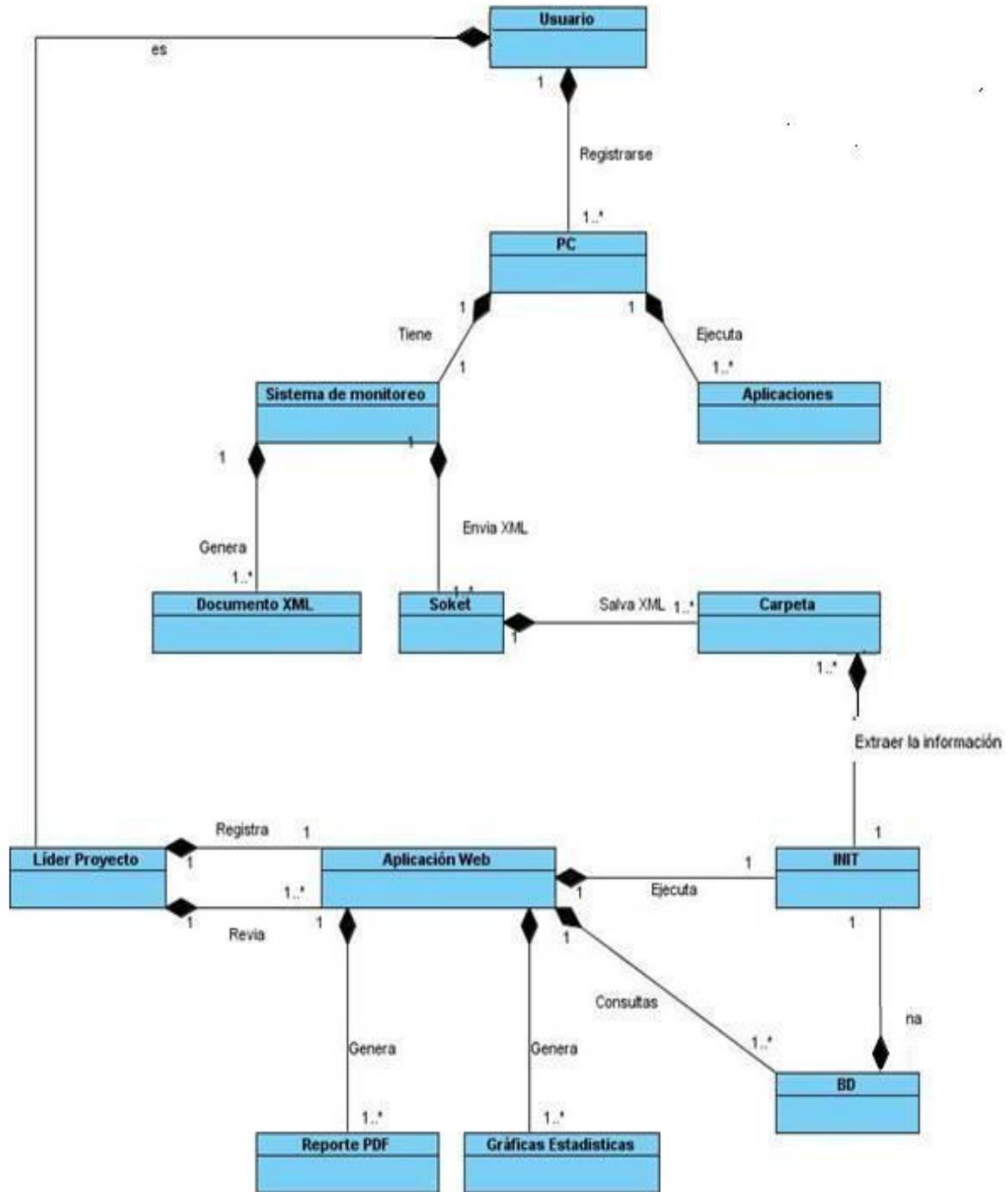
2.2. Modelo de Dominio

El modelo de dominio muestra las clases conceptuales significativas en el dominio del problema, captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará la herramienta. Es considerado un subconjunto del llamado modelo de objetos del negocio.

Durante el desarrollo de la herramienta no se pudo contactar procesos bien definidos en el entorno del negocio. Se hizo difícil determinar los elementos más importantes del sistema y sus interconexiones, así como el establecimiento de las reglas de funcionamiento. Sin embargo, se pueden identificar personas, eventos, transacciones y objetos involucrados en ese entorno que no están bien delimitados, por lo que se hizo necesario un modelado del dominio perteneciente a la solución.

Este modelo ayuda a los usuarios, a utilizar un vocabulario común para poder entender el contexto en que se ubica el sistema y a su vez para capturar correctamente los requisitos y poder construir un sistema correcto, necesitando para ello contar con un firme conocimiento del funcionamiento del objeto de estudio. Este modelo va a contribuir después de identificar algunas clases que se utilizarán en el sistema.

2.2.1 Diagrama de Modelo de Dominio



2.2.2. Descripción de los términos comunes

Se denomina **Usuario** a cualquier persona que trabaje con la aplicación web, sin tener en cuenta la categoría.

Se denomina **PC** al ordenador designado a tener un sistema de monitoreo instalado.

Se denomina **Ventana** al área visual, que contiene algún tipo de interfaz de usuario, mostrando la salida y permitiendo la entrada de datos para uno de varios procesos que se ejecutan simultáneamente.

Se denomina **Sistema de monitoreo** a la aplicación encargada de controlar los procesos y de generar un script en formato de XML.

Se denomina **Documento XML** al documento generado por el sistema de monitoreo.

Se denomina **Socket** a la herramienta encargada de estar a la escucha del envío de los XML.

Se denomina **INIT**, a la herramienta encargada de extraer la información de los XML guardados por el Socket y llenar la base de datos.

Se denomina **BD** al Sistema de gestión de base de datos.

Se denomina **Aplicación Web** a la encargada de hacerle consultas a la BD y generar las gráficas estadísticas y el reporte en formato PDF.

Se denomina **Servidor** al Ordenador encargado de tener el servidor de base de datos (BD): para almacenar los XML, el servidor web, el Socket y el Con_Control.

Se denomina **Líder de Proyecto** a un usuario con ese rol, capaz de autenticarse y de visualizar la información, tanto en gráficas estadísticas como en un reporte en formato PDF.

Se denomina **Reporte PDF** a un reporte generado por la aplicación en formato PDF.

Se denomina **Gráficas Estadísticas** a las gráficas generadas por la aplicación.

2.3. Reutilización de código

La reutilización permite un aprovechamiento más eficiente del código ya construido. El software reutilizable reduce los costes de diseño, codificación y comprobación, porque se amortiza el esfuerzo a lo largo de varios años. Al reducirse la cantidad de código se simplifica también la comprensión, lo cual hace que aumente la probabilidad de que el código sea correcto.

En la implementación de la aplicación se utiliza esta técnica. Se aplican componentes ya existentes para establecer la comunicación entre el cliente y el servidor, el código reutilizable posee funciones de apertura de conexiones, escritura y lectura de datos, y cierre de comunicación.

También se reutiliza código en la aplicación de la tecnología XML para generar la información de forma organizada, se usa un parser¹⁹ implementado, el cual es un módulo encargado de analizar la estructura gramatical del documento.

En la transmisión de datos entre los procesos de monitoreo y la generación de documentos XML, se emplea componentes reutilizables, logrando una eficiente transportación de la información. Con el aprovechamiento de este recurso se logra ahorrar tiempo y esfuerzo en el proceso de desarrollo del software. Es además, una excelente técnica para implementar sistemas que funcionen correctamente.

2.4. Por qué utilizar el sistema X Window System

X Window Systems es un sistema de ventanas orientado a red (network-based window system), independiente del hardware y del SO, y considerado ampliamente como el estándar en este contexto. El éxito de X viene dado en que:

- Es un entorno abierto, ya que el propietario de los derechos es MIT, que lo distribuye libremente. Las fuentes son gratuitas.
- Es un entorno gráfico de ventanas que nace en un momento en el que el mercado comenzó a demandar interfaces gráficas de usuario (GUI).
- Se ha beneficiado también del éxito de Unix como SO, aunque X no se diseñó específicamente para este.
- Es transportable, debido a que existen versiones para casi cualquier plataforma.

¹⁹ Analizador sintáctico que convierte el texto de entrada en otras estructuras (comúnmente árboles).

- X, a nivel de ejecución y debido a su naturaleza de protocolo de red, puede ejecutarse en una máquina y sacar resultados en otra.
- Varios programas pueden utilizar los mismos recursos (ratón, teclado, pantalla).

Todas estas características convierten a X Window en una herramienta poderosa que justifica su utilización para manipular gráficos en SO. Este manipulador de ventanas puede brindar datos importantes sobre lo que sucede en el ordenador, como las dimensiones de una ventana, conocer la aplicación a la que pertenece, saber el estado de visibilidad (totalmente visible, parcialmente visible o totalmente invisible), conocer la ventana padre y las hijas. También permite realizar cálculos analíticos partiendo de la obtención de la resolución de pantalla, etc. A diferencia de las diferentes herramientas de monitoreo estudiadas, con la utilización del sistema X Window se logra obtener una información más precisa sobre lo que el usuario está realizando en la máquina.

2.5. Xlib y XCB

El nivel más bajo de interfaces con X en lenguaje C es Xlib, que permite el acceso a todas las posibilidades del protocolo X. Este ha sido utilizado para el desarrollo de diversas aplicaciones. Se caracteriza por su potencial debido a que estas librerías permiten escribir aplicaciones efectivas sin herramientas de programación adicionales que permitan el uso de ciertas tareas de alto nivel.

A pesar del éxito de Xlib, surgió XCB con el objetivo de sustituir a Xlib. XCB (XC Binding) es una interfaz de programación de aplicaciones de bajo nivel para el servidor X Window. Este proyecto fue iniciado por Bart Massey. Los desarrolladores de la nueva alternativa reconocen a Xlib como un excelente trabajo, pero explican que hay aplicaciones para la que no es ideal.

A pesar de que en la actualidad ha sido utilizado XCB en aplicaciones como Awesome (gestor de ventanas), su extensibilidad aún no abarca el alcance de Xlib y la documentación en la web es mucho menor que la que posee Xlib. Sin embargo, muchas funciones se asemejan a sus homólogas en Xlib, tanto en su nombre, como en los datos que requieren, facilitando la comprensión de la estructura de sus funciones. Por esta razón, se decide utilizar en la primera versión de la solución propuesta, Xlib.

2.6. Estrategia de integración

Para la implementación de un software, es importante tener en cuenta la arquitectura que se va a emplear. En el libro “Ingeniería de Software, un enfoque práctico”, de Roger S. Pressman, quien es una autoridad internacionalmente reconocida en la mejora del proceso del software y en las tecnologías de la Ingeniería del Software, cita tres razones clave por las que la arquitectura de software es importante:

- Las representaciones de la arquitectura de software facilitan la comunicación entre todas las partes interesadas en el desarrollo de un sistema basado en computadora.
- La arquitectura destaca decisiones tempranas de diseño que tendrán un profundo impacto en todo el trabajo de ingeniería del software que sigue, y es tan importante en el éxito final del sistema como una entidad operacional.
- La arquitectura constituye un modelo relativamente pequeño e intelectualmente comprensible de cómo está estructurado el sistema y de cómo trabajan juntos sus componentes.

Basado en estos elementos se realiza el análisis de la arquitectura de la aplicación. En el desempeño de este trabajo, inicialmente se descartó la necesidad de utilizar un modelo Cliente/Servidor, debido a que la información recopilada necesita estar centralizada. Para escoger este esquema, se hizo un estudio sobre las diferentes arquitecturas aplicables para la transportación de ficheros entre distintos ordenadores.

Para el desarrollo de la aplicación cliente, se define la utilización de diversos componentes, con el propósito de facilitar la integración de nuevas funcionalidades de forma independiente. En correspondencia a lo expresado se tomó como arquitectura a utilizar la “Arquitectura centrada en datos”.

En el centro de esta arquitectura se encuentra un almacén de datos (por ejemplo, un documento o una base de datos) al que otros componentes acceden con frecuencia para actualizar, añadir, borrar o bien modificar los datos del almacén. En la solución propuesta el almacén adopta una posición pasiva, esto significa que el software cliente accede a los datos independientemente de cualquier cambio en los mismos, o de las acciones de otro programa sobre la información.

Entre las principales ventajas están:

- Forma eficiente de compartir grandes cantidades de información.

- No hay necesidad de transmitir información entre los distintos componentes del sistema.
- Distintos componentes no necesitan qué datos producen o consumen otros.
- Las actividades de respaldo, seguridad, control de acceso y recuperación de errores están centralizadas.
- El formato de los datos es visible, de modo que es posible desarrollar nuevos componentes con ese formato.

Con la realización de este capítulo, se realiza un análisis sobre la solución propuesta. Se describe de forma general como debe funcionar la aplicación y la importancia de la reutilización de código existente.

Se explica por qué utilizar el sistema de ventanas X Window System, a diferencia de los sistemas de monitoreo estudiados. Para obtener un producto que sea flexible en la incorporación de nuevos componentes se realiza una investigación sobre las diferentes arquitecturas de software, y se decide utilizar una arquitectura centrada en datos.

3. Capítulo 3. Desarrollo ágil de la aplicación propuesta

En el transcurso de este capítulo se describe la lista de reserva del producto y los requisitos no funcionales. Además, se realiza la ingeniería de software a la herramienta, utilizando para ello las metodologías ágiles Scrum y XP. Se explica toda la secuencia del proyecto en forma de Historias de Usuarios, prototipos de interfaces y algunos modelos auxiliares. Se analiza la arquitectura adecuada para desarrollar el sistema y se explica los diversos componentes que la integran.

3.1. Lista de Reserva del Producto

El levantamiento de requisitos es una actividad primordial para el desarrollo de cualquier sistema, en el que se definen las necesidades de los clientes para construir un software. Estos son reflejados en la plantilla Lista de Reserva del Producto (LRP), el cual es el primer artefacto generado, que describe en lenguaje natural las funcionalidades que el sistema debe cumplir para su implementación.

Esta plantilla está conformada por una lista de actividades priorizadas que define el trabajo que se va a realizar. Esta lista puede crecer y modificarse a medida que se obtienen más conocimientos acerca del producto. Con la restricción de que sólo puede cambiarse entre iteraciones. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto.

Teniendo en cuenta lo anteriormente planteado, el sistema deberá ser capaz de:

- Autenticar Usuario.
- Mostrar el reporte general mediante una gráfica.
- Mostrar los reportes de los usuarios mediante gráficas.
- Mostrar los reportes de las aplicaciones utilizadas por los usuarios mediante gráficas.
- Mostrar los reportes por grupos de aplicaciones para un laboratorio determinado mediante gráficas.
- Generar reporte.

3.2. Requisitos No Funcionales

3.2.1. Software

- La aplicación cliente debe funcionar sobre Sistema Operativo GNU/Linux.
- El servidor debe de estar en Ubuntu Server 9.10 o superior.
- En el ordenador del líder de proyecto debe estar instalado el paquete de Adobe Flash Player o alguno que permita la visualización de flash.
- Para la visualización de la web deberá utilizarse Internet Explorer 6.0 o superior y Mozilla Firefox 3.0 o superior.

3.2.2. Usabilidad

- La estructura concebida para la organización de la información agiliza el entendimiento del sistema por parte del usuario.
- Los usuarios serán capaces de alcanzar sus objetivos con un mínimo esfuerzo y obteniendo los resultados máximos.
- El sistema será capaz de solucionar un error cometido por el usuario o sugerir las posibles soluciones, indicándole al usuario las acciones pertinentes a seguir.

3.2.3. Fiabilidad

- En el servidor se garantizará una arquitectura de máxima disponibilidad, tanto del servidor de la aplicación como el de la base de datos. Se garantizarán, además, políticas de respaldo a toda la información, evitando pérdidas en caso de desastres ajenos al sistema.
- Se mantendrá seguridad y control entre usuario, garantizando el acceso de estos sólo al Líder de proyecto o algún designado por este. Las contraseñas podrán cambiarse solo por el propio usuario o por el administrador del sistema.
- El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvast realizadas.

3.2.4. Rendimiento

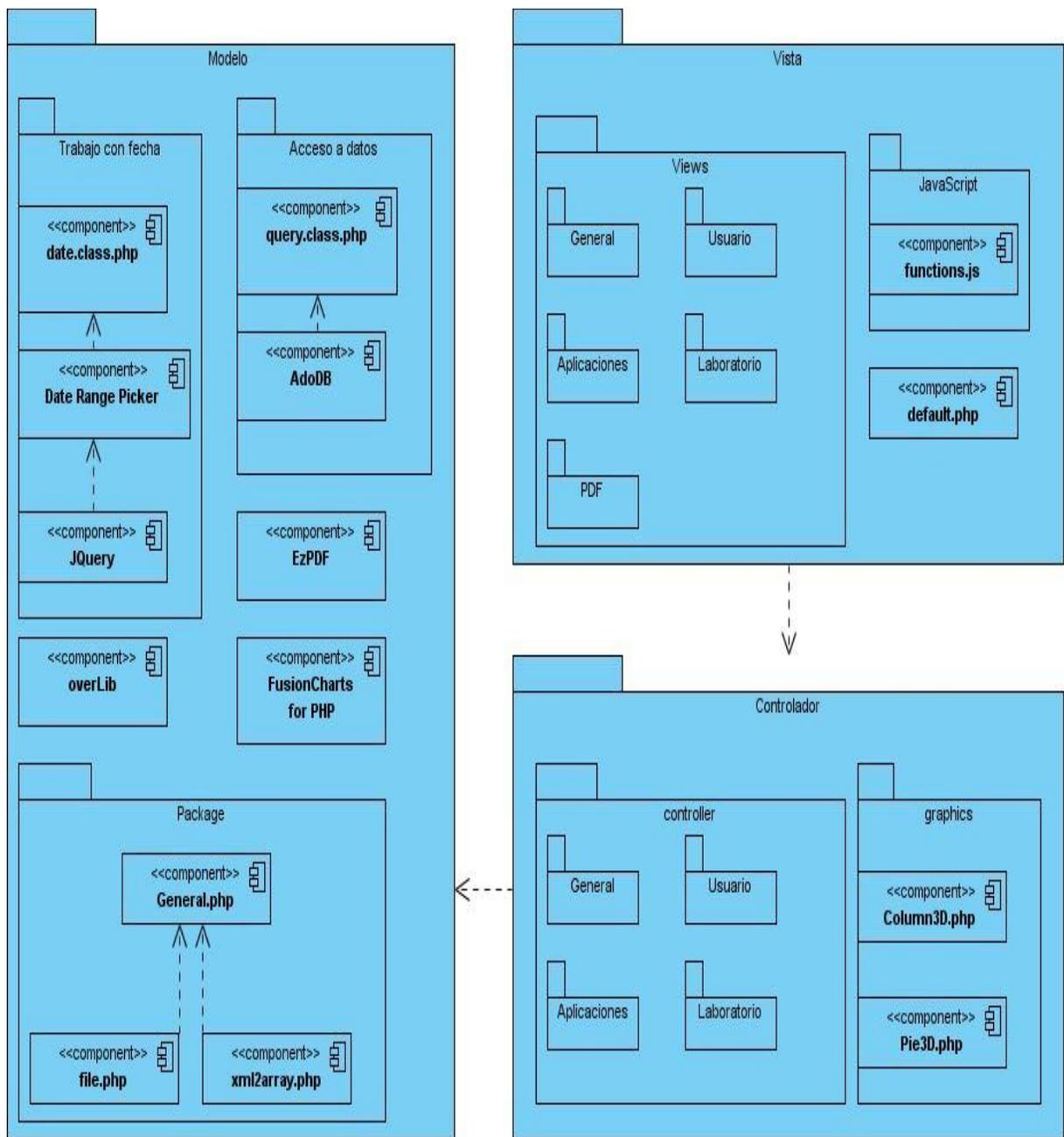
- El sistema respetará buenas prácticas de programación para incrementar el rendimiento en operaciones costosas.

3.3. Descripción de la arquitectura.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como en un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.).

El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. Los componentes en la nueva versión del CMS utilizado se dividen en 3 partes: Model (Modelo), View (Vista) y Controller (Controlador).

En la siguiente figura se muestra el diagrama de componentes que describe la arquitectura del sistema.



Modelo

Producto a la utilización del CMS Joomla para el desarrollo del sistema, todas las clases de acceso a datos y lógica del negocio que se utilizaron son las mismas con las que cuenta el CMS, logrando de esta manera una mayor reutilización del código.

Clases y librerías utilizadas:

- DBQuery: Clase encargada de realizar las conexiones a la base de datos haciendo uso de funcionalidades ya existentes en la librería de acceso a datos adodb.
- jQuery Interactive Date Range Picker: Clase encargada de realizar las operaciones con fechas, siendo el mismo un elegante selector de fechas implementado en jQuery que permite ser incorporado muy fácilmente en una página. No es fácil de modificar pero sí es gratis para ser usado y modificado. Entre las principales funcionalidades que posee están las de obtener la diferencia en días entre dos fechas, conocer si una fecha es mayor que otra y agregar días y/o meses a la fecha actual.
- FusionCharts for PHP: Librería que incluye una serie de clases que sirven para crear imágenes con todo tipo de gráficas dinámicamente desde páginas PHP. Esta ofrece más de 75 tipos de gráficos 2D/3D, los cuales van de gráficos básicos como los de pastel, barras o líneas a gráficos interactivos avanzados, como gráficos 3D, gráficos desplegados, gráficos que se actualizan solos, etc.
- Ezpdf: Librería muy potente que permite crear de manera dinámica documentos PDF con PHP, sin la necesidad de usar algún tipo de módulo. Su uso es intuitivo, solo se necesita incluir la librería a nuestro script y hacer uso de la clase ezPDF para crear nuestro archivo PDF. Esta librería no solo permite crear archivos pdf con texto plano, sino que además nos brinda la posibilidad de darle formato, agregarle color, imágenes y tablas.
- Adodb: ADOdb es un conjunto de librerías de bases de datos con soporte para varios lenguajes. Esta permite a los programadores desarrollar aplicaciones web de una manera portable, rápida y fácil. La ventaja reside en que la base de datos puede cambiar sin necesidad de reescribir cada consulta realizada por la aplicación. Son soportadas las siguientes bases de datos: MySQL, PostgreSQL, Interbase, Firebird, Informix, Oracle, MS SQL, Foxpro, Access, Sybase ASE, FrontBase, DB2, SQLite, Netezza y LDAP.
- Overlib: Librería escrita en JavaScript creada para mejorar nuestras páginas web con pequeñas ventanas "popup" informativas que ayudan a los visitantes y mejoran la usabilidad de nuestras aplicaciones.

Vista

La vista es la responsable de recibir los datos del modelo y mostrarlos al usuario. Cada una de las vistas por lo general tiene asociado un controlador según la funcionalidad que realiza.

En el caso del sistema que se propone las vistas están agrupadas en varios directorios dentro del directorio view del CMS Joomla. Estos son:

- general
- usuarios
- aplicaciones
- laboratorio
- pdf

Todos los directorios correspondientes a las vistas cuentan con un componente en común que se encarga de gestionar todas las interfaces que muestra la vista en cuestión en caso de que posea más de una. Este componente es nombrado *view.html.php*.

Controlador

Los controladores son los ficheros responsables de recibir desde la vista los eventos de entrada y hacer las peticiones al modelo. Los controladores están agrupados en un directorio en la raíz del componente utilizado.

- general
- usuarios
- aplicaciones
- laboratorio

En este directorio se encuentran todos los componentes controladores que se encuentran asociados a las vistas antes mencionadas. Para una mejor asociación entre los componentes controladores y los de las vistas se nombraron igual para lograr una mejor

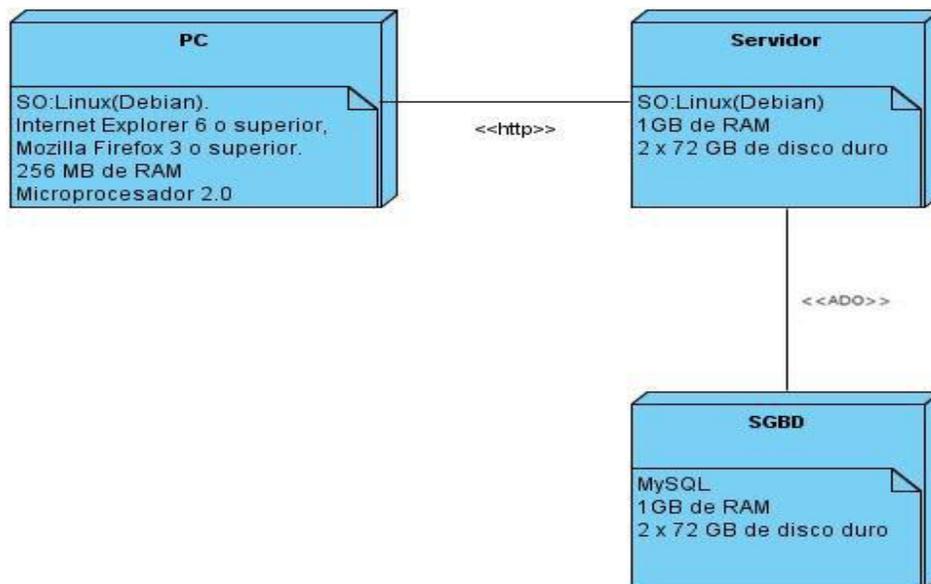
localización. Ejemplo: los controladores asociados al directorio de vistas de usuarios se encuentran en el directorio usuario de los controladores.

Para la construcción de las gráficas se implementaron dos controladores (Column3D.php y Pie3D.php) correspondientes a los dos tipos de gráficas utilizadas por las vistas del sistema, la característica principal de estos es que son parametrizables facilitando una mayor reusabilidad de los mismos.

3.4. Diseño de la base de datos

El CMS Joomla posee características peculiares en cuanto al proceso de instalación de componentes a su núcleo principal. Tanto en módulos desarrollados por la comunidad como en los que se desarrollan por los programadores que mantienen el sistema, las tablas pertenecientes al nuevo módulo se adhieren a la base de datos del sistema, no siempre guardando relación con la misma. En el caso del sistema desarrollado se siguió esta política de desarrollo para la confección de su base de datos.

3.5. Diagrama de despliegue



3.6. Historias de usuarios

Las Historias de usuarios (HU) son utilizadas para la especificación de requerimientos de un sistema. Estas son una forma rápida de administrar las necesidades de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos, debido a que un requerimiento de software es descrito de forma concreta y sencilla utilizando el lenguaje común del usuario. Las HU permiten responder rápidamente a los requerimientos cambiantes y aunque se redactan desde las perspectivas de los clientes, también los desarrolladores pueden brindar ayuda en la identificación de las mismas.

La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Deben poder ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra historia de usuario.

Historia de Usuario	
Número: CETP-01	Nombre Historia de Usuario: Establecer conexión con el servidor X
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reiniel Alvarez Pérez Yosmany Costa Ruiz	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Medio	Puntos Reales: 3
Descripción: Se establece la conexión con el servidor del sistema X Window.	
Observaciones: El sistema X Window trabaja en modo cliente servidor, donde el servidor es el encargado principal de construir cada ventana solicitada por un cliente, para realizar esta operación, el servidor necesita conocer el display que solicita la petición.	
Prototipo de interfaz:	

Historia de Usuario	
Número: CETP-02	Nombre Historia de Usuario: Obtener datos de ventanas
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reiniel Alvarez Pérez Yosmany Costa Ruiz	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Medio	Puntos Reales: 3
Descripción: Se solicita al servidor del sistema X Window los datos de las ventanas que este construye a petición de un cliente (aplicación).	
Observaciones: Los datos a tener en cuenta están relacionados con la ventana contenedora del foco. La ventana puede contener varias subventanas que también serán capturadas, y todas estarán relacionadas con el proceso principal que es el nombre de la aplicación, el cual también es incluido.	
Prototipo de interfaz:	

Historia de Usuario	
Número: CETP-03	Nombre Historia de Usuario: Detectar aplicación no utilizada.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reiniel Alvarez Pérez Yosmany Costa Ruiz	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: Determina si la ventana enfocada está siendo usada o si el usuario ha abierto la misma y no la está utilizando.	
Observaciones:	
Prototipo de interfaz:	

Historia de Usuario	
Número: CETP-04	Nombre Historia de Usuario: Generar documento XML.

Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reiniel Alvarez Pérez Yosmany Costa Ruiz	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: Implementar el componente generador de documentos XML.	
Observaciones: El nombre del documento llevará un formato personalizado evitando que en el mismo día existan documentos con nombres iguales.	
Prototipo de interfaz:	

Historia de Usuario	
Número: CETP-05	Nombre Historia de Usuario: Enviar información entre los componentes.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reiniel Alvarez Pérez Yosmany Costa Ruiz	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: Se integran los componentes del sistema. Luego el componente cliente le envía la información recopilada al componente servidor.	
Observaciones: Los componentes se encuentran en el mismo ordenador.	
Prototipo de interfaz:	

Historia de Usuario	
Número: CETP-06	Nombre Historia de Usuario: Implementar el programa para el componente cliente.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reiniel Alvarez Pérez Yosmany Costa Ruiz	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1

Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: Implementar el programa que envía el fichero XML hasta la computadora donde está el servidor.	
Observaciones:	
Prototipo de interfaz:	

Historia de Usuario	
Número: CETP-07	Nombre Historia de Usuario: Enviar documento XML.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reiniel Alvarez Pérez Yosmany Costa Ruiz	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: El componente cliente le envía la información recopilada al componente servidor. Los componentes se encuentran en computadoras diferentes.	
Observaciones: Se utiliza el protocolo TCP/IP.	
Prototipo de interfaz:	

Historia de Usuario	
Número: CETP-08	Nombre Historia de Usuario: Implementar el programa servidor.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reiniel Alvarez Pérez Yosmany Costa Ruiz	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: Implementar el programa que recibe el fichero XML de la computadora cliente.	
Observaciones:	
Prototipo de interfaz:	

Historia de Usuario	
Número: CETP-09	Nombre Historia de Usuario: Crear script de inicio.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reiniel Alvarez Pérez Yosmany Costa Ruiz	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: Se programa el script que ejecuta el componente que se encuentra en la computadora cliente, este se encarga además de asegurar que el programa esté en ejecución hasta que el usuario abandone la sesión.	
Observaciones: Se implementa en el lenguaje Shell Script.	
Prototipo de interfaz:	

Historia de Usuario	
Número: CETP-10	Nombre Historia de Usuario: Creación del Script principal.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reiniel Alvarez Pérez Yosmany Costa Ruiz	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: Se programa el Script que permitirá realizar todas las operaciones correspondientes a la manipulación de ficheros y directorios. Se encarga de leer los directorios donde se encuentran los reportes en ficheros XML, salvando la información contenida en estos en una base de datos para su posterior análisis. Finalmente, actualiza el estado de los directorios de reportes y crea copias de seguridad de los XML contenidos en las carpetas.	
Observaciones: Se implementa en el lenguaje Java Script.	

Prototipo de interfaz:

Historia de Usuario

Número: CETP-11

Nombre Historia de Usuario: Diseño de la base de datos.

Modificación de Historia de Usuario Número: Ninguna

Usuario: Reiniel Alvarez Pérez
Yosmany Costa Ruiz

Iteración Asignada: 1

Prioridad en Negocio: Alta

Puntos Estimados: 1

Riesgo en Desarrollo: Medio

Puntos Reales: 1

Descripción: Diseño de la base de datos.

Observaciones: La base de datos del sistema no presenta relación con el resto de las tablas del CMS Joomla.

Prototipo de interfaz:

Historia de Usuario

Número: CETP-12

Nombre Historia de Usuario: Autenticación

Modificación de Historia de Usuario Número: Ninguna

Usuario: Reiniel Alvarez Pérez
Yosmany Costa Ruiz

Iteración Asignada: 1

Prioridad en Negocio: Alta

Puntos Estimados: 3

Riesgo en Desarrollo: Medio

Puntos Reales: 3

Descripción: Se le debe dar la posibilidad al Líder del Proyecto o a algún otro responsable designado a autenticarse en el sistema, permitiendo con esto que pueda acceder a las demás opciones del mismo. El sistema debe informar al usuario si dicha actividad ha sido realizada satisfactoriamente, una vez terminado el proceso de autenticación.

Observaciones:

Prototipo de interfaz:

Historia de Usuario	
Número: CETP-13	Nombre Historia de Usuario: Reporte General
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reiniel Alvarez Pérez Yosmany Costa Ruiz	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Medio	Puntos Reales: 3
Descripción: El sistema deberá permitir al Líder del Proyecto o a algún otro responsable designado a visualizar la gráfica correspondiente a los reportes generales por cada uno de los grupos de aplicaciones, teniendo en cuenta una fecha inicio y una fecha fin.	
Observaciones: La gráfica será de tipo pastel, la cual tendrá las opciones de rotar la gráfica, observarla en dos o tres dimensiones, etc.	
Prototipo de interfaz:	

Historia de Usuario	
Número: CETP-14	Nombre Historia de Usuario: Reporte de Usuario
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reiniel Alvarez Pérez Yosmany Costa Ruiz	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Medio	Puntos Reales: 3
Descripción: El sistema deberá permitir al Líder del Proyecto o a algún otro responsable designado a visualizar la gráfica correspondiente al reporte de un usuario en específico del sistema, teniendo en cuenta como parámetro de búsqueda: una fecha inicio, una fecha fin y el usuario al cual se le hará el reporte.	
Observaciones: La gráfica será de tipo barra, en la cual se observará el grupo de aplicaciones contra por ciento de uso del usuario seleccionado.	
Prototipo de interfaz:	

Historia de Usuario	
Número: CETP-15	Nombre Historia de Usuario: Reporte por Aplicación
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reiniel Alvarez Pérez Yosmany Costa Ruiz	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Medio	Puntos Reales: 3
Descripción: El sistema deberá permitir al Líder del Proyecto o a algún otro responsable designado a visualizar la gráfica correspondiente a los reportes de aplicación, teniendo en cuenta como parámetro de búsqueda: una fecha inicio, una fecha fin, un usuario en específico y un grupo de aplicación.	
Observaciones: La gráfica será de tipo barra, en la cual se observará las aplicaciones contra tiempo de uso en segundos del usuario seleccionado.	
Prototipo de interfaz:	

Historia de Usuario	
Número: CETP-16	Nombre Historia de Usuario: Reporte por Laboratorio
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reiniel Alvarez Pérez Yosmany Costa Ruiz	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Medio	Puntos Reales: 3
Descripción: El sistema deberá permitir al Líder del Proyecto o a algún otro responsable designado a visualizar la gráfica correspondiente al reporte de un laboratorio seleccionado, teniendo en cuenta como parámetro de búsqueda: una fecha inicio, una fecha fin y la máscara de red del laboratorio al cual se le generará el reporte.	
Observaciones: La gráfica será de tipo barra, en la cual se observarán los grupos de contra por ciento de uso de las mismas, en el laboratorio seleccionado.	

Prototipo de interfaz:

Historia de Usuario	
Número: CETP-17	Nombre Historia de Usuario: Reporte en PDF
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reiniel Alvarez Pérez Yosmany Costa Ruiz	Iteración Asignada: 4
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Medio	Puntos Reales: 3
Descripción: El sistema deberá permitir al Líder del Proyecto o a algún otro responsable designado a generar reportes en formato PDF con la información obtenida de los reportes anteriores.	
Observaciones: Cada reporte deberá poseer una Descripción bien detallada de la información que muestra.	
Prototipo de interfaz:	

3.7. Tareas de Ingeniería

En la fase de desarrollo se propone la entrega de las tareas realizadas para dar cumplimiento a cada historia de usuario. Estas tareas son llamadas Tareas de Ingeniería.

Tareas de Ingeniería para la historia de usuario CETP-01

Tarea de Ingeniería	
Número Tarea: 1	Número de Historia de Usuario: CETP-01
Nombre Tarea: Estudio de xlib	
Tipo de Tarea: Estudio	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	

Descripción: Se estudia la librería encargada de establecer comunicación con el servidor X.

Observación: Es la librería de más bajo nivel.

Tareas de Ingeniería para la historia de usuario CETP-02

Tarea de Ingeniería	
Número Tarea: 2	Número de Historia de Usuario: CETP-02
Nombre Tarea: Calcular tiempo	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se calcula el tiempo que utiliza el usuario sobre la ventana activa.	
Observación: Si en la ventana del foco no se realiza ninguna acción durante un período específico de tiempo el valor del mismo se iguala a cero.	

Tareas de Ingeniería para la historia de usuario CETP-03

Tarea de Ingeniería	
Número Tarea: 3	Número de Historia de Usuario: CETP-03
Nombre Tarea: Manipular propiedades	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se manipula cualquier cambio en las propiedades de una ventana.	
Observación: Si no existe consumo de recursos en el ordenador, existe alta probabilidad de que no se está trabajando en el mismo.	

Tareas de Ingeniería para la historia de usuario CETP-04

Tarea de Ingeniería

Número Tarea: 4	Número de Historia de Usuario: CETP-04
Nombre Tarea: Diseñar la estructura del XML.	
Tipo de Tarea: Diseño	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se diseña la estructura en que se genera el documento XML.	
Observación: El nombre del documento llevará un formato personalizado evitando que en el mismo día existan documentos con nombres iguales.	

Tarea de Ingeniería	
Número Tarea: 5	Número de Historia de Usuario: CETP-04
Nombre Tarea: Implementar la estructura del XML.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se implementa la estructura en que se genera el documento XML.	
Observación:	

Tareas de Ingeniería para la historia de usuario CETP-05

Tarea de Ingeniería	
Número Tarea: 6	Número de Historia de Usuario: CETP-05
Nombre Tarea: Estudio sobre recursos compartidos.	
Tipo de Tarea: Estudio	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se estudia la manera en que dos programas en ejecución pueden compartir información por medio de recursos compartidos.	
Observación:	

Tarea de Ingeniería	
Número Tarea: 7	Número de Historia de Usuario: CETP-05
Nombre Tarea: Implementación de colas de mensajes en los componentes del sistema.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se implementa funcionalidades de colas de mensajes para establecer la comunicación entre el proceso de monitoreo y el componente generador de XML.	
Observación:	

Tareas de Ingeniería para la historia de usuario CETP-06

Tarea de Ingeniería	
Número Tarea: 8	Número de Historia de Usuario: CETP-06
Nombre Tarea: Estudio de programación con socket.	
Tipo de Tarea: Estudio.	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se estudia la manera en que dos programas que estén en distintos ordenadores puedan establecer comunicación.	
Observación: Utilizar protocolo TCP/IP.	

Tareas de Ingeniería para la historia de usuario CETP-07

Tarea de Ingeniería	
Número Tarea: 9	Número de Historia de Usuario: CETP-07
Nombre Tarea: Cargar y enviar fichero.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez	

Yosmany Costa Ruiz
Descripción: Se lee el fichero, se carga en un buffer y se implementa las funciones de envío.
Observación: El fichero es un documento XML.

Tareas de Ingeniería para la historia de usuario CETP-08

Tarea de Ingeniería	
Número Tarea: 10	Número de Historia de Usuario: CETP-08
Nombre Tarea: Recepción de datos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se reciben los datos enviados por el cliente.	
Observación: La estructura que se espera debe ser la que se envía.	

Tareas de Ingeniería para la historia de usuario CETP-09

Tarea de Ingeniería	
Número Tarea: 11	Número de Historia de Usuario: CETP-09
Nombre Tarea: Estudio del lenguaje Shell Script y del intérprete Bash.	
Tipo de Tarea: Estudio	Puntos Estimados: 1
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se estudia el lenguaje Shell Script y su intérprete Bash para realizar el script de inicio.	
Observación:	

Tarea de Ingeniería	
Número Tarea: 12	Número de Historia de Usuario: CETP-09

Nombre Tarea: Configurar archivos del SO.	
Tipo de Tarea: Configuración.	Puntos Estimados: 1
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se configura el archivo que ejecuta el script al iniciar una sesión.	
Observación:	

Tareas de Ingeniería para la historia de usuario CETP-12

Tarea de Ingeniería	
Número Tarea: 13	Número de Historia de Usuario: CETP-12
Nombre Tarea: Estudiar proceso de autenticación.	
Tipo de Tarea: Estudio	Puntos Estimados: 1
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Realización del mecanismo de autenticación.	
Observación: Se utilizó la autenticación mediante el CMS Joomla.	

Tareas de Ingeniería para la historia de usuario CETP-13

Tarea de Ingeniería	
Número Tarea: 14	Número de Historia de Usuario: CETP-13
Nombre Tarea: Estudiar módulo ArtioFusionCharts	
Tipo de Tarea: Estudio	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Estudio del módulo ArtioFusionCharts para Joomla.	
Observación: Se utilizará el módulo ArtioFusionCharts para el desarrollo de las gráficas de barras y pastel.	

Tarea de Ingeniería	
Número Tarea: 15	Número de Historia de Usuario: CETP-13
Nombre Tarea: Creación de una gráfica de pastel.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Creación de una gráfica de tipo pastel.	
Observación: Se desarrollaron varias gráficas de pastel para lograr que la misma fuese lo más genérica posible.	

Tarea de Ingeniería	
Número Tarea: 16	Número de Historia de Usuario: CETP-13
Nombre Tarea: Selección de datos para la Vista General.	
Tipo de Tarea: Estudio	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se seleccionaron los valores de la base de datos a utilizar para la creación de la Vista General.	
Observación: Se determinó que la Vista General permitirá la visualización de las gráficas correspondientes a los reportes generales por cada uno de los grupos de aplicaciones y para todos los proyectos de la empresa.	

Tarea de Ingeniería	
Número Tarea: 17	Número de Historia de Usuario: CETP-13
Nombre Tarea: Diseño e implementación de la Vista General.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez	

Yosmany Costa Ruiz
Descripción: Se desarrolló el diseño e implementación de la Vista General.
Observación: Se utilizó gráfico de pastel para el desarrollo de la Vista General.

Tareas de Ingeniería para la historia de usuario CETP-14

Tarea de Ingeniería	
Número Tarea: 18	Número de Historia de Usuario: CETP-14
Nombre Tarea: Creación de una gráfica de barras.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Creación de una gráfica de tipo barra.	
Observación: Se desarrollaron varias gráficas de barra para lograr que la misma fuese lo más genérica posible.	

Tarea de Ingeniería	
Número Tarea: 18	Número de Historia de Usuario: CETP-14
Nombre Tarea: Selección de datos para la Vista Usuarios.	
Tipo de Tarea: Estudio.	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se seleccionaron los valores de la base de datos a utilizar para la creación de la Vista Usuarios.	
Observación: Se determinó que la Vista de Usuarios permitirá la visualización de las gráficas correspondientes a los reportes de todos los usuarios pertenecientes a un proyecto determinado.	

Tarea de Ingeniería	
Número Tarea: 20	Número de Historia de Usuario: CETP-14
Nombre Tarea: Diseño e implementación de la Vista de Usuarios	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se desarrolló el diseño e implementación de la Vista de Usuarios.	
Observación: Se utilizó gráfico de pastel para el desarrollo de la Vista de Usuarios.	

Tareas de Ingeniería para la historia de usuario CETP-15

Tarea de Ingeniería	
Número Tarea: 21	Número de Historia de Usuario: CETP-15
Nombre Tarea: Creación de una gráfica de barras.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Creación de una gráfica de tipo barra.	
Observación: Se desarrollaron varias gráficas de barras para lograr que la misma fuese lo más genérica posible.	

Tarea de Ingeniería	
Número Tarea: 22	Número de Historia de Usuario: CETP-15
Nombre Tarea: Selección de datos para la Vista de Aplicaciones.	
Tipo de Tarea: Estudio.	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se seleccionaron los valores de la BD a utilizar para la creación de la Vista	

de Aplicaciones.

Observación: Se determinó que la Vista de Aplicaciones permitirá la visualización de las gráficas, especificando el tipo de aplicaciones que se visualizan.

Tarea de Ingeniería	
Número Tarea: 23	Número de Historia de Usuario: CETP-15
Nombre Tarea: Diseño e implementación de la Vista de Aplicaciones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se desarrolló el diseño e implementación de la Vista de Aplicaciones.	
Observación: Se utilizó gráfico de barras para el desarrollo de la Vista de Aplicaciones.	

Tareas de Ingeniería para la historia de usuario CETP-16

Tarea de Ingeniería	
Número Tarea: 24	Número de Historia de Usuario: CETP-16
Nombre Tarea: Creación de una gráfica de barras.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Creación de una gráfica de tipo barra.	
Observación: Se desarrollaron varias gráficas de barras para lograr que la misma fuese lo más genérica posible.	

Tarea de Ingeniería	
Número Tarea: 25	Número de Historia de Usuario: CETP-16
Nombre Tarea: Selección de datos para la Vista de Laboratorio.	

Tipo de Tarea: Estudio.	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se seleccionaron los valores de la BD a utilizar para la creación de la Vista de Laboratorio.	
Observación: Se determinó que la Vista de Laboratorio permitirá la visualización de las gráficas, especificando el tipo de aplicaciones que se visualizan.	

Tarea de Ingeniería	
Número Tarea: 26	Número de Historia de Usuario: CETP-16
Nombre Tarea: Diseño e implementación de la Vista de Laboratorio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se desarrolló el diseño e implementación de la Vista de Laboratorio.	
Observación: Se utilizó gráfico de barras para el desarrollo de la Vista de Laboratorio.	

Tareas de Ingeniería para la historia de usuario CETP-17

Tarea de Ingeniería	
Número Tarea: 27	Número de Historia de Usuario: CETP-17
Nombre Tarea: Estudiar métodos existentes para la creación de reportes en formato PDF	
Tipo de Tarea: Estudio	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se estudian las formas existentes para la realización de reportes en formato PDF.	
Observación:	

Tarea de Ingeniería	
Número Tarea: 28	Número de Historia de Usuario: CETP-17
Nombre Tarea: Selección de datos para el reporte en formato PDF.	
Tipo de Tarea: Estudio	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se seleccionaron los valores de la BD a utilizar para la creación del reporte en formato PDF.	
Observación: Estos reportes contienen casi la totalidad de la información existente en las vistas General, de Usuarios y de Proyectos.	

Tarea de Ingeniería	
Número Tarea: 29	Número de Historia de Usuario: CETP-17
Nombre Tarea: Diseño e implementación de la vista de reportes en formato PDF.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Programador responsable: Reiniel Alvarez Pérez Yosmany Costa Ruiz	
Descripción: Se desarrolló el diseño e implementación de la Vista para los reportes en formato PDF.	
Observación:	

En este capítulo se ha realizado la descripción de la lista de reserva del producto, los requisitos no funcionales. Además, se realizó la ingeniería de software a la herramienta, utilizando para ello las metodologías ágiles Scrum y XP. También se explicó toda la secuencia del proyecto en forma de Historias de Usuarios, prototipos de interfaces y algunos modelos auxiliares, así como la arquitectura adecuada para desarrollar el sistema y la explicación de los diversos componentes que la integran.

4. Capítulo 4. Validación de la solución propuesta

En este capítulo se describen los casos de pruebas realizados a la aplicación en cada una de las iteraciones, se muestran los resultados, se analizan y se presentan las funcionalidades alcanzadas hasta el período de desarrollo. La plantilla de Plan de pruebas, es lo primero que se realiza cuando el producto va a ser probado. Esta plantilla recoge los pasos y datos de cada uno de los involucrados en el proceso. Además de tener actualizadas las fechas en las que se realiza cada una de las funcionalidades a probar. Con esta plantilla se logra un plan de organización para la realización de las futuras pruebas, se recogen las funcionalidades a probar, y se evalúan de las pruebas realizadas.

4.1. Casos de prueba de aceptación

Uno de los pilares del proceso de desarrollo de un software es el uso de pruebas para comprobar el funcionamiento de los códigos que se implementan. Estas pruebas se realizan entre iteraciones y son las que dan paso a la próxima iteración. Durante el desarrollo de la aplicación propuesta se realizaron un conjunto de pruebas para verificar su correcto funcionamiento. Luego de elaborar el plan de pruebas, para organizar el desarrollo de las mismas, se definieron casos de prueba para historias de usuario, a continuación se hace una Descripción de algunas.

Caso de Prueba de Aceptación	
Código Caso de Prueba: CPA-CETP-01	Nombre Historia de Usuario: Establecer conexión con el servidor X.
Nombre de la persona que realiza la prueba:	
Descripción de la Prueba: Para establecer conexión con el servidor se obtiene el display de conexión y luego se comprueba solicitándole un servicio al servidor.	
Condiciones de Ejecución: Se necesita estar trabajando sobre un entorno gráfico.	
Entrada / Pasos de ejecución: Se tiene que conocer: el nombre del host en el que reside el servidor, el número del display y el número de la pantalla. En esta prueba se define estos valores con NULL, el cual se reemplaza por la cadena almacenada en la variable de entorno DISPLAY.	
Resultado Esperado: Se realice la conexión con el servidor.	

Evaluación de la Prueba: Prueba Satisfactoria

Caso de Prueba de Aceptación	
Código Caso de Prueba: CPA-CETP-02	Nombre Historia de Usuario: Obtener datos de ventanas.
Nombre de la persona que realiza la prueba:	
Descripción de la Prueba: Se calcula el tiempo que utiliza el usuario en cada ventana.	
Condiciones de Ejecución: El usuario abre una ventana, se demora el tiempo que necesita y luego selecciona otra ventana.	
Entrada / Pasos de ejecución: Se toma el tiempo cuando la ventana está en el foco, luego se realiza la misma operación cuando la ventana pierde el foco, y se calcula la diferencia.	
Resultado Esperado: Se obtiene el tiempo que utiliza el usuario en cada ventana.	
Evaluación de la Prueba: Prueba Satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba: CPA-CETP-03	Nombre Historia de Usuario: Generar documento XML.
Nombre de la persona que realiza la prueba:	
Descripción de la Prueba: Se comprueba la existencia de un documento XML.	
Condiciones de Ejecución: Existencia de un documento XML.	
Entrada / Pasos de ejecución: Se entra el nombre del XML, el texto que se va añadir y la etiqueta que recoge el contenido.	
Resultado Esperado: No se crea un nuevo documento XML.	
Evaluación de la Prueba: Prueba Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: CPA-CETP-04	Nombre Historia de Usuario: Generar documento XML.

Nombre de la persona que realiza la prueba:
Descripción de la Prueba: Se genera un documento XML.
Condiciones de Ejecución: No existe un documento XML.
Entrada / Pasos de ejecución: Se entra el nombre del XML, el texto que se va añadir y la etiqueta que recoge el contenido.
Resultado Esperado: Se crea un nuevo documento XML.
Evaluación de la Prueba: Prueba Satisfactoria.

Caso de Prueba de Aceptación	
Código Caso de Prueba: CPA-CETP-05	Nombre Historia de Usuario: Enviar información entre los componentes.
Nombre de la persona que realiza la prueba:	
Descripción de la Prueba: Se establece comunicación entre el sistema que monitorea y el que genera documentos XML, luego el primero le envía mensajes al segundo.	
Condiciones de Ejecución: El proceso que recibe los datos debe esperar la misma estructura de información que el proceso que envía.	
Entrada / Pasos de ejecución: Se ejecuta el proceso que genera documentos XML y este se queda esperando hasta que se ejecute el que envía los datos, luego comienza la transacción de información.	
Resultado Esperado: El componente generador de XML recibe los datos del proceso de monitoreo y se actualiza el documento XML.	
Evaluación de la Prueba: Prueba Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: CPA-CETP-06	Nombre Historia de Usuario: Enviar documento XML.
Nombre de la persona que realiza la prueba:	
Descripción de la Prueba: Se envía el documento XML generado.	
Condiciones de Ejecución: Debe existir al menos un documento XML, así como un servidor esperando, además tanto el cliente como el servidor deben escuchar por el	

mismo puerto.
Entrada / Pasos de ejecución: Se carga el contenido del documento, se calcula el tamaño del mismo y se envía al servidor.
Resultado Esperado: El documento XML es transportado al ordenador donde se encuentra el servidor.
Evaluación de la Prueba: Prueba Satisfactoria.

Caso de Prueba de Aceptación	
Código Caso de Prueba: CPA-CETP-07	Nombre Historia de Usuario: Enviar documento XML.
Nombre de la persona que realiza la prueba:	
Descripción de la Prueba: Se recibe el documento XML generado y se guarda en la carpeta del usuario al que pertenece.	
Condiciones de Ejecución: Debe existir al menos un cliente que envíe al servidor y las carpetas de cada usuario deben estar creadas.	
Entrada / Pasos de ejecución: Se carga el contenido del documento, se calcula el tamaño del mismo y se envía al servidor.	
Resultado Esperado: El documento XML es transportado al ordenador donde se encuentra el servidor.	
Evaluación de la Prueba: Prueba Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: CPA-CETP-08	Nombre Historia de Usuario: Creación del Script principal.
Nombre de la persona que realiza la prueba:	
Descripción de la Prueba: Debe leer y mostrar todos los directorios existentes en una dirección específica.	
Condiciones de Ejecución:	
Entrada / Pasos de ejecución: Se especifica el directorio a leer.	
Resultado Esperado: En caso de existir directorios mostrar los nombres de los mismos,	

en caso contrario mostrar en mensaje informando la no existencia de directorios en la dirección especificada.

Evaluación de la Prueba: Prueba satisfactoria.

Caso de Prueba de Aceptación

Código Caso de Prueba:

CPA-CETP-09

Nombre Historia de Usuario:

Creación del Script principal.

Nombre de la persona que realiza la prueba:

Descripción de la Prueba: Debe eliminar los directorios existentes en una dirección específica y volverlos a crear a partir de un arreglo de nombres. Deberá además mostrar todos los ficheros existentes en un directorio y moverlos a otro directorio específico.

Condiciones de Ejecución: Se especifica el directorio a leer.

Entrada / Pasos de ejecución:

Resultado Esperado: Deben ser eliminados todos los directorios y se deben crear los nuevos partir del arreglo de nombres. Se deberá además mover todos los ficheros existentes de un directorio a otro.

Evaluación de la Prueba: Prueba satisfactoria.

Caso de Prueba de Aceptación

Código Caso de Prueba:

CPA-CETP-09

Nombre Historia de Usuario:

Creación del Script principal.

Nombre de la persona que realiza la prueba:

Descripción de la Prueba: Se leen los reportes y deben ser guardados sus contenidos en la base de datos.

Condiciones de Ejecución: Se debe especificar el directorio donde leer los reportes.

Entrada / Pasos de ejecución: Este script es ejecutado por el sistema según las especificaciones hechas en el crontab del SO. Una vez corrido el script deben guardarse en la BD los datos de los reportes.

Resultado Esperado: Los reportes XML que se encuentran en cada uno de los directorios deben ser leídos, guardando su contenido en la base de datos y después

eliminados de dichos directorios.

Evaluación de la Prueba: Prueba satisfactoria

Caso de Prueba de Aceptación

Código Caso de Prueba:

CPA-CETP-10

Nombre Historia de Usuario:

Reporte General

Nombre de la persona que realiza la prueba:

Descripción de la Prueba: Se selecciona un rango de fecha y se deberá mostrar un resumen en por ciento de cada Grupo de Aplicaciones utilizadas por todos los usuarios contenidos en la base de datos.

Condiciones de Ejecución: Estar autenticado y con permisos suficientes para poder generar reportes.

Entrada / Pasos de ejecución: Se accede al sistema, se selecciona la vista de Reporte General y se selecciona un intervalo de tiempo para posteriormente solicitar el reporte.

Resultado Esperado: Se muestra en una gráfica de tipo pastel los resultados del reporte, en caso contrario informar de que no existen reportes para los filtros señalados.

Evaluación de la Prueba: Prueba satisfactoria

Caso de Prueba de Aceptación

Código Caso de Prueba:

CPA-CETP-11

Nombre Historia de Usuario:

Reporte de Usuario

Nombre de la persona que realiza la prueba:

Descripción de la Prueba: Se selecciona un rango de fecha y un usuario determinado, se deberá mostrar un resumen en por ciento de cada Grupo de Aplicaciones utilizadas por el usuario seleccionado en el tiempo seleccionado.

Condiciones de Ejecución: Estar autenticado y con permisos suficientes para poder generar reportes.

Entrada / Pasos de ejecución: Se accede al sistema, se selecciona la vista de Reporte Usuario, se selecciona un intervalo de tiempo, así como el usuario al cual se le generará

el reporte.
Resultado Esperado: Se muestra en una gráfica de tipo barras, los resultados del reporte, en caso contrario informar de que no existen reportes para los filtros señalados.
Evaluación de la Prueba: Prueba satisfactoria

Caso de Prueba de Aceptación	
Código Caso de Prueba: CPA-CETP-12	Nombre Historia de Usuario: Reporte por Aplicación
Nombre de la persona que realiza la prueba:	
Descripción de la Prueba: Se selecciona un rango de fecha, un usuario determinado y al Grupo al cual se le analizarán las aplicaciones, se deberá mostrar un resumen con el tiempo de uso de las aplicaciones contenidas dentro del grupo, por el usuario y en el tiempo seleccionado.	
Condiciones de Ejecución: Estar autenticado y con permisos suficientes para poder generar reportes.	
Entrada / Pasos de ejecución: Se accede al sistema, se selecciona la vista de Reporte Aplicaciones, se selecciona un intervalo de tiempo, así como el usuario al cual se le generará el reporte.	
Resultado Esperado: Se muestra en una gráfica de tipo barras, los resultados del reporte, en caso contrario informar de que no existen reportes para los filtros señalados.	
Evaluación de la Prueba: Prueba satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba: CPA-CETP-13	Nombre Historia de Usuario: Reporte por Laboratorio
Nombre de la persona que realiza la prueba:	
Descripción de la Prueba: Se selecciona un rango de fecha y la máscara del laboratorio	

al cual se le analizarán los Grupos de Aplicaciones y se deberá mostrar un resumen en por ciento de cada Grupo de Aplicaciones utilizadas por todos los usuarios cuyas direcciones IP coincidan con la máscara seleccionada.
Condiciones de Ejecución: Estar autenticado y con permisos suficientes para poder generar reportes.
Entrada / Pasos de ejecución: Se accede al sistema, se selecciona la vista de Reporte por Laboratorio, se selecciona un intervalo de tiempo, así como la máscara del laboratorio al cual se le generará el reporte.
Resultado Esperado: Se muestra en una gráfica de tipo barras, los resultados del reporte, en caso contrario informar de que no existen reportes para los filtros señalados.
Evaluación de la Prueba: Prueba satisfactoria

Caso de Prueba de Aceptación	
Código Caso de Prueba: CPA-CETP-14	Nombre Historia de Usuario: Reporte en PDF
Nombre de la persona que realiza la prueba:	
Descripción de la Prueba: Seleccionar reportes en formato PDF con información existente en el resto de las vistas.	
Condiciones de Ejecución: Estar autenticado y con permisos suficientes para poder generar reportes.	
Entrada / Pasos de ejecución: Se accede a la vista Reporte PDF y se escoge un reporte de los existentes.	
Resultado Esperado: Se muestra el reporte en formato PDF con la información solicitada.	
Evaluación de la Prueba: Prueba satisfactoria	

En este capítulo se ha mostrado de forma general los aspectos a tener en cuenta para el desarrollo ágil del software, utilizando como metodología de desarrollo a XP y Scrum. Se realizó además la planificación para el desarrollo del sistema, así como el funcionamiento del software a través de la Lista de Reserva de Productos, Historias de Usuarios y prototipos de interfaz de usuario. También se explican las tareas de ingeniería y los estándares de codificación desarrollados en el sistema.

Conclusiones

Con la realización del presente trabajo de diploma, se obtuvo la creación de una herramienta para llevar a cabo de forma eficiente el control del tiempo de producción. El sistema ha sido implementado siguiendo prácticas actuales que permiten un desarrollo robusto y reutilizable.

Las herramientas y tecnologías seleccionadas son software libre, de código abierto y bajo licencias de distribución gratuita en su mayoría, lo cual aporta además a la soberanía tecnológica de la solución. Los lenguajes de programación C, C++, JavaScript, PHP y Shell Script, brindan las herramientas necesarias para el desarrollo de una aplicación como la que se propone. Por su parte las metodologías XP y Scrum contienen las pautas del trabajo a seguir, además de contemplar los artefactos y la documentación que permitirán el proceso de soporte y la creación de nuevas versiones.

Para poder diseñar el sistema se realizó un estudio de los árboles de procesos, el monitoreo de ventanas y cuestiones afines en los Sistemas Operativos Linux, basados en distribuciones Debian.

La solución propuesta fue diseñada en dos partes fundamentales. El cliente que se encarga de recoger la información en los ordenadores personales de los estudiantes y trabajadores de los proyectos. Y el servidor que recoge toda la información de los clientes y realiza reportes y tareas estadísticas. Permite también, al tener los datos centralizados, poder realizar reportes precisos y con un nivel aceptable de actualidad. Quedando como resultado de este trabajo en su versión 1.0, una herramienta para el Control y Evaluación del Tiempo de Producción (CETP).

Recomendaciones

- Adaptar el sistema para ser integrado con otros, como el portal del CESIM con motivo de centralizar las tareas administrativas del centro.
- Implementar mecanismos que permitan a los clientes saber los cambios de dirección IP de los servidores automáticamente.
- Agregarle una funcionalidad para verificar si un estudiante ha asistido a una determinada sesión.
- Ampliar la selección de las máscaras de red, para saber si es de una facultad, la Universidad o un organismo superior.
- Implementar una página de administración para la gestión y configuración de usuarios.

Referencias Bibliográficas

1. Durán, Sergio González. MANUAL BÁSICO DE ADMINISTRACIÓN DE PROCESOS. *LinuxTotal .com.mx*. [En línea] [Citado el: 3 de Diciembre de 2009.] http://www.linuxtotal.com.mx/index.php?cont=info_admon_012.
2. Pérez Costoya, Fernando. Curso del sistema de ventanas X [Citado el: 10 de Noviembre de 2009]. Available from World Wide Web: <http://laurel.datsi.fi.upm.es/~fperez/cursoX/indice.html>
3. The Anjuta Project. Anjuta DevStudio: GNOME Integrated Development Environment. [En línea] 2009. [Citado el: 25 de Febrero de 2010.] <http://anjuta.org/>.
4. The open source, cross platform, free C++ IDE. *Code::Blocks*. [En línea] [Citado el: 24 de Enero de 2010.] <http://www.codeblocks.org/>.
5. Code::Blocks Team. Code::Blocks [Citado el: 6 de Diciembre de 2009] Available from world wide web: <http://www.codeblocks.org>
6. Meyer, Eric. Lenguaje C. [Citado el: 2 de Octubre de 2009]. Available from world wide web: <http://www.ib.cnea.gov.ar/~icom/CursoC/historia.shtml>
8. chuidiang. Programación de sockets en C de Unix/Linux. [Citado el: 10 de febrero de 2010]. Available from world wide web: http://www.chuidiang.com/clinix/sockets/sockets_simp.php
9. *Scribd*. [En línea] Diciembre de 2001. [Citado el: 10 de Febrero de 2010.] <http://www.scribd.com/doc/22283/Manual-De-Programacion-Lenguaje-C>.
10. Montalvo, Marlene Melián. XML el nuevo lenguaje universal. [En línea] [Citado el: 22 de Enero de 2010.] <http://www.bibliociencias.cu/gsd/collect/eventos/index/assoc/HASH0104/f016d031.dir/doc.pdf>.
11. Programación de sockets en C de Unix/Linux. *chuidiang*. [En línea] 4 de Febrero de 2007. [Citado el: 26 de Febrero de 2010.] http://www.chuidiang.com/clinix/sockets/sockets_simp.php.

Bibliografía

Achour, Mehdi. *Manual de PHP.* [Online] junio 04, 2010. [Cited: Junio 05, 2010.] <http://php.net/manual/es/index.php>.

Arrakis. *Guía Beej de programación en redes.* [Online] [Cited: Noviembre 23, 2009.] <http://www.arrakis.es/~dmrq/beej/clientserver.html>.

Barros, Alejandro. Sistemas Colaborativos de gestión de proyectos. [Cited: Enero 13,2010]. <<http://www.alejandrobarrros.cl/content/view/134919/Sistemas-Colaborativos-de-gestion-de-proyectos.html>>.

Campos, Alfredo. *Blog de Alfredo Campos.* [Online] Octubre 01, 2008. [Cited: Noviembre 02, 2009.] <http://alfredocampos.blogspot.com/2008/10/ps-para-monitorear-activamente-los.html>.

Code::Blocks Team. Code::Blocks [Cited: December 06, 2009] Available from world wide web: <http://www.codeblocks.org>

Costoya, Fernando Pérez. Curso del sistema de ventanas X. [Online] [Cited: Enero 13, 2010.] <http://laurel.datsi.fi.upm.es/~fperez/cursoX/indice.html>.

chuidiang. Programación de sockets en C de Unix/Linux. [Cited: 10 de febrero de 2010]. Available from world wide web: http://www.chuidiang.com/clinux/sockets/sockets_simp.php

Curso del sistema de ventanas X. [Online] [Cited: Enero 26, 2010.] <http://laurel.datsi.fi.upm.es/~fperez/cursoX/indice.html>.

Documentation for the X Window System Version 11 Release 6.9 and 7.0 (X11R6.9/X11R7.0). [Online] Diciembre 21, 2005. [Cited: Febrero 9, 2010.] <http://www.x.org/archive/X11R7.0/doc/html/index.html>.

Durán, Sergio González. MANUAL BÁSICO DE ADMINISTRACIÓN DE PROCESOS. LinuxTotal .com.mx. [Online] [Cited el: 3 de Diciembre de 2009.] http://www.linuxtotal.com.mx/index.php?cont=info_admon_012.

El lenguaje JavaScript. [Online] [Cited: Diciembre 10, 2009.] http://www.xpps.net/contenido_xppsnet/areatec/JavaScript.pdf.

El sistema X Window. [Online] [Cited: Octubre 9, 2009.] <http://ditec.um.es/laso/docs/tut-tcpip/3376c49.html>.

Free Software Foundation. Filosofía del Proyecto GNU . *GNU Operating System.* [Online] [Cited: Octubre 7, 2009.] <http://www.gnu.org/philosophy/philosophy.es.html#LicensingFreeSoftware>.

Gayo, Jose Emilio Labra. *Lenguaje XML.* [Online] Marzo 2002. [Cited: Octubre 30, 2009.] <http://petra.euitio.uniovi.es/~labra/cursos/ext02/xml.pdf>.

James Gettys, Keith Packard. The (Re)Architecture of the X Window System. [Online] Julio 2004. [Cited: Octubre 10, 2009.] http://keithp.com/~keithp/talks/xarch_ols2004/.

Maestros del Web. *Apache + PHP + MySql + PhpMyAdmin como módulo de Apache.* [Online] [Cited: Febrero 28, 2010.] <http://www.maestrosdelweb.com/editorial/phpmysqlap/>.

Manrique, Daniel. X Window System Architecture Overview HOWTO. [En línea] Mayo 22, 2001. [Cited: Enero 13, 2010.] <http://www.faqs.org/docs/Linux-HOWTO/XWindow-Overview-HOWTO.html>.

Meyer, Eric. Lenguaje C. [Cited: 02, October 2009]. Available from world wide web: <http://www.ib.cnea.gov.ar/~icom/CursoC/historia.shtml>

Modelo Cliente-Servidor. [Online] [Cited: febrero 27, 2010.] http://www.adelat.org/media/docum/nuke_publico/modelo_clienteservidor.html.

Monitorear procesos en GNU/Linux. *Rincon Informático.* [Online] Enero 19, 2010. [Cited: Febrero 12, 2010.] <http://www.rinconinformatico.net/monitorear-procesos-en-gnulinux>.

Montalvo, Marlene Melián. XML el nuevo lenguaje universal. [Online] [Cited: 22 de Enero de 2010.] <http://www.bibliociencias.cu/gsd/collect/eventos/index/assoc/HASH0104/f016d031.dir/doc.pdf>.

Oracle Corporation. MySQL. *MySQL 5.0 Reference Manual.* [Online] 2010. [Cited: Abril 2, 2010.] <http://dev.mysql.com/doc/refman/5.0/es/index.html>.

Pérez Costoya, Fernando. Curso del sistema de ventanas X [Cited: 10 de Noviembre de 2009]. Available from World Wide Web: <http://laurel.datsi.fi.upm.es/~fperez/cursoX/indice.html>

Programación Básica de Sockets en Unix para Novatos. [Online] [Cited: Enero 30, 2010.] <http://es.tldp.org/Tutoriales/PROG-SOCKETS/prog-sockets.html>.

Programación de sockets en C de Unix/Linux. chuidiang. [Online] 4 de Febrero de 2007. [Cited el: 26 de Febrero de 2010.] http://www.chuidiang.com/clinux/sockets/sockets_simp.php.

Scribd. [Online] Diciembre de 2001. [Cited: 10 de Febrero de 2010.] <http://www.scribd.com/doc/22283/Manual-De-Programacion-Lenguaje-C>.

The Anjuta Project. Anjuta DevStudio: GNOME Integrated Development Environment. [Online] 2009. [Cited el: 25 de Febrero de 2010.] <http://anjuta.org/>.

The open source, cross platform, free C++ IDE. Code::Blocks. [Online] [Cited: 24 de Enero de 2010.] <http://www.codeblocks.org/>.

The X.Org Foundation. Documentation for the X Window System Version 11 Release 6.8.1 (X11R6.8.1). [Online] Septiembre 17, 2004. [Cited: Enero 16, 2010.] <http://www.x.org/archive/X11R6.8.1/doc/>.

Toro, Amador Durán. Programación en X Window. [Online] Junio 1994. [Cited: Octubre 22, 2009.] <http://www.lsi.us.es/cursos/xlib.html>.

The jQuery Project. *jQuery*. [Online] 2010. [Cited: 05 1, 2010.] <http://docs.jquery.com/Types>.

The Open Group Base Specifications Issue 6. [Online] 2004. [Cited: Enero 27, 2010.] <http://www.opengroup.org/onlinepubs/000095399/functions/socket.html>.

Raya, Paco Aldarias. *Lenguaje C.* [Online] Febrero 14, 2004. [Cited: Octubre 28, 2009.] <http://pacoc.iespana.es/c.pdf>.