

**Universidad de las Ciencias Informáticas
Facultad 10**



**Módulo de dotProject para la gestión de estadísticas del
Home Compartido.
GENEST.**

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

Autor: Karel Tito Catá.

Tutor: Ing. Yoandy Pérez Villazón.

Ciudad de La Habana, Cuba, 2009.
“Año del 50 aniversario del triunfo de la Revolución”

Declaración de Autoría

Declaro que soy el único autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Karel Tito Catá

Ing. Yoandy Pérez Villazón

Datos de contacto de los tutores

DATOS GENERALES

Nombre y Apellidos: Yoandy Pérez Villazón

Fecha de nacimiento: 16/04/1984

Carne de Identidad: 84041609944

Dirección: Calle Luz Caballero #125 % Mariana Grajales y Martha Abreu. Colón. Matanzas. Cuba.

Nacionalidad: Cubana.

Teléfono Centro de Trabajo: (+53) (7) 837 3156

Correo electrónico: yvillazon@uci.cu

Fecha de actualización del currículum: Mayo 2009

Institución: Universidad de las Ciencias Informáticas.

RESUMEN

Ingeniero en Ciencias Informáticas de la Universidad de las Ciencias Informáticas (UCI) desde el 30 de mayo del 2008. Una vez graduado se inicia como profesor en la UCI en la disciplina de Física. Actualmente se desempeña como líder del proyecto Mejoras al Sistema dotProject CNTI en la Facultad 10. Ha impartido las asignaturas de Física I y Física II. Referente a la investigación ha desarrollado trabajos de investigación para UCIENCIA e Informática 2009, en este último evento posee dos artículos, uno como autor en la modalidad de póster y otro como coautor en el evento virtual, ambos incluidos en las memorias del evento. Ha ejercido como miembro de diversos tribunales de tesis de grado y a tutorado diversos trabajos a estudiantes de pregrado. En el área productiva a trabajado en diversos proyectos relacionados con la migración a Software Libre.

FORMACIÓN ACADÉMICA

Ingeniería en Ciencias Informáticas, Universidad de las Ciencias Informáticas (UCI), 2008.

CATEGORÍA DOCENTE Y/O CATEGORÍA CIENTÍFICA

Instructor recién graduado / Ingeniero

LÍNEAS DE INVESTIGACIÓN

Software Libre. Herramientas de Gestión de Proyectos.

Las investigaciones más importantes:

- “Metodología de Migración a Software Libre para el MES”, Mayo 2008.
- “Mini-Guía de Migración a Software Libre”, Octubre 2008.
- “Guía Cubana de Migración a Software Libre”, Marzo 2009.

PUBLICACIONES Y MEMORIAS DE EVENTOS

- “Metodología para la Migración a Software Libre del MES”. Yoandy Pérez Villazón. IV Taller de Software Libre. Memorias de la Feria y Convención Internacional Informática 2009. La Habana, Cuba. 2009.
- “Guía Cubana de Migración a Software Libre”. Ramón Paumier Samón, Yoandy Pérez Villazón. IV Taller de Software Libre. Memorias de la Feria y Convención Internacional Informática 2009. La Habana, Cuba. 2009.

Agradecimientos

A mi madre por ser mi mayor motivo de inspiración.

A toda mi familia por la ayuda que siempre me ha brindado.

A todas las personas que confiaron en mi para la realización del presente trabajo en especial a Sheila y Yoany.

A mi tutor Yoandy por su ejemplo como profesional y como persona.

A mis amistades por ayudarme en mi formación personal y profesional.

Dedicatoria

A mi madre por ser mi principal motivo de inspiración.

A mi abuelo Tito por su ejemplo.

A mi familia por su invaluable ayuda.

Resumen

Hoy en día son diversos los procesos que se han automatizado con el auxilio de las tecnologías web. Los procesos de Gestión de Proyectos (GP) no escapan a esta enorme ola de automatización y se apoyan en las Tecnologías Web para ello, principalmente en los Sistemas de Gestión de Proyectos (SGP). El presente trabajo de diploma llevará a cabo el desarrollo de un módulo para una herramienta de este tipo: el dotProject. El módulo es el encargado de analizar estadísticamente un grupo de reportes en formato XML que llegan al servidor donde está instalado el dotProject y que contienen información relacionada con el uso de las aplicaciones por los usuarios del Home Compartido. En las diferentes vistas con que cuenta el módulo se muestran gráficas con el porcentaje del uso de diferentes tipos de aplicaciones que serán analizadas , además del acceso a la información mediante reportes en formato PDF. El módulo está implementado haciendo uso de tecnologías web con el framework que brinda dotProject para su propio desarrollo.

Índice de contenido

Índice de contenido

Introducción	1
Capítulo 1. Fundamentación Teórica	4
Historia y definición de la gestión de proyectos.	4
Las tres restricciones tradicionales.	5
Tendencias de la gestión de proyectos.	7
Los sistemas de Gestión de proyectos.	7
Ejemplos de SGP.	8
OpenProj.	8
ActiveCollab.	9
GanttProject.	9
Microsoft Project.	10
Planner	11
dotProject.	11
Herramientas, lenguajes y tecnologías a utilizar.	12
Lenguajes de programación.	12
Herramientas utilizadas.	14
Metodologías de desarrollo de software.	17
Metodología SXP.	17
PMBOK.	18
Gestión de las Comunicaciones del Proyecto según PMBOK.	19
Infografía.	20
Tipos de infográficos	20
Partes de un infográfico	20
Capítulo 2. Descripción y análisis de la solución propuesta.	22
Amvex	22
Levantamiento de requisitos	24
Propuesta del sistema a implementar.	26
Descripción de la arquitectura.	28
Diseño de la base de datos	33
Guía para la creación de un módulo	33
Creación de un módulo para dotProject en 6 pasos.	34
Capítulo 3. Desarrollo ágil del módulo GENEST.	35
Planificación del proyecto por roles.	36
Historias de usuarios.	36
Plan de versiones	41
Tareas de Ingeniería.	42
Capítulo 4. Validación de la solución propuesta	50

Casos de Pruebas	50
Resultados Obtenidos.....	55
Funcionalidades obtenidas.....	56
Conclusiones	58
Recomendaciones	59
Referencias Bibliográficas	60
Glosario de términos	64
Anexos	68

Introducción

Con el auge de las Tecnologías de la Información y las comunicaciones (TIC) el quehacer diario de las personas se ha revolucionado de forma sustancial, tanto en labores del hogar como en cualquier rama científica.

Hoy en día una de las TIC más utilizadas es la Red de Redes (WWW), la cual no solo nos permite comunicarnos con cualquier parte del mundo con tan solo un click, sino que es la herramienta más recurrida y con mayores potencialidades para el acceso, difusión y generación de información. Uno de sus inminentes usos apunta, entre otras cosas, a la automatización de diferentes procesos que intervienen en la administración de Empresas Productoras de Software. Uno de estos procesos es la gestión de la información asociada a sus procesos de producción por solo citar un ejemplo.

En la actualidad es común el uso de sistemas informáticos que permiten facilitar la gestión de los procesos que se llevan a cabo en las empresas productoras de software, tal es el caso de los SGP. Diversas son las facilidades y ventajas que proporciona el empleo de los SGP, justificando así la gran importancia que han tomado dentro del mundo del desarrollo de software. Su importancia más palpable es la de favorecer en gran medida la toma de decisiones con una mejor eficacia y rapidez. Actualmente existen diversas aplicaciones destinadas a estos fines que constan de muy buena calidad y aceptación a nivel mundial. Entre ellas se encuentran GanttProject (Arleco 2008a)□, OpenProj (Salva Castro 2008)□, ActiveCollab (ActiveCollab 2009)□, Planner (Planner Team 2009)□, MSProject (Arleco 2008b)□ y dotProject (Ernesto Serrano 2009)□ siendo este último uno de los más populares.

El SGP dotProject es una herramienta que permite gestionar los proyectos existentes en una empresa, además de las tareas, eventos y recursos asociados a cada uno de ellos. Generalmente para el líder de proyecto esta labor se torna compleja y prácticamente inabordable sin la ayuda de herramientas como los SGP .

DotProject se perfila como una interesante herramienta para trabajar en entornos colaborativos, permitiendo a los integrantes del equipo trabajar compartiendo información relativa a los proyectos. Contar con dotProject sería definitivamente ventajoso para cualquier empresa productora de software. El mismo facilita el control y gestión de todos los procesos presentes en la creación de un proyecto, permitiendo a los responsables del mismo conocer el estado del proyecto y sus tareas.

El Grupo de Proyecto UNICORNIOS de la Universidad de las Ciencias Informáticas (UCI) tiene alrededor de dos años realizando los procesos de gestión de proyecto con este sistema, teniendo gran aceptación con el mismo; independientemente de algunas particularidades que presenta el proyecto y que no se pueden manejar desde el dotProject. Como parte de los esfuerzos por automatizar la mayor cantidad de los procesos de gestión en el Grupo de Proyecto UNICORNIOS se necesita tener información acerca del uso de las aplicaciones en las computadoras que utilizan el Home Compartido. En estos momentos existe la imposibilidad de gestionar desde el SGP dotProject las estadísticas generadas a partir de la información antes descrita, por la ausencia en el mismo de un módulo con estas características.

Teniendo en cuenta lo descrito anteriormente se plantea como **problema científico**: ¿Cómo desarrollar un módulo para dotproject que permita gestionar información referente al uso de las aplicaciones en las computadoras conectadas al Home Compartido ? Del problema anterior se puede definir que el **objeto de Investigación** de este trabajo de diploma es el proceso de desarrollo de software en aplicaciones web. Acotando el **campo de acción** al Sistema de Gestión de Proyectos dotProject en el proyecto UNICORNIOS.

Se plantea como **objetivo general de la investigación** desarrollar un módulo para el SGP dotProject que permita procesar la información referente al uso de las aplicaciones en las computadoras conectadas al Home Compartido.

Desglosando el mismo en los siguientes **objetivos específicos**:

- Sistematizar sobre los procesos de Gestión de Proyectos y los SGP ,sus características y funcionalidades.
- Analizar y diseñar las funcionalidades del módulo GENEST .
- Implementar las funcionalidades del módulo GENEST.
- Probar las funcionalidades que brindará el módulo GENEST

Para darle cumplimiento al objetivo planteado se plantean las siguientes **tareas**:

- Estudiar los procesos de Gestión de Proyectos y sus características.
- Identificar y estudiar las funcionalidades en los SGP, haciendo énfasis en las relacionadas con la generación de estadísticas.
- Estudiar el proceso de integración al SGP dotProject
- Modelar las funcionalidades que brindará el módulo GENEST.
- Implementar las funcionalidades que brindará el módulo GENEST.
- Probar las funcionalidades que brindará el módulo GENEST.

Para una mejor comprensión de la investigación, el contenido ha sido desglosado en cuatro capítulos, conclusiones generales, recomendaciones, referencias bibliográficas y bibliografía utilizada, además de un glosario de términos en el cual se detallan los términos técnicos y poco claros utilizados en la elaboración del documento además de los anexos que complementan el trabajo realizado.

Los capítulos han sido estructurados de la siguiente manera :

Capítulo I Fundamentación Teórica : En este se definen una serie de conceptos acerca de la Gestión de proyectos y de los SGP incluyendo elementos históricos acerca del surgimiento de los mismos y ejemplos. Se describen las

tecnologías, lenguajes y herramientas a utilizar en la elaboración de la solución.

Capítulo II Análisis y diseño de la solución propuesta : A lo largo de este capítulo se recogen los requerimientos planteados por el cliente, se formula una propuesta de la solución a implementar y se diseña la arquitectura con se implementará el sistema.

Capítulo III Desarrollo ágil del sistema: En este capítulo se realiza la implementación de la solución propuesta en el capítulo anterior y se generan los artefactos correspondientes a la fase de desarrollo según la metodología seleccionada.

Capítulo IV Validación de la solución propuesta: En este capítulo se realizan un conjunto de pruebas necesarias para validar el producto obtenido, además de explicarse en detalle las funcionalidades obtenidas en el mismo.

Capítulo 1. Fundamentación Teórica

Historia y definición de la gestión de proyectos.

Los proyectos existen desde siempre, consisten en un esfuerzo temporal, único y progresivo, que se emprende para crear un producto o un servicio único. Desde las antiguas pirámides de Egipto o los monumentales edificios de la antigua Roma o la antigua Grecia hasta las complejas plataformas tecnológicas de hoy en día, pasando por la construcción de los primeros rascacielos, redes ferroviarias o aeronaves del siglo XX, la innovación siempre los ha acompañado, mejorando y perfeccionando día tras día las diferentes técnicas, teorías y competencias de la administración empresarial, así como el ámbito global de aplicación de todas ellas dentro de las organizaciones.

Numerosas técnicas empleadas actualmente por la mayoría de Directores de Proyectos de todo el mundo tienen ya muchos años de historia desde el momento en que se utilizaron por primera vez. Un buen ejemplo de ello es el Diagrama de Gantt, técnica popular de planificación gráfica que fue desarrollada por Henry Lawrence Gantt en el año 1917 (arqhys.com 2009)□.

La gestión de proyectos se refiere a la organización y administración de recursos de manera tal que se pueda culminar todo el trabajo requerido en un proyecto dentro del alcance, el tiempo, y los costos definidos (Arq. Percy C. Acuña Vigil 2009)□.

Las características o atributos comunes a la mayoría de los proyectos son:

- Tienen un objetivo. La naturaleza de un proyecto debe ser real y debe tener sustento.
- Tienen un calendario de actividades, y deben tener un plan de trabajo.
- Son complejos, están compuestos por múltiples elementos.
- Demandan recursos. Requieren habilidades, conocimientos, capital y equipo de diversas áreas de una organización.
- Tienen una estructura organizacional, en la cual hay roles y responsabilidades.

Según el Project Management Institute (PMI):

- Son temporales. Tiene un principio y un fin.
- Son únicos. Se inician por un único propósito, y porque solo el proyecto usará ese único recurso, tiempo, espacio y demás recursos para ese único momento.
- Requieren de elaboración progresiva. Los procesos y ciclos de vida del proyecto se inician como una idea general y se detallan conforme transcurre el proyecto.

El primer desafío de la gestión de proyectos es asegurarse de que el proyecto sea realizado dentro de los parámetros definidos. El segundo es la asignación y la integración de las acciones necesarias para resolver esos objetivos predefinidos. El proyecto, por lo tanto, es un sistema cuidadosamente seleccionado de actividades para utilizar los recursos (tiempo, dinero, recursos humanos, materiales, energía, espacio, provisiones, comunicación, calidad, riesgo, etc.) con el objetivo de dar cumplimiento a las metas predefinidos (Arq. Percy C. Acuña Vigil 2009)□.

Las tres restricciones tradicionales.

Los proyectos necesitan ser ejecutados y entregados bajo ciertas restricciones. Tradicionalmente, estas restricciones han sido alcance, tiempo y costo. Esto también se conoce como el Triángulo de la Gestión de Proyectos, donde cada lado representa una restricción. Un lado del triángulo no puede ser modificado sin impactar a los otros. Un refinamiento posterior de las restricciones separa la calidad del producto del alcance, y hace de la calidad una cuarta restricción.

La disciplina de la gestión de proyectos consiste en proporcionar las herramientas y técnicas que permiten al equipo de proyecto (no solamente al gerente del proyecto) organizar su trabajo para cumplir con todas esas restricciones.

Tiempo

El tiempo se descompone para propósitos analíticos en el tiempo requerido para completar los componentes del proyecto que es, a su vez, descompuesto en el tiempo requerido para completar cada tarea que contribuye a la finalización de cada componente.

Costo

El costo de desarrollar un proyecto depende de múltiples variables incluyendo costes de mano de obra, costes de materiales, administración de riesgo, infraestructura, equipo y utilidades.

Alcance

Requerimientos especificados para el resultado final.

Fue en los años 50 y 60 del siglo XX cuando la Gestión de Proyectos comenzó a profesionalizarse. Hasta entonces había sido meramente una disciplina. En aquella época, el desarrollo de grandes proyectos militares requería la coordinación del trabajo conjunto de equipos y áreas diferentes en la construcción de sistemas únicos (Angel Nieva López 2009)□.

Pero la innovación en la Gestión de Proyectos no se ha limitado únicamente a la mejora y perfección de sus técnicas y herramientas. El desvío en las planificaciones, costes y la falta, en ocasiones de calidad y utilidad de los entregables en los proyectos, provocó, que en los años 60 surgiera la necesidad de innovar y desarrollar nuevos métodos de organización y trabajo. Fue entonces cuando aparecieron diferentes organizaciones con el objetivo de

desarrollar los conocimientos, metodologías, procesos (lo que se conoce como áreas del conocimiento) y las prácticas necesarias para gestionar esos trabajos, con las mejores garantías de previsibilidad y calidad de los resultados.

Entre estas organizaciones encontramos el Project Management Institute (PMI) y el International Project Management Association (IPMA), ambos fundados en 1965. El PMI es una Asociación Internacional de Profesionales de Gestión de Proyectos, que cuenta con más de 200.000 miembros vigentes con presencia en más de 125 países por todo el mundo y más de 85.000 certificados PMP's (Project Management Professional). Desde su fundación ha estado desarrollando y publicando estándares profesionales para la Gestión de Proyectos, los cuales se recogen actualmente en el PMBOK (Project Management Body of Knowledge).

La innovación en la Gestión de Proyectos siempre ha estado ligada a la innovación en Dirección Estratégica y Organización. Muestra de ello es la evolución que han experimentado tanto la propia Gestión de Proyectos como la figura del Director de Proyectos. Entre los años 1940 y 1970 podemos observar una Gestión de Proyectos tradicional (con sus diagramas de gantt y sus asignaciones de recursos). Se asume que las organizaciones viven en un mundo estático y predecible para desarrollar los proyectos y las decisiones se toman sin darle importancia a la satisfacción del cliente.

El presupuesto, cronograma y la asignación de recursos se manejan a través de una serie de herramientas fijas y el ciclo de vida de proyecto consta solo de cuatro fases: Concepto, Planificación, Ejecución y Cierre. En cambio, en la Gestión de Proyectos moderna (desde 1980 hasta nuestros días) las cosas cambian, ya que se asume que vivimos en mundo caótico y no predecible para desarrollar los proyectos y, al revés que la Gestión de Proyectos tradicional, para la toma de decisiones se tiene en cuenta la opinión y satisfacción del cliente. Nacen nuevas técnicas y herramientas para gestionar los proyectos en los ámbitos de gestión de contratos, análisis de costes, análisis de riesgos, del entorno, negociación y gestión de cambios. Además, se incorporan nuevas fases al ciclo de vida de los proyectos.

Tendencias de la gestión de proyectos.

En la actualidad el reto principal de la Gestión de Proyectos es la globalización, que provoca que la Gestión de Proyectos se expanda hacia entornos internacionales y por tanto, multiculturales. Por ello cada vez más se requieren conocimientos culturales específicos en idiomas y costumbres tanto de los países donde se desarrolla el proyecto como de los diferentes integrantes del equipo, para poder superar con éxito los retos con los cuales las empresas y los proyectos multiculturales se encuentran diariamente. Otros retos actuales a los que se enfrenta la Gestión de Proyectos moderna son las exigencias para conseguir resultados más rápidos y con alta calidad en proyectos, donde existe cada vez más un alto grado de incertidumbre y se multiplican los riesgos, el mantener un buen ambiente en el equipo de proyecto, adaptarse a las continuas reorganizaciones y la excesiva dependencia de la tecnologías de la información (Angel Nieva López 2009)□.

En un futuro cercano la Gestión de Proyectos será una competencia requerida en todas las organizaciones,

todas requerirán que los Gestores de Proyectos estén certificados.

Existirá en pocos años mayor cantidad de oportunidades de trabajo en materia de gestión de de proyectos. No obstante, cada vez existirá mayor competencia y calidad entre los profesionales, que además tendrán que adaptarse a proyectos donde se trabaje a tiempo completo. Veremos como las organizaciones seguirán innovando en la Gestión de Proyectos y desarrollarán e incrementarán el uso de herramientas y evolucionarán nuevas teorías de gestión basadas en el liderazgo para poder afrontar con éxito los proyectos.

Los sistemas de Gestión de proyectos.

En proyectos pertenecientes a las TIC uno de los problemas más frecuentes es la falta de comunicación y de compartir conocimientos en los equipos de trabajo, un elemento que puede ayudar a establecer equipos de trabajo cohesionados de lo que está pasando con el proyecto es desarrollar modelos de interacción y comunicación efectivos, para ello la web 2.0 es un mecanismo muy potente, que permite un mayor grado interacción entre los usuarios (Alejandro Barros 2009)□.

En esta área se han desarrollado diferentes alternativas como las wiki o los blog cuyo uso puede llegar a lograr discretos resultados si de gestión de proyectos se trata.

Poner a disposición de los miembros de un equipo de proyecto una herramienta tipo blog o wiki les permite interactuar de forma simple, manejar la historia y poder reducir las asimetrías de información del proyecto en sus diferentes áreas (técnica, económica, estratégica, etc.).

Los modelos colaborativos de administración presentan importantes ventajas, al establecer mecanismos efectivos de comunicación entre los diferentes miembros de un proyecto, cada uno desde su rol puede conocer lo que está pasando en el proyecto y aportar con comentarios y observaciones.

A diferencia de los modelos colaborativos expuestos anteriormente, los sistemas gestión de proyectos cuentan con una modalidad de uso estructurada y muy jerarquizada que facilita los procesos de gestión de proyectos.

Los sistemas de gestión de proyectos (SGP) son herramientas que se orientan a la administración de forma automatizada de los recursos existentes para desarrollar un producto, cuya producción requiere de un conjunto de actividades o tareas que se desarrollen entre ellas en forma paralela o independiente y para las cuales una herramienta de este tipo cuenta con funcionalidades que automatizan y humanizan el trabajo.

Ejemplos de SGP.

OpenProj

Es multiplataforma, está desarrollado en Java y funciona bajo Windows, GNU/Linux, Unix y Mac OS. De hecho, una de las ventajas de OpenProj es que permite exportar e importar archivos .mpp y .mpx (extensiones de Microsoft Project). Además es completamente gratuito y está traducido a 8 idiomas incluyendo el español.(Salva

Castro 2008)□

- Presenta distintas funcionalidades de gestión , como pueden ser:

- Gestión de calendarios de trabajo.
- Existencia de diferentes vistas del proyecto (Gantt, diagramas de red, de recursos, histogramas, etcétera).
- Gestión de tareas, con niveles de jerarquía y todas las posibilidades de dependencia necesarias.
- Gestión de recursos (tantos humanos como materiales) asignados a un proyecto.
- Gestión de costes, aunque de forma básica.
- Gestión de líneas base para la replanificación controlada del proyecto.
- Funciones de seguimiento de proyectos, ya sea por dedicación de recursos o simplemente por avance por porcentaje (introducción manual).

Como principal defecto podemos destacar la falta de documentación. Tiene una interfaz muy intuitiva y ligera, sin embargo, tiene algunas carencias, como errores en la traducción o que la estética de las ventanas es la de Java por defecto y no se integra con la del sistema.

OpenProj está disponible de dos formas : Para descarga gratuita (OpenProj), o para contratación mediante la modalidad Software como Servicio, denominado Project-ON-Demand, con lo que se tiene la herramienta disponible en Internet para compartir con empleados, clientes y proveedores sin necesidad de complejas instalaciones.

ActiveCollab

Es una aplicación web, esto permite su administración y operación a través de un navegador y desde cualquier computadora con conexión. Fue desarrollado en el lenguaje PHP y usa MySQL como sistema gestor de bases de datos (ActiveCollab 2009)□. La plataforma recomendada para su instalación en un servidor es la siguiente:

- Apache 2.0 con soporte para PHP5.
- MySQL con soporte para InnoDB.

Para su versión 1.0 pasó a ser una aplicación de pago por uso. Un primer plan de precios fue eliminado ante reclamos de usuarios de versiones anteriores del sistema, quedando finalmente dos modos de distribución para su uso.

La desventaja de esta herramienta es que existen traducciones para muchos idiomas excepto en español, y hasta el momento, no presenta soporte para LDAP.

GanttProject

Es una herramienta libre y fácil de usar. Está basada en la creación de diagramas de Gantt y de otros recursos gráficos. Puede organizar el proyecto en un árbol de tareas y asignar recursos humanos que tienen que trabajar en cada tarea, con la posibilidad de crear dependencias entre estas últimas. Es posible generar informes en HTML y PDF, el intercambio de datos con Microsoft Project y con hojas de cálculo (Arleco 2008a)□.

Las principales ventajas que posee esta herramienta son:

- **Facilidad de aprendizaje:** La curva de aprendizaje es bastante baja, cualquier usuario con conocimientos básicos en la gestión de proyecto puede convertirse en un experto en poco tiempo.
- Es una herramienta multiplataforma desarrollada en java.
- **Abrir el código fuente:** GanttProject puede ajustarse a sus necesidades, implementar nuevas características, añadir sus propios informes, etcétera.

Desventajas

- Carece de muchas de las características adicionales que hacen de MS Project la herramienta estándar.
- Poca difusión entre las pequeñas y medianas empresas, para la gestión de sus proyectos.

Microsoft Project

Microsoft Project (MS Project) es un software para administración de proyectos desarrollado y vendido por **MICROSOFT** el cual está creado para asistir a los administradores de proyectos en el diseño de planes, asignación de recursos a tareas, rastreo de progresos y análisis de cargas de trabajo. La primera versión de Microsoft Project fue lanzada para DOS en 1984 por una compañía que trabajaba para Microsoft. Esta última, adquirió todos los derechos del software en 1985 y liberó la versión 2. La primera versión para Windows fue liberada en 1990, y fue llamada versión 1 para Windows. Un dato interesante es notar que la primera versión para DOS introdujo el concepto de Líneas de dependencia (link lines) entre tareas en la gráfica de Gantt. Aunque este software ha sido etiquetado como miembro de la familia Microsoft Office hasta el momento no ha sido incluido en ninguna de las ediciones de paquete de Office (quizá por ser un programa muy especializado) y se encuentra disponible en dos versiones, Standard y Professional (Arleco 2008b)□.

Desventajas de MS Project

- Sólo funciona sobre el sistema operativo Windows, no es libre.
- El 80% de los usuarios de MS Project usa el 20% de sus numerosas opciones.
- No se puede medir la productividad de las máquinas y las personas, tampoco su rendimiento productivo. Alto

costo e inversión.

Planner

Es una herramienta libre, una aplicación de escritorio para la gestión y seguimiento de proyectos de software. Permite la descomposición en tareas y subtareas, dependencias, identificación de la ruta crítica, diagramas de Gantt (Planner Team 2009)□. Entre sus funcionalidades permite definir un calendario, marcar eventos importantes, asignar recursos tanto humanos y materiales, establecer prioridades, entre otras más. En un principio fue desarrollada para GNU/Linux, pero consta de una versión beta disponible para Windows. Además de estas funcionalidades, ofrece:

- Cuadro Intuitivo: Se puede conocer la planificación completa de todo el proyecto, permitiendo establecer diferentes colores, estilos para cada planificación y para cada grupo de personal.
- Acceso a Redes: Otros usuarios pueden abrir el mismo archivo que se está editando con acceso de sólo lectura a través de redes.
- Planificación relacionada a tareas: Cada planificación debe ser adjuntada a una tarea que lo ayuda a rastrear y visualizar todas las planificaciones para una tarea específica.
- Cada archivo puede configurarse con una contraseña para abrir y una contraseña para modificar.
- Puede exportar a archivo de Texto (*.txt), archivo script SQL (*.sql), archivo HTML (*.htm), archivo XML (*.xml), archivo de hoja de cálculo (*.xls), porta-papeles de MS Windows, Formato de Texto Enriquecido (*.rtf), formato Word (*.doc).

dotProject

Está construido haciendo uso de aplicaciones de código abierto. Es una aplicación basada en la web, multiusuario, soporta varios idiomas. Básicamente es un programa al que se le van añadiendo proyectos, y dentro de estos proyectos las tareas que lo conforman (Ernesto Serrano 2009)□. Tiene todas las opciones básicas de cualquier gestor de proyectos:

- Empresas
- Proyectos
- Tareas
- Calendario
- Ficheros
- Contactos

- Usuarios
- Sistema

Es una herramienta que permite gestionar las distintas tareas que componen un proyecto. Además se perfila como una interesante herramienta para trabajar en entornos colaborativos, permitiendo a los integrantes del equipo trabajar compartiendo información relativa a los proyectos.

De forma general las características de la herramienta son:

- Permite la gestión y planificación de proyectos en entornos colaborativos.
- Basado en plataforma web por lo que permite la participación en línea de los miembros de un proyecto.
- Permite la asignación de recursos a un proyecto o a varios, así como la descomposición en tareas.
- Permite visualizar eventos y tareas en el calendario.
- Permite la visualización de informes y estadísticas sobre los proyectos registrados.
- Gestiona las actividades de la empresa al permitir la parcelación del proyecto en tareas.
- Permite a su vez la generación de gran cantidad de informes.

Después de haber analizado las características y funcionalidades existentes en los SGP más importantes, se ha concluido que los mismos de forma general no presentan módulos o funcionalidades donde se gestione estadísticas de cualquier índole en forma de gráficos.

Herramientas, lenguajes y tecnologías a utilizar.

Lenguajes de programación.

XML

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información (w3c 2009b)□.

JavaScript

Javascript es un lenguaje de programación utilizado para crear programas encargados de realizar acciones dentro del ámbito de una página web.

Se trata de un lenguaje de programación del lado del cliente. 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 (Miguel Angel Alvarez 2009)□.

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.

PHP

Es un lenguaje interpretado de alto nivel 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. PHP soporta el trabajo con CGI, permite procesar la información de formularios, generar páginas con contenidos dinámicos, y el trabajo con cookies.

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.

Quizás 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(php.net 2009)□.

CSS

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. CSS se utiliza para dar estilo a documentos

HTML y XML, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento (w3c 2009a)□.

HTML

HTML significa HyperText Markup Language. Es el lenguaje en que se escriben los millones de documentos que hoy existen en el WWW. Cuando accedemos a uno de estos documentos, el cliente (Netscape, IE, Mosaic, Lynx, Mozilla) los interpreta y los muestra. Los documentos HTML son ficheros de texto plano (también conocidos como ASCII) que pueden ser editados con cualquier editor de textos. El nombre de los ficheros escritos en lenguaje HTML suelen tener la extensión html o htm. El lenguaje HTML se basa en la sintaxis SGML. Esto significa que los diferentes elementos (párrafos, encabezamientos, tablas, listas, etc) de un documento para la WWW se señalan intercalando etiquetas que indican al navegador cómo debe mostrarlo (IES Gaviota 2009)□.

Herramientas utilizadas.

DBDesigner.

DBDesigner 4 es un software para el desarrollo de aplicaciones de bases de datos que integra diseño, modelado, creación y mantenimiento en un ambiente simple y fácil de usar.

DBDesigner 4 se compara con productos como Designer de Oracle, Rational Rose de IBM, e ERwin de Computer Associates, pero éste es un proyecto Open Source disponible para Microsoft Windows, Linux(para escritorios KDE y Gnome), y se distribuye bajo licencia GPL.

DBDesigner 4 es desarrollado y optimizado para ser usado con el servidor de bases de datos MySQL (mysql-hispano.org 2008)□.

Entre sus principales características se encuentran:

- Guarda los proyectos en XML nativo.
- Posibilidad de conectividad con otros SGDB a través de plug-ins añadibles (por defecto MySQL y PostgreSQL)
- Conectividad con la base de datos.
- Exportar e importar scripts .SQL brindando la posibilidad de realización de ingeniería inversa.

Quanta Plus.

Quanta Plus (*Quanta+*) es una herramienta libre de desarrollo de páginas web diseñado para el proyecto KDE que rápidamente se ha convertido en un editor maduro que cuenta con varias funcionalidades. Su versión actual es la

3.5 y proporciona un interfaz de múltiples documentos poderoso e intuitivo para los desarrolladores web. Puede incrementar exponencialmente la productividad, a través del uso de acciones personalizadas, guiones y barras de herramientas, con lo que se puede automatizar casi cualquier tarea (Daniel M. Maldonado 2008)□.

Características:

- Usa KIO para FTP y SSH, además de soportar otros protocolos.
- Posee asistentes para creación de tablas, enlaces y páginas en blanco.
- Resalta sintaxis de HTML, Javascript, CSS y varios lenguajes más.
- Contiene un analizador que informa acerca de la correcta creación de nuestras páginas.
- Soporta plugins a través de KParts.
- Los documentos pueden ser previsualizados dentro de la aplicación usando el motor KHTML.
- Es posible preprocesar los documentos a través de un servidor web antes de previsualizar.

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 (Netcraft 2009)□. El servidor Apache es usado por múltiples razones como disponibilidad, facilidad de instalación, pocos recursos hardware necesarios, precio, disponibilidad del código fuente, entre otras. Existen otros aunque suelen estar especializados en nichos concretos de mercado. Corre en una multitud de sistemas operativos, lo que lo hace una herramienta multiplataforma (linux.ciberaula.com 2009)□.

Algunas características:

- Apache es un software gratuito y libre. Esto le da una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor, lo podemos saber.
- Es altamente configurable y de diseño modular. Es muy sencillo ampliar sus capacidades. Actualmente existen muchos módulos y están disponibles para toda la comunidad. Otro aspecto importante es que cualquier persona que posea una experiencia avanzada en la programación de C o Perl puede escribir un módulo para realizar una función determinada.
- 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.

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 (netpecos.org 2009)□.

Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

Metodologías de desarrollo de software.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software.

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 y que papel deben de tener. Detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla (www.um.es 2009)□.

Metodología SXP.

SXP es una metodología de desarrollo de software compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo.

SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que sepamos por dónde andamos. 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 fases 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 este listo para ser entregado.
- Entrega, puesta en marcha.
- 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, lo que nos permite mejorar el diseño cada vez que se le añada una nueva funcionalidad.

SXP esta especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo (Gladys M. Peñalver Romero 2008)□.

PMBOK

PMBOK es un estándar en la gestión de proyectos desarrollado por el PMI. Se encuentra disponible en 11 idiomas: inglés, español, chino simplificado, ruso, coreano, japonés, italiano, alemán, francés, portugués de Brasil y árabe (Mario Torres 2008)□. En 1987, el PMI publicó la primera edición del PMBOK en un intento por documentar y estandarizar la información y prácticas generalmente aceptadas en la gestión de proyectos. La edición actual - la tercera - provee de referencias básicas a cualquiera que esté interesado en la gestión de proyectos. Posee un léxico común y una estructura consistente para el campo de la gestión de proyectos. La Guía del PMBOK es ampliamente aceptada por ser el estándar en la gestión de proyectos.

El PMBOK es una colección de procesos y áreas de conocimiento generalmente aceptadas como las mejores prácticas dentro de la gestión de proyectos, es un estándar reconocido internacionalmente (IEEE Std 1490-2003)(standards.ieee.org 2009)□ que provee los fundamentos de la gestión de proyectos que son aplicables a un amplio rango de proyectos, incluyendo construcción, software e ingeniería.

El PMBOK reconoce cinco procesos básicos y nueve áreas de conocimiento comunes a casi todos los proyectos. Los procesos se traslapan e interactúan a través de un proyecto o fase, son descritos en términos de: Entradas (documentos, planes, diseños, etc.), Herramientas y Técnicas (mecanismos aplicados a las entradas) y Salidas (documentos, productos, etc.). Las nueve áreas del conocimiento mencionadas en el PMBOK son:

- Gestión de la Integración de Proyectos.
- Gestión del Alcance en Proyectos.
- Gestión del Tiempo en Proyectos.
- Gestión de la Calidad en Proyectos.
- Gestión de Costos en Proyectos.
- Gestión del Riesgo en Proyectos.
- Gestión de Recursos Humanos en Proyectos.
- Gestión de la Comunicación en Proyectos.
- Gestión de la Procura (Logística) en Proyectos.

Gestión de las Comunicaciones del Proyecto según PMBOK.

La Gestión de las Comunicaciones del Proyecto es el Área de Conocimiento que define PMBOK con el propósito de analizar los procesos necesarios para asegurar la generación, recogida, distribución, almacenamiento, recuperación y destino final de la información del proyecto en tiempo y forma. Los procesos de Gestión de las Comunicaciones del proyecto proporcionan los enlaces cruciales entre las personas y la información, necesarios para unas comunicaciones exitosas .

Los informes de rendimiento organizan y resumen la información recogida, y presentan los resultados de cualquier análisis en comparación con la línea base para la medición del rendimiento. Los formatos más comunes de los informes de rendimiento incluyen diagramas de barras, histogramas y tablas.

Infografía

La infografía es una representación más visual que la propia de los textos en la que intervienen descripciones, narraciones o interpretaciones, presentadas de manera gráfica normalmente figurativa, que pueden o no coincidir con grafismos abstractos y/o sonidos. La infografía nació como un medio de transmitir información gráficamente. Los mapas y gráficos, son infográficos, es decir unidades menores de la infografía, con la que se presenta una

información completa aunque pueda ser complementaria o de síntesis (Elio Leturia 2009)□.

Tipos de infográficos

Los infográficos pueden dividirse en las categorías de gráficos, mapas, tablas y diagramas.

Los gráficos son los más comúnmente utilizados y presentan información numérica y estadística. Se dividen, a su vez, en gráficos de barra, de torta y de fiebre.

- El gráfico de barras funciona preferentemente con unidades, y lo que hacen es establecer una comparación entre ellas. Las barras presentan el mismo ancho y el alto depende de la cantidad que representen.
- El gráfico de torta (tarta, pastel o queso) indica la división de partes de un todo y sus proporciones son representadas generalmente en porcentajes. Está representado por un círculo que supone un todo y se encuentra dividido en partes. Estas partes no deben ser muchas, especialmente cuando suponen pequeñas partes del todo que se presenta, pues el gráfico se vuelve confuso y la información se ve desordenada.
- El gráfico de fiebre o línea muestra los cambios, expresados en números, a través del tiempo. Los gráficos de fiebre funcionan si:
 - La línea que traza el cambio de cantidades representa un período de tiempo y
 - Si cada cantidad establecida dentro de la línea representa incrementos por igual del tiempo indicado.

Partes de un infográfico

Para que un cuadro gráfico sea considerado completo debe poseer un titular, un texto explicativo corto, un cuerpo de información, una fuente y un crédito de autor (Elio Leturia 2009)□.

- El titular debe ser directo, preferentemente sintético a la vez que expresa el contenido del cuadro. Si se cree conveniente y está determinado en el manual de diseño, el titular puede venir acompañado de una bajada o subtítulo, siempre opcional.
- El texto debe ser pequeño y proveer al lector de toda la explicación necesaria para la comprensión del cuadro. Lo que el cuerpo del cuadro no explique debe ser explicado por dicho texto.
- El cuerpo viene a ser la esencia misma del cuadro, la propia información visual: las barras, la torta, las líneas de fiebre, el mapa, etc. Este cuerpo necesita y presenta información tipográfica explicativa a manera de etiquetas que pueden ser números, fechas o palabras descriptivas.
- La fuente indica de dónde se ha obtenido la información que se presenta en el infográfico y es muy importante, pues señala el origen de la misma.

- El crédito señala al autor o autores del infográfico, tanto de la configuración como de la investigación. La fuente y crédito utilizan, por lo general, una tipografía que no excede los siete puntos y se ubica en un lugar que no distraiga la atención del lector.

Los procesos de gestión de proyectos hoy en día han cobrado mayor importancia en el mundo empresarial por los distintos beneficios que trae consigo. Estos procesos logran mayor eficiencia al automatizarse mediante SGP que logren cubrir todas las especificaciones de los estándares mundiales para la gestión de proyectos como lo es PMBOK. De las nueve áreas del conocimiento que propone PMBOK la gestión de la comunicación en los proyectos y específicamente los informes de rendimientos emitidos de forma clara y concisa mediante infográficos, pueden dar una idea al líder de proyecto del estado de indicadores vitales para el buen funcionamiento del proyecto y mejorar la toma de decisiones para el futuro desempeño del equipo de trabajo. El SGP dotProject al igual que los restantes analizados en este estudio no muestran información de forma general haciendo uso de infográficos, se limitan solo al uso de gráficas de Gantt. De modo general no explotan las potencialidades que brindan el procesamiento gráfico de información.

Capítulo 2. Descripción y análisis de la solución propuesta.

Especificar los requisitos de software es de vital importancia para lograr un desarrollo exitoso y eficiente del mismo. Es la garantía para verificar que las necesidades del cliente sean cubiertas en su totalidad. Lograr identificarlos es una tarea compleja que constituye un hito fundamental en la realización de un producto informático.

AMVEX, fuente de datos de GENEST.

El módulo GENEST se nutre de reportes en haciendo uso del lenguaje XML. Estos reportes en formato XML son elaborados y enviados por una herramienta nombrada AMVEX (Aplicación para el monitoreo de ventanas del sistema X Windows) y que se encuentra instalada en cada una de las computadoras conectadas al Home Compartido que se desean controlar. 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). El formato antes descrito aparece representado en la siguiente figura.

Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.

```
-<report>
  <user> asisinta </user>
  <date> 2009-04-26 </date>
  <ip> 10.33.2.7 </ip>
- <process t="7">
  + <application name="rox" time="89" t="4"></application>
  + <application name="leafpad" time="331" t="2"></application>
  + <application name="mozilla" time="754" t="23"></application>
  + <application name="gkrellm" time="13" t="1"></application>
  + <application name="psi" time="232" t="4"></application>
  - <application name="xpdf" time="68" t="2">
    <item time="12">Xpdf: /home/asisinta/Desktop/chapter02.pdf</item>
    <item time="56">Xpdf: /home/asisinta/Desktop/RPM_v1_02.pdf</item>
  </application>
  - <application name="openOffice.org" time="429" t="2">
    <item time="179">Cap 3 - OpenOffice.org Writer </item>
    <item time="250">Tesis - OpenOffice.org Writer </item>
  </application>
</process>
</report>
```

Figura#1. Ejemplo de reporte en formato XML.

La aplicación funciona la mayor parte del tiempo en modo invisible, se ejecuta como demonio en las computadoras clientes, supervisando las ventanas enfocadas que el usuario utilice durante su estancia en el ordenador. Los datos son recopilados y guardados en formato XML y al producirse el cierre de sesión son enviados a un servidor central.

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.

Una vez que se encuentran los reportes en servidor son analizados por GENEST de forma automática (el script encargado se ejecuta diariamente a la hora señalada por el administrador del servidor en el cron del sistema) y sus datos son guardados en la base de datos para su posterior uso. Para la recepción de los reportes GENEST crea de forma automatizada directorios para cada uno de los usuarios de dotProject, en los cuales se guardarán los reportes que vayan arribando al servidor. Una vez leídos los reportes se eliminan y se actualiza el estado de los directorios, creando nuevamente las carpetas con los usuarios actuales de dotProject; esto garantiza disponer siempre de los directorios de los usuarios que se dan de alta en el sistema y no contar con los que se han dado de baja.

Levantamiento de requisitos

Para recoger los requisitos se utiliza la Lista de Reserva del Producto según la metodología seleccionada; un artefacto que recoge en lenguaje natural las funcionalidades que debe cumplir el sistema para su desarrollo y está conformado por una lista priorizada que define el trabajo que se va a realizar en el proyecto.

Prioridad	Ítem *	Descripción	Estimación	Estimado por
<i>Muy Alta</i>				
	1	Obtener los nombres de todos los directorios donde están los reportes que haya enviado el cliente(killer 532).	0.5	ANA. y PROG.
	2	Obtener los nombres de todos los reportes XML que están en cada uno de los directorios .	0.5	ANA. y PROG.

	3	Obtener los datos de todos los reportes XML que haya enviado el cliente y guardarlos en la base de datos.	2	ANA. y PROG.
	5	Eliminar los reportes una vez que se hayan guardado los datos y actualizar todos los directorios.	0.5	ANA. y PROG.
	6	Obtener información de la base de datos para generar el reporte general.	0.5	ANA. y PROG.
	8	Mostrar el reporte general mediante una gráfica a partir de la información obtenida de la BD.	1	ANA. y PROG.
	9	Obtener información de la base de datos para generar los reportes de usuarios .	0.5	ANA. y PROG.
	10	Mostrar los reportes de los usuarios mediante gráficas a partir de la información obtenida de la BD.	2	ANA. y PROG.
	11	Obtener información de la base de datos para generar los reportes de proyectos.	0.5	ANA. y PROG.
	12	Mostrar los reportes de los proyectos mediante gráficas a partir de la información obtenida de la BD.	2.0	ANA. y PROG.
	13	Obtener información de la base de datos para generar los reportes en formato PDF .	0.5	ANA. y PROG.
	14	Obtener los reportes en formato PDF a partir de la información obtenida de la BD.	1	ANA. y PROG.
Alta				
	1	Crear un paginador de resultados.	1	ANA. y PROG.
RNF				

	1	Tener instalado la plataforma de desarrollo LAMP.	0.3	ANA. y <i>PROG</i>
	2	El sistema deberá funcionar sobre plataforma GNU/Linux.		ANA. y <i>PROG</i>

Propuesta del sistema a implementar.

De forma general se desea implementar un módulo para el SGP dotProject que permita al usuario realizar la selección de diferentes reportes filtrados principalmente por intervalos de tiempo, nombre de usuario, grupo de aplicaciones, aplicaciones y las rutas de las distintas aplicaciones accedidas por él.

Producto a que el volumen de información puede ser abundante se hará uso de infográficos para mostrar la información solicitada. Para esto se utilizarán tres tipo de infográficos : tablas, gráficos de pastel y barras .

Los infográficos usados deberán responder a las siguientes características:

- Deben ser sencillos y no distorsionar la información en un afán de lograr infográficos novedosos desde el punto de vista del diseño.
- La proporción debe ser la correcta .
- Cuando se comparan datos, todos los elementos deben presentar las mismas texturas y los mismos valores.

Las barras serán rectangulares completamente y no con terminaciones redondeadas o triangulares; el gráfico de tarta (queso o pastel) será circular y no ovalado.

Para una mejor interpretación de los resultados que serán graficados se tomó la decisión de clasificar los mismos según las necesidades del cliente. Las aplicaciones que se monitorean con AMVEX se clasifican en los siguientes grupos:

- Navegadores web
- Clientes jabber
- Ofimática
- IDE (Entornos de desarrollo integrado)
- Otros(cualquier aplicación que no entre entre las clasificaciones anteriores)

La interfaz principal del módulo GENEST estará compuesto por un sistema de navegación por pestañas – siendo consecuente con el diseño establecido por dotProject- que dará acceso a cada uno de los diferentes grupos de información representados en las siguientes vistas:

➤ **Vista General.**

Permite al usuario visualizar la gráfica correspondiente a los reportes generales por cada uno de los grupos de aplicaciones y para todos los proyectos de la empresa. La gráfica será de tipo pastel y el usuario podrá seleccionar el intervalo de tiempo en que se medirá dicho reporte.

➤ **Vista de usuarios.**

Permite al usuario visualizar las gráficas correspondientes a los reportes de todos los usuarios pertenecientes a la empresa.

Estos reportes se muestran en tres secciones:

○ **Primera sección: Todos los usuarios.**

Reportes del tiempo de utilización de cada grupo de aplicaciones por cada uno de los usuarios de la empresa. Se utilizará una gráfica de tipo barra y producto al gran número de usuarios que puede existir en la empresa se mostrará la información de forma paginada. Se permitirá seleccionar el intervalo de tiempo en que se medirá dicho reporte.

○ **Segunda sección: Grupo de aplicaciones.**

Reporte de las 5 aplicaciones más usadas por un usuario en específico. Se medirá en un intervalo de tiempo determinado y de un grupo de aplicaciones previamente seleccionado. Se utilizará una gráfica de tipo pastel.

○ **Tercera sección: Ruta de aplicaciones.**

Reporte de las 5 rutas más accedidas de una aplicación específica en un intervalo de tiempo determinado y por un usuario en particular. Se utilizará una gráfica de tipo pastel.

➤ **Vista de proyectos.**

Permite al usuario visualizar las gráficas correspondientes a los distintos reportes de todos los proyectos de la empresa.

Estos reportes se muestran en dos secciones:

- **Primera sección : Por miembros de proyectos.**

Son reportes del tiempo de utilización de cada grupo de aplicaciones por cada uno de los miembros del proyecto que es dueño el usuario que este autenticado. Permite seleccionar el intervalo de tiempo que se medirá para dicho reporte. Se utilizará una gráfica de tipo barra y producto al gran número de usuarios que puede tener el proyecto se mostrará la información de forma paginada.

- **Segunda sección: Todos los proyectos.**

Reportes del tiempo de utilización de cada grupo de aplicaciones por cada uno de los proyectos existentes en la empresa. Se utilizará una gráfica de tipo barra y debido al gran número de proyectos que puede tener la empresa se mostrará la información de forma paginada.

- **Vista de reportes PDF.**

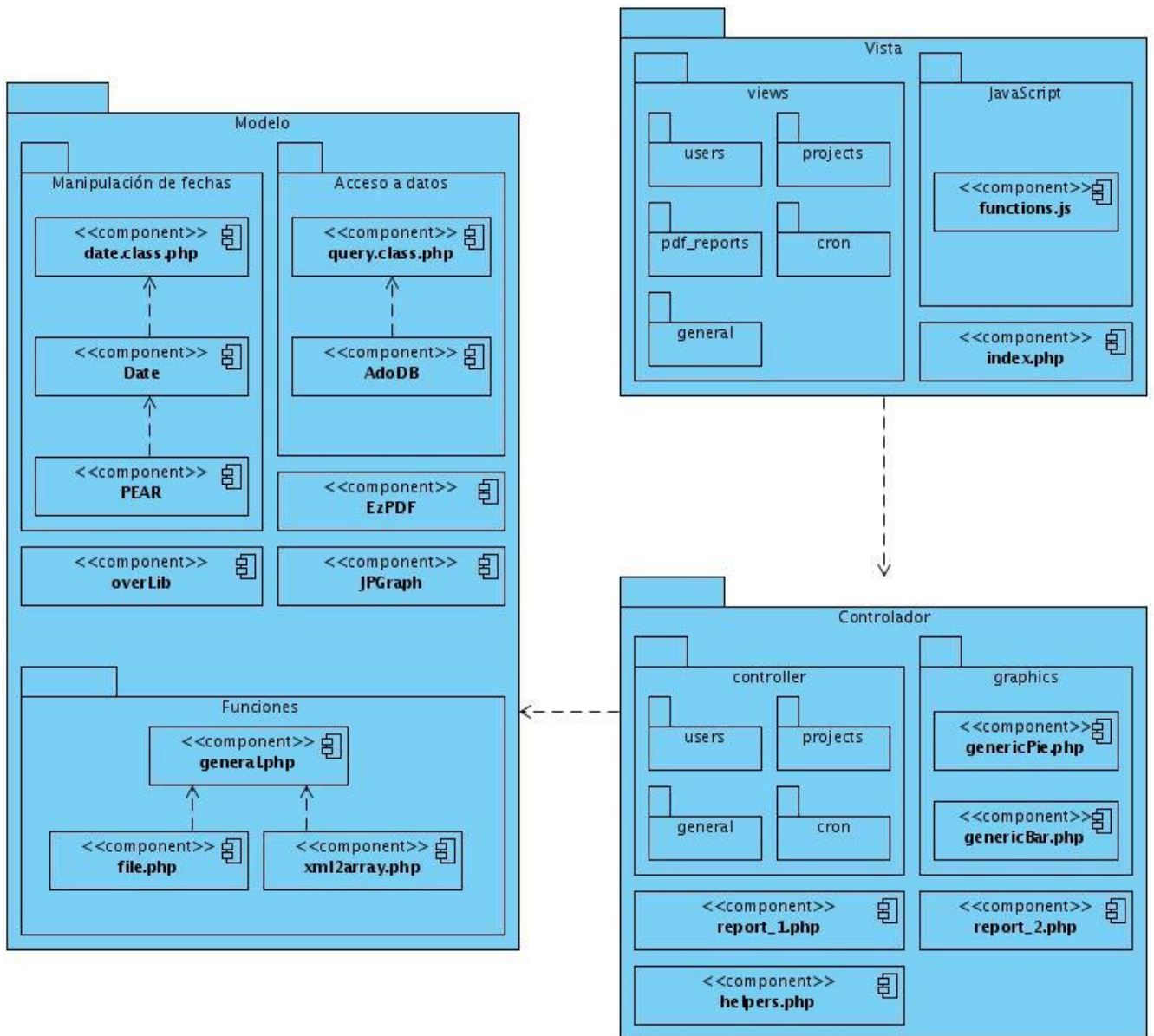
Reportes con parte de la información existente en el resto de las vistas. Cada uno de los reportes posee una descripción bien detallada de la información que muestra.

- **Vista de ejecución manual de CRON**

Posibilita la ejecución manual y desde el mismo módulo del script encargado de leer todos los reportes existentes. Esta vista muestra los logs resultantes de esta operación.

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 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 (Symfony, la guía definitiva 2009)□. En la siguiente figura se muestra el diagrama de componentes que describe la arquitectura del módulo .



Figura#2. Diagrama de componentes.

Modelo: Producto a que el dotProject constituye un framework, todas las clases de acceso a datos y lógica del negocio que utiliza el módulo GENEST son las mismas con las que cuenta el mismo. 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*.

➤ ***CAppUI***

Clase encargada de gestionar los elementos de interfaz de cada usuario que esté autenticado. Manipula la información personal del usuario, como son el nombre completo, compañía, departamento, correo electrónico, identificador de conexión, entre otros.

➤ ***CDate***

Clase encargada de realizar las operaciones con fechas, como bien lo indica su nombre, constituye una capa de abstracción del paquete ***Date*** de la librería ***PEAR***. 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.

➤ ***Jpgraph***

Librería que incluye una serie de clases que sirven para crear imágenes con todo tipo de gráficas dinámicamente desde páginas PHP (jpGraph 2009)□.

La mayoría de las configuraciones de las gráficas vienen con opciones por defecto, así que resulta bastante sencillo obtener resultados rápidamente.

Esta librería proporciona las imágenes con un reducido peso en bytes. Posee soporte para las librerías GD1 y GD2. Permite la confección de diversos tipos de gráficas 2D o 3D, como de puntos, líneas, tartas, barras y cajas. Altamente configurable con distintos tipos de colores, leyendas, tipografías e imágenes de fondo.

➤ ***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 (R&OS 2009)□.

➤ ***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 (Roberto Bárcenas 2009)□.

➤ **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 (overLIB 2009)□.

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 módulo GENEST las vistas están agrupadas en varios directorios dentro del directorio *view* del módulo. Estos son:

- general
- projects
- users
- pdf-reports
- cron

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 *general_view.php* y en el caso de las vistas de *users* y *projects* gestionan las interfaces correspondientes a cada una de las secciones de navegación que tienen estas vistas. En la vista *users* las interfaces correspondientes a las tres secciones que posee la misma se manipulan desde los componentes: *uv_1.php*, *uv_2.php* y *uv_group.php*. En *uv_group.php* se gestiona la interfaz de la primera sección de la vista *users* y es donde se muestran de forma paginada todos los usuarios de la empresa. En *uv_1.php* se gestiona la interfaz de la segunda sección de la vista *users* y manipula la información relacionada con la utilización de las aplicaciones de un usuario específico. En *uv_2.php* se gestiona la interfaz de la tercera y última sección de la vista *users*, es aquí donde se manipula la información de los ficheros de un usuario específico. Además se encuentra la página principal del módulo donde se incluye a su vez el fichero JavaScript correspondiente.

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 varios directorios dentro del directorio *controller* del módulo. Estos son:

- general
- projects

- users
- cron

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 .Ej : los controladores asociados al directorio de vistas *users* se encuentran en el directorio *users* de los controladores.

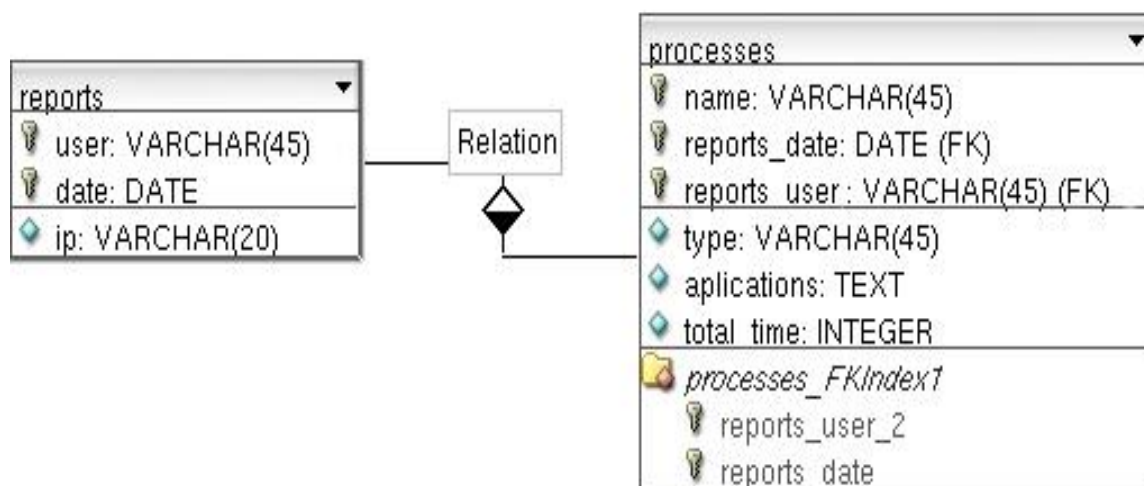
Para la construcción de las gráficas se implementaron dos controladores (*genericBar.php* y *genericPie.php*) correspondientes a los dos tipos de gráficas utilizadas por el módulo, la característica principal de estos es que son parametrizables facilitando una mayor reusabilidad de los mismos. (Ver anexos #12 y #13).

Los controladores *report_1.php* y *report_2.php* corresponden los dos reportes en formato PDF, los mismos aparecen en la raíz del módulos por cuestiones arquitectónicas del dotProject que impiden que su tratamiento sea igual al del resto de los controladores que están relacionados directamente con las vistas.

Diseño de la base de datos

El SGP dotProject posee características peculiares en cuanto al proceso de instalación de módulos 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 módulo GENEST se siguió esta política de desarrollo para la confección de su base de datos.

En la siguiente figura se presenta el diseño de la base de datos del módulo GENEST.



Figura#3. Diseño de la base de datos de módulo GENEST.

Guía para la creación de un módulo

El SGP dotProject es distribuido bajo licencia GPL, lo cual nos permite reutilizar el código y reimplementar funcionalidades ya existentes o implementar nuevas de acuerdo a las necesidades del usuario. El mismo tiene la ventaja de estar implementado con una estructura modular. Esto facilita en gran medida los procesos de mejoras de este software en cuanto al aumento de las funcionalidades que posee. Para poder integrar el módulo GENEST al dotProject primeramente hay tener conocimientos relacionados con el proceso de creación e instalación de un módulo en dotProject.

A continuación veremos los pasos a seguir para la creación de módulos, esto nos dará la posibilidad de tener mayor organización y una homogeneidad a la hora de crear el módulo GENEST.

Creación de un módulo para dotProject en seis pasos.

- Crear un directorio con el nombre del módulo.
- Las imágenes e iconos se deben guardar en un directorio /images dentro de ese mismo directorio.
- Desarrollar un script de instalación nombrado setup.php para instalar, actualizar, eliminar y configurar el módulo. A continuación se muestra el código fuente de un instalador de ejemplo:

```
<?php

$config = array();
$config['mod_name'] = 'Prueba'; // nombre del módulo
$config['mod_version'] = '0.1'; // especificar la versión del módulo
$config['mod_directory'] = 'prueba'; // define donde esta el directorio del módulo
$config['mod_setup_class'] = 'DP_Prueba'; // el nombre de la clase instaladora que se define más adelante en este fichero
$config['mod_type'] = 'user'; // tipo de módulo, como estandar se pone 'user' para los módulos adicionales
$config['mod_ui_icon'] = 'prueba.jpg'; // nombre del icono o logo del módulo
$config['mod_description'] = 'Esto es un módulo de prueba'; // Pequeña descripción del módulo
$config['mod_config'] = false; // determina si se muestra o no el link de configuración en la vista de administración del módulo.

if (@$a == 'setup')
{
    echo dPshowModuleConfig( $config );
}

class DP_Autos
{
    function install()
    {
        /*En esta función se especifican los requerimientos del nuevo módulo,
        desde las tablas que necesite hasta la creación de directorios auxiliares*/
    }
}
```

```

//ejemplo de la creación de una tabla para el módulo
$sql = '(
    prueba_id VARCHAR(45) NOT NULL,
    descripcion TEXT NOT NULL,
    PRIMARY KEY(prueba_id)
)TYPE=MyISAM';

$q = new DBQuery;
$q->createTable('prueba');
$q->createDefinition($sql);
$q->exec();

}

function remove()
{

    /*forma de eliminar los recursos creados por el módulos y que eran necesarios solo para su funcionamiento*/
    //como eliminar la tabla del módulo
    $q = new DBQuery;
    $q->dropTable('prueba');
    $q->exec();
}

}

?>

```

- Crear el fichero index.php donde se encuentra la interfaz general que brindará el módulo.
- Según las observaciones realizadas a todos los módulos del dotproject, se determinó los elementos comunes para los mismos en cuanto a la implementación de su interfaz general (index.php). Se concluyó que la manera utilizada por dotProject para mostrar los elementos de cabecera de los módulos es como se muestra en el siguiente fragmento de código fuente:

```

<?php
$titleBlock = new CTitleBlock($AppUI->_('Módulo de prueba'),'prueba.png', $m, "$m.$a" );
$titleBlock->show();
?>

```

- Por último realizamos los pasos necesarios para la instalación de un módulo en doProject. Ver anexo #11.

El diseño propuesto a partir de los requerimientos planteados por el cliente cumple con los estándares que exige dotProject para la creación de módulos y la navegación en los mismos. El estilo arquitectónico MVC que se propone para el módulo proporcionará robustez al mismo por garantizar que ante posibles cambios se afecten la menor cantidad de componente posibles. En otros aspectos se alcanzará una integración bastante eficiente al

dotProject si seguimos las pautas que al final de este capítulo se plantean.

Capítulo 3. Desarrollo ágil del módulo GENEST.

A partir de la arquitectura propuesta para el módulo GENEST se expondrán los elementos más importantes abordados durante el desarrollo ágil de la solución propuesta. Todo el proceso de desarrollo se describe a través de Historias de Usuarios y Tareas de Ingeniería.

Planificación del proyecto por roles.

SXP define diferentes roles para lograr un exitoso resultado en el proceso de desarrollo de software , estos son:

Rol	Nombre
Gerente	Karel Tito Catá
Cliente	Grupo Unicornios. UCI
Programadores	Karel Tito Catá Yoani Santana Leiva
Analista	Sheila Castro Hernandez
Diseñadores	Karel Tito Catá
Encargado de Pruebas	Karel Tito Catá
Arquitecto	Sheila Castro Hernández

Historias de usuarios.

Las historias de usuario son la técnica utilizada por SXP para especificar los requerimientos de software. Las mismas son escritas por los clientes como las tareas que el sistema debe hacer y su construcción es en lenguaje natural, no excediendo su tamaño de unas pocas líneas de texto.

Historia de usuario U-GENEST-01

Historia de Usuario	
Número: U-GENEST-01	Nombre Historia de Usuario: Gestionar directorios y ficheros.
Modificación de Historia de Usuario Número: 1	
Usuario: Karel Tito Catá	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 3.5
Riesgo en Desarrollo: Bajo	Puntos Reales: 2.5
Descripción: Permite 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 para su posterior análisis y eliminación. Finalmente actualiza el estado de los directorios de reportes.	
Observaciones:	
Prototipo de interfaz:	

Historia de usuario U-GENEST-02

Historia de Usuario	
Número: U-GENEST-02	Nombre Historia de Usuario: Diseño de la base de datos.
Modificación de Historia de Usuario Número:	
Usuario: Karel Tito Catá	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo	Puntos Reales: 1
Descripción: Diseño de la base de datos del módulo GENEST.	
Observaciones: No presentan relación con el resto de las tablas de dotProject por que el estándar de desarrollo de módulos para el mismo posibilita la instalación de las tablas de los nuevos módulos sin que guarden relación con las del sistema.	
Prototipo de interfaz:	

Historia de usuario U-GENEST-03

Historia de Usuario	
Número: U-GENEST-03	Nombre Historia de Usuario: Crear instalador del módulo
Modificación de Historia de Usuario Número: 1	
Usuario: Karel Tito Catá	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo	Puntos Reales: 1
Descripción: Permite la creación del instalador base del módulo GENEST. Se encarga de crear un módulo solo con las configuraciones necesarias para su instalación y sin interfaz alguna.	
Observaciones:	
Prototipo de interfaz:	

Historia de usuario U-GENEST-04

Historia de Usuario	
Número: U-GENEST-04	Nombre Historia de Usuario: Creación de CRON.
Modificación de Historia de Usuario Número: 1	
Usuario: Karel Tito Catá	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo	Puntos Reales: 1
Descripción: Permite la extracción de los datos contenidos en todos los reportes XML y su posterior salva en la base de datos .	
Observaciones:	
Prototipo de interfaz:	

Historia de usuario U-GENEST-05

Historia de Usuario	
Número: U-GENEST-05	Nombre Historia de Usuario: Visualizar reporte general.
Modificación de Historia de Usuario Número: 1	
Usuario: Karel Tito Catá	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1.5
Riesgo en Desarrollo: Bajo	Puntos Reales: 2
Descripción: Permite al usuario visualizar la gráfica correspondiente a los reportes generales por cada uno de los grupos de aplicaciones . Además permite seleccionar en el intervalo de tiempo que se medirá para dicho reporte.	
Observaciones:	
Prototipo de interfaz: Ver anexo #1	

Historia de usuario U-GENEST-06

Historia de Usuario	
Número: U-GENEST-06	Nombre Historia de Usuario: Visualizar reportes de usuarios.
Modificación de Historia de Usuario Número: 1	
Usuario: Karel Tito Catá	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 4
Riesgo en Desarrollo: Bajo	Puntos Reales: 4
Descripción: Permite al usuario visualizar las gráficas correspondiente a los distintos reportes de usuarios. Estos reportes se muestran en tres secciones: <ul style="list-style-type: none"> ● Primera sección: todos los usuarios. ● Segunda sección: grupo de aplicaciones. ● Tercera sección: aplicaciones. 	
Observaciones:	
Prototipo de interfaz: Ver anexo #2, #3 y #4.	

Historia de usuario U-GENEST-07

Historia de Usuario	
Número: U-GENEST-07	Nombre Historia de Usuario: Visualizar reportes de proyectos.
Modificación de Historia de Usuario Número: 1	
Usuario: Karel Tito Catá	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Bajo	Puntos Reales: 2.5
Descripción: Permite al usuario visualizar las gráficas correspondiente a los distintos reportes de proyectos. Estos reportes se muestran en dos secciones: <ul style="list-style-type: none">● Primera sección: por miembros de proyecto.● Segunda sección: todos los proyectos.	
Observaciones: En caso de que el usuario que esté autenticado no sea dueño de ninguno de los proyectos existentes se le debe notificar su estado.	
Prototipo de interfaz: Ver anexo #5 y #6.	

Historia de usuario U-GENEST-08

Historia de Usuario	
Número: U-GENEST-08	Nombre Historia de Usuario: Visualizar reportes en formato PDF.
Modificación de Historia de Usuario Número: 1	
Usuario: Karel Tito Catá	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1.5
Riesgo en Desarrollo: Bajo	Puntos Reales: 1.5
Descripción: Permite al usuario exportar en formato PDF las estadísticas que se muestran en el resto de las vistas.	
Observaciones:	
Prototipo de interfaz: Ver anexo #7	

Historia de usuario U-GENEST-09

Historia de Usuario	
Número: U-GENEST-09	Nombre Historia de Usuario: Ejecutar el CRON de forma manual.
Modificación de Historia de Usuario Número: 1	
Usuario: Karel Tito Catá	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo	Puntos Reales: 0.5
Descripción: Permite que el usuario de forma manual ejecutar el script que ordena al sistema leer los datos de todos los reportes XML y pasarlos a la BD (el sistema de forma automática ejecuta este script pero a una hora indicada del día).	
Observaciones:	
Prototipo de interfaz:	

Plan de versiones

La plantilla de Plan de versiones, es un artefacto generado por la metodología empleada durante el proceso de desarrollo. En este el cliente define el valor que posee el negocio según las características deseadas.

Esta plantilla proporciona ventajas, tales como:

- Define cuales son las historias de usuario más significativas, y las ubican en las iteraciones según esta prioridad.
- Divide el proceso de desarrollo de software en iteraciones, planificando el trabajo a realizar en cada una de ellas.

Plan de releases

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
Iteración 1	Se gestionan los ficheros. Se diseña la base de datos y se crea el instalador.	U-GENEST-01 U-GENEST-02 U-GENEST-03	15-10-2008 al 15-11-2008
Iteración 2	Implementación del cron. Implementación de las vistas: General y Usuario.	U-GENEST-04 U-GENEST-05 U-GENEST-06	24-11-2008 al 31-01-2009
Iteración 3	Implementacion de las vistas : Proyecto, Reportes PDF y Ejecución manual de Cron.	U-GENEST-07 U-GENEST-08 U-GENEST-09	02-02-2009 al 04-03-2009

Tareas de Ingeniería.

En la fase de desarrollo de la metodología SXP 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 U-GENEST-01

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-GENEST-01
Nombre Tarea: Listar directorios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 15-10-2008	Fecha Fin: 18-10-2008
Programador Responsable: Karel Tito Catá.	
Descripción: Se listan todos los directorios existentes en una dirección específica.	
Observación:	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: U-GENEST-01
Nombre Tarea: Eliminar directorios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 20-10-2008	Fecha Fin: 22-10-2008
Programador Responsable: Karel Tito Catá.	
Descripción: Se eliminan todos los directorios existentes en una dirección específica.	
Observación:	

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: U-GENEST-01
Nombre Tarea: Crear directorios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 23-10-2008	Fecha Fin: 25-10-2008
Programador Responsable: Karel Tito Catá.	
Descripción: Dado un arreglo de nombres se crean tantos directorios como nombres existan en el arreglo. Todo esto en una dirección específica.	
Observación:	

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: U-GENEST-01
Nombre Tarea: Listar ficheros.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 27-10-2008	Fecha Fin: 29-10-2008
Programador Responsable: Karel Tito Catá.	
Descripción: Listar los ficheros existentes en una dirección específica.	
Observación: Se listarán solo los ficheros de extensión XML	

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: U-GENEST-01
Nombre Tarea: Eliminar ficheros.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 30-10-2008	Fecha Fin: 01-11-2008
Programador Responsable: Karel Tito Catá.	
Descripción: Eliminar los ficheros existentes en una dirección específica.	
Observación: Se eliminarán todos los ficheros existentes.	

Tareas de Ingeniería para la historia de usuario U-GENEST-02

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-GENEST-02
Nombre Tarea: Diseño de la base de datos del módulo GENEST .	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 03-11-2008	Fecha Fin: 07-11-2008
Programador Responsable: Karel Tito Catá.	
Descripción: Diseño de la base de datos del módulo GENEST .	
Observación:	

Tareas de Ingeniería para la historia de usuario U-GENEST-03

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-GENEST-03
Nombre Tarea: Estudio de la estructura de los módulos del dotProject .	
Tipo de Tarea: Estudio	Puntos Estimados: 0.5
Fecha Inicio: 10-11-2008	Fecha Fin: 12-11-2008
Programador Responsable: Karel Tito Catá.	
Descripción: Estudio de la estructura de los módulos del dotProject para comprender el mecanismo de instalación que se utiliza.	
Observación:	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: U-GENEST-03
Nombre Tarea: Implementación del instalador.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 13-11-2008	Fecha Fin: 15-11-2008
Programador Responsable: Karel Tito Catá.	
Descripción: Implementación del instalador .	
Observación:	

Tareas de Ingeniería para la historia de usuario U-GENEST-04

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-GENEST-04
Nombre Tarea: Implementación del CRON.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 24-11-2008	Fecha Fin: 29-11-2008
Programador Responsable: Karel Tito Catá.	
Descripción: Implementación del CRON.	
Observación:	

Tareas de Ingeniería para la historia de usuario U-GENEST-05

Tarea de Ingeniería	
Número Tarea:1	Número Historia de Usuario: U-GENEST-05
Nombre Tarea: Estudio de la librería JPGRAPH	
Tipo de Tarea: Estudio	Puntos Estimados: 0.5
Fecha Inicio: 01-12-2008	Fecha Fin: 03-12-2008
Programador Responsable: Karel Tito Catá.	
Descripción: Estudio de la librería JPGRAPH, específicamente las gráficas de barra y de pastel.	
Observación:	

Tarea de Ingeniería	
Número Tarea:2	Número Historia de Usuario: U-GENEST-05
Nombre Tarea: Creación de una gráfica de pastel.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 04-12-2008	Fecha Fin: 06-12-2008
Programador Responsable: Karel Tito Catá.	
Descripción: Creación de una gráfica parametrizable de tipo pastel para lograr que sea lo más genérica posible.	
Observación: El objetivo es que sea utilizada por cualquier otra vista que requiera este tipo de gráfica.	

Tarea de Ingeniería	
Número Tarea:3	Número Historia de Usuario: U-GENEST-05

Nombre Tarea: Controlador de la vista general.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 08-12-2008	Fecha Fin: 10-12-2008
Programador Responsable: Karel Tito Catá.	
Descripción: Creación de un fichero que sea el encargado de seleccionar los datos que serán mostrados por la gráfica de la vista general.	
Observación:	

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: U-GENEST-05
Nombre Tarea: Diseño e implementación de la vista general .	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 11-12-2008	Fecha Fin: 13-12-2008
Programador Responsable: Karel Tito Catá.	
Descripción: Diseño e implementación de la vista general .	
Observación:	

Tareas de Ingeniería para la historia de usuario U-GENEST-06

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-GENEST-06
Nombre Tarea: Creación de una gráfica de barras.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 05-01-2009	Fecha Fin: 07-01-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Creación de una gráfica parametrizable de tipo barra para lograr que sea lo más genérica posible.	
Observación: El objetivo es que sea utilizada por cualquier otra vista que requiera este tipo de gráfica.	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: U-GENEST-06
Nombre Tarea: Controlador para la primera sección de la vista de usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 08-01-2009	Fecha Fin: 10-01-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Creación de un fichero que sea el encargado de seleccionar los datos que serán mostrados por la gráfica de la primera sección de la vista de usuarios. Reportes del tiempo de utilización de cada grupo de aplicaciones por cada uno de los usuarios de la empresa. Además permite seleccionar en el intervalo de tiempo que se medirá para dicho reporte.	
Observación:	

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: U-GENEST-06
Nombre Tarea: Diseño e implementación de la vista principal de usuarios.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 12-01-2009	Fecha Fin: 14-01-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Diseño e implementación de la vista principal de usuarios.	
Observación: Esta vista contendrá a las otras tres vistas de usuarios según el tipo que sea seleccionado.	

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: U-GENEST-06
Nombre Tarea: Diseño e implementación de la primera sección de la vista de usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 15-01-2009	Fecha Fin: 17-01-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Diseño e implementación de la primera sección de la vista de usuarios.	
Observación: Esta es la vista que saldrá por defecto en la vista principal de usuarios.	

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: U-GENEST-06
Nombre Tarea: Controlador para la segunda sección de la vista de usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 19-01-2009	Fecha Fin: 21-01-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Creación de un fichero que sea el encargado de seleccionar los datos que serán mostrados por la gráfica de la segunda sección de la vista de usuarios. Reporte de las 5 aplicaciones mas usadas por un usuario específico , en un intervalo de tiempo determinado y de un grupo de aplicaciones previamente seleccionado.	
Observación:	

Tarea de Ingeniería	
Número Tarea: 6	Número Historia de Usuario: U-GENEST-06
Nombre Tarea: Diseño e implementación de la segunda sección de la vista de usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 22-01-2009	Fecha Fin: 24-01-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Diseño e implementación de la segunda sección de la vista de usuarios.	
Observación:	

Tarea de Ingeniería	
Número Tarea:7	Número Historia de Usuario: U-GENEST-06
Nombre Tarea: Controlador para la tercera sección de la vista de usuarios.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 26-01-2009	Fecha Fin: 28-01-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Creación de un fichero que sea el encargado de seleccionar los datos que serán mostrados por la gráfica de la tercera sección de la vista de usuario. Las 5 rutas mas usadas por un usuario específico , en un intervalo de tiempo determinado y de una aplicación previamente seleccionada.	
Observación:	

Tarea de Ingeniería	
Número Tarea:8	Número Historia de Usuario: U-GENEST-06
Nombre Tarea: Diseño e implementación de la tercera sección de la vista de usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 29-01-2009	Fecha Fin: 31-01-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Diseño e implementación de la tercera sección de la vista de usuarios.	
Observación:	

Tareas de Ingeniería para la historia de usuario U-GENEST-07

Tarea de Ingeniería	
Número Tarea:1	Número Historia de Usuario: U-GENEST-07
Nombre Tarea: Diseño e implementación de la vista de principal de proyectos .	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 02-02-2009	Fecha Fin: 04-02-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Diseño e implementación de la vista principal de proyectos.	
Observación: Esta vista contendrá a las otras dos vistas de proyectos según la sección seleccionada.	

Tarea de Ingeniería	
Número Tarea:2	Número Historia de Usuario: U-GENEST-07
Nombre Tarea: Controlador para la primera sección de la vista de proyectos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 05-02-2009	Fecha Fin: 07-02-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Creación de un fichero que será el encargado de seleccionar los datos que serán mostrados por la gráfica de la primera sección de la vista de proyectos. Son reportes del tiempo de utilización de cada grupo de aplicaciones por cada uno de los miembros del proyecto del que es dueño el usuario que este autenticado. Además permite seleccionar en el intervalo de tiempo que se medirá para dicho reporte.	
Observación:	

Tarea de Ingeniería	
Número Tarea:3	Número Historia de Usuario: U-GENEST-07
Nombre Tarea: Diseño e implementación de la primera sección de la vista por proyectos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 09-02-2009	Fecha Fin: 11-02-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Diseño e implementación de la primera sección de la vista por proyectos.	
Observación: Esta es la vista que saldrá por defecto en la vista principal de proyectos.	

Tarea de Ingeniería	
Número Tarea:4	Número Historia de Usuario: U-GENEST-07
Nombre Tarea: Controlador para la segunda sección de la vista de proyectos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 12-02-2009	Fecha Fin: 14-02-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Creación de un fichero que sea el encargado de seleccionar los datos que serán mostrados por la gráfica de la segunda sección de la vista de proyectos. Son reportes del tiempo de utilización de cada grupo de aplicaciones por cada uno de los proyectos existentes en la empresa.	
Observación:	

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: U-GENEST-07
Nombre Tarea: Diseño e implementación de la segunda sección de la vista de proyectos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 16-02-2009	Fecha Fin: 18-02-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Diseño e implementación de la segunda sección de la vista de proyectos.	
Observación:	

Tareas de Ingeniería para la historia de usuario U-GENEST-08

Tarea de Ingeniería	
Número Tarea:1	Número Historia de Usuario: U-GENEST-08
Nombre Tarea: Diseño e implementación de la vista de reportes en formato PDF.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 19-02-2009	Fecha Fin: 21-02-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Diseño e implementación de la vista principal de proyectos. Se muestran en una tabla los diferentes reportes que se pueden realizar. Estos reportes contienen parte de la información existente en las otras vistas.	
Observación:	

Tarea de Ingeniería

Número Tarea: 2	Número Historia de Usuario: U-GENEST-08
Nombre Tarea: Crear reporte general por usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 23-02-2009	Fecha Fin: 25-02-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Tiempo total de utilización de los grupos de aplicaciones para cada usuario	
Observación:	

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: U-GENEST-08
Nombre Tarea: Crear reporte general de proyectos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 26-02-2009	Fecha Fin: 28-02-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Tiempo total de utilización de los grupos de aplicaciones para cada proyecto.	
Observación:	

Tareas de Ingeniería para la historia de usuario U-GENEST-09

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: U-GENEST-09
Nombre Tarea: Diseño e implementación de la vista de ejecución del CRON.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 02-03-2009	Fecha Fin: 04-03-2009
Programador Responsable: Karel Tito Catá.	
Descripción: Se muestran en una tabla los estados de las diferentes acciones que va ejecutando el CRON. Se informa al usuario el estado de lectura de los reportes, y los posibles errores en caso de que ocurran.	
Observación:	

La implementación del módulo GENEST se desarrolló satisfactoriamente a partir del diseño planteado en el capítulo anterior. Se cumplieron con las exigencias desde el punto de vista del diseño de los infográficos que lo conforman y con los estándares exigidos por el framework. Solo basta realizarle las pruebas de aceptación pertinentes para validar su correcta implementación.

Capítulo 4. Validación de la solución propuesta

Los casos de prueba de aceptación tienen como objetivo validar que el sistema cumpla con el funcionamiento esperado y permitir al cliente determinar su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

Casos de Pruebas

Las pruebas de aceptación son definidas por el cliente y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al cliente. La utilización de estas, proporcionan grandes ventajas, permitiendo a los programadores principalmente medir la calidad de su trabajo y garantizar la entrega de un producto con calidad y en correspondencia con las necesidades del cliente.

Caso de Prueba de la historia de usuario U-GENEST-01

Caso de Prueba de Aceptación	
Código Caso de Prueba: U-GENEST-01-01	Nombre Historia de Usuario: U-GENEST-01
Nombre de la persona que realiza la prueba: Karel Tito Catá	
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: U-GENEST-01-02	Nombre Historia de Usuario: U-GENEST-01
Nombre de la persona que realiza la prueba: Karel Tito Catá	
Descripción de la Prueba: Debe leer y mostrar todos los ficheros y existentes dentro de un directorio específico.	
Condiciones de Ejecución: Se especifica el directorio a leer .	
Entrada / Pasos de ejecución:	
Resultado Esperado: En caso de existir ficheros mostrar los nombres de los mismos, en caso contrario mostrar en mensaje informando la no existencia de ficheros en la dirección especificada.	
Evaluación de la Prueba: Prueba satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba: U-GENEST-01-03	Nombre Historia de Usuario: U-GENEST-01

Nombre de la persona que realiza la prueba: Karel Tito Catá
Descripción de la Prueba: Debe eliminar los directorios existentes en una dirección específica y volverlos a crear a partir de un arreglo de nombres, y mostrar todos los ficheros y existentes dentro de un 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.
Evaluación de la Prueba: Prueba satisfactoria

Caso de Prueba de la historia de usuario U-GENEST-03

Caso de Prueba de Aceptación	
Código Caso de Prueba: U-GENEST-03-01	Nombre Historia de Usuario: U-GENEST-03
Nombre de la persona que realiza la prueba: Karel Tito Catá	
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 sistema operativo. 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 , guardado su contenido en la base de datos y después eliminados.	
Evaluación de la Prueba: Prueba satisfactoria	

Caso de Prueba de la historia de usuario U-GENEST-04

Caso de Prueba de Aceptación	
Código Caso de Prueba: U-GENEST-04-01	Nombre Historia de Usuario: U-GENEST-04
Nombre de la persona que realiza la prueba: Karel Tito Catá	
Descripción de la Prueba: Instala el módulo y establece una configuración inicial de sus variables.	
Condiciones de Ejecución: Debe estar autenticado con los permisos suficientes para acceder al módulo de administración del sistema.	
Entrada / Pasos de ejecución: Se accede al módulo de administración del sistema en la vista :”Ver módulos”. Se selecciona el módulo , se instala y se activa .	
Resultado Esperado: El módulo es instalado satisfactoriamente y es guardada en la base de datos su configuración .	
Evaluación de la Prueba: Prueba satisfactoria	

Caso de Prueba de la historia de usuario U-GENEST-05

Caso de Prueba de Aceptación	
Código Caso de Prueba: U-GENEST-05-01	Nombre Historia de Usuario: U-GENEST-05
Nombre de la persona que realiza la prueba: Karel Tito Catá	
Descripción de la Prueba: Selecciona reportes en un rango de tiempo y los muestra clasificados por Grupos de Aplicaciones.	
Condiciones de Ejecución: Estar autenticado y con permisos suficientes para poder acceder al módulo.	
Entrada / Pasos de ejecución: Se accede a la vista 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 que no existen reportes para los filtros señalados.	
Evaluación de la Prueba: Prueba satisfactoria	

Caso de Prueba de la historia de usuario U-GENEST-06

Caso de Prueba de Aceptación	
Código Caso de Prueba: U-GENEST-06-01	Nombre Historia de Usuario: U-GENEST-06
Nombre de la persona que realiza la prueba: Karel Tito Catá	
Descripción de la Prueba: Primera sección: seleccionar reportes en un rango de tiempo y los muestra para cada uno de los usuarios de la empresa clasificados por grupos de aplicaciones.	
Condiciones de Ejecución: Estar autenticado y con permisos suficientes para poder acceder al módulo.	
Entrada / Pasos de ejecución: Se accede a la vista de usuarios y se escoge la primera sección de la vista . Se selecciona un intervalo de tiempo para posteriormente solicitar el reporte.	
Resultado Esperado: Se muestra en una gráfica de tipo barra (en grupos) los resultados del reporte y de forma paginada, en caso contrario informar que no existen reportes para los filtros señalados..	
Evaluación de la Prueba: Prueba satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba: U-GENEST-06-02	Nombre Historia de Usuario: U-GENEST-06
Nombre de la persona que realiza la prueba: Karel Tito Catá	
Descripción de la Prueba: Segunda sección: seleccionar reportes de las 5 aplicaciones más usadas de un grupo de aplicaciones por un usuario específico en un rango de tiempo.	
Condiciones de Ejecución: Estar autenticado y con permisos suficientes para poder acceder al módulo.	
Entrada / Pasos de ejecución: Se accede a la vista de usuarios y se escoge la segunda sección de la vista. Se selecciona un intervalo de tiempo , el usuario y el grupo de aplicaciones para posteriormente solicitar el reporte.	
Resultado Esperado: Se muestra en una gráfica de tipo pastel los resultados del reporte, en caso contrario informar que no existen reportes para los filtros señalados.	
Evaluación de la Prueba: Prueba satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba: U-GENEST-06-03	Nombre Historia de Usuario: U-GENEST-06
Nombre de la persona que realiza la prueba: Karel Tito Catá	
Descripción de la Prueba: Tercera sección: seleccionar reportes de las 5 rutas más accedidas de una aplicación por un usuario específico en un rango de tiempo .	
Condiciones de Ejecución: Estar autenticado y con permisos suficientes para poder acceder al módulo.	
Entrada / Pasos de ejecución: Se accede a la vista de usuarios y se escoge la tercera sección de la vista. Se selecciona un intervalo de tiempo , el usuario y la aplicación para posteriormente solicitar el reporte.	
Resultado Esperado: Se muestra en una gráfica de tipo pastel los resultados del reporte, en caso contrario informar que no existen reportes para los filtros señalados.	
Evaluación de la Prueba: Prueba satisfactoria	

Caso de Prueba de la historia de usuario U-GENEST-07

Caso de Prueba de Aceptación	
Código Caso de Prueba: U-GENEST-07-01	Nombre Historia de Usuario: U-GENEST-07
Nombre de la persona que realiza la prueba: Karel Tito Catá	
Descripción de la Prueba: Primera sección: seleccionar reportes en un rango de tiempo y clasificados por grupos de aplicaciones. Los muestra para cada uno de los usuarios pertenecientes al proyecto del que es dueño el usuario que está autenticado.	
Condiciones de Ejecución: Estar autenticado y con permisos suficientes para poder acceder al módulo y ser dueño de algún proyecto.	
Entrada / Pasos de ejecución: Se accede a la vista de proyectos y se escoge la primera sección de la vista . Se selecciona un intervalo de tiempo para posteriormente solicitar el reporte.	
Resultado Esperado: Se muestra en una gráfica de tipo barra (en grupos) los resultados del reporte y de forma paginada, en caso contrario informar que no existen reportes para los filtros señalados.	
Evaluación de la Prueba: Prueba satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba: U-GENEST-07-02	Nombre Historia de Usuario: U-GENEST-07
Nombre de la persona que realiza la prueba: Karel Tito Catá	
Descripción de la Prueba: Segunda sección: seleccionar reportes en un rango de tiempo y clasificados por grupos de aplicaciones. Los muestra para cada uno de los proyectos existentes en la empresa.	
Condiciones de Ejecución: Estar autenticado y con permisos suficientes para poder acceder al módulo y ser dueño de algún proyecto.	
Entrada / Pasos de ejecución: Se accede a la vista de proyectos y se escoge la segunda sección de la vista. Se selecciona un intervalo de tiempo para posteriormente solicitar el reporte.	
Resultado Esperado: Se muestra en una gráfica de tipo barra (en grupos) los resultados del reporte y de forma paginada, en caso contrario informar que no existen reportes para los filtros señalados.	
Evaluación de la Prueba: Prueba satisfactoria	

Caso de Prueba de la historia de usuario U-GENEST-08

Caso de Prueba de Aceptación	
Código Caso de Prueba: U-GENEST-08-01	Nombre Historia de Usuario: U-GENEST-08
Nombre de la persona que realiza la prueba: Karel Tito Catá	
Descripción de la Prueba: Seleccionar reportes en formato PDF con información existentes en el resto de las vistas.	
Condiciones de Ejecución: Estar autenticado y con permisos suficientes para poder acceder al módulo .	
Entrada / Pasos de ejecución: Se accede a la vista de reportes en formato 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	

Caso de Prueba de la historia de usuario U-GENEST-09

Caso de Prueba de Aceptación	
Código Caso de Prueba: U-GENEST-09-01	Nombre Historia de Usuario: U-GENEST-09
Nombre de la persona que realiza la prueba: Karel Tito Catá	
Descripción de la Prueba: Ejecutar el CRON de forma manual y visualizar los logs que va dejando durante su ejecución.	
Condiciones de Ejecución: Estar autenticado y con permisos suficientes para poder acceder al módulo .	
Entrada / Pasos de ejecución: Se accede a la vista de ejecución manual del CRON y se ejecuta la orden de “Analizar reportes”.	
Resultado Esperado: Se muestran los logs de la ejecución del CRON.	
Evaluación de la Prueba: Prueba satisfactoria	

Resultados Obtenidos.

Como resultado de este trabajo GENEST queda disponible en su versión 0.1. Se obtuvo un módulo que cumple con todas las especificaciones de dotProject desde el punto de vista de diseño. Se logró un alto grado de optimización utilizándose los componentes propios del framework dotProject logrando así una reutilización de código a gran escala.

Funcionalidades obtenidas.

Vista	Descripción
General	Muestra una gráfica con el porcentaje de uso de los distintos grupos de aplicaciones por todos los miembros de la empresa.
Usuarios	<p>Esta vista da acceso a tres secciones más dentro de ella .</p> <ul style="list-style-type: none"> ➤ En la primera se muestra una gráfica con el porcentaje del uso de los distintos grupos de aplicaciones de cada usuario de la empresa. ➤ En la segunda se muestra una gráfica con el porcentaje del uso de las 5 aplicaciones más usadas de un grupo de aplicaciones específico. ➤ En la tercera se muestra una gráfica con el porcentaje de uso de las 5 rutas más accedidas de una aplicación específica.
Proyectos	<p>Esta vista da acceso a dos secciones más dentro de ella :</p> <ul style="list-style-type: none"> ➤ La primera muestra una gráfica con el porcentaje del uso de los grupos de aplicaciones por cada una de los miembros del proyecto que es dueño el usuario que esté autenticado. ➤ La segunda muestra una gráfica de barras con el porcentaje del uso de los grupos de aplicaciones de los proyectos existentes en la empresa.
Reportes en formato PDF	Muestra la lista de reportes que pueden realizarse. Estos reportes contienen casi la totalidad de la información existentes en el resto de las vistas del módulo.
Cron	Posibilita la ejecución de forma manual del script que lee los reportes . Esta vista muestra los logs de todos los procesos que se realizan para leer los datos de los reportes

Conclusiones

Podemos observar que la Gestión de Proyectos siempre ha estado en continua evolución y actualmente está más activa que nunca, teniendo por delante un futuro prometedor donde la innovación jugará un papel decisivo. En el presente trabajo se expusieron elementos de gran importancia relacionados con la gestión de proyectos y los SGP como herramientas web capaces de automatizar los diversos procesos que la forman. Se estudiaron los diferentes tipos de infográficos existentes para ser utilizados en la implementación del módulo GENEST en la muestra del enorme volumen de información que este manipulará. Se diseñó el módulo haciendo uso del estilo arquitectónico MVC, a partir del cual se realizó el desarrollo del módulo con la calidad requerida; que fue comprobada con las distintas pruebas de aceptación a las que se sometió el módulo.

Recomendaciones

- Que se le implementen nuevas funcionalidades al sistema.
- Que el sistema sea desplegado en otros proyectos para mejorar el proceso de planificación en los mismos.
- Que se establezcan cláusulas para regular su uso.

Referencias Bibliográficas

- ActiveCollab. activeCollab - Project Management and Collaboration Tool. 2009. [cited 13 May 2009]. Available from world wide web: <<http://www.activecollab.com/>>.
- Alejandro Barros . Sistemas Colaborativos de gestión de proyectos. 2009. [cited 13 May 2009]. Available from world wide web: <<http://www.alejandrobarrros.cl/content/view/134919/Sistemas-Colaborativos-de-gestion-de-proyectos.html>>.
- Angel Nieva López. Innovación en la Gestión de Proyectos. Una historia en evolución - Noticias.com. 2009. [cited 13 May 2009]. Available from world wide web: <<http://www.noticias.com/opinion/innovacion-gestion-proyectos-historia-evolucion-9h8.html>>.
- Arleco. GanttProject: una buena alternativa de programas de proyectos - AplicacionesEmpresariales.com. 2008a. [cited 13 May 2009]. Available from world wide web: <<http://www.aplicacionesempresariales.com/ganttproject-una-buena-alternativa-de-programas-de-proyectos.html>>.
- Arleco. MS Project, el pionero de los software de proyectos - AplicacionesEmpresariales.com. 2008b. [cited 14 May 2009]. Available from world wide web: <<http://www.aplicacionesempresariales.com/ms-project-el-pionero-de-los-software-de-proyectos.html>>.
- arqhys.com. HENRY GANTT. 2009. [cited 18 May 2009]. Available from world wide web: <<http://www.arqhys.com/articulos/gantt-henry.html>>.
- Arq. Percy C. Acuña Vigil. Gestión de proyectos de desarrollo urbano | Urbano Perú: Perullaqta, El Blog. 2009. [cited 13 May 2009]. Available from world wide web: <<http://www.urbanoperu.com/node/360>>.
- Daniel M. Maldonado. El CoDiGo K » Quanta Plus, la alternativa perfecta para desarrollas nuestras aplicaciones Web. 2008. [cited 15 May 2009]. Available from world wide web: <<http://www.elcodigok.com.ar/2008/04/quanta-plus-la-alternativa-perfecta-para-desarrollas-nuestras-aplicaciones-web/>>.
- Elio Leturia. Leturia, Elio (1998): ¿Qué es infografía? - Revista Latina de Comunicación Social, 4. 2009. [cited 13 May 2009]. Available from world wide web: <<http://www.ull.es/publicaciones/latina/z8/r4el.htm>>.
- Ernesto Serrano. dotProject para planificación de proyectos | Abartia Team. 2009. [cited 13 May 2009]. Available from world wide web: <<http://www.abartiateam.com/dotproject>>.
- Gladys M. Peñalver Romero. MA-GMPR-UR2. Metodología ágil para proyectos de software libre. June 2008.
- IES Gaviota. El lenguaje HTML. 2009. [cited 13 May 2009]. Available from world wide web: <<http://www.juntadeandalucia.es/averroes/iesgaviota/informatica/html.html>>.
- jpGraph. JpGraph - PHP Graph Creating Library. 2009. [cited 14 May 2009]. Available from world wide web: <<http://www.aditus.nu/jpgraph/>>.
- linux.ciberaula.com. Una Introducción a Apache. 2009. [cited 13 May 2009]. Available from world wide web: <http://linux.ciberaula.com/articulo/linux_apache_intro/>.
- Mario Torres. PMBOK: Guía de los Fundamentos de la Dirección de Proyectos | ConstrucGeek. 2008. [cited 18 May 2009]. Available from world wide web: <<http://www.construcgeek.com/blog/pmbok-guia-de-los-fundamentos-de-la-direccion-de-proyectos>>.

- Miguel Angel Alvarez. Qué es Javascript. 2009. [cited 13 May 2009]. Available from world wide web: <<http://www.desarrolloweb.com/articulos/25.php>>.
- mysql-hispano.org. MySQL Hispano - La comunidad de usuarios de MySQL. 2008. [cited 25 May 2009]. Available from world wide web: <<http://www.mysql-hispano.org/index.php?m=read&id=147>>.
- Netcraft. March 2009 Web Server Survey - Netcraft. 2009. [cited 14 May 2009]. Available from world wide web: <http://news.netcraft.com/archives/2009/03/15/march_2009_web_server_survey.html>.
- netpecos.org. MySQL. 2009. [cited 13 May 2009]. Available from world wide web: <http://www.netpecos.org/docs/mysql_postgres/x57.html>.
- overLIB. overLIB - Homepage. 2009. [cited 14 May 2009]. Available from world wide web: <<http://www.bosrup.com/web/overlib/>>.
- php.net. PHP: Hypertext Preprocessor. 2009. [cited 15 May 2009]. Available from world wide web: <<http://www.php.net/>>.
- Planner Team. Software Planner: Award winning Application Lifecycle Management (ALM). 2009. [cited 13 May 2009]. Available from world wide web: <<http://www.softwareplanner.com/>>.
- Roberto Bárcenas. Biblioteca ADOdb para PHP. 2009. [cited 15 May 2009]. Available from world wide web: <<http://www.desarrolloweb.com/articulos/1042.php>>.
- R&OS. R&OS: pdf class. 2009. [cited 14 May 2009]. Available from world wide web: <<http://www.ros.co.nz/pdf/>>.
- Salva Castro. OpenProj 1.01, un MS Project open source que promete. 2008. [cited 13 May 2009]. Available from world wide web: <<http://www.genbeta.com/herramientas/openproj-101-un-ms-project-open-source-que-promete>>.
- standards.ieee.org. IEEE Standards Description: 1490-2003. 2009. [cited 14 May 2009]. Available from world wide web: <http://standards.ieee.org/reading/ieee/std_public/description/se/1490-2003_desc.html>.
- Symfony, la guía definitiva. El patrón MVC | Symfony 1.1, la guía definitiva | LibrosWeb.es. 2009. [cited 18 May 2009]. Available from world wide web: <http://www.librosweb.es/symfony_1_1/capitulo2/el_patron_mvc.html>.
- w3c. Guía Breve de CSS. 2009a. [cited 13 May 2009]. Available from world wide web: <<http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>>.
- w3c. Guía Breve de Tecnologías XML. 2009b. [cited 13 May 2009]. Available from world wide web: <<http://www.w3c.es/divulgacion/guiasbreves/tecnologiasXML>>.
- www.um.es. Apuntes. Ingeniería del software. Sistemas Informaticos. Nivel de madurez software. Informatica Aplicada a la Gestión Pública. 2005/06-2. Universidad de Murcia. Rafael Barzanallana. 2009. [cited 14 May 2009]. Available from world wide web: <<http://www.um.es/docencia/barzana/IAGP/Iagp2.html>>.

Bibliografía

- Mario Torres. PMBOK: Guía de los Fundamentos de la Dirección de Proyectos | ConstrucGeek. 2008. [cited 18 May 2009]. Available from world wide web: <<http://www.construcgeek.com/blog/pmbok-guia-de-los-fundamentos-de-la-direccion-de-proyectos>>.
- R&OS. R&OS: pdf class. 2009. [cited 14 May 2009]. Available from world wide web: <<http://www.ros.co.nz/pdf/>>.
- Symfony, la guía definitiva. El patrón MVC | Symfony 1.1, la guía definitiva | LibrosWeb.es. 2009. [cited 18 May 2009]. Available from world wide web: <http://www.librosweb.es/symfony_1_1/capitulo2/el_patron_mvc.html>.
- php.net. PHP: Hypertext Preprocessor. 2009. [cited 15 May 2009]. Available from world wide web: <<http://www.php.net/>>.
- w3c. Guía Breve de Tecnologías XML. 2009b. [cited 13 May 2009]. Available from world wide web: <<http://www.w3c.es/divulgacion/guiasbreves/tecnologiasXML>>.
- overLIB. overLIB - Homepage. 2009. [cited 14 May 2009]. Available from world wide web: <<http://www.bosrup.com/web/overlib/>>.
- jpGraph. JpGraph - PHP Graph Creating Library. 2009. [cited 14 May 2009]. Available from world wide web: <<http://www.aditus.nu/jpgraph/>>.
- Gladys M. Peñalver Romero. MA-GMPR-UR2. Metodología ágil para proyectos de software libre. June 2008.
- Elio Leturia. Leturia, Elio (1998): ¿Qué es infografía? - Revista Latina de Comunicación Social, 4. 2009. [cited 13 May 2009]. Available from world wide web: <<http://www.ull.es/publicaciones/latina/z8/r4el.htm>>.

Glosario de términos

API: interfaz de programación de aplicaciones o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Artefactos: Un artefacto es un producto tangible resultante del proceso de desarrollo de software. Algunos artefactos como los casos de uso, diagrama de clases u otros modelos UML ayudan a la descripción de la función, la arquitectura o el diseño del software.

ASCII: El código ASCII (acrónimo inglés de American Standard Code for Information Interchange) pronunciado generalmente [áski], es un código de caracteres basado en el alfabeto latino tal como se usa en inglés moderno y en otras lenguas occidentales.

Blog: Un blog, o en español también una bitácora, es un sitio web periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente.

Cookies: Una cookie (pronunciado ['ku.ki]; literalmente galleta) es un fragmento de información que se almacena en el disco duro del visitante de una página web a través de su navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en posteriores visitas.

Columns: es un conjunto de valores de datos de un simple tipo particular, uno por cada fila de la tabla. Las columnas proporcionan la estructura según la cual se componen las filas. El término campo es frecuentemente intercambiable con el de columna, aunque muchos consideran más correcto usar el término campo (o valor de campo) para referirse específicamente al simple elemento que existe en la intersección entre una fila y una columna.

Demonio: Un demonio, daemon o daemon (de sus siglas en inglés Disk And Execution MONitor), es un tipo especial de proceso informático que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario (es un proceso no interactivo). Este tipo de programas se ejecutan de forma continua (infinita), vale decir, que aunque se intente cerrar o matar el proceso, este continuará en ejecución o se reiniciará automáticamente. Todo esto sin intervención de terceros y sin dependencia de consola alguna.

Editor de textos: Un editor de texto es un programa que permite crear y modificar archivos digitales compuestos únicamente por texto sin formato, conocidos comúnmente como archivos de texto o texto plano. El programa lee el archivo e interpreta los bytes leídos según el código de caracteres que usa el editor.

FTP: Sigla en inglés de File Transfer Protocol (Protocolo de Transferencia de Archivos) en informática, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor.

Gnome: es un entorno de escritorio e infraestructura de desarrollo para sistemas operativos Unix/Linux, compuesto enteramente de software libre.

Forma parte oficial del proyecto GNU. Nació como una alternativa a KDE bajo el nombre de GNU Network Object Model Environment.

GNU/Linux: es el término empleado para referirse al sistema operativo similar a Unix que utiliza como base las herramientas de sistema de GNU y el núcleo Linux.

GPL: 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 a mediados de los 80, y está 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.

InnoDB : InnoDB es una tecnología de almacenamiento de datos de fuente abierta para la base de datos MySQL, incluido como formato de tabla estándar en todas las distribuciones de MySQL AB a partir de las versiones 4.0. Su característica principal es que soporta transacciones de tipo ACID y bloqueo de registros e integridad referencial.

Java: Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

KDE: es un entorno de Escritorio contemporáneo para estaciones de trabajo Unix. KDE llena la necesidad de un escritorio amigable para estaciones de trabajo Unix, similar a los escritorios de MacOSX o Windows.

KHTML: es el motor de renderizado HTML libre desarrollado para el proyecto KDE. Fue creado para el navegador web de KDE, Konqueror.

KIO : acrónimo del inglés KDE Input/Output, es parte de la arquitectura del entorno KDE . Permite acceso a archivos, sitios web y a otras fuentes con un API simple y consistente.

Kparts: En programación, KParts es el nombre del componente framework para el entorno de escritorio KDE. A un componente individual se le denomina una KPart. Los KPart son análogos a los componentes Bonobo en GNOME.

LDAP: Lightweight Directory Access Protocol(Protocolo Ligero de Acceso a Directorios) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

librería: En ciencias de la computación, una biblioteca (o librería) es un conjunto de subprogramas utilizados para desarrollar software. Las bibliotecas contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de éstos.

Licencia de software: Una licencia de software es un contrato entre el titular del derecho de autor (propietario) y el usuario del programa informático (usuario final), para utilizar éste en una forma determinada y de conformidad con unas condiciones convenidas.

Log: Un log es un registro oficial de eventos durante un periodo de tiempo en particular. Para los profesionales en seguridad informática es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué (who, what, when, where y why, W5) un evento ocurre para un dispositivo en particular o aplicación.

Paquete: Un paquete de software es una serie de programas que se distribuyen conjuntamente. Algunas de las razones suelen ser que el funcionamiento de cada uno complementa a o requiere de otros, además de que sus objetivos están relacionados como estrategia de mercadotecnia.

PC : ordenador personal (en inglés, Personal Computer) es el término genérico utilizado para referirse a microcomputadoras que son compatibles con las especificaciones de IBM.

PDF:(acrónimo del inglés Portable Document Format, formato de documento portátil) es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems. Este formato es de tipo compuesto (imagen vectorial, mapa de bits y texto).

Pear: PHP Extension and Application Repository, es un entorno de desarrollo y sistema de distribución para componentes de código PHP.

PMBOK: es una colección de procesos y áreas de conocimiento generalmente aceptadas como las mejores prácticas

dentro de la gestión de proyectos. El PMBOK es un estándar reconocido internacionalmente (IEEE Std 1490-2003) que provee los fundamentos de la gestión de proyectos que son aplicables a un amplio rango de proyectos.

PMI: Project Management Institute. Está actualmente considerado la asociación profesional para la gestión de proyectos sin fines de lucro más grande del mundo, formada por más 260.000 miembros alrededor de 171 países.

PostgreSQL: PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD.

Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG .

Script: es un guión o conjunto de instrucciones. Permiten la automatización de tareas creando pequeñas utilidades.

Socket: designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiarse cualquier flujo de datos, generalmente de manera fiable y ordenada.

Un socket queda definido por una dirección IP, un protocolo y un número de puerto.

SSH:(Secure SHell, en español: intérprete de órdenes seguro) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos si tenemos un Servidor X (en sistemas Unix) corriendo.

TIC : Tecnologías de la información y las Comunicaciones.

UNIX: Unix (registrado oficialmente como UNIX®) es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T,

WEB 2.0: El término, Web 2.0 fue acuñado por Tim O'Reilly en 2004 para referirse a una segunda generación en la historia de la Web basada en comunidades de usuarios y una gama especial de servicios, como las redes sociales, los blogs y los wikis.

Wiki : Un wiki, o una wiki, es un sitio web cuyas páginas web pueden ser editadas por múltiples voluntarios a través del navegador web.

WWW : World Wide Web o Red Global Mundial es un sistema de documentos de hipertexto y/o hipermedios enlazados y accesibles a través de Internet.

Anexos

Anexo #1: Prototipo de la Historia de Usuario visualizar reporte general.

dotProject 2.1-rc2 dotProject.net
FREE SOFTWARE

Empresas | Proyectos | Tareas | Calendario | Ficheros | Contactos | Foros | Tickets | Usuarios | Sistema | Ayuda | Reportes | Inventario - Nuevo Elemento -

Bienvenido Karelito Postmar Ayuda | Mis datos | **A realizar** | Hoy | Salir

Estadísticas

con pestañas : plano

Vista general | Vista de usuario | Vista de proyecto | Reportes en formato PDF | Ejecutar el cron

Seleccione un intervalo

Fecha de Inicio:

Fecha de Finalización:

Vista general

% de uso de los grupos de aplicaciones

Grupo de Aplicaciones	Porcentaje
jabber	34%
ide	23%
ofimatica	15%
otros	26%
web	2%

<http://localhost/proyecto/dotproject-26-01-09/index.php?m=report#>

Anexo #2: Prototipo de la Historia de Usuario visualizar reportes de usuarios.

dotProject 2.1-rc2 dotProject.net
FREE SOFTWARE

Empresas | Proyectos | Tareas | Calendario | Ficheros | Contactos | Foros | Tickets | Usuarios | Sistema | Ayuda | Reportes | Inventario | jaiien - Nuevo Elemento - ▾

Bienvenido Karelito Postmar Ayuda | Mis datos | **A realizar** | Hoy | Salir

Estadísticas

con pestañas : plano

Vista general | **Vista de usuario** | Vista de proyecto | Reportes en formato PDF | Ejecutar el CRON

Opciones

Seleccione un intervalo

Fecha de Inicio: 26/04/2009

Fecha de Finalización: 26/04/2009

Enviar

Todos los usuarios

Statistics of users for applications group

Aplicación	Usuarios
Web	18
Jabber	15
IDE	30
Ofimática	38
Otros	0

grserrano

Página [10] de [10]

Anexo #3: Prototipo de la Historia de Usuario visualizar reportes de usuarios.

dotProject 2.1-rc2 dotProject.net
FREE SOFTWARE

Empresas | Proyectos | Tareas | Calendario | Ficheros | Contactos | Foros | Tickets | Usuarios | Sistema | Ayuda | Reportes | Inventario | Jailen - Nuevo Elemento -

Bienvenido Karelito Postmar Ayuda | Mis datos | **A realizar** | Hoy | Salir

Estadísticas

con pestañas : plano

Vista general | Vista de usuario | Vista de proyecto | Reportes en formato PDF | Ejecutar el CRON

Opciones

Seleccione un intervalo

Fecha de Inicio: 01/03/2009

Fecha de Finalización: 26/04/2009

Group: OFFIMATIC

Nombre de usuario: yrserrano

Enviar

Estadísticas por aplicaciones

% de uso de los grupos de aplicaciones

Aplicación	Porcentaje
gnnumeric	65%
openoffice	35%

Anexo #4: Prototipo de la Historia de Usuario visualizar reportes de usuarios.



Estadísticas

con pestañas : plano

Vista general | Vista de usuario | Vista de proyecto | Reportes en formato PDF | Ejecutar el CRON

Opciones

Seleccione un intervalo

Fecha de Inicio

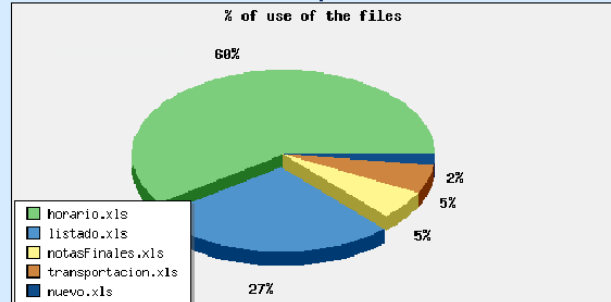
Fecha de Finalización

Aplicación

Nombre de usuario



Estadísticas por ficheros



Anexo #5: Prototipo de la Historia de Usuario Visualizar reportes de proyectos.

dotProject 2.1-rc2 dotProject.net
FREE SOFTWARE

Empresas | Proyectos | Tareas | Calendario | Ficheros | Contactos | Foros | Tickets | Usuarios | Sistema | Ayuda | Reportes | Inventario - Nuevo Elemento -

Bienvenido Karelito Postmar Ayuda | Mis datos | **A realizar** | Hoy | Salir

Estadísticas

con pestañas : plano

Vista general | Vista de usuario | **Vista de proyecto** | Reportes en formato PDF | Ejecutar el cron

Opciones

Seleccione un intervalo

Fecha de Inicio: 02/02/2009

Fecha de Finalización: 20/03/2009

Mi equipo de proyecto

Statistics of users for applications group

Group	Web	Jabber	IDE	Ofimatica	Otros
ktito	2	18	24	37	22
rpelegrino	2	28	28	18	28

Pagina [1] de [1]

Done

Anexo #6: Prototipo de la Historia de Usuario Visualizar reportes de proyectos.

dotProject 2.1-rc2 dotProject.net
FREE SOFTWARE

Empresas | Proyectos | Tareas | Calendario | Ficheros | Contactos | Foros | Tickets | Usuarios | Sistema | Ayuda | Reportes | Inventario - Nuevo Elemento -

Bienvenido Karelito Postmar Ayuda | Mis datos | **A realizar** | Hoy | Salir

Estadísticas

con pestañas : plano

Vista general | **Vista de usuario** | Vista de proyecto | Reportes en formato PDF | Ejecutar el cron

Opciones

Seleccione un intervalo

Fecha de Inicio: 02/02/2009

Fecha de Finalización: 20/03/2009

Todos los proyectos

Statistics of users for applications group

Group	Web	Jabber	IDE	Ofimatica	Otros
Arquitectura	2	26	28	21	27
Prensa	2	30	30	17	23

Pagina [1] de [1]

Done

Anexo #7: Prototipo de la Historia de Usuario Visualizar reportes en formato PDF.

dotProject 2.1-rc2 dotProject.net
FREE SOFTWARE



Empresas | Proyectos | Tareas | Calendario | Ficheros | Contactos | Foros | Tickets | Usuarios | Sistema | Ayuda | Reportes | Inventario - Nuevo Elemento -

Bienvenido Karelito Postmar Ayuda | Mis datos | **A realizar** | Hoy | Salir

Estadísticas

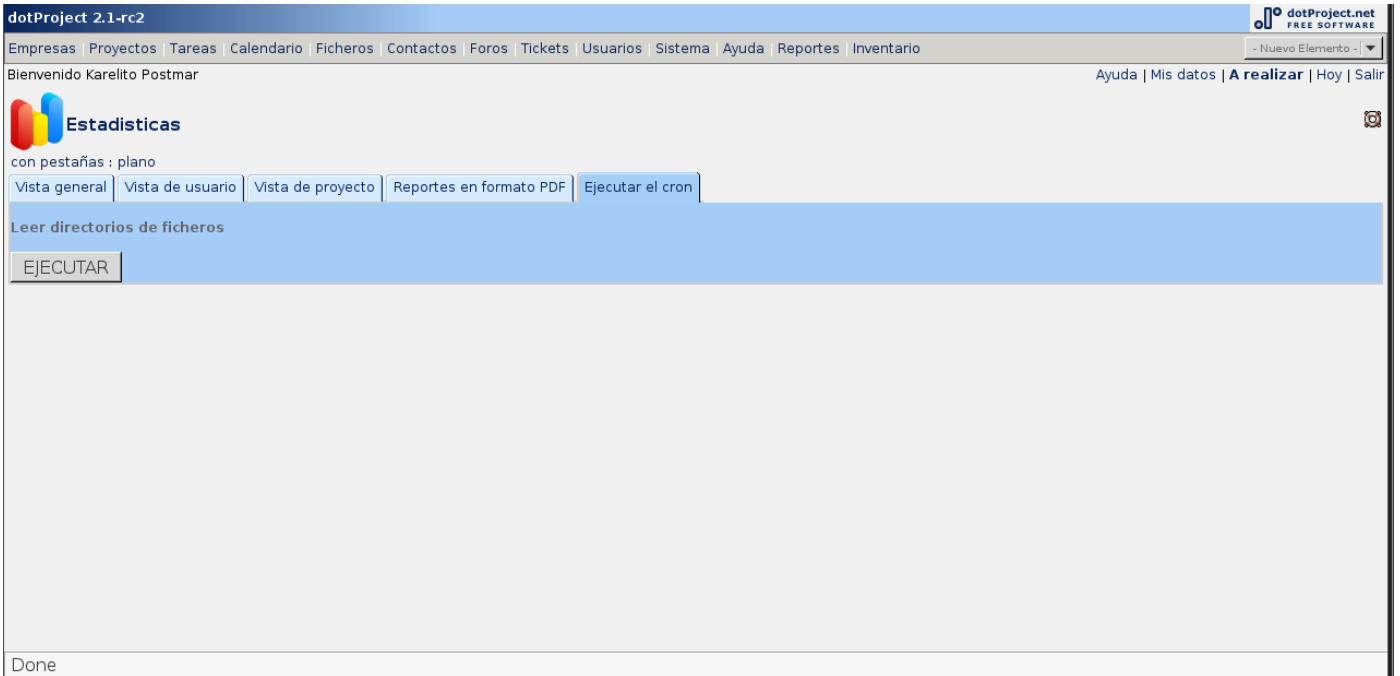
con pestañas : plano

Vista general | Vista de usuario | Vista de proyecto | Reportes en formato PDF | Ejecutar el cron

	Descripción
	Tiempo total de utilización de las aplicaciones para cada usuario
	Total time of use of applications for each project

Done

Anexo #8: Prototipo de la Historia de Usuario Ejecutar el CRON de forma manual.



Anexo #9: Código fuente del instalador de GENEST

```
<?php
/**
 * Name: Report
 * Directory:report
 * Version 0.1
 * Type: user
 * UI Icon: ?
 */
require_once(DP_BASE_DIR.' /modules/report/includes/functions/file.php');
$config = array();

$config['mod_name'] = 'report';
$config['mod_version'] = '0.1';
$config['mod_directory'] = 'report';
$config['mod_setup_class'] = 'DP_Report';
$config['mod_type'] = 'admin';
$config['mod_ui_name'] = 'Report';
$config['mod_ui_icon'] = 'charts.png';
$config['mod_description'] = 'Estadísticas del Home compartido';
$config['mod_config'] = false;

if (@$a == 'setup')
{
    echo dPshowModuleConfig( $config );
}

class DP_Report
{
    function install()
    {
        //creación de la tabla reports
        $sql = '(
            user VARCHAR(45) NOT NULL,
            date DATE NOT NULL,
            ip VARCHAR(45) NOT NULL,
            PRIMARY KEY(user,date)
        )TYPE=MyISAM';
    }
}
```

```

$q = new DBQuery;
$q->createTable('reports');
$q->createDefinition($sql1);
$q->exec();
$q->clear();

//creación de la tabla processes
$sql2 = '(
    name VARCHAR(45) NOT NULL,
    type VARCHAR(45) NOT NULL,
    reports_date DATE NOT NULL,
    reports_user VARCHAR(45),
    aplicaciones TEXT NOT NULL,
    total_time INTEGER UNSIGNED NOT NULL,
    PRIMARY KEY(name,reports_date,reports_user),
    INDEX process_FKIndex1(reports_user, reports_date)
)TYPE=MyISAM';

$q = new DBQuery;
$q->createTable('processes');
$q->createDefinition($sql2);
$q->exec();
$q->clear();

//creación de la tabla ip
$sql3 = '(
    ip VARCHAR(45) NOT NULL,
    reports_date DATE NOT NULL,
    reports_user VARCHAR(45) NOT NULL,
    PRIMARY KEY(ip,reports_date,reports_user),
    INDEX process_FKIndex1(reports_user, reports_date)
)TYPE=MyISAM';

$q = new DBQuery;
$q->createTable('ip');
$q->createDefinition($sql3);
$q->exec();
$q->clear();

$q = new DBQuery;
$q->addQuery('user_username');
$q->addTable('users');
$q->exec();
$q->clear();

$users = array();
while ( $row = $q->fetchRow()
    {
        $user = $row['user_username'];
        //crear el directorio donde se guardarán los reportes de cada usuario
        createFolder(DP_BASE_DIR.REPORTS_PATH,$user);
    }
    $q->clear();
    return db_error();
}

function remove()
{
    //eliminar tabla processes
    $q = new DBQuery;
    $q->dropTable('processes');
    $q->exec();
    $q->clear();

    //eliminar tabla reports
    $q = new DBQuery;

```



```

$q->dropTable('reports');
$q->exec();
$q->clear();

//eliminar tabla ip
$q = new DBQuery;
$q->dropTable('ip');
$q->exec();
$q->clear();

$q->addQuery('user_username');
$q->addTable('users');
$q->exec();

$users = array();
while ( $row = $q->fetchRow()
    {
        $user = $row['user_username'];
        //Eliminar los ficheros de cada uno de los directorios de los usuarios
        delFileDirectory(DP_BASE_DIR.REPORTS_PATH.$user);
        //Eliminar los directorios de cada uno de los usuarios.
        delFolder(DP_BASE_DIR.REPORTS_PATH.$user);
    }

$q->clear();
return db_error();
}
}
?>

```

Anexo #10: Estándar de codificación utilizado.

El código debe ser legible y estar escrito una sola vez, pero revisado y modificado varias veces. Lo cual permite que el código sea comprensible, y por tanto, que pueda ser mantenible. Aunque el rendimiento es importante, si hay una elección entre el rendimiento y la facilidad de lectura, la lectura debe tener preferencia.

Sangrado y longitud de la línea

La forma más elegante para marcar indentación (al principio del renglón) es la utilización de Tabs. Solamente se puede utilizar Tabs para la indentación o sangrado. Cada nivel de indentación puede ser solamente un Tab.

Modo de configuración de las líneas o las entradas de otros editores.

Dado que los editores modernos ya no se limitan a 80 caracteres de ancho de línea, las longitudes de línea son menos importantes que la lectura. Ya las líneas de 75 caracteres debe ser consideradas obsoletas y las líneas de más de 100 caracteres de longitud tienen que romperse.

Estructuras de control

Estas incluyen **if**, **for**, **while**, **switch**, etc. A continuación se muestra un ejemplo de declaración, ya que es el más complicado de ellos:

```

<?php
    if ((condición1) || (condición2)){
        acción1;
    }

```

```

    } elseif ((condición3) && (condición4)){
        acción2;
    } else {
        acción_por_defecto;
    }
?>

```

Las declaraciones de control deben tener un espacio entre las palabras clave de control y la apertura de paréntesis, para distinguirlos de las llamadas a funciones.

Es recomendable utilizar siempre llaves, incluso en situaciones en las que son técnicamente opcional ya que estas aumentan la legibilidad y disminuye la probabilidad de introducir errores lógicos cuando se añaden nuevas líneas.

Para declaraciones utilizando switch:

```

<?php
switch (condición) {
    case 1:
        acción1;
        break;
    case 2:
        acción2;
        break;
    default:
        acción_por_defecto;
        break;
}
?>

```

Funciones

Las funciones deben ser, sin espacios entre el nombre de la función, la apertura de paréntesis, y el primer parámetro, comas y espacios entre cada parámetro, y ningún espacio entre el último parámetro, el paréntesis de cierre, y el punto y coma. He aquí un ejemplo:

```

<?php
$var = foo($bar, $variable, $pvar);
?>

```

Como se muestra en el ejemplo anterior, debe haber un espacio a ambos lados de un signo de igualdad utilizado para asignar el valor de retorno de una función a una variable. En el caso de un bloque de asignaciones, se puede introducir más espacio para facilitar la lectura e interpretación del código, a continuación veremos un ejemplo:

```

<?php
$short = foo($bar);
$long_variable = foo($variable);
?>

```

Definición de funciones

La declaración de funciones sigue el “estilo K & R”, el cual permite ahorrar espacio vertical de lectura. Ejemplo:

```

<?php
function fooFunction($arg1,$arg2){
    if (condición) {
        definiciones;
    }
}

```

```

    }
    return $val;
}
?>

```

Los argumentos de las funciones que tengan valores por defecto se colocan al final de la lista de argumentos que posee la función. Ejemplo:

```

<?php
function connect(&$dsn, $persistent = false) {
    if (is_array($dsn)) {
        $dsninfo = &$dsn;
    } else {
        $dsninfo = DB::parseDSN($dsn);
    }

    if (!$dsninfo || !$dsninfo['phptype']) {
        return $this->raiseError();
    }
    return true;
}
?>

```

Comentarios

Completar la documentación en línea con bloques de comentarios proporcionados. Estos bloques de comentarios deben seguir el estilo C (`/* */`) ó el estándar de C++ (`//`).

Una regla general es que si nos fijamos en una sección de código es necesario comentar antes que se olvide cómo funciona.

Etiquetas de Código PHP

Siempre se debe usar `<?php` para abrir y `?>` para cerrar, y de esta forma delimitar el código PHP, nunca utilice `<?>`.

Etiquetas necesarias que tienen contenido variable

- Descripciones breves

Todos los bloques de comentarios deben ser breves descripciones, preferentemente en una oración pequeña.

- Versiones de PHP

Uno de los siguientes bloques de comentarios debe ir en el nivel de página:

- PHP versión 4
- PHP versión 5
- PHP versión 4 y 5

- Autor

No existe ninguna norma para determinar cuándo un nuevo contribuyente de código debe añadirse a la lista de los autores de un determinado archivo fuente. En general, los cambios deben caer en la categoría sustancial, es decir, alrededor de 10% al 20% de los cambios en el código. Las excepciones pueden ser hechas para la reescritura de funciones o en su contribución.

- Derechos de autor.

Los desarrolladores de módulos son libres de aplicar los derechos de autor independientemente de que usted desee. El año debe estar en formato de cuatro dígitos y, si es un período de años se trata de usar un guión entre la primera y el año más reciente. El titular del derecho de autor puede ser usted, una lista de personas, una empresa, el Grupo de PHP, etc. Ejemplos:

- * @copyright 2003 John Doe and Jennifer Buck
- * @copyright 2001-2004 John Doe
- * @copyright 1997-2004 The PHP Group
- * @copyright 2001-2004 XYZ Corporation

Para el código básico se recomienda que se aplique la siguiente declaración sobre derecho de autor:

- * @copyright 2001-2007 The dotProject Development Team.

➤ Ver

Añadir una etiqueta @see cuando quiera referirse a los usuarios o a otras secciones de la documentación del paquete.

Nombres convencionales

➤ Clases

Las clases deben tener nombres descriptivos. Evite el uso de abreviaturas cuando sea posible. Los nombres de clase siempre debe comenzar con una letra mayúscula. Las clases fundamentales son con el prefijo C. El módulo principal de nombres de clase deben comenzar con el nombre del módulo. Ejemplos de buenos nombres de clase son los siguientes:

- CAppUI
- MyModuleClass

➤ Métodos y funciones

Las funciones tienen además del nombre del paquete un prefijo para evitar las colisiones entre los nombres de paquetes. La letra inicial del nombre (tras el prefijo) es minúscula, y cada letra que comienza una nueva "palabra" es mayúscula. Algunos ejemplos son:

- Connect()
- GetData()
- BuildSomeWidget()
- XML_RPC_serializeData()

Los miembros privados de la clase (es decir, los miembros de la clase que van a utilizarse sólo dentro de la misma clase en que se declaró; realmente PHP no soporta todavía ejecutar variables privadas) están precedidos por un solo subrayado. Por ejemplo:

- _sort()
- _initTree()
- \$this->_status

En PHP5 los miembros protegidos de la clase (es decir, los miembros de la clase que van a utilizarse sólo dentro de la misma clase en la que se declaran o en subclases que se extienden desde la misma) no están precedidos por ningún subrayado. Por ejemplo:

- protected \$somevar
- protected function initTree()

➤ Constantes

Todas las constantes deben ser siempre en mayúsculas y con subrayados para separar palabras. Estas poseen un prefijo de nombres constantes con el nombre de la clase o módulo en el que se utiliza. Las únicas constantes que se exceptúan de la norma de todas las mayúsculas y siempre deben estar en minúsculas son las constantes **true**, **false** y **null**.

➤ Variables Globales

Si el módulo tiene que definir las variables globales, debe empezar su nombre en mayúsculas y con el nombre de la clase o del módulo.

Anexo #11: Pasos para la instalación de módulos al dotProject.

El SGP dotProject dispone de diferentes módulos que aumentan sus funcionalidades básicas.

1) Existe un grupo de módulos para dotProject que se pueden descargar desde la dirección <http://sourceforge.net/projects/dotmods> en caso de que no sea un módulo creado por nosotros mismos.

2) Copiar el directorio del módulo en el directorio *modules* del dotproject.

Ej. Directorio: dotproject/modules/módulo_de_prueba

3) Activar el módulo.

Para activar los módulos dtproject brinda una interfaz bastante amigable con el usuario.

La ruta a seguir para activar los módulos es la siguiente:

- Sistema
- Ver módulos
- Seleccionar módulo a instalar
- Activar el módulo.
- Darle visibilidad módulo (para que aparezca en el menú general)

Anexo #12: Código fuente del *controlador* genericPie.php.

```
<?php
include ('../../../../../lib/jpgraph/src/jpgraph.php');
include ('../../../../../lib/jpgraph/src/jpgraph_pie.php');
include ('../../../../../lib/jpgraph/src/jpgraph_pie3d.php');

extract($_GET);
$data=split(",",$data);
array_pop($data);
$legend=split(",",$legend);
array_pop($legend);
$default_colours=array('palegreen3','steelblue3','khaki1','tan3','dodgerblue4');

$graph = new PieGraph(500,250,"auto");
$graph->title->Set($title);
$graph->title->SetFont(FF_FONT1,FS_BOLD);
$graph->title->SetColor('black');
$graph->legend->SetShadow('darkgray@0.5');
$graph->legend->SetFillColor('white');
$graph->legend->SetPos(0.005,0.999,"left","bottom");
$graph->SetBackgroundImage('../background.png',BGIMG_FILLFRAME);

$p1 = new PiePlot3D($data);
$p1->value->SetColor("black","darkred");
$p1->value->SetFont(FF_FONT1,FS_BOLD);
$p1->ExplodeSlice(1);
$p1->SetCenter(0.45);
$p1->SetLegends($legend);
```

```

$p1->SetSliceColors($default_colours);
$graph->Add($p1);
$graph->Stroke();
?>

```

Anexo #13: Código fuente del *controlador* genericBar.php.

```

<?php

include("../..../lib/jpgraph/src/jpgraph.php");
include("../..../lib/jpgraph/src/jpgraph_bar.php");

extract($_GET);

$web=split(",", $webs);
array_pop($web);

$ofi=split(",", $ofis);
array_pop($ofi);

$ide=split(",", $ides);
array_pop($ide);

$jab=split(",", $jabs);
array_pop($jab);

$otros=split(",", $otross);
array_pop($otros);

$user=split(",", $users);
array_pop($user);

$count=count($otros);

for($i=0;$i<$count;$i++)
{
    $total[]=$otros[$i]+$web[$i]+$jab[$i]+$ofi[$i]+$ide[$i];
}

for($i=0;$i<$count;$i++)
{
    $web[$i]=($web[$i]*100)/$total[$i];
    $jab[$i]=($jab[$i]*100)/$total[$i];
    $ide[$i]=($ide[$i]*100)/$total[$i];
    $ofi[$i]=($ofi[$i]*100)/$total[$i];
    $otros[$i]=($otros[$i]*100)/$total[$i];
}

$graph = new Graph(470,250,'auto');
$graph->SetScale("textlin");
$graph->img->SetMargin(40,80,30,40);

$graph->legend->Pos(0.02,0.15);
$graph->legend->SetShadow('darkgray@0.5');
$graph->legend->SetFillColor('white');

$graph->xaxis->SetTickLabels($gDateLocale->GetShortMonth());
$graph->SetBackgroundImage('../background.png',BGIMG_FILLFRAME);

$graph->xaxis->SetTickLabels($user);
$graph->xaxis->SetFont(FF_FONT1,FS_BOLD);
$graph->xaxis->SetColor('black');

$graph->yaxis->SetFont(FF_FONT1,FS_BOLD);
$graph->yaxis->SetColor('black');

```

```
$graph->ygrid->SetColor('black@0.5');

$graph->title->Set('Statistics of users for applications group');
$graph->title->SetMargin(4);
$graph->title->SetFont(FF_FONT1,FS_BOLD);
$graph->title->SetColor('black');

$bplot1 = new BarPlot($web);
$bplot2 = new BarPlot($jab);
$bplot3 = new BarPlot($ide);
$bplot4 = new BarPlot($ofi);
$bplot5 = new BarPlot($otros);

$bplot1->SetFillColor('palegreen3@0.4');
$bplot2->SetFillColor('steelblue3@0.4');
$bplot3->SetFillColor('khaki1@0.4');
$bplot4->SetFillColor('tan3@0.4');
$bplot5->SetFillColor('dodgerblue4@0.4');

$bplot1->SetLegend('Web');
$bplot2->SetLegend('Jabber');
$bplot3->SetLegend('IDE');
$bplot4->SetLegend('Ofimatica');
$bplot5->SetLegend('Otros');

$bplot1->SetShadow('black@0.3');
$bplot2->SetShadow('black@0.3');
$bplot3->SetShadow('black@0.3');
$bplot4->SetShadow('black@0.3');
$bplot5->SetShadow('black@0.3');

$gbarplot = new GroupBarPlot(array($bplot1,$bplot2,$bplot3,$bplot4,$bplot5));
$gbarplot->Setwidth(0.6);
$graph->Add($gbarplot);

$graph->Stroke();
?>
```