

Universidad de las Ciencias Informáticas

Facultad 4



**Título: “Herramienta para calcular indicadores de
productividad de esfuerzo en una
Universidad Productiva.”**

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

Autor: José Alejandro Lugo García.

Tutores: Dra. Ana María García Pérez.
Ing. Ramsés Delgado Martínez.

Junio 2008

"La gestión de la calidad es un proceso que no debe estorbar a quienes la construyen."



DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos 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 _____.

José Alejandro Lugo García

Ramsés Delgado Martínez

Ana María García Pérez

Firma del Autor

Firma del Tutor

Firma del Tutor

AGRADECIMIENTOS

Detrás de cada logro está la mano amiga, el consejo oportuno, la idea renovadora. El resultado de este trabajo no hubiera sido posible sin la ayuda de ciertas personas de las cuales estoy y viviré eternamente agradecido. Cada hora que aquí se esconde estuvo salpicada con su valiosa contribución:

A mi familia por su guía, amor y apoyo incondicional en cada momento de mi vida y los valores que han sembrado en mí.

Mis tutores la Dra. Ana María García y el Ing. Ramsés Delgado Martínez, gracias por la confianza que recibí para esta novedosa tarea.

Al eterno líder de cien batallas, Fidel Castro Ruz, por haber ideado tanta nobleza y humanismo.

Mis profes, aquellos que me inculcaron la sed insaciable de conocer el por qué de las cosas.

A mis colegas del proyecto Prisiones: Hermes Miguel Velázquez Domínguez, Franklin Rivero Duharte, Luis Alberto Pimentel González y en especial a mi amigo Damián Pérez Alfonso. Al piquete de Nova Linux: Ángel Goñi Oramas, Yoandy Rodríguez Martínez y Mijail Hurtado Fedorovich. A Luis Miguel Guerra Nieves del proyecto Aduana.

Al equipo de desarrollo del proyecto StatSVN, por su preciado aporte a la comunidad del Software Libre mundial.

A los compañeros de la Dirección de Calidad de Software UCI.

A todo aquel que de una forma u otra contribuyó en la realización de este trabajo...

Muchas gracias.

DEDICATORIA

A mi abuela Olga: siempre estarás en mi corazón...

A mi abuelo César: mi ángel de la guarda.

A mi mamá, la mente más maravillosa que he conocido.

A mi papá: de ti saqué la perseverancia.

A mi abuela Carmen: tu espíritu es ternura.

A mi abuelo Pepe: sé que me querías mucho.

A mis también padres, Soraya y Manolo: forjadores del carácter y guías incondicionales.

A mis tíos, César y Yolanda: enciclopedias de la razón y ejemplos a seguir.

A mis hermanos, Anabel, Yenier y Mirell: mi sangre alentadora.

A Yilo: hermosa mujer, amiga, confidente, mi amor.

A Martica, osbrújuli: Tus hechizos hacen bien al corazón.

A Tamia y Adriana: resplandores del rocío al amanecer.

A mis amigos: energía que revitaliza y tesoro más valioso.

A los que alguna vez desearon estudiar pero la vida se los impidió...

A todas las personas que me abrieron su corazón y confiaron en mí...

RESUMEN

El presente Trabajo de Diploma se relaciona con el tema de la estimación en proyectos de desarrollo de software.

Se caracteriza el estado del arte en dicho tema demostrándose que la determinación del tamaño es un elemento clave. Se exponen las métricas para tamaño más usadas, el software existente utilizado para el control de versiones de proyectos, así como un estudio de diversas tecnologías y herramientas, que permitió decidir como objetivo lograr dos resultados útiles que sirven de base para las estimaciones de nuevos proyectos: un **Servicio de Integración de Estadísticas** acoplado a un Sistema de Control de Versiones y un **Sistema de Gestión de Indicadores de Productividad** basado en las estadísticas colectadas.

Para dichos resultados se documenta el análisis y diseño con diagramas de casos de uso, diagramas de clases y de implantación.

Para implementar el Servicio de Integración de Estadísticas se utilizó código fuente proveniente de un proyecto de software libre, al que se realizaron modificaciones y adiciones a fin de lograr coleccionar datos provenientes de los desarrollos reales de los proyectos e insertar estos datos en una base de datos relacional que se puede considerar como Línea Base de la Organización.

Los indicadores de productividad están preparados para ser obtenidos en función de las plataformas de desarrollo, ya que se emplean las líneas de código como medida del tamaño. Cada programador (equipo, producto, etc.) tendrá una productividad calculada en función de la plataforma, por lo que se sientan las bases para una mejora en la gestión de la organización a partir de su madurez actual.

En el trabajo se describen los pasos para la implantación y cómo se emplean las herramientas desarrolladas.

Los principales usuarios son los especialistas de la Dirección de Calidad de la Universidad de las Ciencias Informáticas y los jefes de proyectos y arquitectos.

Palabras claves: estimación, línea base, control de versiones, calidad del software.

SUMMARY

This work is related to the theme of estimation in software development projects.

It characterizes the state of the art in this issue showing that the determination of size is a key element. It outlines the most commonly used metric for sizing, the existing software used for version control of projects and a study of various technologies and tools, which enabled us to decide two results which will serve as useful basis for the estimation of new projects: A ***Statistical Integration Service*** coupled to a version control system and a ***Management System for Productivity Indicators*** based on the collected statistics.

For those results analysis and design were documented with use cases diagrams, class diagrams and deployment diagrams.

To implement the integration services of Statistics we used source code from a free software project, where changes and additions were made to collect data from actual developments projects and insert the data into a *relational database* that may be considered as a *baseline* for the Organization.

The productivity indicators are ready to be mobilized on the basis of development platforms, as they use *lines of code* as a measure for size. Each programmer (equipment, product, etc.) will have a productivity calculated on the basis of the platform, which lays the groundwork for an improvement in the management of the organization from its current maturity.

This work describes the steps for the implementation of our project and how users can use the developed tools.

The main users are specialists from the Directorate of Quality at the University of Informatics Science as well as project managers and architects.

Keywords: estimation, baseline, version control, quality software.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: LA PROBLEMÁTICA DE LA ESTIMACIÓN EN PROYECTOS DE DESARROLLO DE SOFTWARE.	6
1.1 ESTIMACIÓN DEL ESFUERZO EN PROYECTOS DE DESARROLLO DE SOFTWARE.....	6
1.1.1 Caracterización de métricas en entornos de producción de software	7
1.1.2. Automatización de la gestión de la Calidad de una Organización de software partiendo de la gestión de configuración	12
1.2 SOFTWARE EXISTENTE UTILIZADO PARA EL CONTROL DE VERSIONES EN UN PROYECTO	14
1.2.1 Borland StarTeam	14
1.2.2 Rational Clear Case	15
1.2.3 Microsoft Visual Source Safe	16
1.2.4 Plastic SCM.....	17
1.2.5 CVS.....	18
1.2.6. SVN.....	19
1.2.7 StatSVN	19
1.3 TECNOLOGÍAS Y HERRAMIENTAS PROPUESTAS PARA EL DESARROLLO DEL SISTEMA.....	20
1.3.1 La tecnología Java. Entornos de Desarrollo Integrados.	20
1.3.2 Plugins & Frameworks.	22
1.3.3 Sistemas gestores de Base de Datos (SGBD)	25
1.4 FUNDAMENTACIÓN DE LA METODOLOGÍA DE DESARROLLO. HERRAMIENTAS.	27
1.4.1 Proceso unificado de desarrollo (RUP).....	27
1.4.2 Lenguaje de modelado.....	29
1.4.3 Visual Paradigm for UML	30
CONCLUSIONES DEL CAPÍTULO.....	32
CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LAS HERRAMIENTAS PROPUESTAS.....	33
2.1 INTRODUCCIÓN	33
2.2 MODELO DEL NEGOCIO	33
2.2.1 Descripción del proceso.....	36
2.3 REQUISITOS NO FUNCIONALES DE LAS HERRAMIENTAS PROPUESTAS.	39
2.4 SISTEMA DE GESTIÓN DE LOS INDICADORES DE PRODUCTIVIDAD	40
2.4.1 Requisitos funcionales	40

2.4.2 Definición de los actores del sistema.....	41
2.4.3 Listado de los Casos de Uso del Sistema. Paquete “Datos del desarrollo”.....	42
2.4.4 Listado de los Casos de Uso del Sistema. Paquete “Datos del Proyecto”.....	44
2.4.5 Diagrama de Casos de Uso.....	48
2.4.6. Especificaciones de los casos de uso. Paquete “Datos del Desarrollo”.....	50
2.4.7. Especificaciones de los casos de uso. Paquete “Datos del Proyecto”.....	64
2.5 SERVICIO DE INTEGRACIÓN DE ESTADÍSTICAS DEL STATSVN (SVNSTATBASH).....	89
2.5.1 Requisitos funcionales.....	89
2.5.2 Modelo de la BD.....	89
2.5.3 Descripción de las tablas.....	90
2.5.4 Arquitectura del Servicio.....	93
CONCLUSIONES DEL CAPÍTULO.....	96
CAPÍTULO 3: GUÍAS PARA LA IMPLANTACIÓN Y USO DE LAS HERRAMIENTAS	
DESARROLLADAS.....	97
3.1 INTRODUCCIÓN.....	97
3.2 BASE DE DATOS DEL SOFTWARE DE GESTIÓN.....	97
3.3 NACE UN PROYECTO.....	97
3.4 LA UTILIZACIÓN DE SUBVERSION.....	99
3.5 RECOLECTANDO LOS TAMAÑOS.....	100
3.6 MODELO DE IMPLANTACIÓN.....	110
CONCLUSIONES DEL CAPÍTULO.....	111
CONCLUSIONES.....	112
RECOMENDACIONES.....	113
REFERENCIAS BIBLIOGRÁFICAS.....	114
ANEXOS.....	119
ANEXO 1. CÓDIGO DEL SERVICIO SVNSTATBASH.....	119
ANEXO 2. INSTALACIÓN Y CONFIGURACIÓN DE LA BD DEL SGIP SOBRE UBUNTU 8.04.....	124
ANEXO 3. GRÁFICOS DE LA SUITE HTML GENERADA POR SVNSTATBASH. (PROYECTO SGIP).....	128
ANEXO 4. FICHERO DE DESCRIPCIÓN DEL PROYECTO.....	131

INTRODUCCIÓN

A partir del surgimiento de las sociedades productivas el hombre se dedicó a elaborar herramientas y medios con los que satisfacer no solo sus necesidades vitales, sino también para garantizar la complacencia de sus semejantes. Así por ejemplo, desde antes de nuestra era, aquellos constructores de casas que no las hicieran con buena resistencia enfrentaban penas rigurosas por el peligro de muerte o derrumbe sobre sus ocupantes. Esto obligaba a los ejecutores a tener en cuenta ciertos principios de construcción para así lograr el desarrollo de sus obras de forma satisfactoria. Existiendo tal retórica y con el decursar del tiempo, fue surgiendo un nuevo concepto que comenzó a aplicarse a todos los bienes a los que el hombre diera origen. El término de calidad, responde a la necesidad de controlar el conjunto de atributos o propiedades de un objeto lo que permitirá emitir un juicio de valor acerca de él, o sea, “el grado en el que un conjunto de características inherentes cumple con los requisitos” [ISO 9000]

El mundo actual recibe con agrado los beneficios que aportan las nuevas tecnologías de la información y las comunicaciones, más conocidas por TICs. Se han hecho posibles el acceso, control y manipulación de la información a una escala nunca antes vista. Para facilitar esta vorágine, las empresas productoras de software se ven impulsadas al desarrollo de los recursos que respalden este proceso. La calidad ha pasado a formar parte importante en la esfera de la producción de herramientas informáticas ya que “si un equipo de software aplica la calidad a todas las actividades de la ingeniería de software, reducirá la cantidad de trabajo repetido que deba realizar. Esto supondrá costes más bajos y, lo que es más importante, mejorará el tiempo de llegada al mercado” [Pressman 2001]

La calidad, enfocada en términos cuantitativos, ofrece la visión de la rapidez con que un producto es desarrollado. Aunque para ello se hace necesario establecer indicadores que avalen el cumplimiento de la tarea y que permitan poder utilizarlos a fin de emitir el estado de desarrollo en que se encuentra la misma.

Dada la complejidad de las aplicaciones que se elaboraran en nuestros días, cualquier empresa que desee acelerar el proceso de desarrollo de software con calidad en su seno, necesitará automatizar una métrica afable con los intereses del usuario que permita controlar y centralizar la información relativa al desarrollo de cualquier tipo de herramienta de software; sin esta condición sería tortuoso tratar de manejar los indicadores que se relacionan con la calidad del proceso, incidiendo de manera negativa en tiempo y valiosos recursos.

Desde el triunfo revolucionario cubano de 1959, la máxima dirección de nuestro país ha encaminado sus esfuerzos en incrementar la cultura general integral de sus habitantes. Esto ha posibilitado obtener al cabo de años de duro batallar, una inestimable fuerza de capital humano. Actualmente, nuestro país se encuentra inmerso en la Informatización de la Sociedad. Proceso que, utilizando de forma ordenada y masiva las Tecnologías de la Información y las Comunicaciones, contribuye a la satisfacción de las necesidades de múltiples esferas de la población que encuentran en la informática la herramienta para catalizar el proceso de desarrollo nacional.

Dadas las condiciones actuales siguientes:

- Existencia del Ministerio de la Informática y las Comunicaciones.
- Universalización de las carreras de Ingeniería Informática.
- Sólido soporte tecnológico en los polos científicos y universitarios de recursos informáticos.
- Propuesta del Programa Nacional de Informatización de la Sociedad Cubana.

Nos encontramos ante un momento fecundo y propicio para la elaboración de soluciones tecnológicas basadas en aplicaciones informáticas y las relacionadas con las de gestión de la calidad del software no escapan a esta concepción.

La Universidad de las Ciencias Informáticas (UCI), surgida en momento crucial de la Batalla de Ideas, vislumbra para Cuba una oportunidad importante en el mercado mundial de software y ha venido desarrollando numerosos trabajos con éxito acercándola cada vez más a este propósito. Solo que esta meta es posible alcanzarla siguiendo principios donde la calidad del software juegue el papel rector en el desarrollo de sus proyectos. Un aporte a la obtención de indicadores que midan esta calidad es el que pretende dar este trabajo, proponiendo una “Herramienta para calcular Indicadores de Productividad de Esfuerzo en una Universidad Productiva”, con la que se quiere tratar de dar solución a puntos débiles que existen en la estimación del esfuerzo de los distintos proyectos que puedan estarse desarrollando dentro de dicha institución.

Por otra parte nuestra Isla, país en vías de desarrollo, ha visto al igual que otras naciones desarrolladas y del tercer mundo, la utilización del software libre como un camino interesante para construir herramientas informáticas. Sobre todo cuando la situación cubana se torna más compleja. “Las grandes empresas desarrolladoras de software propietario tienen su domicilio legal en Estados Unidos de América, por lo que se atienen a la legislación sobre exportaciones de su país, y si a esto se le suma que en las licencias de sus programas se exige a los licenciarios cumplir esas mismas

regulaciones, nos encontramos ante una imposibilidad de adquirir un programa de esta clase.” [Rodríguez 2006].

Dada la trascendencia de impulsar una sociedad justa, el software libre hace alusión a ciertos principios, conocidos como las cuatro libertades básicas:

- Usar el programa, con cualquier propósito.
- Estudiar cómo funciona el programa, y adaptarlo a tus necesidades.
- Distribuir copias.
- Mejorar el programa y hacer públicas las mejoras a los demás.

El software libre provee una base sólida para brindar la satisfacción de usuarios novatos y expertos, tanto clientes como profesionales de la informática y está ganando terreno en muchos aspectos de la vida social y económica. El uso de las tecnologías que respaldan este novedoso movimiento ha sido un principio seguido en el trabajo que mostramos, promoviendo que el fruto del mismo mantenga estas cuatro libertades. La razón principal no se encuentra solamente en la independencia de licencias privativas de software que su uso implica, sino que hace gala de las ideas de nuestro Comandante en Jefe Fidel que, aplicadas al mundo de la informática, sugerirían esta vez: un mundo mejor para el software también es posible.

Por tanto, para la realización de este trabajo de diploma se identificó el siguiente **problema**: ¿Cómo obtener indicadores de productividad de esfuerzo en el entorno de desarrollo de proyectos de software de una Universidad Productiva utilizando una herramienta sobre software libre que conforme la línea base en ayuda a la gestión de la calidad?

De esta manera, el **objeto de estudio** fueron los proyectos de software que se llevan a cabo dentro de la UCI como Universidad Productiva.

El **campo de acción** incluyó las herramientas de control de versiones, gestión de proyectos de software, bases de datos y software libre.

Todos estos puntos analizados conllevaron a plantear el **objetivo general** de este trabajo: Desarrollar herramientas sobre software libre, que encuestando un sistema de control de versiones, habiliten la información necesaria para obtener indicadores de productividad de esfuerzo en una Universidad Productiva.

Teniendo en cuenta este propósito, se establecieron los siguientes **objetivos específicos**:

- Proponer una métrica que mida el tamaño del código fuente desarrollado en un lenguaje de programación específico, e implementar su cálculo.
- Definir la herramienta de control de versiones a utilizar para implementar, asociado a ella, el cálculo del tamaño de los componentes que se vayan liberando por los programadores.
- Construir un software de gestión que sea capaz de capturar datos de los entornos productivos, dejando lista la información para poder emitir reportes que muestren indicadores de productividad de esfuerzo en proyectos de desarrollo.

Para resolver la problemática planteada se desarrollaron un conjunto de **tareas**:

- Analizar las distintas métricas orientadas al tamaño existentes, para valorar su nivel de implicación en la gestión de la calidad de los proyectos de software.
- Realizar un estudio sobre el Software Libre en general y las herramientas que lo acompañan que podrían ser de utilidad para el desarrollo del software propuesto.
- Revisar la bibliografía científica sobre las herramientas de control de versiones existentes a nivel mundial.
- Estudiar el funcionamiento de las herramientas de control de versiones más utilizadas en el entorno productivo de la UCI, proponiendo la que se va a utilizar para el cálculo de los tamaños.
- Seleccionar un gestor de bases de datos relacional libre con características adecuadas para la seguridad y control de los datos que manejará la aplicación.
- Seleccionar un lenguaje y un Entorno de Desarrollo Integrado (IDE) libre y multiplataforma para llevar a cabo las tareas de implementación.
- Indagar por la metodología adecuada para el desarrollo de sistemas de software como el que se quiere construir.
- Analizar y diseñar las herramientas.
- Implementar los casos de usos derivados del análisis y diseño.
- Probar el software elaborado para verificar que cumpla con todos los requisitos solicitados.

Con este trabajo se esperan los siguientes **aportes prácticos**:

- Una herramienta integrada a un sistema de control de versiones que calcula los tamaños de los componentes liberados.
- La automatización del trabajo de recolectar datos sobre la velocidad de desarrollo que realizan manualmente programadores y personal de calidad hoy día.
- Una base de datos con los datos que se requieren para conformar una línea base que permite obtener el esfuerzo actual de proyectos de desarrollo de software.
- Sentar las bases para estimar con mayor precisión el esfuerzo de nuevos proyectos, utilizando los indicadores de productividad que arroje la línea base.

Este trabajo de diploma se estructura en 3 Capítulos.

El primero se dedica a una caracterización del estado del arte en la estimación de proyectos de software, considerando las métricas para tamaño más usadas, el software existente utilizado para el control de versiones en un proyecto, así como un estudio de diversas tecnologías y herramientas que permitió decidir cuales se utilizaron para el desarrollo del sistema.

El segundo capítulo describe el análisis y diseño de las herramientas propuestas: Servicio de Integración de Estadísticas del Sistema de Control de Versiones y el Sistema de Gestión de Indicadores.

El tercer capítulo describe los pasos para la implantación y cómo se emplean las herramientas desarrolladas.

Este trabajo ha sido validado por especialistas de la Dirección de Calidad de la Universidad de las Ciencias Informáticas.

CAPÍTULO 1: La problemática de la estimación en proyectos de desarrollo de software.

En este capítulo se examina el estado del arte en cuanto a la aplicación de técnicas de recolección de datos relacionados con la productividad y el costo en entornos de desarrollo de software.

Se describe como se han utilizado históricamente varias métricas orientadas a medir el tamaño de lo que se produce, así como la factibilidad de automatizar el cálculo del tamaño producido.

Se exponen las características principales del software existente utilizado para el control de versiones en un proyecto, así como un estudio de diversas tecnologías y herramientas que permitió decidir cuales se utilizaron para cumplir con los objetivos del trabajo.

1.1 Estimación del esfuerzo en proyectos de desarrollo de software

La gestión tiene vital importancia para que un proyecto alcance su éxito. Este deberá desarrollarse con una planificación rigurosa y detallada para cada etapa donde el resultado final será tener el control de los recursos necesarios para la realización del proyecto. Este plan además se actualizará en la medida que avance el proyecto a fin de conocer las necesidades reales. Estimar costos y plazos para las tareas a realizar es uno de los objetivos fundamentales dentro de esta gestión.

Para Pressman [Pressman 2001] “la planificación implica la estimación, su intento por determinar cuánto dinero, esfuerzo, recursos y tiempo supondrá construir un sistema o producto específico de software” y la importancia que le da la responde con la siguiente pregunta “¿Podría construir una casa sin saber cuánto estaría dispuesto a gastar? Por supuesto que no y puesto que la mayoría de los sistemas y productos basados en computadora cuestan considerablemente más que construir una casa grande, podría ser razonable desarrollar y estimar antes de empezar a construir el software”.

Está claro que la estimación nunca será una ciencia exacta ya que son demasiados los factores que están en juego: técnicos, humanos, el entorno o políticas que puedan afectar el curso normal de desarrollo incidiendo todo esto sobre el costo final y el esfuerzo involucrado.

1.1.1 Caracterización de métricas en entornos de producción de software

En el campo de la programación por computadoras, ha sido una realidad la falta de precisión y la ambigüedad en las unidades de medida de calidad y productividad, sin embargo se reconoce que el progreso científico en cualquier campo es totalmente dependiente de la habilidad de medir cantidades en forma precisa.

La forma de dimensionar, tanto los productos, como los proyectos y el proceso en si, es un tema verdaderamente interesante y el logro de resultados en este sentido puede constituir un fuerte impulso para la gestión de la calidad en la industria del software actual.

De hecho es conocido que grandes proyectos han fracasado al no estar a tiempo o dentro del presupuesto por una mala estimación del esfuerzo o la duración, o de las capacidades requeridas de los ingenieros y de la empresa. Pero tener una métrica que sea capaz de ser usada para varios tipos de lenguajes seria de gran ayuda al mundo del software actual.

Respecto a trabajos de medición de métricas de software podemos citar, por ejemplo: las métricas de complejidad de McCabe [McCabe 1976], la métrica de longitud de código fuente, métricas de punto de función [Alb86, Cap91], métricas para productos web [Men01, Ols01], entre otros.

Algunos organismos y empresas privadas han realizado trabajos en el campo de las métricas del software para dominios de aplicaciones específicos, inclusive se proveen de las herramientas necesarias que aplican dichas métricas y/o dan soporte al proceso de análisis.

Sin embargo, la comunidad de ingenieros del software no ha acordado un conjunto de métricas universalmente aceptadas por el área como ha sucedido en disciplinas más tradicionales como la física o la química.

Pero para poder estimar el esfuerzo de desarrollo de nuevos proyectos hay que reunir información sobre cuanto "tamaño" se logra hacer por persona-mes involucrada en el proyecto, por lo que la determinación de una unidad de medida para el tamaño es la piedra angular de cualquier herramienta utilizada para la estimación. [García 2005]

Las métricas para tamaño del software.

Varias escuelas han emergido en el intento de definir una medida de tamaño para el software, las que respetan las “líneas de código”, las que analizan subelementos de estas líneas separadamente (operadores/operandos) [Halstead 1977], así como las que abandonan las líneas completamente para caer en las funciones [Albretch 1979], los objetos [Boehm 1999] y hasta los casos de uso [Sparks 2000].

El temprano trabajo de [Jones 1986] es un excelente artículo donde aparecen reflejadas las principales dificultades de las líneas de código como medida y lo paradójica que resulta esta métrica como criterio de productividad. Ya en este trabajo se enuncia que “...el desarrollo de un producto de software no solo consiste de tareas de codificación y estas tareas también deberían ser medidas”.

Según Pressman [Pressman 2000] una métrica tiene como características deseables las siguientes:

1. **Simple y fácil de calcular:** Debe ser de ayuda al ingeniero y si es posible, deberá ser diseñada para poderla computar automáticamente.
2. **Empírica y persuasiva:** Debe obtenerse de la práctica y disminuciones o incrementos de sus valores indicar algún síntoma.
3. **Consistente:** Debe ser objetivamente calculada con la menor ambigüedad posible, no importa la persona que la tome.

Sin embargo, un enfoque dirigido a unidades de trabajo, subtareas o completamiento de fases en el ciclo de vida no es tampoco la solución deseada si queremos que la medida cumpla los requerimientos enunciados en el párrafo anterior. La medición de la productividad en los ambientes actuales es una tarea compleja pues debería servir para reflejar el costo de cambiar un programa cuando las circunstancias cambian y reflejar los costos acumulados de una versión simple de programa.

El objetivo principal de la mejora del proceso de obtención de versiones de programas es bajar el costo del desarrollo por medio de un incremento de la velocidad en el logro de la satisfacción de las necesidades de los clientes. No puede ser simplemente velocidad de desarrollo en forma absoluta, sino relativa a la satisfacción plena de la necesidad que da origen al desarrollo del producto software.

La cantidad de retrabajo o “vueltas atrás” en el proceso deberá ser minimizada. Hoy en día la tecnología que nos apoya en este camino lo hace en el sentido puramente técnico, como tecnología dura, pero la tecnología blanda está por definir. El “costo por defecto”, sin una medida que tome en

cuenta la satisfacción del cliente, solo producirá errores de interpretación que conducen a una gestión no efectiva de la calidad. [García 2005]

Establecer una medida de tamaño standard tiene las siguientes ventajas:

Comparabilidad – Al utilizarse la misma métrica en todos los proyectos, se pueden realizar comparaciones entre ellos, y mejor aún, si es una métrica estándar en la industria, podría compararse ésta contra otras.

Administrar la productividad – Al obtener los tamaños producidos y los esfuerzos requeridos, entonces es posible establecer indicadores de productividad, con los cuales se pueden trazar planes de mejora.

Administración de calidad – De manera similar al disponer del tamaño y el número de defectos que se entregaron en un desarrollo, entonces se pueden establecer indicadores de calidad. Con una historia suficiente de proyectos, se pueden estimar proyectos futuros.

Valuar el SW de una organización – Una vez que se tiene el tamaño de cada aplicación dentro de la organización, es posible evaluar mejor los activos disponibles.

A continuación ofrecemos una breve panorámica de varias medidas posibles para esta “unidad de tamaño”.

Líneas de código

LOC (Lines Of Code) o SLOC (Source Lines Of Code), como también se le conoce, fue el primer método para medir el tamaño, cuando los programadores debían rendir cuentas sobre el monto de trabajo realizado.

Hoy en día sigue siendo una medida atractiva, habida cuenta que la mayoría de los editores de programas proporcionan esta medida de manera automática, o en su defecto, resulta muy fácil hallarla si los programas se codifican con lenguajes de programación estructurados.

Los principales problemas de utilizar líneas de código como métrica para estimación del esfuerzo son la falta de una *definición universal de línea de código*, su dependencia del lenguaje de desarrollo y la dificultad de estimar en fases tempranas del desarrollo la cantidad de líneas que tendrá una aplicación.

Puntos de función

Medida propuesta por Albretch [Albretch 1979] donde se cuantifica el tamaño y la complejidad de un sistema de software en términos de las funciones de usuario que este desarrolla. Algunas de las características fundamentales de los puntos de función son:

- Posibilita que la medida sea independiente del lenguaje o herramienta de desarrollo.
- Los puntos de función pueden ser estimados a partir de la especificación de requisitos o especificaciones de diseño, haciendo posible de este modo la estimación del esfuerzo de desarrollo en etapas tempranas del mismo.
- Al estar basados en una visión externa del usuario del sistema, los usuarios no técnicos del software poseen un mejor entendimiento de lo que los puntos de función están midiendo.

Esta medida es muy utilizada y con éxito en numerosas organizaciones que desarrollan productos de software en forma masiva. No obstante, el análisis por puntos de función está diseñado para medir aplicaciones de negocios, no es apropiado para aplicaciones técnicas o científicas dada la complejidad de ciertos algoritmos que las mismas no están diseñadas para manejar. La principal crítica que recibe esta medida es la de requerir una dedicación adicional en los proyectos de desarrollo de software, que suelen desenvolverse con presupuestos ajustados. Su implantación en una empresa no acostumbrada a su uso suele agregar esfuerzo y requerir un fuerte compromiso de la dirección. Suele ser vista por los desarrolladores como un mecanismo de control de su trabajo. Otros aspectos negativos serían:

- Resulta arduo formar al personal en su utilización y más todavía mantener unos criterios homogéneos de recuento.
- Carece de precisión cuando se trata de proyectos pequeños. Por debajo de unos 100 Puntos de Función (PF) resulta demasiado poco fiable.
- Para resultar realmente útil, una organización de desarrollo y mantenimiento de software debería actualizar continuamente el tamaño de su línea base, pero hacerlo resulta muy costoso especialmente si mantiene software adquirido a terceros.
- El factor de ajuste calculado a partir de las características generales del sistema resulta a veces ambiguo.

Puntos de característica

Una ampliación del punto de función es la llamada *métrica de puntos de características*. Propuesta por Caper Jones [Jones 1991] para aplicaciones de software de tiempo real, de control de procesos y

empotradas, esta métrica suple las carencias de la anterior acomodándose a situaciones donde la complejidad del algoritmo es alta.

Este método considera los mismos elementos que considera Albrecht [Albrecht 1979] en su análisis por puntos de función, sólo que añade la variable "número de algoritmos" y elimina los niveles de complejidad, así, cada cuenta es pesada por un valor único para ese componente.

Puntos de Casos de Uso

Basado en el trabajo realizado por Gustav Karner, miembro de Rational Software, [Sparks y Kapczynski 1999] muestran un interesante artículo que describe cómo utilizar los métodos de Karner en conjunto con otros métodos de estimación. Para esto propone lo que se conoce como Puntos de Caso de Uso, que consiste en contabilizar esta métrica a partir del análisis de casos de uso del sistema. El mejor momento sería durante las fases tempranas de un proyecto software orientado a objetos donde se capture el alcance teniendo en cuenta los casos de uso, clasificando a cada uno en simple, medio o complejo para producir el conteo de puntos. También se pueden ajustar los puntos de caso de uso según los requerimientos técnicos del proyecto y las características del personal, los cuales se pueden convertir directamente a horas para obtener una idea del plan del proyecto.

PSP y TSP

Ideado por Watts S. Humphrey, PSP (Personal Software Process), es usado por los ingenieros de software para recolectar y analizar datos sobre su trabajo [Humphrey 2001]. Es un conjunto estructurado de descripciones de proceso, mediciones y métodos que pueden ayudar a que los ingenieros mejoren su rendimiento personal. Proporcionan las formas, guiones y estándares que les ayudan a estimar y planificar su trabajo. Muestra cómo definir procesos y cómo medir su calidad y su productividad. Los estudios publicados generalmente usan datos recolectados usando PSP para esbozar las conclusiones cuantitativas sobre el impacto en el comportamiento de los programadores y la calidad del producto.

También de [Humphrey 2000], TSP (Team Software Process), es un proceso para equipos de 2 a 20 ingenieros de software entrenados en el uso de PSP. Propone un conjunto de procesos estructurados que indican qué hacer en cada fase del desarrollo del proyecto y muestra cómo conectar cada fase para construir un producto completo. El desarrollo de TSP comenzó en 1996 aunque la herramienta estuvo disponible a finales del 2000.

Si bien ambos procesos incluyen reconocidísimas técnicas para una buena organización, aún requieren de mucho trabajo por parte del ingeniero para lograr llevar el control necesario de si mismo y del equipo respectivamente a pesar de que en la actualidad se cuenta con sistemas de gestión para estos procesos como Process Dashboard.

El enfoque actual de recolección y análisis de indicadores, al realizarse manualmente, tiene sus repercusiones en tiempo, pues consume a los programadores parte del tiempo de desarrollo. Además implica disparidades en las interpretaciones de las métricas dado que cada programador interpreta las métricas a su manera.

1.1.2. Automatización de la gestión de la Calidad de una Organización de software partiendo de la gestión de configuración

El trabajo de [García 2005] propone diseñar, implementar y validar un proceso automático de formación de los indicadores de productividad y costo, a partir de la producción de componentes de la organización de software. Para lograr esto será necesario crear una herramienta que calcule una métrica para tamaño del software, acoplable a un sistema de gestión de configuración.

De manera general, la solución está basada en el diseño e implementación de un marco de trabajo conformado por un conjunto de herramientas y el determinado flujo de información entre ellas. (Ver Figura 1.1). Para mayor comprensión de este diagrama, se representan los procesos de manera general por círculos, las herramientas que se deben implementar por rombos y los sistemas ya existentes por rectángulos. Dentro de este marco el flujo comenzará y terminará en el cliente; desde que éste hace un pedido de software a la organización hasta que se le hace la entrega del producto.

Dimensión del Problema. En este proceso, como bien indica su nombre, el objetivo fundamental será dimensionar el problema a resolver dada la petición del cliente. Este módulo usa como información principal las métricas e indicadores obtenidos de proyectos anteriores y puede implementar técnicas de Inteligencia Artificial para la toma de decisiones, a fin de brindar datos importantes en la estimación de tiempo y recursos. (Este subsistema no es objetivo del presente diploma).

Recolector de Métricas. Este artefacto mediará entre la Producción y la Gestión de Configuración, donde la información recolectada formará parte de los datos guardados por el Sistema de Gestión de Indicadores (el Recolector y el Sistema de Gestión de Indicadores son nuestros objetivos).

Tablero de Mando. Herramienta que le facilitará al líder de proyecto y a los directivos tener una visión clara y abarcadora del avance del proceso. Aquí se podrán apreciar indicadores a nivel de proyecto así como índices globales de la empresa tales como los índices de defectos reportados por los clientes y los índices de productividad y de costos.

(Este subsistema tampoco es objetivo del presente diploma).

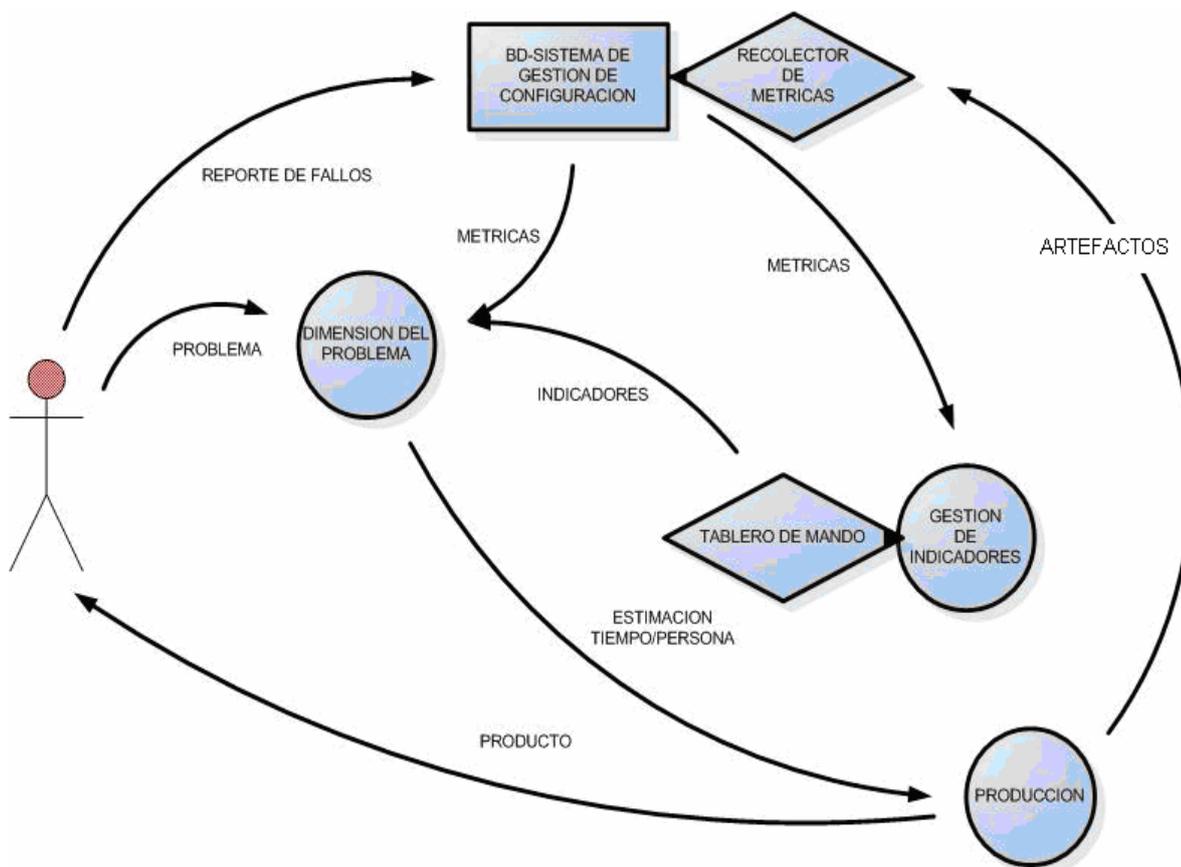


Figura 1.1 Flujo de Trabajo del Marco Propuesto.

Una herramienta para el control de versiones dispone de utilidades que permiten conocer los datos concernientes a los archivos que un desarrollador, ya sea analista, diseñador o programador, va subiendo hacia la base de datos del servidor de Gestión de Configuración. Por ello se analizarán a continuación algunas de estas herramientas a fin de seleccionar la que se empleará para el cálculo integrado de los tamaños.

1.2 Software existente utilizado para el control de versiones en un proyecto

Cuando construimos software de computadora, surgen cambios y debido a esto la necesidad de controlarlos eficazmente se hace evidente. La Gestión de la Configuración del Software (GCS) es un conjunto de actividades diseñadas para controlar el cambio identificando los productos del trabajo que probablemente cambien, estableciendo relaciones entre ellos, definiendo mecanismos para gestionar distintas versiones de estos productos, controlando los cambios realizados, y auditando e informando de los cambios realizados. [Pressman 2001]

Para Pressman [Pressman 1997] es importante la gestión de la configuración del software (GCS/SCM) ya que “si no controlamos el cambio él nos controlará a nosotros”. Y esto nunca es bueno. Es muy fácil que un flujo de cambios incontrolados lleve al caos a un proyecto de software correcto. Por esta razón la gestión de la configuración del software es una parte esencial de una buena gestión del proyecto y una práctica formal de la ingeniería del software.”

Se llama *control de versiones* a la gestión de versiones (revisiones) de todos los elementos de configuración que forman la línea base de un producto o una configuración del mismo [Spinellis 2005]. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado junto a las posibles especializaciones realizadas para algún cliente específico.

El proceso de control de versiones para un proyecto se puede realizar de forma manual aunque se recomienda la utilización de herramientas que faciliten esta gestión.

1.2.1 Borland StarTeam

Borland StarTeam proporciona una solución completa para la gestión de los cambios y la configuración que incluye la integración con la gestión de requerimientos, tracking de defectos, versiones de código fuente y seguimiento de proyectos y tareas. [Borland 2007]

Entre las características principales de esta herramienta se encuentran:

- *Control centralizado de los proyectos y tareas:* Al proporcionar un repositorio común, StarTeam mantiene actualizado los elementos del proyecto y mejora la productividad y comunicación del equipo.
- *Sistema de alto rendimiento para la gestión de la configuración y del cambio para equipos de todos los tamaños:* StarTeam proporciona la funcionalidad, seguridad y escalabilidad necesaria para cualquier equipo de desarrollo independientemente de su tamaño y distribución.
- *Acceso flexible, aunque seguro:* Los miembros del equipo pueden acceder desde y como

quieran, ya sea desde Web, IDE, escritorio o línea de comandos.

- *Reutilización de software entre proyectos:* Por medio de la capacidad de búsqueda, los miembros del equipo pueden compartir y reutilizar la información almacenada en StarTeam.
- *Mejora de la eficacia:* Por medio de flujos de trabajo modificables se obtiene la máxima eficacia durante el ciclo de vida de la aplicación.
- *Alta seguridad y protección de los proyectos:* StarTeam asegura que todos los elementos se almacenan de forma segura, controlando el acceso y las modificaciones en función de reglas establecidas.
- *Integración:* Se integra dentro de los IDEs de CodeGear, Eclipse y Visual Studio.

Una de las principales ventajas de StarTeam es que es muy fácil de utilizar y puede ser usado no solo por programadores sino también por aquellos roles que generan documentación de vital importancia.

Una de las mayores dificultades de los productos Borland es su elevado costo. Borland StarTeam 2005 Enterprise Advantage cuesta alrededor de 1340 euros [Mercado PC 2007].

1.2.2 Rational Clear Case

IBM Rational Clear Case es una herramienta de control de versiones básica de gran fiabilidad que resulta ideal para equipos de proyectos medianos y pequeños, proporciona una gestión del ciclo de vida y control de los activos de desarrollo de software. Con un control integrado de versiones, una gestión del espacio de trabajo automatizado, un soporte de desarrollo en paralelo, una gestión de línea base y gestión de builds y releases, Rational ClearCase proporciona las funciones necesarias para crear, actualizar, ofrecer, reutilizar y mantener los activos más importantes del negocio.

Otras de sus características son:

- Fácil instalación con asistentes de iniciación intuitivos para automatizar la configuración del servidor.
- Integración transparente con Rational ClearQuest para obtener un acceso directo al seguimiento de cambios y defectos.
- Interfaz web para acceder fácilmente desde cualquier navegador web estándar. Integrado con los IDE líderes en el sector, como WebSphere Studio, Eclipse y Microsoft .NET.
- Soporte para desarrollo en paralelo, que incluye bifurcaciones automáticas.

Una de las ventajas claves de adoptar herramientas de IBM Rational al nivel de toda la empresa de FNF es la capacidad de perfeccionar y acelerar la capacitación. [IBM 2007]

Su alto costo constituye una desventaja, IBM Rational ClearCase Floating User License + SW Maintenance 12 Months se eleva a alrededor de 5000 Euros con impuestos incluidos. [IBM 2007]

1.2.3 Microsoft Visual Source Safe

Microsoft Visual SourceSafe es una herramienta que permite el control de versiones para proyectos de equipo y proyectos individuales. Forma parte de Microsoft Visual Studio aunque está siendo sustituida por el Visual Studio Team Foundation Server.

Según MSDN [MSDN 2007] Visual Source Safe incluye, como mínimo, las siguientes funciones:

- Ayuda al equipo a evitar la pérdida accidental de archivos.
- Permite realizar un seguimiento de las versiones anteriores de un archivo.
- Admite la bifurcación, el uso compartido, la combinación y la administración de versiones de archivos.
- Realiza el seguimiento de las versiones de proyectos completos.
- Realiza el seguimiento del código modular (un archivo que se reutiliza, o se comparte, en varios proyectos).

Para las personas que desarrollan programas bajo el sistema operativo Windows, resulta una herramienta útil ya que se integra fuertemente con el entorno de desarrollo integrado o IDE de Visual Studio permitiendo un manejo relativamente simple de versiones sobre una computadora individual y en equipos de trabajo relativamente pequeños.

La principal desventaja de Visual SourceSafe reside en el método de acceso a los archivos compartidos que constituyen su repositorio mediante el protocolo SMB que no impide que éstos sean manipulados de manera externa al producto por cualquier persona que tenga acceso al mismo, provocando corrupción de datos. Este mismo tipo de acceso a archivos compartidos provoca que en equipos de trabajo grandes, el acceso concurrente pueda ser particularmente lento. A esto podemos añadir el carácter propietario de la herramienta y más al ser producto de Microsoft, el mismo no puede ser comercializado con Cuba. [Delgado 2006]

1.2.4 Plastic SCM

Plastic SCM es un sistema comercial desarrollado por Codice Software. Intenta centrarse en el desarrollo paralelo, la bifurcación, fusión y seguridad.

La herramienta Plastic SCM 1.5 soporta muchas novedades con respecto a su anterior versión 1.0. Las características más importantes se enumeran a continuación: [Codice Software 2007]

- *Gráfico de ramas:* Es una de las características más esperadas de la nueva versión de Plastic SCM. A partir de ahora será posible visualizar la evolución de un proyecto mediante esta nueva visualización, que junto al árbol de versiones en 3D marca una diferencia sustancial con todos los mecanismos de visualización existentes en el ámbito del control de versiones.
- *Soporte de SQL Server:* Plastic SCM, como sistema de nueva generación, se apoya en un backend de base de datos relacional que puede ser reemplazable y configurable por el usuario. La primera versión oficial de Plastic SCM incorporaba únicamente soporte para Firebird (<http://www.firebirdsql.org>), un sistema gestor de base de datos open source muy potente y fiable. Aunque la apuesta de Plastic por Firebird sigue siendo clara, se ha incorporado soporte de SQL Server (versiones 2005 y superiores) de modo que las empresas puedan aprovechar mejor su infraestructura existente.
- *Integración con CruiseControl:* La nueva release 1.5 de Plastic se integra con la versión 2.7 de CruiseControl permitiendo a los usuarios automatizar tareas de compilación tanto desde Java como .NET.
- *Mejoras en el instalador de Linux:* La versión 1.5 incluye mejoras en el instalador de Linux. A partir de ahora instalar y poner en marcha Plastic con sistemas con Ubuntu, Suse y Fedora es aún mucho más sencillo.

Plastic SCM ha sido desarrollado con tecnología .NET por lo que funciona en cualquier versión de Windows desde 2003 hasta Vista. Es totalmente compatible con Mono y eso le permite ser utilizado en Linux, MacOS y Solaris.

Como todo sistema hecho con tecnología Microsoft esta herramienta siempre impone sus limitantes ya que no constituye un software completamente libre.

1.2.5 CVS

CVS (Concurrent Versions System) implementa un sistema de control de versiones como aplicación informática. Como tal se encarga de mantener un registro de todo el trabajo y los cambios realizados en los archivos, de código fuente principalmente. Este se ha hecho popular en el mundo del software libre donde sus desarrolladores distribuyen y difunden el sistema bajo la licencia GPL.

Algunas características de este sistema son descritas a continuación:

- *Arquitectura cliente-servidor.* El servidor guarda la versión actual del proyecto y su historial. Los clientes se conectan al servidor para descargar una copia fiel del proyecto, trabajar con la misma y subir los cambios utilizando comandos GNU.
- *El cliente y servidor pueden estar en la misma máquina.*
- *Puede mantener distintas ramas de un proyecto:* Una versión difundida de un proyecto de programa puede formar una rama y ser utilizada para corregir errores.
- *Emite reportes útiles y accesibles al cliente:* Comparar diferentes versiones de archivos, solicitar una historia completa de los cambios, o sacar una "foto" histórica del proyecto son opciones del SVN.
- *Viene integrado por defecto en Eclipse.*

Una de las principales ventajas es que, como muchos proyectos de código abierto, permite el acceso de lectura anónimo, significando que los clientes pueden sacar y comparar versiones sin necesidad de teclear una contraseña, solamente el ingreso de cambios requiere una contraseña en estos casos. Otra ventaja se incluye en que es un programa totalmente gratis que contiene en sí los principios del software libre amparado bajo la licencia GPL.

Constituye una limitante que los archivos en el repositorio sobre la plataforma CVS no pueden ser renombrados, estos deben ser eliminados y luego volver a agregarlos con el nuevo nombre. El protocolo CVS no provee una manera de que los directorios puedan ser eliminados o renombrados, cada archivo en cada subdirectorío debe ser eliminado y re-agregado con el nuevo nombre. Soporte limitado para archivos Unicode con nombres de archivo no ASCII.

1.2.6. SVN

Subversion o SVN como también se le conoce, es un software para control de versiones diseñado específicamente para reemplazar al popular CVS, el cual, para muchos expertos, posee varias deficiencias. Se encuentra bajo licencia de tipo Apache/BSD y es software libre.

Una característica importante de Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.

Otras características son:

- *Usa el modelo Copy-Modify-Merge.*
- *Trata siempre de optimizar el uso de ancho de banda y de hacer todas las operaciones posibles sin necesidad de conexión:* Esto se hace guardando una copia intacta del estado del repositorio y otra para que modifiquemos. De esta manera hay algunas operaciones que se pueden hacer en modo desconectado (Entre ellas la más importante el diff).
- *El aplicar un changeset es proporcional al tamaño de los cambios y no de los archivos enteros como pasa en otros SCMs.*

Uno de los conceptos más fuertes del Subversion es que es un filesystem versionado, o sea un sistema de archivo organizado jerárquicamente de directorios y sus contenidos, por lo que provee las operaciones comunes referentes al mismo como copiar, crear, mover, borrar archivos o directorios.

Una de las ventajas más agradables del mismo es que goza de una alta disponibilidad de clientes, tanto para distintos sistemas operativos y plataformas como de consola o gráficos. Además maneja otros detalles interesantes como el tipo de fin de línea, puede reconocer si el archivo a versionar es binario y de qué tipo o si es ejecutable. Todas estas características hacen de Subversion la herramienta ideal a utilizar.

1.2.7 StatSVN

En el año 2006 surgió en Internet un proyecto de software libre nombrado StatSVN (Estadísticas basadas en Subversion) liderado por Jason Kaley y Gunter Mussbacher. Hecha sobre tecnología Java, la herramienta que lleva el mismo nombre retorna información que existe en un repositorio de Subversion y es capaz de generar varias tablas y gráficos que describen la fase de implementación de un proyecto dado (Línea del tiempo sobre las líneas de código, contribución de cada programador,

etc). La versión actual del proyecto (0.4.0) es capaz de generar una suite estática HTML o documentos XDOC donde aparecen contenidas las tablas y gráficas descritas. La herramienta, de código abierto, está liberada bajo los términos de la licencia LGPL basada en StatCVS, otra herramienta similar pero destinada a repositorios CVS, ambas disponibles en SourceForge.net, conocido sitio donde se alojan cientos de proyectos de distribución libre.

Según [Kealey y Mussbacher 2006] una administración de proyectos efectiva requiere de un software preciso y confiable, presto a hacer informes de decisiones sobre el proyecto en cuestión. Se propone StatSVN como herramienta de ejemplo en la obtención de estadísticas que ayuden a disponer de información relativa al proceso y estado de un proyecto, pudiendo medir incluso la productividad de programadores durante la fase de implementación de un proyecto de desarrollo. StatSVN proporciona una métrica basada en LOC para proyectos de software que utilicen SVN como herramienta de control de configuración.

1.3 Tecnologías y Herramientas propuestas para el desarrollo del sistema.

Existen múltiples variantes para el desarrollo de software en nuestros días. Las tecnologías y herramientas propietarias han invadido este mercado y desde hace más de una década, una nueva corriente ha surgido haciéndole frente, demostrando ser una variante interesante para aquellos países en vías de desarrollo. “La aparición del software libre en este contexto es bien complicada y atrevida, pretende cambiar un modelo tradicional y se proclama como una alternativa de calidad superior” [Rodríguez 2006]

1.3.1 La tecnología Java. Entornos de Desarrollo Integrados.

Impulsada por Sun Microsystems y bajo licencia de software GPL/Java Community Process [Stallman 2006], Java es reconocida mundialmente como una tecnología madura, muy eficaz y sorprendentemente versátil. La portabilidad y la seguridad son dos criterios que apuestan a su favor. Está compuesta por dos partes fundamentales: la plataforma y el lenguaje de programación.

La plataforma

Una plataforma es descrita como la combinación del hardware y software que hace funcionar al ordenador. La característica más notoria de la plataforma Java es que solo hace referencia al software que se ejecutará sobre las restantes plataformas de hardware. Esto le da un sentido muy amplio de independencia, razón por lo que es tan valorado en el mundo de la informática, ya que se deseará escribir software lo más portable posible. Java permite este requisito.

Algunas de las principales ventajas de esta plataforma es que, prácticamente todo el Java de Sun es software libre (a excepción de la biblioteca de clases de Sun que se requiere para ejecutar los programas). Aunque la compilación en código máquina nativo también es posible, las aplicaciones Java están típicamente compiladas en bytecode lo cual se traduce a una potente herramienta multiplataforma. La máquina virtual (*JVM-Java Virtual Machine*) permite esta portabilidad a todo tipo de hardware y para una gran cantidad de sistemas operativos, Linux entre ellos.

El lenguaje

Java es un potente lenguaje de programación. Su compañía Sun lo describe como “simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico” [Jalón 1999]

Desde sus inicios Java surgió como un lenguaje de programación ideal para entornos de red, esto explica por qué fue pensado para ser seguro. “Los programas escritos en Java no pueden ser atacados por virus, pues para que estos tengan efecto deben utilizar rutinas de acceso directo a memoria, que Java no tiene [Becerril 1998].” Tiene amplio uso para Internet por lo integrado que tiene el protocolo TCP/IP.

Otro detalle importante es que “Java incorpora muchos aspectos que en cualquier otro lenguaje son extensiones propiedad de empresas de software o fabricantes de ordenadores (threads, ejecución remota, componentes, seguridad, acceso a bases de datos, etc.)” [Jalón 1999]. Se considera de primerísimo nivel al ser diseñado para que un programa escrito en él sea ejecutado independientemente de la plataforma (sistema operativo, hardware y software) sobre la que se esté implementando.

Para él se han construido varios IDEs (Entornos de Desarrollo Integrados) que lo soportan e incluyen numerosas herramientas que extienden su funcionalidad.

Eclipse

Eclipse fue desarrollado originalmente por IBM como sucesor a su familia de herramientas de VisualAge. Ahora es impulsado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Considerado por muchos como la principal estructura programática de open source, es un IDE muy ventajoso para las actividades de desarrollo de la tecnología Java ya que ofrece un compilador interno para este lenguaje y un modelo completo de sus archivos fuente, permitiendo técnicas avanzadas de refactorización y análisis de código. Provee interacción con otras aplicaciones informáticas que aportan funcionalidades, también conocidas como *plugins* o *módulos*, permitiendo la reutilización y extensibilidad. Con la utilización de los mismos, Eclipse puede adaptarse para desarrollo con otros lenguajes de programación como C/C++ y Python. También permite la creación de plugins personalizados para luego poder ser añadidos al entorno. Esto no sucede en la mayoría de los IDEs, donde la utilidad de la herramienta viene con un valor estático agregado, como es el caso de Visual Studio .NET, desarrollado por Microsoft bajo sistema operativo Windows, o incluso NetBeans, de Sun Microsystems, donde se incluyen solamente ciertos paquetes extras, también llamados Add-on Packs, específicos para llevar a cabo las tareas primordiales de desarrollo.

Catalogado como “una especie de herramienta universal” por el proyecto que da su nombre, Eclipse provee al desarrollador de plugins y plataformas (frameworks) muy ricas para el desarrollo de aplicaciones gráficas, manipulación de modelos de software, aplicaciones para la Web y acceso a información en bases de datos.

1.3.2 Plugins & Frameworks.

Al ser Eclipse un IDE en constante crecimiento y al que pueden aportar numerosos programadores alrededor del mundo, la Eclipse Foundation ha ido propiciando el desarrollo de numerosos proyectos para la creación de plugins y frameworks. A continuación se exponen algunos de estos proyectos, cuyos productos tanto Open Source como comerciales, están disponibles para el resto de la comunidad desde la Web.

GEF (Graphic Editing Framework - Plataforma para la edición gráfica)

Es un plugin de eclipse para el desarrollo de editores visuales que pueden ir desde procesadores de texto hasta editores de diagramas UML, interfaces gráficas para el usuario (GUI - Graphic User Interface – Interfaz gráfica de usuarios), etc. Dado que los editores realizados con GEF "viven" dentro de eclipse, además de poder ser usados conjuntamente con otros plugins, hacen uso de su interfaz gráfica personalizable y profesional.

EMF (Eclipse Modeling Framework – Plataforma de Modelado Eclipse)

Una plataforma de modelado y generación de código para construir herramientas y otras aplicaciones basadas en un modelo de datos estructurado desde una especificación de modelo descrita en XML (XML Metadata Interchange - XML de Intercambio de Metadatos).

VE (Visual Editor – Editor Visual)

El proyecto Eclipse Visual Editor es una plataforma de desarrollo abierta para el suministro de los marcos de creación de constructores de GUI. Ejemplo de estos marcos lo constituyen las implementaciones de herramientas extensibles como Swing/JFC y SWT/RCP. Con esto se elimina el arraigado mito de que “Eclipse no tiene soporte para desarrollo de aplicaciones de escritorio”.

WTP (Web Tools Platform – Plataforma para herramientas Web)

Extiende la plataforma Eclipse con herramientas para desarrollar aplicaciones Web en Java EE. Está compuesta de: Editores de fuentes para HTML, Java Script, CSS, JSP, SQL, XML, DTD, XSD y WSDL; Editores gráficos para XSD y WSDL; proyectos de naturaleza Java EE, constructores y modelos y un navegador de Java EE; un explorador y asistente para servicios Web y una herramienta de pruebas WS-I (*Web Services Interoperability*); herramientas para acceso a base de datos, filtrado y modelos; y herramientas para manejo de servidores de pruebas unitarias.

Subclipse

Subclipse es un plugin para Eclipse que proporciona una serie de herramientas mediante las cuales se puede interactuar con un sistema de control de versiones de tipo Subversion. Los proyectos que emplean el mismo mapeados en Eclipse realizan su configuración desde una perspectiva denominada *SVN Repository*. La configuración de dichos mapeos se realiza de forma similar a los mapeos de sistemas de control de versiones CVS.

Subclipse proporciona una interfaz gráfica parecida al que Eclipse ofrece para su interacción con su sistema de control de versiones CVS. Su principal responsabilidad se centra en permitir al usuario tener sus proyectos en un repositorio común al equipo de desarrollo donde podrá trabajar con dichos proyectos, usando las funcionalidades que los sistemas de control de versiones Subversion ofrecen.

Crystal Reports for Eclipse

Desde mediados del año 2006 el proveedor líder mundial en soluciones de inteligencia de negocios Business Objects, ofreció la disponibilidad de Crystal Reports para el entorno de desarrollo Eclipse.

Mediante este nuevo plugin, Crystal Reports ofrece a los programadores un diseño para informes y un ambiente de implementación cien por ciento basado en Java. Esta herramienta facilita enormemente a las compañías la creación de soluciones completas para la generación de reportes, con las que podrán monitorear y optimizar su desempeño empresarial.

Utilizando Crystal Reports dentro del popular IDE Eclipse se podrá:

- Acceder con facilidad a los datos y formatearlos para convertirlos en información dinámica.
- Reducir al mínimo el esfuerzo de codificación manual asociado a la integración de datos dinámicos dentro de una página JSP.
- Reducir el trabajo de mantenimiento de las aplicaciones desvinculando los aspectos de presentación y conectividad de datos del núcleo de código de la aplicación.

Hibernate

Hibernate ha ganado una amplia aceptación dentro de la comunidad de desarrollo de la tecnología Java por su integración y facilidad para el trabajo con las bases de datos. Aunque es un framework de mapeo objeto-relacional para esta tecnología, también está disponible para .NET con el nombre de NHibernate. Es software libre y distribuido bajo los términos de la licencia GNU LGPL.

Ofrece grandes facilidades al poder declarar objetos a partir del esquema relacional originado en una base de datos. Para esto facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación utilizando archivos declarativos XML que permiten establecer estas relaciones.

Una de las ventajas de la utilización de este framework es que libera al desarrollador del manejo explícito de los datos que resultan de la ejecución de sentencias SQL, ganando así en comprensión. Para esto Hibernate convierte los datos entre los tipos utilizados por Java y los definidos por SQL manteniendo la portabilidad entre todas las bases de datos con un ligero incremento en el tiempo de ejecución. Esto es factible ya que incluye una caché de consultas con la posibilidad de obtener rápidamente resultados que ya habían sido consultados previamente. Está diseñado para ser flexible en cuanto al sistema de tablas utilizado y así poder adaptarse a su uso sobre una base de datos ya existente. Tiene la funcionalidad de crear la propia base de datos a partir de la información de diseño disponible. Hibernate ofrece además un lenguaje de consulta de datos llamado HQL (Hibernate Query Language) al mismo tiempo que una API (conocida como Criteria) para construir las consultas programáticamente.

Otras características de este framework son que se integra dentro de los servicios de una plataforma J2EE siendo capaz de obtener conexiones a través de objetos DataSource vía JNDI (Java Naming and Directory Interface), ejecutar sus transacciones dentro de un entorno JTA (Java Transaction API). Aparte de esto ofrece también vía de integración con otras tecnologías, por ejemplo JMX (Java Management Extensions) y JCA (Java Cryptography Architecture). [Informática 2006]

1.3.3 Sistemas gestores de Base de Datos (SGBD)

Los sistemas gestores de base de datos brindan un importante servicio al ser los encargados de almacenar información y permitir a los usuarios su disponibilidad en caso de ser necesitada. Son numerosos los software que en la actualidad ofrecen esta utilidad. Entre SGBD libres, gratuitos y comerciales podemos escoger cuál de ellos será el más conveniente, dependiendo de factores tales como seguridad, rapidez en acceso a la información, poder de réplica e integridad.

Dentro de sistemas gestores de base de datos comerciales tenemos: Microsoft SQL Server, Microsoft Access y Oracle como los más conocidos, aunque también se mencionan: IBM Informix, Interbase de CodeGear, filial de la Borland, MAGIC, Fox Pro, y Paradox. Sobre SGBD gratuitos podemos hacer mención a: Microsoft SQL Server Compact Edition y Sybase ASE Express Edition para Linux. Por último y más aceptados últimamente por las ventajas del uso del software libre a nivel mundial, están los SGBD libres: MySQL, SQLite, Apache Derby, Firebird y PostgreSQL.

PostgreSQL

PostgreSQL es un sistema gestor de base de datos relacional altamente aceptado en calidad de software libre. Liberado bajo licencia BSD, está dirigido por una comunidad de desarrolladores y organizaciones comerciales que trabajan en su constante perfeccionamiento. Esta comunidad es denominada PGDG (PostgreSQL Global Development Group). PostgreSQL cuenta con características que lo hacen ser preferido por encima del resto. Entre ellas está su alto rendimiento, estabilidad y capacidad para lidiar con grandes volúmenes de datos en sus almacenes. El número de prestigiosas empresas que aceptan su uso crece exponencialmente cada día, incluyendo agencias bancarias donde la seguridad de los datos ejerce un papel primordial.

Las características que se le asocian a este producto, basada en opiniones de reconocidos especialistas a lo ancho de Internet y detalles publicados por el propio sitio Web oficial de la comunidad son:

- Posee gran escalabilidad: Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta (en algunos benchmarks se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL).
- Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz y ofreciendo soluciones en campos en los que por ejemplo MySQL no podría.
- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle.
- Ventajosa capacidad de almacenamiento. Según el sitio oficial [PostgreSql.org 2007] Postgre tiene tamaño ilimitado para la base de datos y número de filas en una tabla, (dependiendo de la capacidad del disco duro claro está), un máximo de 32 TeraBytes por tabla, 1Gb por campo y 400 Gb por fila, lo que rompe con el mito muy difundido entre sitios de Internet que afirman que solo posee 8 Kb para esta última.

Por otra parte, según Lockhart [Lockhart 1998] este gestor ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema: *clases*, *herencia*, *tipos*, *funciones*.

Otras características aportan potencia y flexibilidad adicional:

- Restricciones (Constraints)
- Disparadores (triggers)
- Reglas (rules)
- Integridad transaccional

Estas cualidades ubican a Postgre en la definición de las Bases de Datos identificadas como objeto-relacionales. Nótese que éstas son diferentes de las referidas como orientadas a objetos, que en general no son bien aprovechables para soportar lenguajes de Bases de Datos relacionales tradicionales. Postgre tiene algunas características que son propias del mundo de las bases de datos orientadas a objetos. De hecho, algunas Bases de Datos comerciales han incorporado recientemente características en las que Postgre fue pionera. Sin duda un gestor de base de datos confiable y escalable.

1.4 Fundamentación de la Metodología de desarrollo. Herramientas.

Según Jacobson “los desarrolladores de software no trabajan de manera independiente, interaccionan unos con otros y con los usuarios” de ahí la necesidad de establecer una metodología rectora de un proceso que “promete hacerse más crítico, especialmente en empresas u organizaciones en las cuales los sistemas de software son esenciales”. [Jacobson 2000]

Hacer software es una tarea de mucha dedicación, en la que no se está exento de cometer fallos; se corren muchos riesgos y puede llegar a ser un proceso difícil de controlar si no existe como puente una metodología que proporcione una guía acertada de las actividades a desarrollar para obtener un producto de calidad, escalable y de fácil reutilización. Entre las metodologías de desarrollo de software que hoy se consideran más importantes se encuentran: XP, MSF y RUP.

Cada metodología ofrece un marco adecuado para la construcción de software basado en sus requisitos. Así por ejemplo, XP (eXtreme Programming) es la más destacada de los *procesos ágiles de desarrollo de software*, la cual ofrece un enfoque adecuado para aquellos proyectos de corto plazo. La misma plantea que adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos [Beck 2005]. MSF (Microsoft Solutions Framework) por su parte puede ser aplicado tanto en proyectos grandes como pequeños pero orientado a construir e implantar sistemas empresariales distribuidos basados en herramientas y tecnologías de Microsoft, lo cual para nuestro país constituye una desventaja. RUP por otro lado es una metodología adoptada universalmente y es muy bien recibida por aquellos equipos de desarrollo con poca experiencia y muy bien utilizada en aquellos con alto grado de maestría en la construcción de software. A esto podemos añadir que es más adaptable para proyectos de largo plazo, los cuales aportan más ingresos y requieren de una guía acertada durante el ciclo de vida de desarrollo del software. Expuesto lo anterior, esta última será la metodología a emplear para el desarrollo del sistema que da a estudio esta tesis.

1.4.1 Proceso unificado de desarrollo (RUP)

El proceso unificado de desarrollo o Rational Unified Process (RUP) como también se le conoce, ofrece todo un marco para el desarrollo del software con carácter iterativo e incremental. Estas dos últimas palabras le han hecho famoso al punto de conferirle ser la metodología más refinada y documentada, donde sus raíces se montan a más de 30 años de unir experiencias de distintas compañías dedicadas a la producción de software, cuyos servicios fueron comprados por la poderosa empresa Rational, de donde obtiene su nombre.

Según Jacobson, Booch y Rumbaugh el Proceso Unificado es más que un simple proceso, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto [Jacobson 2000], por lo que constituye una metodología unificada, cohesiva y comprensiva para el desarrollo de sistemas de software. RUP es Orientada a Objetos y proporciona una línea base para todos los aspectos y estados del desarrollo de un software. Esto redundará, si se cumplen bien sus directivas, en un producto de alta calidad, obtenido en el tiempo previsto y dentro del presupuesto asignado.

El principio por que se rige RUP es que todo este proceso está compuesto por ciclos que bien se pueden repetir a lo largo del tiempo de vida donde se desarrolla el software. Cada uno de estos ciclos contiene cuatro fases bien definidas: Inicio, Elaboración, Contrucción y Transición. Cada fase tiene como objetivo enfocar al equipo de desarrollo en cumplir una tarea determinada y para esto tendrá asociado varios hitos. El cumplimiento de estos hitos dará la señal oportuna para moverse a la siguiente fase. Por su parte, sobre cada fase tienen lugar los cinco flujos de trabajo: requisitos, análisis, diseño, implementación y prueba. Al finalizar el ciclo se muestra un release del producto.

RUP tiene tres características fundamentales:

- *Iterativo e incremental*: Se dividen los proyectos grandes en proyectos más pequeños o mini-proyectos, cada mini-proyecto comprende una iteración que produce un incremento. Una iteración puede abarcar todos los flujos de trabajo en el proceso.
- *Guiado por los casos de uso*: Emplea los casos de uso para manejar el desarrollo del proceso desde el inicio del proyecto hasta el despliegue.
- *Centrado en la arquitectura*: El proceso busca entender la estática más significativa y los aspectos dinámicos en términos de la arquitectura del software. La arquitectura es una función de las necesidades de los usuarios y se captura en la esencia de los casos de uso.

Las fases que propone RUP se muestran a continuación:

- *Inicio*: El propósito de esta fase es definir la visión y alcance del proyecto. Se identifican todas las entidades externas con las que se trata (actores) y se define la interacción de las partes a un alto nivel de abstracción. Sus hitos principales: las partes interesadas deben acordar el alcance y la estimación de tiempo y costo, además de la comprensión de los requerimientos plasmados en casos de uso.

- *Elaboración:* Durante la misma se especifican en detalle la gran parte de los casos de uso y se diseña la arquitectura base del sistema. Los hitos: ¿es estable y realista la visión del producto, arquitectura y plan de proyecto? ¿están de acuerdo con el plan todas las personas involucradas? ¿las pruebas de ejecución convencen de que los riesgos han sido abordados y resueltos?
- *Construcción:* El software es diseñado, escrito y probado. El producto se mueve desde una visión de línea base de arquitectura a un sistema apto para ser entregado al cliente. El énfasis está en la producción eficiente y no tanto ya en la creación intelectual. El hito principal: obtener un producto beta que debe decidirse si puede ponerse en ejecución sin mayores riesgos.
- *Transición:* Se entrega el sistema a los usuarios finales teniendo como meta principal la satisfacción del cliente. Una vez instalado el producto surgirán nuevos elementos que implicarán nuevos desarrollos (ciclos). Obtener autosuficiencia de parte de los usuarios con el sistema a utilizar constituye el hito fundamental a cumplir en esta fase.

La última fase analizada culmina con un postmortem dedicado a aprender lecciones para ciclos futuros, contribuyendo a afianzar conocimientos y buscar refinamientos en la aplicación de RUP.

1.4.2 Lenguaje de modelado

Las ramas más antiguas de la ingeniería han encontrado útil desde hace tiempo representar los diseños mediante dibujos. Desde los inicios del software, los programadores han encapsulado sus conceptos en diversos tipos de dibujos o, más ampliamente, de modelos. La comunidad del software precisa de una forma de comunicar sus modelos, no solo entre los miembros de un proyecto, sino a todas las personas involucradas en él y, con el paso del tiempo, a desarrolladores de futuras generaciones. [Jacobson 2000]

Los sistemas construidos con programación orientada a objetos (POO) proponen un mejor diseño, más comprensible, del entorno de desarrollo. Conocer un lenguaje orientado a objetos es un primer paso necesario pero insuficiente para crear sistemas de objetos. Se requiere además analizar y diseñar desde la perspectiva de los objetos. UML (Unified Modeling Language) es el Lenguaje Unificado de Construcción de Modelos con que se construyen sistemas por medio de conceptos orientados a objetos. El mismo ofrece un standard de modelado para procesos de negocio, funciones del sistema, aspectos bien definidos como esquemas de bases de datos, expresiones de lenguajes de programación y componentes de software reutilizables. Con UML se pueden modelar desde complejos sistemas de software para instituciones hasta aquellos basados en web o de tiempo real. Es un lenguaje muy expresivo, que cubre todas las vistas necesarias para desarrollar un sistema.

1.4.3 Visual Paradigm for UML

Desarrollada con tecnología Java, Visual Paradigm for UML es una poderosa herramienta profesional que se distingue por su fácil integración con el IDE Eclipse. Como su nombre lo indica utiliza UML como lenguaje de modelado. Soporta el ciclo de vida completo del desarrollo del software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue, ayudando a catalizar la construcción de mejores aplicaciones, con calidad y a un menor coste. Está disponible en múltiples ediciones, cada una dedicada a necesidades específicas: Enterprise, Professional, Community, Standard, Modeler y Personal.

Aunque Visual Paradigm puede descargarse gratuitamente desde la Web en una versión Trial que permite utilizarlo con acceso a todas sus funcionalidades por tiempo ilimitado, a la hora de generar diagramas, los exporta con una marca de agua de fondo que numerosas veces dificulta la lectura, situación que se elimina al ser adquirida la licencia original suministrada por la Empresa. Por tanto, Visual Paradigm for UML no es considerado software libre, pero por su portabilidad entre plataformas Windows, Linux, Mac OS y una licencia de coste menor comparada con otras como por ejemplo Rational Rose, la hacen ser preferida como herramienta de modelado en muchas instituciones donde se desarrolla software de calidad pues está a la altura de predecesoras como ArgoUML y el ya mencionado Rational Rose de IBM.

Entre los múltiples diagramas que permite elaborar se encuentran:

- Diagrama de casos de uso.
- Diagrama de clases.
- Diagramas de secuencia.
- Diagramas de colaboración.
- Diagrama de estado.
- Diagrama de actividad.
- Elementos de ayuda.
- Diagramas de componentes.
- Diagramas de implementación

El modelado de procesos del negocio entra también en las operaciones que esta robusta herramienta brinda. Un detalle novedoso es que permite capturar actores y casos de uso a partir del análisis textual de los requisitos del software, así como la creación de distintos diagramas relacionados con este proceso. Por último y al igual que Rational Rose, permite extender la representación de la futura

aplicación con la generación del modelo de datos de la base de datos a partir del modelo de objetos con su respectiva ingeniería inversa.

Lista de características adicionales:

- Soporte de UML versión 2.1
- Estereotipos Web.
- Modelado colaborativo con CVS y Subversion.
- Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XMI.
- Ingeniería de ida y vuelta.
- Ingeniería inversa - Código a modelo, código a diagrama.
- Ingeniería inversa Java, C++, Esquemas XML, XML, .NET exe/dll, CORBA IDL.
- Generación de código - Modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Diagramas EJB - Visualización de sistemas EJB.
- Generación de código y despliegue de EJB's - Generación de beans para el desarrollo y despliegue de aplicaciones.
- Diagramas de flujo de datos.
- Soporte ORM - Generación de objetos Java desde la base de datos.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- Importación y exportación de ficheros XMI.
- Integración con Visio - Dibujo de diagramas UML con plantillas (stencils) de MS Visio.
- Editor de figuras.

Conclusiones del Capítulo

En este capítulo se hace un estudio sobre la situación actual en la que se encuentra la estimación del esfuerzo en proyectos de desarrollo de software.

La piedra angular en la determinación del esfuerzo radica en una definición de la medición de los tamaños desarrollados. Estos datos, junto a otros del entorno productivo, permiten la formación de indicadores de productividad.

El presente trabajo ha desplegado la implementación de una versión inicial de herramienta integrada a la gestión de la configuración para medir automáticamente los tamaños producidos en la fase de construcción del software, y una versión inicial del Sistema de Gestión de Indicadores de Productividad, donde como parte del tamaño producido se considera también la etapa de análisis. Ambas herramientas sientan las bases para la implementación de un framework más general, que incluya la automatización de la estimación del tamaño y esfuerzo requeridos para la solución de un nuevo problema, y todo un sistema de Tablero de Mando o Cuadro de Mando Integral para la Organización del Software.

Elegir la vía de desarrollar una herramienta sobre software libre, integrada a un sistema de control de versiones, permite obtener un sistema de software caracterizado por:

- Fácil acceso a indicadores de productividad de esfuerzo.
- Capaz de emitir reportes sobre estos indicadores en una Universidad Productiva.
- Independencia de licencias privativas de software.

Se opta por el uso de Subversion como sistema de control de versiones y de StatSVN como apoyo en la obtención de los tamaños; la tecnología Java y Eclipse como herramientas de desarrollo; PostgreSQL en su papel como gestor de base de datos para el almacenamiento de la información necesaria. El enfoque iterativo de RUP nos ha guiado en la selección del núcleo básico de funcionalidades a implementar en las primeras versiones de nuestras herramientas.

Se adopta UML como lenguaje para representar los modelos de análisis y diseño y Visual Paradigm for UML como herramienta para la diagramación de dichos modelos.

CAPÍTULO 2: Análisis y Diseño de las Herramientas Propuestas.

2.1 Introducción

En este capítulo se describe la concepción del proyecto de desarrollo de software.

Se enumeran los requisitos y se documentan los artefactos del análisis y diseño de las herramientas:

- Sistema de Gestión de los Indicadores de Productividad. (SGIP)
- Servicio de Integración de Estadísticas del StatSVN (SVNStatBash)

2.2 Modelo del Negocio

El negocio surge a partir de la necesidad de conocer indicadores de productividad de esfuerzo dentro del marco de la Universidad de las Ciencias Informáticas donde se desarrollan múltiples proyectos productivos para la elaboración de productos software.

Cuando a esta institución se le solicita la construcción de un producto informático se constituye un nuevo proyecto productivo, donde se designa a un equipo que se encargará de llevar a cabo el ciclo de vida de desarrollo y entregar al cliente el producto solicitado.

Cada equipo de proyecto cuenta con un jefe de proyecto, el cual dirige todo el proceso de producción y coordina el trabajo de los analistas, diseñadores, probadores, documentadores y programadores que forman parte del mismo.

Durante el proceso de desarrollo del producto software se generan documentos que avalan las expectativas que se pretenden alcanzar. Entre estos documentos se encuentran aquellos relacionados con la descripción de los casos de uso, que guía la elaboración de los componentes asociados, en los que intervendrá aquel o aquellos programador(es) designados por el Jefe de Proyecto.

Se dice que el producto software alcanza una versión apta para su entrega y uso cuando se implementan y prueban todos los casos de uso especificados en los documentos que los describen. Es usual hacer un documento de especificaciones por módulo del producto software, el cual contiene

varios casos de uso que se van codificando en varios archivos de código fuente durante la construcción del producto.

Dos aspectos esenciales atendidos por la dirección de los proyectos son:

- *Esfuerzo a emplear para el desarrollo.*
- *Productividad de los trabajadores.*

En particular se emplea la fórmula: [Pressman 2000]

$$\text{ESFUERZO} = \text{TIEMPO} \times \text{PERSONAS}$$

La *productividad de los trabajadores* es la relación de cuánto tamaño se produce en función del esfuerzo, y se define con la fórmula:

$$\text{PRODUCTIVIDAD} = \text{TAMAÑO} / \text{ESFUERZO}$$

En el capítulo 1 se examinaron diferentes métricas que han sido utilizadas para medir el tamaño del software, entre ellas se describieron los puntos de función, puntos de casos de uso y de característica, que si bien suponen una buena aproximación al arte de evaluar productos de software ya que “hacen que la medida sea independiente del lenguaje o herramienta utilizada en el desarrollo del proyecto” [Albrecht 1979] [Sparks y Kapczynski 1999], adolecen de la facilidad para hacer el cómputo automático y requieren intervención del hombre para mejorar la precisión con que se estima.

Además, al haber identificado la herramienta StatSVN, que permite recolectar estadísticas a partir de un repositorio de Subversion, se decidió basarse en la medición del tamaño con líneas de código, con las que se puede medir además la *productividad de los trabajadores*. De esta forma, tanto el tiempo como el tamaño producido se calculan automáticamente durante el trabajo y no a partir de la actividad controlada por personas que registren los eventos durante el desarrollo. Hoy día, la UCI estima la *productividad de sus trabajadores* basado en las preguntas que el analista hace al programador en el módulo donde trabaja. No existen por lo general herramientas integradas al control de las versiones y por ende el grado de subjetividad con que esta estimación se realiza es alto.

El uso de herramientas que apoyen la Gestión de Configuración de Software es imprescindible para un desarrollo organizado y responsable y ellas mismas nos ofrecen el mecanismo adecuado para el control de la actividad sobre las líneas de código (tamaño) y el tiempo.

Teniendo en cuenta que un producto de software es infinitamente versionable, el tamaño y tiempo de producción deben ser calculados a partir de las fechas de obtención de las versiones de sus componentes y teniendo en cuenta la fecha en que se comenzó el proyecto como inicio absoluto.

Se hace necesario poder medir el tamaño de todos los componentes desarrollados, pues con esto es posible obtener el tamaño del producto en su totalidad. Para este propósito el Subversion cuenta en su base de datos del repositorio información que, con ayuda del StatSVN, puede ser accesible y contribuir a la obtención del tamaño.

El indicador de productividad (tamaño producido por persona-mes) se deberá poder obtener para:

- Un equipo.
- Un producto.
- Un programador.
- Toda la universidad.

Por otra parte el Departamento de Calidad de la Universidad está especialmente interesado en controlar el tamaño de los productos que ya se han construido en la organización, ya que partiendo de la obtención de estos indicadores se podrá entonces estimar el tiempo y recursos a emplear en la realización de nuevos proyectos semejantes a los ya realizados. Esto redundará en una mejor planificación y gestión de los proyectos actuales.

La productividad ha de reflejar fielmente el nivel de aceptación de los productos elaborados por parte de los clientes, de esta manera, deberán dejarse de considerar aquellos componentes, que por el nivel de defectos que poseen deben borrarse del repositorio y por ende dejarán de ser tenidos en cuenta en los tamaños entregados.

El modelo del negocio que debe ser implementado se muestra en la figura 2.1

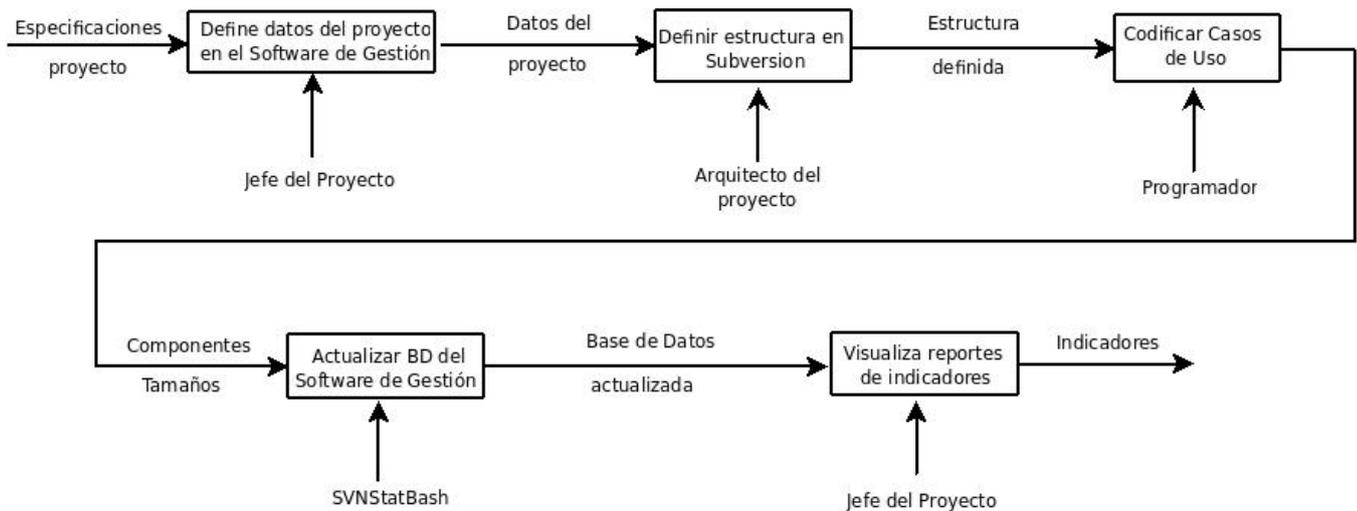


Figura 2.1 Proceso de obtención de indicadores utilizando control de versiones sobre código fuente y datos iniciales sobre documentos del proyecto.

2.2.1 Descripción del proceso

El proyecto consiste en la construcción de un producto software, donde se utiliza una base de datos para guardar los datos del proyecto y un repositorio de Subversion para guardar / consultar los componentes construidos del producto.

El jefe del proyecto define los datos generales del proyecto en el software de gestión de indicadores. Los datos a definir deben contener, entre otros: la fecha de inicio del proyecto, qué nombres se le ha dado a los documentos de especificaciones de casos de uso, la cantidad de personas que trabajan en el análisis de cada documento de especificaciones. Esta información debe ser almacenada en la base de datos del sistema de gestión.

El arquitecto del proyecto define la estructura arbórea del repositorio para la construcción del código fuente, para ello define los nodos y asocia qué documentos de especificaciones se usarán para desarrollar código en cada nodo. La estructura en principio es libre, es decir, el arquitecto puede definir los nodos en función de la plataforma de desarrollo y utilizando una organización conveniente al proyecto y a las capacidades de sus programadores. Por ejemplo se pueden designar varios nodos al desarrollo de los casos de uso de un mismo documento de especificaciones cuando los nodos se utilizan para “las capas” del software, pero es incorrecto almacenar en el mismo nodo códigos correspondientes a documentos diferentes ya que la localización futura de los componentes para reuso se dificultaría. Luego también debe definir qué programadores van a trabajar en cada nodo. Como quiera que va a existir información sobre los casos de uso que se especifican en cada documento de

especificaciones, resultará posible determinar estadísticas de desarrollo por casos de uso contenidos en los documentos.

La base de datos del sistema de gestión de indicadores se actualiza diariamente a través de un servicio de integración de estadísticas del StatSVN (SVNStatBash), desarrollado en este trabajo, que inserta los tamaños del desarrollo de la jornada de ese día, utilizando la información que proporciona StatSVN sobre el repositorio de Subversion en cuanto a los nodos de su estructura arbórea.

El sistema de gestión usa los datos del proyecto definido para calcular el tiempo total de desarrollo de cada documento de especificaciones, sumando el tiempo que se demoró la etapa de análisis (desde la fecha de inicio del proyecto hasta la fecha de registro del documento de especificaciones) al tiempo que se demora su desarrollo (tomando el tiempo desde la fecha en que se registró el documento de especificaciones hasta que se consulta al repositorio). Una vez obtenidos los datos de tamaños y de tiempos, el sistema de gestión podrá ser capaz de emitir los indicadores de productividad y de esfuerzo de la organización.

Por ende, la base de datos actualizada funciona como *Línea Base de la Organización del Software*.

Asumiendo lo siguiente para SVN:

Un Componente = Un fichero (con código fuente)

La manera en la que se calcula el *Tamaño de un Producto* (Tam (P)) es la siguiente:

$$\text{Tam (P)} = \text{Tam (Comp1)} + \text{Tam (Comp2)} + \dots + \text{Tam (Compn)}$$

Donde:

Tam (Comp i): Tamaño del componente i

n: Cantidad de componentes del producto

Para tomar en cuenta un producto de software en su versión N, se consideran las correspondientes versiones N de los componentes que lo forman.

El *tamaño* del componente se expresa como la cantidad de líneas de código que tiene el fichero, información que es perfectamente obtenible a partir del repositorio.

Llevar líneas base por plataforma de software tiene sentido usando la línea de código como medida del tamaño ya que se controlarían proyectos que usan los mismos tipos de recursos asociados al lenguaje.

La fecha de inicio del proyecto rige el inicio de la etapa de análisis. Durante el análisis se producen los documentos de especificaciones de casos de uso.

La fecha de registro del documento de especificaciones del caso de uso rige el inicio del tiempo de desarrollo para ese documento.

Por ello, la diferencia en días entre la fecha de inicio del proyecto y la fecha de registro del documento de especificación del caso de uso que lo contiene, es el *tiempo de análisis* del módulo especificado en dicho documento.

El *tiempo del desarrollo* de un módulo se obtiene considerando la fecha de registro del documento de especificación y la fecha actual, donde se están analizando las estadísticas.

Entonces el Tiempo Total de Producción de un Módulo es:

Tmpo (Módulo i) = Tiempo de análisis (Módulo i) + Tiempo de desarrollo (Módulo i)

Donde:

Tiempo de análisis = Fecha de inicio del producto – Fecha de registro de documento de especificación
--

Tiempo de desarrollo = Fecha de registro del documento de especificación – Fecha actual.
--

Es importante hacer notar que el tiempo de desarrollo se actualiza cada día, hasta que se defina una fecha de liberación del producto en su totalidad. Por consiguiente la productividad cambia día a día con los tamaños de los componentes registrados y las personas participantes, y al cierre del proyecto se puede contar con las estadísticas finales.

2.3 Requisitos no funcionales de las herramientas propuestas.

- Requerimientos de software:
 - JRE (Java Runtime Environment) 5.0 o superior.
 - Gestor de bases de datos: PostgreSQL 8.3.
 - Servidor para control de configuración Subversión 1.4.4 o superior.
 - Clientes para Subversion: TortoiseSVN (plataforma Windows). RapidSVN (plataforma UNIX)

- Requerimientos mínimos de hardware:
 - PC Servidor: Pentium 4, CPU 3.00 Ghz, 2 Gb RAM
 - PC Cliente: Pentium 3, 128 Mb RAM.
 - Computadoras conectadas a una red de al menos 100 Mb/s.

- Requerimientos de apariencia o interfaz externa:

El sistema debe ser:

 - Amigable.
 - Fácil de comprender.
 - Interactivo.

- Restricciones en el diseño e implementación:
 - La calidad del sistema va a estar dada en gran medida por el tiempo de respuesta. La aplicación debe responder a las peticiones de los usuarios en el menor tiempo posible. Para garantizar el desarrollo de la aplicación se utilizará como guía el Proceso Unificado de Desarrollo (RUP). Para visualizar, especificar, construir y documentar los artefactos se utilizará el lenguaje UML (Unified Modeling Language) y como herramienta de apoyo a este lenguaje de modelación se utilizará Visual Paradigm Suite 3.0. La aplicación será programada en Java utilizando el IDE Eclipse 3.2.

- Requerimientos de seguridad:
 - El sistema de gestión debe ser capaz de crear usuarios y niveles de privilegios para acceder a la información. En particular debe restringir las operaciones de escritura

sobre el registro de tamaños sólo al Servicio de Integración de Estadísticas del StatSVN (SVNStatBash).

- El sistema deberá poder ser auditado. Para esto se llevará un registro de quiénes han accedido al sistema, datos como la hora de entrada, hora de salida y las operaciones siempre deberán chequearse contra un nombre de usuario.
- Se deberá garantizar el respaldo del servidor de la base de datos con fuentes de corriente alternativas utilizando para esto UPSs o grupo electrógeno a fin de que toda la universidad pueda acceder a este recurso aún en condiciones anómalas.
- Los servidores deberán permanecer en un lugar donde solamente tenga acceso personal autorizado. Se propone llave magnética para esto.
- Requerimientos de soporte:
 - La aplicación deberá poseer una ayuda interactiva con el usuario.
 - Se mostrará en cada ventana un acceso de cualquier tipo a la ayuda donde el usuario podrá evacuar la duda relacionada.
- Requerimientos de portabilidad:
 - El software se realizará utilizando tecnología multiplataforma como es el caso de Java, el gestor de BD PostgreSQL y el Gestor de Configuración Subversion.

2.4 Sistema de Gestión de los Indicadores de Productividad

El software de gestión es el encargado de manejar la información relativa al proyecto y emitir los reportes de productividad.

A continuación se exponen los requisitos funcionales, el diagrama de casos de uso y sus especificaciones.

2.4.1 Requisitos funcionales

RF1: El sistema de gestión debe manejar una base de datos con datos relativos a los equipos de desarrollo, programadores, productos a construir, documentos de especificaciones de casos de uso, casos de uso y la jerarquía de carpetas existentes en el repositorio.

- RF1.1: Insertar la información en la base de datos.

- RF1.2: Modificar la información en las tablas correspondientes.
- RF1.3 Eliminar la información deseada.
- RF1.4: Listar los datos de las tablas.

RF2: El sistema de gestión debe calcular el tiempo total de producción de módulo.

- El tiempo total de producción es la suma del tiempo de análisis más el tiempo de desarrollo.

RF3: El sistema de gestión debe emitir reportes relacionados con la información almacenada en la base de datos.

- RF3.1: Tamaño producido por persona-mes (Tam/PM). Se suman todos los tamaños y se suman las personas que participaron en la elaboración de los documentos más los programadores que intervinieron en el desarrollo. Se obtienen los tiempos totales de producción. Se multiplica la cantidad de personas por la cantidad de tiempos, y por último: $Tam/PM = \text{Suma de tamaños} / \text{Suma de Personas} \times \text{Suma de Tiempos}$. (el tiempo se expresa en meses).
- RF3.2: Este indicador se deberá poder obtener para: cada equipo, cada producto, cada caso de uso, cada programador, toda la universidad, a partir de definir una versión X de un producto (en una fecha dada).

2.4.2 Definición de los actores del sistema

Los actores que interactúan con el Sistema de Gestión de Indicadores de Productividad (SGIP) son:

Actor	Descripción
Jefe de proyecto	Persona encargada de manejar la organización del proyecto productivo. Controla los principales datos de los artefactos y los integrantes que interactúan con el producto software.
Arquitecto	Persona encargada de definir la organización física del proyecto por jerarquía de carpetas. Agrupa en cada una de estas los ficheros fuentes y ensamblados que sustentarán el ejecutable final del producto informático. Debe asociar qué documento de especificación de casos de usos se implementa en cada carpeta. Se encarga de subir esta jerarquía de carpetas al repositorio de control de versiones

	(Subversion) y mantener la asociación documentos-carpetas actualizada.
--	--

▪ **Diagrama de Paquetes**

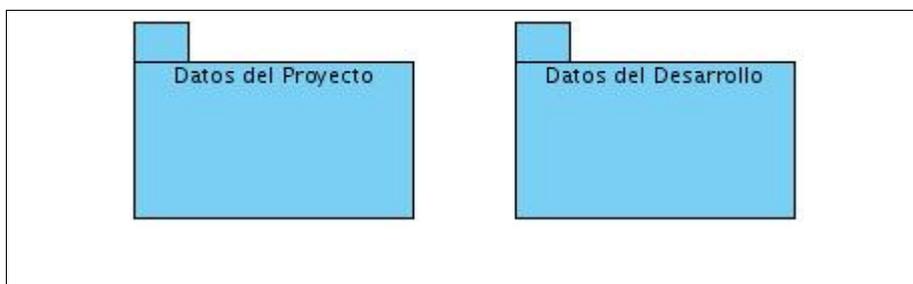


Fig. 2.2 Diagrama de Paquetes del Sistema de Gestión de Indicadores de Productividad.

Paquete	Descripción
Datos del proyecto	En este paquete el Jefe de Proyecto puede registrar y administrar la información relativa a los productos, documentos y casos de uso que se manejan en dicho proyecto. Asimismo el Arquitecto definirá qué documentos se estarán implementando en cada carpeta del repositorio de control de versiones (Subversion).
Datos del desarrollo	En este paquete el Jefe de Proyecto puede registrar y administrar la información relativa a los equipos y programadores que trabajan en dicho proyecto.

2.4.3 Listado de los Casos de Uso del Sistema. Paquete “Datos del desarrollo”

CU-1	Insertar Equipo
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto solicita insertar un equipo en la base de datos del software de gestión.
Referencia	RF 1.1

CU-2	Insertar programadores del equipo
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto selecciona la opción de insertarle los programadores que tendrá un mismo equipo de desarrollo.
Referencia	RF 1.1

CU-3	Administrar Equipo
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto solicita administrar los datos de los equipos existentes. Para esto el sistema le muestra los equipos que se han registrado en la base de datos del Software de Gestión de Indicadores del proyecto.
Referencia	RF 1.4

CU-4	Actualizar equipo
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto selecciona un registro de la tabla de equipos existentes y modifica los valores del mismo.
Referencia	RF 1.2

CU-5	Eliminar equipo
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de proyecto selecciona uno o varios registros de la tabla de equipos existentes y elige la opción de eliminarlos.
Referencia	RF 1.3

CU-6	Insertar Programador
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto solicita insertar un nuevo programador en la base de datos del software de gestión. Para ello el sistema lista los equipos existentes para que el Jefe de proyecto pueda decidir a qué equipo pertenecerá el nuevo programador.
Referencia	RF 1.1

CU-7	Administrar programadores
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto solicita administrar los datos de los programadores existentes. Para esto el sistema le muestra los programadores que se han registrado en la base de datos del software de gestión de indicadores del proyecto y su equipo correspondiente.
Referencia	RF 1.4

CU-8	Actualizar programador
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto selecciona un registro de la tabla de programadores existentes y modifica los valores del mismo.
Referencia	RF 1.2

CU-9	Eliminar programador
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto selecciona uno o varios registros de la tabla de programadores existentes y elige la opción de eliminarlos.
Referencia	RF 1.3

2.4.4 Listado de los Casos de Uso del Sistema. Paquete “Datos del Proyecto”

CU-10	Insertar Producto
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto solicita insertar un producto en la base de datos del software de gestión.
Referencia	RF 1.1

CU-11	Administrar Productos
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto solicita administrar los datos de los productos de software existentes. Para esto el sistema le muestra los productos que se han registrado en la base de datos del software de gestión de

	indicadores del proyecto.
Referencia	RF 1.4

CU-12	Actualizar producto
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto selecciona un registro de la tabla de productos existentes y modifica los valores del mismo.
Referencia	RF 1.2

CU-13	Eliminar producto
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto selecciona uno o varios registros de la tabla de productos existentes y elige la opción de eliminarlos.
Referencia	RF 1.3

CU-14	Insertar Documento
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto solicita insertar un nuevo documento en la base de datos del software de gestión. Para ello el sistema lista los productos existentes para que el Jefe de Proyecto pueda decidir a qué producto pertenecerá el nuevo documento.
Referencia	RF 1.1

CU-15	Administrar Documentos
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto solicita administrar los datos de los documentos de software existentes. Para esto el sistema le muestra los documentos que se han registrado en la base de datos del software de gestión de indicadores del proyecto.
Referencia	RF 1.4

CU-16	Actualizar documento
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto selecciona un registro de la tabla de documentos existentes y modifica los valores del mismo.
Referencia	RF 1.2

CU-17	Eliminar documento
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto selecciona uno o varios registros de la tabla de documentos existentes y elige la opción de eliminarlos.
Referencia	RF 1.3

CU-18	Insertar Caso de Uso
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto solicita insertar un nuevo caso de uso en la base de datos del software de gestión. Para ello el sistema lista los documentos existentes para que el Jefe de proyecto pueda decidir a qué documento pertenecerá el nuevo caso de uso.
Referencia	RF 1.1

CU-19	Administrar Casos de Uso
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto solicita administrar los datos de los casos de uso existentes. Para esto el sistema le muestra los casos de uso que se han registrado en la base de datos del Software de Gestión de Indicadores del proyecto y a qué documento pertenece.
Referencia	RF 1.4

CU-20	Actualizar caso de uso
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto selecciona un registro de la tabla de casos de uso existentes y modifica los valores del mismo.
Referencia	RF 1.2

CU-21	Eliminar caso de uso
Actor	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto selecciona uno o varios registros de la tabla de casos de uso existentes y elige la opción de eliminarlos.
Referencia	RF 1.3

CU-22	Insertar Documento en Carpeta
Actor	Arquitecto
Descripción	El caso de uso se inicia cuando el arquitecto define qué documentos se estarán implementando en cada carpeta del repositorio de control de versiones (Subversion).
Referencia	RF 1.1

CU-23	Administrar Carpetas
Actor	Arquitecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto solicita administrar las carpetas del repositorio existentes. Para esto el sistema le muestra la relación de carpetas que han sido registradas en la base de datos del Software de Gestión de Indicadores del proyecto y con qué documento ha sido asociado su desarrollo.
Referencia	RF 1.4

CU-24	Eliminar carpeta
Actor	Arquitecto
Descripción	El caso de uso se inicia cuando el Jefe de Proyecto selecciona uno o varios registros de la tabla que muestra las asociaciones de carpetas del repositorio contra documentos existentes y elige la opción de eliminarlos.
Referencia	RF 1.3

2.4.5 Diagrama de Casos de Uso

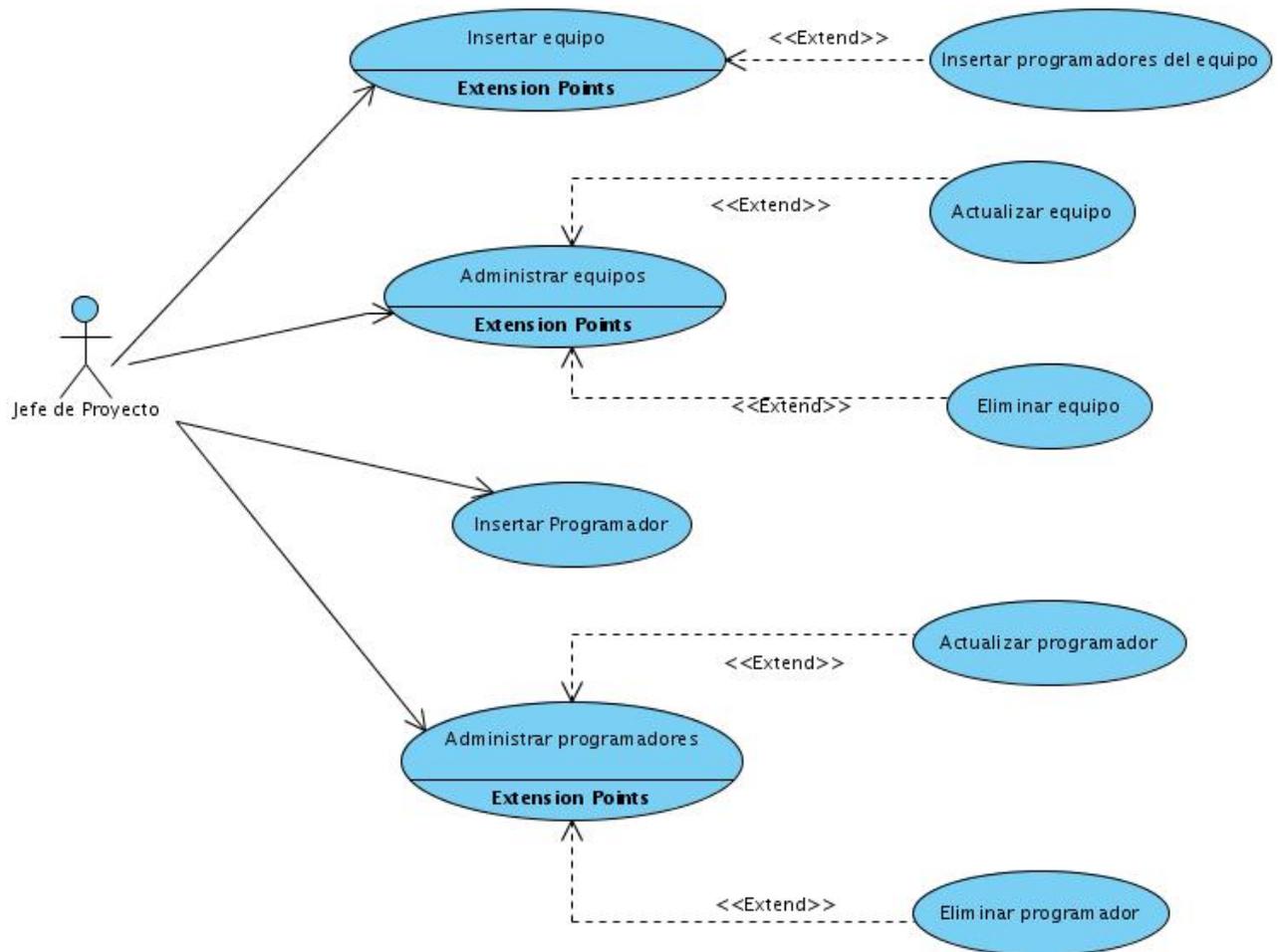


Fig. 2.3 Diagrama de Casos de Uso del Paquete Datos del Desarrollo.

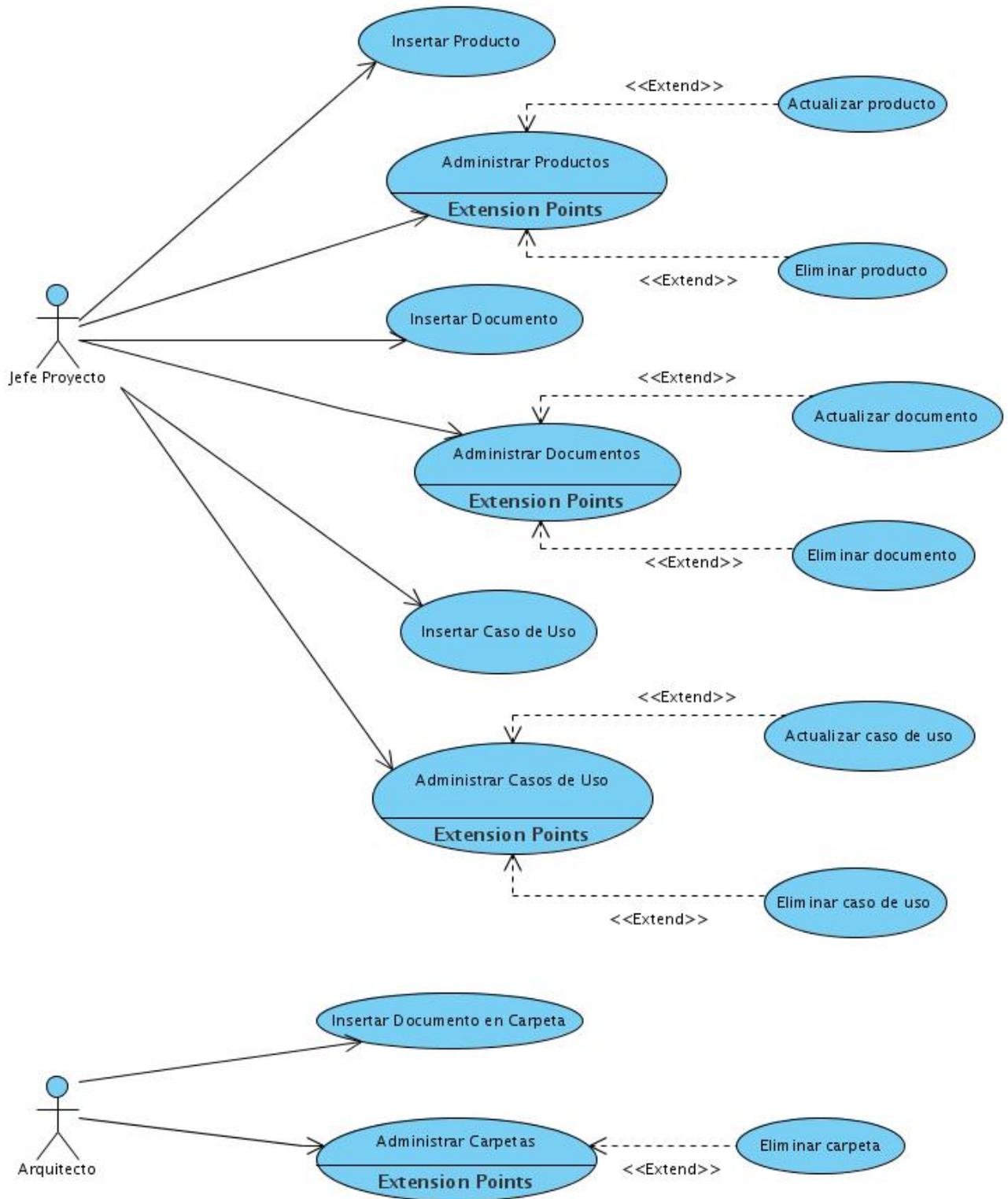
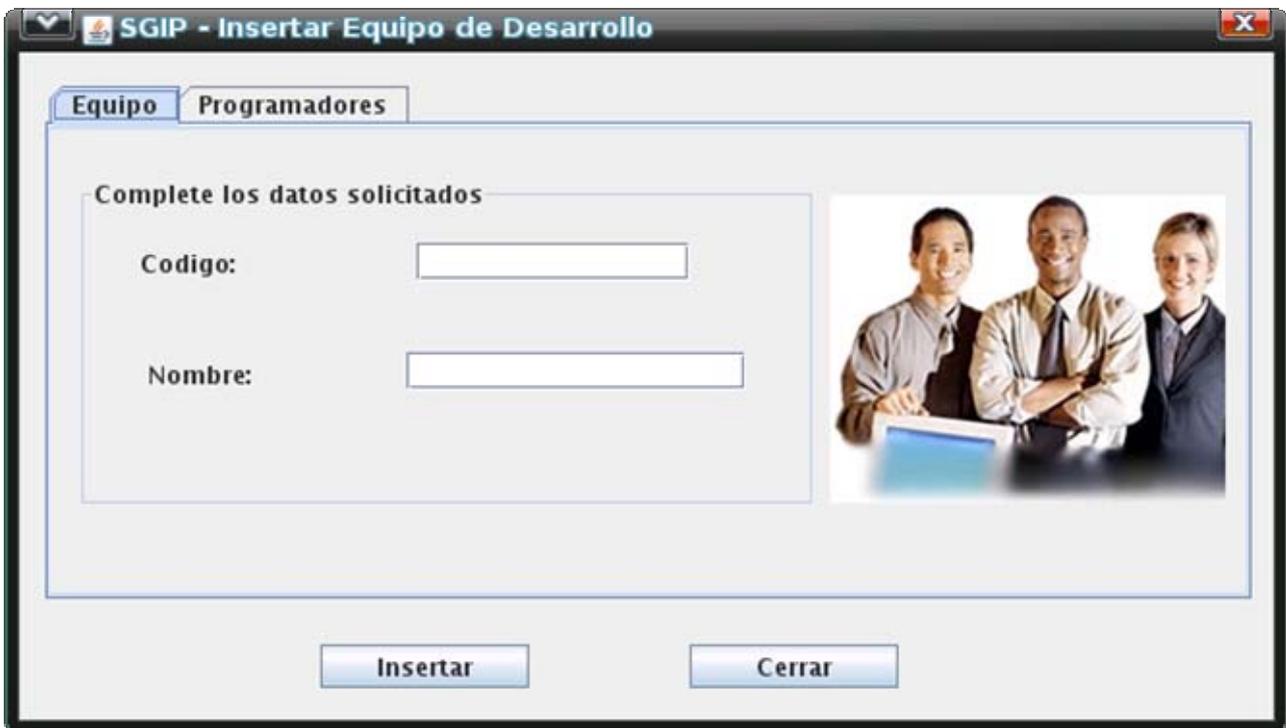


Fig. 2.4 Diagrama de Casos de Uso del Paquete Datos del Proyecto.

2.4.6. Especificaciones de los casos de uso. Paquete “Datos del Desarrollo”

2.4.6.1 CU: Insertar Equipo

Prototipo de interfaz de usuario:



CU-1	Insertar Equipo
Propósito	Tiene como objetivo insertar los datos de un equipo en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto va al menú Datos de Desarrollo de la página principal del software de gestión, selecciona la opción Equipo y dentro el submenú Insertar Equipo. Introduce los datos guardándose los mismos en la base de datos.
Referencias	RF 1.1

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
1. Selecciona la opción Insertar Equipo	2. Muestra una pestaña del formulario solicitando los siguientes datos del equipo: <ul style="list-style-type: none"> ● Código

	● Nombre
3. Introduce los datos solicitados	
4. Selecciona la pestaña Programadores	5. Invoca al CU Insertar programadores del equipo
5. Selecciona la opción Insertar	6. Comprueba que todos los campos de entrada de datos de la pestaña Equipo tengan valores.
	7. Almacena en la BD la información introducida.
	8. El sistema muestra un mensaje “Equipo insertado satisfactoriamente.”
9. Selecciona el botón OK.	10. Ir a la acción 2 del CU.
11. Selecciona el botón CERRAR	12. Cierra la ventana.
Flujo alternativo 6a Campos en blanco	
Acción del actor	Respuesta del sistema
	6a.1. Comprueba que hay campos en blanco.
	6a.2. Muestra un mensaje de error “Campos en blanco”.
6a.3. Selecciona el botón OK.	6a.4. Ir a la acción 2 del CU.
Flujo alternativo 7a Registro existente en la BD	
Acción del actor	Respuesta del sistema
	7a.1. Comprueba que hay un registro con el mismo identificador en la base de datos.
	7a.2. Muestra un mensaje de error “Ya ha sido registrado un equipo con este identificador”.
7a.3. Selecciona el botón OK.	7a.4. Ir a la acción 2 del CU.

2.4.6.2 CU: Insertar programadores del equipo

Prototipo de interfaz de usuario:



CU-2	Insertar programadores del equipo
Propósito	Tiene como objetivo insertar los datos de los programadores de un mismo equipo en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto va al menú Datos de Desarrollo de la página principal del software de gestión, selecciona la opción Equipo y dentro el submenú Insertar Equipo. En la pestaña Programadores introduce los datos correspondientes, almacenándolos en la base de datos.
Referencias	RF 1.1

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
	<p>1. Muestra una pestaña del formulario solicitando los siguientes datos del programador:</p> <ul style="list-style-type: none"> ● Alias

	<ul style="list-style-type: none"> ● Nombre ● Apellidos <p>Muestra una tabla con los programadores que van a ser adicionados al equipo, poniendo al final de cada fila la opción de seleccionar para eliminar.</p>
2. Introduce los datos solicitados	
4. Selecciona el botón Adicionar	5. Comprueba que todos los campos de entrada de datos de la pestaña Programadores tengan valores.
	6. Los datos pasan a formar parte de la tabla.
7. Selecciona una celda de la tabla para editar.	8. Permite editar la celda seleccionada.
9. Selecciona el botón Eliminar	10. Comprueba que al menos existe un programador marcado para eliminar.
	11. Emite un mensaje preguntando si desea eliminar el/los programador(es) seleccionados.
12. Selecciona el botón SI	13. Elimina los programadores seleccionados de la tabla. Ir a la acción 1 del CU.
14. Selecciona el botón CERRAR	15. Cierra la ventana.
Flujo alternativo 5a Campos en blanco	
Acción del actor	Respuesta del sistema
	5a.1. Comprueba que hay campos en blanco.
	5a.2. Muestra un mensaje de error "Campos en blanco".
5a.3. Selecciona el botón OK.	5a.4. Ir a la acción 1 del CU.
Flujo alternativo 10a No existe programador marcado para eliminar	
Acción del actor	Respuesta del sistema
	10a.1. Comprueba que no existe al menos un programador seleccionado en la tabla para eliminar.
	10a.2. Muestra un mensaje de error. "No hay programadores seleccionados para eliminar"
10a.3. Selecciona el botón OK.	10a.4. Ir a la acción 1 del CU.
Flujo alternativo 11a NO	
Acción del actor	Respuesta del sistema
11a.1. Selecciona la opción NO	11a.2. Ir a la acción 1 del CU.

Flujo alterno 4a Alias repetido	
Acción del actor	Respuesta del sistema
	4a.1. Comprueba que ya existe un programador en la tabla con el mismo alias de usuario.
	4a.2. Muestra un mensaje de error “Ya existe un programador con ese alias, utilice otro”.
4a.3. Selecciona el botón OK.	4a.4. Ir a la acción 1 del CU.

2.4.6.3 CU: Administrar equipos

Prototipo de interfaz de usuario:



CU-3	Administrar equipos
Propósito	Tiene como objetivo visualizar, actualizar o eliminar los datos de un equipo en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto va al menú Datos de Desarrollo de la página principal del software de gestión, selecciona la opción

	Equipo y dentro el submenú Administrar Equipo.
Referencias	FR 1.2, RF 1.3, RF 1.4

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
1. Selecciona el botón Refrescar	2. Se listan en una tabla los equipos existentes en la BD, especificando los siguientes datos: <ul style="list-style-type: none"> ● Código ● Nombre ● Opción de eliminar
3. Selecciona el botón Eliminar	4. Se invoca al CU Eliminar equipo .
Sección Actualizar Equipo	
Acción del actor	Respuesta del sistema
	5. Muestra en el formulario una sección para actualizar el equipo seleccionado. Se solicitan los siguientes campos: <ul style="list-style-type: none"> ● Nuevo nombre
6. Selecciona un registro de la tabla de equipos existentes	7. Muestra el nombre actual del equipo seleccionado por el campo de entrada de texto.
8. Introduce el nuevo valor del campo.	
9. Selecciona el botón Actualizar.	10. Invoca al CU Actualizar equipo .
11. Selecciona el botón CERRAR	12. Cierra la ventana.

2.4.6.4 CU: Actualizar equipo

CU-4	Actualizar equipo
Propósito	Tiene como objetivo actualizar los datos de un equipo en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto selecciona el botón Actualizar de la ventana Administrar Equipo.
Referencias	RF 1.2

Tabla de eventos

Flujo básico	
--------------	--

Flujo alternativo 1a No existe registro marcado	
Acción del actor	Respuesta del sistema
	1. Comprueba que exista un registro marcado en la tabla de equipos existentes.
	2. Comprueba que los campos de entrada de textos con los datos a actualizar contengan valores.
	3. Actualiza el registro correspondiente en la tabla de la BD.
	4. Emite un mensaje "Equipo actualizado satisfactoriamente".
5. Selecciona el botón OK.	6. Ir a la acción 2 del CU Administrar Equipo .
Flujo alternativo 2a Campos en blanco	
Acción del actor	Respuesta del sistema
	1a.1. Comprueba que no existe al menos un equipo seleccionado en la tabla para actualizar.
	1a.2. Muestra un mensaje de error. "No hay ningún equipo seleccionado para actualizar"
1a.3. Selecciona el botón OK.	1a.4. Ir a la acción 2 del CU Administrar Equipo .
Flujo alternativo 2a Campos en blanco	
Acción del actor	Respuesta del sistema
	2a.1. Comprueba que hay campos en blanco.
	2a.2. Muestra un mensaje de error "Campos en blanco".
2a.3. Selecciona el botón OK.	2a.4. Ir a la acción 2 del CU Administrar Equipo .

2.4.6.5 CU: Eliminar equipo

CU-5	Eliminar equipo
Propósito	Tiene como objetivo eliminar un equipo en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto selecciona el botón Eliminar de la ventana Administrar Equipo.
Referencias	RF 1.3

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
	1. Comprueba que exista al menos un registro marcado en la tabla de equipos existentes para eliminar.
	2. Emite un mensaje preguntando si desea eliminar el/los equipos(es) seleccionados.
3. Selecciona el botón SI	3. Elimina el/los registro(s) correspondiente(s) en la tabla de la BD.
	4. Emite un mensaje "Equipo(s) eliminado(s) satisfactoriamente".
5. Selecciona el botón OK.	6. Ir a la acción 2 del CU Administrar Equipo .
Flujo alternativo 1a No existe registro marcado	
Acción del actor	Respuesta del sistema
	1a.1. Comprueba que no existe al menos un equipo seleccionado en la tabla para eliminar.
	1a.2. Muestra un mensaje de error. "No hay equipos seleccionados para eliminar"
1a.3. Selecciona el botón OK.	1a.4. Ir a la acción 2 del CU Administrar Equipo .
Flujo alternativo 2a NO	
Acción del actor	Respuesta del sistema
2a.1. Selecciona la opción NO	2a.2. Ir a la acción 2 del CU Administrar Equipo .

2.4.6.6 CU: Insertar Programador

Prototipo de interfaz de usuario:



CU-6	Insertar Programador
Propósito	Tiene como objetivo insertar los datos de un programador en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto va al menú Datos de Desarrollo de la página principal del software de gestión, selecciona la opción Programador y dentro el submenú Insertar Programador. Introduce los datos guardándose los mismos en la base de datos.
Referencias	RF 1.1, RF 1.4

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
	1. Muestra un formulario con una tabla donde se listan

	<p>los siguientes datos de los equipos existentes:</p> <ul style="list-style-type: none"> ● Código ● Nombre
	<p>2. Muestra una sección del formulario solicitando los siguientes datos del programador:</p> <ul style="list-style-type: none"> ● Alias ● Nombre(s) ● Apellidos
3. Selecciona un equipo	
4. Introduce los datos solicitados	
5. Selecciona la opción Insertar	6. Comprueba que haya un registro seleccionado en la tabla donde se listan los equipos.
	7. Comprueba que todos los campos de entrada de datos tengan valores.
	8. Almacena en la BD la información introducida.
	9. El sistema muestra un mensaje "Programador insertado satisfactoriamente."
10. Selecciona el botón OK.	11. Ir a la acción 1,2 del CU.
12. Selecciona el botón CERRAR	13. Cierra la ventana.
Flujo alternativo 6a Registro no seleccionado en la tabla	
Acción del actor	Respuesta del sistema
	6a.1. Comprueba que no hay un registro seleccionado en la tabla donde se listan los equipos.
	6a.2. Muestra un mensaje de error "Debe seleccionar a qué equipo pertenecerá este programador".
6a.3. Selecciona el botón OK.	6a.4. Ir a la acción 1,2 del CU.
Flujo alternativo 7a Campos en blanco	
Acción del actor	Respuesta del sistema
	7a.1. Comprueba que hay campos en blanco.
	7a.2. Muestra un mensaje de error "Campos en blanco".
7a.3. Selecciona el botón OK.	7a.4. Ir a la acción 1,2 del CU.
Flujo alternativo 8a Registro existente en la BD	
Acción del actor	Respuesta del sistema

	8a.1. Comprueba que hay un registro con el mismo identificador en la base de datos.
	8a.2. Muestra un mensaje de error “Ya ha sido registrado un programador con este identificador”.
8a.3. Selecciona el botón OK.	8a.4. Ir a la acción 1,2 del CU.

2.4.6.7 CU: Administrar programadores

Prototipo de interfaz de usuario:



CU-7	Administrar programadores
Propósito	Tiene como objetivo visualizar, actualizar o eliminar los datos de un programador en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto va al menú Datos de Desarrollo de la página principal del software de gestión, selecciona la opción Programador y dentro el submenú Administrar programadores.

Referencias	FR 1.2, RF 1.3, RF 1.4
-------------	------------------------

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
1. Selecciona el botón Refrescar	2. Se listan en una tabla los programadores existentes en la BD, especificando los siguientes datos: <ul style="list-style-type: none"> ● Alias ● Nombre(s) ● Apellidos ● Equipo al que pertenece ● Opción de eliminar
3. Selecciona el botón Eliminar	4. Se invoca al CU Eliminar programador .
Sección Actualizar Programador	
Acción del actor	Respuesta del sistema
	5. Muestra en el formulario una sección para actualizar el programador seleccionado. Se solicitan los siguientes campos: <ul style="list-style-type: none"> ● Nuevo nombre ● Nuevos apellidos
6. Selecciona un registro de la tabla de programadores existentes	7. Muestra los datos actuales del programador seleccionado por los campos de entrada de texto.
8. Introduce nuevos valores para los campos.	
9. Selecciona el botón Actualizar.	10. Invoca al CU Actualizar programador .
11. Selecciona el botón CERRAR	12. Cierra la ventana.

2.4.6.8 CU: Actualizar programador

CU-8	Actualizar programador
Propósito	Tiene como objetivo actualizar los datos de un programador en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto selecciona el botón

	Actualizar de la ventana Administrar Programadores.
Referencias	RF 1.2

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
	1. Comprueba que exista un registro marcado en la tabla de programadores existentes.
	2. Comprueba que los campos de entrada de textos con los datos a actualizar contengan valores.
	3. Actualiza el registro correspondiente en la tabla de la BD.
	4. Emite un mensaje "Programador actualizado satisfactoriamente".
5. Selecciona el botón OK.	6. Ir a la acción 2 del CU Administrar Equipo .
Flujo alternativo 1a No existe registro marcado	
Acción del actor	Respuesta del sistema
	1a.1. Comprueba que no existe al menos un programador seleccionado en la tabla para actualizar.
	1a.2. Muestra un mensaje de error. "No hay ningún programador seleccionado para actualizar"
1a.3. Selecciona el botón OK.	1a.4. Ir a la acción 2 del CU Administrar programadores .
Flujo alternativo 2a Campos en blanco	
Acción del actor	Respuesta del sistema
	2a.1. Comprueba que hay campos en blanco.
	2a.2. Muestra un mensaje de error "Campos en blanco".
2a.3. Selecciona el botón OK.	2a.4. Ir a la acción 2 del CU Administrar programadores .

2.4.6.9 CU: Eliminar programador

CU-9	Eliminar programador
Propósito	Tiene como objetivo eliminar un programador en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto selecciona el botón Eliminar de la ventana Administrar programadores.
Referencias	RF 1.3

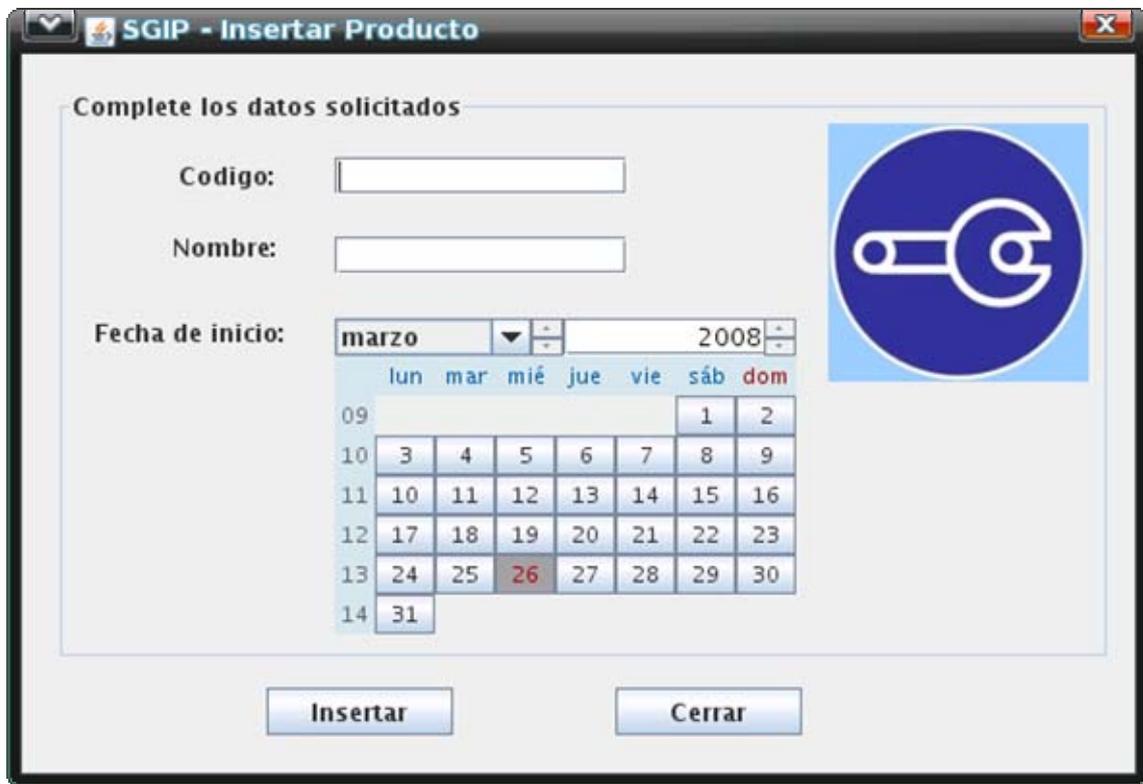
Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
	1. Comprueba que exista al menos un registro marcado en la tabla de programadores existentes para eliminar.
	2. Emite un mensaje preguntando si desea eliminar el/los programador(es) seleccionados.
3. Selecciona el botón SI	4. Elimina el/los registro(s) correspondiente(s) en la tabla de la BD.
	5. Emite un mensaje "Programador(es) eliminado(s) satisfactoriamente".
6. Selecciona el botón OK.	7. Ir a la acción 2 del CU Administrar programadores .
Flujo alternativo 1a No existe registro marcado	
Acción del actor	Respuesta del sistema
	1a.1. Comprueba que no existe al menos un programador seleccionado en la tabla para eliminar.
	1a.2. Muestra un mensaje de error. "No hay programadores seleccionados para eliminar"
1a.3. Selecciona el botón OK.	1a.4. Ir a la acción 2 del CU Administrar programadores .
Flujo alternativo 2a NO	
Acción del actor	Respuesta del sistema
2a.1. Selecciona la opción NO	2a.2. Ir a la acción 2 del CU Administrar programadores .

2.4.7. Especificaciones de los casos de uso. Paquete “Datos del Proyecto”

2.4.7.1 CU: Insertar Producto

Prototipo de interfaz de usuario:



CU-10	Insertar Producto
Propósito	Tiene como objetivo insertar los datos de un producto software en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto va al menú Datos del Proyecto de la página principal del software de gestión, selecciona la opción Producto y dentro el submenú Insertar Producto. Introduce los datos guardándose los mismos en la base de datos.
Referencias	RF 1.1

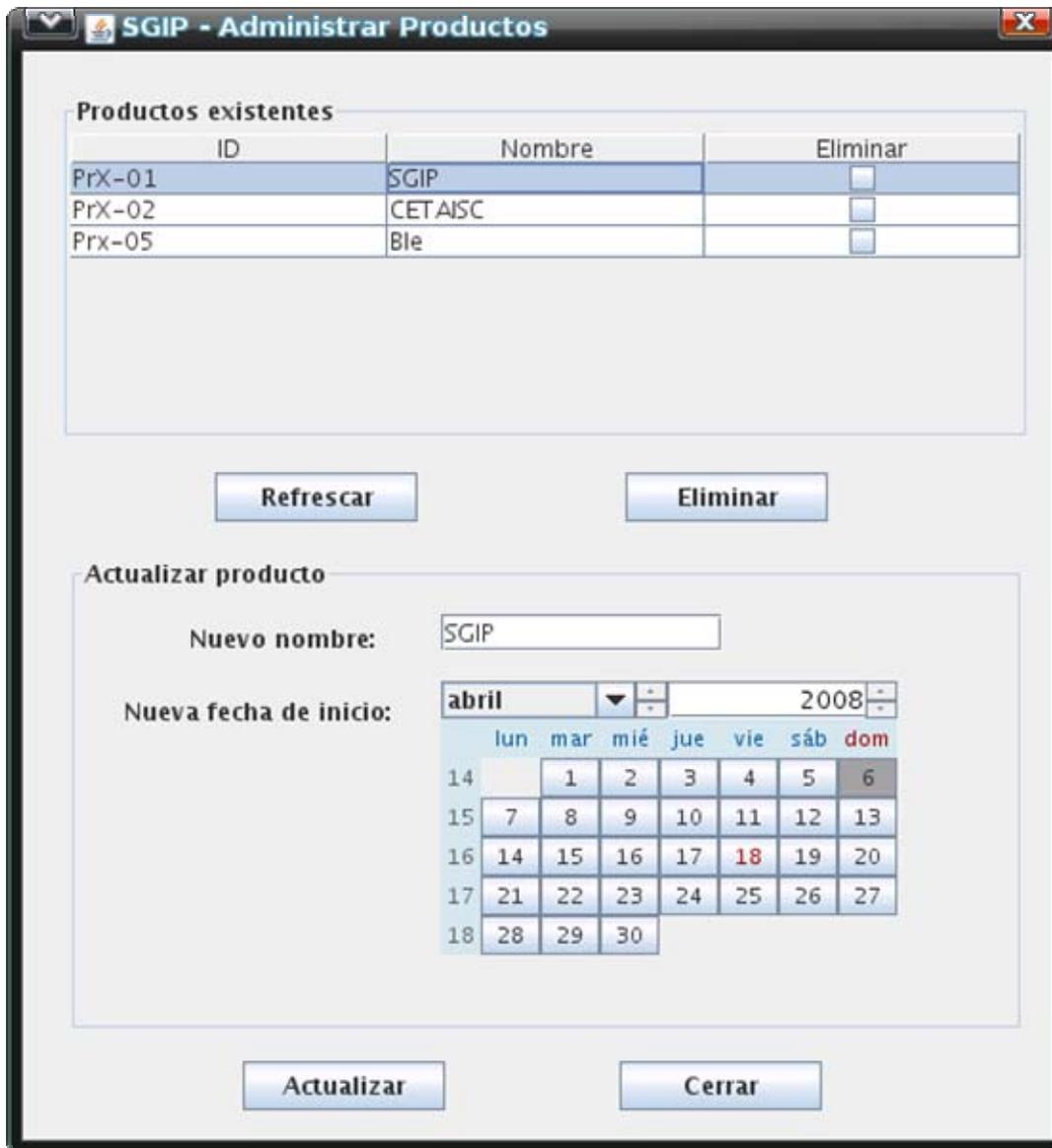
Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema

1. Selecciona la opción Insertar Producto	2. Muestra una pestaña del formulario solicitando los siguientes datos del producto: <ul style="list-style-type: none"> ● Código ● Nombre ● Fecha de inicio
3. Introduce los datos solicitados	
4. Selecciona la opción Insertar	6. Comprueba que todos los campos de entrada de datos tengan valores.
	7. Almacena en la BD la información introducida.
	8. El sistema muestra un mensaje “Producto insertado satisfactoriamente.”
9. Selecciona el botón OK.	10. Ir a la acción 2 del CU.
11. Selecciona el botón CERRAR	12. Cierra la ventana.
Flujo alternativo 6a Campos en blanco	
Acción del actor	Respuesta del sistema
	6a.1. Comprueba que hay campos en blanco.
	6a.2. Muestra un mensaje de error “Campos en blanco”.
6a.3. Selecciona el botón OK.	6a.4. Ir a la acción 2 del CU.
Flujo alternativo 7a Registro existente en la BD	
Acción del actor	Respuesta del sistema
	7a.1. Comprueba que hay un registro con el mismo identificador en la base de datos.
	7a.2. Muestra un mensaje de error “Ya ha sido registrado un producto con este identificador”.
7a.3. Selecciona el botón OK.	7a.4. Ir a la acción 2 del CU.

2.4.7.2 CU: Administrar Productos

Prototipo de interfaz de usuario:



CU-11	Administrar Productos
Propósito	Tiene como objetivo visualizar, actualizar o eliminar los datos de un producto software en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto va al menú Datos del proyecto de la página principal del software de gestión, selecciona la opción Producto y dentro el submenú Administrar Productos.

Referencias	FR 1.2, RF 1.3, RF 1.4
-------------	------------------------

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
1. Selecciona el botón Refrescar	2. Se listan en una tabla los productos existentes en la BD, especificando los siguientes datos: <ul style="list-style-type: none"> ● Código ● Nombre ● Opción de eliminar
3. Selecciona el botón Eliminar	4. Se invoca al CU Eliminar producto .
Sección Actualizar equipo	
Acción del actor	Respuesta del sistema
	5. Muestra en el formulario una sección para actualizar el equipo seleccionado. Se solicitan los siguientes campos: <ul style="list-style-type: none"> ● Nuevo nombre ● Nueva fecha de inicio
6. Selecciona un registro de la tabla de productos existentes	7. Muestra los datos actuales del producto seleccionado por los campos de entrada de texto.
8. Introduce nuevos valores para los campos.	
9. Selecciona el botón Actualizar.	10. Invoca al CU Actualizar producto .
11. Selecciona el botón CERRAR	12. Cierra la ventana.

2.4.7.3 CU: Actualizar producto

CU-12	Actualizar producto
Propósito	Tiene como objetivo actualizar los datos de un producto en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto selecciona el botón Actualizar de la ventana Administrar Productos.
Referencias	RF 1.2

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
	1. Comprueba que exista un registro marcado en la tabla de productos existentes.
	2. Comprueba que los campos de entrada de textos con los datos a actualizar contengan valores.
	3. Actualiza el registro correspondiente en la tabla de la BD.
	4. Emite un mensaje “Producto actualizado satisfactoriamente”.
5. Selecciona el botón OK.	6. Ir a la acción 2 del CU Administrar Productos .
Flujo alternativo 1a No existe registro marcado	
Acción del actor	Respuesta del sistema
	1a.1. Comprueba que no existe al menos un producto seleccionado en la tabla para actualizar.
	1a.2. Muestra un mensaje de error. “No hay ningún producto seleccionado para actualizar”
1a.3. Selecciona el botón OK.	1a.4. Ir a la acción 2 del CU Administrar Productos .
Flujo alternativo 2a Campos en blanco	
Acción del actor	Respuesta del sistema
	2a.1. Comprueba que hay campos en blanco.
	2a.2. Muestra un mensaje de error “Campos en blanco”.
2a.3. Selecciona el botón OK.	2a.4. Ir a la acción 2 del CU Administrar Productos .

2.4.7.4 CU: Eliminar producto

CU-13	Eliminar producto
Propósito	Tiene como objetivo eliminar un producto en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto selecciona el botón Eliminar de la ventana Administrar Productos.
Referencias	RF 1.3

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
	1. Comprueba que exista al menos un registro marcado en la tabla de productos existentes para eliminar.
	2. Emite un mensaje preguntando si desea eliminar el/los producto(s) seleccionados.
3. Selecciona el botón SI	3. Elimina el/los registro(s) correspondiente(s) en la tabla de la BD.
	4. Emite un mensaje "Producto(s) eliminado(s) satisfactoriamente".
5. Selecciona el botón OK.	6. Ir a la acción 2 del CU Administrar Productos .
Flujo alternativo 1a No existe registro marcado	
Acción del actor	Respuesta del sistema
	1a.1. Comprueba que no existe al menos un producto seleccionado en la tabla para eliminar.
	1a.2. Muestra un mensaje de error. "No hay productos seleccionados para eliminar"
1a.3. Selecciona el botón OK.	1a.4. Ir a la acción 2 del CU Administrar Productos .
Flujo alternativo 2a NO	
Acción del actor	Respuesta del sistema
2a.1. Selecciona la opción NO	2a.2. Ir a la acción 2 del CU Administrar Equipo .

2.4.7.5 CU: Insertar Documento

Prototipo de interfaz de usuario:

SGIP - Insertar Documento

Seleccione el producto al que perteneciera este documento:

Productos existentes

ID	Nombre
PrX-01	SGIP
PrX-02	CETAISC
Prx-05	Ble

Complete los datos solicitados

Codigo:

Nombre:

Cantidad de personas:

Fecha de creacion: abril 2008

	lun	mar	mié	jue	vie	sáb	dom
14		1	2	3	4	5	6
15	7	8	9	10	11	12	13
16	14	15	16	17	18	19	20
17	21	22	23	24	25	26	27
18	28	29	30				

Insertar **Cerrar**

CU-14	Insertar Documento
Propósito	Tiene como objetivo insertar los datos de un documento en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto va al menú Datos del proyecto de la página principal del software de gestión, selecciona la opción Documento y dentro el submenú Insertar Documento. Introduce los datos guardándose los mismos en la base de datos.

Referencias	RF 1.1, RF 1.4
-------------	----------------

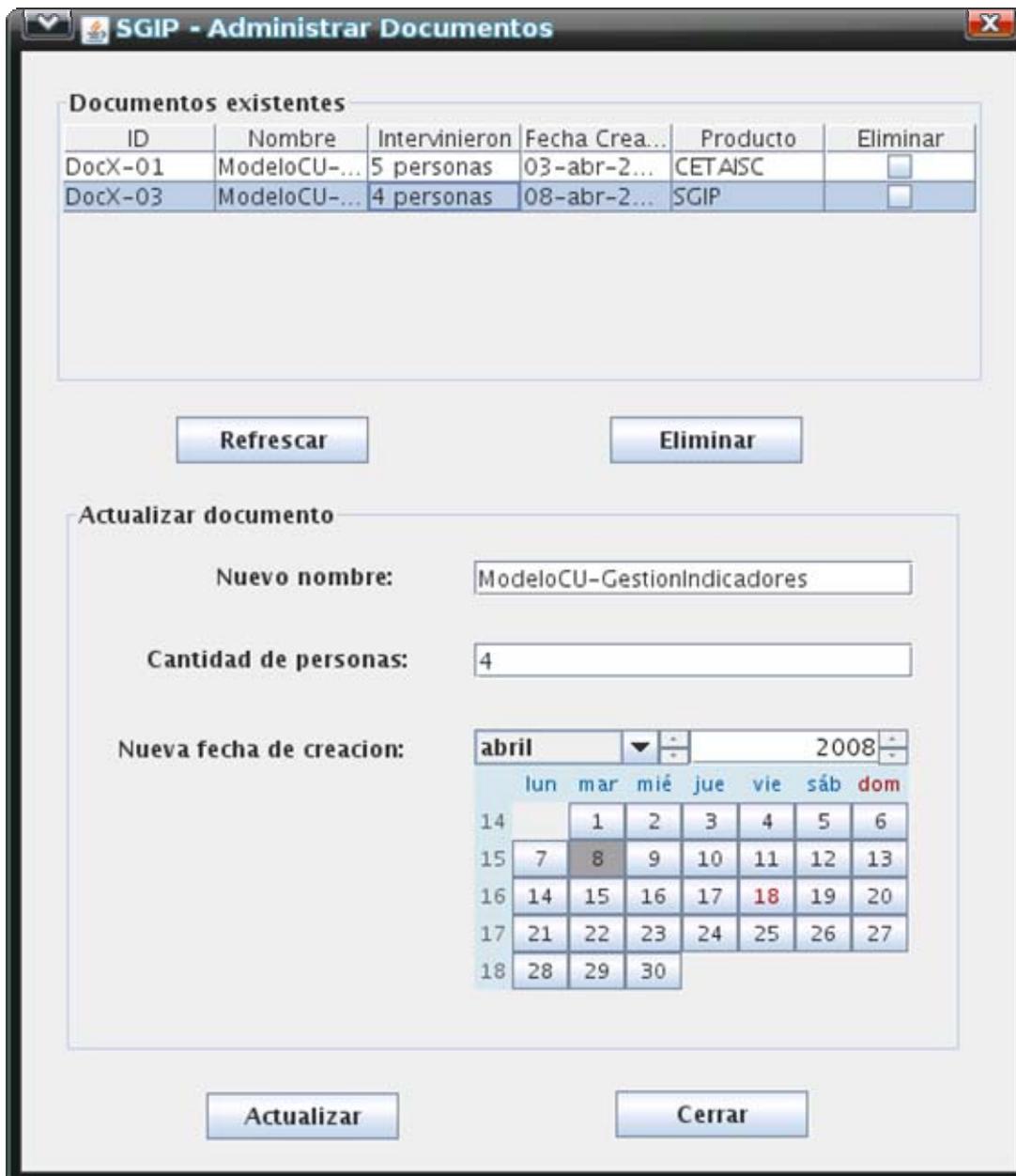
Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
	1. Muestra un formulario con una tabla donde se listan los siguientes datos de los productos existentes: <ul style="list-style-type: none"> ● Código ● Nombre
	2. Muestra una sección del formulario solicitando los siguientes datos del documento: <ul style="list-style-type: none"> ● Código ● Nombre ● Cantidad de personas que intervinieron en su elaboración ● Fecha de creación
3. Selecciona un equipo	
4. Introduce los datos solicitados	
5. Selecciona la opción Insertar	6. Comprueba que haya un registro seleccionado en la tabla donde se listan los productos.
	7. Comprueba que todos los campos de entrada de datos tengan valores.
	8. Almacena en la BD la información introducida.
	9. El sistema muestra un mensaje "Documento insertado satisfactoriamente."
10. Selecciona el botón OK.	11. Ir a la acción 1,2 del CU.
12. Selecciona el botón CERRAR	13. Cierra la ventana.
Flujo alternativo 6a Registro no seleccionado en la tabla	
Acción del actor	Respuesta del sistema
	6a.1. Comprueba que no hay un registro seleccionado en la tabla donde se listan los productos.
	6a.2. Muestra un mensaje de error "Debe seleccionar a qué producto pertenecerá este documento".
6a.3. Selecciona el botón OK.	6a.4. Ir a la acción 1,2 del CU.

Flujo alterno 7a Campos en blanco	
Acción del actor	Respuesta del sistema
	7a.1. Comprueba que hay campos en blanco.
	7a.2. Muestra un mensaje de error “Campos en blanco”.
7a.3. Selecciona el botón OK.	7a.4. Ir a la acción 1,2 del CU.
Flujo alterno 4a Entrada de texto en campo numérico	
Acción del actor	Respuesta del sistema
4a.1. Intenta introducir letras en el campo de entrada de texto Cantidad de personas.	4a.2. Inhabilita la opción de escribir sobre este campo.
	4a.3. Ir a la acción 1,2 del CU.
Flujo alterno 8a Registro existente en la BD	
Acción del actor	Respuesta del sistema
	8a.1. Comprueba que hay un registro con el mismo identificador en la base de datos.
	8a.2. Muestra un mensaje de error “Ya ha sido registrado un documento con este identificador”.
8a.3. Selecciona el botón OK.	8a.4. Ir a la acción 2 del CU.

2.4.7.6 CU: Administrar Documentos

Prototipo de interfaz de usuario:



CU-15	Administrar Documentos
Propósito	Tiene como objetivo visualizar, actualizar o eliminar los datos de un documento en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto va al menú Datos del proyecto de la página principal del software de gestión, selecciona la opción

	Documento y dentro el submenú Administrar Documentos.
Referencias	FR 1.2, RF 1.3, RF 1.4

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
1. Selecciona el botón Refrescar	2. Se listan en una tabla los documentos existentes en la BD, especificando los siguientes datos: <ul style="list-style-type: none"> ● Identificador ● Nombre ● Cantidad de personas que intervinieron en su elaboración ● Fecha de creación ● Producto al que pertenece ● Opción de eliminar
3. Selecciona el botón Eliminar	4. Se invoca al CU Eliminar documento .
Sección Actualizar documento	
Acción del actor	Respuesta del sistema
	5. Muestra en el formulario una sección para actualizar el documento seleccionado. Se solicitan los siguientes campos: <ul style="list-style-type: none"> ● Nuevo nombre ● Cantidad de personas ● Nueva fecha de creación
6. Selecciona un registro de la tabla de documentos existentes	7. Muestra los datos actuales del documento seleccionado por los campos de entrada de texto.
8. Introduce nuevos valores para los campos.	
9. Selecciona el botón Actualizar.	10. Invoca al CU Actualizar documento .
11. Selecciona el botón CERRAR	12. Cierra la ventana.
Flujo alternativo 8a Entrada de texto en campo numérico	
Acción del actor	Respuesta del sistema
8a.1. Intenta introducir letras en el campo de entrada de texto Cantidad de personas.	8a.2. Inhabilita la opción de escribir sobre este campo. Ir a la acción 2 del CU.

2.4.7.7 CU: Actualizar documento

CU-16	Actualizar documento
Propósito	Tiene como objetivo actualizar los datos de un documento en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto selecciona el botón Actualizar de la ventana Administrar Documentos.
Referencias	RF 1.2

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
	1. Comprueba que exista un registro marcado en la tabla de documentos existentes.
	2. Comprueba que los campos de entrada de textos con los datos a actualizar contengan valores.
	3. Actualiza el registro correspondiente en la tabla de la BD.
	4. Emite un mensaje "Documento actualizado satisfactoriamente".
5. Selecciona el botón OK.	6. Ir a la acción 2 del CU Administrar Documentos .
Flujo alternativo 1a No existe registro marcado	
Acción del actor	Respuesta del sistema
	1a.1. Comprueba que no existe al menos un documento seleccionado en la tabla para actualizar.
	1a.2. Muestra un mensaje de error. "No hay ningún documento seleccionado para actualizar"
1a.3. Selecciona el botón OK.	1a.4. Ir a la acción 2 del CU Administrar Documentos .
Flujo alternativo 2a Campos en blanco	
Acción del actor	Respuesta del sistema
	2a.1. Comprueba que hay campos en blanco.
	2a.2. Muestra un mensaje de error "Campos en blanco".

2a.3. Selecciona el botón OK.	2a.4. Ir a la acción 2 del CU Administrar Documentos .
-------------------------------	---

2.4.7.8 CU: Eliminar documento

CU-17	Eliminar documento
Propósito	Tiene como objetivo eliminar un documento en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto selecciona el botón Eliminar de la ventana Administrar Documentos.
Referencias	RF 1.3

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
	1. Comprueba que exista al menos un registro marcado en la tabla de documentos existentes para eliminar.
	2. Emite un mensaje preguntando si desea eliminar el/los documento(s) seleccionados.
3. Selecciona el botón SI	3. Elimina el/los registro(s) correspondiente(s) en la tabla de la BD.
	4. Emite un mensaje "Documento(s) eliminado(s) satisfactoriamente".
5. Selecciona el botón OK.	6. Ir a la acción 2 del CU Administrar Documentos .
Flujo alternativo 1a No existe registro marcado	
Acción del actor	Respuesta del sistema
	1a.1. Comprueba que no existe al menos un documento seleccionado en la tabla para eliminar.
	1a.2. Muestra un mensaje de error. "No hay documentos seleccionados para eliminar"
1a.3. Selecciona el botón OK.	1a.4. Ir a la acción 2 del CU Administrar Documentos .
Flujo alternativo 2a NO	
Acción del actor	Respuesta del sistema

2a.1. Selecciona la opción NO	2a.2. Ir a la acción 2 del CU Administrar Documentos .
-------------------------------	---

2.4.7.9 CU: Insertar Caso de Uso

Prototipo de interfaz de usuario:



CU-18	Insertar Caso de Uso
Propósito	Tiene como objetivo insertar los datos de los casos de uso que componen a un documento en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto va al menú Datos del proyecto de la página principal del software de gestión, selecciona la opción Caso de Uso y dentro el submenú Insertar Caso de uso. Introduce los datos guardándose los mismos en la base de datos.
Referencias	RF 1.1, RF 1.4

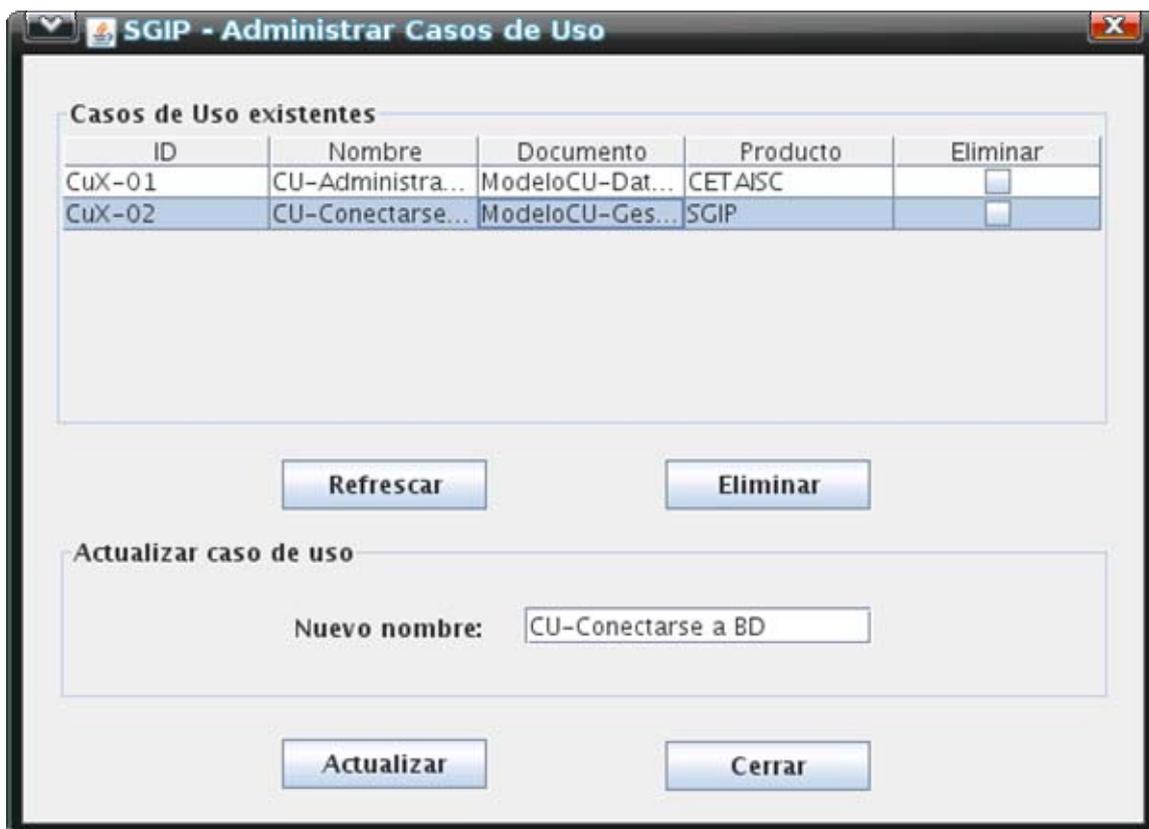
Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
	1. Muestra un formulario con una tabla donde se listan los siguientes datos de los documentos existentes: <ul style="list-style-type: none"> ● Código ● Nombre ● Fecha ● Producto al que pertenece
	2. Muestra una sección del formulario solicitando los siguientes datos del caso de uso: <ul style="list-style-type: none"> ● Código ● Nombre
3. Selecciona un documento	
4. Introduce los datos solicitados	
5. Selecciona la opción Insertar	6. Comprueba que haya un registro seleccionado en la tabla donde se listan los documentos.
	7. Comprueba que todos los campos de entrada de datos tengan valores.
	8. Almacena en la BD la información introducida.
	9. El sistema muestra un mensaje “Caso de uso insertado satisfactoriamente.”
10. Selecciona el botón OK.	11. Ir a la acción 1,2 del CU.
12. Selecciona el botón CERRAR	13. Cierra la ventana.
Flujo alternativo 6a Registro no seleccionado en la tabla	
Acción del actor	Respuesta del sistema
	6a.1. Comprueba que no hay un registro seleccionado en la tabla donde se listan los documentos.
	6a.2. Muestra un mensaje de error “Debe seleccionar a qué documento pertenecerá este caso de uso”.
6a.3. Selecciona el botón OK.	6a.4. Ir a la acción 1,2 del CU.
Flujo alternativo 7a Campos en blanco	
Acción del actor	Respuesta del sistema
	7a.1. Comprueba que hay campos en blanco.

	7a.2. Muestra un mensaje de error “Campos en blanco”.
7a.3. Selecciona el botón OK.	7a.4. Ir a la acción 2 del CU.
Flujo alternativo 8a Registro existente en la BD	
Acción del actor	Respuesta del sistema
	8a.1. Comprueba que hay un registro con el mismo identificador en la base de datos.
	8a.2. Muestra un mensaje de error “Ya ha sido registrado un caso de uso con este identificador”.
8a.3. Selecciona el botón OK.	8a.4. Ir a la acción 2 del CU.

2.4.7.10 CU: Administrar Casos de Uso

Prototipo de interfaz de usuario:



CU-19	Administrar Casos de Uso
Propósito	Tiene como objetivo visualizar, actualizar o eliminar los datos de un caso de uso en el sistema de gestión.

Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto va al menú Datos del proyecto de la página principal del software de gestión, selecciona la opción Caso de Uso y dentro el submenú Administrar Casos de Uso.
Referencias	FR 1.2, RF 1.3, RF 1.4

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
1. Selecciona el botón Refrescar	2. Se listan en una tabla los casos de uso existentes en la BD, especificando los siguientes datos: <ul style="list-style-type: none"> ● Código ● Nombre ● Documento al que pertenece ● Producto al que pertenece
3. Selecciona el botón Eliminar	4. Se invoca al CU Eliminar caso de uso .
Sección Actualizar caso de uso	
Acción del actor	Respuesta del sistema
	5. Muestra en el formulario una sección para actualizar el caso de uso seleccionado. Se solicitan los siguientes campos: <ul style="list-style-type: none"> ● Nuevo nombre
6. Selecciona un registro de la tabla de casos de uso existentes	7. Muestra los datos actuales del caso de uso seleccionado por los campos de entrada de texto.
8. Introduce nuevos valores para los campos.	
9. Selecciona el botón Actualizar.	10. Invoca al CU Actualizar caso de uso .
11. Selecciona el botón CERRAR	12. Cierra la ventana.

2.4.7.11 CU: Actualizar caso de uso

CU-20	Actualizar caso de uso
Propósito	Tiene como objetivo actualizar los datos de un caso de uso en el sistema de gestión.

Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto selecciona el botón Actualizar de la ventana Administrar Casos de Uso.
Referencias	RF 1.2

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
	1. Comprueba que exista un registro marcado en la tabla de casos de uso existentes.
	2. Comprueba que los campos de entrada de textos con los datos a actualizar contengan valores.
	3. Actualiza el registro correspondiente en la tabla de la BD.
	4. Emite un mensaje “Caso de uso actualizado satisfactoriamente”.
5. Selecciona el botón OK.	6. Ir a la acción 2 del CU Administrar Casos de Uso .
Flujo alternativo 1a No existe registro marcado	
Acción del actor	Respuesta del sistema
	1a.1. Comprueba que no existe al menos un caso de uso seleccionado en la tabla para actualizar.
	1a.2. Muestra un mensaje de error. “No hay ningún caso de uso seleccionado para actualizar”
1a.3. Selecciona el botón OK.	1a.4. Ir a la acción 2 del CU Administrar Casos de Uso .
Flujo alternativo 2a Campos en blanco	
Acción del actor	Respuesta del sistema
	2a.1. Comprueba que hay campos en blanco.
	2a.2. Muestra un mensaje de error “Campos en blanco”.
2a.3. Selecciona el botón OK.	2a.4. Ir a la acción 2 del CU Administrar Casos de Uso .

2.4.7.12 CU: Eliminar caso de uso

CU-21	Eliminar caso de uso
Propósito	Tiene como objetivo eliminar un caso de uso en el sistema de gestión.
Actores	Jefe de Proyecto
Descripción	El caso de uso se inicia cuando el Jefe de un proyecto selecciona el botón Eliminar de la ventana Administrar Casos de Uso.
Referencias	RF 1.3

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
	1. Comprueba que exista al menos un registro marcado en la tabla de casos de uso existentes para eliminar.
	2. Emite un mensaje preguntando si desea eliminar el/los caso(s) de uso(s) seleccionados.
3. Selecciona el botón SI	3. Elimina el/los registro(s) correspondiente(s) en la tabla de la BD.
	4. Emite un mensaje "Caso(s) de uso(s) eliminado(s) satisfactoriamente".
5. Selecciona el botón OK.	6. Ir a la acción 2 del CU Administrar Casos de Uso .
Flujo alternativo 1a No existe registro marcado	
Acción del actor	Respuesta del sistema
	1a.1. Comprueba que no existe al menos un caso de uso seleccionado en la tabla para eliminar.
	1a.2. Muestra un mensaje de error. "No hay documentos seleccionados para eliminar"
1a.3. Selecciona el botón OK.	1a.4. Ir a la acción 2 del CU Administrar Casos de Uso .
Flujo alternativo 2a NO	
Acción del actor	Respuesta del sistema
2a.1. Selecciona la opción NO	2a.2. Ir a la acción 2 del CU Administrar Casos de Uso .

2.4.7.13 CU: Insertar Documento en Carpeta

Prototipo de interfaz de usuario:

SGIP - Asociar documentos con carpetas del repositorio

Seleccione un documento de especificacion

Documentos existentes

ID	Nombre	Fecha	Producto
Doc-01	Descripcion-CU1	06-may-2008	SGIP
Doc-02	Documento2	06-may-2008	SGIP
Doc-03	Especificacion2 1	02-may-2008	SIGEP

Repositorio del proyecto

URL:

Seleccione la carpeta "src" del proyecto:

Carpeta(s) donde se implementa el documento seleccionado...

Direccion relativa:

CU-22	Insertar Documento en Carpeta
Propósito	Tiene como objetivo insertar en el sistema de gestión los datos de los nodos (carpetas) que componen la jerarquía del repositorio y asociar qué documentos se estarán implementando en cada uno.

Actores	Arquitecto
Descripción	El caso de uso se inicia cuando el arquitecto de un proyecto va al menú Datos del proyecto de la página principal del software de gestión, selecciona la opción Repositorio y dentro el submenú Asociar documento a carpetas.
Referencias	RF 1.1, RF 1.4

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
	1. Se listan en una tabla los documentos existentes en la BD, especificando los siguientes datos: <ul style="list-style-type: none"> ● Código ● Nombre ● Fecha ● Producto al que pertenece
	2. Muestra una sección del formulario solicitando los siguientes datos del repositorio: <ul style="list-style-type: none"> ● URL
3. Selecciona un documento de la tabla.	
4. Introduce los datos solicitados	
5. Selecciona el botón Examinar	5. Muestra un OpenFileDialog con las carpetas que existen en el disco local. Se aplica un filtro para listar solo las carpetas.
6. Selecciona el botón Abrir	7. Define la carpeta raíz donde se maneja el código fuente del desarrollo del proyecto de software en Java. (usualmente "src")
	8. Muestra por el control de entrada de texto de solo lectura, ubicado al lado del botón Examinar, la dirección completa donde se encuentra la carpeta contenedora de "src".
	9. Muestra por el Jtree la jerarquía de las subcarpetas del proyecto. La raíz del árbol tendrá el nombre de la carpeta contenedora.
10. Selecciona un nodo (carpeta) del árbol	11. Muestra por un control de entrada de texto de solo

de carpetas del Jtree.	lectura la dirección del nodo seleccionado.
12. Selecciona el botón Aceptar	13. Comprueba que se haya seleccionado la carpeta que define la raíz del repositorio.
	14. Comprueba que haya un registro seleccionado en la tabla donde se listan los documentos.
	15. Comprueba que el control de entrada de texto de solo lectura con la dirección del nodo seleccionado tenga valor.
	16. Almacena en la BD la información introducida.
	17. El sistema muestra un mensaje "Carpeta insertada satisfactoriamente."
18. Selecciona el botón OK.	19. Ir a la acción 1 del CU.
20. Selecciona el botón CERRAR	21. Cierra la ventana.
Flujo alternativo 14a Registro no seleccionado en la tabla	
Acción del actor	Respuesta del sistema
	14a.1. Comprueba que no hay un registro seleccionado en la tabla donde se listan los documentos.
	14a.2. Muestra un mensaje de error "Debe seleccionar un documento".
14a.3. Selecciona el botón OK.	14a.4. Ir a la acción 1 del CU.
Flujo alternativo 15a Campos en blanco	
Acción del actor	Respuesta del sistema
	15a.1. Comprueba que al menos uno o los dos controles de entrada de texto donde se muestra el camino y nombre del nodo seleccionado tienen campos en blanco.
	15a.2. Muestra un mensaje de error "Debe seleccionar una carpeta del proyecto para establecer la relación".
15a.3. Selecciona el botón OK.	15a.4. Ir a la acción 1 del CU.
Flujo alternativo 10a Selecciona la carpeta raíz del árbol de directorios	
Acción del actor	Respuesta del sistema
10a.1. Selecciona la raíz del árbol de carpetas del proyecto	10a.2. Muestra un mensaje de advertencia por el control de entrada de texto de solo lectura referente a

	Dirección relativa: "No valido. Seleccione dentro de la jerarquía inferior".
10a.3. Selecciona el botón OK.	10a.4. Ir a la acción 1 del CU.
Flujo alterno 10b Selecciona una carpeta que contiene ".svn" en la dirección.	
Acción del actor	Respuesta del sistema
10b.1. Selecciona la carpeta ".svn" dentro de la jerarquía de carpetas del proyecto mostradas por el Jtree.	10b.2. Muestra un mensaje de advertencia por los dos controles de entrada de textos de solo lectura "La carpeta seleccionada no es válida para implementar documentos porque pertenece al control de versiones del proyecto".
10b.3. Selecciona el botón OK.	10b.4. Ir a la acción 1 del CU.
Flujo alterno 13a Carpeta raíz del repositorio no seleccionada	
Acción del actor	Respuesta del sistema
	13a.1. Comprueba que el campo de entrada de texto con la dirección de la carpeta raíz del proyecto está en blanco.
	13a.2. Muestra un mensaje de error "Debe seleccionar la carpeta raíz donde se maneja el desarrollo del proyecto".
13a.3. Selecciona el botón OK.	13a.4. Ir a la acción 1 del CU.
Flujo alterno 16a Registro existente en la BD	
Acción del actor	Respuesta del sistema
	16a.1. Comprueba que hay un registro con el mismo identificador en la base de datos.
	16a.2. Muestra un mensaje de error "Ya ha sido registrada una carpeta con esta dirección para el repositorio especificado.".
16a.3. Selecciona el botón OK.	16a.4. Ir a la acción 1 del CU.

2.4.7.14 CU: Administrar Carpetas

Prototipo de interfaz de usuario:



CU-23	Administrar Carpetas
Propósito	Tiene como objetivo visualizar/eliminar los datos de una carpeta existente en el repositorio del proyecto en el sistema de gestión.
Actores	Arquitecto
Descripción	El caso de uso se inicia cuando el arquitecto va al menú Datos del proyecto de la página principal del software de gestión, selecciona la opción Repositorio y dentro el submenú Administrar carpetas.
Referencias	RF 1.3, RF 1.4

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
1. Selecciona el botón Refrescar	2. Se listan en una tabla las carpetas del repositorio existentes en la BD, especificando los siguientes datos: <ul style="list-style-type: none"> ● URL Repositorio. ● Dirección relativa ● Documento al que pertenece

	<ul style="list-style-type: none"> ● Producto al que pertenece ● Opción de eliminar
3. Selecciona el botón Eliminar	4. Se invoca al CU Eliminar carpeta .
5. Selecciona el botón CERRAR	6. Cierra la ventana.

2.4.7.15 CU: Eliminar carpeta

CU-24	Eliminar carpeta
Propósito	Tiene como objetivo eliminar una carpeta existente en el repositorio del proyecto en el sistema de gestión.
Actores	Arquitecto
Descripción	El caso de uso se inicia cuando el arquitecto selecciona el botón Eliminar de la ventana Administrar carpetas del repositorio.
Referencias	RF 1.3

Tabla de eventos

Flujo básico	
Acción del actor	Respuesta del sistema
	1. Comprueba que exista al menos un registro marcado en la tabla de carpetas del repositorio existentes para eliminar.
	2. Emite un mensaje preguntando si desea eliminar la(s) carpeta(s) seleccionada(s).
3. Selecciona el botón SI	3. Elimina el/los registro(s) correspondiente(s) en la tabla de la BD.
	4. Emite un mensaje "Carpeta(s) eliminada(s) satisfactoriamente".
5. Selecciona el botón OK.	6. Ir a la acción 2 del CU Administrar Carpetas .
Flujo alternativo 1a No existe registro marcado	
Acción del actor	Respuesta del sistema
	1a.1. Comprueba que no existe al menos una carpeta seleccionada en la tabla para eliminar.
	1a.2. Muestra un mensaje de error. "No hay carpetas seleccionadas para eliminar"

1a.3. Selecciona el botón OK.	1a.4. Ir a la acción 2 del CU Administrar Carpetas .
Flujo alternativo 2a NO	
Acción del actor	Respuesta del sistema
2a.1. Selecciona la opción NO	2a.2. Ir a la acción 2 del CU Administrar Carpetas .

2.5 Servicio de Integración de Estadísticas del StatSVN (SVNStatBash)

Este servicio es un archivo bash (.sh) para plataforma UNIX. Se ejecutará a una hora determinada del día y se encargará de procesar los datos de las estadísticas del repositorio de Subversion (productividad de los programadores en la jornada), insertando esta información en la base de datos del Sistema de Gestión de Indicadores. También generará una suite HTML con reportes extras sobre la actividad de los programadores sobre el repositorio.

A continuación se exponen los requisitos funcionales del servicio, se explica el modelo de la base de datos del software de gestión donde este servicio inserta los tamaños, se muestra la Arquitectura y varios diagramas de clases.

2.5.1 Requisitos funcionales

RF1: Reutilizar el código fuente de StatCVS y StatSVN para aprovechar las funcionalidades que ya hacen estas aplicaciones relativas a la obtención de las estadísticas por jornada de la productividad de los programadores en cada nodo del repositorio.

RF2: Insertar los tamaños de desarrollo de la jornada diaria en la base de datos.

- RF2.1: Verificar la concordancia entre la información existente en la base de datos del Software de Gestión de Indicadores de Productividad (SGIP) y el repositorio en torno a programadores y carpetas.
- RF2.2: Si la verificación de RF2.1 arroja que existen discordancias no se inserta el tamaño correspondiente a la carpeta o programador no definidos en el software de gestión.
- RF2.3: Registrar en un archivo LOG los estados de las operaciones de inserción de los tamaños de desarrollo diarios (ya sean éxitos o fallos).

2.5.2 Modelo de la BD

Esta base de datos contiene datos de los proyectos y de la Organización del Software.

Se identificaron las siguientes entidades persistentes:

- Equipo.
- Programador.
- Producto
- Documento de Especificación de Caso de Uso.
- Caso de Uso
- Nodo (del árbol del repositorio de Subversion)

En la figura 2.5 se muestra el diseño correspondiente:

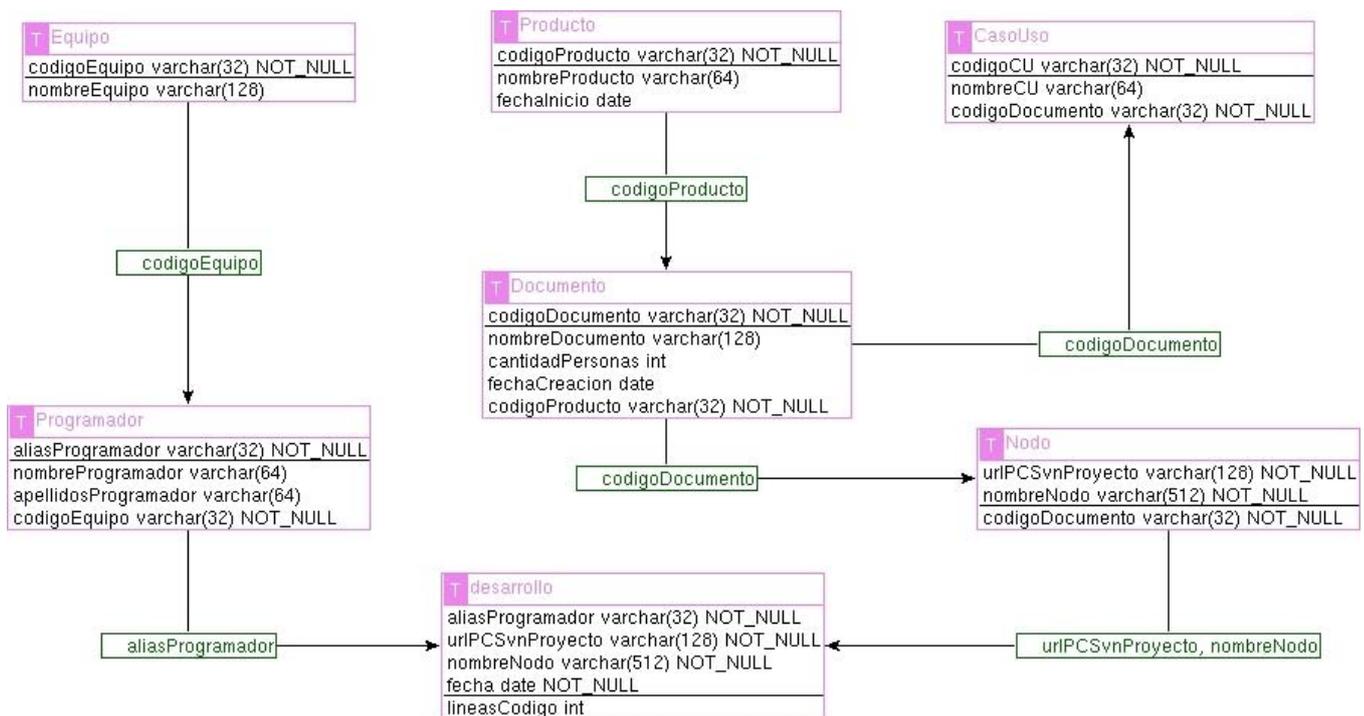


Figura 2.5 Base de datos donde se almacena la información necesaria para obtener los indicadores de productividad de esfuerzo de la Universidad.
(Diseño realizado con la herramienta de software libre Ferret)

2.5.3 Descripción de las tablas

Nombre: Equipo		
Descripción: Tabla que contiene la información de los equipos de desarrollo de software de la universidad.		
Atributo	Tipo	Descripción
codigoEquipo	varchar(32)	Identificador de la tabla Equipo.

nombreEquipo	varchar(128)	Nombre que identifica al equipo o proyecto.
--------------	--------------	---

Nombre: Programador		
Descripción: Tabla que contiene la información de los programadores de los proyectos.		
Atributo	Tipo	Descripción
aliasProgramador	varchar(32)	Alias que obtiene el programador en el repositorio de Subversion. Identificador de la tabla Programador.
nombreProgramador	varchar(64)	Nombre del programador.
apellidosProgramador	varchar(64)	Apellidos del programador
codigoEquipo	varchar(32)	Llave foránea que identifica a qué equipo o proyecto pertenece un programador.

Nombre: Desarrollo		
Descripción: Tabla que contiene la información relativa al desarrollo incremental que tienen los programadores en los diferentes nodos del repositorio de subversion.		
Atributo	Tipo	Descripción
aliasProgramador	varchar(32)	Alias del programador asociado al desarrollo insertado. Identificador de la tabla desarrollo.
urlPCSVnProyecto	varchar(128)	URL de la PC que contiene el repositorio al que pertenece el desarrollo insertado. Identificador de la tabla desarrollo.
nombreNodo	varchar(512)	Nombre del nodo del repositorio al que pertenece el desarrollo insertado. Identificador de la tabla desarrollo.
fecha	date	Fecha del desarrollo registrado. Identificador de la tabla desarrollo.
lineasCodigo	int	Tamaño asociado al desarrollo medido por cantidad de líneas de código.

Nombre: Nodo		
Descripción: Tabla que contiene la información de los nodos del repositorio de Subversion.		
Atributo	Tipo	Descripción
uriPCSVnProyecto	varchar(128)	Identificador de la tabla Nodo. Dirección URL donde queda ubicado el repositorio SVN.
nombreNodo	varchar(512)	Identificador de la tabla Nodo. Dirección relativa de la carpeta del repositorio.
codigoDocumento	varchar(32)	Llave foránea que identifica a qué documento tributa el contenido del nodo.

Nombre: Documento		
Descripción: Tabla que contiene la información de los documentos asociados a los productos.		
Atributo	Tipo	Descripción
codigoDocumento	varchar(32)	Identificador de la tabla Documento.
nombreDocumento	varchar(128)	Nombre que identifica al documento.
fechaCreación	date	Fecha en que se crea el documento.
cantidadpersonas	int	Cantidad de personas que intervinieron en la elaboración del documento.
codigoProducto	varchar(32)	Llave foránea que identifica a qué producto corresponde el documento.

Nombre: CasoUso		
Descripción: Tabla que recoge la información de los casos de uso contenidos en los documentos.		
Atributo	Tipo	Descripción
codigoCU	varchar(32)	Identificador de la tabla CasoUso.
nombreCU	varchar(64)	Nombre del caso de uso.
codigoDocumento	varchar(32)	Llave foránea. Identifica a qué documento pertenece.

Nombre: Producto		
Descripción: Tabla que contiene la información de los productos elaborados en la universidad.		
Atributo	Tipo	Descripción
codigoProducto	varchar(32)	Identificador de la tabla Producto.
nombreProducto	varchar(64)	Nombre del producto.
fechaInicio	date	Fecha oficial en que se decide comenzar el proyecto.

2.5.4 Arquitectura del Servicio

El servicio SVNStatBash se implementó de la siguiente manera:

1. Modificando el código fuente de StatCVS ¹, por medio de:
 - a) La adición de un paquete nombrado “**net.sf.statcvs.export**” con 3 nuevas clases, denominadas: **StatSVNReport**, **RepoNode** y **AuthorExport**.
 - b) La modificación de la clase **AuthorsforDirectoryTableReport**.

2. Modificando el código fuente de StatSVN ², por medio de:
 - a) La adición de un paquete “**net.sf.statsvn.fileObjectHandler**” con la clase **DataFileHandler**, encargada de serializar la información contenida en la clase **StatSVNReport**.
 - b) La modificación de la clase **Main** del paquete “**net.sf.statsvn**”.

Estas modificaciones tienen por objetivo permitir obtener una lista de los nodos del repositorio con cada uno de los tamaños por autores y la fecha del reporte. Esta información se hace persistir como objeto serializable utilizando la clase contenedora StatSVNReport.

3. Construyendo una aplicación, denominada SIEStatSVN, que lee (del objeto serializable) la clase contenedora con la lista anterior e inserta la información deseada en la base de datos, haciendo las comprobaciones pertinentes sobre la existencia de los datos del proyecto requeridos. Esta aplicación es un .jar que usa Hibernate.

¹ Código fuente con fecha 21 de abril del 2008. Disponible en <https://sourceforge.net/projects/statcvs/>

² Código fuente con fecha 4 de mayo del 2008. Disponible en <https://sourceforge.net/projects/statsvn/>

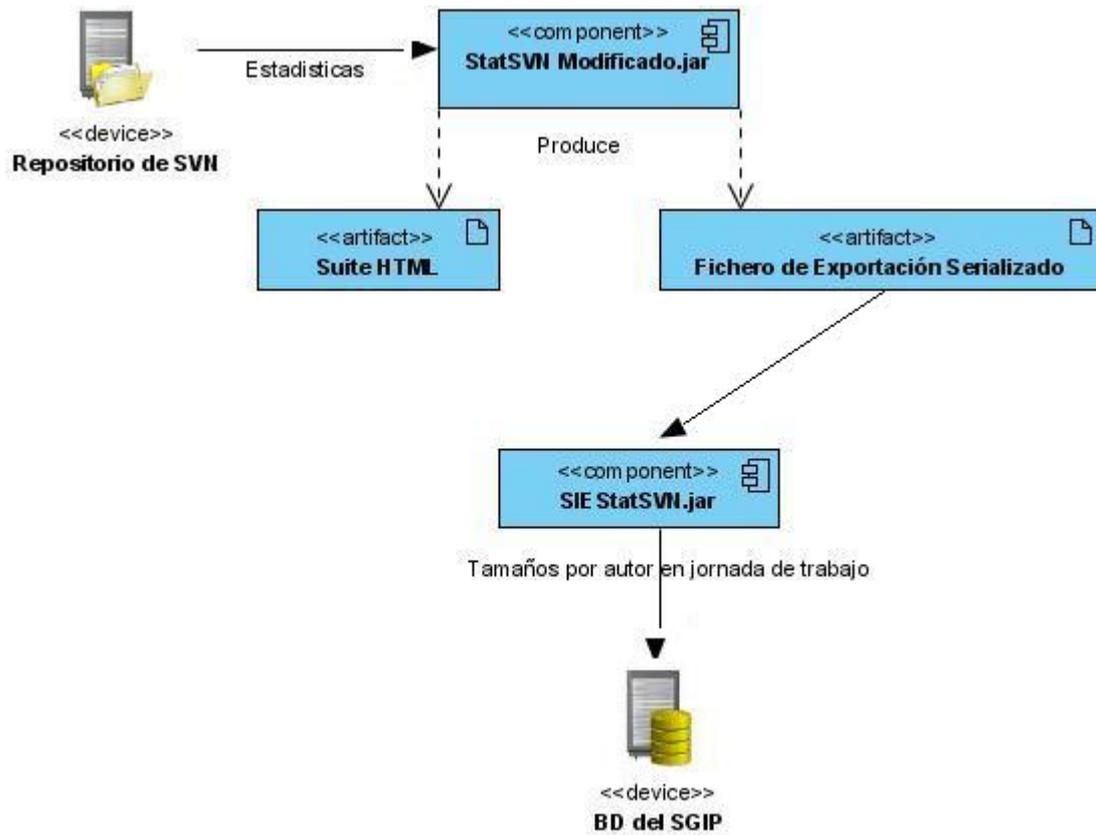


Figura 2.6 Arquitectura del Servicio SVNStatBash.
 (StatSVN Modificado.jar y SIE StatSVN.jar son aplicaciones de consola)

El código del servicio SVNStatBash aparece en el Anexo 1.

La figura 2.7 muestra el diagrama UML de clases correspondiente a las modificaciones realizadas al código de StatSVN.

Conclusiones del Capítulo

Este capítulo describe la concepción de las herramientas necesarias para obtener los indicadores de productividad de esfuerzo, mostrando el análisis y diseño de los artefactos correspondientes para implementar las mismas.

Se explican las fórmulas requeridas para obtener los indicadores que brinda la herramienta SGIP. Este sistema se documenta a través de las especificaciones de sus casos de uso y el diseño de su interfaz para el usuario.

Se describe el servicio SVNStatBash el cual se documenta a través de los diagramas de clases que implementan las aplicaciones construidas para el mismo.

Se explica el diseño de los esquemas para la base de datos del SGIP.

Toda esta documentación se elaboró utilizando las herramientas Visual Paradigm for UML Enterprise Edition y Ferret.

CAPÍTULO 3: Guías para la implantación y uso de las herramientas desarrolladas.

3.1 Introducción

En el presente capítulo se describen los pasos necesarios para el empleo por los proyectos productivos, de las herramientas construidas para garantizar la obtención de indicadores de productividad de esfuerzo. Se explica en qué momento se debe utilizar cada una y los deberes del Jefe de Proyecto y Arquitecto para que la tarea marche sin contratiempos.

3.2 Base de Datos del Software de Gestión

La instalación de la base de datos del software de gestión de los indicadores es la primera tarea que se debe realizar. Esta base de datos estará montada sobre un servidor Ubuntu Linux 8.04 y gestor PostgreSQL 8.3. Los pasos para su instalación y configuración se describen en el anexo 2 de este documento.

3.3 Nace un proyecto

Antes de comenzar a utilizar las herramientas propuestas en este trabajo es recomendable tener en cuenta que:

- El primer paso en la creación de todo proyecto es conformar el equipo de trabajo que llevará a cabo el desarrollo del producto software.
- En las fases de análisis y diseño de un proyecto surgen los documentos de especificación de los casos de uso que serán implementados por los programadores miembros del equipo de trabajo.

Una vez concluidas las fases de análisis y diseño, el Jefe del Proyecto y el Arquitecto están en condiciones de utilizar el Software de Gestión de Indicadores de Productividad (SGIP).

Para esto el jefe del proyecto debe rellenar algunos formularios que pueden ser accedidos fácilmente desde el menú de la ventana principal del SGIP.

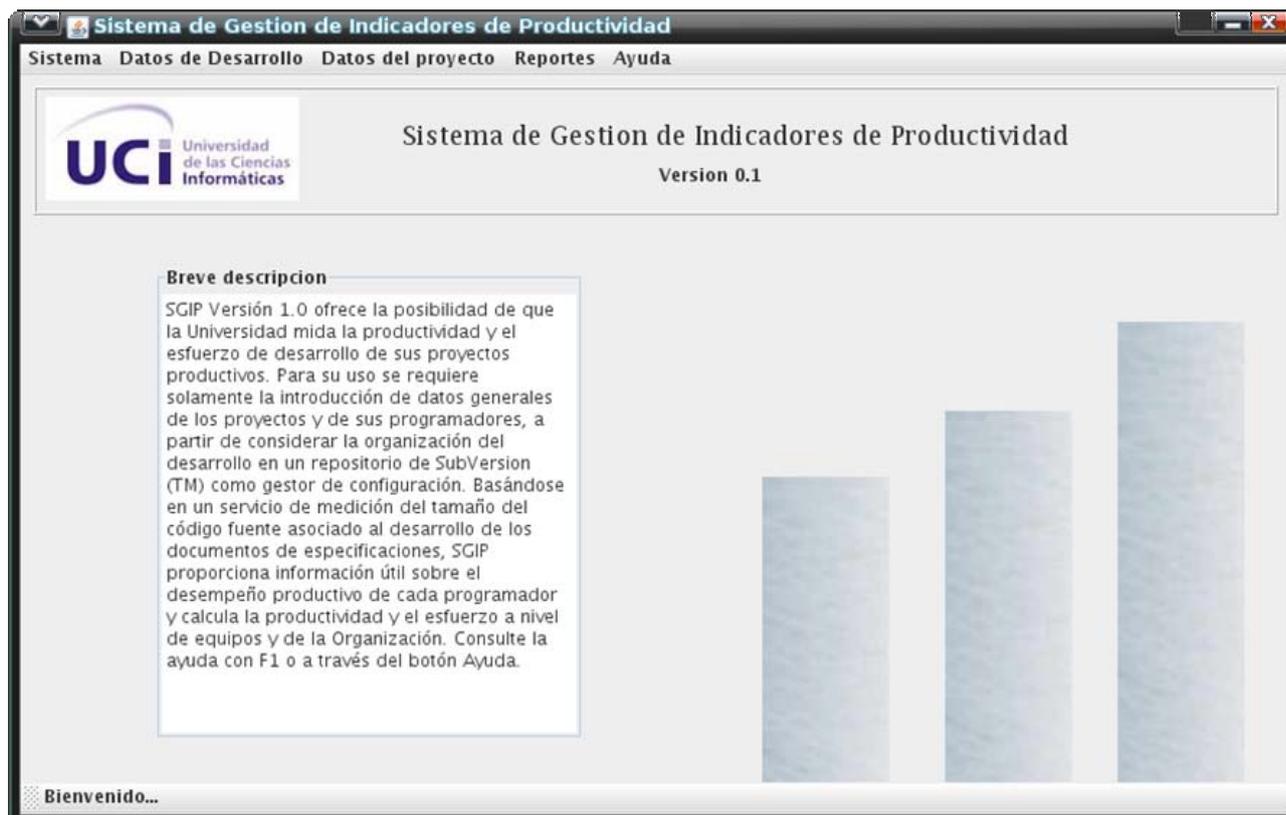


Figura 3.1 Ventana principal del SGIP.

En “Datos de Desarrollo” el Jefe de Proyecto maneja la información relativa a su equipo de desarrollo y los programadores que lo conforman. A la hora de insertar un equipo de desarrollo este usuario puede o no añadir los programadores que tendrá su equipo. Si desea añadir más adelante algún nuevo programador, puede escoger la opción Insertar Programador y definir a qué equipo pertenecerá. Igualmente puede administrar (eliminar o actualizar) tanto los equipos como el equipo que está bajo su mando y los programadores que lo conforman.

En “Datos del Proyecto” el Jefe de Proyecto maneja la información relativa el producto que se está construyendo, los documentos de especificación de dicho proyecto y los casos de uso que pertenecen a un documento.

Para cuando se terminen de definir los documentos de especificación, ya el arquitecto debe tener organizada la estructura del proyecto en las carpetas que contendrán los ficheros de código fuente. En este punto su misión será asociar qué documentos de especificación se estarán implementando dentro de cada carpeta. En la sección “Datos del proyecto/Repositorio/Asociar documento a carpetas” se puede establecer esta relación. Las carpetas que se registren en esta ventana son aquellas a las que

se desea contabilizar el tamaño producido en ellas. En la figura 3.2 aparece un ejemplo de estructura de carpetas de un proyecto en Java, estructura que es cargada en SGIP para permitir su uso en estas definiciones.

Cuando la arquitectura del proyecto está definida el arquitecto procede a alojarla en el repositorio de Subversion.

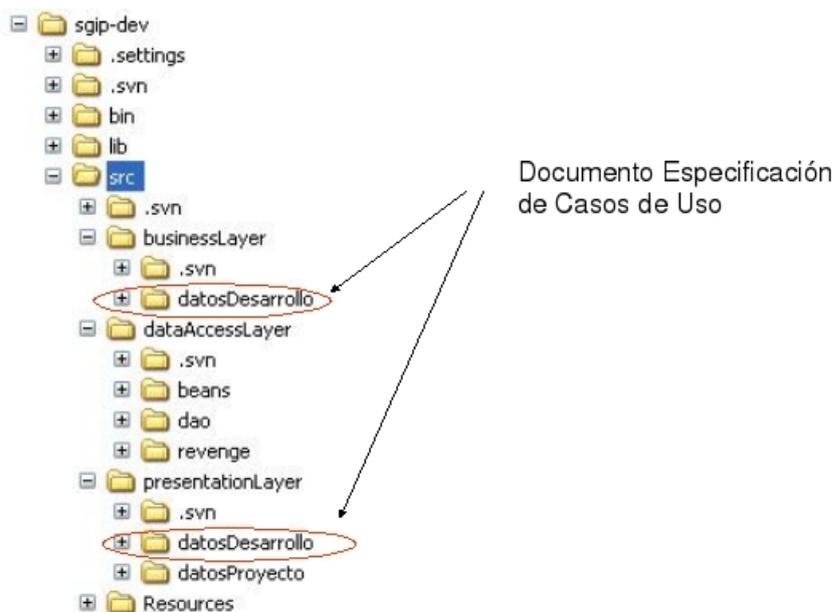


Figura 3.2 Estructura de carpetas de un proyecto en Java. Ejemplo de un documento asociado a dos carpetas.

3.4 La utilización de Subversion

El presente trabajo maneja la recolección de indicadores de productividad utilizando repositorios Subversion, por lo que cada proyecto del cual se quiera recoger estas estadísticas, debe utilizar un servidor de control de configuración SVN. Cada desarrollador debe tener instalado en su PC algún cliente para comunicarse con el repositorio. Con la utilización de TortoiseSVN para el caso de Windows o de RapidSVN en entornos UNIX bastará para realizar las operaciones básicas sobre el mismo. También son válidos aquellos clientes integrados a los IDEs como es el caso de Subclipse, plugin del Eclipse.

El repositorio SVN que se cree debe tener políticas de seguridad implementadas donde cada desarrollador contará con una credencial (alias y contraseña) que lo identificará para realizar las operaciones de subida de cambios en aquellas carpetas donde el administrador del repositorio le estableció el acceso. Este alias o nombre de usuario debe ser el mismo registrado en la ventana de

programadores del software de gestión, en caso contrario no se podrán registrar los tamaños asociados a ese programador.

Cualquier cambio en la estructura de carpetas del repositorio puede ser reflejado en el software de gestión en las ventanas “Administrar carpetas” y “Asociar documento a carpetas”.

El software de gestión podrá ser utilizado indistintamente desde entornos Windows o UNIX.

3.5 Recolectando los tamaños

Una vez definidos los datos de la fase de análisis y diseño del proyecto que se necesitan en el software de gestión, es de interés comenzar a recolectar los tamaños producidos de cada programador por carpetas existentes en el repositorio de gestión de configuración. Este paso se realiza haciendo uso del script **SVNStatBash.sh**, que con el auxilio del administrador de tareas programadas de Ubuntu, se pondrá a ejecutar una sola vez al día a una hora determinada (preferiblemente 12 am). Al hacer esta operación estamos convirtiendo el script en un servicio, el cual correrá automáticamente desde la PC con Ubuntu Linux Server. Como salidas se obtendrá una suite HTML con datos de interés sobre la actividad de los programadores en el repositorio SVN (ver anexo 3) y las líneas de código producidas por cada programador en cada nodo del repositorio.

La ejecución de este servicio necesita de algunos artefactos y cierta organización, detalles que se explican a continuación. Se propone mantener la siguiente estructura de carpetas en el servidor donde corra el mismo (Ver figura 3.3). El nombre de las carpetas puede estar sujeto a los deseos del usuario.

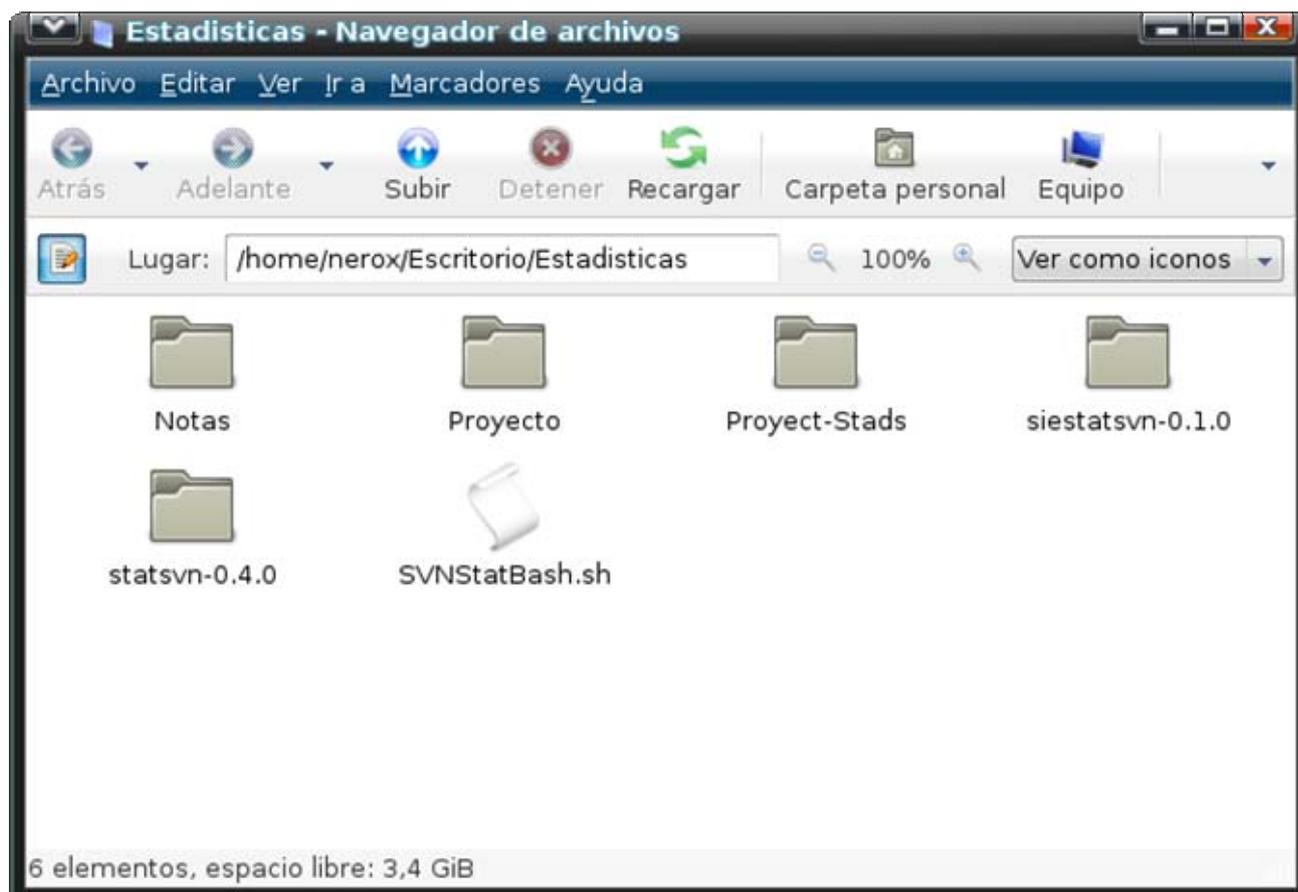


Figura 3.3 Estructura de carpetas para la ejecución de SVNStatBash.

En la carpeta *Notas* irá un archivo txt que contiene fragmentos de HTML que describen las características de interés del proyecto. Esta descripción irá como encabezado en la suite HTML generada. (Ver anexo 4)

La carpeta *Proyecto* se creará vacía. La misma se utilizará para recoger la copia de trabajo del proyecto alojado en el repositorio SVN.

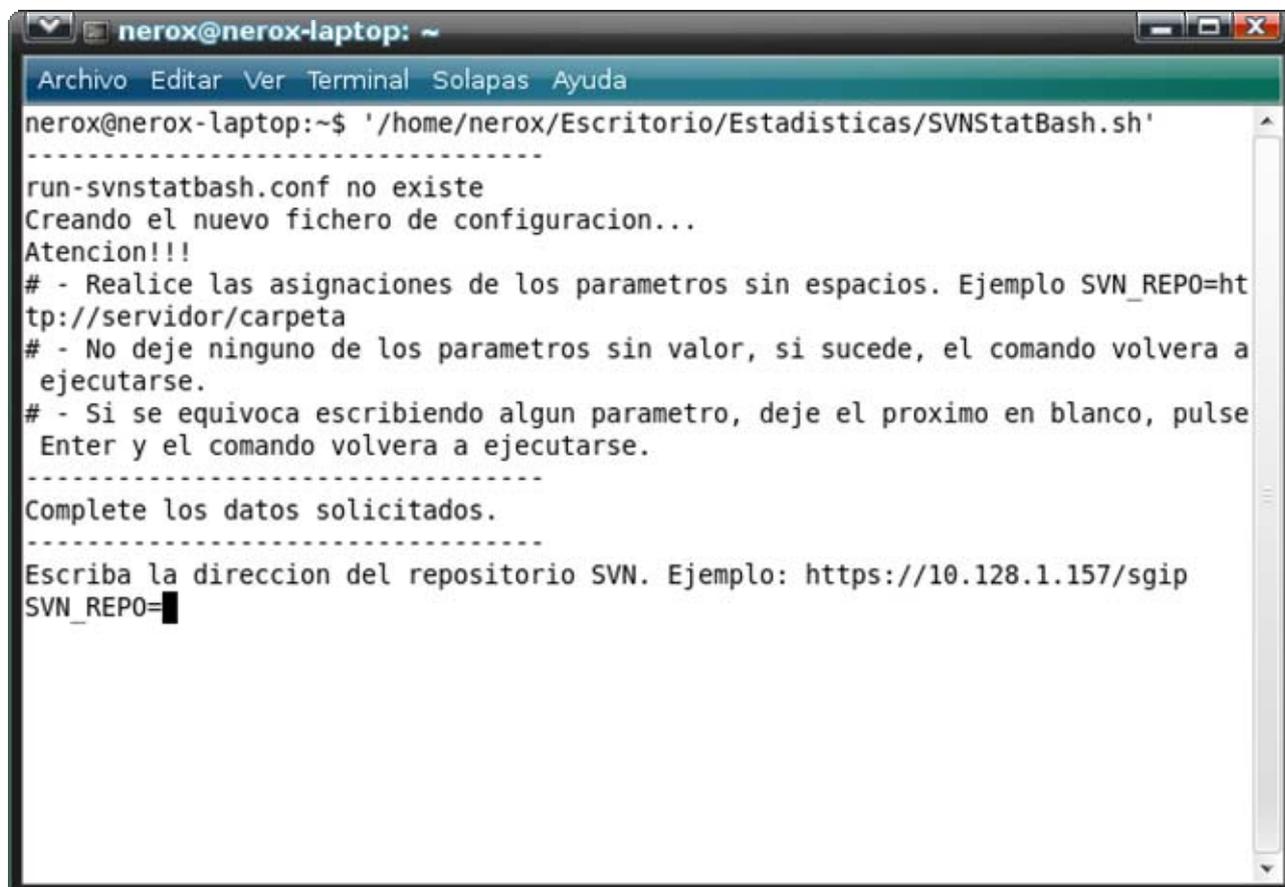
La carpeta *Proyect-Stads* se creará vacía. Contendrá la suite HTML.

La carpeta *siestatsvn-0.1.0* contendrá el ejecutable *siestatsvn.jar*. Dentro de esta propia carpeta aparecerá otra llamada "lib" que se debe dejar intacta junto con su contenido ya que dentro de ella se encuentran las librerías necesarias para hacer las operaciones de inserción de los tamaños en la BD del SGIP.

En la carpeta *statsvn-0.4.0* aparece el *statsvn.jar*, responsable de generar la suite HTML y exportar los datos necesarios a ser utilizados por el *siestatsvn.jar*.

Por último aparece el script SVNStatBash.sh.

Antes de poner a ejecutar este script como un servicio en Ubuntu se debe correr manualmente una primera vez utilizando para ello el usuario \$HOME.



```
nerox@nerox-laptop: ~  
Archivo Editar Ver Terminal Solapas Ayuda  
nerox@nerox-laptop:~$ '/home/nerox/Escritorio/Estadisticas/SVNStatBash.sh'  
-----  
run-svnstatbash.conf no existe  
Creando el nuevo fichero de configuracion...  
Atencion!!!  
# - Realice las asignaciones de los parametros sin espacios. Ejemplo SVN_REPO=ht  
tp://servidor/carpeta  
# - No deje ninguno de los parametros sin valor, si sucede, el comando volvera a  
ejecutarse.  
# - Si se equivoca escribiendo algun parametro, deje el proximo en blanco, pulse  
Enter y el comando volvera a ejecutarse.  
-----  
Complete los datos solicitados.  
-----  
Escriba la direccion del repositorio SVN. Ejemplo: https://10.128.1.157/sgip  
SVN_REPO=
```

Figura 3.4 Ejecución por primera vez de SVNStatBash.

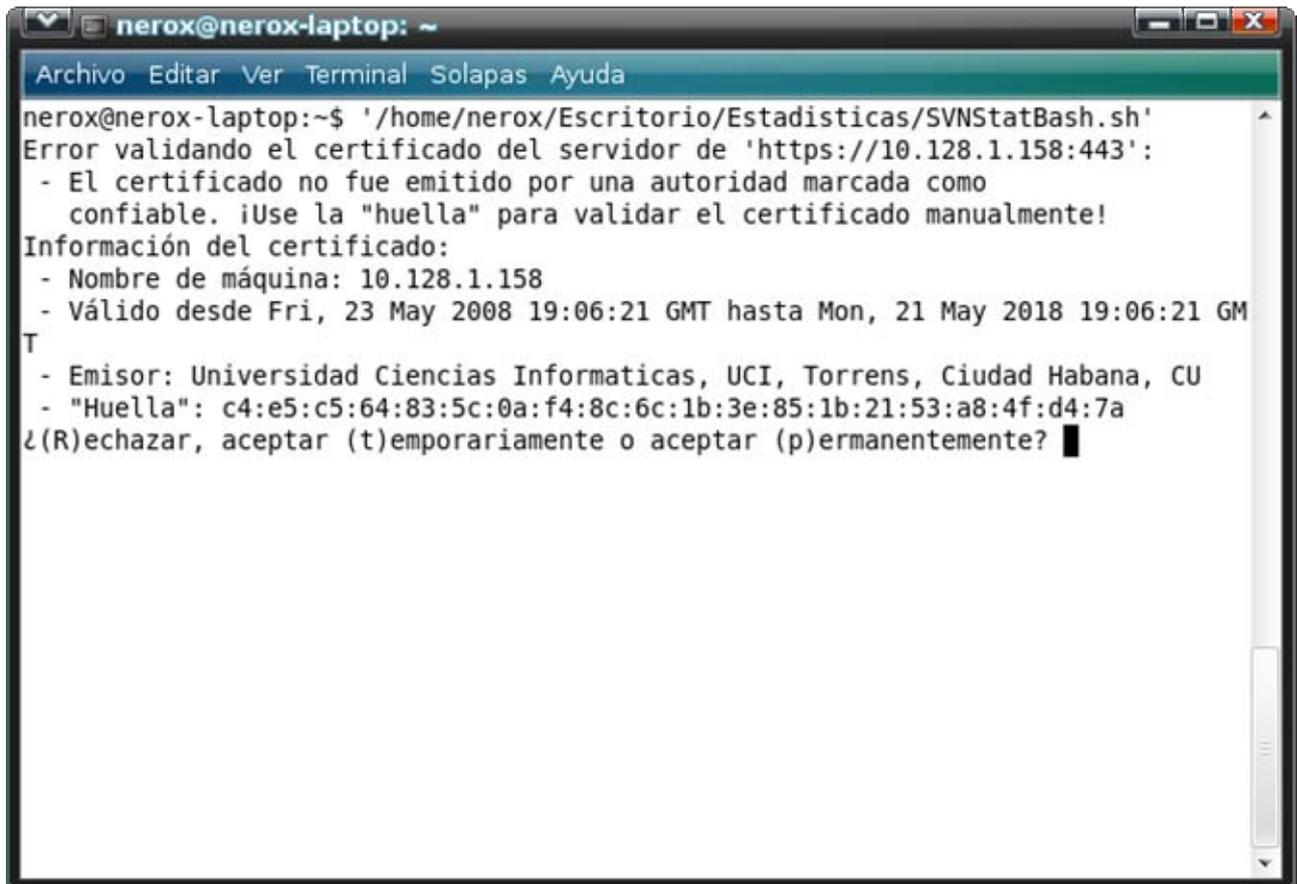
El script solicitará ciertos parámetros para poder construir un fichero de configuración. Si este archivo no existe, el script no podrá ejecutarse y pedirá al usuario que nuevamente se le pasen los parámetros para crear el archivo. Los parámetros solicitados son los siguientes:

- P1: Dirección del repositorio SVN.
- P2: Una credencial (usuario y contraseña) con nivel de acceso a todo el repositorio y de solo lectura.
- P3: Dirección de la carpeta donde se realizará la copia de trabajo del contenido del repositorio (working copy).
- P4: Nombre de la carpeta raíz de la copia de trabajo.

- P5: Dirección de la carpeta donde se desea generar la suite HTML con los reportes del repositorio.
- P6: Dirección de la carpeta donde se encuentra el ejecutable del StatSVN.
- P7: Dirección de la carpeta donde se encuentra el ejecutable del SieStatSVN.
- P8: Título que se muestra al usuario cuando se genera la suite HTML. Generalmente el nombre del proyecto.
- P9: Dirección del fichero de notas. (Descripción del proyecto que se muestra en la suite HTML generada).

Una vez que se suministran estos datos, el script le indicará al usuario el lugar donde ha sido creado el fichero de configuración y los pasos que debe seguir si desea encuestar otro repositorio.

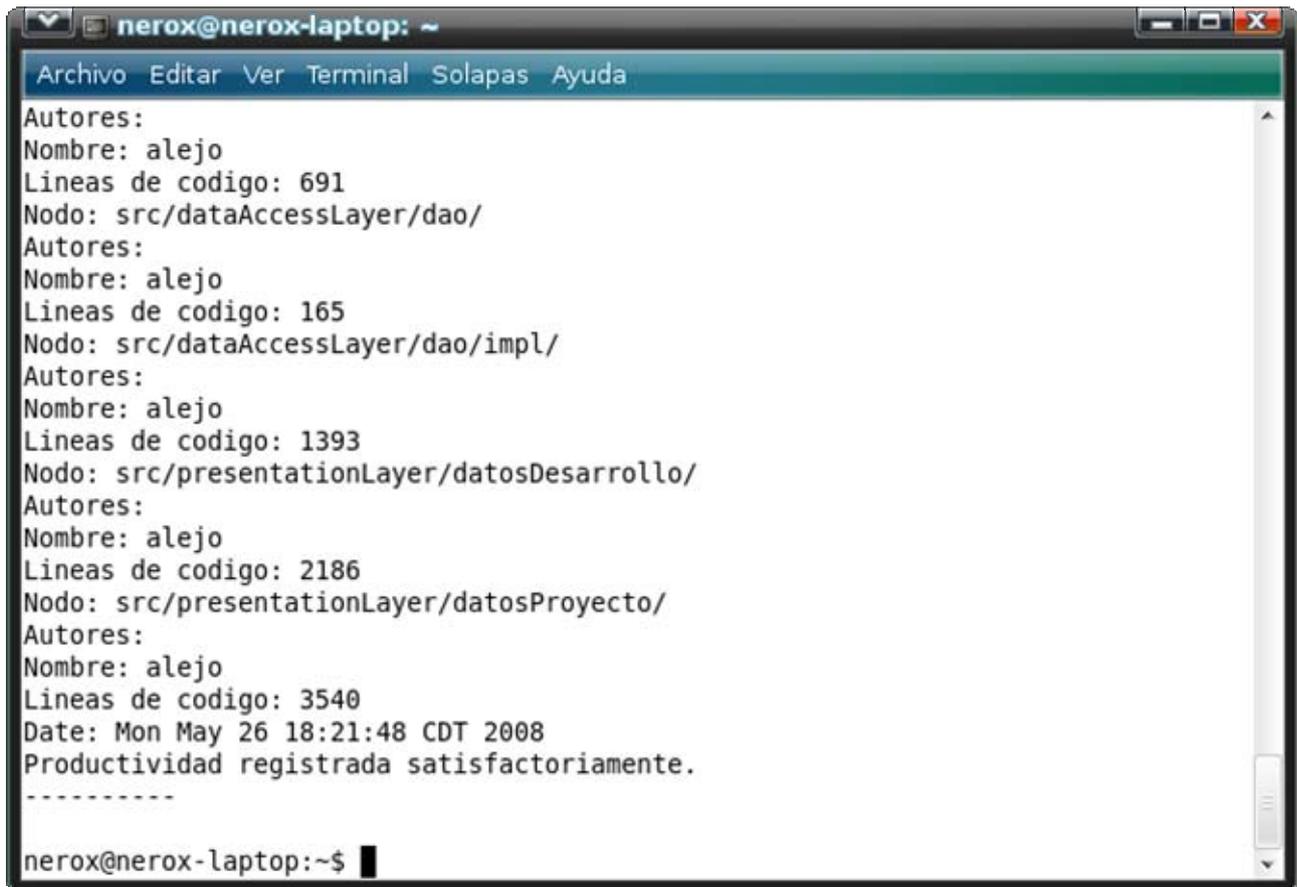
Si el administrador del repositorio SVN utiliza conexión segura con autenticación SSL sobre Apache (ejemplo: <https://serverIP/repo>), es muy recomendable volver a ejecutar este script manualmente. Esto se debe a que se pedirá al usuario cuánto tiempo desea aceptar este certificado de seguridad. Cuando esto suceda se selecciona la opción p (permanentemente).



```
nerox@nerox-laptop: ~  
Archivo Editar Ver Terminal Solapas Ayuda  
nerox@nerox-laptop:~$ '/home/nerox/Escritorio/Estadisticas/SVNStatBash.sh'  
Error validando el certificado del servidor de 'https://10.128.1.158:443':  
- El certificado no fue emitido por una autoridad marcada como  
  confiable. ¡Use la "huella" para validar el certificado manualmente!  
Información del certificado:  
- Nombre de máquina: 10.128.1.158  
- Válido desde Fri, 23 May 2008 19:06:21 GMT hasta Mon, 21 May 2018 19:06:21 GM  
T  
- Emisor: Universidad Ciencias Informaticas, UCI, Torrens, Ciudad Habana, CU  
- "Huella": c4:e5:c5:64:83:5c:0a:f4:8c:6c:1b:3e:85:1b:21:53:a8:4f:d4:7a  
¿(R)echazar, aceptar (t)emporariamente o aceptar (p)ermanentemente? █
```

Figura 3.5 Aceptando el certificado SSL.

Una vez pulsado Enter, el script volverá a ejecutarse y esta vez ya estará listo para ejercer su funcionalidad. (Ver figura 3.6)

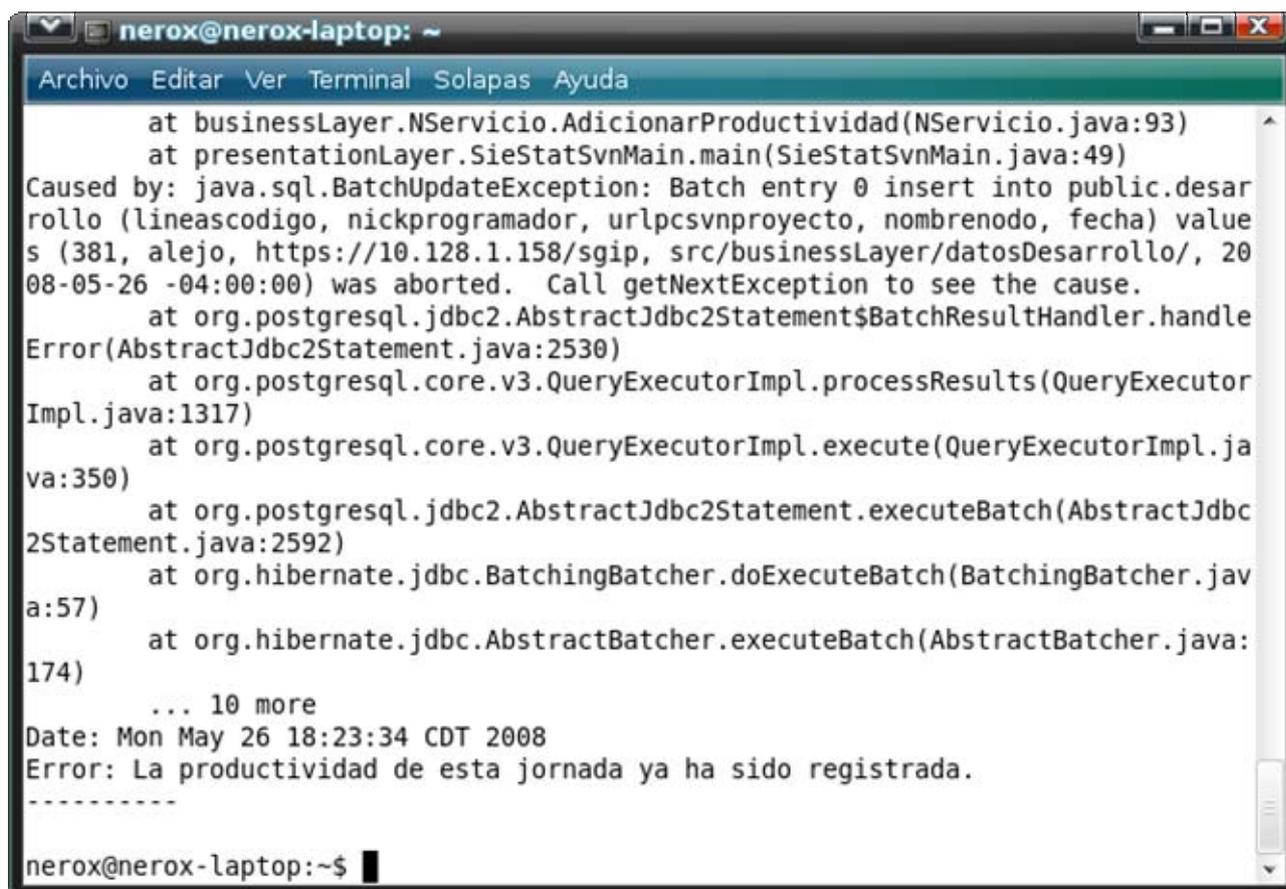
A screenshot of a terminal window titled 'nerox@nerox-laptop: ~'. The window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Terminal', 'Solapas', and 'Ayuda'. The terminal output shows four entries for different directories, each with the author 'alejo', the number of lines of code, and the node path. The last entry includes a date and a productivity message. The prompt 'nerox@nerox-laptop:~\$' is visible at the bottom.

```
nerox@nerox-laptop: ~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
Autores:
Nombre: alejo
Lineas de codigo: 691
Nodo: src/dataAccessLayer/dao/
Autores:
Nombre: alejo
Lineas de codigo: 165
Nodo: src/dataAccessLayer/dao/impl/
Autores:
Nombre: alejo
Lineas de codigo: 1393
Nodo: src/presentationLayer/datosDesarrollo/
Autores:
Nombre: alejo
Lineas de codigo: 2186
Nodo: src/presentationLayer/datosProyecto/
Autores:
Nombre: alejo
Lineas de codigo: 3540
Date: Mon May 26 18:21:48 CDT 2008
Productividad registrada satisfactoriamente.
-----
nerox@nerox-laptop:~$
```

Figura 3.6 Productividad registrada por SVNStatBash.

Si el script se ejecuta manualmente por tercera vez devolverá un mensaje de error indicando que la productividad de la jornada ya ha sido registrada. Esto es normal, ya que la instrucción está elaborada para que registre los tamaños una vez en el día (en la base de datos el campo fecha forma parte de la llave en la tabla que guarda los tamaños).

En la figura 3.8 se puede ver el tipo de mensaje correspondiente a este suceso.



```
nerox@nerox-laptop: ~
Archivo Editar Ver Terminal Solapas Ayuda
    at businessLayer.NServicio.AdicionarProductividad(NServicio.java:93)
    at presentationLayer.SieStatSvnMain.main(SieStatSvnMain.java:49)
Caused by: java.sql.BatchUpdateException: Batch entry 0 insert into public.desarrollo (lineascodigo, nickprogramador, urlpcsvnproyecto, nombrenodo, fecha) values (381, alejo, https://10.128.1.158/sgip, src/businessLayer/datosDesarrollo/, 2008-05-26 -04:00:00) was aborted. Call getNextException to see the cause.
    at org.postgresql.jdbc2.AbstractJdbc2Statement$BatchResultHandler.handleError(AbstractJdbc2Statement.java:2530)
    at org.postgresql.core.v3.QueryExecutorImpl.processResults(QueryExecutorImpl.java:1317)
    at org.postgresql.core.v3.QueryExecutorImpl.execute(QueryExecutorImpl.java:350)
    at org.postgresql.jdbc2.AbstractJdbc2Statement.executeBatch(AbstractJdbc2Statement.java:2592)
    at org.hibernate.jdbc.BatchingBatcher.doExecuteBatch(BatchingBatcher.java:57)
    at org.hibernate.jdbc.AbstractBatcher.executeBatch(AbstractBatcher.java:174)
    ... 10 more
Date: Mon May 26 18:23:34 CDT 2008
Error: La productividad de esta jornada ya ha sido registrada.
-----
nerox@nerox-laptop:~$
```

Figura 3.7 Error de SVNStatBash cuando trata de ejecutarse 2 veces el mismo día.

Si se han realizado todos estos pasos y conseguido el mismo resultado aquí descrito, el script ya está listo para ponerse a ejecutar como tarea programada (servicio) en Ubuntu 8.04. Para esto es necesario instalar la utilidad `gnome-schedule`. Una vez hecho esto, dirigirse a Sistema/Preferencias/Tareas Programadas.

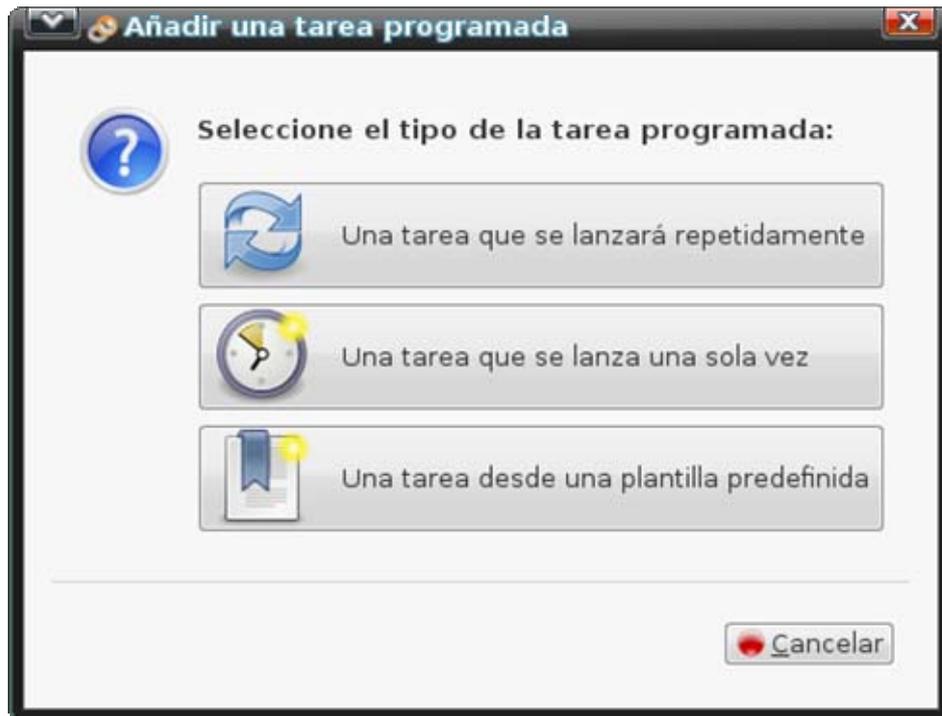


Figura 3.8 Ventana de opciones del administrador de tareas programadas de Ubuntu.

Seleccionar la opción “Una tarea que se lanzará repetidamente” y completar los datos de la ventana que se muestra de la siguiente manera:

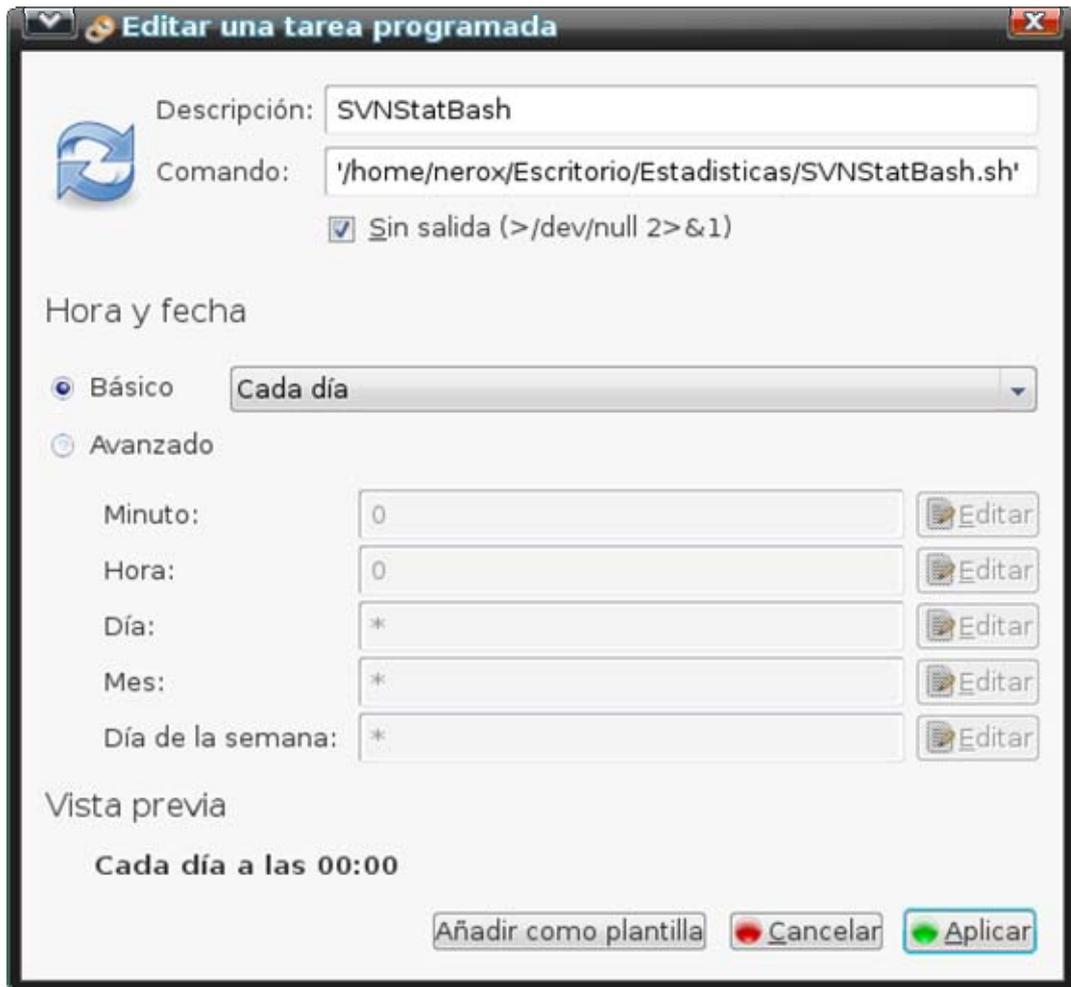


Figura 3.9 Configuración de la tarea programada.

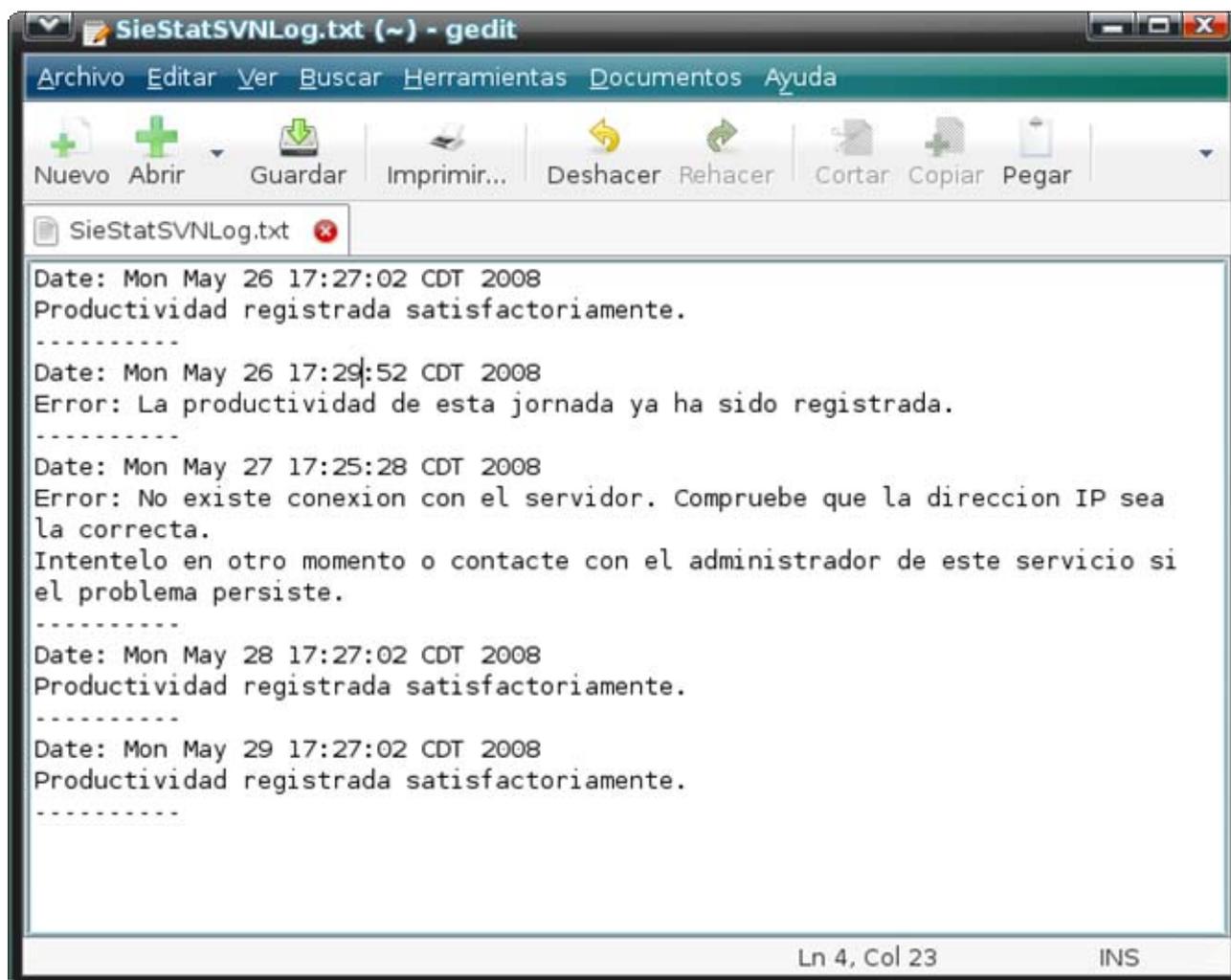
De esta manera todos los días a las doce de la noche SVNStatBash se lanzará como tarea programada y la productividad de los programadores en el día será recogida por el sistema. (Ver figura 3.10)

	nickprogramador [PK] character var	urlpcsvnproyecto [PK] character varying	nombrenodo [PK] character varying(512)	fecha [PK] date	lineascodigo integer
1	alejo	https://10.128.1.158/sgip	src/businessLayer/datosDesarrollo/	2008-05-10	266
2	alejo	https://10.128.1.158/sgip	src/businessLayer/datosDesarrollo/	2008-05-11	266
3	alejo	https://10.128.1.158/sgip	src/businessLayer/datosDesarrollo/	2008-05-12	294
4	alejo	https://10.128.1.158/sgip	src/businessLayer/datosDesarrollo/	2008-05-13	306
5	alejo	https://10.128.1.158/sgip	src/businessLayer/datosDesarrollo/	2008-05-14	379
6	alejo	https://10.128.1.158/sgip	src/businessLayer/datosDesarrollo/	2008-05-23	379
7	alejo	https://10.128.1.158/sgip	src/businessLayer/datosDesarrollo/	2008-05-24	379
8	alejo	https://10.128.1.158/sgip	src/businessLayer/datosDesarrollo/	2008-05-25	379
9	alejo	https://10.128.1.158/sgip	src/businessLayer/datosDesarrollo/	2008-05-26	381
10	alejo	https://10.128.1.158/sgip	src/businessLayer/datosProyecto/	2008-05-10	498
11	alejo	https://10.128.1.158/sgip	src/businessLayer/datosProyecto/	2008-05-11	498
12	alejo	https://10.128.1.158/sgip	src/businessLayer/datosProyecto/	2008-05-12	498
13	alejo	https://10.128.1.158/sgip	src/businessLayer/datosProyecto/	2008-05-13	498
14	alejo	https://10.128.1.158/sgip	src/businessLayer/datosProyecto/	2008-05-14	691
15	alejo	https://10.128.1.158/sgip	src/businessLayer/datosProyecto/	2008-05-23	691
16	alejo	https://10.128.1.158/sgip	src/businessLayer/datosProyecto/	2008-05-24	691
17	alejo	https://10.128.1.158/sgip	src/businessLayer/datosProyecto/	2008-05-25	691

55 filas.

Figura 3.10 Registro de tamaños producidos en la base de datos del sistema.

Como constancia de las operaciones que realiza este script, el usuario puede consultar el fichero log que se genera en \$HOME con el nombre SieStatSVNLog.txt.



```
SieStatSVNLog.txt
Date: Mon May 26 17:27:02 CDT 2008
Productividad registrada satisfactoriamente.
-----
Date: Mon May 26 17:29:52 CDT 2008
Error: La productividad de esta jornada ya ha sido registrada.
-----
Date: Mon May 27 17:25:28 CDT 2008
Error: No existe conexion con el servidor. Compruebe que la direccion IP sea
la correcta.
Intentelo en otro momento o contacte con el administrador de este servicio si
el problema persiste.
-----
Date: Mon May 28 17:27:02 CDT 2008
Productividad registrada satisfactoriamente.
-----
Date: Mon May 29 17:27:02 CDT 2008
Productividad registrada satisfactoriamente.
-----
Ln 4, Col 23  INS
```

Figura 3.11 SieStatSVNLog.txt.

Con SVNStatBash ejecutándose todos los días, el Jefe de Proyecto y las instancias interesadas están listas para obtener reportes sobre los indicadores de productividad de esfuerzo que podrán accederse desde la ventana principal del SGIP.

3.6 Modelo de implantación

La arquitectura que se propone para la implantación a nivel de la Universidad se puede apreciar en la figura 3.12.

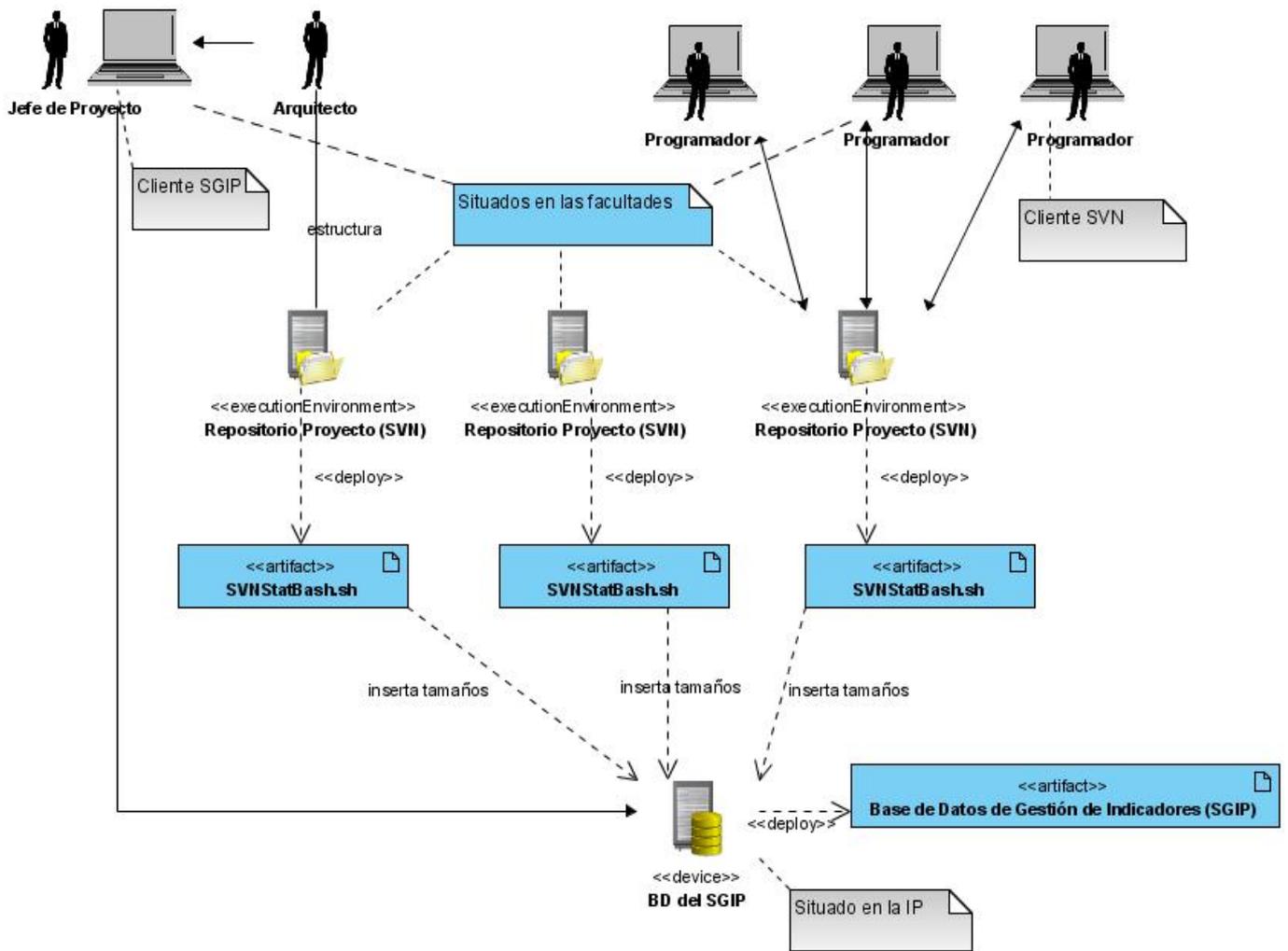


Figura 3.12 Implantación en la Universidad.

Conclusiones del Capítulo

En este capítulo se documentaron los pasos necesarios para el empleo por los proyectos productivos del servicio de recolección de estadísticas (SVNStatBash) y la correcta instalación de las aplicaciones clientes.

Se muestra cada paso por medio de imágenes ilustrativas y por ultimo se dibuja un diagrama general de la arquitectura propuesta para hacer funcionar este proyecto en la Universidad.

CONCLUSIONES

Este trabajo realiza una contribución al objetivo de poder estimar y controlar la productividad del trabajo en un entorno de desarrollo que lo requiere con mucha fuerza, al estar llamado a producir software con calidad en el menor tiempo posible.

El servicio de recolección de estadísticas a partir de los repositorios está completamente implementado. El software de gestión se encuentra implementado hasta la gestión de toda la información necesaria, faltando solamente la construcción de los reportes.

Los indicadores de productividad están preparados para ser obtenidos en función de las plataformas de desarrollo de la organización. Cada programador (equipo, producto, etc.) tendrá una productividad calculada en función de la plataforma.

Se ha obtenido un Sistema de Gestión de Indicadores en su versión 1.0 que tiene las siguientes bondades:

- Permite liberar personal del proyecto que se dedica a labores de control de la productividad.
- Permite la obtención de indicadores básicos del trabajo de la organización para un sistema de control de versiones específico.
- Sienta las bases para una mejora en la gestión de la organización a partir de su ambiente actual.
- Tiene como novedad incluir el tiempo de producción al añadir al tiempo de desarrollo los tiempos de duración de la etapa de análisis.
- Es una herramienta que permite controlar el desarrollo al hacer comparaciones de productividad entre programadores.

RECOMENDACIONES

- Implementar mecanismos de seguridad eficientes en ambas aplicaciones.

- Añadirle al Sistema de Gestión de Indicadores una interfaz que permita nutrir la base de datos (la línea base) con datos sobre los componentes ya creados que sean útiles para crear una disciplina de reuso:
 - aplicación donde está insertado
 - fecha de creación
 - descripción de su funcionalidad
 - lugar donde está salvado
 - plataforma de desarrollo

Datos que deberán definir los jefes de proyecto con cada producción de componente que sea reutilizable.

También se deberán incluir datos sobre la cantidad de defectos reportada a cada componente, a fin de poder medir el costo de los defectos.

- Desarrollar la cultura de gestión y sus herramientas asociadas para el aprovechamiento de estos indicadores, que influirán en las políticas de formación de personal.

- Dar continuidad a la investigación en los temas que han quedado abiertos: el desarrollo del Sistema de Dimensionamiento de Problemas y el desarrollo del Sistema de Cuadro de Mando Integral.

- Desarrollar un DataWareHouse para implementar la Línea Base, a partir de nutrir datos a la Línea Base con las diferentes bases de datos de cualquier sistema de repositorio.

REFERENCIAS BIBLIOGRÁFICAS

[Jones 1991]

Jones, Capers. Applied Software Measurement - Assuring Productivity and Quality. McGraw-Hill, 1991, ISBN 0-07-032813-7.

[Albrecht 1979]

Albrecht, A.J. "Measuring application development productivity", Proc. Joint Share/Guide/IBM Symposium on Application Development, Monterey, Calif., Oct.14-17, pp. 83-92. 1979.

[McCabe 1976]

McCabe, T., A Software Complexity Measure, IEEE Trans. Software Engineering, vol. 2, pp. 308-320. Diciembre 1976.

[Sparks y Kapczynski 1999]

Sparks, Steve; Kapczynski, Kara. The art of sizing projects. ITWorld.com. Diciembre 1999.

[Pressman 1997]

Pressman, Roger, "Ingeniería de Software: un Enfoque Práctico". 5ta Edición, 1997.

[Pressman 2001]

Pressman, Roger S. 2001. "Ingeniería de Software: un enfoque práctico". Quinta Edición. s.l. : McGraw-Hill, 2001.

[Humphrey 2001]

Humphrey, W. S. Introducción al Proceso Software Personal. Addison Wesley & Pearson Education. S. A., Madrid, 2001.

[Spinellis 2005]

- Spinellis, Diomidis. Software engineering glossary, version control, part I. IEEE Software, 22(5):107, September/October 2005. [En línea] [Citado el 28 de noviembre 2007] Disponible en: <http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=1504673>

- Spinellis, Diomidis. Software engineering glossary, version control, part 2. IEEE Software, 22(6):c2–c3, November/December 2005. [En línea] [Citado el 28 de noviembre 2007] Disponible en: <http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=1524926>

[Humphrey 2000]

Humphrey, W. S. Introduction to the Team Software Process. Addison Wesley, New York, 2000.

[Jacobson 2000]

Jacobson, Ivar; BOOCH, Grady; RUMBAUGH, James. El Proceso Unificado de Desarrollo de Software. Pearson Addison-Wesley. 2000.

[Jones 1986]

Jones, C. Programming Productivity, Mc Graw Hill, 1986.

[Beck 2005]

Beck, Kent; Andres Cynthia. Getting Started with XP: Toe Dipping, Racing Dives, and Cannonballs. Kent Beck and Cynthia Andres. All rights reserved. Año 2005.

[Craig 1999]

Larman, Craig. 1999. UML y Patrones. Mexico : s.n., 1999.

[García 2005]

García Pérez, Ana María. Una nueva métrica para tamaño en ayuda a la gestión de la calidad del software. Palacio de las Convenciones, La Habana, Cuba, 2do Taller de Calidad en la Informática y las Comunicaciones. Informática 2005.

[García 2006]

García Pérez, Ana María; Shelton Nadal, Ronald; Díaz Guerra, Taidy; de la Torre Oliva, Suny; Consuegra Rodríguez, Dialys Aliaska. Automatización de la gestión de la calidad de una organización de software partiendo de la gestión de configuración. Memorias del Evento UCIENCIA. pp 3-4. 2006

[Delgado 2006]

Delgado Martínez, Ramsés. Herramienta de apoyo a la gestión de configuración. Propuesta arquitectónica. Universidad de las Ciencias Informáticas UCI. Dirección de Calidad. Cuba. pp 29. 12 de noviembre de 2007.

[Jalón 1999]

García de Jalón, Javier; Ignacio Rodríguez, José; Mingo, Iñigo; Imaz, Aitor; Brazales, Alfonso; Larzabal, Alberto; Calleja, Jesús; García, Jon. "Aprenda Java como si estuviera en primero". Universidad de Navarra. 140 páginas. 1999.

[Kealey y Mussbacher 2006]

Kealey, Jason; Mussbacher, Gunter. StatSVN: Statistics for SVN Repositories Based on the Open Source Project StatCVS. [En línea] [Citado el: 8 febrero 2008] Disponible en: http://www.softwareengineering.ca/statsvn/CSI5140_StatSVN.pdf

[Borland 2007]

Borland Starteam. A Complete Software Change & Configuration Management (SCM) Tool. [En línea] [Citado el: 27 de noviembre 2007] Disponible en: <http://www.borland.com/us/products/starteam/index.html>

[Becerril 1998]

Becerril C. Francisco. Java a su alcance. Ed. Mc Graw Hill. 1998.

[Lockhart 1998]

The PostgreSQL Administrator's Guide , The Administrator's Guide , Editado por Thomas Lockhart, 1998-10-01, The PostgreSQL Global Development Group.

[Amat 2006]

Amat Salas, Joan. La Contabilidad de Gestión y los indicadores de control [En línea] [Citado el 12 de noviembre de 2007] Disponible en: <http://capacitacionencostos.blogia.com/2006/092004-la-contabilidad-de-gestion-y-los-indicadores-de-control.php>

[Mercado PC 2007]

Mercado PC 2007. BORLAND ACT.STARTEAM 2005 ENTERPRISE ADVANTAGE [En línea] [Citado el 12 de noviembre de 2007] Disponible en:

<http://www.mercadopc.com/logic/servlet/item?iditem=29961>

[IBM 2006]

IBM 2006. Fidelity National Financial obtiene mejoras en el desarrollo y en la reutilización de software con la Plataforma de Desarrollo de Software IBM Rational y Flashline [En línea] [Citado el 12 de noviembre de 2007] Disponible en: http://www-306.ibm.com/software/mx/success/case_10.shtml

[IBM 2006]

IBM 2006. Productos, Software, Comprar Rational ClearCase. [En línea] [Citado el 12 de noviembre de 2007] Disponible en: [https://www-](https://www-112.ibm.com/software/howtobuy/buyingtools/paexpress/Express?P0=E1&part_number=D5315LL.&catalogLocale=es_ES&Locale=es_ES&country=ESP&S_TACT=none&S_CMP=none)

[112.ibm.com/software/howtobuy/buyingtools/paexpress/Express?P0=E1&part_number=D5315LL.&catalogLocale=es_ES&Locale=es_ES&country=ESP&S_TACT=none&S_CMP=none](https://www-112.ibm.com/software/howtobuy/buyingtools/paexpress/Express?P0=E1&part_number=D5315LL.&catalogLocale=es_ES&Locale=es_ES&country=ESP&S_TACT=none&S_CMP=none)

[MSDN 2007]

MSDN Online. 2007. Microsoft Visual Source Safe. Descripción del producto [En línea] [Citado el 12 de noviembre de 2007] Disponible en: [http://msdn2.microsoft.com/es-](http://msdn2.microsoft.com/es-es/library/3h0544kx(VS.80).aspx)

[es/library/3h0544kx\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/3h0544kx(VS.80).aspx)

[Codice Software 2007]

Codice Software Blog. ¡Nuevo Plastic SCM 1.5! [En línea] [Citado el 12 de noviembre de 2007]

Disponible en: <http://codicesoftware-es.blogspot.com/2007/07/nuevo-plastic-scm-15.html>

[Rodríguez 2006]

Rodríguez Figueredo, Héctor; Batista Soler, Otto. Análisis jurídico del software libre. Sus particularidades en Cuba. [En línea] [Citado el: 16 de noviembre de 2007] Disponible en:

http://www.informaticahabana.com/evento_virtual/files/SWL27.pdf

[Stallman 2006]

Stallman, Richard. Stallman: novedades en la GPLv3 [En línea] [Citado el: 16 de noviembre de 2007]
Disponible en: <http://www.exactas.org/index.php?name=News&file=article&sid=24>

[Informática 2006]

Informática, C. T. D. Arquitectura de desarrollo, 2006. [En línea] [Citado el 16 de noviembre de 2007].
Disponible en: http://tic.csic.es/documentos/Arquitectura_desarrollo_framework_csic_feb06.pdf

[PostgreSql 2007]

PostgreSQL, Official Site. Frequently Asked Questions (FAQ) for PostgreSQL. [En línea] [Citado el 19 de noviembre de 2007]. Disponible en: <http://www.postgresql.org/docs/faqs.FAQ.html#item4.4>

[ISO 9000]

ISO 9000:2000 Documentation. Quality Manual and Operational Procedures,(2000), Jack Kanholm, USD 390, American Society for Quality (ASQ), PO Box 3066, Milwaukee, WI 53201 - 3066, USA, Tel +1 414 272 8575, Fax +1 414 272 1734, E-mail: asq.asq@org [En línea] [Citado el 19 de noviembre de 2007] Disponible en: <http://qualitypress.asq.org> o <http://www.asq.org>

ANEXOS

Anexo 1. Código del servicio SVNStatBash

```
#!/bin/bash

if [ -f /$HOME/.run-svnstatbash.conf ]
then
  . /$HOME/.run-svnstatbash.conf
else
  echo "-----"
  echo "run-svnstatbash.conf no existe"
  echo "Creando el nuevo fichero de configuracion..."
  echo "Atencion!!!"

# - Realice las asignaciones de los parametros sin espacios. Ejemplo
SVN_REPO=http://servidor/carpeta

# - No deje ninguno de los parametros sin valor, si sucede, el comando volvera a ejecutarse.

# - Si se equivoca escribiendo algun parametro, deje el proximo en blanco, pulse Enter y el comando
volvera a ejecutarse."

  echo "-----"
  echo "Complete los datos solicitados."
  echo "-----"

function validar()
{
  if [ -z $1 ]
  then
    echo "Error: No se aceptan parametros sin valor. El comando volvera ahora a
ejecutarse. Presione cualquier tecla para continuar..."
  read
  exec $0
}
```

```
        fi
    }

echo "Escriba la direccion del repositorio SVN. Ejemplo: https://10.128.1.157/sgip"

read -p "SVN_REPO=" SVN_REPO

validar $SVN_REPO

echo "-----"

echo "Autenticacion SVN"

#NOTA IMPORTANTE: Se recomienda utilizar un usuario que tenga:

1- Permiso de solo lectura ya que las credenciales suministradas pueden ser vistas por otros usuarios
no autorizados a realizar modificaciones.

2- Acceso a leer todo el repositorio."

read -p "SVN_USER=" SVN_USER

validar $SVN_USER

read -p "SVN_PASS=" SVN_PASS

validar $SVN_PASS

echo "-----"

echo "Direccion de la carpeta donde desea realizar la copia de trabajo (working copy del repositorio).
Ejemplo: $HOME/Proyecto"

read -p "SVN_CO_DIR=" SVN_CO_DIR

validar $SVN_CO_DIR

echo "-----"

echo "Nombre de la carpeta raiz de la copia de trabajo.

Ejemplo: Si en SVN_REPO puso https://10.128.1.157/sgip/src el nombre seria src. En caso de ser
https://10.128.1.157/sgip seria sgip"

read -p "SVN_ROOT_NODE_NAME=" SVN_ROOT_NODE_NAME

validar $SVN_ROOT_NODE_NAME

echo "-----"

echo "Direccion de la carpeta donde se encuentra el ejecutable del StatSVN. Ejemplo
```

```
$HOME/statsvn-0.4.0"

read -p "STAT_SVN_DIR=" STAT_SVN_DIR

validar $STAT_SVN_DIR

echo "-----"

echo "Direccion de la carpeta donde se encuentra el ejecutable del SieStatSVN. Ejemplo
$HOME/siestatsvn-0.1.0"

read -p "SIESTAT_SVN_DIR=" SIESTAT_SVN_DIR

validar $SIESTAT_SVN_DIR

echo "-----"

echo "Direccion de la carpeta donde desea generar la suite HTML con los reportes. Ejemplo:
$HOME/Reportes"

read -p "STAT_SVN_REPORT_DIR=" STAT_SVN_REPORT_DIR

validar $STAT_SVN_REPORT_DIR

echo "-----"

echo "Titulo que se muestra al usuario cuando se genera la suite HTML. Generalmente el nombre del
proyecto."

read -p "REPORT_TITLE=" REPORT_TITLE

validar $REPORT_TITLE

echo "-----"

echo "Direccion del fichero de notas. Ejemplo $HOME/noteFile.txt"

read -p "NOTES_FILE_DIR=" NOTES_FILE_DIR

validar $NOTES_FILE_DIR
cat >/$HOME/.run-svnstatbash.conf <<EOF

#-----
#run-svnstatbash.conf - Archivo de configuracion del SIEStatSVN (Servicio de Integracion de
Estadisticas de repositorios Subversion)

#Atencion!!!

# - Realice las asignaciones de las variables sin espacios. Ejemplo
SVN_REPO=http://servidor/carpeta

# - No deje ninguna de las variables sin valor.
```

```
#-----  
#Direccion del repositorio SVN. Ejemplo: https://10.128.1.157/sgip  
SVN_REPO=$SVN_REPO  
#Autenticacion SVN  
#NOTA IMPORTANTE: Se recomienda utilizar un usuario que tenga:  
#1- Permiso de solo lectura ya que las credenciales suministradas pueden ser vistas por otros usuarios no autorizados a realizar modificaciones.  
#2- Acceso a leer todo el repositorio.  
SVN_USER=$SVN_USER  
SVN_PASS=$SVN_PASS  
#Direccion de la carpeta donde desea realizar la copia de trabajo (working copy del repositorio).  
Ejemplo: $HOME/Proyecto  
SVN_CO_DIR=$SVN_CO_DIR  
#Nombre de la carpeta raiz de la copia de trabajo.  
#Ejemplo: Si en SVN_REPO puso https://10.128.1.157/sgip/src el nombre seria src. En caso de ser https://10.128.1.157/#sgip seria sgip  
SVN_ROOT_NODE_NAME=$SVN_ROOT_NODE_NAME  
#Direccion de la carpeta donde se encuentra el ejecutable del StatSVN. Ejemplo $HOME/statsvn-0.4.0  
STAT_SVN_DIR=$STAT_SVN_DIR  
#Direccion de la carpeta donde se encuentra el ejecutable del SieStatSVN. Ejemplo $HOME/siestatsvn-0.1.0  
SIESTAT_SVN_DIR=$SIESTAT_SVN_DIR  
#Direccion de la carpeta donde desea generar la suite HTML con los reportes. Ejemplo: $HOME/Reportes  
STAT_SVN_REPORT_DIR=$STAT_SVN_REPORT_DIR  
#Titulo que se muestra al usuario cuando se genera la suite HTML. Generalmente el nombre del proyecto.  
REPORT_TITLE=$REPORT_TITLE
```

```
#Direccion del fichero de notas.

NOTES_FILE_DIR=$NOTES_FILE_DIR

EOF

echo "-----"

echo "El fichero run-svnstatbash.conf ha sido creado con los parametros especificados en $HOME
como archivo oculto. Presione CTRL+H para ver los archivos ocultos en dicho directorio. Si al ejecutar
este comando una vez creado el fichero se genera algun error, verifique que los parametros
suministrados esten correctos, realice los cambios pertinentes, guarde e intente de nuevo. Si desea
encuestar otro repositorio modifique los parametros correspondientes en este mismo fichero."

exit 1

fi

if [ -d $SVN_CO_DIR ]

then

cd $SVN_CO_DIR

else

mkdir -p $SVN_CO_DIR

cd $SVN_CO_DIR

fi

svn co --username "${SVN_USER}" --password "${SVN_PASS}" "${SVN_REPO}"

cd "$SVN_CO_DIR/$SVN_ROOT_NODE_NAME"

svn log -v --xml > "logfile.log"

cd $STAT_SVN_REPORT_DIR

rm -rf *

java -jar "$STAT_SVN_DIR/statsvn.jar" -verbose -title "$REPORT_TITLE" -notes
"$NOTES_FILE_DIR" "$SVN_CO_DIR/$SVN_ROOT_NODE_NAME/logfile.log"
"$SVN_CO_DIR/$SVN_ROOT_NODE_NAME"

java -jar "$SIESTAT_SVN_DIR/siestatsvn.jar" "$SVN_REPO"
```

Anexo 2. Instalación y configuración de la BD del SGIP sobre Ubuntu 8.04.

Dirigirse a Sistema/Administración/Gestor de paquetes Synaptic.

- En el menú Editar-> Marcar paquetes por tarea. Seleccionar PostgreSQL Database
- Instalar pgAdmin3: Cliente para conectarse al servidor Postgre.

Desde la línea de comandos de la consola y superusuario root abrimos el archivo postgresql.conf con el comando: `gedit /etc/postgresql/8.3/main/postgresql.conf`

Buscamos la cadena "listen_addresses" le quitamos el comentario y le ponemos el valor "*" Quedaría así:

```
listen_addresses = '*'
```

Guardamos y cerramos el fichero.

Desde la consola permaneciendo como superusuario root abrimos el siguiente fichero: `gedit /etc/postgresql/8.3/main/pg_hba.conf`

Asegurarse de poner este detalle que se muestra a continuación:

```
# TYPE DATABASE USER CIDR-ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all ident sameuser
# IPv4 local connections:
host all all 127.0.0.1/32 trust
```

Guardamos, cerramos y reiniciamos el servidor de PostgreSQL desde la consola con el comando: `/etc/init.d/postgresql-8.3 restart`

Ahora podemos ir a Aplicaciones / Herramientas del Sistema y abrir el pgAdmin III el que nos permitirá conectarnos a nuestro gestor de base de datos PostgreSQL. Al abrirse el mismo lo primero que tenemos que hacer es realizar una nueva conexión con el servidor. En este caso con el local que tenemos instalado en nuestra PC. Para esto marcamos Servidores y en el menú Archivo seleccionamos Añadir Servidor. Suministramos los siguientes valores para la ventana:

Dirección: localhost

Descripción: La que uno quiera

Nombre de usuario: postgres

Aceptamos y ya nos hemos conectado al servidor.

Si quiere seguir conociendo otros detalles sobre la seguridad debe cerrar el pgAdmin para poder ilustrar lo que viene a continuación.

Cuestiones de seguridad (Opcional)

Pero la manera en que hemos puesto este acceso es público (**trust**), si queremos establecer que el acceso al gestor de base de datos se realice por contraseña debemos ir a Roles de Login y haciendo clic derecho del usuario postgres en Propiedades podemos establecerle una contraseña para el mismo. Aceptamos y vamos de nuevo al archivo pg_hba.conf escribiendo en la consola:

```
gedit /etc/postgresql/8.3/main/pg_hba.conf
```

y establecemos los siguientes cambios:

```
# Database administrative login by UNIX sockets
local all postgres ident sameuser -> password

# TYPE DATABASE USER CIDR-ADDRESS METHOD

# "local" is for Unix domain socket connections only
local all all ident sameuser
# IPv4 local connections:
host all all 127.0.0.1/32 trust -> md5
#Máscara de subred para permitir conexiones dentro del lab solamente
host all all 10.128.1.0/24 (trust o md5)
#Máscara de subred para permitir conexiones desde cualquier IP
#host all all 0.0.0.0 0.0.0.0 md5
# IPv6 local connections:
host all all ::1/128 md5
```

El método md5 se utiliza cuando queremos proteger algo con contraseña. Caso contrario cuando queremos poner la conexión de dominio público donde bastaría con cambiar md5 por trust.

Después de haber hecho todos los pasos anteriores debemos reiniciar el servidor de PostgreSQL con el comando: `/etc/init.d/postgresql-8.3 restart`

Ahora vamos nuevamente a Aplicaciones / Herramientas del Sistema y abrimos el pgAdmin III. Esta vez para conectarnos con el gestor (clic derecho Conectar) nos pedirá la contraseña que le pusimos anteriormente a postgres. Se la proporcionamos y listo, ya tenemos configurado satisfactoriamente el pgAdmin III con nuestro gestor de base de datos relacional PostgreSQL y esta vez con seguridad incluida.

En el ícono Bases de Datos hacemos clic derecho y escogemos la opción Nueva Base de Datos...Especificamos el nombre BDSGIP-UCI y pulsamos el botón Aceptar. Marcamos el ícono de la nueva base de datos creada y en el menú Herramientas seleccionamos Herramienta para consultas. Hecho esto se abre una ventana donde correremos el script SQL que creará las tablas y reglas de la base de datos. Para abrir el script seleccionamos el ícono Abrir archivo en la barra inferior al menú. Antes de correr el script debemos asegurarnos que el mismo se ejecutará para la base de datos BDSGIP-UCI, esto se hace comprobando que el combobox de esta ventana tenga el siguiente texto asignado: "BDSGIP-UCI on [postgres@localhost:5432](#)". (Ver figura 2.1)

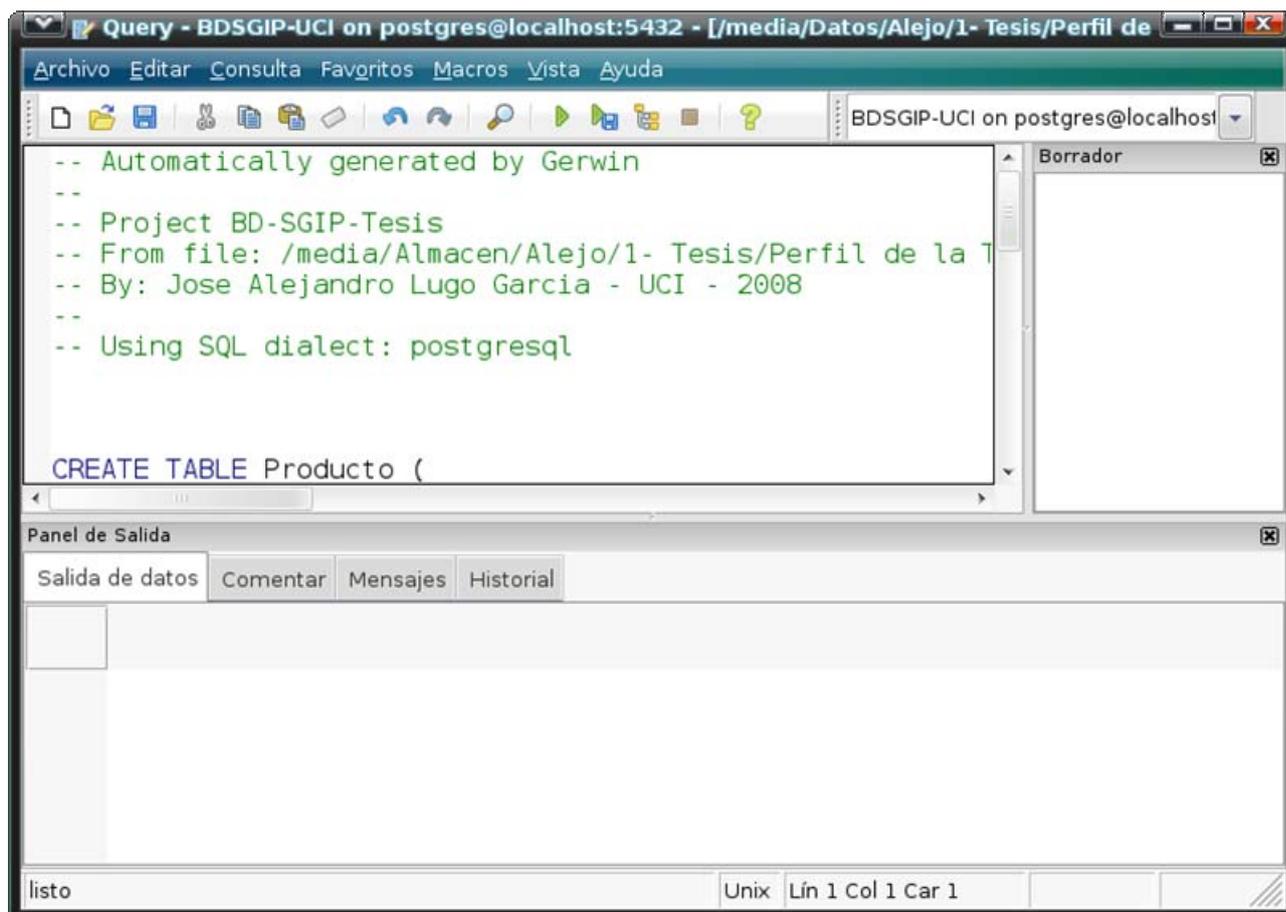


Figura 2.1 Ejecutar script para crear la base de datos.

Una vez verificado esto, procederemos a ejecutarlo en el menú Consulta/Ejecutar. El gestor nos advierte del resultado de la operación. Si todo resultó sin problemas, la base de datos ya estará lista para ser utilizada por los clientes SGIP y el servicio SVNStatBash.

Anexo 3. Gráficos de la suite HTML generada por SVNStatBash. (Proyecto SGIP)

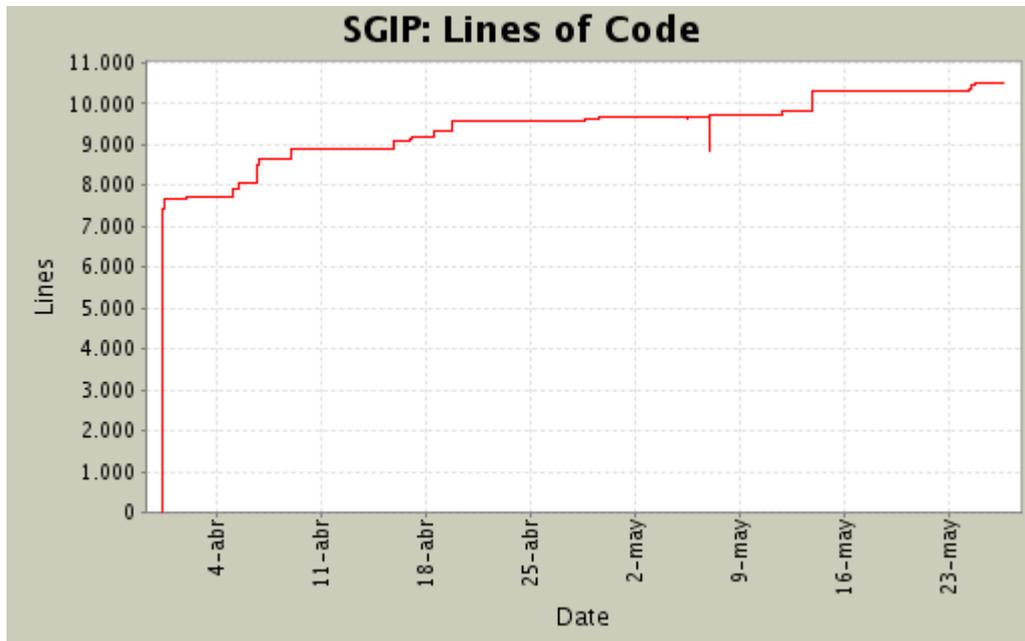


Figura 3.1 Líneas totales de código producidas en el proyecto por fecha.

File Types			
Type	Files	LOC	LOC per file
*.java	40 (21.9%)	9055 (94.3%)	226.3
*.xml	17 (9.3%)	409 (4.3%)	24.0
*.properties	2 (1.1%)	66 (0.7%)	33.0
*.prefs	1 (0.5%)	4 (0.0%)	4.0
Others	3 (1.6%)	72 (0.7%)	24.0
Non-Code Files	120 (65.6%)	0 (0.0%)	0.0
Totals	183 (100.0%)	9606 (100.0%)	52.4

Figura 3.2 Datos sobre los tipos de ficheros existentes en el proyecto.

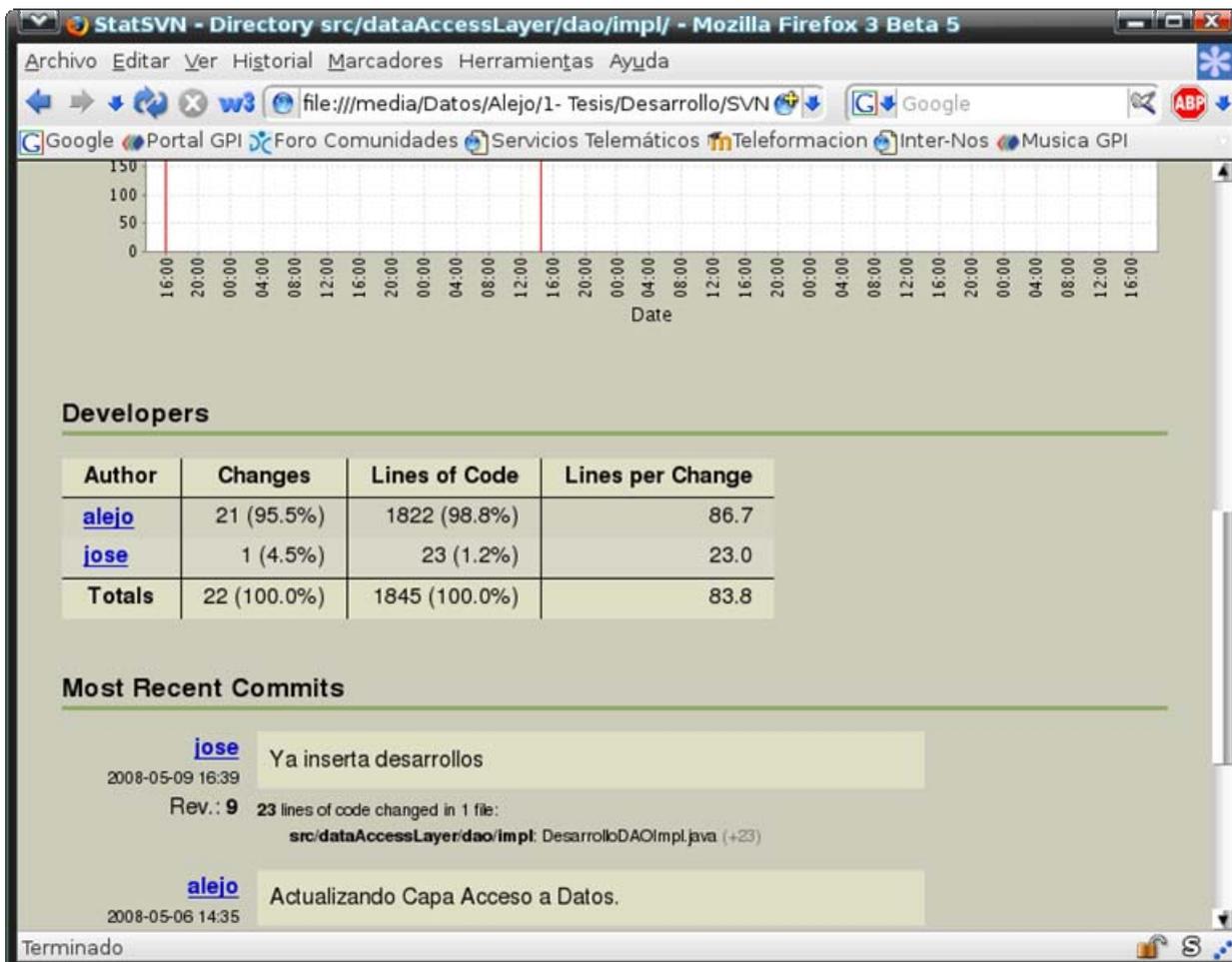


Figura 3.3 Aportes de cada programador en un nodo del repositorio.

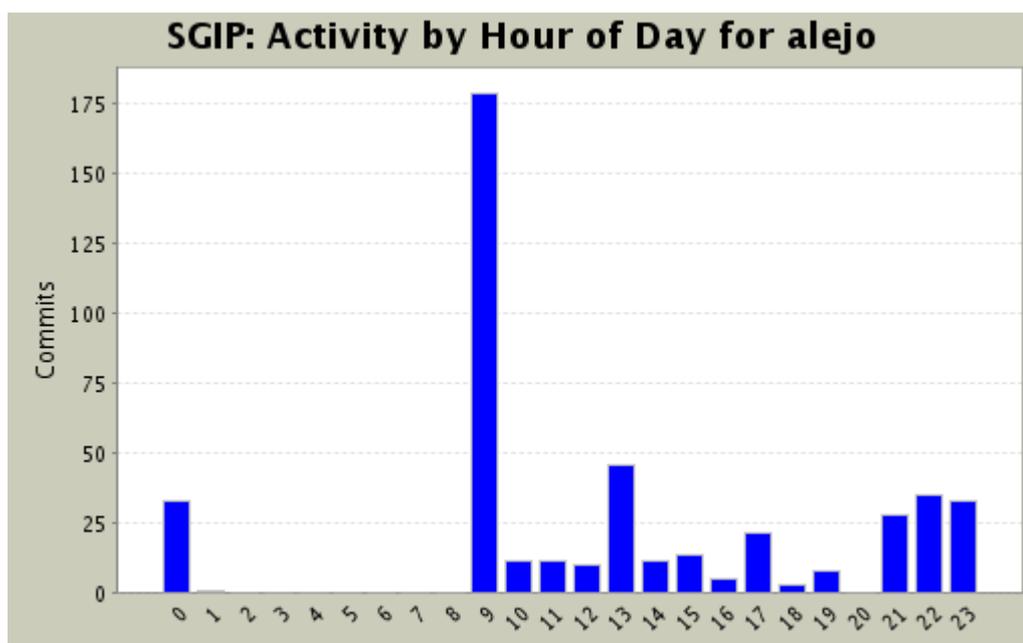


Figura 3.4 Productividad del programador "alejo" en el día.

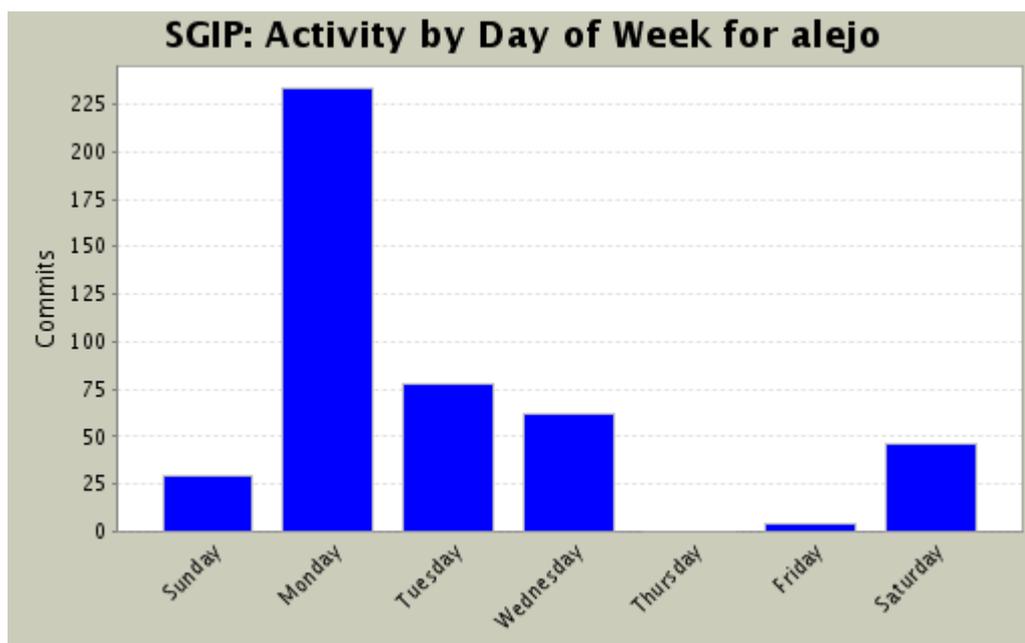


Figura 3.5 Productividad del programador “alejo” en la semana.

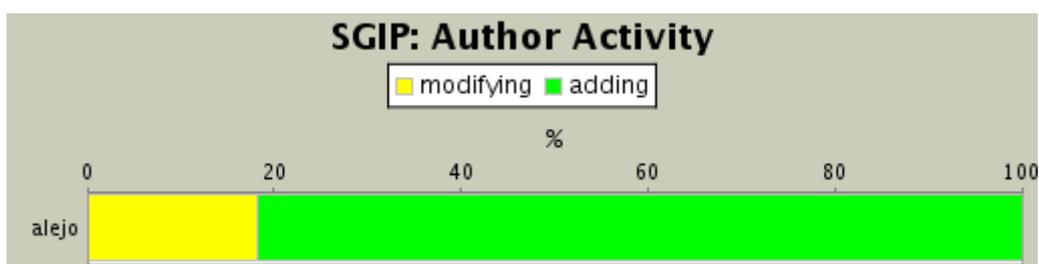


Figura 3.6 Operaciones en las que participa el usuario.

Anexo 4. Fichero de descripción del proyecto.

El archivo txt expuesto a continuación es un fragmento de HTML y será la descripción del proyecto en la suite HTML generada por SVNStatBash.

```
<p>

Estas son las estadísticas de desarrollo, creadas por la herramienta StatSVN, para el proyecto

<a href="https://10.128.1.157/sgip" target="_blank">SGIP</a> - Sistema de Gestion de Indicadores
de Productividad.

</p>

<p>

<strong>Autor:</strong>

Jose Alejandro Lugo Garcia. (<a href="mailto:jalugo@estudiantes.uci.cu">Contacto</a>).

</p>

<p>

<strong>Organizacion:</strong>

Universidad de las Ciencias Informaticas, Cuba. (<a href="http://www.uci.cu" target="_blank">Sitio
Web</a>).

</p>

<p>

<strong>Nota:</strong>

Puede aprender mas sobre como generar estos reportes

visitando el siguiente <a href="http://svn.statsvn.org/statsvnwiki/index.php/User_Manual"
target="_blank">Manual</a>.

</p>
```

Figura 4.1 Fichero notesFile.txt



Figura 4.2 Vista de cómo queda la descripción en la suite HTML.