

**SEPROSO.**  
**Plan de medidas.**

*Francisco Javier Delgado del Hoyo*

*Yuri Torres de la Sierra*

*Rubén Martínez García*

*Abel Lozoya de Diego*

Diciembre, 2008

# Revisiones del documento

## Historial de revisiones del documento

VERSIÓN	FECHA	DESCRIPCIÓN	AUTOR
0.1	16/11/08	Recopilación de Información Inicial.	Francisco
0.5	18/11/08	Asociación del documento al Plan de Proyecto.	Rubén

# Indice

<b>Revisiones del documento</b>	<b>i</b>
<b>1 Introducción.</b>	<b>1</b>
1.1 Propósito. . . . .	1
1.2 Ámbito. . . . .	1
1.3 Definiciones. . . . .	1
1.4 Referencias. . . . .	1
1.5 Visión general. . . . .	2
<b>2 Gestión de objetivos y subobjetivos.</b>	<b>3</b>
<b>3 Métricas.</b>	<b>4</b>
3.1 Métricas. . . . .	5
<b>4 Métricas primitivas.</b>	<b>7</b>
<b>A Plantilla de registro de esfuerzo por actividad</b>	<b>9</b>
<b>B Herramientas</b>	<b>10</b>

# Indice de tablas

3.1	Métricas WMC.	5
3.2	Métricas Duración.	6
3.3	Métricas DIT.	6
4.1	Métricas SLOC.	7
4.2	Métricas SLOCc.	8
4.3	Métricas Número de Casos de Uso.	8

# Capítulo 1

## Introducción.

### 1.1 Propósito.

El propósito de este Plan de Medición de Software es detallar cuales son las métricas (formas de medir el software) primitivas que serán recogidas, así como las métricas asociadas a la revisión de calidad del proyecto. Además se especifica una colección de objetivos del proyecto que permite monitorizar el progreso del mismo.

### 1.2 Ámbito.

Este plan define un programa simple de métricas, incluyendo los objetivos de la medición, las métricas utilizadas, y las métricas primitivas necesarias para realizar el seguimiento del progreso.

El plan está basado en los requisitos software especificados en el documento SRS.

### 1.3 Definiciones.

Véase el Glosario.

### 1.4 Referencias.

1. Artefactos: SRS, SSS, Plan de Desarrollo Software, Glosario. Grupo III.
2. UPEDU Measurement Plan: <http://www.upedu.org/upedu/>

## 1.5 Visión general.

A lo largo del resto de este documento especificaremos todos los detalles del Plan de Medición de Software, dividido en las siguientes partes o temas:

- **Objetivos de la medición:** objetivos del programa de medición en este proyecto, en cuanto a logros, mejoras y calidad.
- **Métricas:** medidas a obtener a intervalos regulares de tiempo del proyecto para lograr los objetivos.
- **Métricas primitivas:** medidas recogidas automáticamente o manualmente para calcular las métricas anteriores.

## Capítulo 2

# Gestión de objetivos y subobjetivos.

Hay dos aproximaciones para garantizar buena calidad en el producto final. La primera consiste en garantizar el proceso de desarrollo (secuencia de actividades) y segunda en garantizar la calidad del producto obtenido (artefactos del proceso, incluyendo software, documentos y modelos). Mediante la medición pretendemos asegurar en ambos casos una calidad suficiente.

Más concretamente, los objetivos del programa de medición para este proyecto son los siguientes:

- Evaluar la calidad del producto (en todos los artefactos).
- Asegurar un seguimiento pormenorizado del proyecto (progreso de las fases según lo estimado y estado de las actividades realizadas)
- Evaluar aspectos de los requisitos: impacto de los cambios, tareas implicadas y nivel de realización.
- Facilitar la planificación y estimación para los nuevos proyectos: comparar relativamente las distintas fases realizadas y la importancia de cada actividad en ellas.

## Capítulo 3

# Métricas.

A continuación se enumeran las métricas que serán recogidas del proyecto a intervalos regulares para conseguir los objetivos de la sección anterior. Como también vimos, las métricas se pueden dividir entre las que permiten evaluar el producto o el proyecto, para conseguir calidad en el desarrollo.

### ***PRODUCTO***

Estableceremos algunas métricas para evaluar diferentes atributos de los tres elementos fundamentales del producto: el código, los documentos y los modelos.

- **Código:** tamaño (SLOC), tamaño para control de configuraciones (SLOCc), profundidad del árbol de herencia (DIT)
- **Documento:** tamaño (número de páginas), esfuerzo (horas de tiempo del equipo para producir y cambiar), volatilidad (número de páginas cambiadas), trazabilidad (checklist en revisión)
- **Modelo:** tamaño (número de casos de uso, casos de prueba, actores, clases o paquetes, WMC)

### ***PROCESO***

Para asegurar la calidad del proceso necesitamos realizar las mediciones sobre las actividades de más bajo nivel. Mantendremos un registro de cada actividad con los valores del esfuerzo empleado y las estimaciones iniciales planificadas, lo que nos permite medir la calidad de la planificación inicial.

- **Esfuerzo:** horas de esfuerzo del equipo.



Tabla 3.1: Métricas WMC.

Nombre	WMC (Weighted Methods for Class)
Definición	Número de métodos implementados en una clase.
Objetivo	Indica el esfuerzo y tiempo necesario para desarrollar y mantener la clase. Aquella clase con más métodos es la que presenta un mayor impacto potencial en sus hijos, ya que heredan todos los métodos. Al mismo tiempo limitan su reutilización ya que son más específicas de la aplicación. Por lo tanto con esta métrica medimos la usabilidad y la reutilización de las clases.
Proceso de análisis	Se debe recoger cuando una actividad produzca un Modelo de Clases de Diseño. Se calculará con la herramienta CASE de modelado utilizada. Será añadida a la Base de Datos de Medidas del proyecto.
Responsabilidad	Equipo de desarrollo.

- **Duración:** tiempo transcurrido desde el inicio.
- **Salida:** cantidad y tamaño de los artefactos (incluidos los defectos detectados)

### 3.1 Métricas.

Las principales métricas son las siguientes:

- WMC [3.1](#).
- Duración [3.2](#).
- DIT [3.3](#).

Tabla 3.2: Métricas Duración.

Nombre	Duración.
<b>Definición</b>	Diferencia entre la fecha de finalización y la de comienzo de una actividad. Se expresa en minutos y las fechas se utilizan con una resolución de 15 minutos.
<b>Objetivo</b>	Nos indica el esfuerzo y coste empleado en la realización de una actividad. El esfuerzo se utiliza tanto en la medición del proceso como del producto.
<b>Proceso de análisis</b>	Cada trabajador rellena la misma plantilla de estimación de esfuerzo (incluida en la sección 5.1) cada vez que termina una actividad. Además es una de las entidades fundamentales del proyecto. Se creará relleno la plantilla manualmente y será entregada al Gestor de Proyecto, que las recogerá una vez por semana y que posteriormente las añadirá a la Base de Datos de Medidas del proyecto.
<b>Responsabilidad</b>	Cada trabajador encargado de una actividad.

Tabla 3.3: Métricas DIT.

Nombre	DIT (Deep Inheritance Tree).
<b>Definición</b>	Máxima longitud del camino desde una clase hasta su clase raíz en el sistema.
<b>Objetivo</b>	Nos indica el número de super-clases que afectan a la clase, midiendo el nivel de dependencia y con ello la reusabilidad, mantenimiento, portabilidad, etc...
<b>Proceso de análisis</b>	Se calcula manualmente estudiando los diagramas de clases de la herramienta, comenzando por la clase base y ascendiendo por el árbol hasta la clase raíz. Se hace cada vez que una actividad actualice el Diagrama de Clases. Se recoge para cada clase con herencia y posteriormente se añade a la Base de Datos de Medidas del proyecto.
<b>Responsabilidad</b>	Equipo de desarrollo.

## Capítulo 4

# Métricas primitivas.

Las siguientes métricas son básicas o primitivas porque se recogen y calculan manual o automáticamente para calcular otras métricas de la sección 3.3.

Tabla 4.1: Métricas SLOC.

Nombre	SLOC (Source Line of Code).
Definición	Número de líneas de código desarrolladas.
Proceso de análisis	Se recogen mediante la herramienta de desarrollo utilizada, justo al final de cada iteración, detallando el módulo o fichero específico al que pertenecen. Es un número entero que se almacena en la Base de Datos de Medidas.
Responsabilidad	Equipo de desarrollo.

Tabla 4.2: Métricas SLOCc.

<b>Nombre</b>	<b>SLOCc (Source Line of Code).</b>
<b>Definición</b>	Número de líneas de código desarrolladas para la línea base actual.
<b>Proceso de análisis</b>	Se calculan con la herramienta de desarrollo utilizada. Se recogen cada vez que la línea base de desarrollo cambia, detallando el módulo o fichero específico al que pertenecen. Es un número entero que se almacena en la Base de Datos de Medidas.
<b>Responsabilidad</b>	Equipo de desarrollo.

Tabla 4.3: Métricas Número de Casos de Uso.

<b>Nombre</b>	<b>Número de Casos de Uso.</b>
<b>Definición</b>	Número de casos de uso identificados en el modelo.
<b>Proceso de análisis</b>	Son recogidos manualmente por simple inspección visual del Diagrama de Casos de Uso. Se hace una vez por cada actividad que actualice el Modelo de Casos de Uso del proyecto. Es un número entero que se almacena en la Base de Datos de Medidas.
<b>Responsabilidad</b>	Equipo de desarrollo.

## Apéndice A

# Plantilla de registro de esfuerzo por actividad

Para evaluar la calidad del proceso de desarrollo utilizamos como métrica el esfuerzo dedicado realmente a cada una de las actividades planificadas. Para obtener esta métrica los miembros del equipo de desarrollo deben completar manualmente y con una breve descripción los siguientes conceptos:

- **Nombre:** persona que desarrolló la actividad.
- **Trabajador:** rol que desempeñaba en el proceso.
- **Actividad:** nombre o ID de la actividad que ha finalizado.
- **Artefacto:** salidas producidas de la actividad (código, documento o modelo).
- **Fecha:** fecha del informe.
- **Fecha de inicio:** día y hora de comienzo de la actividad (resolución de 15 min)
- **Fecha de finalización:** día y hora de terminación de la actividad (resolución de 15 min)
- **Duración:** horas empleadas en total.

## Apéndice B

# Herramientas

Para el cálculo de las métricas nos apoyaremos en herramientas de desarrollo software ya existentes y Open Source. Más concretamente utilizaremos:

- **Eclipse PHP (PDT)** como entorno de desarrollo para el lenguaje PHP que permite medir líneas de código automáticamente.
- **StarUML** como herramienta de modelado UML para crear diagramas UML que permiten realizar manualmente algunas mediciones por simple inspección visual.