



**UNIVERSIDAD DE CASTILLA-LA MANCHA**  
**ESCUELA SUPERIOR DE INFORMÁTICA**

**INGENIERÍA**  
**EN INFORMÁTICA**

**ANTEPROYECTO**

**DESGLOSA**

Un sistema de visualización 3D para dar soporte al Desarrollo Global de Software

Jose Domingo López López

**Mayo, 2011**

# ÍNDICE

<b>1</b>	<b>Introducción</b>	<b>1</b>
<b>2</b>	<b>Fundamentos teóricos</b>	<b>2</b>
2.1	Desarrollo Global de Software . . . . .	2
2.2	Métricas software . . . . .	2
2.3	Visualización de métricas . . . . .	3
<b>3</b>	<b>Objetivos</b>	<b>4</b>
3.1	Objetivo principal . . . . .	4
3.2	Objetivos parciales . . . . .	5
<b>4</b>	<b>Método y fases de trabajo</b>	<b>6</b>
<b>5</b>	<b>Medios que se pretenden utilizar</b>	<b>7</b>
5.1	Medios hardware . . . . .	7
5.2	Medios software . . . . .	7
	<b>Bibliografía</b>	<b>8</b>

# 1. INTRODUCCIÓN

La **globalización**, según la Real Academia Española, es la “*tendencia de los mercados y de las empresas a extenderse, alcanzando una dimensión mundial que sobrepasa las fronteras nacionales*”. Otras fuentes [1] buscan una definición más explícita y definen este concepto como “*un proceso económico, tecnológico, social, político y cultural de interacción e integración entre personas, empresas y gobiernos de todo el mundo*”. No cabe duda de que este proceso o tendencia ha desencadenado una gran cantidad de cambios no sólo en la forma de trabajo tradicional de la industria y el comercio, si no también en el modo en el que los productos son diseñados, construidos, probados y distribuidos [15].

El fenómeno de la globalización también está afectando a la industria del software. A consecuencia de ello, el software está siendo desarrollado por personas que no se encuentran co-localizadas en las mismas ubicaciones, es decir, se encuentran distribuidas o dispersas geográficamente. Esta dispersión geográfica puede limitarse a un área cerrada y próxima (por ejemplo, distintos edificios en una misma localidad) o expandirse por todo el mundo evadiendo cualquier frontera. Además, los integrantes de un equipo de desarrollo pueden pertenecer a una misma organización o a organizaciones diferentes [10]. Cuando varios equipos de desarrollo, que trabajan en un mismo proyecto, pertenecen a distintas organizaciones y además están dispersos geográficamente por todo el mundo, se habla de **Desarrollo Global de Software** (en inglés, Global Software Development o GSD) [8].

El GSD introduce una serie de características y complejidades que afectan, tanto positiva como negativamente, al desarrollo y gestión de proyectos. Por un lado, este paradigma permite a las organizaciones el abstraerse de la distancia geográfica y minimizar costes [11], pero es susceptible de numerosos problemas [14] como las diferencias lingüísticas, culturales, horarias, etcétera [12].

Con el fin de proporcionar una solución que permita a las organizaciones conocer el estado de los proyectos de desarrollo global en los que se encuentran involucradas, y simplificar las tareas de gestión de las personas a cargo de los mismos, se propone este **Proyecto Fin de Carrera**, en adelante PFC, que consistirá en construir una herramienta enmarcada en un proyecto de investigación que será utilizada por **Indra Sistemas, S.A.**.

## 2. FUNDAMENTOS TEÓRICOS

Para la realización de este PFC es necesario definir y estudiar tres conceptos fundamentales:

- Qué es el Desarrollo Global de Software (GSD).
- Qué son las métricas y su uso en el ciclo de vida del software.
- Principios para la representación visual de métricas.

### 2.1. Desarrollo Global de Software

Sahay Sundeep [17] define el GSD como *“una forma coordinada de desarrollo de software realizado en lugares geográficamente distantes atravesando fronteras nacionales, implicando interacción síncrona (en tiempo real) y asíncrona”*. Otros autores [18] lo definen como *“una forma de desarrollar software utilizando equipos situados en múltiples localizaciones geográficas. Esos equipos pueden pertenecer a la misma organización o, por el contrario, pueden existir colaboraciones o outsourcing entre varias organizaciones”*. Tomando como base ambas definiciones, se puede generalizar y definir el GSD como *un conjunto de actividades que permite desarrollar software mediante equipos geográficamente distantes, que pueden pertenecer a diferentes organizaciones, y que comparten flujos de trabajo y los entregables generados en cada una de las fases que los conforman*. Esta última parte cobra fuerza si se reflexiona acerca de que el ciclo de desarrollo del software se compone de una serie de fases relacionadas entre sí (análisis, diseño, implementación, pruebas, despliegue, mantenimiento y soporte) y que un producto software no es únicamente el código fuente que se ejecuta en una o varias máquinas, si no que está formado por todos los artefactos que se han ido generando a lo largo de su ciclo de vida (diagramas, modelos, documentación, etcétera).

Por tanto, tal y como se ha definido el GSD, se infiere que el desarrollo global afecta tanto al proceso de desarrollo del software como a la gestión de proyectos, y es susceptible de problemas tales como las diferencias horarias, culturales o lingüísticas, coordinación, motivación, etcétera [7]. Por el contrario, también ofrece una serie de ventajas como la disponibilidad de recursos especializados y la reducción de costes, que prometen, entre otros, beneficios económicos [8].

### 2.2. Métricas software

Una **métrica** es *“un sistema de medida que incluye la entidad que se está midiendo, una unidad de medida y un valor”* [5] y se define para uno o más atributos comunes para un conjunto de entidades. Un atributo es una propiedad, física o abstracta, que se puede medir.

Las **métricas software** son medidas cuantificables que se utilizan para medir diferentes características o atributos de un producto software o un proceso de desarrollo software [19]. Según

[16] y [4], el uso de métricas en el software es una buena práctica para conocer qué ocurre durante el proceso de desarrollo y mantenimiento del software, predecir lo que puede ocurrir y, por consiguiente, tomar las mejores decisiones cuando se considere oportuno.

## 2.3. Visualización de métricas

Los sistemas de visualización deben proporcionar vistas intuitivas, efectivas y eficaces que hagan del análisis de datos una tarea rápida y precisa. Para lograrlo, se hace uso de una cualidad del sistema visual humano denominado *proceso pre-atentivo* [6]. Los procesos pre-atentivos son procesos que “saltan a la vista”, siendo detectados inmediatamente y evitando que el usuario deba centrar su atención en una determinada región de una imagen para captar la presencia o ausencia de una característica determinada [2].

Existen cuatro categorías básicas de características que se procesan de forma pre-atentiva [20]: color, forma, movimiento y localización espacial. De cada una de estas características se puede obtener un gran conjunto de subcaracterísticas como curvatura, tamaño, orientación, longitud, etcétera.

Para representar una **métrica software** visualmente es necesario qué entidad se quiere escenificar por medio de un modelo gráfico y cuáles son sus atributos más relevantes. De este modo, es posible establecer un conjunto de asociaciones entre los atributos de la entidad y las subcaracterísticas del modelo gráfico. Este concepto recibe el nombre de *vista polimétrica* [13] y permite que los modelos gráficos adopten una determinada forma, orientación o color en función de los valores de las métricas, beneficiándose así de los procesos pre-atentivos.

## 3. OBJETIVOS

A continuación, y bajo el subtítulo de “objetivo principal”, se describirá la meta final que se persigue en la elaboración de este PFC, que consiste en obtener un producto software. Un producto software no se constituye únicamente de código fuente ejecutable, sino que también se compone de la documentación, diagramas y modelos -entre otros- que se han generado en todas y cada una de las fases que conforman todo el proceso de desarrollo del software. Ello acarrea la adquisición de una serie de competencias y la compleción de un conjunto de tareas que serán expuestas bajo el título de “objetivos parciales”.

### 3.1. Objetivo principal

El objetivo principal del PFC consiste en elaborar una **herramienta** que permita la visualización de algunas **métricas software** utilizadas en el contexto empresarial bajo el paradigma del **Desarrollo Global de Software**. Para ello, será necesario definir un conjunto de **metáforas visuales** capaces de escenificar las características más relevantes de un producto o un proceso software. Para lograr una herramienta intuitiva y utilizable, se consideran, *a priori*, dos niveles visuales de abstracción: *nivel de fábrica* y *nivel de proyecto*.

- **Nivel de fábrica.** El GSD implica la colaboración entre varias organizaciones y cada organización puede estar compuesta por una o más fábricas de software. A su vez, una fábrica puede estar desarrollando varios proyectos, o parte de ellos, de forma simultánea. En este nivel se podrá acceder a la información relativa a una fábrica determinada o a un conjunto de fábricas.
- **Nivel de proyecto.** Un proyecto puede estar desarrollándose en varias fábricas pertenecientes a distintas organizaciones. Desde este nivel se podrá visualizar la información relativa a los proyectos software.

Para permitir que las distintas organizaciones y fábricas involucradas en un conjunto común de proyectos puedan utilizar el sistema de visualización y manejar la información globalizada, la herramienta se diseñará como una **aplicación web**, que deberá ser accesible por medio de un navegador web convencional.

Además, a fin de desarrollar un sistema extensible y que satisfaga los requisitos de visualización más exigentes, se desarrollará un **motor gráfico** basado alguna librería de gráficos 3D, que será integrado en la aplicación web. El motor gráfico implementará un conjunto de **metáforas de visualización** que representarán de un modo visual los datos contenidos en los atributos de distintas métricas software.

Para conseguir este objetivo se deben llevar a cabo los objetivos parciales que se abordan en las siguiente subsección.

## 3.2. Objetivos parciales

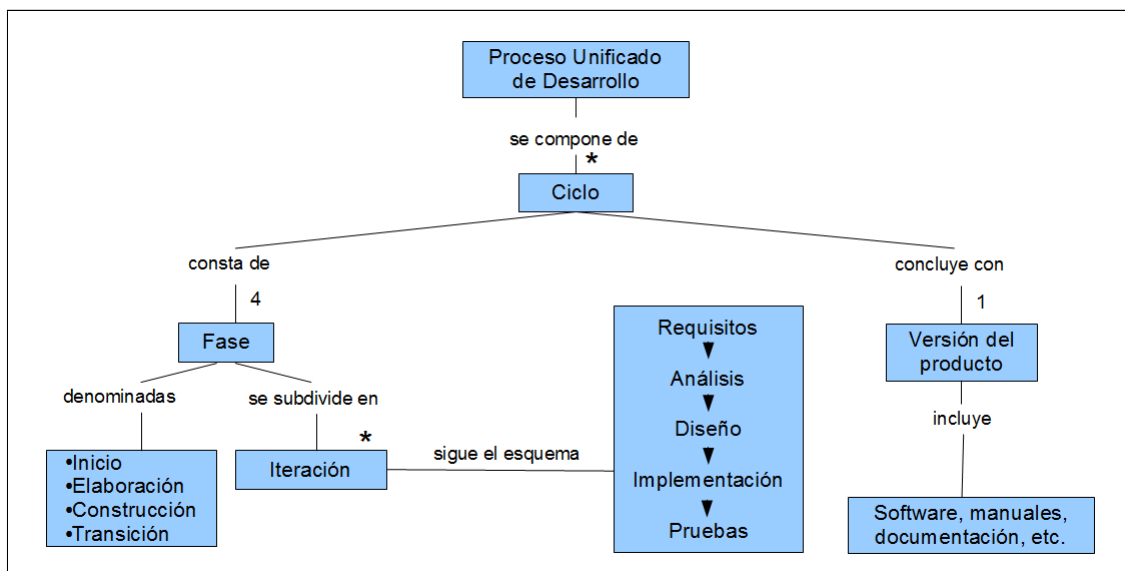
La consecución del objetivo principal acarrea la adquisición y compleción de las siguientes competencias y tareas:

- Seleccionar un conjunto de **métricas software** utilizadas en el contexto empresarial bajo el paradigma del GSD.
- Definir **metáforas de visualización** para cada uno de los niveles de abstracción que se desean escenificar.
- Analizar las distintas técnicas de visualización empleadas para escenificar métricas.
- Establecer un mecanismo de asociación entre las subcaracterísticas de las metáforas de visualización y los atributos de las métricas software.
- Analizar el mundo real, en el contexto del GSD, para lograr un diseño conceptual que abstraiga la información relevante. Este punto implica el diseño de una base de datos para la gestión de la lógica de negocio utilizada por la herramienta.
- Investigar sobre librerías para la generación de gráficos 3D y técnicas de visualización de datos empleando gráficos 3D.
- Definir una arquitectura estable y extensible de un motor gráfico que permita la representación de modelos para escenificar metáforas de visualización, empleando buenas prácticas de programación así como patrones de diseño.
- Desarrollar un **motor gráfico** para la visualización de datos empleando modelos en 3D.
- Analizar tecnologías y *frameworks* para el desarrollo de aplicaciones Web y tecnologías para la generación de gráficos y modelos de visualización en la Web.
- Definir una arquitectura estable y extensible de una aplicación web, utilizando patrones de diseño.
- Desarrollar una **aplicación web** para la visualización de métricas software empleando técnicas basadas en gráficos en dos y tres dimensiones.
- Integrar del motor gráfico desarrollado en la aplicación web.
- Desarrollar las **pruebas unitarias** del software.
- Elaborar la documentación técnica asociada al proyecto.

## 4. MÉTODO Y FASES DE TRABAJO

Dada la naturaleza del proyecto, en el que se irán definiendo requisitos y funcionalidades a lo largo de todo su ciclo de vida, se ha optado por utilizar una metodología de desarrollo de software que permita adaptarse a este caso de estudio. Por estas razones, se ha seleccionado el **Proceso Unificado de Desarrollo** (en adelante PUD) como metodología de trabajo.

El PUD [9] es una evolución del Proceso Unificado de Rational, que define un “conjunto de actividades necesarias para transformar los requisitos de usuario en un sistema software”. Es un marco genérico que puede especializarse para una gran variedad de sistemas de software cualesquiera que sean el área de aplicación o el tamaño del proyecto [3]. Sus caracteriza debido a que está dirigido por casos de uso, centrado en la arquitectura, y es iterativo e incremental (Fig. 4.1).



**Figura 4.1:** Mapa conceptual del PUD

Para llevar esta metodología a la práctica se establecerán una serie de ciclos, dentro de los cuales se definirá un conjunto de requisitos funcionales a los que habrá que dar solución. Los requisitos funcionales negociados dentro de cada ciclo conforman los hitos que deberán alcanzarse en dicho ciclo. Los ciclos serán divididos a su vez en iteraciones en las que se abordarán un subconjunto de requisitos funcionales. Para concluir una iteración será necesario planificar reuniones con los responsables del proyecto de **Indra Sistemas S.A.**, que deberán validar y verificar que se han alcanzado los hitos establecidos para dicha iteración. Si la validación y verificación es de carácter positivo, se obtendrá una nueva versión del software. Este proceso se aplicará hasta obtener la versión final del software.



## 5. MEDIOS QUE SE PRETENDEN UTILIZAR

Los medios y recursos necesarios para el desarrollo y despliegue del PFC se categorizan en *medios hardware* y *medios software*.

### 5.1. Medios hardware

Tanto para el desarrollo como el despliegue del sistema bastará con un ordenador sobremesa o portátil. Dicho ordenador deberá estar dotado de una conexión a Internet y actuar como servidor web para permitir que los usuarios puedan acceder a la aplicación por medio de un navegador web convencional.

La tarjeta gráfica del sistema requiere especial mención. Ésta no tiene por qué poseer una potencia extraordinaria, pero es de vital importancia que disponga de sus controladores gráficos debidamente instalados para una correcta ejecución del motor gráfico.

### 5.2. Medios software

Tanto la aplicación web como el motor gráfico serán desarrollados en tecnologías basadas en Java y, debido a su naturaleza *multiplataforma*, se espera que no existan restricciones en cuanto al sistema operativo que el ordenador debe ejecutar. Sin embargo, éste debe tener debidamente instalado el entorno de ejecución Java (JRE o *Java Runtime Environment*)

Por otro lado, son necesarias las siguientes herramientas:

- JDK 1.4+ (*Java Development Kit*).
- Un entorno integrado de desarrollo (IDE o *Integrated Development Environment*). Se utilizará **Eclipse EE Helios**.
- Un navegador web con JavaScript habilitado para que se pueda establecer una comunicación entre la aplicación web y el motor gráfico, que estará incrustado en un applet, y poder beneficiarse de las características de la tecnología AJAX (*Asynchronous JavaScript And XML*).
- Un sistema para la gestión y construcción de proyectos. Se hará uso de **Maven 2**.
- Un servidor web capaz de desplegar aplicaciones web con estructura WAR (*Web-Archive*). **Apache Tomcat 7** es la solución más factible gracias a su licencia, eficiencia y rendimiento.
- Un sistema gestor de bases de datos. Se empleará **MySQL Community Server 5.x**.
- **Visual Paradigm** para modelado y diseño de diagramas UML.
- $\text{\LaTeX}$  para la elaboración de documentación técnica asociada al proyecto.
- GIMP e Inkscape para el diseño y edición de figuras, imágenes y gráficos.

# BIBLIOGRAFÍA

- [1] Croucher, S.: *Globalization and belonging: the politics of identity in a changing world*. pág. 10, 2004.
- [2] Ducasse, S., S. Denier, F. Balmas, A. Bergel, J. Laval, K. Mordal-Manet y F. Bellingard: *Visualization of Practices and Metrics*, March 2010.
- [3] Espinoza, J. M. O.: *El proceso unificado de desarrollo de software*.
- [4] Fenton, N. E. y S. L. Pfleeger: *Software metrics: A rigorous approach*. 1997.
- [5] Geisler, E.: *The Metrics of Science and Technology: Evaluation of research, development and innovation*. Quorum Books, 2000.
- [6] Healey, C. G., K. S. Booth y J. T. Ennis: *Visualizing real-time multivariate data using preattentive processing*. ACM TOMACS, 5, 1995.
- [7] Herbsleb, J. D. y A. Mockus: *An empirical study of global software development: Distance and speed*. IEEE Software, 2001.
- [8] Herbsleb, J. D. y D. Moitra: *Global software development*. IEEE Software, March/April:16–20, 2001.
- [9] Jacobson, I., G. Booch y J. Rumbaugh: *El proceso unificado de desarrollo de software*. Addison Wesley, 2000.
- [10] Katzy, B., R. Evaristo y I. Zigurs: *Knowledge management in virtual projects: A research agenda*. 1999.
- [11] Kobitzsch, W., D. Rombach y R. L. Feldmann: *Outsourcing in india*. IEEE Software, 18(2):78–86, March/April 2001.
- [12] Krishna, S., S. Sahay y G. Walsham: *Managing cross-cultural issues in global software outsourcing*. Commun. ACM, págs. 62–66, 2004.
- [13] Lanza, M.: *Polymetric views: A lightweight visual approach to reverse engineering*. IEEE Transactions on Software Engineering, 29, September 2003.
- [14] Layman, L., L. Williams, D. Damian y H. Bures: *Essential communication practices for extreme programming in a global software development team*. Information and Software Technology, 48(9):781–794, 2006.
- [15] O'Hara-Devereaux, M. y R. Johansen: *Global Work: Bridging Distance, Culture, and Time*. Jossey-Bass Publishers, 1994.
- [16] Pfleeger, S. L.: *Assessing software measurement*. IEEE Software, págs. 25–26, March/April 1997.

- [17] Sahay, S., B. Nicholson y S. Krishna: *Global IT Outsourcing: Software Development across Borders*. Cambridge University Press, November 2003.
- [18] Sangwan, R., M. Bass, N. Mullick, D. J. Paulish y J. Kazmeier: *Global Software Development Handbook*. Auerbach Publications, 2007.
- [19] Shaik, A., C. Reddy y A. Damodaran: *Statistical analysis for object oriented design software security metrics*. International Journal of Engineering Science and Technology, 2:1136–1142, 2010.
- [20] Ware, C.: *Information Visualization. Perception for design*. Morgan Kaufmann, February 2004.



## **Escuela Superior de Informática**

D. Jose Domingo López López, con D.N.I. nº 71219116-F, alumno de la Escuela Superior de Informática.

### **EXPONE:**

— Que se encuentra matriculado del PFC.

### **SOLICITA:**

— Que la Comisión Académica evalúe el anteproyecto de fin de carrera que acompaña, titulado *DESGLOSA: Un sistema de visualización 3D para dar soporte al Desarrollo Global de Software*.

Ciudad Real, a 6 de mayo de 2011

El alumno,

VºBº El director,

Fdo.:

Fdo.:

**SR. DIRECTOR DE LA ESCUELA SUPERIOR DE INFORMÁTICA**