



**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**

## DESGLOSA

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

© 2011 Jose Domingo López López (Alumno del PFC)

M<sup>a</sup> Ángeles Moraga de la Rubia (Director del PFC)

En virtud del Artículo 7 del Real Decreto Legislativo 1/1996, de 12 de abril por el que se aprueba el Texto Refundido de la Ley de Propiedad Intelectual, modificado por la Ley 23/2006 de 7 de julio, este PFC se considera una obra en colaboración entre las diferentes partes. Por tanto la propiedad intelectual de este PFC será compartida con iguales porcentajes entre el alumno y el director del PFC.

Este documento ha sido compuesto con L<sup>A</sup>T<sub>E</sub>X y sus imágenes generadas con Inkscape y GIMP.

# ÍNDICE

<b>1</b>	<b>Introducción</b>	<b>1</b>
<b>2</b>	<b>Fundamentos teóricos</b>	<b>1</b>
2.1	Desarrollo Global de Software . . . . .	1
2.2	Visualización de métricas . . . . .	2
<b>3</b>	<b>Objetivos</b>	<b>3</b>
3.1	Objetivo principal . . . . .	3
3.2	Objetivos secundarios . . . . .	4
<b>4</b>	<b>Método y fases de trabajo</b>	<b>5</b>
4.1	Dirigido por casos de uso . . . . .	5
4.2	Centrado en la arquitectura . . . . .	5
4.3	Iterativo e incremental . . . . .	5
<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>
	<b>Anexos</b>	<b>9</b>
<b>A</b>	<b>Solicitud de evaluación</b>	<b>9</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 [10]. Por lo tanto, la industria del software y las empresas que dependen del software también se ven afectados por este movimiento y sus consecuencias.

Debido al fenómeno de la globalización, 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 distribución 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 [8]. Como consecuencia de estas dos características surge un nuevo paradigma denominado **Desarrollo Global de Software** (en inglés, Global Software Development o GSD).

Este nuevo paradigma introduce una serie de características y complejidades que puede afectar tanto al desarrollo como a la gestión de proyectos. Es de vital importancia que las empresas manejen métricas e indicadores que les permitan comprobar si determinados factores están siendo llevados a cabo con normalidad o que reflejen si existe algún tipo de riesgo o anomalía. Para ello se propone este Proyecto Fin de Carrera, en adelante PFC, que tratará de dar una solución para la definición de dichas métricas y dotarlas de una representación visual intuitiva, eficaz y eficiente.

## 2. FUNDAMENTOS TEÓRICOS

### 2.1. Desarrollo Global de Software

Sahay Sundeep (2003) 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*”. Por otro lado, Parastoo Mohagheghi (2004) lo define como el “*modo de desarrollo de software que permite que los proyectos, productos o sistemas atraviesen las fronteras de las compañías y países, por medio de la coordinación de equipos distribuidos con distintas localizaciones geográficas y husos horarios*”. 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 entregables*. 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 lo conforman 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 distribuido 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 [5]. Pero también proporciona una serie de ventajas muy importantes que prometen, entre otros, beneficios económicos [6]. A continuación se enumeran algunos de ellos:

- **Reducción de costes:** gracias a la subcontratación de empresas o equipos de desarrollo ubicados en localidades o países en los que la mano de obra es más económica.
- **Tiempos de lanzamiento al mercado más cortos:** ya que se pueden subcontratar varios equipos de desarrollo para desempeñar distintas actividades de un modo paralelo o concurrente.
- **Mejora de la calidad:** se debe a que determinadas funcionalidades pueden ser desarrolladas por expertos ajenos a la empresa que desarrolla el software.
- **Escalabilidad:** hace posible afrontar la alza o baja, tanto temporal como permanente, de demanda de productos

## 2.2. Visualización de métricas

Los sistemas de visualización deben proporcionar mecanismos que permitan que el análisis de datos sean tareas sean rápidas y precisas. Para lograrlo, se hace uso de una cualidad del sistema visual humano denominado *proceso pre-atentivo* [4]. 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 [12]: 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.

Por otro lado, gracias a las *vistas polimétricas* [9] es posible representar una métrica de un modo visual. El método consiste en establecer un conjunto de asociaciones entre cada una de las dimensiones de la métrica con cada una de las subcaracterísticas de un modelo gráfico. De este modo, los modelos tomarán una determinada forma, orientación o color en función de los valores de las métricas, permitiendo así hacer uso de los beneficios 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, si no 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 secundarios”.

### 3.1. Objetivo principal

El objetivo principal del PFC consiste en elaborar una **aplicación web** que permita la visualización de indicadores y métricas utilizadas en el contexto empresarial bajo el paradigma del **Desarrollo Global de Software**, escenificadas por un conjunto de metáforas previamente definidas. La visualización se llevará a cabo mediante tecnologías basadas en gráficos 2D y 3D, utilizándose en cada momento la que se considere oportuna en función de las necesidades de representación de cada métrica.

A fin de desarrollar un sistema extensible y que satisfaga los requisitos de visualización más exigentes, se desarrollará un **motor gráfico 3D** basado en *Java* y *OpenGL*, que será integrado en la aplicación Web mediante un applet. Asimismo, el usuario deberá ser capaz de interactuar con dicho applet por medio de un navegador Web convencional.

El motor gráfico implementará un conjunto de **metáforas de visualización** y permitirá que sean utilizadas para representar las distintas dimensiones de los datos o información que se quiera representar visualmente. Por otro lado, se dejarán abiertas determinadas guías para permitir la incorporación de nuevos modelos y metáforas de un modo fácil, viable, rápido y poco costoso por parte del desarrollador.

La visualización se podrá hacer en base a dos niveles de abstracción:

- **Nivel de fábrica**, en el que se podrá acceder a información relativa a una fábrica determinada o a un conjunto de fábricas. Una fábrica puede estar desarrollando varios proyectos.
- **Nivel de proyecto**, desde el que se podrá visualizar la información de un proyecto. Un proyecto puede estar desarrollándose en varias fábricas.

## 3.2. Objetivos secundarios

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

- Revisión sistemática de las distintas técnicas de visualización empleadas para escenificar indicadores y métricas que representan información.
- Análisis del mundo real, en el contexto del GSD, para lograr un diseño conceptual que abstraiga la información relevante.
- Revisión sistemática de librerías para la generación de gráficos 3D y técnicas de visualización de datos empleando gráficos 3D.
- Definición de 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.
- Desarrollo de un motor gráfico para la visualización de datos empleando modelos en 3D.
- Revisión sistemática de tecnologías y *frameworks* para desarrollo de aplicaciones Web y tecnologías para la generación de gráficos y modelos de visualización en la Web.
- Definición de una arquitectura estable y extensible de una aplicación Web, utilizando patrones de diseño.
- Diseño de una base de datos para la gestión de la lógica de negocio utilizada por el sistema a implementar.
- Desarrollo de una aplicación Web para la visualización de información empleando técnicas basadas en gráficos en tres dimensiones.
- Integración del motor gráfico basado en OpenGL con la aplicación Web.
- Desarrollo de pruebas unitarias para el software.
- Elaboración de 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 [7] 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 principales características se abordan en las siguientes subsecciones.

### 4.1. Dirigido por casos de uso

Para poder desarrollar un sistema es necesario saber qué necesitan sus usuarios [3]. Un usuario puede ser un ser humano u otro sistema que interacciona con el sistema que se está desarrollando. Las necesidades de un usuario se denominan requisitos funcionales y se representan por medio de casos de uso.

Los casos de uso guían el proceso de desarrollo desde la especificación de requisitos hasta las pruebas y se utilizan para crear modelos que permitan la construcción e implementación de los mismos. Todos los casos de uso juntos constituyen el **modelo de casos de uso** [11].

### 4.2. Centrado en la arquitectura

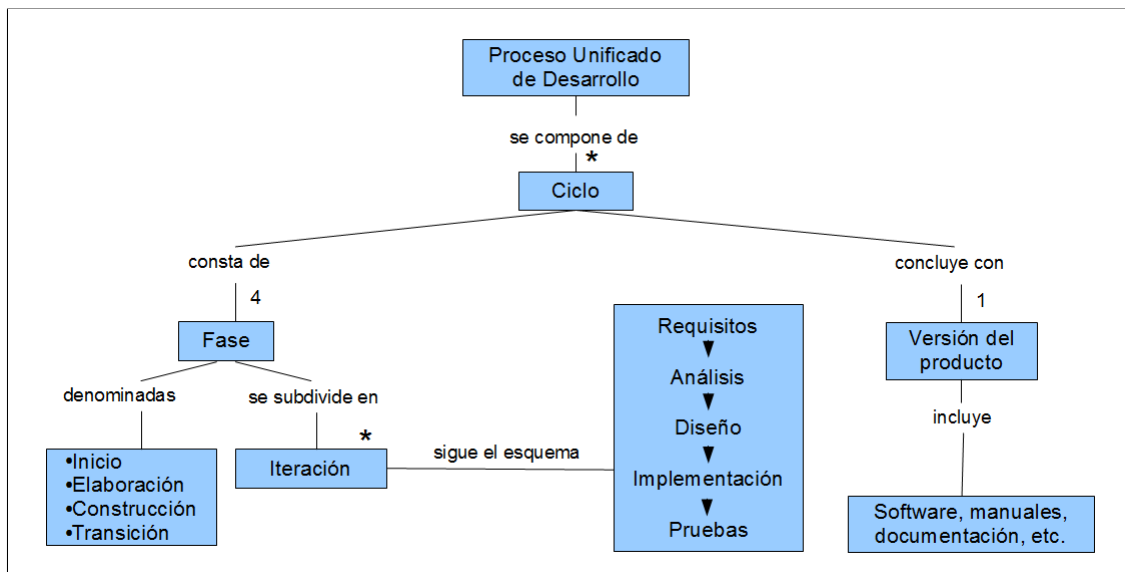
Un sistema software puede contemplarse desde varios puntos de vista. La arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema, y debe estar profundamente relacionada con los casos de uso ya que debe permitir el desarrollo de los mismos [11]. Por esta razón, la arquitectura y los casos de uso deben desarrollarse en paralelo.

### 4.3. Iterativo e incremental

Partiendo del dicho latino *divide et impera* (en castellano, *divide y vencerás*), el ciclo de vida del proyecto se divide en ciclos. Cada ciclo consta de cuatro fases (inicio, elaboración, construcción y transición) que a su vez se dividen en iteraciones (Fig. 4.1).

Cada iteración da resultado a una nueva versión del proyecto, aborda un conjunto de casos de uso que amplían la funcionalidad global del producto desarrollado y trata los riesgos más importantes. Si la iteración cumple sus objetivos se continúa con la próxima, en caso contrario se vuelve





**Figura 4.1:** Mapa conceptual del PUD

a las primeras etapas de la misma y se toman otras decisiones que permitan cumplir sus objetivos. Las iteraciones deben ser seleccionadas y ejecutarse de forma controlada y siguiendo un flujo de trabajo [11]. Este flujo suele constar de las típicas etapas de captación de requisitos, análisis, diseño, implementación, pruebas y despliegue.

## 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] Healey, C. G., K. S. Booth y J. T. Ennis: *Visualizing real-time multivariate data using preattentive processing*. ACM TOMACS, 5, 1995.
- [5] Herbsleb, J. D. y A. Mockus: *An empirical study of global software development: Distance and speed*. IEEE Software, 2001.
- [6] Herbsleb, J. D. y D. Moitra: *Global software development*. IEEE Software, March/April:16–20, 2001.
- [7] Jacobson, I., G. Booch y J. Rumbaugh: *El proceso unificado de desarrollo de software*. Addison Wesley, 2000.
- [8] Katzy, B., R. Evaristo y I. Zigers: *Knowledge management in virtual projects: A research agenda*. 1999.
- [9] Lanza, M.: *Polymetric views: A lightweight visual approach to reverse engineering*. IEEE Transactions on Software Engineering, 29, September 2003.
- [10] O'Hara-Devereaux, M. y R. Johansen: *Global Work: Bridging Distance, Culture, and Time*. Jossey-Bass Publishers, 1994.
- [11] Torossi, G.: *El Proceso Unificado de Desarrollo de Software*.
- [12] Ware, C.: *Information Visualization. Perception for design*. Morgan Kaufmann, February 2004.

# ANEXOS

## A. SOLICITUD DE EVALUACIÓN



### 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 4 de mayo de 2011

El alumno,

VºBº El director,

VºBº El tutor académico

Fdo.:

Fdo.:

Fdo.:

SR. DIRECTOR DE LA ESCUELA SUPERIOR DE INFORMÁTICA