



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

INGENIERÍA
EN INFORMÁTICA

ANTEPROYECTO

desGLosa
Un sistema de visualización 3D para **D**esarrollo **G**lobal de **S**oftware

Jose Domingo López López

Octubre, 2010

desGLosa

Un sistema de visualización 3D para Desarrollo Global de Software

© 2010 Jose Domingo López López

M^a Ángeles Moraga de la Rubia (Tutor académico)

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^AT_EX y sus imágenes generadas con GIMP.

ÍNDICE

1	Introducción	1
2	Objetivos	3
2.1	Objetivo principal	3
2.2	Objetivos secundarios	3
3	Método y fases de trabajo	4
4	Medios que se pretenden utilizar	6
4.1	Tecnologías para desarrollo y diseño web	6
4.2	Tecnologías para visualización	7
	Bibliografía	9
	Anexos	10
A	Solicitud de evaluación	10
B	Contrato de Propiedad Intelectual	11

1. INTRODUCCIÓN

Años atrás, las organizaciones y empresas se preocupaban de la calidad de los procesos que se seguían para desarrollar software basándose en la premisa de que “*con un proceso de calidad se obtiene un producto de calidad*” [6], pero dados los problemas que han surgido en sistemas ya implantados y en producción, y las pérdidas que éstos han ocasionado, ha sido necesario actualizar la percepción que se tiene acerca de la calidad del software. Por ello, las organizaciones comenzaron a preocuparse por la calidad del producto software desde el punto de vista de los clientes y usuarios finales, abarcando un mayor número de *stakeholders*, con el fin de aumentar el grado de satisfacción y confianza por parte de los mismos.

Es por esto por lo que se trató de asegurar la funcionalidad y usabilidad de los sistemas profundizando en área de *testing*, pero se han quedado sin cubrir aspectos de seguridad y mantenibilidad. Y es que un producto software no es únicamente el código que se ejecuta sobre una máquina, si no que lo son también los artefactos que se han ido produciendo a lo largo de todo el ciclo de vida: modelos, diagramas, documentos, etc.

Para paliar algunos de los problemas expuestos anteriormente, se comenzaron a utilizar dos estándares:

- **ISO/IEC 9126** [4], que define el modelo de calidad de un producto, incluyendo las características, subcaracterísticas y métricas (de calidad interna, calidad externa y calidad en uso) que han de tenerse en cuenta, pero sin proporcionar una metodología para su evaluación. Las características que propone son funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.
- **ISO/IEC 14598** [3], que proporciona guías y requisitos para el proceso de evaluación, cubriendo así los aspectos de los que carece la norma ISO/IEC 9126.

Pero entre ambos estándares existen distintas incompatibilidades e inconsistencias -tales como diferencias en el vocabulario, términos y conceptos, la necesidad de añadir recomendaciones, metodologías, guías y nuevas necesidades para especificar las dimensiones de la calidad del software, entre otros- que motivan la creación de la familia **ISO/IEC 25000**, denominada SQuaRe (Software product Quality Requirements and Evaluation).

La norma ISO/IEC 25000 establece criterios para la especificación de requisitos de calidad de productos software y su evaluación, un modelo de calidad y métricas para los distintos atributos de dicho modelo. Nótese que no es necesario alcanzar una calidad perfecta o absoluta, sino la necesaria y suficiente para cada contexto de uso y en función de las necesidades de los *stakeholders*, que son todas las entidades involucradas en el producto: clientes, usuarios finales, factorías y empresas de desarrollo, etcétera.

La aparición de esta norma, cuyo objetivo es reemplazar a las normas ISO/IEC 9126 e ISO/IEC 14598, junto con las nuevas necesidades que surgen en las empresas para controlar y evaluar la calidad de los desarrollos informáticos, motivan la aparición del proyecto **MEDUSAS** (Mejora y Evaluación del Diseño, Usabilidad, Seguridad y Mantenibilidad del Software). MEDUSAS (Fig. 1.1) es un entorno metodológico e instrumental basado en la ISO/IEC 25000 para satisfacer dichas

necesidades, y evaluar la calidad del software teniendo en cuenta todos y cada uno de sus artefactos: código fuente, diseño, diagramas, modelos, etc.

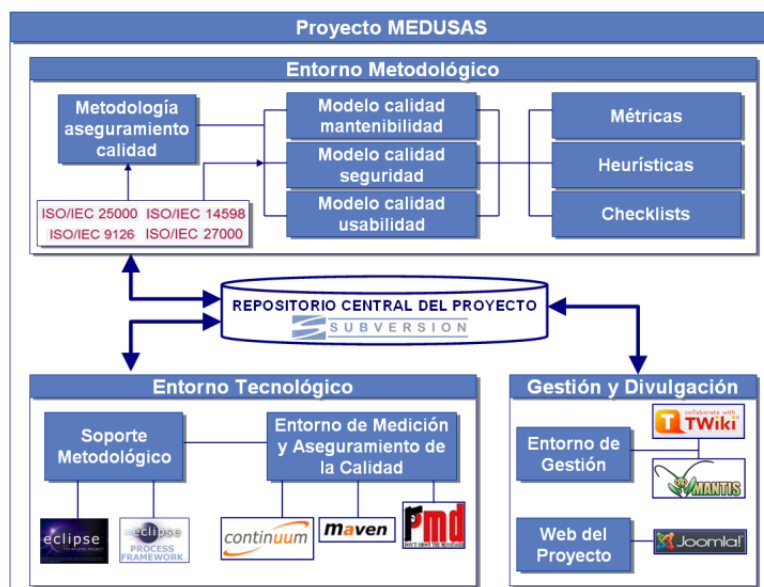


Figura 1.1: Arquitectura del proyecto MEDUSAS

Aquí es donde surge la motivación del Proyecto Fin de Carrera (en adelante PFC), que consistirá en realizar un cuadro de mando para la visualización de la calidad del software, integrado con el entorno MEDUSAS. Como resultado de la integración, se dispondrá de un sistema completo que permitirá automatizar la adquisición, presentación y análisis [2] de las métricas que definen la calidad del producto software.

2. OBJETIVOS

Los objetivos que se esperan lograr con la realización de este PFC se han dividido en dos categorías. En la primera de ellas, como objetivo principal, se introducirá el producto final que se espera obtener y, por último, como objetivos secundarios, se expondrán los objetivos que se irán alcanzando en las distintas fases de elaboración del PFC.

2.1. Objetivo principal

El objetivo principal del PFC consiste en elaborar una **aplicación web** para la visualización de los datos obtenidos como resultado de las operaciones de medición y evaluación de la calidad del producto software del entorno MEDUSAS, que está basado en la nueva familia de normas ISO/IEC 25000. Para ello, se desarrollará un proyecto Java que será capaz de representar gráficamente la información procedente de los resultados de evaluación de la calidad del producto software.

Con el paso del tiempo, sobre todo si el proceso de evaluación se encuentra automatizado o semi-automatizado, la cantidad de información generada puede ser muy grande. Por esta razón, será necesario presentarla en distintos niveles de detalle o abstracción. Estos niveles se denominan operativo, táctico y estratégico, y permitirán al usuario o evaluador realizar un análisis eficiente e intuitivo de la calidad del producto software, ya sea de una versión concreta del producto software, o de la evolución que se ha producido a lo largo de distintas versiones, ya sea mediante historiales o tendencias.

La aplicación web será desarrollada utilizando tecnologías actuales empleadas en desarrollo y diseño web (Ver Sección 4) con el fin de obtener una RIA (Rich Internet Application) [7]. Deberá gozar de una alta usabilidad y navegabilidad, tratando de maximizar la experiencia de usuario.

Una RIA se fundamenta en una arquitectura cliente/servidor asíncrona, segura y escalable, disminuyendo la carga de trabajo en el servidor y siendo accesible desde cualquier navegador web. Así, estará disponible para cualquier persona desde cualquier rincón del planeta, siempre y cuando posea una conexión a Internet, tratando de maximizar la experiencia de usuario y producir un incremento en la productividad.

2.2. Objetivos secundarios

Para la consecución del objetivo principal será necesario llevar a cabo los siguientes objetivos secundarios:

- Realizar un estudio referente a las normas de calidad del software existentes.
- Investigar sobre las principales formas de visualización de calidad del software.
- Estudiar los principales entornos y medios actuales para la visualización de la calidad del software, así como las librerías gráficas que permiten su implementación.

- Investigar sobre los principales *frameworks* y tecnologías empleados en desarrollo y diseño web.
- Implementar un entorno que permita visualizar la calidad de un producto software de acuerdo a las características de calidad impuestas por la ISO/IEC 25000.
- Integrar dicho entorno de visualización con un entorno real de medición de la calidad del producto software, proporcionado por Alarcos Quality Center.

3. 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 genérica 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 [5] 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. Sus principales características son las siguientes:

- **Dirigido por casos de uso.** Para poder desarrollar un sistema es necesario saber qué necesitan sus usuarios. 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 los modelos que permitan la construcción e implementación de los mismos. Todos los casos de uso juntos constituyen el **modelo de casos de uso**.
- **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. Por esta razón, la arquitectura y los casos de uso deben desarrollarse en paralelo.
- **Iterativo e incremental.** Partiendo del dicho latino *divide et impera* (en castellano, *divide y vencerás*), el ciclo de vida del proyecto (Fig. 3.1) se divide en ciclos. Cada ciclo consta de cuatro fases que a su vez se dividen en iteraciones. Las fases de un ciclo son las siguientes:
 - **Inicio o Factibilidad.** Se obtiene un modelo de casos de uso simplificado, se identifican riesgos potenciales y se estima el proyecto de manera aproximada.
 - **Elaboración.** Se mejora el modelo de casos de uso y se diseña la arquitectura del sistema. A continuación, se desarrollan los casos de uso más críticos que se identificaron en la fase de inicio y se obtiene una *línea base* de la arquitectura para planificar las actividades y estimar los recursos necesarios para terminar el proyecto.
 - **Construcción.** Se crea el producto en base a las dos fases anteriores.

- **Transición.** Implica la corrección de errores y *bugs*, así como el mantenimiento del sistema.

En cierto modo, una iteración se corresponde con un mini-proyecto que resulta en un *incremento*. En cada iteración se 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 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. Este flujo de trabajo consta de cinco etapas:

- **Requisitos.** El equipo de desarrollo y los clientes establecen comunicaciones por diferentes medios para especificar qué es lo que el usuario espera del sistema que se va a desarrollar.
- **Análisis.** Se identifican y especifican los casos de uso relevantes.
- **Diseño.** Se crea un diseño utilizando la arquitectura seleccionada como guía para tratar de dotar al sistema de las funcionalidades representadas por los casos de uso identificados en la etapa de análisis.
- **Implementación.** Se implementan las decisiones tomadas en la etapa de diseño, con el fin de construir las funcionalidades que satisfacen los casos de uso identificados en la etapa de análisis.
- **Pruebas.** Se verifica que los cambios realizados en el sistema satisfacen los casos de uso.

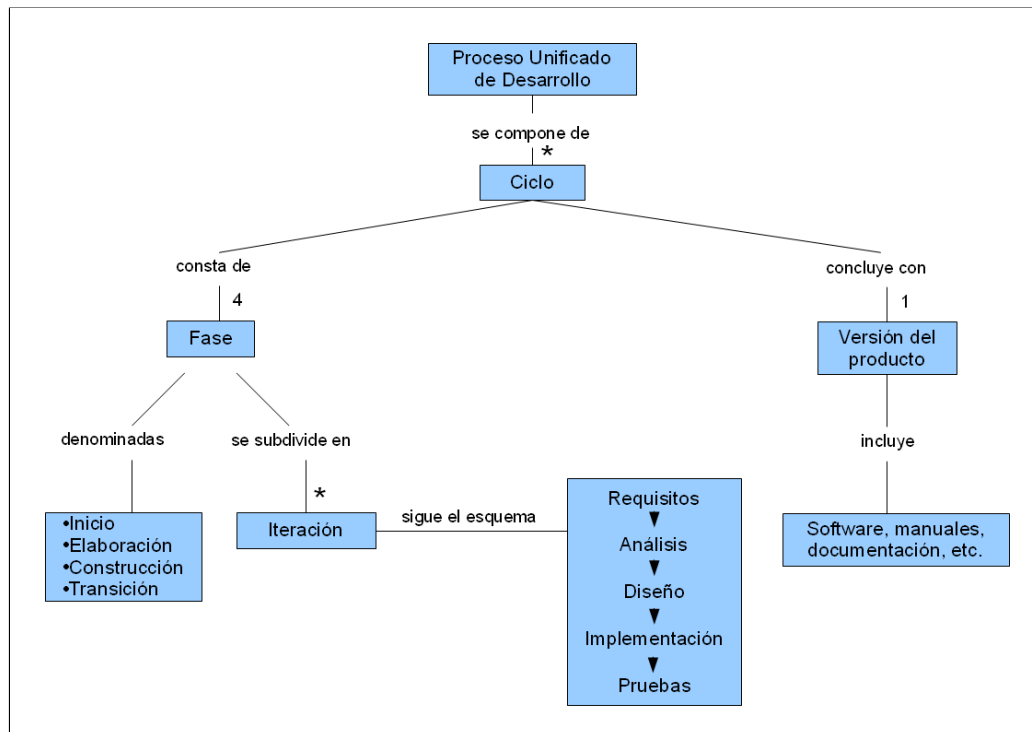


Figura 3.1: Mapa conceptual del PUD

4. MEDIOS QUE SE PRETENDEN UTILIZAR

Como ya se ha explicado anteriormente (Ver Sección 2.1), se elaborará una aplicación web utilizando tecnologías actuales empleadas en desarrollo y diseño web con el fin de obtener una RIA.

Una aplicación web tiene dos contextos bien diferenciados: el lado del cliente (client-side) y el lado del servidor (server-side). Para poder dotar una aplicación web de las características de una aplicación de escritorio es necesario trabajar desde el lado del cliente. Para ello, se dispone de tecnologías como CSS, Ajax y JavaScript que nos permiten cambiar dinámicamente el contenido que el usuario recibe en su navegador web a través del DOM¹.

A continuación se exponen algunas de las tecnologías y librerías que han sido consideradas para elaborar el PFC, no siendo necesario utilizar todas y cada una de ellas, y dejando abierta la posibilidad de utilizar alguna que se estudie después de haber presentado el anteproyecto.

4.1. Tecnologías para desarrollo y diseño web

Para poder desarrollar la aplicación web descrita en los objetivos del PFC (Ver Sección 2.1) se han considerado tecnologías que se ejecutan tanto en el lado del cliente, como en el lado del servidor.

- **HTML 5** (HyperText Markup Language, versión 5). Es la quinta revisión de HTML. Incorpora, de forma nativa, numerosas características y funcionalidades que le permiten no depender de programas externos o *plugins* del navegador. No obstante, aún tiene ciertos problemas de compatibilidad con determinados navegadores, por lo que no se descarta el uso de HTML4.
- **DWR: Direct Web Remoting** [9]. Es una librería que permite ejecutar funciones Java desde JavaScript y viceversa. Además, es posible integrarlo con tecnologías como jQuery, Struts e Hibernate.
- **Struts2**. Es un *framework* de desarrollo web basado en el patrón MVC (Modelo-Vista-Controlador), permitiendo así reducir el acoplamiento entre las tres capas de las que se compone un sistema software.
- **jQuery** [12] y **Dojo** [8]. Son *frameworks* para JavaScript que proveen al programador de un mayor nivel de abstracción para el desarrollo de interfaces gráficas de usuario en aplicaciones web.

¹DOM, acrónimo de Document Object Model, es un API a través del cuál se puede modificar dinámicamente el contenido, estructura y estilo de los documentos HTML y XML.

4.2. Tecnologías para visualización

En este documento se denomina como tecnologías o librerías de visualización a aquellas que permiten la generación de modelos o diagramas que nos permiten representar información de un modo gráfico. A continuación se enumeran las características que debe poseer una librería de este tipo:

- Generación dinámica de diagramas.
- Posibilidad de elección entre una gran colección de modelos (diagramas de barras, sectores, kiviati, etcétera).
- Permitir la interacción por parte del usuario.
- Visualización atractiva.
- Integración fácil en la aplicación web, desarrollada en Java.
- Gratuito y, preferiblemente, libre para poder desarrollar nuevos modelos si fuera necesario.
- Una licencia conforme a nuestros requisitos y objetivos.

Algunas de las librerías que se proponen para su estudio, a fin de satisfacer los requisitos anteriormente expresados, y pudiendo añadir nuevas librerías en un futuro, son las siguientes [1]:

- **DojoX Charting.** Es una extensión para Dojo (Ver Sección 4.1) que permite la generación de gráficas y diagramas.
- **Google Visualization API [10]** (Fig. 4.1). Es una librería basada en JavaScript y que pertenece a Google. Goza de un buen soporte por parte del equipo de desarrollo y una comunidad de desarrolladores que aportan una gran cantidad documentación y nuevos modelos.

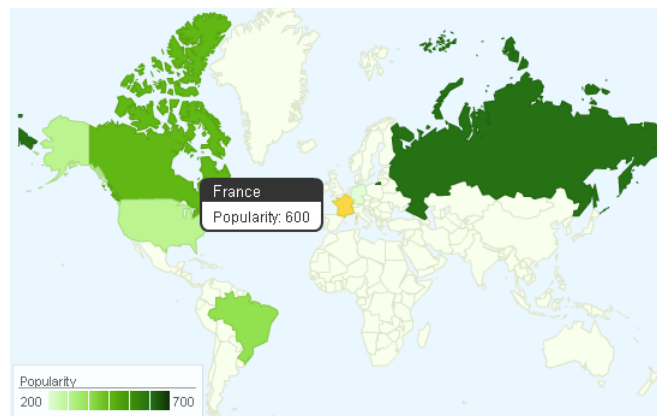


Figura 4.1: Geomapa generado con Google Visualization API

- **The JavaScript InfoVis Toolkit [11].** Está desarrollada por completo en JavaScript y goza de un gran potencial para representar y manejar grafos, *treemaps* y *sunbursts*.

- **Open Flash Chart** [13] (Fig. 4.2). Es una librería libre y gratuita, con licencia LGPL, que se basa en flash para generar los diagramas. Además, incluye un plug-in para ser integrado con numerosos lenguajes, entre ellos Java y el framework Struts2.

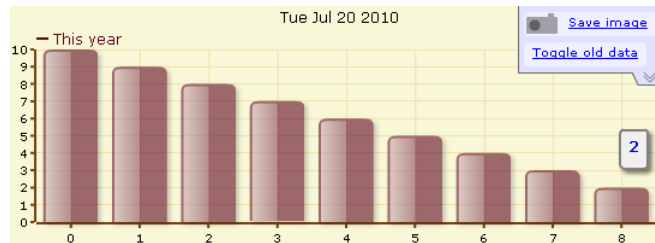


Figura 4.2: Diagrama de barras generado con Open Flash Chart

- **JOGL: JavaOpenGL.** Es un *binding* de Java para el API de OpenGL (JSR-231), que permite desarrollar aplicaciones que manejen gráficos 3D. JOGL implementa todas las funciones de OpenGL 1.3 - 3.0, 3.1, ES 1.x y ES 2.x, y se integra con AWT y Swing.
- **WebGL: OpenGL ES 2.0 for the Web.** Es una especificación estándar para un API basada en OpenGL ES 2.0, que permite desplegar gráficos en 3D a través del elemento *canvas* de HTML 5.

BIBLIOGRAFÍA

- [1] 22 Awesome Visualization Libraries: Charts and Diagrams. <http://www.onextrapixel.com/2009/11/24/22-awesome-visualization-libraries-charts-and-diagrams/>, Nov. 2009.
- [2] Giles, A. y G. Daich: *Metrics tools*. Crosstalk, 1995. <http://www.stsc.hill.af.mil/crosstalk/1995/02/Metrics.asp>.
- [3] ISO: *ISO/IEC 14598 Software engineering. Product evaluation*.
- [4] ISO: *ISO/IEC 9126 Software engineering. Product quality*. 1991.
- [5] Jacobson, I., G. Booch y J. Rumbaugh: *El proceso unificado de desarrollo de software*. Addison Wesley, 2000.
- [6] Mellado, D., M. Rodríguez, J. Verdugo, M. Piattini y E. Fernández-Medina: *Evaluación de la calidad y seguridad en productos software*.
- [7] Nascimbene, C.: *¿Qué son las Rich Internet Applications?*, Dic. 2005. <http://www.canal-ar.com.ar/noticias/noticiamuestra.asp?Id=2639>.
- [8] Página web de Dojo Toolkit. <http://www.dojotoolkit.org/>.
- [9] Página web de DWR: Easy Ajax for Java. <http://directwebremoting.org/dwr/index.html>.
- [10] Página web de Google Visualization API. http://code.google.com/intl/es-ES/apis/visualization/interactive_charts.html.
- [11] Página web de JavaScript InfoVis Toolkit. <http://thejit.org/>.
- [12] Página web de jQuery. <http://jquery.com/>.
- [13] Página web de Open Flash Chart v2. <http://teethgrinder.co.uk/open-flash-chart-2/>.

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 *Cuadro de Mando para la Visualización de la Calidad del Software, Integrado con un Entorno de Medición*.

Ciudad Real, a 7 de marzo de 2011

El alumno,

VºBº El director,

VºBº El tutor académico

Fdo.:

Fdo.:

Fdo.:

SR. DIRECTOR DE LA ESCUELA SUPERIOR DE INFORMÁTICA

B. CONTRATO DE PROPIEDAD INTELECTUAL

En Ciudad Real, a 7 de marzo de 2011

REUNIDOS

DE UNA PARTE, **D. Jose Domingo López López** mayor de edad, con D.N.I. número **71219116-F**, Alumno de la Escuela Superior de Informática, y en adelante el **“ALUMNO DEL PFC”**.

DE OTRA PARTE, **Dña. María de los Ángeles Moraga de la Rubia** mayor de edad, con D.N.I. número **5683662-V**, Profesora de la Escuela Superior de Informática, y en adelante el **“TUTOR ACADÉMICO DEL PFC”**.

DE OTRA PARTE, **D. Moisés Rodríguez Monje** mayor de edad, con D.N.I. número **5928188-F**, Director Técnico de la empresa Alarcos Quality Center (en adelante AQC), y en adelante el **“DIRECTOR DEL PFC”**.

Todos ellos reconociéndose mutuamente capacidad suficiente para la celebración del presente Contrato,

EXPONEN

PRIMERO: El objeto del presente contrato es la determinación de la propiedad intelectual del PFC titulado **“Cuadro de Mano para la Visualización de la Calidad del Software, Integrado con un Entorno de Medición”**, y del resultado de la integración del mismo con las herramientas usadas por AQC. Así como, establecer el compromiso de la confidencialidad que el **ALUMNO DEL PFC** contraerá con AQC.

SEGUNDO: La propiedad intelectual del producto software resultante del **PFC** integrado con las herramientas y entornos de la empresa **AQC** pertenecerá única y exclusivamente a **AQC**, no pudiendo el **ALUMNO DEL PFC** utilizarlas sin el consentimiento expreso de la empresa. Sin embargo, la propiedad intelectual del **PFC** de manera aislada, así como de los documentos, diagramas, modelos, etc. que lo acompañen será compartida a partes iguales entre el **ALUMNO DEL PFC**, el **TUTOR DEL PFC** y el **DIRECTOR DEL PFC**.

TERCERO: El **ALUMNO DEL PFC** trabajará para la realización de su **PFC** con documentación, productos software y entornos de la empresa **AQC** y por ello, por medio de la presente, se compromete de forma irrevocable ante **AQC** a:

1. Asumir las políticas, los objetivos y las directrices establecidas por **AQC**, y particularmente la política de calidad, las directrices y procedimientos incluidos o referenciados en el manual de calidad de **AQC**.

2. Actuar de forma que se preserve una imagen positiva y eficaz de **AQC** guardando al debida diligencia en relación con todas las labores efectuadas, actuando en todo momento de buena fe y en aras al leal cumplimiento de mis obligaciones para con **AQC**.
3. Evitar intervenir en cualquier actividad que pueda disminuir la confianza en la competencia, imparcialidad, juicio o integridad operativa de **AQC**.
4. Asegurar la imparcialidad y estar libre de toda presión indebida, comercial, financiera o de otra índole, que pueda influir en el juicio técnico empleado en la realización de sus actividades en **AQC**.
5. No alegar su calidad de colaborador de **AQC** con fines ajenos a las misiones que se le hayan confiado por **AQC**. No admitir ningún tipo de retribución ajena a **AQC** por cualquier actividad realizada en nombre de **AQC** o en calidad de persona cualificada por **AQC**.
6. En particular, durante la realización de las actividades del **PFC**:
 - Detectar desviaciones / no conformidades basadas en evidencias objetivas y no en opiniones personales o valoraciones subjetivas.
 - No adoptar nunca actitudes de prepotencia ante la empresa.
 - No tener inconveniente en modificar una opinión si la empresa le demuestra que estaba equivocado.
 - No ceder ante presiones o coacciones que pueda recibir si está convencido de sus criterios y no le demuestran con contrario.
7. Participar, en la medida de lo posible, en las actividades, reuniones o cursos de armonización de criterios siempre y cuando sea requerido por **AQC**. Compartir con el personal de **AQC**, su conocimiento, experiencia, ideas y sugerencias.
8. Avisar a **AQC** de cualquier cambio o interrupción en sus actividades profesionales.
9. Durante su relación con **AQC** y tras ella, guardar el más estricto secreto sobre todos los asuntos, documentos y registros confidenciales de carácter técnico, organizativo, comercial y económico de **AQC** y aquella que pudiera tener conocimiento dentro de marco de sus actividades. Tratando como documentación confidencial:
 - Solicitud, formal o informal, de un proyecto. Queda incluido el hecho de la existencia de la solicitud.
 - La entrega por una organización que ha solicitado o ha realizado un proyecto con **AQC**.
 - Los datos de los productos o servicios o de las organizaciones que ha solicitado o ha realizado un proyecto con **AQC**.
 - Toda la correspondencia entre **AQC** y las organizaciones con que trabaja o tiene relación.
 - Toda la correspondencia entre **AQC** y sus subcontratistas y viceversa.
 - La que los documentos del sistema señalen como confidencial.

- Los documentos del sistema de gestión de **AQC**.
 - Todos los documentos que **AQC** le facilite para la realización exitosa de su **PFC**, así como las herramientas y modelos con los que el **ALUMNO DEL PFC** deba integrar su trabajo.
10. Adoptar todas las precauciones para evitar que se divulguen, directa o indirectamente, por causa propia o de las personas de las que es responsable, documentos o informaciones de los que pudiera tener conocimiento dentro del marco de sus actividades en los proyectos de **AQC**. Estas precauciones incluyen:
- Manipular y mantener debidamente archivada la documentación.
 - No dejar incontrolada documentación confidencial en salas de reuniones y visitas.
 - Evitar que personas ajenas accedan a lugares de trabajo y archivos.
11. En caso de acceso y tratamiento de datos de carácter personal, por exigencias el trabajo que desempeñe, asegurará la confidencialidad de los mismos respetando y cumpliendo en todo momento los procedimientos y directrices instaurados en **AQC** al efecto y la legislación vigente aplicable.
12. Respetar la titularidad de **AQC** sobre cualquier proyecto, informe, documento e imágenes que realice en el desempeño de las tareas que se le encomiendan durante el **PFC** (respetando y protegiendo los derechos de propiedad intelectual o industrial de **AQC**).

Y en prueba de cuando antecede, las Partes suscriben el Contrato, en cuatro ejemplares y a un solo efecto, en el lugar y fecha señalados en el encabezamiento.

EL ALUMNO DEL PFC

EL TUTOR DEL PFC

EL DIRECTOR DEL PFC