

# TESINA DE LICENCIATURA

**TITULO:** Ambiente de Certificación de Calidad en Pequeñas y Medianas Empresas de Desarrollo de Soft

**AUTORES:** Daniel Nanni, Augusto Soncini

**DIRECTOR:** Lic. Patricia Pesado

**CODIRECTOR:**

**CARRERA:** Licenciatura en Sistemas

## Resumen

El desarrollo de software representa una actividad económica de suma importancia, ya que ofrece múltiples fuentes de ingresos y empleo y se perfila como una de las oportunidades más importantes en los países en vía de desarrollo. Pero en muchos países, las empresas de desarrollo de software normalmente pequeñas y medianas empresas (PYMES), no están preparadas para competir internacionalmente. El sector informático se enfrenta a una serie de problemas como la dependencia tecnológica y metodológica (especialmente de EEUU), la falta de formación sobre los procesos del ciclo de vida del software y sobre la calidad del mismo.

Por lo que en la mayoría de los casos asistimos a una construcción de software de baja calidad, totalmente artesanal, con tiempo de desarrollo inapropiados y la correspondiente insatisfacción de los clientes y usuarios finales. Ya que el papel del software crece de manera exponencial, urge la necesidad de articular una estrategia para conseguir la construcción de productos de calidad.

Esto se logra en gran medida con la utilización de modelos internacionales tales como CMMI o SPICE, o alguno de los modelos emergentes tales como MoProSoft.

## Líneas de Investigación

Ingeniería de Software.  
Calidad del Software.  
Modelos de procesos existentes.  
Organización de las PYMES de Desarrollo de Software.  
Modelo MoProSoft.

## Conclusiones

En la actualidad la calidad de los procesos de desarrollo de software se convirtió en un factor primordial para las empresas a la hora de obtener productos de calidad y ser competitivos en un mercado cada vez más demandante. Para las PYMES, en particular, poder llegar a definir procesos de calidad es aún muy costoso y difícil de lograr.

Luego de nuestro análisis de los modelos de procesos existentes, concluimos que MoProSoft es el modelo que representa una de las mejores opciones para la pequeña y mediana industria, ya que presenta de manera pragmática las mejores prácticas para el desarrollo de software. Por ello, consideramos que la herramienta construida para el soporte del proceso de recolección y evaluación de datos basada en MoProSoft, va a resultar de gran ayuda en el camino de certificación de calidad en las PYMES de software.

## Trabajos Realizados

Recopilación de las normas internacionales de calidad más importantes del mercado.  
Análisis de las características de las medianas y pequeñas empresas de desarrollo de software.  
Análisis de modelos de calidad emergentes para las PYMES.

Estudio del modelo MoProSoft, incluyendo el análisis de un cuestionario a aplicar para el relevamiento del proceso seleccionado.  
Construcción de una herramienta Web para el relevamiento y evaluación de aspectos de calidad como apoyo al proceso de certificación basado en MoProSoft.

## Trabajos Futuros

-Continuar investigando sobre modelos internacionales existentes y nuevas iniciativas sobre modelos para PYMES.

-Continuar refinando la herramienta para el soporte del proceso de recolección y evaluación de datos.

**Fecha de la presentación:** Diciembre de 2008



UNLP  
Facultad de Informática  
Tesina de Grado

Ambiente de Certificación de Calidad en Pequeñas  
y Medianas Empresas de Desarrollo de Soft

*Augusto Soncini,  
Daniel Nanni*

*Director*

*Lic. Patricia Pesado*

*Colaboración*

*C.C. Silvia Esponda*

# 1. CONTENIDO

<b>1. CONTENIDO</b> .....	<b>2</b>
<b>2. INTRODUCCIÓN</b> .....	<b>4</b>
2.1. DE QUE TRATA ESTA TESINA .....	4
2.2. SITUACIÓN ACTUAL .....	4
2.3. FORMULACIÓN DEL PROBLEMA .....	4
<b>3. CALIDAD DEL PROCESO DE DESARROLLO DE SOFTWARE</b> .....	<b>7</b>
3.1. INTRODUCCIÓN .....	7
3.2. EVOLUCIÓN .....	8
3.3. ORGANIZACIONES MADURAS E INMADURAS .....	8
3.4. VENTAJAS.....	10
3.5. QUÉ IMPLICA UN PROCESO DE DESARROLLO DE SOFTWARE .....	11
3.6. CONCLUSIONES.....	11
<b>4. ESTUDIO DE NORMAS INTERNACIONALES EXISTENTES</b> .....	<b>14</b>
4.1. CMM - CMMI .....	14
<i>Introducción</i> .....	14
<i>Descripción del modelo CMM</i> .....	14
<i>Descripción del modelo CMM Integrado</i> .....	17
4.2. ISO/IEC 15504 .....	19
4.3. MR MPS .....	25
4.4. SIMEP – SW .....	27
4.5. METRICA V.3 .....	28
<b>5. ESTUDIO DEL MODELO MOPROSOFT</b> .....	<b>32</b>
5.1. ¿QUÉ ES MoPROSOFT?.....	32
5.2. HISTORIA Y OBJETIVO .....	32
5.3. MODELO.....	33
5.4. ESQUEMA GENERAL .....	33
5.5. MoPROSOFT E ISO/IEC 29110 .....	38
<b>6. ANÁLISIS Y DESARROLLO DE LA HERRAMIENTA WEB</b> .....	<b>42</b>
6.1. SOLUCIÓN PROPUESTA .....	42
6.2. ALCANCE.....	42
6.3. SEGURIDAD .....	42
6.3.1. <i>Autenticación de usuario</i> .....	42
6.3.2. <i>Administración de usuarios</i> .....	42
6.3.3. <i>Administración de permisos</i> .....	43
6.4. CARACTERÍSTICAS TÉCNICAS .....	43
6.4.1. <i>Arquitectura</i> .....	43
6.4.2. <i>Base de Datos</i> .....	43
6.5. FUNCIONALIDAD BÁSICA.....	43
<b>7. DESCRIPCIÓN FUNCIONAL</b> .....	<b>45</b>

7.1.	PERFIL USUARIO FINAL .....	45
7.1.1.	<i>Realizar una evaluación.</i> .....	45
7.1.2.	<i>Obtener conclusiones sobre las evaluaciones.</i> .....	45
7.1.3.	<i>Generar reportes.</i> .....	45
7.1.4.	<i>Consultar evaluaciones históricas.</i> .....	45
7.2.	PERFIL ADMINISTRADOR.....	45
7.2.1.	<i>Administrar cuestionarios.</i> .....	45
7.2.2.	<i>Administrar preguntas.</i> .....	46
7.2.3.	<i>Administrar respuestas.</i> .....	46
7.2.4.	<i>Búsquedas sobre el sistema.</i> .....	46
<b>8.</b>	<b>ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE .....</b>	<b>47</b>
8.1.	REQUERIMIENTOS MÓDULO USUARIO FINAL. ....	47
8.1.1.	<i>Listar cuestionarios.</i> .....	47
8.1.2.	<i>Listar evaluaciones.</i> .....	47
8.1.3.	<i>Realizar evaluación.</i> .....	47
8.1.4.	<i>Continuar una evaluación.</i> .....	47
8.1.5.	<i>Guardar evaluación parcialmente terminada.</i> .....	47
8.1.6.	<i>Ver detalle de evaluación.</i> .....	47
8.1.7.	<i>Imprimir detalle de evaluación.</i> .....	47
8.1.8.	<i>Ver conclusiones de evaluación.</i> .....	48
8.1.9.	<i>Imprimir conclusiones de evaluación.</i> .....	48
8.2.	REQUERIMIENTOS MÓDULO ADMINISTRADOR. ....	49
8.2.1.	<i>Dar de alta un cuestionario.</i> .....	49
8.2.2.	<i>Dar de baja un cuestionario.</i> .....	49
8.2.3.	<i>Modificar un cuestionario.</i> .....	49
8.2.4.	<i>Dar de alta una pregunta.</i> .....	49
8.2.5.	<i>Modificar una pregunta.</i> .....	50
8.2.6.	<i>Dar de baja una pregunta.</i> .....	50
8.2.7.	<i>Dar de alta tipos de respuestas específicas.</i> .....	50
8.2.8.	<i>Buscar cuestionarios.</i> .....	50
8.2.9.	<i>Buscar preguntas.</i> .....	51
8.2.10.	<i>Generar cuestionario impreso.</i> .....	51
8.2.11.	<i>ABM de datos de dominio.</i> .....	51
<b>9.</b>	<b>GUÍA DE USO.....</b>	<b>52</b>
9.1.	CREAR UN CUESTIONARIO.....	52
9.1.1.	<i>Ingreso Al sistema.</i> .....	52
9.1.2.	<i>Menú principal del usuario Administrador.</i> .....	53
9.1.3.	<i>Crear cuestionario.</i> .....	54
9.2.	RESPONDER UN CUESTIONARIO.....	69
9.2.1.	<i>Ingreso Al sistema.</i> .....	69
9.2.2.	<i>Menú principal.</i> .....	70
9.2.3.	<i>Listar Cuestionarios.</i> .....	71
<b>10.</b>	<b>CONCLUSIÓN.....</b>	<b>76</b>
<b>11.</b>	<b>BIBLIOGRAFÍA.....</b>	<b>78</b>

## 2. INTRODUCCIÓN

### 2.1. DE QUE TRATA ESTA TESINA

---

El objetivo de esta tesina consiste en la construcción de una herramienta Web para el relevamiento y evaluación de aspectos de calidad, como apoyo al proceso de certificación de PYMES de desarrollo de software, basada en el modelo MoProSoft.

Para lograr esto, se desarrollaron las siguientes actividades:

- Recopilación de las normas internacionales de calidad más importantes del mercado.
- Análisis de las características de las medianas y pequeñas empresas de desarrollo de software.
- Análisis de modelos de calidad emergentes para las PYMES.
- Estudio del modelo **MoProSoft**, incluyendo el análisis de un cuestionario a aplicar para el relevamiento del proceso seleccionado.

### 2.2. SITUACIÓN ACTUAL

---

El desarrollo de software representa una actividad económica de suma importancia, ya que ofrece múltiples fuentes de ingresos y empleo y se perfila como una de las oportunidades más importantes en los países en vía de desarrollo. Pero en muchos países, las empresas de desarrollo de software normalmente pequeñas y medianas empresas (PYMES), no están preparadas para competir internacionalmente. El sector informático se enfrenta a una serie de problemas como la dependencia tecnológica y metodológica (especialmente de EEUU), la falta de formación sobre los procesos del ciclo de vida del software y sobre la calidad del mismo. Por lo que en la mayoría de los casos asistimos a una construcción de software de baja calidad, totalmente artesanal, con tiempo de desarrollo inapropiados y la correspondiente insatisfacción de los clientes y usuarios finales. Ya que el papel del software crece de manera exponencial, urge la necesidad de articular una estrategia para conseguir la construcción de productos de calidad.

### 2.3. FORMULACIÓN DEL PROBLEMA

---

La calidad de los productos está íntimamente ligada a la calidad de los procesos utilizados para desarrollarlos y gestionarlos. A nivel internacional se han desarrollado varios modelos para la mejora y evaluación de procesos de software, destacando el modelo ISO/IEC 15504(ISO 2004) y, especialmente, los modelos CMM y CMMI desarrollados por el SEI (Software Engineering Institute) de EEUU.

El problema de muchos de estos modelos es que están pensados para grandes empresas de desarrollo de software, y su adopción por parte de las Pymes es difícil y costosa.

Otro aspecto a tener en cuenta, es el alto costo de las evaluaciones y el alto costo de los cursos sobre los métodos y modelos de evaluación. Un informe del propio (SEI) insiste en lo caro y difícil de implementar que resulta CMMI y SCAMPI( método evaluador asociado a CMMI). Si esto es así en las propias organizaciones norteamericanas, implica que en los países iberoamericanos alcanza un costo prohibitivo.

Las pequeñas y medianas empresas de desarrollo de software necesitan certificar calidad para posicionarse competitivamente en el mercado nacional e internacional. No obstante la madurez del proceso en estas organizaciones todavía se encuentra en un estado crítico.

En los últimos años han aparecido un gran número de estándares y propuestas internacionales y regionales relacionadas con la mejora de los procesos de software para medianas y pequeñas empresas de desarrollo de software.

Esto evidencia el creciente interés en la comunidad de Ingeniería del Software en abordar el tema. Este interés se suscita porque la industria del software en la mayoría de los países y en particular los iberoamericanos, está formada en gran parte por este tipo de empresas. Es importante entonces fortalecerlas con prácticas y guías eficientes de Ingeniería del Software adaptadas a su tamaño y tipo de negocio.

A principios del presente siglo XXI la comunidad de Ingeniería del Software (industria e investigadores) ha expresado especial interés en la mejora de procesos de software (conocida por las siglas SPI, del término en inglés Software Process Improvement) en medianas y pequeñas empresas desarrolladoras de software con menos de 50 empleados (a las cuales se hace referencia por la sigla VSEs del término en inglés Very Small Software Enterprises). Esto se evidencia por el creciente número de artículos que tratan el tema según el análisis de la tendencia de las publicaciones de mejora de procesos en pequeñas y medianas empresas presentado en [1], así como por la aparición de un gran número de estándares y propuestas internacionales y regionales relacionadas con SPI para VSEs.

Este creciente interés acerca de SPI en VSEs, se suscita como se mencionó anteriormente, porque este tipo de empresas son una pieza muy importante en el engranaje de la economía mundial, además la industria del software en la mayoría de los países está formada por tejido industrial compuesto en gran parte por VSEs. Según estudios presentados en [2], [3] y [4], tanto en Europa como en Iberoamérica, la gran mayoría de las compañías del sector de las tecnologías de la información son pequeñas o medianas.

Este tipo de organizaciones desarrollan productos que, para su construcción, necesitan estrategias, prácticas y/o guías eficientes de ingeniería de software adaptadas a su tamaño y tipo de negocio. Siguiendo esta premisa es importante entonces ayudar a estas organizaciones a entender y usar las prácticas ofrecidas por los estándares y propuestas internacionales y regionales relacionadas con SPI para VSEs.. Con la información pertinente sobre SPI en VSEs, las pequeñas organizaciones cuentan con más elementos de juicio para facilitar la adopción e implantación de las propuestas relacionadas con mejora de procesos.

Algunos países están intentando abordar el problema descrito, con algunas iniciativas dignas de destacar como el modelo MoProSoft de México, modelo en el cual se centra nuestra tesina, el modelo MR mps de Brasil o el modelo Competisoft de Colombia.

## **3. CALIDAD DEL PROCESO DE DESARROLLO DE SOFTWARE**

### **3.1. INTRODUCCIÓN**

---

Luego de décadas de promesas incumplidas sobre mejoras de productividad y producción gracias a la aplicación de nuevas metodologías y tecnologías de software, organizaciones industriales y gubernamentales llegaron a la conclusión de que su fundamental problema es la incapacidad de manejar el proceso de desarrollo de software. Los beneficios de la utilización de nuevos métodos y herramientas no se pueden obtener dentro del desorden de un proyecto indisciplinado y caótico. En muchas organizaciones, los proyectos frecuentemente llevan un tiempo excesivo y llegan hasta duplicar el tiempo planeado por el presupuesto. En tales casos, la organización no está proveyendo la infraestructura y soporte necesarios para ayudar al proyecto a evitar estos problemas.

Aún en organizaciones indisciplinadas, algunos proyectos de software producen excelentes resultados. Cuando tales proyectos tienen éxito, generalmente es debido a los esfuerzos heroicos de un equipo dedicado, en lugar de provenir de repetir métodos probados de una organización con un proceso de desarrollo del software.

En la ausencia de un proceso de software que abarque a toda la organización, la repetición de resultados depende enteramente de tener a los mismos individuos disponibles para el próximo proyecto. El éxito que descansa sólo en la habilidad de individuos específicos no provee la base de una mejora de productividad y calidad a largo plazo dentro de una organización.

La calidad de los procesos de software se está convirtiendo en un elemento estratégico de las grandes organizaciones debido a su fuerte impacto en la competitividad de las empresas, el siguiente paso es que estos procesos lleguen hasta las PYMES.

Un proceso de software puede definirse como un conjunto de actividades métodos prácticas y transformaciones que la gente emplea para desarrollar y mantener el software y los productos asociados (por ejemplo, planes de proyectos, documentos de diseño, código casos de testeo y manuales de usuario).

La premisa fundamental de manejar procesos de software es que la calidad del producto de software es ampliamente determinada por la calidad de los procesos usados para desarrollarlo y mantenerlo. Un efectivo proceso de software debe estar ligado a la gente, las herramientas y los métodos integrados todos juntos.

La capacidad de los procesos de software describe el rango de expectativa de resultados que pueden ser logrados siguiendo un proceso de software. La capacidad de un proceso de software proporciona un significado de predicción, el resultado más probable a ser esperado del próximo proyecto emprendido por la organización.

Otro concepto importante a tener en cuenta es el denominado maduración de los procesos de software, que se podría definir como el alcance para el cual un proceso especificado es explícitamente definido, manejado, medido, controlado, y efectivo. La

maduración implica un potencial de crecimiento en capacidad, e indican la riqueza de un proceso de una organización de software y la consistencia con la cual es aplicada en proyectos a través la organización.

La maduración de los procesos de software implica que la capacidad de los mismos debería ir creciendo. El mejoramiento requiere un fuerte manejo del soporte y una consistencia enfocada a largo plazo. Cuando una organización de software madura, necesita una infraestructura y una cultura que soporte estos métodos, prácticas y procedimientos de manera que pueda subsistir luego de ser originalmente definida.

### **3.2. EVOLUCIÓN**

---

Las empresas que desarrollan software no pueden ignorar que su negocio es un negocio de software, y que el modelo que cada una adopte para las actividades de desarrollo y mantenimiento tiene implicaciones relevantes en la eficiencia general de los desarrollos que emprenden.

Según como afrontan las organizaciones el desarrollo del software, éste puede comportarse como factor de riesgo o amenaza; o por el contrario como una poderosa oportunidad de negocio. Todas las empresas quieren producir más rápido, mejor y con menores costes, y sin duda esto es posible porque la naturaleza del software no es origen de riesgos y problemas, sino una fuente de oportunidades. Cada vez más directivos están comprendiendo que la forma de gestión del desarrollo de software, puede hacer de la materia prima de su negocio un material arisco de resultados impredecibles, o una ventaja competitiva. La evolución hacia entornos de ingeniería del software requiere cambios severos en la organización, así como el convencimiento, implicación y empuje de la dirección. Pero sobre todo el diseño de un modelo de producción propio que sepa aprovechar la personalidad de la organización, y responder a las particularidades de su negocio.

Las organizaciones que desarrollan o mantienen software pueden optar por trabajar "a la buena de Dios", o por seguir una metodología. Hacerlo a la buena de Dios no es tan raro. Es la primera forma que se empleó para desarrollar programas. En realidad, cuando en la segunda mitad del siglo pasado la industria del hardware puso en la calle máquinas que se podían programar, los pioneros de nuestra profesión, atraídos por el encanto de diseñar y construir artefactos útiles, y verlos funcionar, fueron los primeros en remangarse frente al teclado. Fueron también los primeros en pasar noches en vela programando, prometiendo que "tan sólo es cuestión de un par de días más." Aunque las cinco décadas de vida del software ya han sido suficientes para experimentar los excesos y los errores de la infancia y la juventud, la resistencia al cambio es el mejor aliado de la inercia, y por eso se produce una cierta impermeabilidad a la experiencia. En cualquier caso es una opción: trabajar sin ninguna metodología.

### **3.3. ORGANIZACIONES MADURAS E INMADURAS**

---

En una organización inmadura, los procesos del software generalmente son improvisados por los programadores y sus gerentes durante el curso del proyecto. Aún si el proceso del software ha sido especificado, no es rigurosamente seguido o cumplido.

Una organización inmadura es reaccionaria, y los gerentes generalmente enfocan su atención en resolver crisis inmediatas y urgentes (mejor conocido como “apagar incendios”). Las agendas y presupuestos son excedidos frecuentemente porque no se basan en estimaciones realistas. Cuando se imponen fechas de entrega estrictas, la funcionalidad del producto y la calidad generalmente son dejadas de lado para cumplir con los plazos.

En una organización inmadura, no existe una base objetiva para juzgar la calidad de los productos o para resolver problemas del proceso o de los productos. Por tal razón la calidad es difícil de pronosticar. Las actividades relacionadas con el mejoramiento de la calidad tales como revisiones y pruebas frecuentemente son reducidas o eliminadas cuando los proyectos se exceden de la agenda.

Por otro lado, una organización de software madura posee la habilidad de gerenciar los procesos de desarrollo y mantenimiento del software a nivel organización. El proceso del software es precisamente comunicado tanto al personal existente como a los nuevos empleados, y las actividades de trabajo son llevadas a cabo de acuerdo con el plan del proceso.

Los procesos obligatorios son adecuados para ser utilizados y consistentes con la forma en que el trabajo realmente es llevado a cabo. Tales procesos definidos son actualizados cuando se considera necesario, y se le realizan mejoras a través de pruebas piloto controladas y/o análisis de costo-beneficio. Los roles y responsabilidades dentro del proceso son establecidos claramente durante el proceso y en toda la organización.

En una organización madura, los gerentes monitorean la calidad del software y la satisfacción del cliente. No existe ningún principio objetivo o cuantitativo para juzgar la calidad y analizar problemas del producto y el proceso.

Las agendas, planes y presupuestos se basan en información histórica y son realistas; los resultados esperados de costo, agenda, funcionalidad, y calidad del producto generalmente son alcanzados. En general, un proceso disciplinado es consistentemente seguido pues todos los participantes entienden la importancia de hacerlo y la infraestructura necesaria para apoyar tal proceso existe dentro de la organización.

El SEI, (Software Engineering Institute) define una escala del 1 al 5 para determinar el grado de madurez de una organización, y en consecuencia el nivel de garantía que ofrece en cuanto a calidad, predictibilidad en los resultados y eficiencia en el desarrollo de software, este tipo de organizaciones, las denominadas inmaduras, se quedan, lógicamente, en el 1. Que, no quiere decir que necesariamente van a producir mal software, de forma ineficiente y con retraso, sino que las probabilidades de cumplir las tres facetas son bajas. Hay equipos que lo consiguen, pero son pocos. La razón es sencilla, los resultados en estos casos son tan buenos como las personas que los desarrollan, y lo cierto es que los buenos programadores escasean. A este modo de producción se le ha venido a llamar "programación heroica", porque todo el peso del resultado descansa sobre el "saber-hacer" de los programadores. El éxito o fracaso de

las organizaciones que trabajan sin metodologías depende del conocimiento tácito de su personal, pero teniendo en cuenta que se trata del conocimiento que cada uno traía ya de la calle, o del que adquiere *motu proprio*, porque estamos hablando de "ninguna metodología". Este es sin duda el modelo con el que empiezan muchas empresas "start-up": un equipo de emprendedores con talento, capaces de construir sistemas de software con calidad. Los resultados serán tan buenos como ellos los sepan hacer. El cumplimiento de agendas dependerá de su capacidad de previsión y organización. Pero no hay que engañarse; en este caso no se trata de empresas que saben hacer software, sino de personas que saben hacer software. Y esta situación dibuja el guión común a todas las pequeñas empresas de desarrollo de software que surgen de cero, impulsadas tan sólo por el empuje de sus emprendedores: pueden llegar todo lo lejos que la combinación del talento y de la capacidad de marketing de sus creadores les permitan. O sea, en ocasiones acabarán cerrando, y en otras alcanzarán un nivel de estabilidad tan mediocre o tan brillante como dicha combinación les permita. Y si una vez logrado ese nivel de equilibrio desean proyectar mayor crecimiento a su organización se enfrentan a un reto importante: pasar de personas que saben hacer software a empresa que sabe hacer software. Salto que supone que no van a ser ya ellos, sino su empresa quien deberá producir de forma eficiente y repetible en todos sus proyectos, resultados de calidad. Aquí es donde entra en juego la gestión por procesos.

### 3.4. VENTAJAS

---

La gestión por procesos encierra cuatro factores fundamentales:

**Repetibilidad de resultados.** Al conseguir que la calidad del resultado sea consecuencia de la calidad del proceso, producir aplicando el mismo proceso garantiza la homogeneidad de los resultados.

**Escalabilidad.** Es una consecuencia de la repetibilidad. No sólo un equipo consigue resultados homogéneos en todos los proyectos, sino que los obtienen todos los equipos.

**Mejora continua.** Al aplicar meta-procesos que trabajan sobre los propios procesos de producción, midiendo y analizando los resultados se obtienen los criterios de gestión necesarios para aplicar medidas que mejoran de forma continua la eficiencia y calidad de los procesos base, y por tanto de los resultados.

Un **know-how propio**, consiguiendo finalmente una empresa que sabe hacer, porque su modelo de procesos termina conteniendo un activo valioso de la organización: la clave para hacer las cosas bien, con eficiencia y de forma homogénea.

Un ejemplo claro que ilustra las diferencias entre ambos sistemas de producción es el montaje de un mueble en *kit*. Si dejamos el proceso a un lado (el manual con las instrucciones de cómo proceder) tendremos un sistema perfecto de producción

heroica: personas más tecnología; del que cabrá esperar los resultados propios de estos entornos. La productividad y la calidad no son homogéneas. Hay quien ensambla la estantería o el mueble, en cuestión de minutos, y quien necesita toda una tarde. Algunos obtienen estanterías robustas y perfectas, y otros terminan el montaje con peor suerte, afirmando que la estantería está bien montada pero que las piezas venían mal cortadas de fábrica, o que los tornillos son de mala calidad, o quizá sin ni siquiera percatarse de los defectos con los que ha dejado el mueble. En definitiva esta forma de trabajo produce resultados, pero para una empresa de montaje de muebles no resultaría viable un sistema de producción que no pudiera garantizar eficiencia en los costes, y resultados con una calidad homogénea.

Al añadir procesos al sistema se consigue que todas las personas ejecuten el trabajo siguiendo las mismas pautas, y que los parámetros de ejecución y los resultados puedan ser medidos de forma homogénea en toda la organización.

De esta forma se reduce drásticamente la heterogeneidad de los resultados, tanto en la eficiencia de los tiempos invertidos, como en la calidad obtenida, que empieza a ser proporcional a la calidad de los procesos que se hayan diseñado.

### **3.5. QUÉ IMPLICA UN PROCESO DE DESARROLLO DE SOFTWARE**

---

Un marco de procesos determinará, por ejemplo, la necesidad de gestionar adecuadamente las modificaciones de requisitos, o de realizar un diseño detallado antes de comenzar la codificación. El valor de los modelos de procesos radica en el conocimiento que aportan al señalar las actividades cuya omisión incrementa las posibilidades de fracaso en los proyectos.

Sin embargo el marco de procesos no dirá, por ejemplo, que deba emplearse un sistema basado en documentos o en base de datos para la gestión de los requisitos; o si el diseño debe realizarse con diagramas IDEF o UML.

Este es el campo de la técnica de la ingeniería del software.

Una organización posee un proceso de desarrollo de software de calidad cuando el mismo es respaldado mediante documentación y capacitación del personal; y el proceso es continuamente monitoreado y mejorado por sus usuarios. La maduración de un proceso implica que la productividad y calidad resultantes del proceso de software de la organización pueden ser mejoradas con el tiempo mediante ganancias consistentes en la disciplina alcanzada mediante la utilización de tal proceso.

A medida que una organización de software gana en calidad de su proceso de software, institucionaliza su proceso mediante políticas, estándares, y estructuras organizativas. La institucionalización requiere la construcción de una infraestructura y una cultura corporativa que sustente los métodos, prácticas y procedimientos del negocio de tal forma que perduren aún después de que los individuos que los definieron se hayan ido de la organización.

### **3.6. CONCLUSIONES**

---

La calidad de los procesos de software se está convirtiendo en un elemento estratégico de las grandes organizaciones debido a su fuerte impacto en la competitividad de las empresas, el siguiente paso es que estos procesos lleguen hasta las PYMES.

Cuando una organización encara proyectos de desarrollo de software, tiene la necesidad de seleccionar un proceso para su desarrollo, no es suficiente dejar que el conocimiento y el esfuerzo de los individuos involucrados sean aplicados en forma uniforme. Es necesario tener presente que la calidad de un producto es directamente proporcional a la del proceso con que es generado.

En este marco, un modelo de procesos es un conjunto estructurado de elementos que describen las características de procesos efectivos y de calidad, indicando “qué hacer”, no “cómo hacer” ni “quién lo hace”.

En los últimos años en que los estándares de calidad internacionales se han reorientado hacia el aseguramiento de la calidad, han aparecido en la industria del software dos importantes modelos de referencia: Modelo CMM (Capability Maturity Model) y su sucesor CMMI y Modelo SPICE (Software Process Improvement and Capability determination).

Estadísticas del Software Engineering Institute (SEI) reportan que la gran mayoría de las empresas que aplican CMMI, no alcanzan los últimos niveles. Esto da cuenta de la rigurosidad del modelo y de las pocas posibilidades que tendrían las PYMES de ubicarse en los niveles superiores.

Las grandes empresas ya consolidadas pueden disponer a su voluntad, y de hecho lo hacen, de la implantación de otras metodologías o esquemas propios para lograr la calidad y la mejora continua de sus procesos productivos, sin embargo, esto no resulta tan fácil para las PYMES, las cuales necesitan modelos que definan requisitos básicos de la organización y permitan poder llegar a certificaciones que le confieran prestigio ante sus clientes. Conceptualmente, el auge de los procesos de calidad se debe a que el software está cada vez más “inmerso” en los productos y servicios que se producen. Además, representa un elemento clave en la innovación de las empresas, tanto en productos como en procesos, es un componente de alto riesgo derivado de la capacidad y madurez de los procesos que intervienen en su desarrollo y es una actividad esencialmente artesanal.

Cada vez más organizaciones están inmersas en el proceso de implementación de estos modelos en sus organizaciones de software, e incluso han sido ya certificadas en alguno de los modelos internacionalmente reconocidos.

Se debe tener en cuenta que implementar en una organización un modelo de procesos para el desarrollo de software, implica un cambio cultural importante que debe ser bien manejado.

El reto principal, consiste en extender los modelos de calidad de procesos de software a las PYMES, que son mayoría en el sector de desarrollo de software y que, de forma directa o indirecta, vía subcontratación, representa más del 70% del software que se produce. Y es que mientras las grandes empresas se encuentran ya adelantadas en todo lo que afecta a la mejora de sus procesos de software, las PYMES del sector se encuentran realmente rezagadas. Aquí es donde se hacen fuerte las nuevas normas de certificación enfocadas a la pequeña industria, como es nuestro caso de estudio: **MoProSoft**.

Para finalizar, podemos decir que para lograr el éxito en la construcción de software, se deben seguir procesos de calidad, que guíen en todas las etapas del desarrollo, facilitando la realización de productos de software de calidad. Esto se logra en gran medida con la utilización de modelos internacionales tales como CMMI o SPICE, o alguno de los modelos emergentes tales como MoProSoft.

## 4. ESTUDIO DE NORMAS INTERNACIONALES EXISTENTES

### 4.1. CMM - CMMI

---

#### Introducción

---

Durante las últimas décadas, el desarrollo de software, evidenció la necesidad de un marco en el cual ordenar y sistematizar los procesos de desarrollo y gestión de los proyectos. Durante varios años el Departamento de Defensa de EEUU financió numerosas investigaciones que arrojaron como resultados estadísticos, la escasez de proyectos que terminaban exitosamente dentro del tiempo, costo y funcionalidad prevista.

En respuesta a esta situación, el Departamento de Defensa de los EEUU apoyó la formación del SEI (Software Engineering Institute, Carnegie Mellon University) con el propósito de estudiar el problema y desarrollar modelos con el objetivo de encontrar alguna solución.

En 1991, el SEI publica el modelo CMM (Capability Maturity Model). El modelo está orientado a la mejora de los procesos relacionados con el desarrollo de software y de gestión.

CMM ha servido como marco de referencia para la implementación de mejoras de procesos en organizaciones en muchas partes del mundo y se han gastado miles de millones de dólares en estas implementaciones.

Posteriormente, el SEI desarrolló modelos similares a CMM para otras disciplinas, entre las cuales figuraban la ingeniería de sistemas (SE-CMM, *Systems Engineering Capability Maturity Model*), la adquisición de software (SA-CMM, *Software Acquisition Capability Maturity Model*), las personas (P-CMM, *People Asistente para la Evaluación de CMMI-SW Capability Maturity Model*), y el desarrollo integrado de productos (IPD-CMM, *Integrated Product Development Capability Maturity Model*).

A mediados de la década del 90, el SEI decide unificar los modelos de ingeniería de software (SW-CMM, también conocido como CMM), de ingeniería de sistemas (SE-CMM) y de desarrollo integrado de productos (IPD-CMM), dando origen a CMMI (Integración del Modelo de Madurez de Capacidad, en inglés *Capability Maturity Model Integration*) en el año 2002.

El nuevo modelo CMMI brinda un marco con una estructura común para todas las disciplinas y agrega una nueva forma de representación además de la conocida representación por *Niveles*. La nueva forma de representación se llama *Continua* y está orientada a medir la mejora en los procesos de manera individual en vez de hacerlo de manera conjunta como la representación por niveles.

#### Descripción del modelo CMM

---

Los aspectos claves del modelo son, por un lado, la clasificación de las organizaciones en maduras e inmaduras y, luego, la prescripción del camino a seguir por una organización inmadura para evolucionar y convertirse en una organización madura.

El modelo entiende por organización inmadura aquella que lleva adelante sus proyectos sin una definición previa de los procesos a seguir. Estos proyectos frecuentemente sobrepasan sus presupuestos y tiempos de terminación debido a que son iniciados con estimaciones poco realistas, sin una planificación adecuada, y son llevados adelante sin ningún tipo de gestión. En general estos proyectos no terminan, o terminan con una disminución importante en la calidad esperada del producto.

Por organizaciones maduras el modelo entiende a aquellas que desarrollan sus proyectos en forma planeada. Los presupuestos asignados y el tiempo previsto son los necesarios porque se parte de estimaciones metódicas y basadas en datos de proyectos previos, con roles y responsabilidades bien definidos.

Para que una organización se convierta en madura debe evolucionar con el tiempo alcanzando sucesivos niveles de madurez.

El modelo identifica los siguientes **niveles de madurez** dentro de los cuales puede encontrarse una organización:

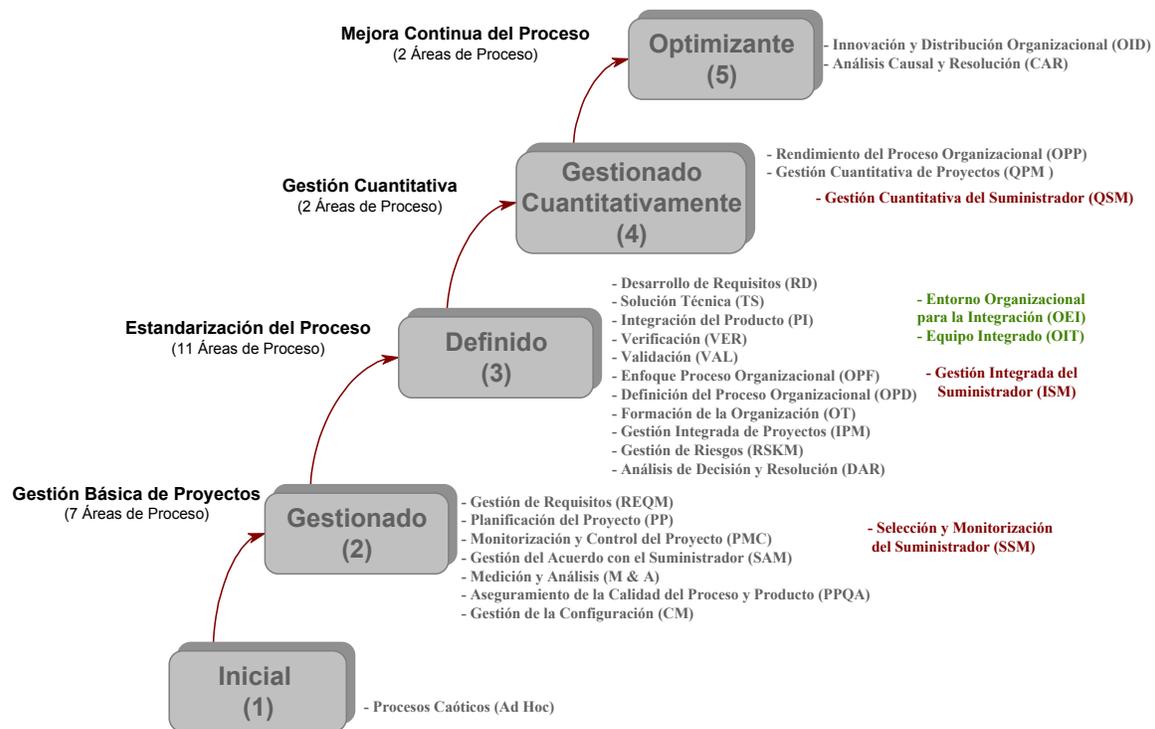


Figura 4.1

**Nivel 1 – Inicial:** el proceso de software es impredecible, sin control y reactivo. El éxito de los proyectos depende del talento de los individuos.

**Nivel 2 – Repetible:** procesos de administración básicos establecidos para lograr el seguimiento de los costos, tareas y funcionalidad. Los procesos existentes hacen que se puedan repetir éxitos en proyectos de similares características.

**Nivel 3 – Definido:** además de las definiciones del nivel anterior, son incorporadas actividades de administración de ingeniería en forma documentada, estandarizada e integradas en una familia de procesos normalizados de la organización. Los proyectos utilizan una versión adaptada de esas normas para su desarrollo.

**Nivel 4 – Gestionado:** se llevan adelante los proyectos en forma controlada con métricas que permiten mediciones confiables de los procesos y productos.

**Nivel 5 - Optimizado:** incluye la mejora continua de procesos a partir de la comparación y análisis de mediciones sucesivas de los proyectos.

En general, una organización mejora notablemente su rendimiento a medida que la misma incrementa su nivel de madurez. La figura 4.2 muestra de manera conceptual las mejoras en el rendimiento para cada nivel contemplando los factores Tiempo y Costo [Paulk et al, 1993].

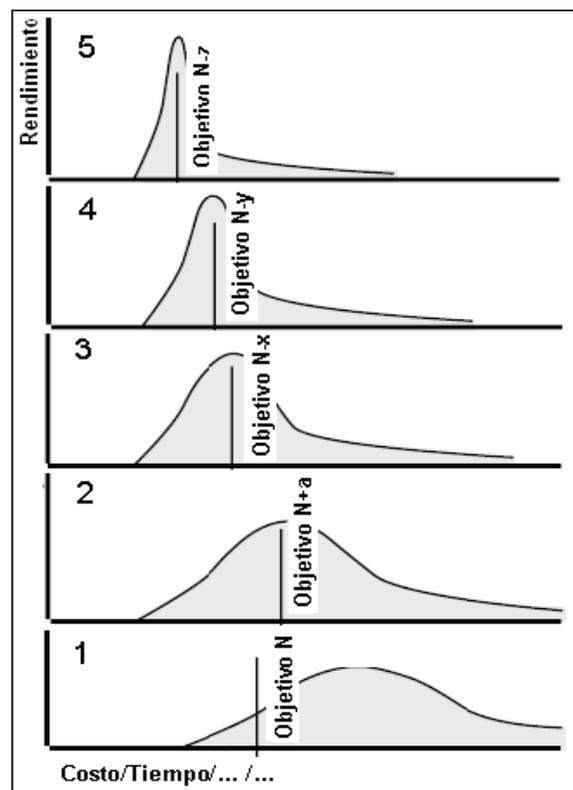


Figura 4.2

Del gráfico, se desprenden las siguientes observaciones:

- En las organizaciones que se encuentran en el nivel 1, los objetivos propuestos generalmente son ampliamente excedidos por la realidad.
- En las organizaciones que se encuentran en el nivel 2, se establecen objetivos más acordes a la realidad.
- En las organizaciones que se encuentran los niveles 3, 4, y 5, existe una menor dispersión de la realidad con respecto a los objetivos propuestos, y la performance mejora con cada nivel.

Algunas organizaciones que adoptaron el modelo fueron: Accenture, AT&T, Boeing, Ericsson, Fuji Xerox, Motorola, Nasa, United Airlines. En Argentina se puede mencionar a Thinksoft Argentina S.A., Planta Motorola Córdoba y la consultora Hexacta entre otros.

El gran problema que trae aparejado este modelo para las organizaciones, es en lo referente a los costos y tiempos necesarios para la preparación previa a su adopción o a una acreditación. Más acentuado se encuentra este problema en organizaciones pequeñas, en donde los recursos humanos, económicos y temporales, suelen ser muy inferiores que en las grandes organizaciones. Suelen utilizar como un paso preparatorio el concepto de “Evaluación Interna”. Esta evaluación es difícil de llevar a cabo y no existe un soporte adecuado de herramientas que faciliten las cosas.

### **Descripción del modelo CMM Integrado**

---

Además de la integración de varias disciplinas, los modelos CMMI introdujeron otra novedad que afectaba a su implantación:

CMMI al igual que CMM tiene dos utilidades. Puede servir tanto como guía para la mejora en una organización, o como criterio para evaluar su nivel; pero mientras CMM centraba estas dos finalidades en la dimensión de la madurez de la organización, CMMI introduce una segunda dimensión, también válida para guiar las actividades de mejora y para evaluar a las organizaciones: la capacidad de los procesos.

CMMI establece 6 niveles para determinar la capacidad de un proceso:

**Nivel 0.- Incompleto:** el proceso no se realiza.

**Nivel 1.- Realizado:** se lleva a cabo el proceso, consiguiendo transformar elementos de entrada identificados, en productos de salida.

**Nivel 2.- Gestionado:** el proceso se ejecuta siempre de la misma manera, de una forma gestionada.

**Nivel 3.- Definido:** el proceso está definido en la organización y se ejecuta siempre.

**Nivel 4 – Cuantificadamente gestionado:** la ejecución del proceso tiene institucionalizado en la organización un sistema de medición objetivo y cuantificable de su capacidad.

**Nivel 5.- Optimizado:** el proceso, que se ejecuta siempre, está definido en la organización, se mide y está integrado en un plan, también institucionalizado, de mejora continua basada en las mediciones de los procesos.

De esta forma los modelos CMMI presentan 2 versiones:

**Versión escalonada:** guía a la organización sobre las áreas de procesos que debe ir abordando, las prácticas que debe implantar y los objetivos que debe alcanzar para ir consiguiendo los sucesivos niveles de madurez.

**Versión continua:** permite cierta libertad a la organización sobre las áreas de proceso que desea mejorar, y le orienta para ir elevando su nivel de capacidad. Consecuentemente al evaluar a una organización se puede hacer sobre la dimensión de su madurez, estableciendo cuál es el nivel que ha alcanzado, o sobre la dimensión de la capacidad, reflejando cuál es el nivel de cada una de las áreas de proceso contempladas por el modelo.

### **Puntos fuertes y débiles**

No cabe duda de que CMMI, tanto por su difusión y resultados reportados, es un modelo exitoso obteniendo brillantes resultados en incremento de la productividad, más rápida respuesta al mercado, reducción de defectos, disminución de costes y planificaciones fiables.

Son numerosos los casos de estudio, por ejemplo [5][6][7] probando la mejora sustancial que supone la participación en una iniciativa de mejora de procesos basado en modelos de madurez y realizando interesantes comparativas entre metodologías.

Sin embargo, por lo reciente de los nuevos estándares emergentes, han quedado anticuados y aún hay pocos resultados cuantitativos publicados sobre CMMI. SEI publicó un informe especial [8] que, si bien no es estadísticamente significativo, presenta ‘evidencia cuantitativa creíble’ de las mejoras en rendimiento y calidad obtenidas en 12 casos de estudio, tanto en coste (ahorros y disminución en coste de encontrar y reparar errores), planificación (disminución del tiempo necesario para terminar tareas y aumento de la fiabilidad de las predicciones sobre estimaciones), calidad (reducción de tasa de defectos), satisfacción del cliente como retorno de la inversión.

Entre sus fortalezas podríamos destacar:

- Inclusión de las prácticas de institucionalización, que permiten asegurar que los procesos asociados con cada área de proceso serán efectivos, repetibles y duraderos.
- Guía paso a paso para la mejora, a través de niveles de madurez y capacidad (frente a ISO).
- Transición del ‘aprendizaje individual’ al ‘aprendizaje de la organización’ por mejora continua, lecciones aprendidas y uso de bibliotecas y bases de datos de proyectos mejorados.

Algunas de sus debilidades son:

- El CMMI puede llegar a ser excesivamente detallado para algunas organizaciones.
- Puede ser considerado prescriptivo.
- Requiere mayor inversión para ser completamente implementado.
- Puede ser difícil de entender.

Existen influencias negativas de CMMI en el proceso de mejora, en sus dos representaciones, como por ejemplo dar la idea de que sólo se pueden mejorar áreas de proceso del actual nivel de madurez, centrarse más en alcanzar el siguiente nivel de madurez, más que la mejora medible de los objetivos de negocio de la organización, o que se preste excesiva atención a aspectos de gestión, dejando a un lado aspectos técnicos, o que podamos mejorar áreas de proceso según nuestro interés obviando las relaciones y dependencias entre ellas.

### **Análisis de la aplicación de CMMI en pequeñas empresas:**

- No existencia de una guía a medida de pequeñas organizaciones. Reconocen que inicialmente está dirigido a grandes corporaciones. Sin embargo la estructura organizativa del modelo permite seleccionar sólo aquellas áreas de proceso que puedan ser asumidas por pequeñas empresas. Crecimiento cuasi-exponencial del número de áreas y prácticas, tiempo, recursos y costes, pero si se alinean los procesos a las necesidades de la organización, se beneficiarán de un proceso estructurado.
- CMMI es demasiado normativo, en especial con pequeñas organizaciones que, además, funcionan y evolucionan de distinta manera que las grandes.
- CMMI parece escrito para organizaciones ya maduras y vagamente escrito para ser usado en valoraciones.

## **4.2. ISO/IEC 15504**

---

### **Introducción**

**ISO (Organización Internacional de Normalización)** es una federación mundial de organismos nacionales de normalización (organismos miembros de ISO). El trabajo de preparación de las normas internacionales normalmente se realiza a través de los comités técnicos de ISO.

Debido a la alta competencia del mercado de desarrollo de software, a la difícil tarea de identificar los riesgos, cumplir con el calendario, controlar los costos y mejorar la eficiencia y calidad, es que en el año 1993 la comisión ISO/IEC JTC1 aprobó un programa de trabajo para el desarrollo de un modelo que fuera la base de un futuro estándar internacional para la evaluación de los procesos del ciclo de vida del software. Este trabajo recibió el nombre de proyecto SPICE (Software Process Improvement and Capability dEtermination), y en junio de 1995, con la publicación de su primer borrador,

desde ISO fueron invitadas diferentes organizaciones para aplicarlo y valorar sus resultados.

En 1998, pasada la fase de proyecto, el trabajo pasó a la fase de informe técnico con la denominación ISO/IEC TR 15504. La instrucción técnica consta de 9 apartados, recogidos en volúmenes independientes que se han ido publicando como redacción definitiva del estándar internacional ISO/IEC 15504 durante el periodo 2003 - 2005.

Esta norma rige la evaluación de proceso del software, proporcionando un marco para todos los aspectos de una evaluación sobre la capacidad de los procesos de una organización. Este engloba un modelo de referencia para los procesos y sus potencialidades sobre la base de la experiencia de compañías grandes, medianas y pequeñas.

SPICE provee:

- Un marco de referencia para determinar las fortalezas y debilidades de los procesos;
- Un marco de referencia para mejorar los procesos de software, y medir sus mejoras;
- Un marco de referencia para los que adquieren un sistema para evaluar la capacidad de los proveedores de sistemas;
- Un marco de referencia para determinar los riesgos de negocio para una empresa que considera desarrollar un nuevo producto de software o servicio.
- Una ruta para la armonización o migración de los modelos de evaluación de procesos con referencia al modelo de procesos y capacidad. La organización o unidad informática puede utilizar SPICE para los siguientes casos:
  - Modo de nivel de capacidad o potencialidad. Le permite a una organización adquirente determinar el nivel de capacidad de una empresa que le suministra un sistema de software.
  - Modo de mejoramiento de procesos. Para ayudar a una organización a mejorar su propio departamento de desarrollo de software y los procesos de mantenimiento.
  - Modo auto-evaluación. Para ayudar a una organización determinar su habilidad de implementar un nuevo proyecto de software

### **Descripción del Modelo SPICE**

El modelo describe los procesos que una organización puede ejecutar, adquirir, suplir, desarrollar, operar, evolucionar, brindar soporte de software y todas las prácticas genéricas que caracterizan las potencialidades de estos procesos.

La arquitectura del modelo organiza las prácticas en números de categorías usando diferentes tipos de aproximaciones. La arquitectura distingue entre:

- Prácticas base, son las actividades esenciales de un proceso específico, agrupado por categorías de procedimientos y procesos de acuerdo al tipo de actividad que direccionan.
- Prácticas genéricas, aplicables a cualquier proceso, que representa las actividades necesarias para administrar el "proceso" y mejorar su potencialidad.

El modelo agrupa a los procesos en cinco categorías:

1. Procesos Cliente - Proveedor (Customer - Supplier) esta categoría consiste en los procesos que directamente impactan al cliente, al soporte de desarrollo y a la transición del software al cliente.
2. Procesos de Ingeniería (Engineering) esta categoría consiste, a los procesos que directamente especifican, implementa, y mantienen un sistema, un producto de software y la documentación del usuario.
3. Procesos de Proyecto (Project) esta categoría consiste en los procesos establecidos dentro del proyecto, coordinación y administración de los recursos para producir un producto o proveer un servicio para satisfacer al cliente.
4. Procesos de Soporte (Support) esta categoría consiste en los procedimientos que establecen y soportan el desempeño de los otros procesos del proyecto.
5. Procesos de la Organización (Organization) esta categoría consiste en los procesos que establecen las metas de negocio de la organización, los procesos de desarrollo y recursos que ayudan a la organización alcanzar dichas metas.

La categoría de proceso "Cliente-Proveedor" consiste en los procesos que están más cerca al cliente. "Ingeniería" consiste en los procesos que construye el producto que es entregado al cliente. El "proyecto" consiste en los procesos que administra el desarrollo de los productos. "Soporte" brinda apoyo a las otras categorías de procesos. Finalmente la categoría "Organizacional" son los procesos que construyen la infraestructura organizacional sobre la cual cada uno de los procesos descritos pueden potencialmente ser ejecutados y maximizar los beneficios para la organización.

Las categorías de procesos están relacionadas y cada categoría es una capa que precede a la otra, como se indica en la figura 4.2.1.

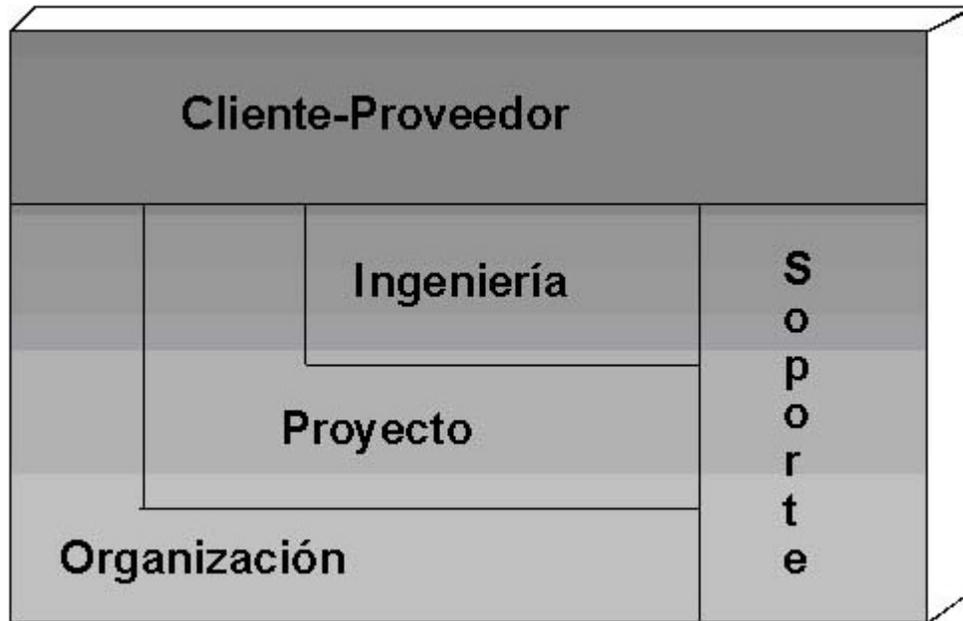


Figura 4.2.1

Cada proceso en el modelo es descrito en términos de prácticas de base, cada una son actividades únicas de ingeniería de software o de administración. La categoría de los Procesos y las prácticas base están agrupadas por tipo de actividad.

La evolución de la capacidad de los procesos (capability process) esta expresado en términos de niveles de capacidad, características comunes, y prácticas genéricas. Un nivel de capacidad es un conjunto de actividades que trabajan juntas para proveer una mejor ejecución de los procesos. Cada nivel provee una mejor y más compleja ejecución de los procesos que el nivel predecesor.

**Existen seis niveles de capacidad en el modelo:**

- Nivel 0; Incompleto: el proceso no ha sido implementado o falla en la consecución de su propósito. No hay evidencia del logro de los resultados esperados.
- Nivel 1; Realizado Informalmente: Las prácticas bases de los procesos generalmente son ejecutados. La ejecución de dichas prácticas puede no ser rigurosamente planificadas y seguidas. La ejecución de las prácticas dependerá del conocimiento y esfuerzo personal. Se identifica algunos procesos.
- Nivel 2; Planificado y Seguido: La ejecución de las prácticas bases en los procesos son planificadas y seguidas. La performance estará acorde con los

procedimientos especificados. La primera distinción entre el nivel 1 y el 2 es que la ejecución de los procesos está planificada y administrada y progresan hacia un proceso bien definido.

- Nivel 3; Bien Definido; El proceso se realiza y se gestiona utilizando procedimientos definidos. Cada implementación de un proceso se hace utilizando procedimientos creados según un estándar y debidamente documentados, además, se dispone de los recursos necesarios para alcanzar los propósitos establecidos. La principal diferencia con el Nivel 2, es que se utiliza un proceso definido y con capacidad para alcanzar los resultados esperados.
- Nivel 4; Cuantitativamente Controlado: Mediciones detalladas de la performance o ejecución son recabadas y evaluadas. Es conocida en forma cuantitativa la performance de los procesos y es posible su predicción. Las prácticas son objetivamente administradas. La calidad de las mismas es cuantitativamente conocida.
- Nivel 5; Mejoramiento Continuo: Son establecidos en forma cuantitativa procesos y metas eficientes, basados en los objetivos de la organización. En forma continua los procesos se van mejorando mediante la retroalimentación (feedback) obtenido por los resultados de procesos definidos y por ideas pilotos y tecnologías novedosas.

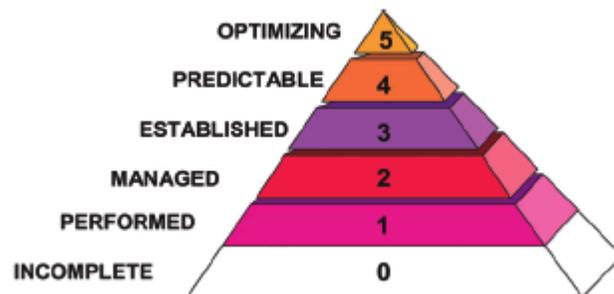


Figura 4.2.2

## **Puntos fuertes y puntos débiles**

SPICE ha sido un proyecto de lenta maduración, que ha variado mucho desde su borrador inicial.

Muchas de las críticas o mejoras propuestas a lo largo de estos años por diversos autores, han sido recogidas así como ciertos aspectos positivos (como la representación continua del modelo) han sido adoptadas por otros.

Cabe destacar el considerable número de estudios de evaluación empírica realizados, pudiendo revisar sus conclusiones más significativas en [9], en cuanto a validez predictiva, veracidad de la capacidad, demostración de su capacidad para identificación de fuerzas, debilidades y riesgos así como dirigir y priorizar el proceso de mejora. Trabajos previos en la misma línea pueden revisarse en [10].

Una de las mayores contribuciones del modelo SPICE consiste en ser el primer modelo de procesos de dos dimensiones, dimensiones independientes para los procesos y la capacidad.

Como puntos débiles podemos enumerar:

- La dimensión capacidad ha alcanzado un alto grado de dificultad y existen solapamientos con la dimensión procesos.
- La complejidad de las evaluaciones (y por consiguiente el costo) es significativamente más alta que en otros modelos.

En [11] se afirma que el desarrollo de ISO 15504 ha acercado a lo mejor de los expertos internacionales en evaluación de procesos, y a través de la sinergia de estas relaciones ha llevado a significativos avances en el estado del arte y en la argumentación teórica del proceso. Así, en [12] encontramos un reconocimiento a la influencia de SPICE en el desarrollo de CMMI.

## **Diferencias con CMMI**

Algunas diferencias con CMM que merecen ser mencionadas, son que SPICE contempla controles que CMM no tiene, como por ejemplo la preocupación de contratación de un proveedor de software, establecimiento de contratos comerciales entre ambas partes, preocupación con el servicio de mantenimiento, preocupación con relación a la entrega de la instalación de software y evaluación de la satisfacción del cliente.

Por último, merecen ser destacados los estudios sobre implantación de SPICE en pequeñas empresas [13], y de lo inadecuado del uso de SPICE para metodologías ágiles [14].

### 4.3. MR MPS

---

Algunos países tales como Brasil, se han preocupado por la calidad de los procesos de desarrollo de software para su industria, que en su gran mayoría está compuesta por empresas pequeñas, prueba de esto es el modelo “MR mps”.

Estudios sobre la calidad en el sector de software brasileño muestran la necesidad de un esfuerzo significativo capaz de aumentar la madurez de los procesos de software de las empresas. El Proyecto MPS.Br - Mejora de Proceso del Software Brasileño - es una iniciativa que comprende Universidades, grupos de investigación y empresas, bajo la coordinación de la Sociedad SOFTEX (Sociedad para Promoción de la Excelencia del Software Brasileño).

El proyecto tiene como finalidad la definición y la diseminación de un Modelo de Referencia y un Modelo de Negocio para la mejora del proceso de software (MR.MPS y MN.MPS, respectivamente). El Modelo de Negocio tiene gran potencial de replicabilidad en Brasil y en otros países de características semejantes en lo que se refiere a la industria de software.

El Proyecto es estructurante y promueve la calificación de un grupo amplio de empresas compatible con los estándares de calidad aceptados internacionalmente por la comunidad de software, a costos accesibles para la mayoría de las empresas brasileñas, siendo adecuado al perfil y la cultura de cada una de ellas.

Este proyecto consiste en la elaboración de un Modelo de Referencia de Procesos, basado en los conceptos de madurez y capacidad de proceso, para evaluación y mejora de la calidad y productividad de productos y servicios de software. Por un lado, pretende una mayor utilización por la industria nacional y por otro, una capacitación para definición de ajustes y adaptaciones de los modelos internacionales para alineamiento con la realidad y decisiones de la política de software brasileña.

“MPSps Br” [15], basado en ISO/IEC 12207:2002, CMMI e ISO/IEC 15504:2003, define e implementa un modelo para la mejora de procesos de software. Pretende dar respuesta a la pregunta ¿Cómo mejorar radicalmente los procesos de software en Brasil, con foco en un número significativo de PyMES de forma que estas obtengan un nivel de madurez 2 o 3 a un costo accesible?

El proyecto “MPS Br”, desarrolla dos modelos:

- Modelo de Referencia para la mejora del proceso del software – “MR mps”
- Modelo de Negocio para la mejora del proceso del software – “MN mps”.

“MN mps” define los elementos e interacciones involucrados para la certificación de la empresa a través de la implementación de “MR mps” de dos maneras:

- personalizada para una empresa

- conjunta entre un grupo de empresas (logrando así costos más accesibles para PyMES).

“MR mps” comprende diferentes niveles de madurez y un método de evaluación. Los niveles de madurez están organizados en dos dimensiones: de capacidad y de proceso.

La madurez del proceso define 7 niveles. A cada nivel de madurez se le atribuyen áreas de proceso con base en los niveles de CMMI, para posibilitar la implementación gradual y adecuada en las PyMES brasileñas. A continuación se muestra un cuadro con los diferentes niveles y que procesos se corresponden con cada uno.

<b>7 Niveles</b>	<b>21 Procesos</b>
<b>A – Optimizado</b>	Implantación de Innovaciones en la Organización – IIO Análisis de Causas y Resolución - ARC
<b>B – Gestionado Cuantitativamente</b>	Desempeño del Proceso Organizacional - DEP Gestión Cuantitativa del Proyecto – GQP
<b>C – Definido</b>	Gestión de Riesgos - GRI Análisis de Decisión y Resolución – ADR
<b>D – Ampliamente Definido</b>	Desarrollo de Requisitos - DRE Solución Técnica - STE Validación - VAL Verificación - VER Integración del Producto – ITP
<b>E – Parcialmente Definido</b>	Entrenamiento - TRE Definición del Proceso Organizacional – DFP Evaluación y Mejora del Proceso Organizacional – AMP Adaptación del Proceso para Gestión de Proyecto – APG
<b>F – Gestionado</b>	Gestión de Configuración - GCO Aseguramiento de la Calidad – GQA Medición – MED Adquisición - AQU
<b>G – Parcialmente Gestionado</b>	Gestión de Proyecto - GPR Gestión de Requisitos – GRE

El método de evaluación, a partir de indicadores, asigna un nivel de implementación de una práctica relacionada a un área de proceso.

En cuanto al modelo de evaluación, “MR mps” lo define de la intersección de las dimensiones de madurez y cumplimiento del proceso relacionándolo con el esquema de niveles de madurez de CMMI, en su representación escalonada.

#### 4.4. SIMEP – SW

Teniendo como premisa que las empresas de desarrollo de software de Latinoamérica deben implementar proyectos de mejoramiento de procesos de desarrollo y de calidad, existen diferentes emprendimientos. Es así como en Colombia nace SIMEP –SW .

En el marco del proyecto SIMEP-SW se ha definido una estrategia de mejora, la cual intenta cubrir dos esfuerzos: el de alivianar requisitos y guiar en el proceso de mejora, así como el de generar un conjunto de recomendaciones prácticas para la implementación de los requisitos del proceso software.

Es por esto que el proyecto Sistema Integral para el Mejoramiento de los Procesos de Desarrollo de Software en Colombia, SIMEP-SW busca proporcionar a las empresas del sector informático de Colombia las herramientas necesarias para motivarlas a mejorar sus procesos de desarrollo de software con el objetivo de facilitar el posicionamiento y la competitividad en mercados nacionales e internacionales.

El proyecto pretende crear, aplicar y probar un sistema de mejoramiento que integre elementos de modelos de calidad, mejoramiento y evaluación reconocidos internacionalmente, adaptados a las características propias de la industria del software colombiana y que pueda ser replicado a industrias de características similares a nivel nacional e internacional [16].

Se pretende que los proyectos de mejoramiento que realicen las empresas de Colombia sigan un modelo nacional coherente a las características propias de la idiosincrasia y al contexto socio-económico del país.

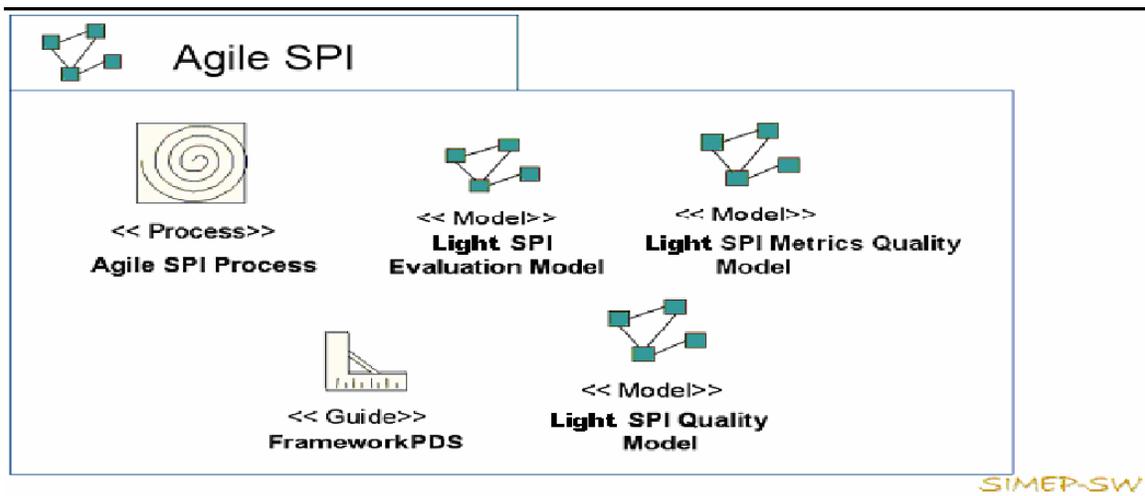


Figura 4.4.1.

El resultado del proyecto SIMEP-SW es Agile SPI (Software Process Agile Improvement) [17], con la premisa esencial que los modelos utilizados sean ligeros y basados en estándares internacionales, acordes a las características, idiosincrasia y circunstancias de la realidad socio- económica de la naciente industria del software Colombiano.

La arquitectura preliminar de Agile SPI, se presenta en la figura 4.4.1, de la cual se observan los siguientes componentes:

- Agile SPI Process: Un proceso ágil que guía a un programa de mejora de procesos.
- Light SPI Evaluation Model: Un modelo ligero de evaluación del proceso productivo.
- Light SPI Metrics Quality Model: Un modelo ligero de métricas del proceso productivo.
- Framework PDS: Un marco conceptual y tecnológico para soportar procesos.
- Light SPI Quality Model: Un modelo de calidad ligero.

SIMEP-SW basa su estrategia de mejora en proporcionar a la organización un proceso ágil que guía a un programa de mejora de procesos. Para esto es indispensable contar con un modelo de evaluación ligero ya que, para poder promover la mejora de los procesos de software, es muy importante establecer previamente un marco de evaluación con el fin de conocer sus puntos fuertes y débiles. La evaluación de los procesos software tiene como objetivo detectar aspectos de un proceso de software que se pueden mejorar [14].

#### **4.5. METRICA V.3**

---

**METRICA** es una Metodología de Planificación, Desarrollo y Mantenimiento de Sistemas de información. Promovida por el Ministerio de Administraciones Públicas del gobierno español para la sistematización de actividades del ciclo de vida de los proyectos software en el ámbito de las administraciones públicas. Esta metodología propia está basada en el Modelo de Procesos del Ciclo de vida de desarrollo ISO/IEC 12207 (*Information Technology - Software Life Cycle Processes*) así como en la norma ISO/IEC 15504 SPICE (*Software Process Improvement And Assurance Standards Capability Determination*)

La metodología METRICA Versión 3 ofrece a las Organizaciones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software dentro del marco que permite alcanzar los siguientes objetivos:

- Proporcionar o definir Sistemas de Información que ayuden a conseguir los fines de la organización mediante la definición de un marco estratégico para el desarrollo de los mismos.
- Dotar a la organización de productos software que satisfagan las necesidades de los usuarios dando una mayor importancia al análisis de requisitos.

- Mejorar la productividad de los departamentos de Sistemas y Tecnologías de la Información y las Comunicaciones, permitiendo una mayor capacidad de adaptación a los cambios y teniendo en cuenta la reutilización en la medida de lo posible.
- Facilitar la comunicación y entendimiento entre los distintos participantes en la producción de software a lo largo del ciclo de vida del proyecto, teniendo en cuenta su papel y responsabilidad, así como las necesidades de todos y cada uno de ellos.
- Facilitar la operación, mantenimiento y uso de los productos de software obtenidos.

La nueva versión de MÉTRICA contempla el desarrollo de Sistemas de Información para las distintas tecnologías que actualmente están conviviendo y los aspectos de gestión que aseguran que un proyecto cumple sus objetivos en términos de calidad, costo y plazos.

Su punto de partida es la versión anterior de MÉTRICA de la cual se han conservado la adaptabilidad, flexibilidad y sencillez, así como la estructura de actividades y tareas, si bien las fases y módulos de MÉTRICA versión 2.1 han dado paso a la división en Procesos, más adecuada a la entrada-transformación-salida que se produce en cada una de las divisiones del ciclo de vida de un proyecto. Para cada tarea se detallan los participantes que intervienen, los productos de entrada y de salida así como las técnicas y prácticas a emplear para su obtención.

En la elaboración de MÉTRICA Versión 3 se han tenido en cuenta los métodos de desarrollo más extendidos, así como los últimos estándares de ingeniería del software y calidad, además de referencias específicas en cuanto a seguridad y gestión de proyectos. También se ha tenido en cuenta la experiencia de los usuarios de las versiones anteriores para solventar los problemas o deficiencias detectados.

En una única estructura la metodología MÉTRICA Versión 3 cubre distintos tipos de desarrollos: estructurado y orientado a objetos, facilitando a través de interfaces la realización de los procesos de apoyo u organizativos: Gestión de Proyectos, Gestión de Configuración, Aseguramiento de Calidad y Seguridad.

La automatización de las actividades propuestas en la estructura de MÉTRICA Versión 3 es posible ya que sus técnicas están soportadas por una amplia variedad de herramientas de ayuda al desarrollo.

Además, para facilitar la utilización de dicha metodología existe una herramienta de software, *Gestor Metodológico*, de ayuda a la aplicación de la metodología en cada proyecto concreto y que permite adaptar la estructura de MÉTRICA Versión 3 de acuerdo a las características del mismo, permitiendo el seguimiento y control de sus actividades y tareas realizadas por distintos perfiles de usuario asignados a los participantes por el jefe de proyecto.

Se ha desarrollado también un software, *Selector de Herramientas*, que ayuda a seleccionar entre las CASE del mercado la que mejor se adapta a las necesidades de cada proyecto teniendo en cuenta las características de cada organización.

Así mismo se ha elaborado un curso de autoformación, que sirva para aprender todos los conceptos y elementos de la metodología MÉTRICA Versión 3 contemplando varios niveles y diversos perfiles[19].

## Procesos

MÉTRICA Versión 3 tiene un enfoque **orientado al proceso**, ya que la tendencia general en los estándares se encamina en este sentido y por ello, como ya se ha dicho, se ha enmarcado dentro de la norma ISO 12.207, que se centra en la clasificación y definición de los procesos del ciclo de vida del software. Como punto de partida y atendiendo a dicha norma, MÉTRICA Versión 3 cubre el Proceso de Desarrollo y el Proceso de Mantenimiento de Sistemas de Información. Dichos procesos son:

- Planificación de Sistemas de Información (PSI).
- Desarrollo de Sistemas de Información (DSI). Debido a su complejidad, está a su vez dividido en cinco procesos:
  - Estudio de Viabilidad del Sistema (EVS).
  - Análisis del Sistema de Información (ASI).
  - Diseño del Sistema de Información (DSI).
  - Construcción del Sistema de Información (CSI).
  - Implantación y aceptación del Sistema (IAS).
- Mantenimiento de Sistemas de Información (MSI).

## Interfaces

La estructura de MÉTRICA Versión 3 incluye también un conjunto de interfaces que definen una serie de actividades de tipo organizativo o de soporte al proceso de desarrollo y a los productos, que en el caso de existir en la organización se deberán aplicar para enriquecer o influir en la ejecución de las actividades de los procesos principales de la metodología y que si no existen habrá que realizar para complementar y garantizar el éxito del proyecto desarrollado con MÉTRICA Versión 3.

Las interfaces descritas en la metodología son:

- Gestión de Proyectos (GP)
- Seguridad (SEG)
- Aseguramiento de la Calidad (CAL)
- Gestión de la Configuración (GC)

Todo proyecto **Métrica 3** consta de un conjunto de fases que se desglosan en múltiples puntos cuya cronología hay que seguir con claridad para ir avanzando en el desarrollo del proyecto. El figura 4.5.1 muestra el esquema general de estas fases:

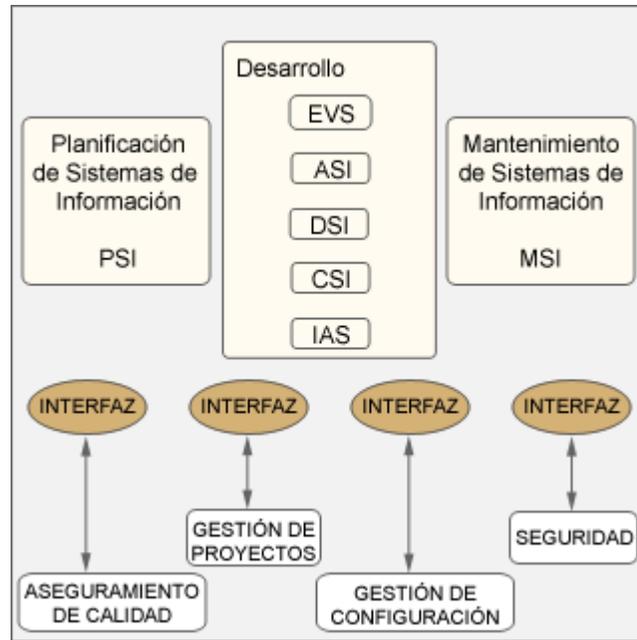


Figura 4.5.1

## 5. ESTUDIO DEL MODELO MOPROSOFT

### 5.1. ¿QUÉ ES MOPROSOFT?

---

Es un modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. El mismo fue desarrollado por la Asociación Mexicana para la Calidad en Ingeniería de Software a través de la Facultad de Ciencias de la Universidad Nacional Autónoma de México (UNAM) y a solicitud de la Secretaría de Economía para obtener una norma que resulte apropiada a las características de tamaño de la gran mayoría de empresas mexicanas de desarrollo y mantenimiento de software. MoProSoft es el nombre del modelo en la comunidad universitaria y profesional, y la norma técnica a la que da contenido es la NMX-059/01-NYCE-2005 que fue declarada Norma Mexicana el 15 de agosto de 2005.

MoProSoft considera que los modelos de evaluación y mejora CMMI e ISO/IEC 15504 no resultan apropiados para empresas pequeñas y medianas de desarrollo y mantenimiento de software. El desarrollo del modelo fue inspirado en la norma ISO/IEC 15504 y las áreas de procesos de los niveles 2 y 3 del modelo SW-CMM, así como también en estándares reconocidos internacionalmente como [SWEBOOK] y [PMBOOK]

### 5.2. HISTORIA Y OBJETIVO

---

MoProSoft debe su origen al Programa para el Desarrollo de la Industria del Software (PROSOFT). PROSOFT tiene siete líneas estratégicas, siendo la siguiente la que dio origen a MoProSoft:

"Alcanzar niveles internacionales en capacidad de procesos".

Al comenzar el desarrollo de esta línea se evaluó la adopción de los modelos: ISO 9000, ISO 15504, SW-CMM, como resultado de dicha evaluación, concluyeron lo siguiente:

"Ninguno de los estándares o modelos cumple con los requisitos expresados por la industria mejicana"

Se entiende por esto que la industria mejicana de desarrollo de software, en su gran mayoría está constituida por pequeñas o medianas empresas, y se decidió la elaboración de un modelo adecuado para las características de dichas empresas, que se basaría en los modelos evaluados. En base a esta decisión la Secretaría de Economía encargó la elaboración de dicho modelo a la Asociación Mexicana para la Calidad en Ingeniería del Software (AMCIS) en colaboración con la Universidad Nacional Autónoma de México (UNAM). La primera versión de MoProSoft se publicó en diciembre de 2002.

### 5.3. MODELO

---

El modelo que propone MoProSoft está enfocado en **procesos** y considera tres niveles básicos de la estructura de una organización en los que se agrupan los diferentes procesos:

- Alta Dirección
- Gestión
- Operación

La categoría Alta Dirección incluye el proceso:

- Gestión de Negocio.

La categoría Gestión incluye los procesos:

- Gestión de Procesos
- Gestión de Proyectos
- Gestión de Recursos (Recursos Humanos y Ambiente de Trabajo, Bienes Servicios e Infraestructura, Conocimiento de la Organización).

La categoría Operación incluye los procesos:

- Administración de Proyectos Específicos
- Desarrollo y Mantenimiento de *Software*.

### 5.4. ESQUEMA GENERAL

---

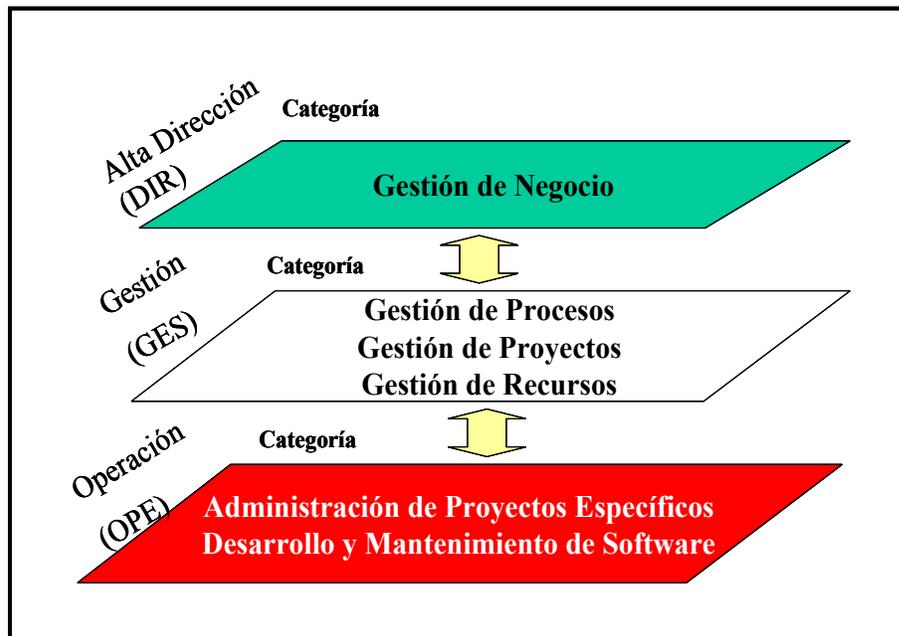


Figura 5.4.1

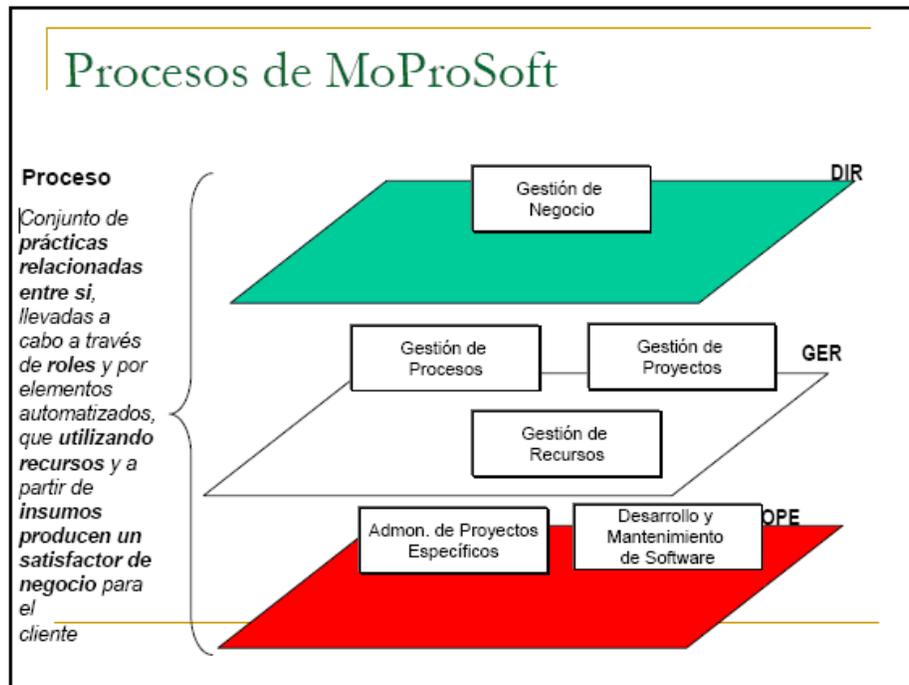


Figura 5.4.2

MoProSoft identifica los procesos empleados por las empresas de desarrollo y mantenimiento de software y los clasifica en tres categorías:

**Categoría de alta dirección (DIR)**

Aborda las prácticas de la alta dirección relativas a la gestión del negocio. Proporciona alineación a los procesos de la categoría de gerencia (GER) y se retroalimenta de la información que éstos generan.

**Categoría de Gerencia (GER)**

Aborda las prácticas de gestión de procesos, proyectos y recursos en función de las alineaciones establecidas a través de los procesos de alta dirección (DIR). Proporciona los elementos para el funcionamiento de los procesos de la siguiente categoría (Operación), recibe y evalúa la información que generan, y comunica los resultados a los procesos de alta dirección.

**Categoría de Operación (OPE)**

Aborda las prácticas para los proyectos de desarrollo y mantenimiento de software. Los procesos de esta categoría realizan las actividades de acuerdo con los elementos proporcionados por los de gerencia, y remite a ésta la información y los productos generados.

## **Patrón de procesos**

MoProSoft establece y emplea un patrón para definir cada proceso. El patrón de procesos es una agrupación esquemática de los elementos que configuran un proceso. Está formado por tres partes:

- Definición general del proceso
- Prácticas
- Guías de ajuste.

## **Definición general del proceso**

Agrupación de los elementos:

- **Proceso:** nombre del proceso, precedido por el acrónimo establecido en la definición de los elementos de la estructura del modelo de procesos.
- **Categoría:** nombre de la categoría a la que pertenece el proceso, y el acrónimo entre paréntesis.
- **Propósito:** objetivos generales medibles y resultados esperados por la implantación efectiva del proceso.
- **Descripción:** descripción general de las actividades y productos que componen el flujo de trabajo del proceso.
- **Objetivos:** objetivos específicos cuya finalidad es asegurar el cumplimiento del propósito del proceso. Los objetivos se identifican como 01, 02, etc.
- **Indicadores:** definición de los indicadores para evaluar la efectividad del cumplimiento de los objetivos del proceso. Los indicadores se identifican como I1, I2, etc. y entre paréntesis se especifica una o más identificaciones de los objetivos a los que dan respuesta.
- **Metas cuantitativas:** valor numérico o rango de satisfacción por indicador.
- **Responsabilidad y autoridad:** Responsabilidad identifica al rol principal responsable por la ejecución del proceso. Autoridad identifica al rol responsable para validar la ejecución del proceso y el cumplimiento de su propósito.
- **Subprocesos (opcional):** lista de procesos que componen al proceso definido.
- **Procesos relacionados:** nombres de los procesos relacionados

- Entradas: nombre del producto o recurso de entrada, e indicación de su origen o fuente.
- Salidas, Nombre del producto o recurso, descripción y características e indicación del destino o destinatario.
- Productos internos: nombre del producto generado y utilizado en el propio proceso, descripción y características del producto.
- Referencias bibliográficas: bibliografía que sustenta el proceso: normas, modelos de referencia, libros y otras fuentes.

### **Prácticas**

En las prácticas se identifican los roles implicados en el proceso y la formación requerida, se describen las actividades en detalle y se asocian a los objetivos del proceso, se presenta un diagrama con el flujo del trabajo, se describen las verificaciones y validaciones requeridas, se listan los productos que se incorporan a la base de conocimiento, se identifican los recursos de infraestructura necesarios para apoyar las actividades, se establecen las mediciones del proceso, así como las prácticas para la capacitación, manejo de situaciones excepcionales y uso de las lecciones aprendidas.

### **Guías de ajuste**

En las Guías de ajuste se sugieren modificaciones al proceso pero que no afectan a los objetivos del mismo.

### **Procesos**

#### **Categoría alta dirección (DIR)**

- Gestión de negocio

Establecer la razón de ser de la organización, sus objetivos y las condiciones para lograrlos, para lo cual es necesario considerar las necesidades de los clientes, así como evaluar los resultados para poder proponer cambios que permitan la mejora continua. Adicionalmente habilita a la organización para responder a un ambiente de cambio y a sus miembros para trabajar en función de los objetivos establecidos.

#### **Categoría Gerencia (GER)**

- Gestión de procesos

Establecer los procesos de la organización, en función de los procesos requeridos identificados en el Plan Estratégico. Así como definir, planificar e implantar las actividades de mejora en los mismos.

- Gestión de proyectos

Asegurar que los proyectos contribuyan al cumplimiento de los objetivos y estrategias de la organización.

- Gestión de recursos

Conseguir y dotar a la organización de los recursos humanos, infraestructura, ambiente de trabajo y proveedores, así como crear y mantener la Base de Conocimiento de la organización. La finalidad es apoyar el cumplimiento de los objetivos del Plan Estratégico de la organización. Las actividades de este proceso se apoyan en tres subprocesos:

- Recursos humanos y ambiente de trabajo

El propósito de Recursos Humanos y Ambiente de Trabajo es proporcionar los recursos humanos adecuados para cumplir las responsabilidades asignadas a los roles dentro de la organización, así como la evaluación del ambiente de trabajo.

- Bienes , servicios e infraestructura

El propósito de Bienes, Servicios e Infraestructura es proporcionar proveedores de bienes, servicios e infraestructura que satisfagan los requisitos de adquisición de los procesos y proyectos.

- Conocimiento de la organización

El propósito de Conocimiento de la Organización es mantener disponible y administrar la base de conocimiento que contiene la información y los productos generados por la organización.

### **Categoría Operación (OPE)**

- Administración de proyectos específicos

Establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados.

- Desarrollo y mantenimiento de software

Realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software nuevos o modificados cumpliendo con los requerimientos especificados.

En la figura 5.4.3 se puede observar un diagrama de relación entre los procesos.

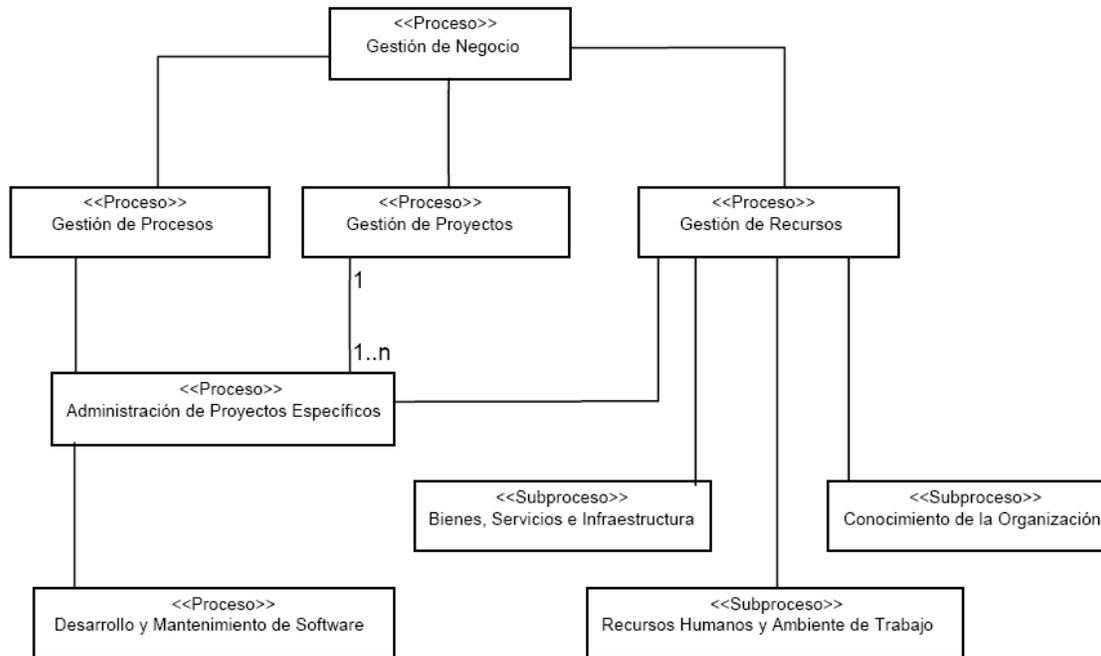


Figura 5.4.3

## 5.5. MOPROSOFT E ISO/IEC 29110

### Introducción

Como se mencionó anteriormente, las PyMEs y micro PyMEs de software desean mejorar la calidad de sus procesos de desarrollo, mediante la implementación de modelos de mejora de procesos.

Dado que la mayoría de los modelos existentes para las PyMEs carecen de un aval internacional, ISO decide crear el grupo de trabajo denominado SC7-WG24 con el objetivo de que sus estándares para la mejora de procesos de software sean más accesibles a este tipo de empresas.

## **Origen de ISO/IEC 29110 y relación con MoProSoft**

En el año 2005 ISO crea un grupo de trabajo denominado SC7-WG24, para definir procesos de software centrándose en las VSEs (Very Small Enterprises), con el objetivo principal de estructurar el futuro estándar ISO para la mejora de procesos en pequeñas empresas: el ISO/IEC 29110 for VSE. Dicho estándar debe establecer un marco común para describir perfiles (conjunto de procesos para ayudar a aplicar una norma ISO) evaluables del ciclo de vida software en las pequeñas empresas.

En una reunión del grupo ISO WG24 llevada a cabo en mayo del año 2006, con la participación de varios países entre los que se destacan USA, India y México se decide tomar la norma mexicana MoProSoft como base para su trabajo.

En octubre del mismo año se entrega al grupo ISO WG24 la traducción al inglés de la norma mexicana NMX-I-059-NYCE, seleccionando como Perfil Básico de procesos: Administración de Proyectos Específicos Desarrollo y Mantenimiento de Software

Durante el 2007 fueron generadas las primeras versiones del futuro estándar ISO para la mejora de procesos en pequeñas empresas, el ISO/IEC 29110 for VSE (Very Small Enterprise).

En las siguientes secciones, se presenta una visión general del mismo.

### **Alcance de ISO/IEC 29110**

- Proporcionar a las VSEs un reconocimiento como productoras de software de calidad a costes reducidos.
- Elaborar guías fáciles de entender, asequibles y utilizables por VSEs.
- Producir un conjunto de perfiles que construyan o mejoren procesos.
- Permitir perfiles y niveles para dominios específicos.
- Dar ejemplos para fomentar a las VSEs la adopción y seguimiento de los procesos.
- Ayudar a que las VSEs puedan trabajar y puedan ser evaluadas en forma conjunta.
- Elaborar perfiles y guías conformes con ISO/IEC 12207, 15504 y/o ISO 9001:2000

### **Estructura de ISO/IEC 29110**

En la figura 5.5.1 se puede observar la estructura que presenta el estándar ISO/IEC 29110 determinada por el SC7-WG24.

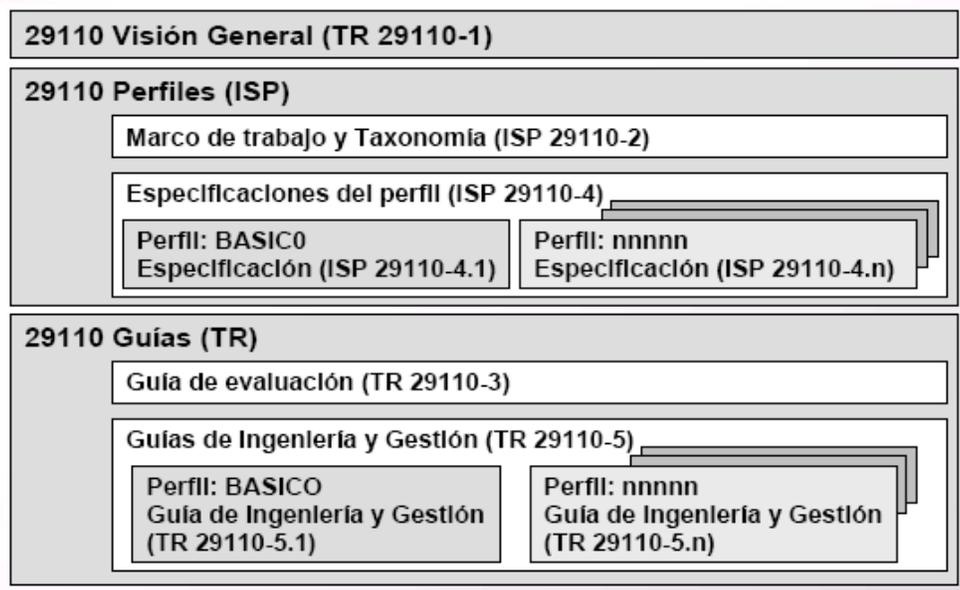


Figura 5.5.1

A continuación, se comentan cada uno de los documentos que componen el estándar, mostrado en la figura anterior.

### Visión General

Este informe incluye los conceptos principales necesarios para comprender y utilizar los documentos de ISO/IEC 29110.

### Perfiles

Los perfiles se definen con el propósito de empaquetar referencias a y/o partes de otros documentos de manera formal, con el fin de adaptarlos a las necesidades y características de las VSE. Preparar un perfil implica producir dos tipos de documentos:

- Marco de trabajo y taxonomía (TR29110-2): Especifica los elementos comunes a todos los perfiles (estructura, conformidad, evaluación) e introduce la taxonomía (catálogo) de los perfiles ISO/IEC 29110.
- Especificaciones de perfil (TR29110-4): Proporciona la composición definitiva de un perfil, los enlaces normativos al subconjunto normativo de estándares usados en el perfil, y los enlaces informativos (referencias) a documentos de “entrada”. Para cada perfil existe un documento de este tipo. Un ejemplo de una especificación de perfil es el documento 29110-4.1 (Especificación - Perfil Básico). Su objetivo es definir una guía de gestión de proyectos y desarrollo de software, adaptada a las necesidades de las VSE, para un subconjunto de procesos de ISO/IEC 12207.

## **Guías**

Las guías contienen directrices de aplicación sobre cómo realizar los procesos para alcanzar los niveles de madurez. Se desarrollan para la implantación de los procesos y para la evaluación.

- Guías de evaluación (TR29110-3): describe el proceso a seguir para realizar una evaluación que determine las capacidades de proceso y la madurez organizacional.
- Guías de ingeniería de gestión (TR29110-5): Las guías de ingeniería y gestión proporcionan orientación sobre la implementación y uso de un perfil. Para cada perfil, existe una guía de este tipo.  
Un ejemplo de una guía de ingeniería y gestión es el documento 29110-5.1 (Guía de ingeniería y gestión – Perfil básico). Este perfil se compone de dos procesos: Gestión de Proyectos e Implementación de Software.

## **Ubicación de MoProSoft en la estructura ISO/IEC 29110**

Los procesos de la categoría de Operación de MoProSoft, están explotados dentro de la estructura que presenta el estándar ISO/IEC 29110 en la parte 5.1 ( Gestión de Proyectos e Implementación de Software).

## **6. ANÁLISIS Y DESARROLLO DE LA HERRAMIENTA WEB**

### **6.1. SOLUCIÓN PROPUESTA**

---

Para el soporte del proceso de recolección y evaluación de datos de las organizaciones se desarrolló una Herramienta Web lo suficientemente genérica como para soportar distintos aspectos a evaluar (Por ejemplo: Categoría — Proceso - Subproceso)

### **6.2. ALCANCE**

---

La herramienta consta de funcionalidad dividida en dos perfiles bien diferenciados, que en conjunto sirven para cumplir con los objetivos propuestos.

Por un lado se provee de la funcionalidad correspondiente a un usuario “Final” el cual a través de la herramienta contestará un cuestionario a modo de relevamiento de la situación de la organización.

Por otro la funcionalidad correspondiente a un usuario “Administrador”, el cual es el encargado básicamente de realizar la confección de los cuestionarios que posteriormente serán usados por los usuarios finales.

### **6.3. SEGURIDAD**

---

El sistema cuenta con un perfil de seguridad, el cual permite dar el soporte necesario para brindar seguridad a nivel de aplicación, mediante roles, y permisos, así como seguridad en las comunicaciones.

#### **6.3.1. Autenticación de usuario**

---

El contexto de usuario en una aplicación describe las características y perfil del usuario junto con los permisos que este posee para realizar diversas tareas en una aplicación. Dado que los servicios se ejecutan en un servidor de aplicación y serán invocados por clientes ubicados en otros servidores o HOST, se requiere un mecanismo para asociar cada requerimiento a un servicio con una sesión previamente establecida con el servidor durante el proceso de autenticación. Para esto la mejor alternativa es la utilización de la sesión del servidor de aplicaciones. Este mecanismo es posible utilizarlo tanto para aplicaciones clientes WEB como para aplicaciones ricas (STANDALONE) o aplicaciones externas.

#### **6.3.2. Administración de usuarios**

---

Se provee de la funcionalidad necesaria para la administración de usuarios, creación modificación y eliminación de los mismos.

### **6.3.3. Administración de permisos**

---

Permite dado un usuario existente, la asignación de permisos sobre cierta funcionalidad.

El sistema tiene la particularidad que el menú se construye en base a los permisos sobre la funcionalidad que fueron otorgados al usuario.

## **6.4. CARACTERÍSTICAS TÉCNICAS**

---

### **6.4.1. Arquitectura**

---

La herramienta de evaluación es un componente WEB, el cual brinda la posibilidad de acceso desde cualquier punto de la intranet o inclusive, internet. Siendo esta la única precondition tecnológica, se optó por una arquitectura de 3 “capas Web” (WebClient / WebServer / DBServer). El WebClient puede ser cualquier navegador de internet (WebBrowser), como Microsoft Internet Explorer, Mozilla Firefox, NetScape Navigator, etc.

Basándose en el experticement de los desarrolladores sobre tecnologías OpenSource y el lenguaje de programación Java, se decidió usar como contenedor Web (“WebServer”) Tomcat v5.0 (Apache Software Foundation), quizás el contenedor Web no-comercial más utilizado del mundo. Esta elección brinda la posibilidad de instalarse en cualquier plataforma de SO. Sobre él se programó la interfaz de usuario (páginas jsp, servlets y componentes javascript) y la lógica del negocio (componentes Java). El mapeo de la lógica de negocios sobre el modelo de datos se hace a través del framework de persistencia Hibernate v3.0.

Como DBServer se optó por MySQL 4.1, un motor liviano y simple que cumple con creces las necesidades técnicas del proyecto.

### **6.4.2. Base de Datos**

---

El modelo de datos es un modelo relacional separado en Modelo Lógico y Modelo Físico. A su vez cuenta con un submódulo no-relacional que representa el estadio temporal de los objetos mientras están en curso las operaciones del sistema.

## **6.5. FUNCIONALIDAD BÁSICA**

---

El sistema cuenta con dos perfiles claramente divididos: el de Administrador y el de Usuario. El Administrador puede crear, listar, modificar y eliminar cuestionarios. Sobre los cuestionarios puede listar, agregar, modificar y eliminar preguntas, como también asignarle una categoría de evaluación. Para cada pregunta puede listar, crear, modificar y eliminar las posibles respuestas, como también asignarle un nivel

específico de certificación. Para cada respuesta es posible definir su contenido, dominio, importancia y asociarla con la pregunta desencadenante.

Para los Usuarios las funcionalidades cubiertas son las de listar, responder o consultar los cuestionarios que tenga asociados.

Para ambos perfiles se da la posibilidad de realizar impresiones de cuestionarios.

## **7. DESCRIPCIÓN FUNCIONAL**

La siguiente sección muestra un detalle de alto nivel de la funcionalidad que provee la herramienta desarrollada. Se encuentra dividido en dos grupos. Por un lado la funcionalidad provista para el usuario que realiza una evaluación. Por otro la funcionalidad correspondiente al administrador del sistema.

### **7.1. PERFIL USUARIO FINAL**

---

#### **7.1.1. Realizar una evaluación.**

---

Esta funcionalidad incluye, realizar una evaluación de un cuestionario determinado, desde el principio o poder continuar una evaluación parcialmente completada.

#### **7.1.2. Obtener conclusiones sobre las evaluaciones.**

---

El sistema le brinda al usuario final, la posibilidad de ver ciertas conclusiones sobre la evaluación realizada.

#### **7.1.3. Generar reportes.**

---

Esta funcionalidad permite al usuario final, poder ver reportes en pantalla e impresos, de los detalles de evaluaciones realizadas y de las conclusiones obtenidas de las mismas.

#### **7.1.4. Consultar evaluaciones históricas.**

---

El sistema brinda la posibilidad al usuario final de poder consultar evaluaciones anteriormente realizadas.

### **7.2. PERFIL ADMINISTRADOR.**

---

#### **7.2.1. Administrar cuestionarios.**

---

Esta funcionalidad permite al usuario administrador, tanto la creación de un cuestionario nuevo como la modificación de uno existente.

### **7.2.2. Administrar preguntas.**

---

Esta funcionalidad hace referencia a la creación de preguntas nuevas dentro de un cuestionario, a la modificación de cualquier pregunta existente y al borrado de las mismas.

### **7.2.3. Administrar respuestas.**

---

El sistema provee la posibilidad que el administrador cree tipos de respuestas nuevas para una pregunta determinada.

### **7.2.4. Búsquedas sobre el sistema.**

---

Esta funcionalidad permite al usuario administrador realizar búsquedas tanto de cuestionarios existentes como de preguntas dentro de un cuestionario determinado.

## **8. ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE**

La siguiente sección muestra los requerimientos que satisface el componente, en un nivel de detalle más bajo que el utilizado en el capítulo de **descripción funcional**. Al igual que el capítulo anterior, se encuentra dividido en dos grupos. Por un lado los requerimientos satisfechos para el usuario que realiza una evaluación. Por otro los requerimientos satisfechos para el usuario administrador del sistema.

### **8.1. REQUERIMIENTOS MÓDULO USUARIO FINAL.**

---

#### **8.1.1. Listar cuestionarios.**

---

Realizar la búsqueda de cuestionarios existentes en el sistema.

#### **8.1.2. Listar evaluaciones.**

---

Realizar la búsqueda de evaluaciones parcial o totalmente terminadas, existentes en el sistema.

#### **8.1.3. Realizar evaluación.**

---

Realizar la evaluación de un cuestionario determinado.

#### **8.1.4. Continuar una evaluación.**

---

Continuar realizando la evaluación de un cuestionario determinado que se encuentra parcialmente completado.

#### **8.1.5. Guardar evaluación parcialmente terminada.**

---

Guardar una evaluación que se encuentra parcialmente terminada para su posterior continuación.

#### **8.1.6. Ver detalle de evaluación.**

---

Ver el detalle completo de una evaluación que se encuentra totalmente terminada.

#### **8.1.7. Imprimir detalle de evaluación.**

---

Dar la posibilidad mediante el sistema de generar la impresión del detalle completo de una evaluación totalmente terminada.

#### **8.1.8. Ver conclusiones de evaluación.**

---

Dar la posibilidad de ver un detalle completo de las conclusiones que brinda el sistema sobre una evaluación que se encuentra totalmente terminada.

#### **8.1.9. Imprimir conclusiones de evaluación.**

---

Dar la posibilidad de generar la impresión del detalle completo de las conclusiones que brinda el sistema sobre una evaluación totalmente terminada.

## **8.2. REQUERIMIENTOS MÓDULO ADMINISTRADOR.**

---

### **8.2.1. Dar de alta un cuestionario.**

---

Dar de alta un nuevo cuestionario en el sistema.

El alta es lo suficientemente genérica como para soportar distintos aspectos a evaluar como ser: Categoría – Subcategoría – Proceso

Los datos de entrada necesarios para dar de alta un nuevo cuestionario son:

- **Nombre.**
- **Categoría:** Este dato representa la categoría para la cual se definirá el cuestionario. Por ejemplo: “Categoría de Gestión”.
- **Proceso:** Este dato representa dentro de la categoría para la cual se va a definir el cuestionario, a que proceso pertenece. Por ejemplo: “Gestión de proyectos”.
- **Subproceso:** Este dato representa cual es el subproceso, si es que existe, al que pertenece el cuestionario. Por ejemplo: “Administración de proyectos específicos”.
- **Preguntas:** Preguntas correspondientes al cuestionario. La especificación de las preguntas se hará en un requerimiento posterior.

### **8.2.2. Dar de baja un cuestionario.**

---

Posibilidad de dar de baja cualquier cuestionario existente en el sistema dejando de estar disponible para nuevas evaluaciones.

Para el caso que existan evaluaciones parcialmente completadas correspondientes a un cuestionario a eliminar, se debe proveer un mecanismo que permita por parte del usuario final de la aplicación, la terminación de la evaluación.

### **8.2.3. Modificar un cuestionario.**

---

Posibilidad de modificar cualquier cuestionario existente en el sistema.

Para el caso que existan evaluaciones parcialmente completadas correspondientes a un cuestionario a modificar, se debe poder mantener para estas evaluaciones el cuestionario original.

### **8.2.4. Dar de alta una pregunta.**

---

Dar de alta preguntas dentro de un cuestionario determinado.

Se debe permitir tanto el alta de las preguntas en el momento en el cual se está creando el cuestionario como también la posibilidad de agregar preguntas a un cuestionario existente.

Los datos de entrada necesarios para dar de alta una nueva pregunta son:

- **Nivel:** este dato determina a qué nivel de certificación corresponde la pregunta.
- **Actividad:** determina a que actividad corresponde la pregunta, por ejemplo: planeación, realización, etc.
- **Texto:** representa el texto de la pregunta.
- **Nombre corto:** nombre corto con el cual se puede identificar la pregunta.
- **Ayuda contextual:** representa un texto de ayuda al usuario que está realizando la evaluación para poder contestar la pregunta.
- **Respuesta:** representa el tipo de respuesta correspondiente a esta pregunta.

#### **8.2.5. Modificar una pregunta.**

---

Dar la posibilidad de modificar cualquiera de los datos correspondientes a una pregunta de un cuestionario determinado.

Vale aclarar que la modificación de una pregunta de un cuestionario, puede afectar a las evaluaciones realizadas de ese cuestionario o a las evaluaciones parcialmente terminadas.

Estas evaluaciones no deben ser afectadas por esta modificación, deben respetar el cuestionario original.

#### **8.2.6. Dar de baja una pregunta.**

---

Posibilidad de eliminar una pregunta de un cuestionario determinado.

Vale aclarar que la eliminación de una pregunta de un cuestionario, puede afectar a las evaluaciones realizadas de ese cuestionario o a las evaluaciones parcialmente terminadas.

Estas evaluaciones no se verán afectadas por esta eliminación, deben respetar el cuestionario original.

#### **8.2.7. Dar de alta tipos de respuestas específicas.**

---

Brindar la posibilidad de dar de alta opciones de respuestas específicas que se ajusten a una pregunta determinada.

En el caso que ninguna de los tipos de respuestas que provee el sistema se adapte a una pregunta determinada, se debe proveer la posibilidad de generar respuestas nuevas que se ajusten a la pregunta en construcción.

Cada opción de respuesta tendrá que tener asociado un peso determinado el cual sirve para realizar la evaluación final.

#### **8.2.8. Buscar cuestionarios.**

---

Realizar la búsqueda por nombre de cuestionarios dentro del sistema.

### **8.2.9. Buscar preguntas.**

---

Realizar la búsqueda de preguntas dentro de un cuestionario determinado.

### **8.2.10. Generar cuestionario impreso.**

---

Dar la posibilidad de generar la impresión de un cuestionario.

### **8.2.11. ABM de datos de dominio.**

---

Realizar el alta baja y modificación de cualquiera de los datos de dominio. Los datos de dominio son los que permiten en gran medida que el sistema desarrollado sea lo suficientemente genérico para soportar distintos aspectos a evaluar. Como ejemplo de datos de dominio podemos citar: niveles de certificación a los que pueden pertenecer las preguntas, categoría, respuestas generales del tipo SI/NO, etc.

## 9. GUÍA DE USO

Si bien la tesina de grado viene acompañada de un manual de usuario de la aplicación desarrollada, en la siguiente sección, se desea mostrar un ejemplo concreto en el cual por un lado, un usuario con perfil de Administrador crea un cuestionario y los pasos que debe seguir un usuario Final para responder el mismo.

### 9.1. CREAR UN CUESTIONARIO.

---

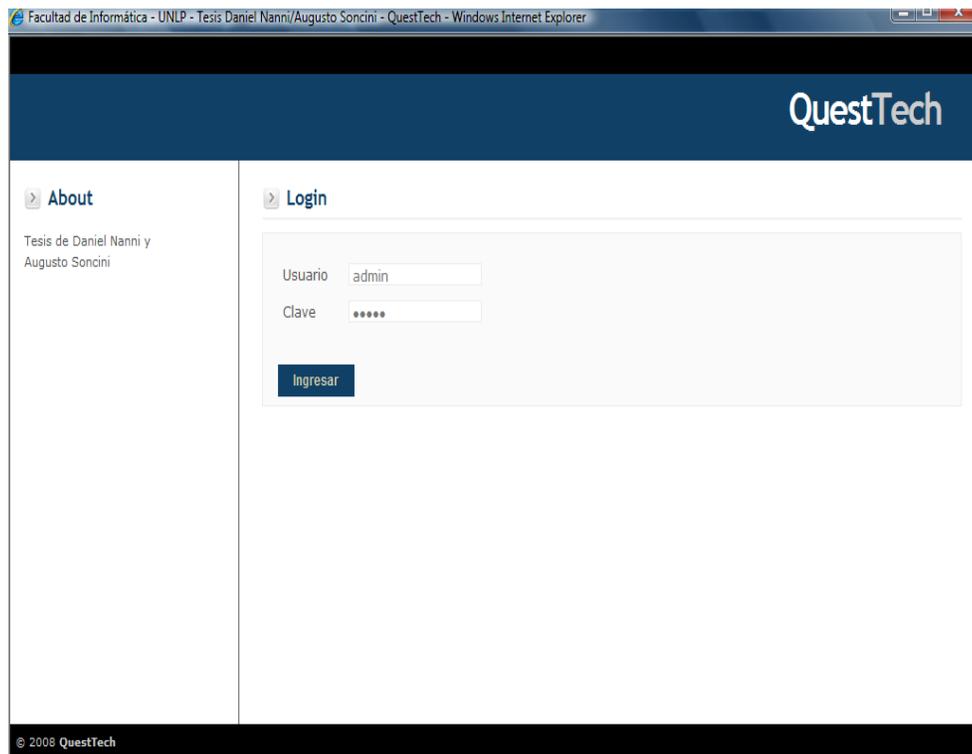
Esta funcionalidad pertenece a un usuario Administrador. El sistema es genérico dando la posibilidad de crear cualquier tipo de cuestionario.

A continuación se muestra un ejemplo de los pasos a seguir para la creación de un cuestionario.

#### 9.1.1. Ingreso Al sistema.

---

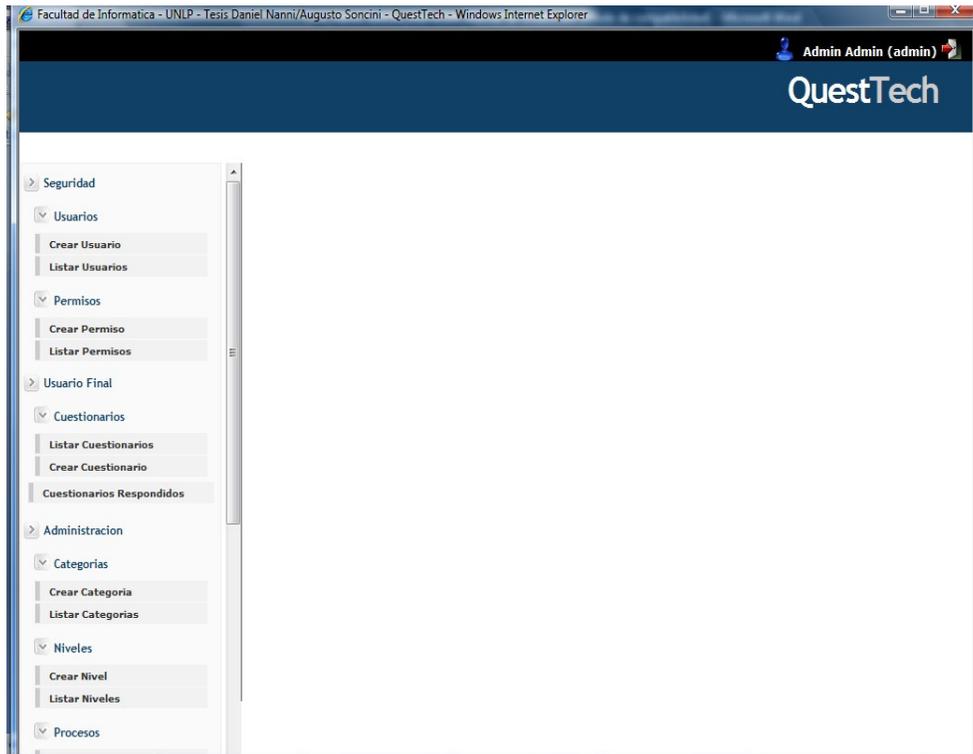
La figura siguiente muestra la ventana de ingreso al sistema.

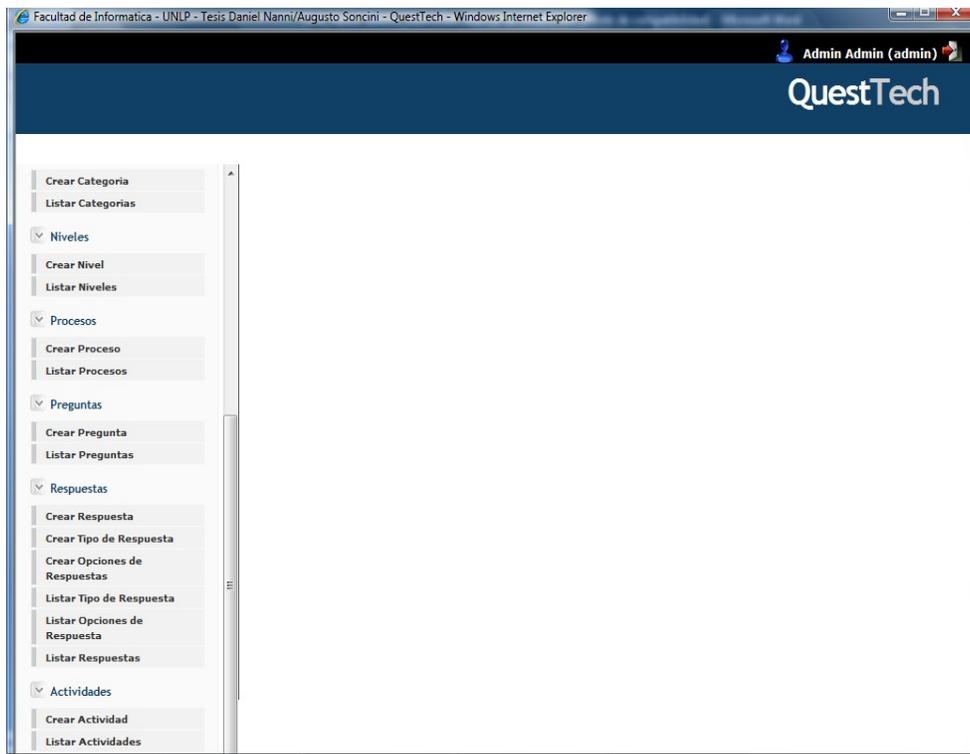


### 9.1.2. Menú principal del usuario Administrador.

---

El menú general de la aplicación se encuentra en la sección izquierda de la pantalla. Se agrupa por un nivel que engloba la funcionalidad en forma jerárquica como se muestra en las figuras.

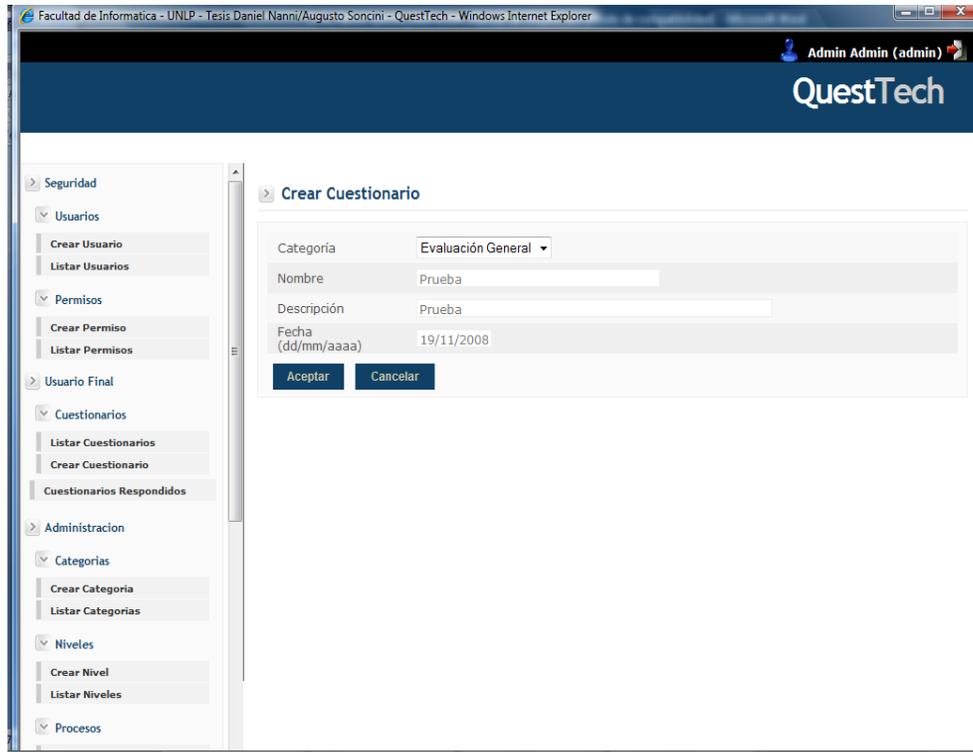




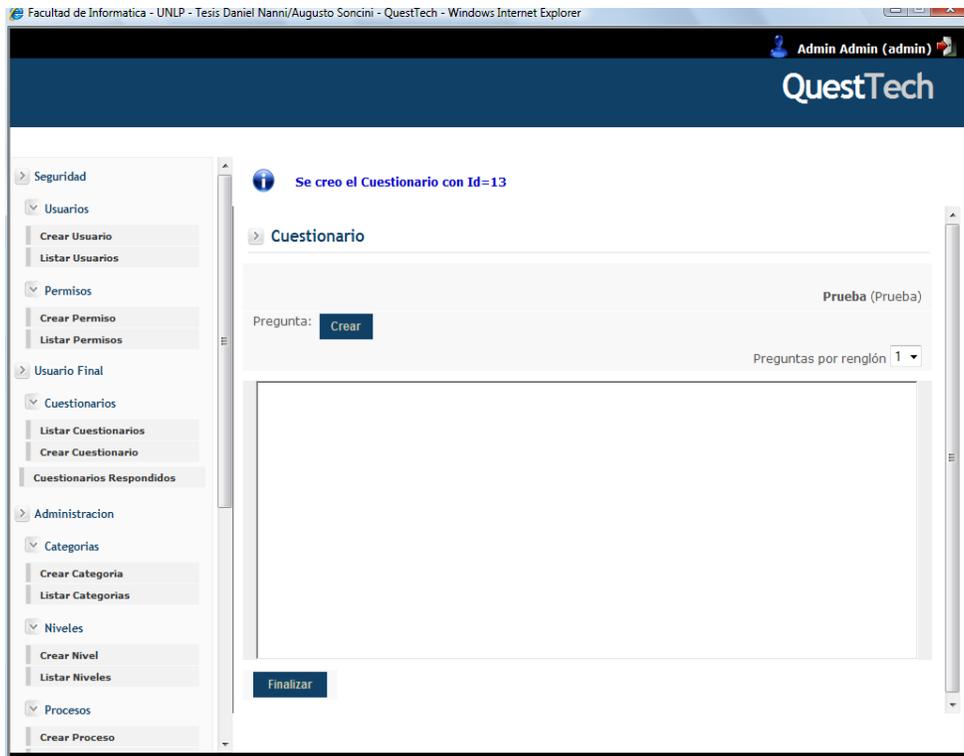
### 9.1.3. Crear cuestionario.

---

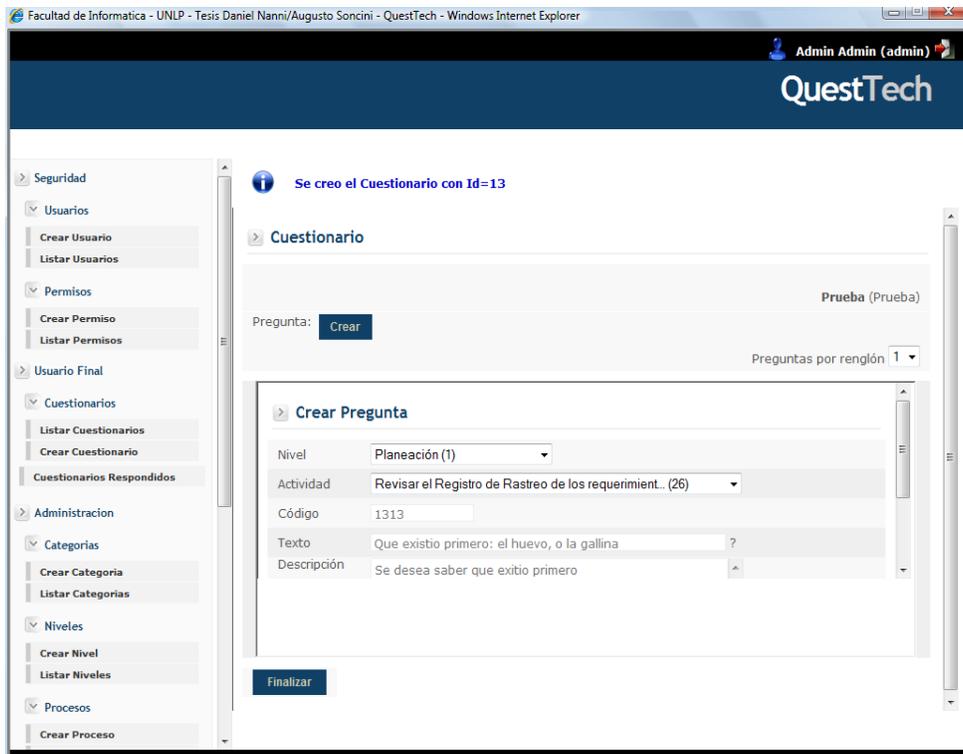
Dentro del submenú Cuestionarios, si se selecciona la opción **Crear Cuestionario**, se abre una pantalla como la de la siguiente figura.



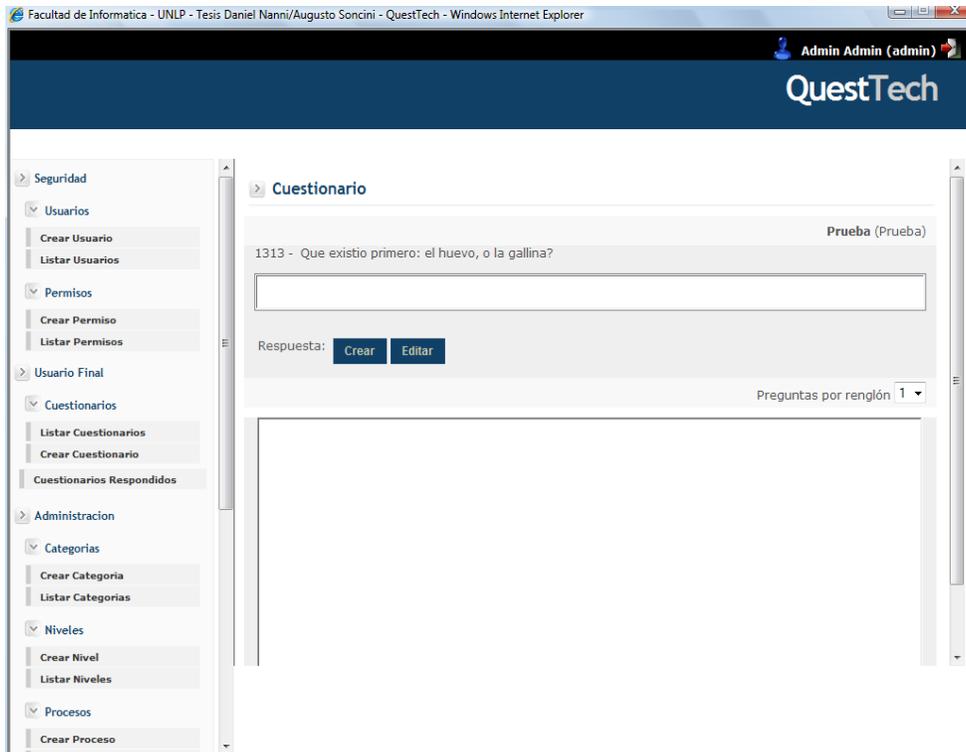
Una vez creado el Cuestionario, se cuenta con la opción para definir las preguntas y respuestas.  
Para crear la pregunta, se debe presionar el botón Crear, como se muestra en la figura siguiente.



En el panel inferior se permite definir las propiedades de la pregunta: “Nivel”, “Actividad”, “Código”, “Texto” y “Descripción”, como se muestra en la figura siguiente.

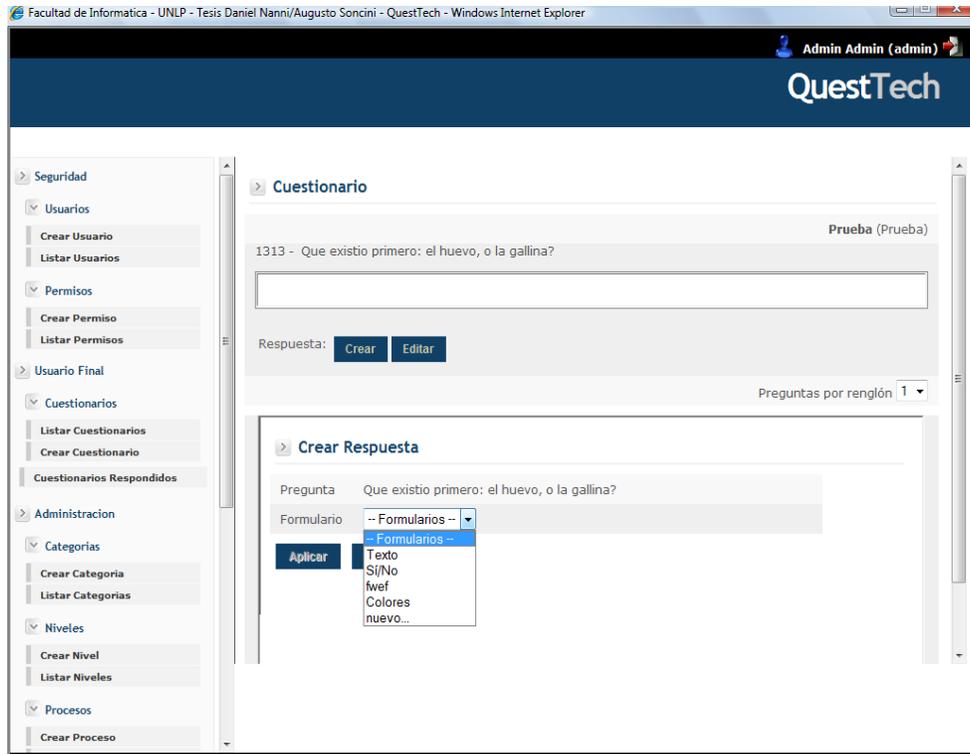


Una vez creada la pregunta, en el panel superior se la puede ver incluida en el grafo del cuestionario.  
Para cada pregunta aparecerán las opciones para crear y editar una respuesta (botones “Crear” y “Editar”).

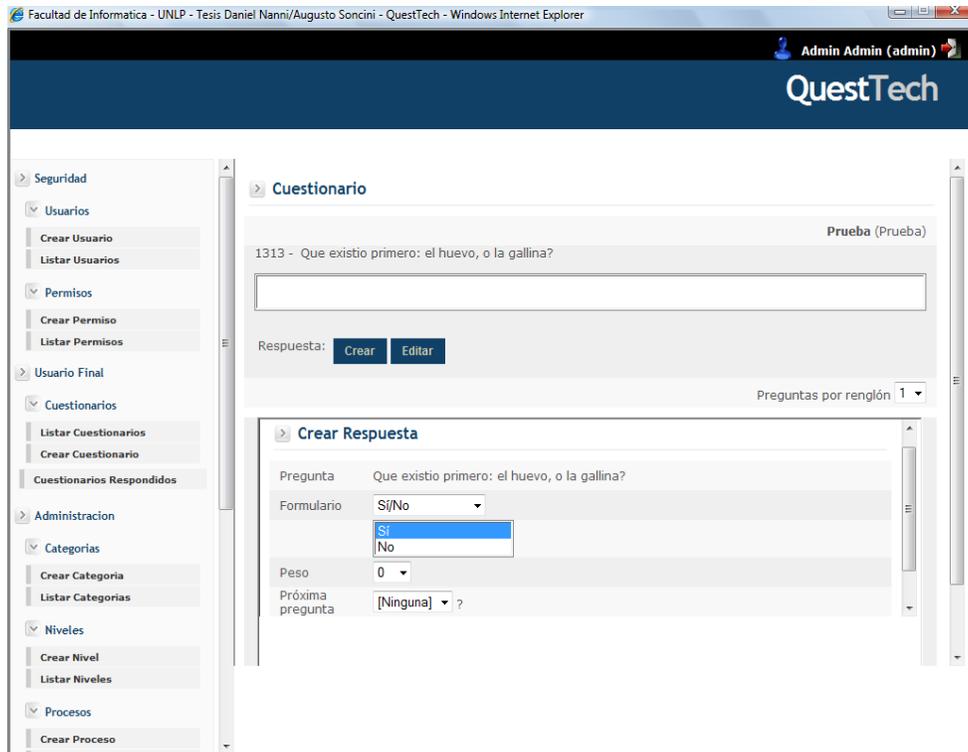


Cuando el usuario opta por “Crear Respuesta” para la Pregunta, en el panel inferior aparecen las opciones de respuestas, tipificadas por “Formularios”. Se distinguen ya algunos predefinidos”:

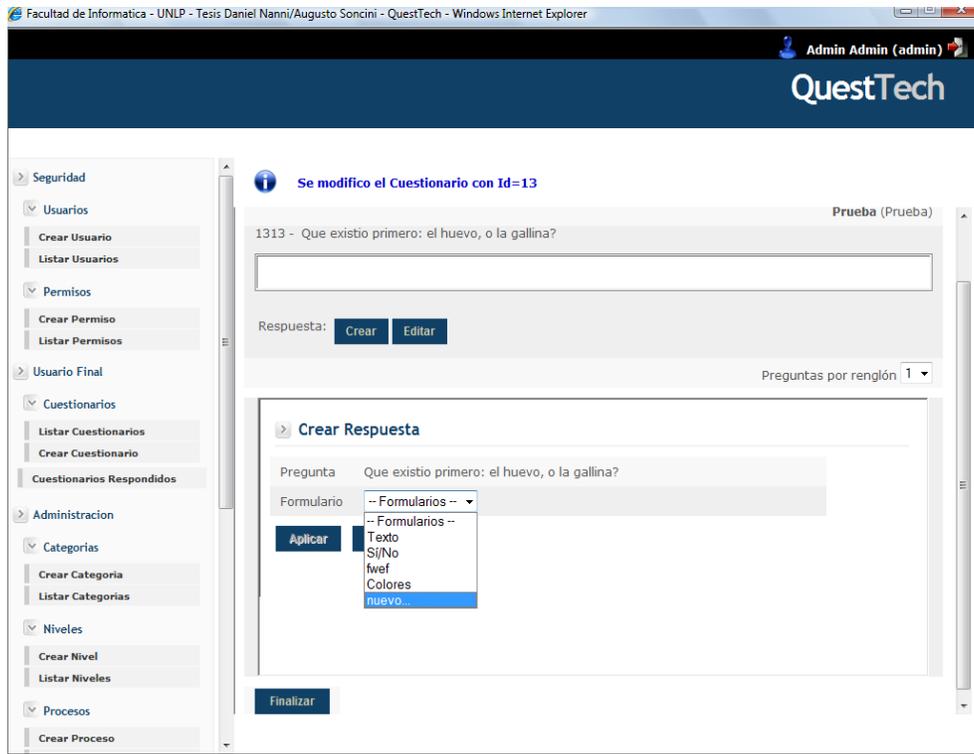
- “Texto”: que indica que la respuesta requerida es un texto a ser ingresado por el usuario
- “Sí/No”: que indica que la respuesta puede ser afirmativa o negativa
- Etc...
- “nuevo...”: que es la opción que permite crear nuevos formularios, según la necesidad de la pregunta (p.e. Colores=Rojo, Azul, Verde; Nombres=Juan, Pedro, Nicolas; etc).



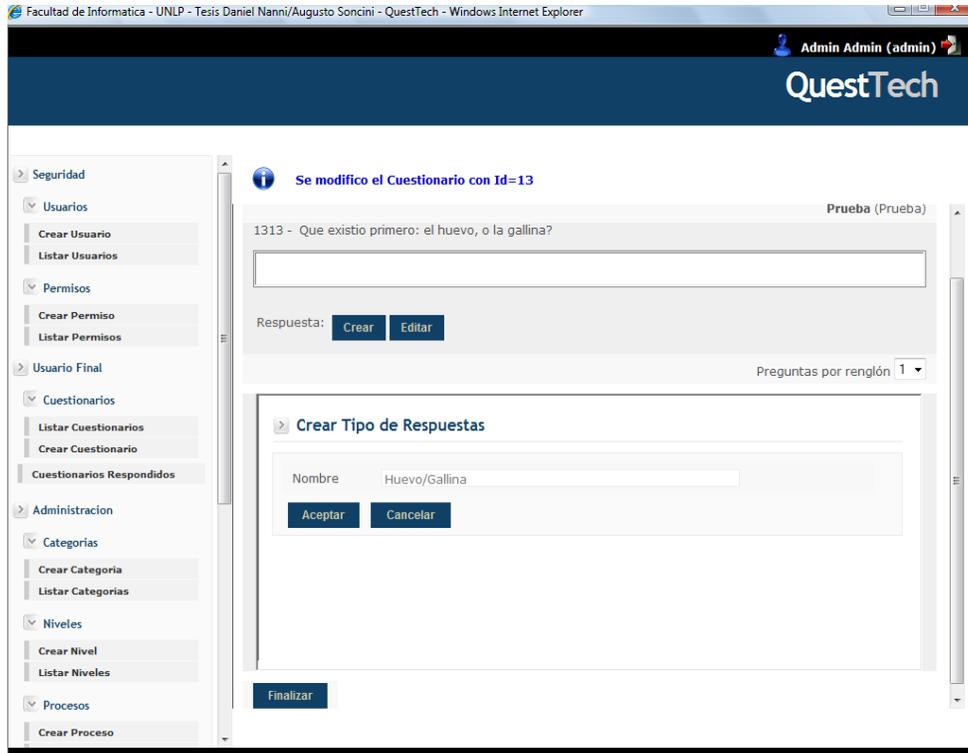
Si se opta por un tipo de respuesta de formulario, exceptuando la de tipo “Texto”, a medida que se van seleccionado las opciones, se le debe asignar un peso (valor de 0 a 10), para que se pueda hacer una evaluación una vez que el usuario final responda las preguntas, tal como se ve en la figura siguiente.



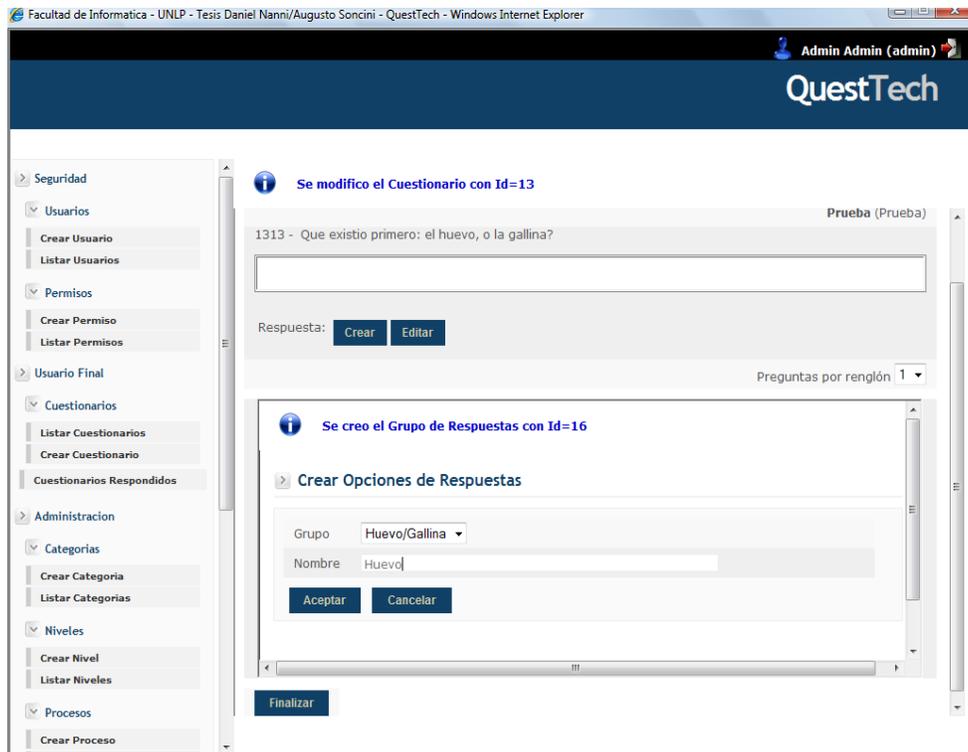
Si el formulario de respuesta que se quiere ofrecer para la pregunta no apareciera entre las opciones, está la posibilidad de crear uno (opción “nuevo...”), como se ve en la figura siguiente

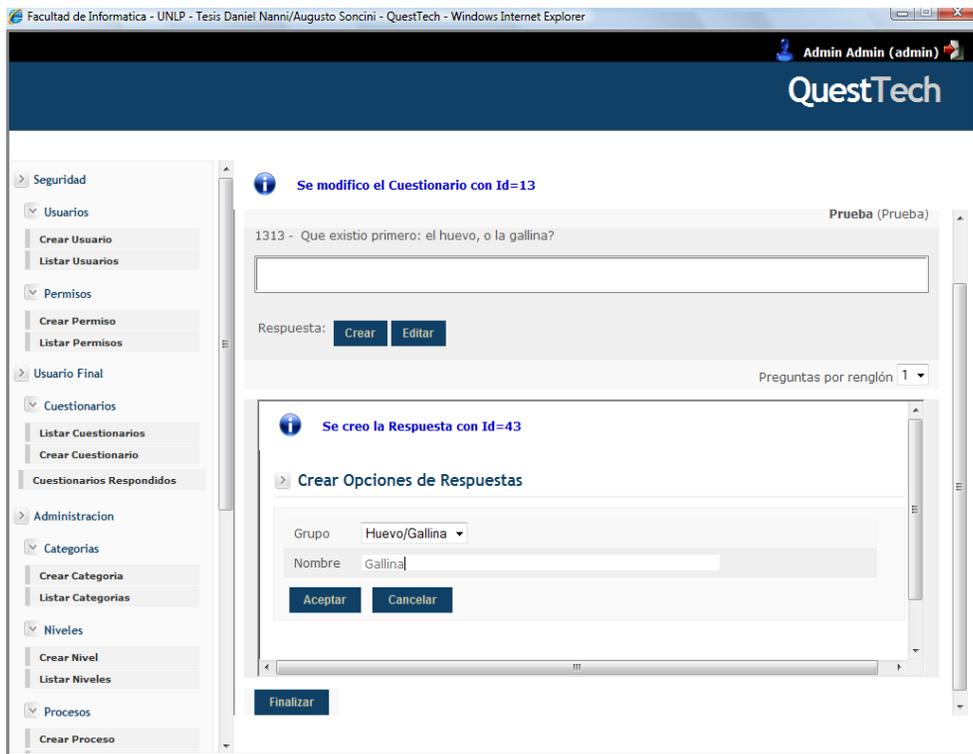


Al crear un nuevo Tipo de Respuesta, lo primero que se presenta en el panel inferior es la opción para asignarle un nombre, como se muestra en la figura siguiente

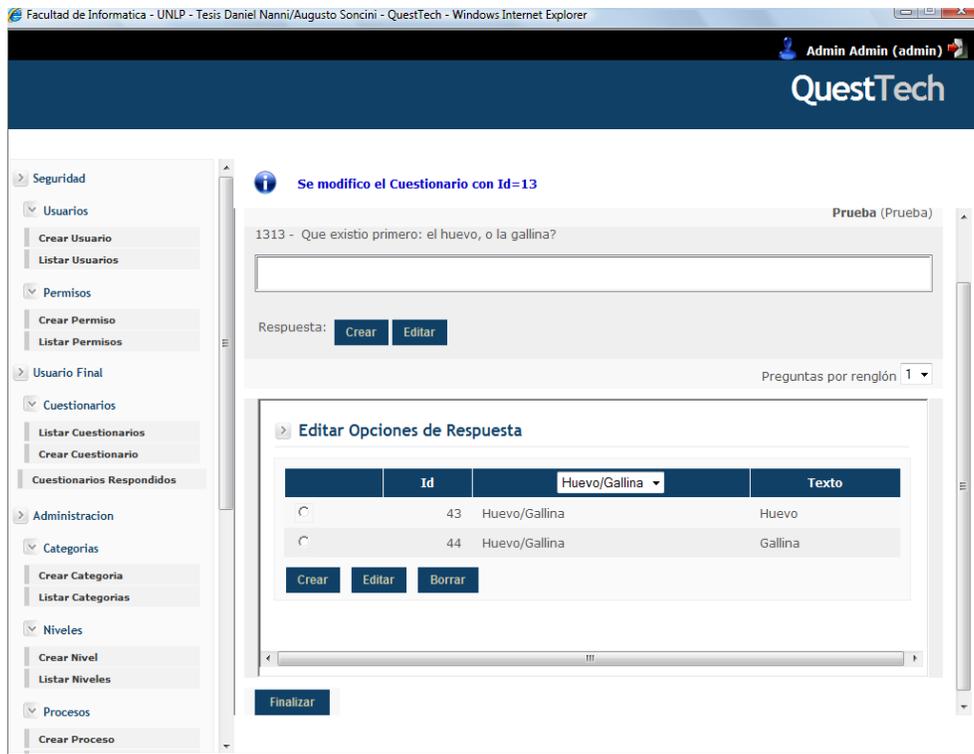


Al Tipo de Respuesta se van a poder ir asociando tantas Opciones como sean necesarias, tal como se ve en las figuras siguientes.

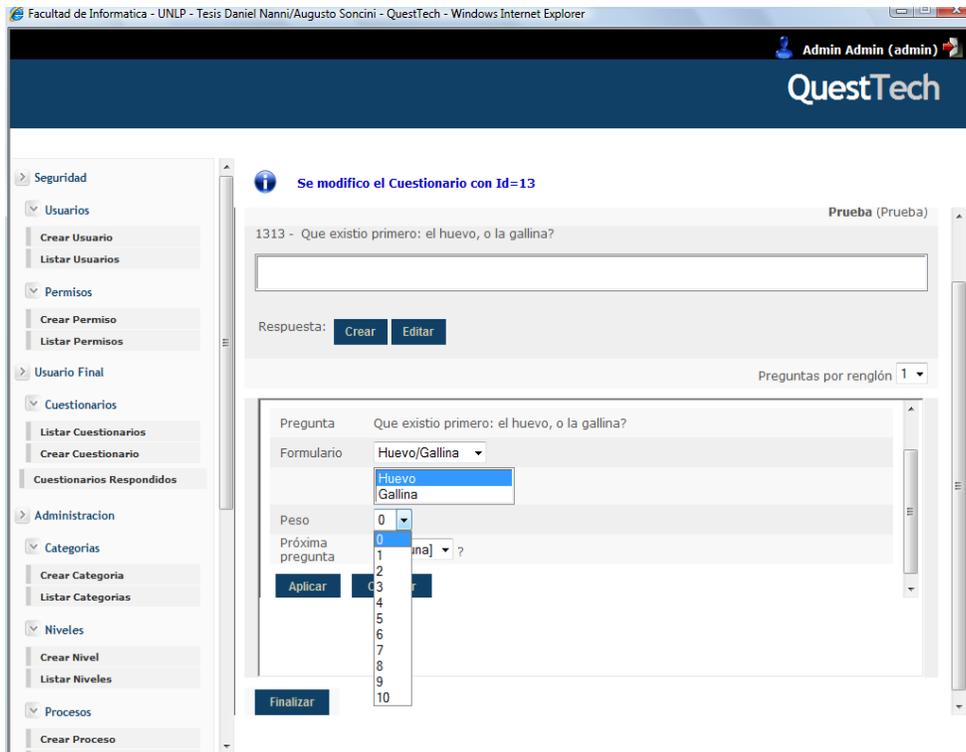




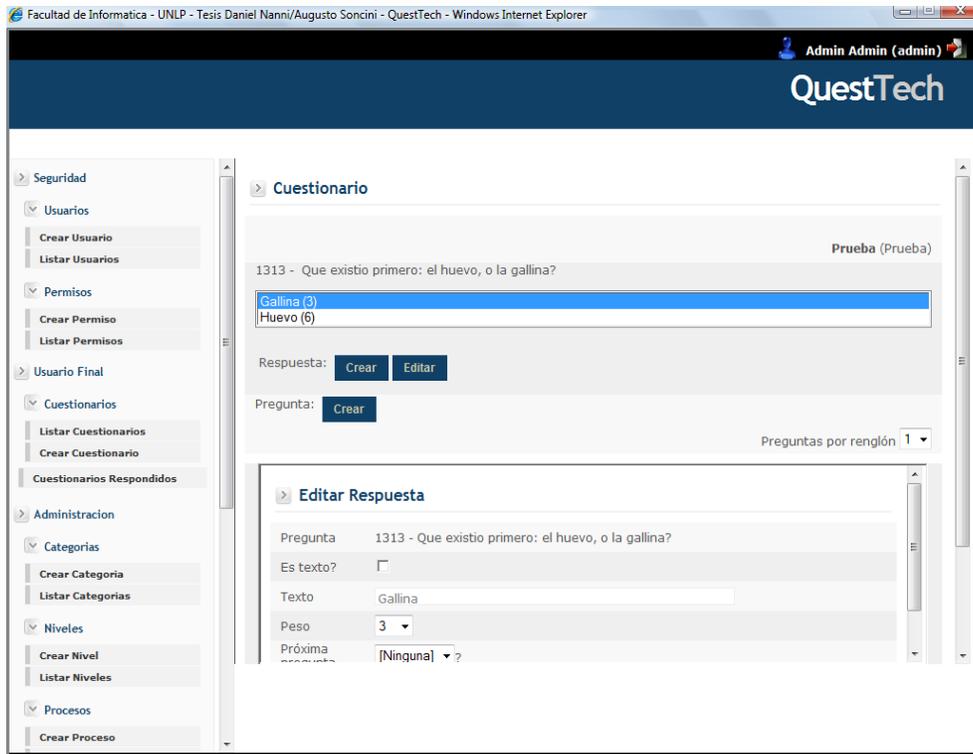
Una vez asignadas todas las opciones necesarias, estas se listan para poder seguir administrándolas, de ser necesario como muestra la figura siguiente.



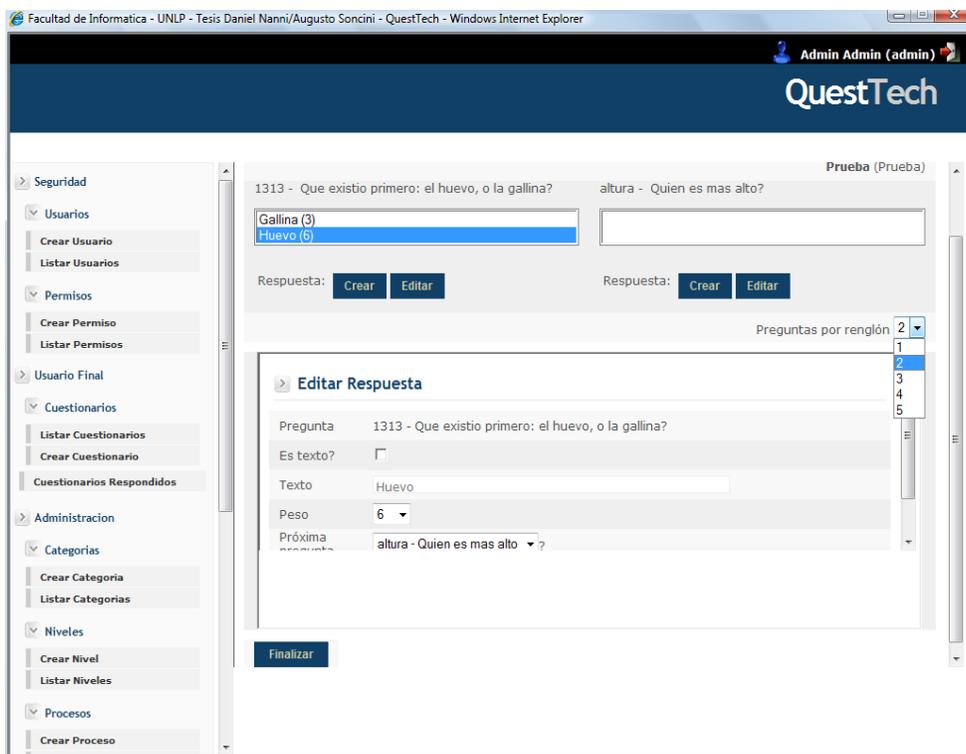
Ahora sí, para la Pregunta que se está agregando al Cuestionario, figura disponible como Formulario de respuesta el creado en los pasos previos, con sus diferentes opciones, como se ve en la figura siguiente



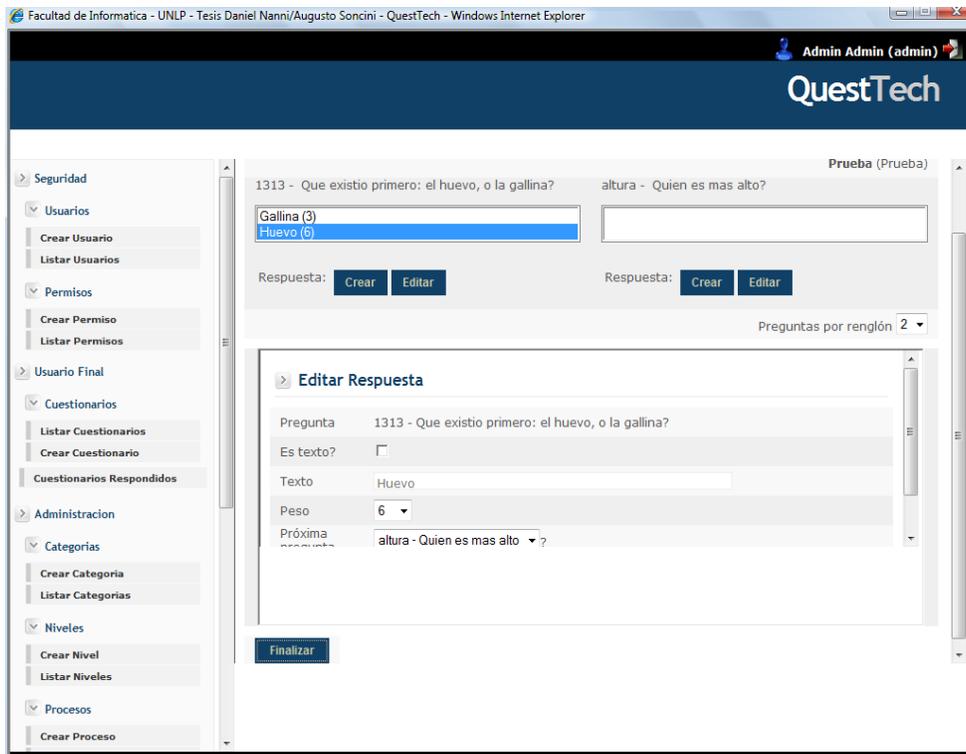
Para completar el grafo del cuestionario, se debe ir asignando las preguntas que le siguen a cada respuesta. Se selecciona la respuesta y en el panel inferior se presenta la opción de editar la misma. Además de poder modificar cualquiera de sus atributos, se puede asignar la “Próxima Pregunta”, como se ve en la figura siguiente



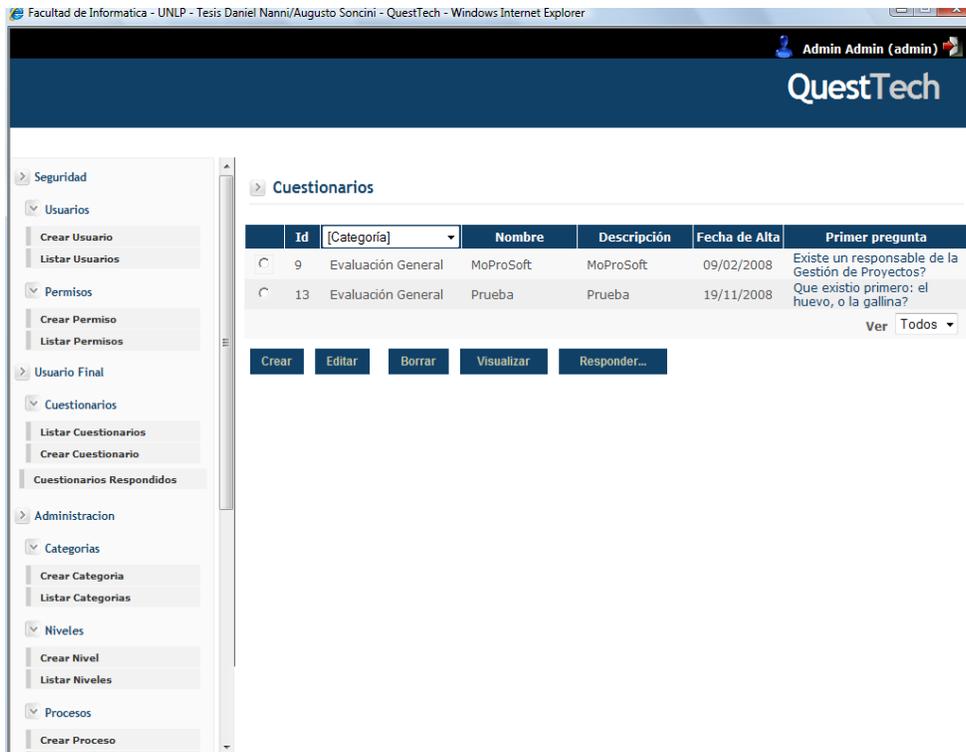
Para una presentación más cómoda del grafo del Cuestionario, se puede determinar, las “Preguntas por renglón” a ser mostradas, como se ve en la figura siguiente.



Una vez definidas todas las alternativas para el Cuestionario, presionado el botón “Finalizar” que aparece al pie izquierdo del panel inferior se da por terminado el proceso de creación, como se ve en la siguiente figura.



En el listado de Cuestionarios, ya se encontrará disponible el cuestionario creado, para su manipulación.



## 9.2. RESPONDER UN CUESTIONARIO.

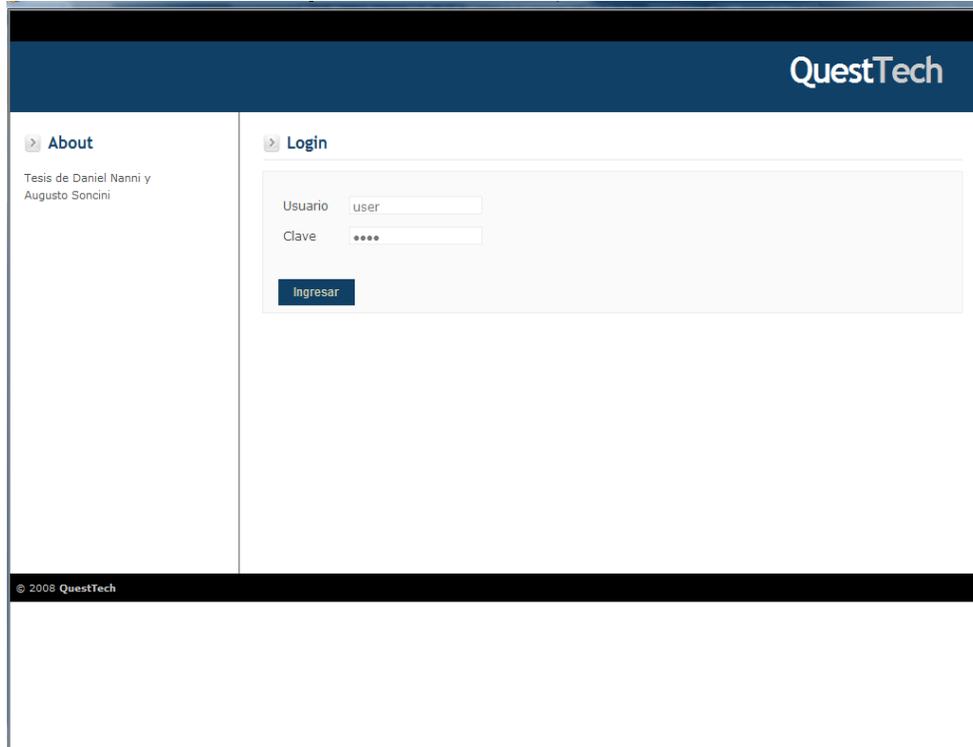
---

Esta funcionalidad es propia del usuario final. En las siguientes secciones se describe a modo de ejemplo los pasos a seguir.

### 9.2.1. Ingreso Al sistema.

---

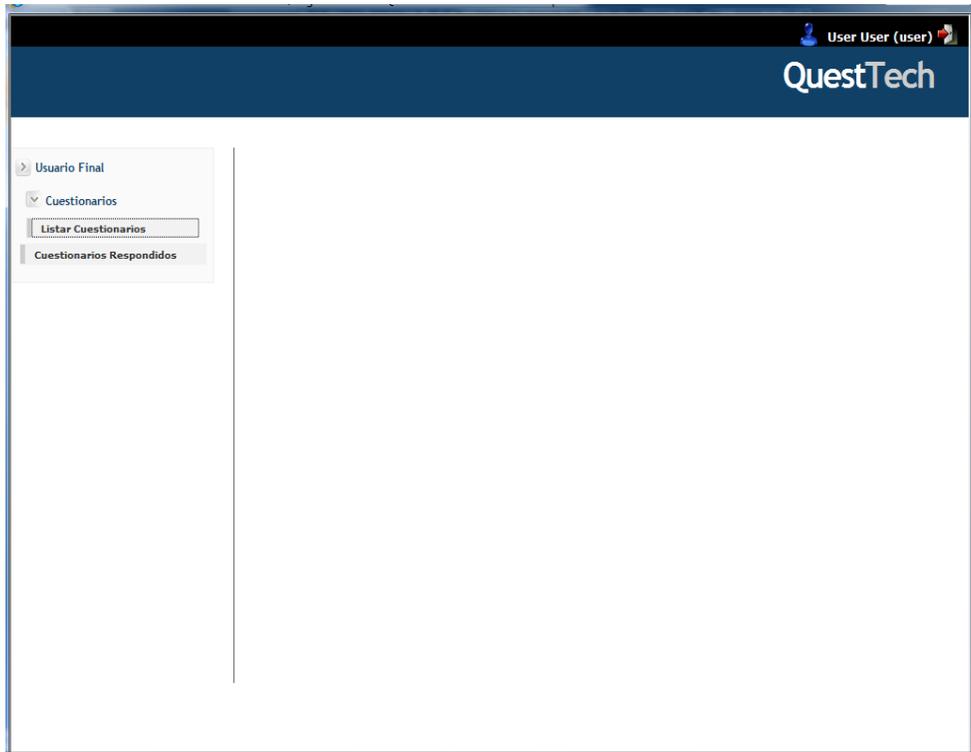
En la siguiente figura se muestra la ventana de ingreso al sistema. El usuario debe haber sido creado previamente por un usuario con perfil de administrador.



### 9.2.2. Menú principal.

---

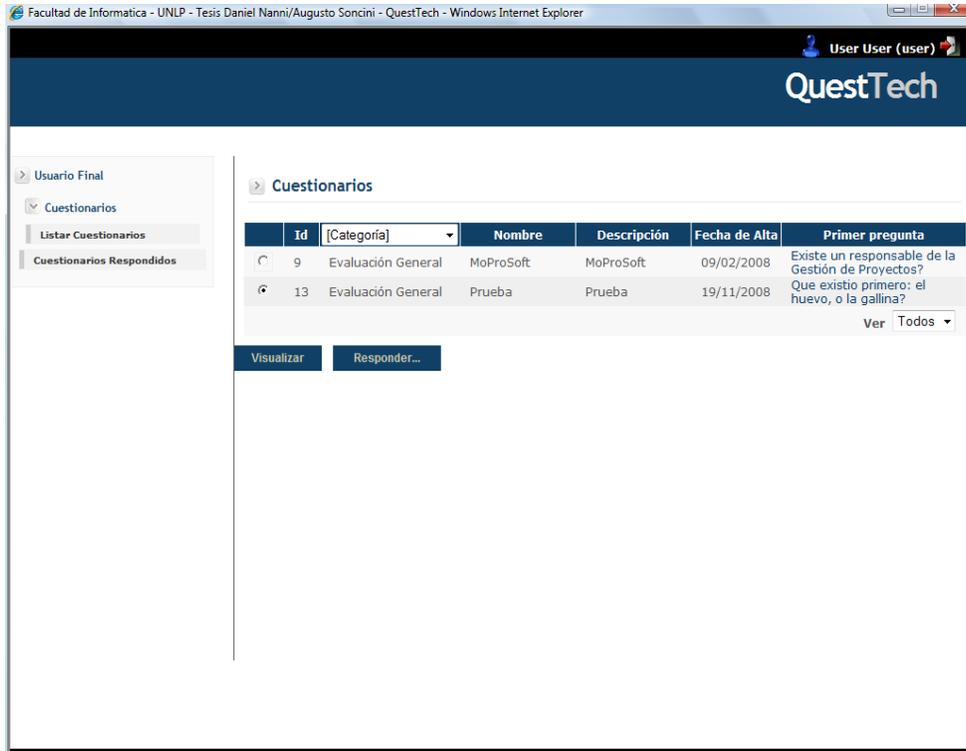
Como se ve en la figura 10.1.2, el usuario final corresponde a un usuario simple del sistema, que sólo tiene permisos para responder cuestionarios nuevos, y operar sobre sus cuestionarios respondidos.



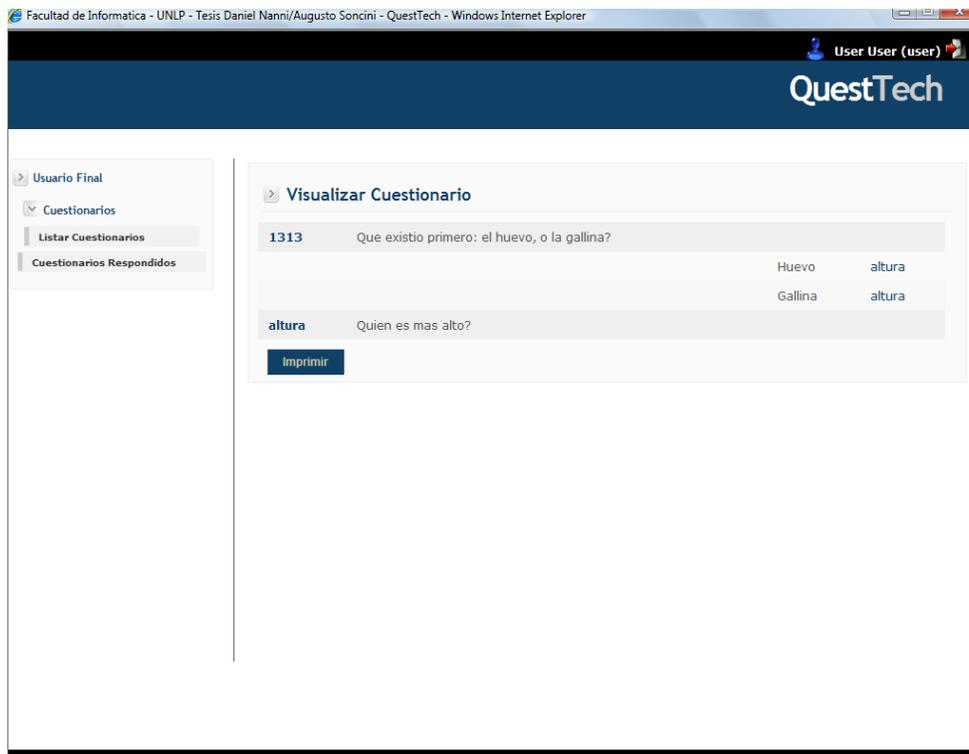
### 9.2.3. Listar Cuestionarios

---

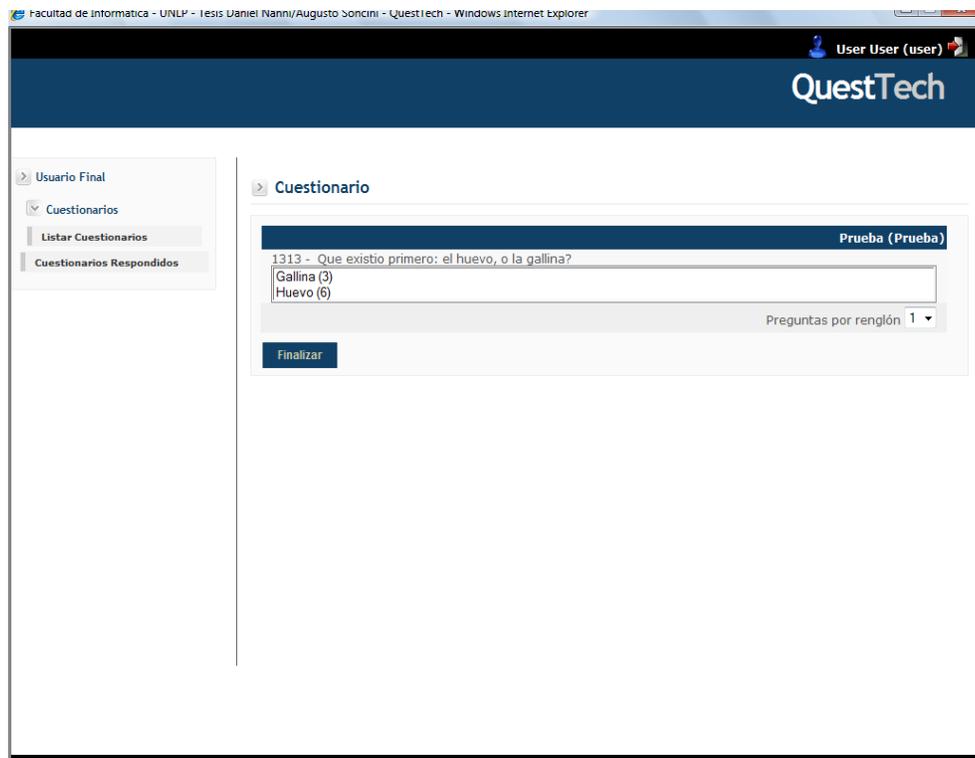
Desde esta opción del menú se listan los Cuestionarios disponibles para ser completados, como se ve en la siguiente figura.



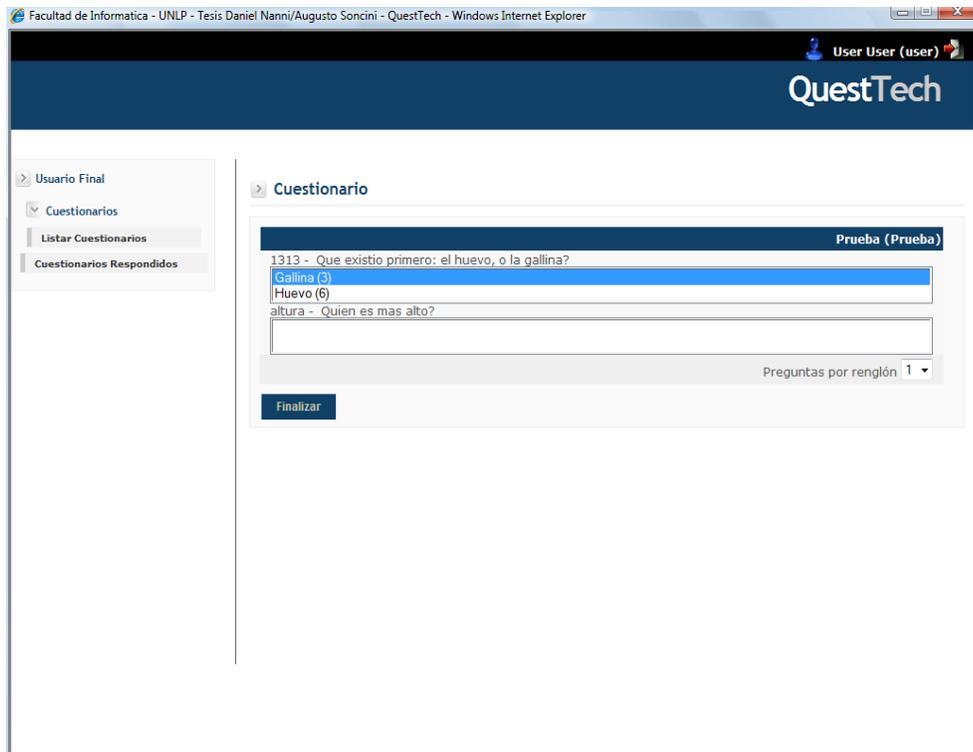
Si se opta por “Visualizar” el Cuestionario, se accede a la pantalla con el detalle del mismo, desde el cual puede imprimirlo, como se ve en la siguiente figura.



Si se opta por “Responder” el cuestionario, la aplicación comenzará a presentar una a una las preguntas, y pidiendo interacción con el usuario para que las responda, como se ve en las figuras siguientes



La aplicación irá guiando al usuario hacia la siguiente pregunta que debe responder de acuerdo a sus respuestas.



Los cuestionarios que el usuario haya ido completando (parcial o totalmente), pueden ser accedidos desde la opción “Cuestionarios Respondidos”, como se ve en las figuras siguientes.

Facultad de Informatica - UNLP - Tesis Daniel Nanni/Augusto Soncini - QuestTech - Windows Internet Explorer

User User (user)

# QuestTech

Usuario Final

- Questionarios
- Listar Questionarios
- Questionarios Respondidos

## Respuestas de Questionarios

	Id	Questionario	Fecha de Alta	Estado
<input type="radio"/>	50	MoProSoft	03/07/2008	Abierto
<input type="radio"/>	51	MoProSoft	15/07/2008	Abierto
<input type="radio"/>	55	Prueba	27/11/2008	Abierto

Ver Todos ▾

[Editar](#) [Visualizar](#) [Borrar](#)

## 10. CONCLUSIÓN

Sin duda, el campo de la mejora y evaluación de procesos de desarrollo de software es muy prolífico y emergente, un campo muy activo de trabajo.

Finalmente las organizaciones llegaron a la conclusión de que su fundamental problema es la incapacidad de manejar procesos para el desarrollo de software. Los beneficios de la utilización de nuevos métodos y herramientas no se pueden obtener dentro del desorden de un proyecto indisciplinado y caótico. En muchas organizaciones, los proyectos frecuentemente llevan un tiempo excesivo y llegan hasta duplicar el tiempo planeado por el presupuesto. En tales casos, la organización no está proveyendo la infraestructura y soporte necesarios para ayudar al proyecto a evitar estos problemas.

En los últimos años en que los estándares de calidad internacionales se han reorientado hacia el aseguramiento de la calidad, han aparecido en la industria del software dos importantes modelos de referencia: Modelo CMM y su sucesor CMMI y Modelo SPICE.

Estos modelos son muy rigurosos y difíciles de implantar para las organizaciones de desarrollo de software y más aun para las pequeñas y medianas empresas, en donde según diferentes estadísticas se demuestra que son mayoría.

Cabe resaltar en este sentido el esfuerzo que están realizando los países iberoamericanos para afrontar esto para el tipo de industria de desarrollo de software que poseen, que como se mencionó con anterioridad, mayormente, son industrias pequeñas y medianas, para las cuales los estándares internacionales como CMMI e ISO se tornan prácticamente imposibles de implementar.

En particular MoProSoft, nuestro caso de estudio, define un conjunto pequeño de procesos que abarcan las responsabilidades de la alta dirección, gestión y operación. Es un modelo integrado, en el cual: las salidas de un proceso están claramente dirigidas como entradas a otros procesos; las prácticas de planeación, seguimiento y evaluación fueron incluidas en todos los procesos de gestión y administración; los objetivos, los indicadores, las ediciones y las metas cuantitativas fueron incorporadas de manera congruente y práctica en todos los procesos; las verificaciones, validaciones y pruebas están incluidas de manera explícita dentro de las actividades de los procesos; y existe una base de conocimiento que resguarda todos los documentos y productos generados por los procesos.

En conclusión, es un modelo que, de manera pragmática, presenta las mejores prácticas para la industria de software, convirtiéndose en una muy buena opción para la pequeña y mediana industria.

Como corolario de este análisis, se construyó una herramienta Web para el soporte del proceso de recolección y evaluación de datos de las organizaciones basada en el

modelo MoProSoft, lo suficientemente genérica como para soportar distintos aspectos a evaluar tales como Categoría, Proceso o Subproceso, con el objetivo de servir de apoyo al proceso de certificación de PYMES de desarrollo de software.

## 11. BIBLIOGRAFÍA

[1] Pino, F., F. Garcia, and M. Piattini, Revisión sistemática de mejora de procesos software en micro, pequeñas y medianas empresas. Revista Española de Innovación, Calidad e Ingeniería del Software (REICIS), 2006. Vol. 2(1) Abril pp. 6-23.

[2] ESI. Europe Software Institute. 2007. [www.esi.es/en/main/iitmark.html](http://www.esi.es/en/main/iitmark.html).

[3] Mayer&Bunge. Panorama de la Industria del Software en Latinoamérica. Mayer & Bunge Informática LTDA. Brasil. 2004. [www.mbi.com.br/200409\\_panorama\\_industria\\_software\\_america\\_latina.pdf](http://www.mbi.com.br/200409_panorama_industria_software_america_latina.pdf).

[4] Fayad, M.E., M. Laitinen, and R.P. Ward, Software Engineering in the Small. Communications of the ACM, 2000. Vol. 43(3) March pp. 115-118.

[5] Goldenson D.R., Gibson D.L.: Demonstrating the Impact and Benefits of CMMI: An Update and Preliminary Results. CMU/SEI-2003-SR-009 (Oct. 2003)

[6] El-Emam K., Goldenson D., McCurley J., Herbsleb J.: Modeling the Likelihood of Software Process Improvement: An Exploratory Study. Empirical Software Engineering 6, 3. 207-229 (Sep. 2001)

[7] Harter D. E., Krishnan M. S., Slaughter S. A.: Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development. Management Science 46, 4: 451-466 (Abr. 2000)

[8] Goldenson D.R., Gibson D.L., Ferguson R.W.: Why Should I Switch to CMMI®? Initial Evidence about Impact and Value Added. 3<sup>rd</sup>. Annual CMMI Technology Conference and User Group: Track on Impact and Benefits of CMMI. Pittsburgh (Nov. 2003).

[9] Jung H-W., Hunter R., Goldenson DR., El-Eman K.: Findings from Phase 2 of the SPICE Trials. Software Process Improvement and Practice. 6: 205-242 (2001)

[10] El-Emam K., Birk A.: Validating the ISO/IEC 15504 Measure of Software Development Process Capability. The Journal of Systems and Software, 51 119-149 (2000)

[11] Rout T.P.: ISO/IEC 15504 – Evolution to an International Standard. Software Process Improvement and Practice. 8: 27-40 (2003)

[12] Dorling A., Kitson D.H.: The Impact of ISO/IEC 15504 on CMM Integration Effort. TickIT International. 2Q99. Pag.9 (1999)

[13] Tuffley A., Grove B., McNair G.: SPICE for Small Organizations. Software Process: Improvement and Practice. Vol.9 Issue 1. 23-31 (Sep 2004).

[14] Krishnan M.S., Kellner M. I.: Measuring Process Consistency: Implications for Reducing Software Defects. IEEE Transactions on Software Engineering 25, 6: 800-815 (Nov/ Dec 1999)

[15] Weber K. Rocha A. Modelo de Referência para Melhoria de Processo de Software: uma abordagem brasileira. Proc. of the QUATIC 2004, 73-78.

[16] Hurtado J. y otros. SIMEP-SW- Sistema Integral de Mejoramiento de los Procesos de Desarrollo Software en Colombia. Colciencias. 2003.

[17] Hurtado J. El modelo integral de mejoramiento Agile SPI. Universidad del Cauca. 2004.

[18] García F. FMESP: Marco de Trabajo Integrado para el Modelado y la Medición de los Procesos. Universidad Castilla-La Mancha. 2004.

[19] <http://www.csi.map.es/csi/metrica3/introduccion.pdf>

[Kulpa M.K, 2003] Kulpa M.K., Johnson K.A.: Interpreting the CMMI: A Process Improvement Approach.

[Wang Y., 2002] Wang Y., Software Engineering Standards: Review and Perspectives.

[J. Sanders, 1994] J. Sanders, E. Curran. Software Quality.

[M. O. Tingey, 1997] Comparing ISO 9000, Malcom Baldrige and the SEI CMM for software.

[Daniel Galin, 2003] Software Quality Assurance: From Theory to Implementation.

[Stephen Kan, 2003] Metrics and Models in Software Quality Engineering(2<sup>nd</sup> Edition).

[Paittini, 2003] Paittini, Garcia Rubio.Calidad en el desarrollo y mantenimiento del software.

[Mary Beth Chrissis] CMMI: Guidelines for Process Integration and Product Improvement Mary Beth Chrissis, et al Software IEEE Computer Society Press

#### Sitios Web de interés:

<http://www.monografias.com/trabajos16/calidad-sw-pymes/calidad-sw-pymes.shtml>

[http://www.bvs.sld.cu/revistas/aci/vol3\\_3\\_95/aci05395.htm](http://www.bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm)

[http://www.uhu.es/eyda.marin/apuntes/valora/tema10VRI\\_I.ppt](http://www.uhu.es/eyda.marin/apuntes/valora/tema10VRI_I.ppt)

[www.sc.ehu.es/jiwdocoj/remis/docs/DelaVillaadis2004.doc](http://www.sc.ehu.es/jiwdocoj/remis/docs/DelaVillaadis2004.doc)

[www.it-mentor.com.ar/pdf/CMM CulturaOrg.pdf](http://www.it-mentor.com.ar/pdf/CMM CulturaOrg.pdf)

<http://www.usm.edu.ec/abedini/spice/spice5.htm>

<http://ieeexplore.ieee.org/iel5/9907/34417/01642455.pdf>

<http://www.csi.map.es/csi/metrica3/>

<http://www.unap.cl/~setcheve/Metrica/m/index.html>

[http://inf.ucsp.edu.pe/summerschool/Formal%20Method%20in%20Software%20Development\\_files/MoproSoft/MoProsoft%20Escuela%20Verano%20Peru.pdf](http://inf.ucsp.edu.pe/summerschool/Formal%20Method%20in%20Software%20Development_files/MoproSoft/MoProsoft%20Escuela%20Verano%20Peru.pdf)

<http://www.iie.org.mx/boletin032003/ind.pdf>

This document was created with Win2PDF available at <http://www.win2pdf.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.  
This page will not be added after purchasing Win2PDF.