



Proyecto final de carrera  
Ingeniería Informática de Gestión 2012/13

Consultor: Oriol Martí Girona  
Autor: Marcelo Tello Helbling

Memoria

*Este proyecto está dedicado a mis padres, a mis hermanos y a mi pareja, por su apoyo incondicional en esta etapa de mi vida y a Oriol, mi consultor, por su gran ayuda durante la elaboración.*

*Barcelona, 2 de Enero de 2013*

## Resumen

El proyecto CMMI UP surge de la necesidad por parte de las organizaciones de una herramienta que soporte las actividades que se llevan a cabo para afrontar el proyecto de implantación CMMI en el marco de un proyecto interno.

Un proyecto interno de la talla de una implantación CMMI supone afrontar unos costes internos bastante elevados (costes directos) y los que supone interferir en las actividades habituales que desempeñan (costes indirectos).

Por ejemplo, al gasto directo del propio equipo del proyecto de implantación, el proveedor de auditoría y certificación, se le une los gastos derivados del impacto que supone interferir la metodología propia de los proyectos en curso y de las actividades de comunicación interna.

Para el equipo encargado de asumir la difícil tarea de abordar un proyecto de estas características se les plantean una serie de actividades como las detalladas a continuación:

- Gestionar y planificar el proyecto interno de implantación
- Coordinar la agenda con el equipo de auditoría que certificará a la organización.
- Gestionar la comunicación con el equipo de los proyectos seleccionados, informe del avance a dirección, etc.
- Revisar los procesos actuales de la compañía y proponer su alineamiento con el modelo CMMI.
- Revisar y verificar el estado de las tareas encomendadas a cada uno de los equipos de los proyectos seleccionados.
- Informar y tomar decisiones de la viabilidad de cada una de las fases del proyecto.
- Recopilar la base de datos de evidencias para su presentación al equipo de auditoría externo.

CMMI UP aporta una herramienta para el equipo de proyecto de CMMI que les permitirá gestionar de una manera centralizada, todas las actividades anteriormente descritas, así como obtener una visión clara del estado del proyecto y los objetivos abordados.



## Índice de contenido

<b>Resumen .....</b>	<b>2</b>
<b>Índice de contenido.....</b>	<b>3</b>
<b>Índice de ilustraciones.....</b>	<b>5</b>
<b>Introducción.....</b>	<b>7</b>
¿Qué es CMMI y qué pretende? .....	9
Justificación .....	11
Objetivos.....	14
Enfoque y método aplicado.....	15
Recursos .....	16
Planificación del proyecto .....	17
Productos obtenidos .....	19
<b>Análisis previo y especificación de requisitos .....</b>	<b>20</b>
Modelo de negocio .....	20
Debilidades del modelo.....	23
Requisitos .....	24
Descripción del sistema.....	31
Identificación de actores .....	32
Relación de los subsistemas con los actores .....	33
Detalle y funcionalidades de los subsistemas.....	34
Documentación formal de casos de uso .....	37
Modelo estático : Diagrama de clases.....	59
<b>Diseño del sistema .....</b>	<b>64</b>
Decisiones de arquitectura .....	66
Modelo estático : Diagrama de clases.....	73
Clases frontera y gestoras CMMI Core .....	75
Clases frontera y gestoras CMMI Proyectos, revisiones incidencias .....	76
Diseño de casos de uso : Diagramas de actividad de procesos .....	77
Diseño de casos de uso : Diagramas de secuencia .....	81
Diseño E/R CMMI Core.....	86
<b>Diseño de la interfaz gráfica .....</b>	<b>87</b>
Pantalla principal y menú .....	87
Login.....	88

DashBoard .....	89
Pantallas CMMI: Core .....	90
Pantallas CMMI: Proyectos, Roles .....	94
Pantallas CMMI: Revisiones .....	96
<b>Valoración económica .....</b>	<b>101</b>
<b>Conclusiones .....</b>	<b>103</b>
<b>Apéndice A: Glosario .....</b>	<b>104</b>
<b>Apéndice B: Bibliografía .....</b>	<b>118</b>

## Índice de ilustraciones

Ilustración 1 Historia de los modelos de madurez .....	8
Ilustración 2 Constelaciones CMMI.....	9
Ilustración 3 Áreas de interés CMMI.....	9
Ilustración 4 Empresas certificadas en España .....	12
Ilustración 5 Ciclo de vida clásico.....	15
Ilustración 6 Entregables del proyecto .....	18
Ilustración 7 Metodología SCRUM .....	20
Ilustración 8 Esquema de una implantación CMMI .....	22
Ilustración 9 Diagrama de paquetes CMMI UP.....	31
Ilustración 10 Relación de actores y subsistemas .....	33
Ilustración 11 Componentes CMMI UP .....	34
Ilustración 12 Componentes en la representación de casos de uso .....	38
Ilustración 13 Casos de uso subsistema conexión .....	44
Ilustración 14 Casos de uso del subsistema CMMI .....	47
Ilustración 15 Casos de uso de Proyectos .....	50
Ilustración 16 Casos de uso tareas y coberturas .....	52
Ilustración 17 Diagrama de casos de uso del subsistema de revisiones y acciones correctivas ..	57
Ilustración 18 Casos de uso Cuadro de mandos .....	58
Ilustración 19 Diagrama de clases CMMI .....	59
Ilustración 20 Ejemplo información CMMI.....	60
Ilustración 21 Ejemplo Información CMMI (Cont.) .....	61
Ilustración 22 Ejemplo Información CMMI (Cont.) .....	62
Ilustración 23 Diagrama de clases Revisiones .....	63
Ilustración 24 Diagrama de la estructura de la aplicación.....	70
Ilustración 25 Flujo a través de los componentes.....	71
Ilustración 26 Esquema general de la arquitectura .....	72
Ilustración 27 Diagrama de clases CMMI .....	73
Ilustración 28 Diagrama de clases Revisiones .....	74
Ilustración 30 Detalle de la relación de componentes modelo, vista, controlador .....	75
Ilustración 29 Clases frontera y gestoras CMMI Core .....	75
Ilustración 31 Clases frontera y gestoras CMMI Proyectos revisiones e incidencias .....	76
Ilustración 32 Diagrama de actividad para la gestión de usuarios .....	78
Ilustración 33 Diagrama de actividad para la revisión de una tarea .....	79
Ilustración 34 Diagrama de actividad para insertar evidencias .....	80
Ilustración 35 Diagrama de secuencia para listas .....	82
Ilustración 36 Diagrama de secuencia para Altas .....	83
Ilustración 37 Diagrama de secuencias para 'bajas' .....	84
Ilustración 38 Diagrama de secuencia para Dashboard .....	85
Ilustración 39 E-R CMMI Core.....	86
Ilustración 40 Prototipo pantalla Menú .....	87

Ilustración 41 Prototipo Pantalla Login .....	88
Ilustración 42 Prototipo DashBoard .....	89
Ilustración 43 Prototipo lista de Categorías.....	90
Ilustración 44 Prototipo Alta de Categoría .....	91
Ilustración 45 Prototipo lista niveles de Madurez .....	92
Ilustración 46 Prototipo lista áreas de proceso .....	93
Ilustración 47 Prototipo lista Roles .....	94
Ilustración 48 Prototipo lista de proyectos.....	94
Ilustración 49 Prototipo alta de proyectos .....	95
Ilustración 50 Prototipo lista tipos de revisiones .....	96
Ilustración 51 Prototipo lista tareas de revisión .....	96
Ilustración 52 Prototipo Alta de tareas de revisión.....	97
Ilustración 53 Prototipo matriz de revisiones.....	98
Ilustración 54 Prototipo lista de cobertura de tareas .....	99
Ilustración 55 Prototipo calendario de revisiones .....	100

## Introducción

---

*“La calidad nunca es un accidente; siempre es el resultado de un esfuerzo de inteligencia”.* John Ruskin (1819-1900) Crítico y escritor británico.

Durante los años ochenta, el departamento de defensa de los EEUU consciente de la problemática que tenía en los encargos de desarrollo de software (presupuestos, bajo nivel de calidad, incumplimiento de fechas, etc.) , se aventuró a crear un comité de expertos que analizara dicha situación y propusiera una solución.

Dichos expertos, concluyeron como mejor solución, la creación de un instituto de ingeniería de software que diera una respuesta efectiva al cúmulo de problemas que originaba la mala calidad del software, así como la imposibilidad de planificar los tiempos y coste del desarrollo. Fue entonces cuando en 1994, el congreso de los EEUU fundó el Instituto de Ingeniería de Software (SEI) constituido como instituto federal para la investigación y desarrollo administrado por la Universidad de Carnegie Mellon.



Uno de los primeros acometidos del SEI, fue la creación de un modelo de procesos para el desarrollo y mantenimiento de sistemas de software (SW-CMM), que ha evolucionado hasta nuestros días como el modelo CMMI (Capability Mature Model Integration). Este modelo se sustenta sobre los siguientes criterios:

- La calidad de un producto o sistema es consecuencia directa de los procesos empleados en su desarrollo.
- Las organizaciones que desarrollan software presentan un atributo denominado madurez, cuya medida es proporcional a los niveles de capacidad e institucionalización de los procesos que emplean en su trabajo

Dicho de otro modo, para desarrollar software de calidad, es preciso que a totalidad de los procesos empleados en el desarrollo, sean de calidad.

## History of CMMs

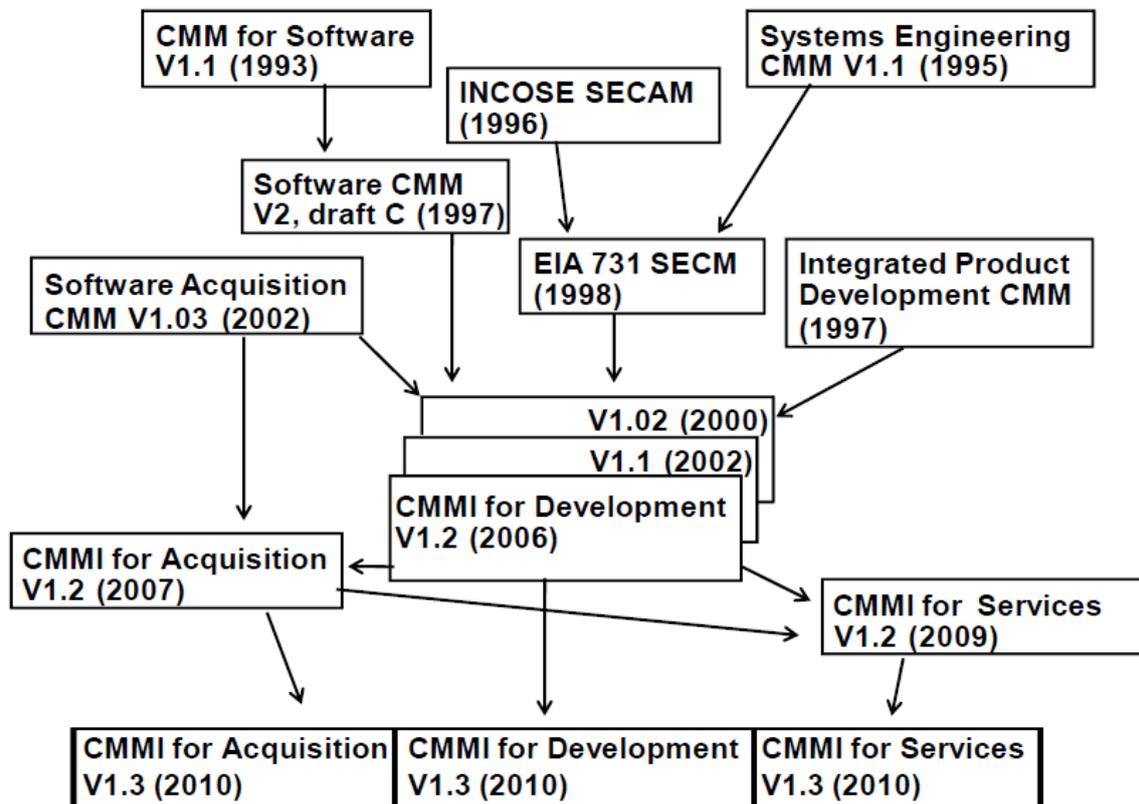


Ilustración 1 Historia de los modelos de madurez

## ¿Qué es CMMI y qué pretende?

Como hemos visto en el apartado anterior, CMMI surge de la necesidad de alinear el trabajo de los proveedores de software hacia su cliente y de plantearles las buenas prácticas que deben cumplir para producir un software con la calidad exigida y en los plazos establecidos.



En el mercado actual, los modelos de madurez, estándares, metodologías y guías, ayudan a las organizaciones a llevar a cabo los objetivos de negocio. No obstante, la mayoría de actividades se centran en una parte específica del negocio, y no realizan un aproximamiento sistemático de los problemas que la mayoría de organizaciones tienen. Esta visión reducida del problema, provoca que muchas empresas arrastren eternamente las dificultades iniciales.

Este redactado de buenas prácticas del modelo CMMI lo que propone es una reingeniería de procesos aplicados al producto o al servicio. Dicha reingeniería es totalmente compatible con un proyecto en particular, un departamento o bien toda la empresa dedicada a la fabricación de software y da la posibilidad de que cada uno de estos procesos fluya de una forma alineada para llevar a cabo el tan complicado proceso de elaboración, mantenimiento y operación de sistemas de software.



Ilustración 2 Constelaciones CMMI

En realidad, CMMI cubre tres áreas de interés: Desarrollo, Adquisición y Servicios, pero en el marco de nuestro proyecto únicamente nos centraremos en, como así lo denominan, la “constelación” de para el desarrollo.

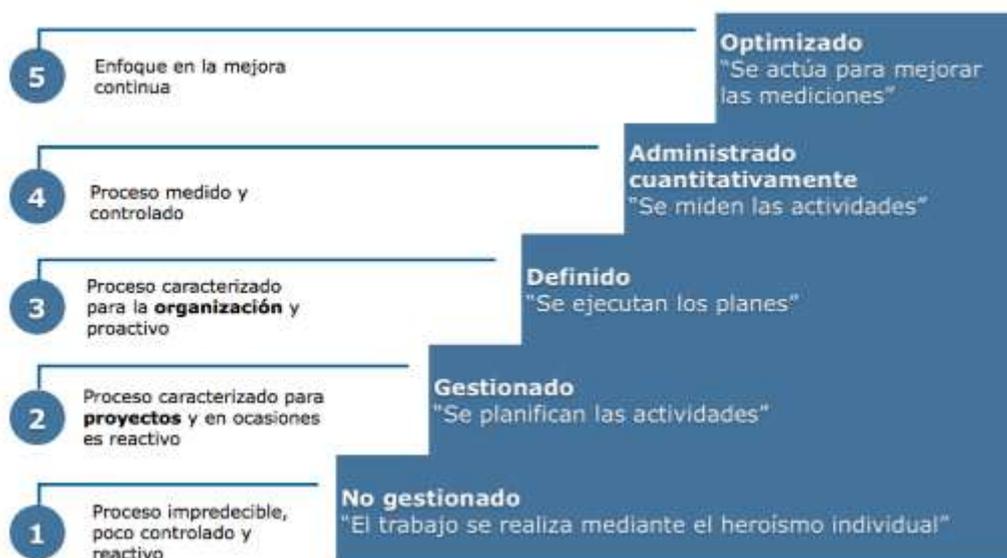
CMMI para el desarrollo de software, contiene 22 áreas de proceso de las cuales, 16 son áreas específicas del núcleo (core), 1 es un área compartida y 5 son áreas específicas de desarrollo.

Como observamos en la ilustración siguiente, la constelación de Desarrollo, posee 4 categorías en los que se distribuyen todos los procesos documentados: Ingeniería, Gestión de proyectos, Gestión de procesos y Soporte, esta última posee los procesos transversales de la organización.



Ilustración 3 Áreas de proceso de la constelación de Desarrollo

Asimismo, todo el modelo CMMI está organizado de tal forma que sus buenas prácticas tienen como objetivo situarnos en uno de estos 5 niveles de madurez.



## Justificación

La situación actual del desarrollo de software afronta un escenario poco alentador. En situaciones de crisis, las grandes empresas que contratan los servicios tecnológicos han sufrido recortes y miran con lupa cualquier gasto en esta partida contable. No es de extrañar que la empresa que posea una certificación CMMI, posee un valor añadido muy competitivo y una presentación excelente de cara a las ofertas de sus clientes.

Algunos ejemplos extraídos a través del siguiente blog de Javier Garzás ([www.javiergarzas.com](http://www.javiergarzas.com)), podemos observar como en el 2010 muchos de los pliegos emitidos por las grandes organizaciones ya incorporan como requisito poseer una certificación CMMI.

- [Ministerio de Ciencia e Innovación](#) (2010). Requiere de CMMI e ISO 15504.
- [Ministerio de Industria, Justicia y Red.es](#) (2010). Obligatoria nivel 3 en CMMI o ISO 15504.
- [Dirección general de patrimonio](#). Subdirección general de compras (2010). Empresas certificadas tanto en CMMI como en ISO 15504
- [Dirección General de Sistemas de Información Sanitaria](#) (2010). Servicio madrileño de salud. Comunidad de Madrid. Requiere de CMMI e ISO 15504.
- [Isdefe](#) (Ingeniería de Sistemas para la Defensa de España) (2010). Requiere sólo de CMMI
- Ayuntamientos: [Ayuntamiento de Benidorm](#) (Alicante), requiere sólo de ISO 15504, [Ayuntamiento de Lorca](#) (Murcia), Requiere sólo de ISO 15504, etc.

Cada vez más son las empresas que exigen una garantía en forma de certificación.

No obstante, no todas las empresas proveedoras están preparadas para afrontar este proyecto de cambio. Los procesos de la organización pasarán a evaluarse y posiblemente alterar sus flujos actuales a la vez que nuevas herramientas pueden surgir e imponerse de forma corporativa. Surgirán nuevos roles y quizás se eliminen otros, y sobre todo, la empresa puede estar sometida a un nivel de presión constante por parte de los partidarios y los detractores de este nuevo modelo.

No toda la organización está preparada para este cambio cultural. Es por este motivo que algunos puntos clave de esta implantación debe ir acompañada de:

- Un proyecto estudiado y planificado al detalle junto con expertos en la materia.
- Un equipo con conocimiento y experiencia en este campo.
- Implicación del comité de dirección de un modo público y transparente.
- Realizar una evaluación previa para saber el estado o nivel de madurez actual.
- **Definir las herramientas de gestión no sólo las que se utilizarán de manera estándar en el contexto de CMMI sino las que se utilizarán para dar soporte al proyecto de implantación.**

En la siguiente ilustración podemos observar los datos referidos al año 2010 y que nos muestran el número de organizaciones certificadas en CMMI clasificadas por nivel de madurez en España.

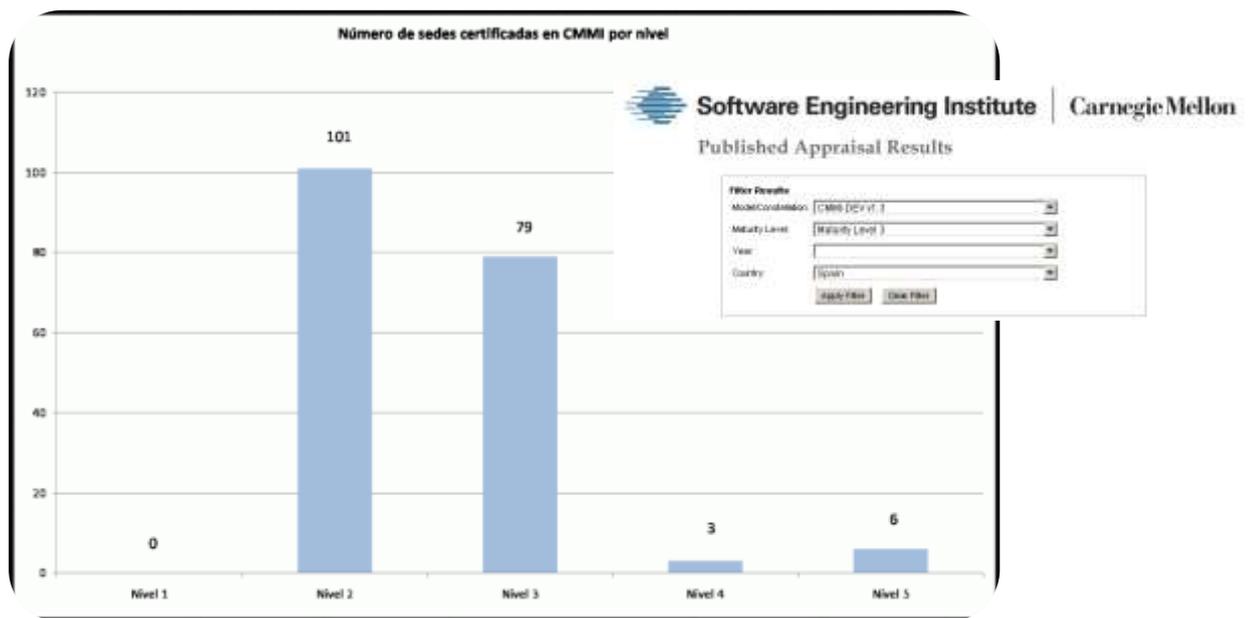


Ilustración 4 Empresas certificadas en España

Como se puede observar, la mayoría de organizaciones optan por los niveles 2 y 3 de CMMI, que coincide con la exigencia de las ofertas y pliegos publicados por las grandes administraciones y empresas públicas.

El riesgo que supone alcanzar un nivel 4 o 5 debe ser revisado con lupa, puesto que la relación coste-beneficio no siempre está a favor de la empresa proveedora.

Estos datos son publicados frecuentemente por el SEI a través del siguiente enlace:

<https://sas.sei.cmu.edu/pars/pars.aspx>

## Objetivos

El objetivo primordial de este proyecto es llevar a cabo las tareas de Análisis de Requerimientos, Especificación y Diseño comprendidos en el ciclo de vida del software.

El análisis de requerimientos se llevará a cabo con la información aportada por la propia compañía en cuanto a necesidades y problemáticas encontradas durante una implantación real a la que se le darán las soluciones apropiadas para disponer de una herramienta que permita optimizar ciertas actividades para la gestión de una implantación CMMI.

Esta solución, no sólo debe aportar una solución práctica al problema, sino que debe argumentarse en un contexto de ahorro de costes y retorno de la inversión, que es la prioridad de cualquier organización, por tanto, es muy importante recalcar que cada especificación de requisito englobe cierto enfoque de este tipo.



## Enfoque y método aplicado

La metodología escogida para la realización del proyecto es la llamada “metodología en cascada”.

### Ventajas

- Calidad del producto alta
- Permite detectar problemas de viabilidad del proyecto antes de proseguir con fases posteriores.
- Reducción de costes de desarrollo debido a la alta calidad en la fase de requisitos.

### Inconvenientes

- Los problemas surgidos en alguna fase implican un rediseño o reprogramación y su consecuente aumento de costes.
- Requisitos con un alto grado de detalle y fiabilidad (comentado como riesgo a controlar).
- El producto final no es visible hasta el final.

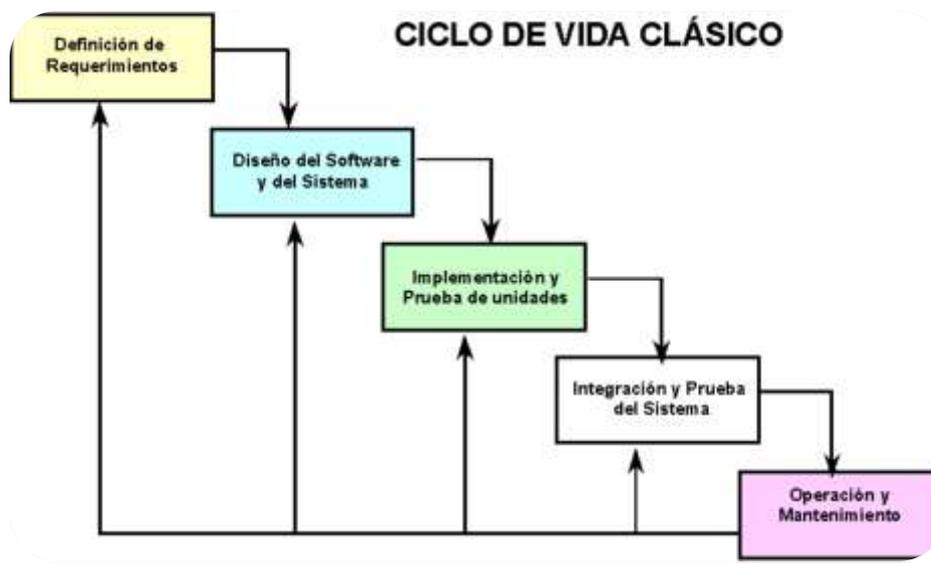


Ilustración 5 Ciclo de vida clásico

## Recursos

Los recursos necesarios para llevar a cabo el proyecto son:

### Personal

- Única persona para la elaboración de todas las tareas descritas en el apartado 'Tareas y planificación'

### Herramientas

- PC de Sobremesa con conexión a Internet
- Software Ofimático Office : Word, Project, Excel, PowerPoint
- Software para el diseño de base de datos: DB-Designer
- Software para la elaboración de prototipos: Visio, Yii Framework, etc.
- Software para la elaboración de scripts de persistencia : MySQL

### Documentación:

- Toda la documentación detallada en la bibliografía.

## Planificación del proyecto

Para realizar la planificación, y, siguiendo las buenas prácticas en gestión de proyectos, hemos realizado una descomposición estructural de actividades (WBS).

Descomposición estructural de actividades (WBS)		
Código de actividad	Nombre de actividad (Nivel I)	Nombre de actividad (Nivel II)
PAC-1	Plan de trabajo	
1.1		Propuesta
1.2		Descripción
1.3		Objetivos
1.4		Análisis situación actual
1.5		Análisis y gestión de riesgos
1.6		Elección de metodología
1.7		Descomposición de tareas
1.8		Planificación
		Elaborar entregable

Descomposición estructural de actividades (WBS)		
Código de actividad	Nombre de actividad (Nivel I)	Nombre de actividad (Nivel II)
PAC-2	Especificación y análisis de requisitos	
2.1		Introducción y marco del proyecto
2.2		Elaboración de requisitos
2.3		Descripción del sistema
2.4		Descripción de los subsistemas
2.5		Documentación de actores y casos de uso
2.6		Diagramas necesarios
2.7		Elaboración entregable Análisis

Descomposición estructural de actividades (WBS)		
Código de actividad	Nombre de actividad (Nivel I)	Nombre de actividad (Nivel II)
PAC-3	Diseño técnico	
3.1		Diseño técnico de los subsistemas
3.2		Diagramas (colaboración y secuencia)
3.3		Diseño de casos de uso
3.4		Base de Datos: Diseño de persistencia y E/R
3.5		Prototipo Interface de Usuario
3.7		Entregable Diseño

Descomposición estructural de actividades (WBS)		
Código de actividad	Nombre de actividad (Nivel I)	Nombre de actividad (Nivel II)
PAC-4	Memoria y presentación	
4.1		Revisión de entregables
4.2		Elaboración de anexos
4.3		Bibliografía y glosario de términos
4.4		Revisión y entregable Memoria
4.5		Elaboración presentación



**ENTREGABLES**



Ilustración 6 Entregables del proyecto

## Productos obtenidos

Una vez abordado el proyecto en la totalidad de sus fases obtenemos los siguientes entregables:

- Documento de análisis y especificación

En este documento hemos contemplado los requisitos de la organización en la que nos hemos basado para abordar dicho proyecto. En el podemos situar el proyecto de una forma realista puesto que se trata de una empresa real con necesidades reales.

- Documento de diseño

Partiendo del documento anterior, realizamos el diseño del sistema abordando la mayoría de los requisitos funcionales. Con ello conseguimos un entregable que proporcionará la base para la implementación posterior, la cual, no hemos contemplado en el marco de este proyecto.

- Memoria del proyecto

En este documento presentamos todo el proyecto conjunto con los anexos correspondientes, bibliografía, glosario.

- Presentación del proyecto

Con esta presentación pretendemos transmitir de una forma sintetizada cómo hemos enfocado el proyecto y qué hemos pretendido abordar con el mismo. De una manera gráfica y resumida, intentamos comunicar al oyente todos los aspectos más relevantes del proyecto.

- Prototipo realista del proyecto

Para evaluar la viabilidad y hacer una estimación realista del proyecto en términos económicos, hemos iniciado un prototipo del mismo. Aunque no entra dentro del marco del proyecto, hemos querido mencionarlo.

## Análisis previo y especificación de requisitos

### Modelo de negocio

El caso que nos ocupa, viene a solucionar una situación en la que una empresa que desarrolla software debe abordar una implantación de CMMI en el marco de un proyecto interno de mejora de la competitividad.

Dicho proyecto interno se aborda bajo el paraguas de una metodología iterativa e incremental llamada SCRUM. La particularidad de esta metodología, en el contexto de la organización, es que el producto final no es en sí un producto, sino una “garantía” de que toda la lista de tareas encomendadas para el cumplimiento de los objetivos que marca CMMI dentro de sus respectivas áreas se han cumplido.

Una vez un comité directivo escoge las condiciones bajo las cuales deben ser escogidos los proyectos candidatos para abordar la auditoría se planifica un kick-off del proyecto en el que se presentan, a los interesados, los objetivos, riesgos e hitos del proyecto.

Las fases del proyecto de implantación se basan en iteraciones acotadas en el tiempo, durante el cual, el equipo (tanto interno como el de los diversos proyectos candidatos) trabaja para acercarse cada vez más a la consecución de las metas marcadas por CMMI. Estas iteraciones son las que SCRUM denomina SPRINT.

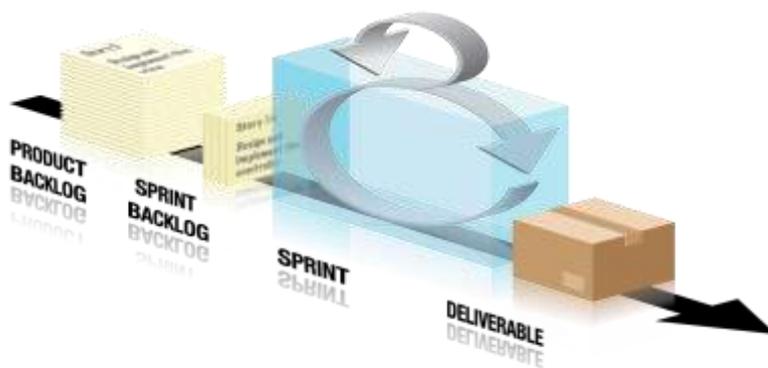


Ilustración 7 Metodología SCRUM

En el marco del proyecto de implantación, el SPRINT está compuesto por tres actividades principales:

**Desarrollo:**

Durante esta actividad, el equipo interno de CMMI especifica y estudia las actividades que deben realizarse para un área concreta. Se revisa el repositorio documental donde se haya CMMI y se asocian actividades para el cumplimiento de los objetivos marcados, tanto genéricos como específicos. En esta etapa intervienen expertos en función del área que se trate: Jefes de proyecto, Ingenieros de software, Personal de áreas de calidad, etc.

**Despliegue:**

La fase despliegue contemplan todas las actividades a realizar por los diferentes equipos de los proyectos seleccionados y en las que se han trabajado en la fase de desarrollo. El impacto de esta fase en los proyectos seleccionados y para una primera implantación de CMMI es bastante alto, dado que tendrán que crear documentación específica, alterar procedimientos, usar nuevas herramientas, etc. En esta fase, se incorporan unas figuras llamadas 'multiplicadores', que son personas formadas por el equipo interno del proyecto de implantación para el asesoramiento a los equipos de proyectos para la consecución de las actividades a realizar. Cada multiplicador puede tener asignados de uno a tres proyectos.

**Revisión:**

Durante este intervalo de tiempo, el equipo interno del proyecto de implantación recopila la información que debería haberse realizado y la cruza con los diferentes proyectos. Esta matriz es revisada y puesta al día con diferentes estados: Realizado, No realizado, No procede, etc.

Cuando una revisión da un resultado negativo, el revisor propone una acción correctiva al equipo de proyecto. Esta acción correctiva debe ser llevada a cabo para su nueva revisión.

Esta fase es muy importante ya que marca el indicador de progreso del proyecto, y en consecuencia, el punto de situación en relación al nivel de madurez que se haya marcado la organización.

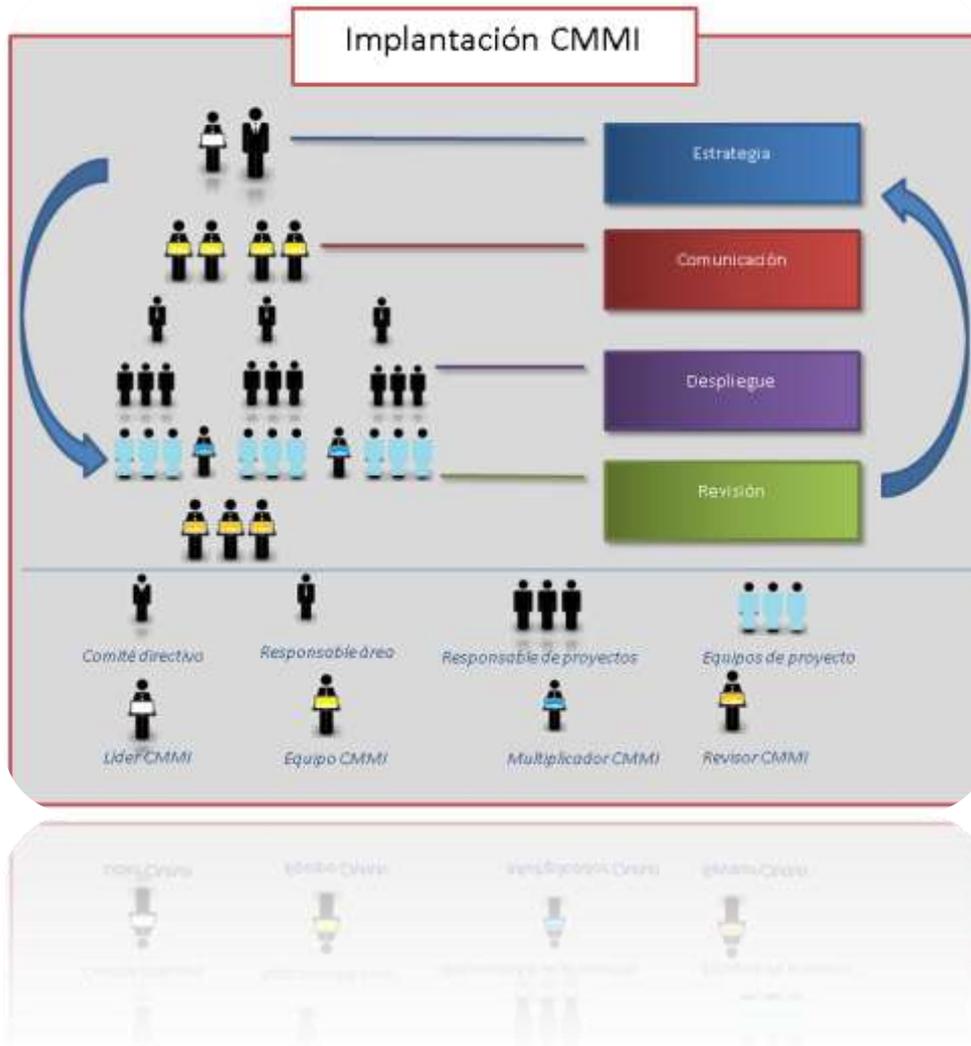
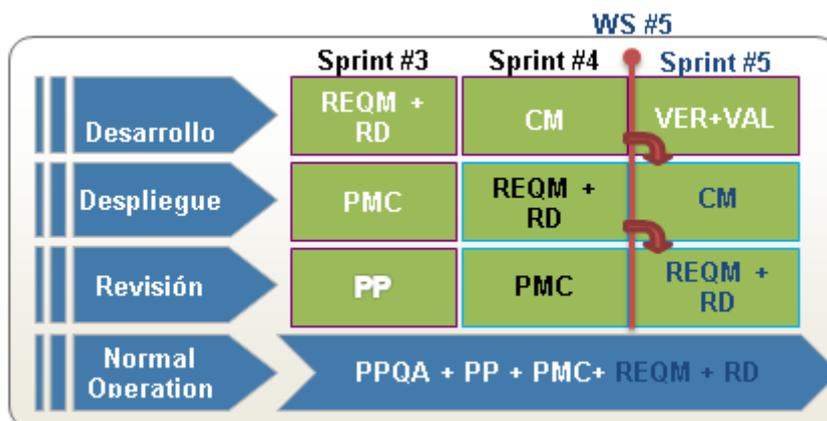


Ilustración 8 Esquema de una implantación CMMI

En el siguiente esquema podemos ver un ejemplo de lo comentado anteriormente:



Además de las fases iterativas, se realizan los llamados WorkShops. En estas reuniones de trabajo se desarrollan las siguientes actividades:

- **Un representante de cada área (que engloba diferentes proyectos) presenta los resultados del despliegue** llevado a cabo e iniciado en el sprint anterior. Este representante comenta de una forma genérica, el grado de consecución de las tareas encomendadas y el motivo por el cual, los proyectos bajo su tutela, no ha llevado a cabo las que no ha podido abordar.
- **Un representante del equipo interno de implantación presenta los resultados de las revisiones** realizadas al despliegue efectuado en el sprint anterior junto con el avance total de la implantación.
- **Un representante del equipo interno de implantación describe las tareas que deberán abordarse para el siguiente sprint** y que ya han sido alineadas con los objetivos del área concreta de CMMI.
- **Se forman a los llamados ‘multiplicadores’,** que son las personas que darán apoyo y asesoramiento al equipo de los diferentes proyectos en la consecución de las actividades a realizar.

### Debilidades del modelo

A raíz de las actividades mencionadas anteriormente y de las cuales he participado como multiplicador, he podido deducir alguno de los requisitos que he contrastado con el personal del equipo CMMI. Por otro lado, se ha realizado una labor posterior de entrevistas con el equipo, para averiguar las necesidades de los diversos actores participantes en la implantación.

A modo resumen, podríamos destapar las siguientes debilidades:

Equipo de proyectos seleccionados

- Los equipos de proyecto (analistas, jefes de proyecto, gestores de calidad, testers) no tienen claro qué tareas deben cumplir y con qué objetivo.
- Los multiplicadores no disponen de un repositorio amigable y con documentación fácilmente organizada donde figuren las tareas y la información sobre el modelo de madurez CMMI

- En ocasiones, los roles de los equipos de proyecto son mixtos, lo que dificulta la asignación de tareas a realizar.

#### Equipo de revisiones de tareas

- Las revisiones se llevan a cabo por diferentes personas (en función del área) y con herramientas o métodos distintos, lo que dificulta la centralización y el tratamiento de la información.
- La manipulación de la información es compleja con las herramientas ofimáticas habituales.
- Existe una tendencia a concentrar las tareas y revisiones al comienzo de los sprints, lo que crea una presión añadida al equipo de revisiones.

#### Equipo de gestión de proyecto CMMI

- Existen dificultades en la integración y explotación de los datos revisados para su presentación en los WorkShops.
- Las herramientas usadas para presentar los indicadores de progreso se hacen lentas al tener que manipular muchos datos.
- Falta de una visión global de la situación y progreso.

### Requisitos

En este apartado haremos una relación de los requisitos que debe contemplar este proyecto para dar solución a las necesidades de la empresa.

Por un lado, enumeramos los requisitos funcionales, que son los que definirán el comportamiento del sistema y por otro los requisitos no funcionales, que son aquellos en los que intervienen en aspectos cualitativos, de rendimiento, etc.

#### Lista de Requisitos funcionales

Subsistema	Identificador	Descripción
Conexión	RF_0.1	Gestión de usuarios y permisos
CMMI Core	RF_1.1	Gestión de objetivos genéricos
	RF_1.2	Gestión de prácticas genéricas
	RF_1.3	Gestión de áreas de proceso
	RF_1.4	Gestión de categorías de áreas de proceso

	RF_1.5	Gestión de niveles de madurez
Proyectos candidatos	RF_2.1	Mantenimiento de proyectos
	RF_2.2	Mantenimiento de roles
	RF_2.3	Mantenimiento de departamentos
	RF_2.4	Equipos de proyecto
Tareas de revisión y cobertura	RF_3.1	Gestión de tipos de revisiones
	RF_3.2	Tareas de revisión
	RF_3.3	Coberturas de tarea
Revisiones y evidencias	RF_4.1	Generación de tareas de revisión
	RF_4.2	Agenda de revisiones y acciones de revisión
	RF_4.3	Incidencias y acciones correctivas
DashBoard	RF_5.1	Cuadro de mandos con gráficas y posibilidad de impresión

### Detalle de requisitos funcionales

#### RF 0.1: Gestión de usuarios

**Descripción del requisito:** Este requisito es fundamental para el funcionamiento de la aplicación, pues desde él, el sistema validará las credenciales y dará permiso a las diferentes opciones de menú.

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades para un usuario:

- Datos demográficos
- Datos para la validación de credenciales
- Datos para el acceso a las diferentes opciones del menú principal de la aplicación.

#### RF 1.1: Gestión de objetivos genéricos

**Descripción del requisito:** Este requisito pertenece a la información del modelo CMMI. Son los objetivos transversales que debe cumplir la organización

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades :

- Código y descripción

#### RF 1.2: Gestión de prácticas genéricas

**Descripción del requisito:** Este requisito pertenece a la información del modelo CMMI. Son las prácticas que deben cumplirse para cumplir con los objetivos genéricos

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Código y descripción de práctica
- Objetivo para el cual se realiza la práctica

### RF 1.3: Gestión de áreas de proceso

**Descripción del requisito:** Este requisito pertenece a la información del modelo CMMI. Son las áreas de proceso que engloba el modelo de madurez

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Código y descripción
- Categoría a la que pertenece y nivel de madurez al que pertenece

### RF 1.4: Gestión de categorías de áreas de proceso

**Descripción del requisito:** Este requisito pertenece a la información del modelo CMMI. Son las categorías en las que se clasifican las áreas de proceso

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Código y descripción.

### RF 1.5: Gestión de niveles de madurez

**Descripción del requisito:** Este requisito pertenece a la información del modelo CMMI. Son los niveles de madurez del modelo

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Número de nivel (1-5) y descripción.

### RF 2.1: Gestión de proyectos candidatos

**Descripción del requisito:** Con este requisito podremos gestionar los proyectos candidatos a la auditoría CMMI

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Código y descripción
- Departamento al que pertenece
- Candidato/No candidato
- Fechas de vigencia

### RF 2.1: Gestión de roles de proyecto

**Descripción del requisito:** Con este requisito podremos asignar roles a los diferentes miembros del equipo de un proyecto

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Código y descripción

### RF 2.3: Gestión de departamentos

**Descripción del requisito:** Cada proyecto pertenece a una unidad funcional o departamento. A través de este requisito pretendemos gestionar las diferentes unidades para un proyecto

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Código y descripción

### RF 2.1: Gestión de equipos de proyecto

**Descripción del requisito:** Con este requisito podremos asignar miembros y roles a los proyectos

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Usuario, Rol, proyecto

### RF 2.1: Gestión de proyectos candidatos

**Descripción del requisito:** Con este requisito podremos gestionar los proyectos candidatos a la auditoría CMMI

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Código y descripción
- Departamento al que pertenece
- Candidato/No candidato
- Fechas de vigencia

### RF 3.1: Gestión de tipos de revisiones

**Descripción del requisito:** Con este requisito podremos gestionar los tipos de revisiones que se realizan (frecuencia)

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Código y descripción
- Frecuencia de días
- Única/No única

### RF 3.2: Gestión de tareas de revisión

**Descripción del requisito:** Con este requisito podremos gestionar las diferentes tareas que se llevan a cabo para posteriormente revisarlas en el contexto del proyecto.

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Código y descripción
- Tipo de revisión

### RF 3.3: Gestión de coberturas de revisión

**Descripción del requisito:** Con este requisito podremos gestionar las diferentes tareas que se llevan a cabo para posteriormente revisarlas en el contexto del proyecto.

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Tarea que dará cobertura
- Área de proceso a la que dará cobertura
- Objetivo específico al que dará cobertura.

### RF 4.1: Generación de tareas de revisión

**Descripción del requisito:** Con este requisito podremos realizar la matriz de revisiones que comprende los proyectos y las diferentes tareas que deberán realizar.

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Fecha hasta la cual se realizará la generación de tarea.
- Proyectos de los cuales se quiere realizar la generación.

### RF 4.2: Agenda de revisiones y acciones de revisión

**Descripción del requisito:** Una vez generada la agenda de revisiones, podremos seleccionar proyecto y tarea y actualizar su estado así como comentarios y evidencias

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Seleccionar estado de revisión
- Evidencia de tarea cumplida.
- Almacenar timestamp de revisión.

### RF 4.3: Incidencias y acciones correctivas

**Descripción del requisito:** Por una parte, durante la tarea de revisión es posible generar una incidencia que debe ser corregida por un miembro del equipo de proyecto. Cada miembro del proyecto tendrá un buzón donde constan las incidencias reportadas por el revisor y la acción correctiva detallada.

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Incidencia, revisión asociada
- Acción correctiva aconsejada
- Responsable de la acción correctiva.

#### RF 5.1: Cuadro de mandos

**Descripción del requisito:** Se debe tener acceso a un cuadro de mandos con diferentes gráficas como la posibilidad de imprimir

**Requisitos de almacenamiento:** Deben tenerse en cuenta las siguientes propiedades:

- Gráficas porcentuales de cobertura del modelo en el total de proyecto de implantación
- Gráficas porcentuales de cobertura del modelo en proyectos concretos.
- Gráfico de tareas cumplidas/no cumplidas/ en espera, etc.

#### Requisitos no funcionales

Identificador	Clasificación	Descripción
RNF_1	Rendimiento	La aplicación debe tener un rendimiento óptimo para el uso compartido por 10-20 usuarios concurrentes
RNF_2	Disponibilidad	La disponibilidad para el usuario debe ser en horario laboral, dejando el resto del horario para tareas de mantenimiento.
RNF_3	Seguridad	La aplicación debe cumplir con los estándares de seguridad corporativos en el contexto de una intranet.
RNF_4	Usabilidad	La aplicación debe hacer uso de las buenas prácticas en usabilidad y adecuar su look & feel al corporativo.
RNF_5	Estabilidad	Sistema estable en entornos de servidor web.
RNF_6	Portabilidad	Aplicación disponible en múltiples plataformas
RNF_7	Interoperabilidad	La aplicación deberá integrarse con el sistema de usuarios LDAP corporativo.
RNF_8	Escalabilidad	La implementación debe proporcionar mecanismos para que la aplicación sea fácilmente escalable.
RNF_9	Concurrencia	La concurrencia deberá contemplar 10-20 usuarios.
RNF_10	Mantenimiento	

### Requisitos empresariales

Identificador	Clasificación	Descripción
RE_1	Económico	Reducción del gasto del proyecto
RE_2	Estratégico	Reforzar la competitividad en el mercado

## Descripción del sistema

CMMI UP es un software clasificado como empresarial. Este tipo de software está orientado a mejorar la productividad o a medirla. En este sentido, como hemos mencionado anteriormente, el sistema se compone de diversos módulos funcionales que ayudará a cada actor del sistema a mejorar su cometido y en términos generales mejorar la productividad y aumentar la eficiencia de la implantación CMMI.

A continuación podemos ver un diagrama de paquetes que ilustra esta descomposición.

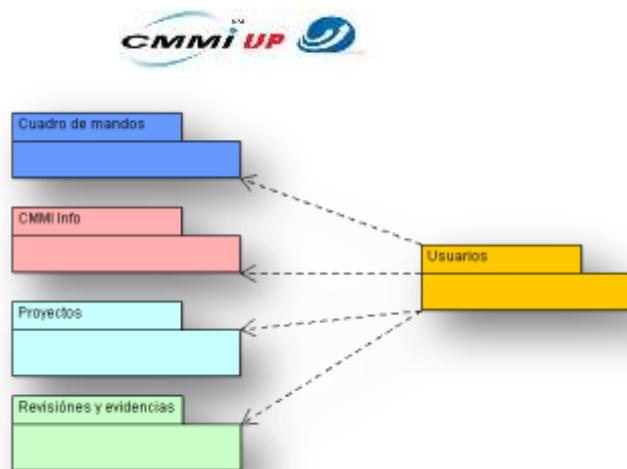


Ilustración 9 Diagrama de paquetes CMMI UP

## Identificación de actores



### Coordinador del equipo CMMI

El coordinador del equipo CMMI es en realidad el jefe del proyecto. Su misión es asegurarse que el proyecto se lleva a cabo en los términos establecidos tanto temporales como de coste.



### Experto en CMMI

El Experto en CMMI es el que mejor conoce el modelo de madurez y las metodologías actuales de la empresa. Junto con los expertos en las áreas de ingeniería de software acuerda las actividades que deben realizarse para que se cumplan los objetivos CMMI.



### Multiplicador CMMI

Los multiplicadores asisten a los WorkShops informativos y dan apoyo durante el despliegue de los sprints a los diferentes equipos. Son conocedores de cómo deben realizarse las tareas encomendadas.



### Revisor CMMI

Los revisores son las personas que diariamente revisan que las tareas encomendadas se han cumplido. Para ello, en una matriz de proyectos/tareas, van apuntando el estado de la tarea. Si existe una tarea que no está bien realizada o no hecha, dan de alta un registro en una base de datos de incidencias corporativa para que se resuelvan.



### Usuario de Proyecto involucrado

Denominamos así a cualquier miembro de un equipo de proyecto que esté interesado en acceder al repositorio documental de CMMI.

## Relación de los subsistemas con los actores

En el siguiente diagrama se representa la relación de los diferentes subsistemas con los actores.

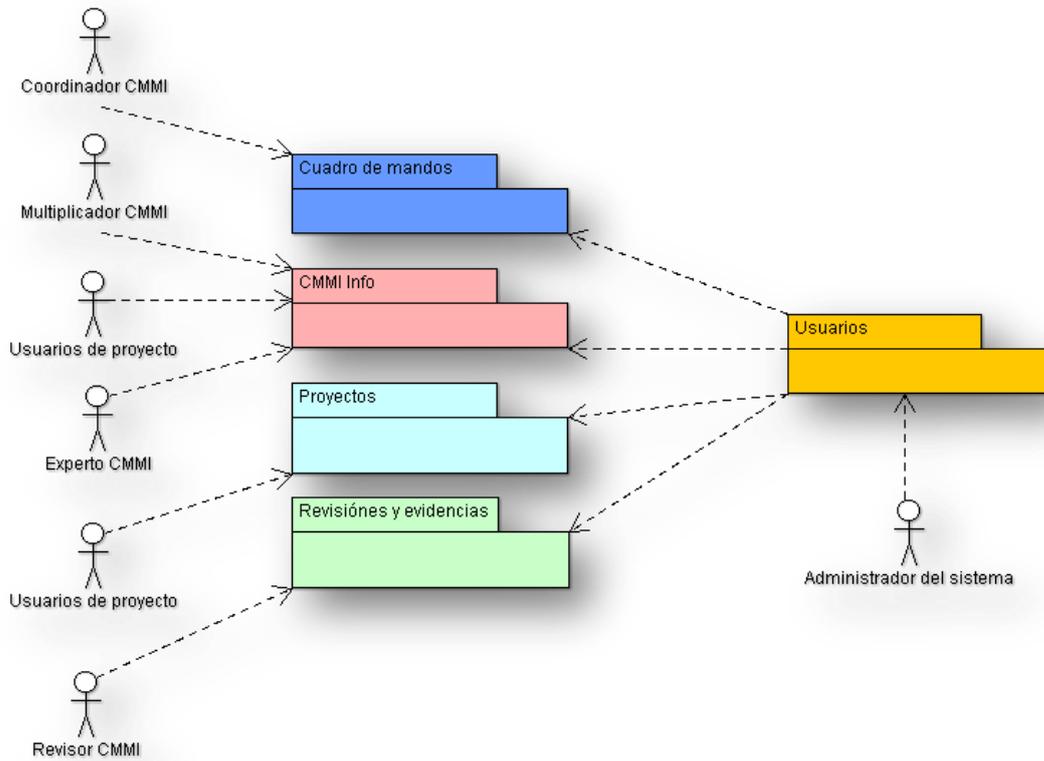


Ilustración 10 Relación de actores y subsistemas

## Detalle y funcionalidades de los subsistemas

A continuación se representa un gráfico más ilustrativo de los componentes y funcionalidades del sistema.

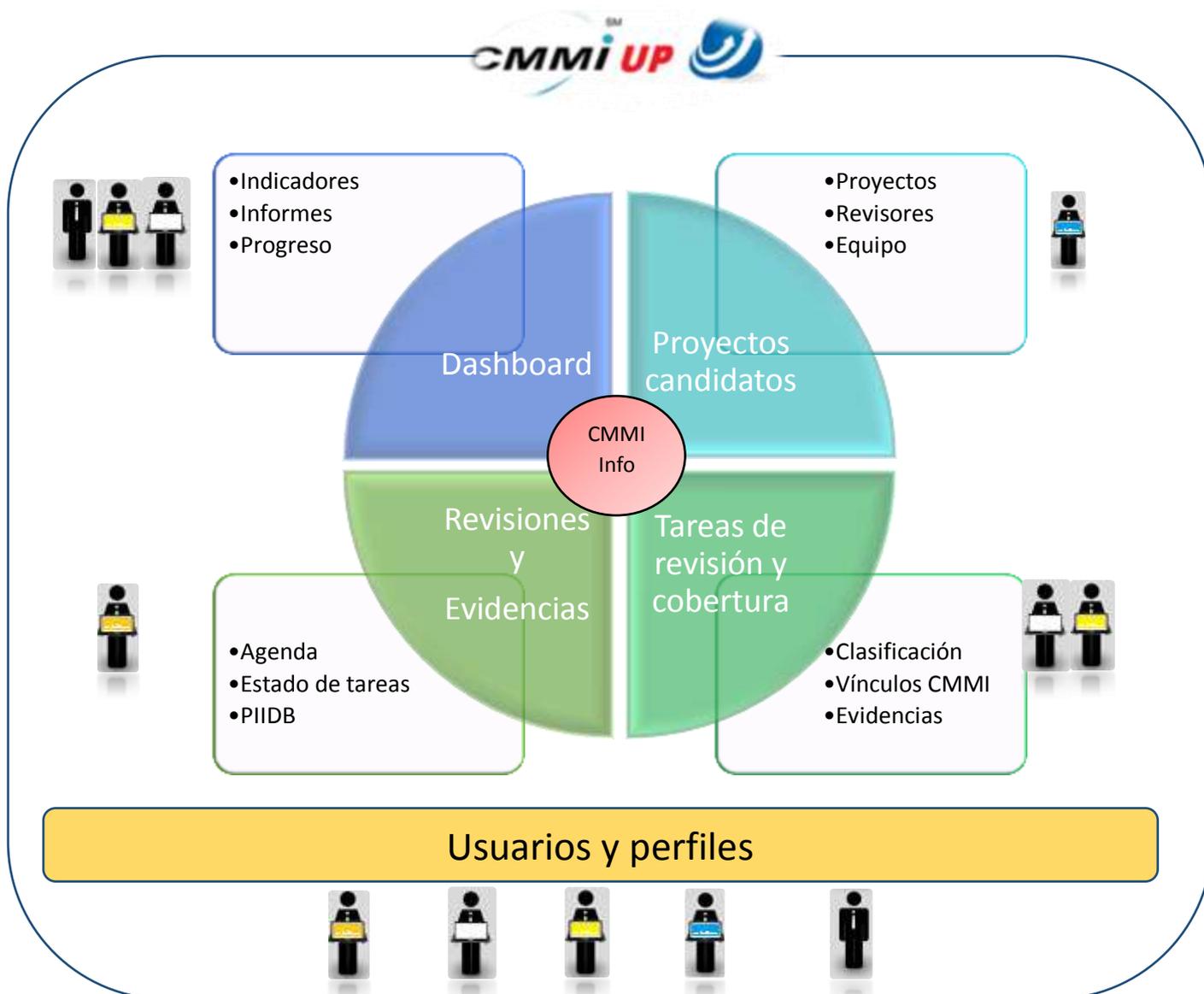


Ilustración 11 Componentes CMMI UP

## Subsistema de gestión de usuarios

### Descripción:

El subsistema de gestión de usuarios proporciona las funcionalidades básicas para realizar el mantenimiento de los usuarios. Estos usuarios son los que posteriormente se asignarán al rol pertinente.

### Funcionalidades:

- Mantenimiento de usuarios
- Gestionar contraseñas

## Subsistema de información CMMI: CMMI Info

### Descripción

El subsistema de información CMMI (CMMI Info) es dónde se documentan todos los procesos CMMI. El software vendrá preinstalado con toda la información acerca del modelo de madurez para cada uno de los niveles publicados hasta la fecha. No obstante, el usuario podrá dar de alta y mantener la información a su gusto.

### Funcionalidades:

Las funcionalidades principales de este subsistema son:

- Mantenimiento de niveles de madurez
- Mantenimiento de áreas de competencias
- Mantenimiento objetivos genéricos
- Mantenimiento de objetivos específicos
- Asociación de objetivos a áreas
- Asociación de áreas a niveles.

## Subsistema de gestión de proyectos candidatos

### Descripción:

En este subsistema se reúnen las funcionalidades para mantener los proyectos que van a ser revisados y los que finalmente serán auditados para la certificación CMMI.

### Funcionalidades:

- Mantenimiento de áreas departamentales
- Mantenimiento de proyectos
- Mantenimiento de equipos de trabajo
- Asociación de revisores a los proyectos.

## Subsistema de configuración de tareas de revisión y cobertura

### Descripción:

Este subsistema posee las funcionalidades para configurar los tipos de revisiones (únicas, mensuales, etc.), la descripción de las diferentes tareas y a qué objetivo del modelo CMMI da cobertura.

### Funcionalidades:

- Alta de tipos de revisiones
- Detalle de tareas
- Coberturas a los objetivos CMMI

## Subsistema de revisión y evidencias

### Descripción:

En este subsistema se proporcionan las funcionalidades para la gestión del resultado de las revisiones. Gestionar las incidencias encontradas en la revisión y comunicarlas a los diferentes equipos de proyecto son las principales actividades de este subsistema.

### Funcionalidades:

- Añadir incidencias y proponer acciones correctivas
- Comunicar a los equipos de proyecto
- Buzón de comunicación de incidencias (Equipo de proyecto y revisor).

## Subsistema de indicadores

### Descripción:

El subsistema de indicadores proporciona un cuadro de mandos con las gráficas más usadas en este tipo de implantaciones.

### Funcionalidades:

- Añadir gráfica
- Imprimir gráfica

## Documentación formal de casos de uso

La interacción del usuario con el sistema la esquematizaremos a través de la documentación formal de los casos de uso según el Lenguaje de Modelado Unificado (UML).

En esta representación se presentan una serie de atributos que se detallan a continuación.

**Actor:** Se llama así a toda entidad externa al sistema que interactúa con él. Por ejemplo, pidiendo una funcionalidad. No necesariamente tiene que ser una persona, en ocasiones se representan sistemas externos.

**Relación de asociación:** Indica la participación del actor en el caso de uso.

**Relación de extensión (extend):** Se define como la relación de dependencia entre dos casos de uso en la cual se aporta alguna funcionalidad extra. El caso de uso principal puede funcionar (aunque sin la funcionalidad extra) sin el caso de uso secundario.

**Relación de inclusión (include):** Esta propiedad es similar al anterior, a diferencia de que una relación de inclusión obliga a que el conjunto del proceso (todos los casos de uso que se comunican) sean llamados. En otras palabras, el caso de uso principal no puede funcionar si el segundo.

**Puntos de extensión:** Los puntos de extensión pueden corresponderse a la enumeración de pasos en el caso de uso base, así los casos de uso extendidos pueden extender cualquier a de dichos pasos.

**Escenario:** Representa el flujo exitoso más simple. Pueden existir escenarios alternativos o de excepción.

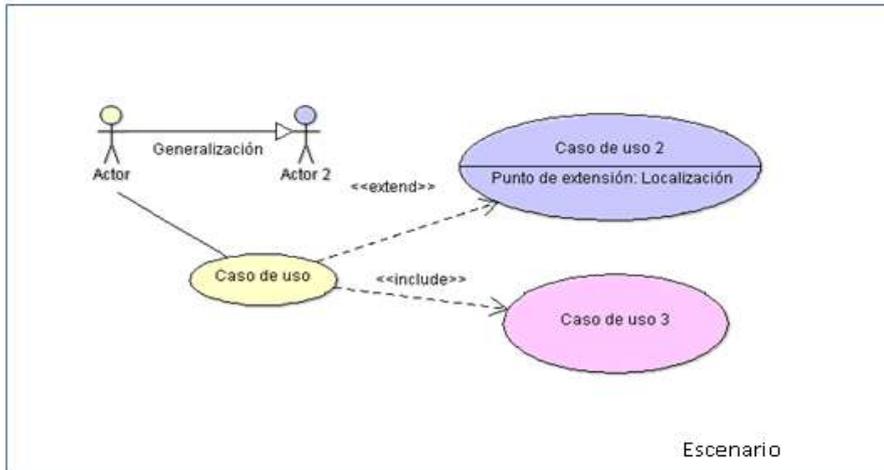


Ilustración 12 Componentes en la representación de casos de uso

En esta primera tabla realizaremos una clasificación de los casos de uso organizados por subsistema:

Caso de uso		
Subsistema	Código	Descripción
Gestión de usuarios	GU_CU_001	Conexión
	GU_CU_002	Comprobación permisos
	GU_CU_003	Lista de usuarios
	GU_CU_004	Alta de usuario
	GU_CU_005	Modificación de usuarios
	GU_CU_006	Baja de usuario

Subsistema	Código	Descripción
Información CMMI	CM_CU_001	Lista de objetivos genéricos
	CM_CU_002	Alta de objetivo genérico
	CM_CU_003	Baja de objetivo genérico
	CM_CU_004	Modificación de objetivo genérico
	CM_CU_005	Lista de prácticas genéricas
	CM_CU_006	Alta de prácticas genéricas
	CM_CU_007	Baja de prácticas genéricas
	CM_CU_008	Modificación de prácticas genéricas
	CM_CU_009	Lista de categorías de áreas de proceso
	CM_CU_010	Alta de categoría de áreas de proceso
	CM_CU_011	Baja de categorías de áreas de proceso

	CM_CU_012	Modificación de categorías de áreas de proceso
	CM_CU_013	Lista de niveles de madurez
	CM_CU_014	Alta de niveles de madurez
	CM_CU_015	Baja de niveles de madurez
	CM_CU_016	Modificación de niveles de madurez
	CM_CU_017	Lista de áreas de proceso
	CM_CU_018	Alta de áreas de proceso
	CM_CU_019	Baja de áreas de proceso
	CM_CU_020	Modificación de áreas de proceso

Subsistema	Código	Descripción
<b>Gestión de proyectos candidatos</b>	PJ_CU_001	Lista
	PJ_CU_002	Alta de roles
	PJ_CU_003	Baja de roles
	PJ_CU_004	Modificación de roles
	PJ_CU_005	Lista de departamentos
	PJ_CU_006	Alta de departamentos
	PJ_CU_007	Baja de departamentos
	PJ_CU_008	Modificación de departamentos
	PJ_CU_009	Lista de proyectos
	PJ_CU_010	Alta de proyecto
	PJ_CU_011	Baja de proyecto
	PJ_CU_012	Modificación de proyecto
	PJ_CU_013	Lista de equipos de proyecto
	PJ_CU_014	Alta de equipo de proyecto
	PJ_CU_015	Baja de equipo de proyecto
	PJ_CU_016	Modificación de equipos de proyecto

Subsistema	Código	Descripción
<b>Tareas de revisión y cobertura</b>	RV_CU_001	Lista de tipos de revisiones
	RV_CU_002	Alta de tipos de revisiones
	RV_CU_003	Baja de tipos de revisiones
	RV_CU_004	Modificación de tipo de revisiones
	RV_CU_005	Lista de tareas de revisión
	RV_CU_006	Alta de tareas de revisión

RV_CU_007	Baja de tarea de revisión
RV_CU_008	Modificación de tarea de revisión
RV_CU_009	Lista de coberturas de tarea
RV_CU_010	Alta de cobertura de tarea
RV_CU_011	Baja de cobertura de tarea
RV_CU_012	Modificación de cobertura de tarea

Subsistema	Código	Descripción
<b>Revisiones y evidencias</b>	EV_CU_001	Matriz de revisiones
	EV_CU_002	Generar revisiones
	EV_CU_003	Cambiar estado revisiones
	EV_CU_004	Añadir evidencias
	EV_CU_005	Eliminar evidencia
	EV_CU_006	Modificar evidencia
	EV_CU_007	Consultar PIIDB
	EV_CU_008	Añadir incidencia-acción correctiva
	EV_CU_009	Consultar incidencias-acciones correctivas
	EV_CU_010	Cambiar estado incidencia-acción correctiva

Subsistema	Código	Descripción
<b>Indicadores</b>	IN_CU_001	Consultar indicador/Gráfica
	IN_CU_002	Imprimir indicador/Gráfica
	IN_CU_003	Añadir indicador
	IN_CU_004	Eliminar indicador

## Subsistema de gestión de usuarios

Caso de uso	GU_CU_001
Nombre	Conexión al sistema
Requerimiento relacionado	RF_0.1
Actores	Todos los actores
Precondición	Acceder a la URL de conexión de la aplicación
Postcondición	El usuario es validado y se conocen los permisos para mostrar los menús (a través del GU_CU_002)
Escenario principal	El actor accede a la URL de la aplicación e introduce su usuario y contraseña. El sistema valida las credenciales y el actor accede al sistema con los menús a los cuales ha sido autorizado.
Escenario secundario	La validación de credenciales ha sido incorrecta. El sistema devuelve un mensaje indicando el problema.

Caso de uso	GU_CU_002
Nombre	Validar credenciales y obtener permisos
Requerimiento relacionado	RF_0.1
Actores	Todos los actores
Precondición	Ha sido llamado por el caso de uso GU_CU_001
Postcondición	El usuario es validado y se conocen los permisos. El menú de la aplicación se estructura en base a estos permisos.
Escenario principal	El actor accede a la URL de la aplicación e introduce su usuario y contraseña. El sistema valida las credenciales y el actor accede al sistema con los menús a los cuales ha sido autorizado.
Escenario secundario	La validación de credenciales ha sido incorrecta. El sistema devuelve código de error al GU_CU_001.

Caso de uso	GU_CU_003
Nombre	Lista de usuarios
Requerimiento relacionado	RF_0.1
Actores	Administrador
Precondición	Credenciales correctas y acceso al menú de usuarios.
Postcondición	Se le muestra una lista de los usuarios dados de alta en el sistema. El usuario ha gestionado (consultado, modificado o dado de baja algún usuario)
Escenario principal	El actor accede a la URL de la aplicación e introduce su usuario y contraseña. El sistema valida las credenciales y el actor accede al menú de usuarios.
Escenario secundario	-

Caso de uso	GU_CU_004
Nombre	Alta de usuario
Requerimiento relacionado	RF_0.1
Actores	Administrador
Precondición	El caso de uso ha sido llamado por GU_CU_003
Postcondición	Se ha dado de alta un nuevo usuario en el sistema
Escenario principal	El actor accede al formulario de alta de usuarios, completa los campos obligatorios y guarda el registro.
Escenario secundario	El usuario que quiere dar de alta existe. El sistema le muestra un mensaje de aviso.

Caso de uso	GU_CU_005
Nombre	Baja de usuario
Requerimiento relacionado	RF_0.1
Actores	Administrador
Precondición	El caso de uso ha sido llamado por GU_CU_003
Postcondición	El usuario se elimina (lógicamente) del sistema.
Escenario principal	EL usuario accede a la lista de usuarios, selecciona un usuario y pulsa la opción para darlo de baja.

Escenario secundario	-
Caso de uso	GU_CU_006
Nombre	Modificación de usuario
Requerimiento relacionado	RF_0.1
Actores	Administrador
Precondición	El caso de uso ha sido llamado por GU_CU_003
Postcondición	Alguna propiedad del usuario ha sido modificada.
Escenario principal	EL usuario accede a la lista de usuarios, selecciona un usuario y pulsa la opción para editarlo. Posteriormente lo guarda
Escenario secundario	-

Diagrama de casos de uso del subsistema de conexión

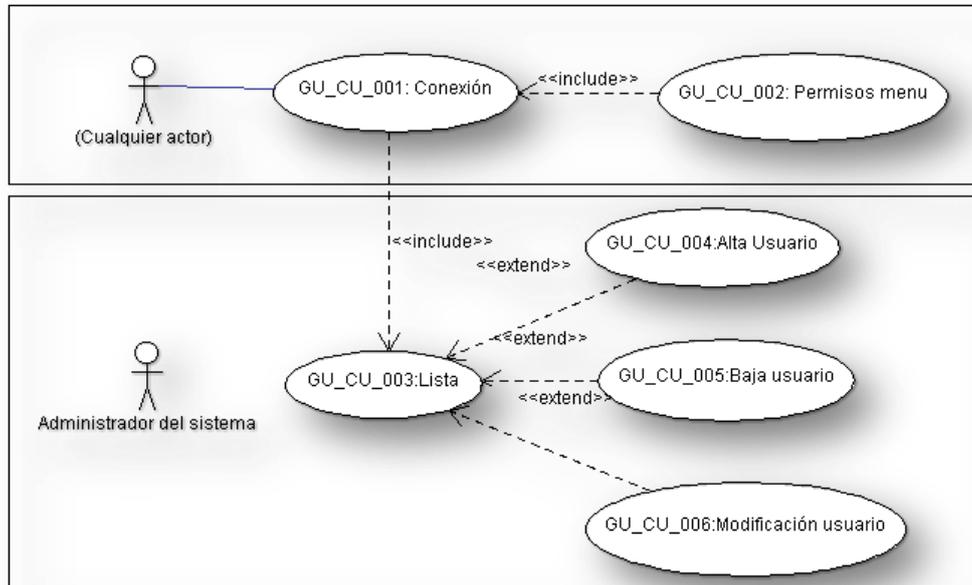


Ilustración 13 Casos de uso subsistema conexión

## Subsistema de información CMMI

\* Todos los casos de uso del subsistema CMMI son similares. Se detalla un ejemplo de mantenimiento de una entidad.

Caso de uso	CM_CU_001
Nombre	Lista de objetivos genéricos
Requerimiento relacionado	RF_1.1
Actores	Experto CMMI
Precondición	Credenciales correctas y acceso al menú Lista de o.g.
Postcondición	Se le muestra una lista de los objetivos genéricos definidos en el sistema. El usuario ha gestionado (consultado, modificado o dado de baja algún objetivo genérico)
Escenario principal	El actor accede a la URL de la aplicación e introduce su usuario y contraseña. El sistema valida las credenciales y el actor accede al menú CMMI – Objetivos genéricos.
Escenario secundario	-

Caso de uso	CM_CU_002
Nombre	Alta de objetivo genérico
Requerimiento relacionado	RF_1.1
Actores	Experto CMMI
Precondición	El caso de uso ha sido llamado por CM_CU_001
Postcondición	Se ha dado de alta un nuevo objetivo genérico
Escenario principal	El actor accede al formulario de alta de objetivos genéricos, completa los campos obligatorios y guarda el registro.
Escenario secundario	El objetivo genérico que quiere dar de alta existe. El sistema le muestra un mensaje de aviso.

Caso de uso	CM_CU_003
Nombre	Baja de objetivo genérico
Requerimiento relacionado	RF_1.1
Actores	Administrador
Precondición	El caso de uso ha sido llamado por CM_CU_001
Postcondición	El objetivo genérico se elimina (lógicamente) del sistema
Escenario principal	EL usuario accede a la lista de objetivos genéricos, selecciona uno y pulsa la opción para darlo de baja.
Escenario secundario	-

Caso de uso	CM_CU_004
Nombre	Modificación de un objetivo genérico
Requerimiento relacionado	RF_1.1
Actores	Experto CMMI
Precondición	El caso de uso ha sido llamado por CM_CU_001
Postcondición	Alguna propiedad del objetivo genérico ha sido modificada.
Escenario principal	EL usuario accede a la lista de objetivos genéricos, selecciona uno y pulsa la opción para editarlo. Posteriormente lo guarda
Escenario secundario	-

Diagrama de casos de uso del subsistema CMMI

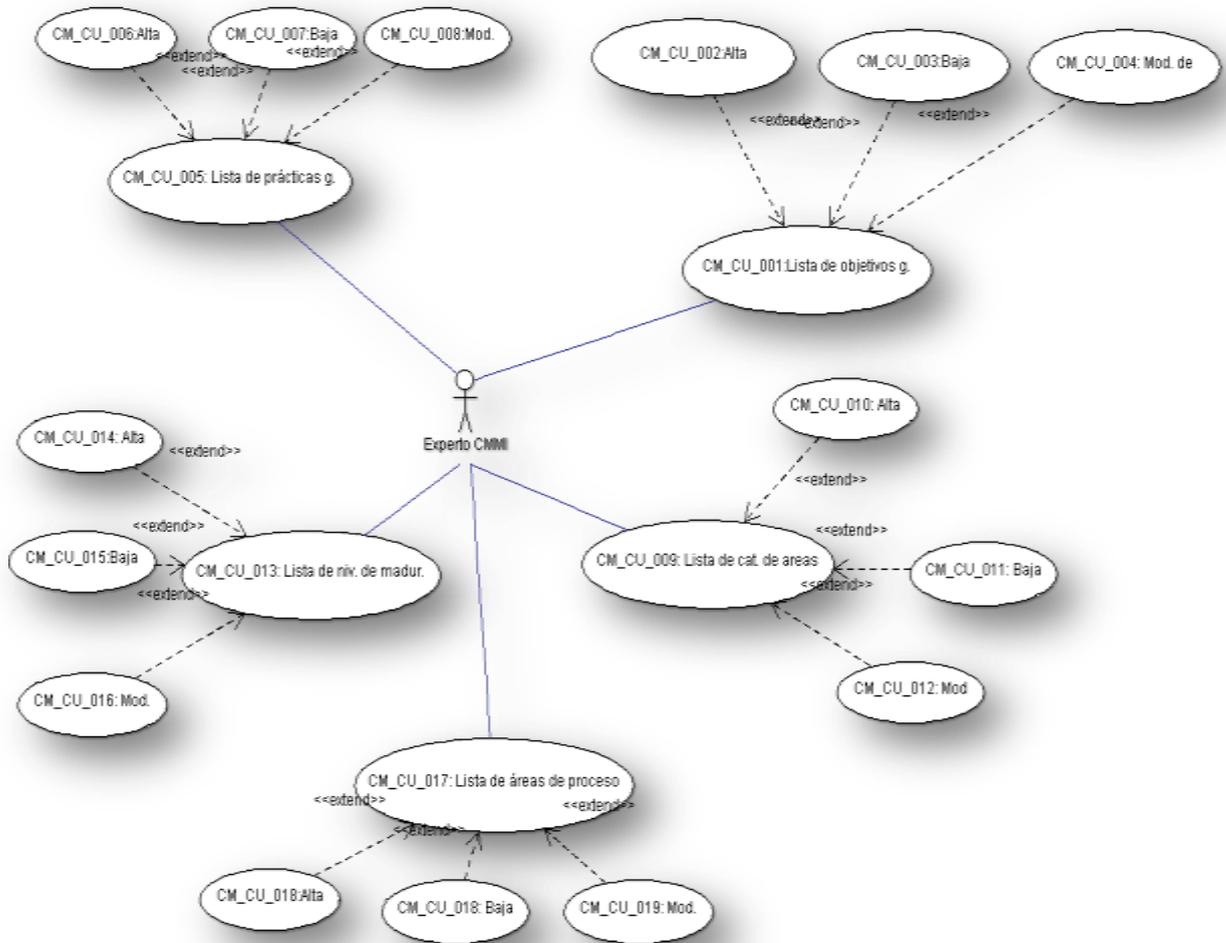


Ilustración 14 Casos de uso del subsistema CMMI

## Subsistema de gestión de proyectos candidatos

\* Todos los casos de uso del subsistema CMMI son similares. Se detalla un ejemplo de mantenimiento de una entidad.

Caso de uso	PJ_CU_009
Nombre	Lista de proyectos candidatos
Requerimiento relacionado	RF_2.1
Actores	Administrador, Líder equipo CMMI
Precondición	Credenciales correctas y acceso al menú Lista de proyectos
Postcondición	Se le muestra una lista de los proyectos definidos en el sistema. El usuario ha gestionado (consultado, modificado o dado de baja algún proyectos )
Escenario principal	El actor accede a la URL de la aplicación e introduce su usuario y contraseña. El sistema valida las credenciales y el actor accede al menú CMMI – proyectos candidatos
Escenario secundario	-

Caso de uso	PJ_CU_010
Nombre	Alta de proyectos candidatos
Requerimiento relacionado	RF_2.1
Actores	Administrador, Líder equipo CMMI
Precondición	El caso de uso ha sido llamado por PJ_CU_001
Postcondición	Se ha dado de alta un nuevo proyectos candidatos
Escenario principal	El actor accede al formulario de alta de proyectos candidatos, completa los campos obligatorios y guarda el registro.
Escenario secundario	El proyecto candidatos que quiere dar de alta existe. El sistema le muestra un mensaje de aviso.

Caso de uso	PJ_CU_011
Nombre	Baja de proyectos candidatos
Requerimiento relacionado	RF_2.1
Actores	Administrador, Líder equipo CMMI
Precondición	El caso de uso ha sido llamado por PJ_CU_001
Postcondición	El proyecto candidatos se elimina (lógicamente) del sistema
Escenario principal	EL usuario accede a la lista proyectos candidatos, selecciona uno y pulsa la opción para darlo de baja.
Escenario secundario	-

Caso de uso	PJ_CU_012
Nombre	Modificación de un proyecto candidato
Requerimiento relacionado	RF_2.1
Actores	Administrador, Líder equipo CMMI
Precondición	El caso de uso ha sido llamado por PJ_CU_001
Postcondición	Alguna propiedad del proyecto candidatos ha sido modificada.
Escenario principal	EL usuario accede a la lista de proyectos candidatos, selecciona uno y pulsa la opción para editarlo. Posteriormente lo guarda
Escenario secundario	-

Diagrama de casos de uso del subsistema de proyectos candidatos

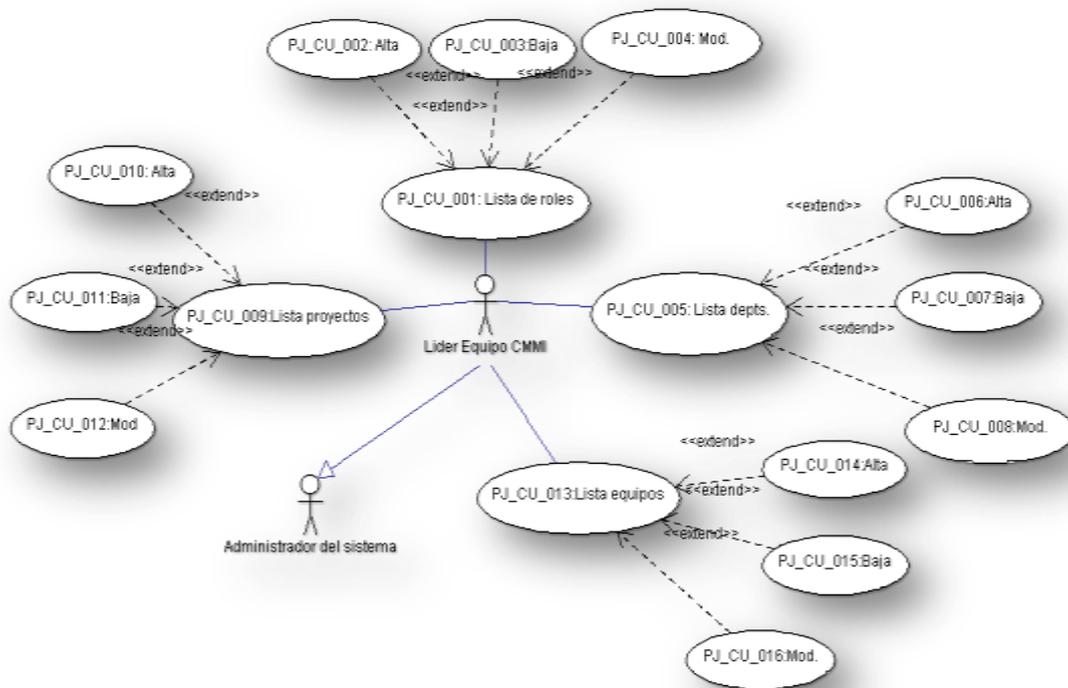


Ilustración 15 Casos de uso de Proyectos

Subsistema de configuración de tareas y coberturas

\* Todos los casos de uso del subsistema CMMI son similares. Se detalla un ejemplo de mantenimiento de una entidad.

Caso de uso	RV_CU_001
Nombre	Lista de tipos de revisiones
Requerimiento relacionado	RF_3.1
Actores	Administrador, Revisor CMMI
Precondición	Credenciales correctas y acceso al menú Lista de tipos de revisiones.
Postcondición	Se le muestra una lista de los tipos de revisiones definidos en el sistema. El usuario ha gestionado (consultado, modificado o dado de baja algún tipo de revisión)
Escenario principal	El actor accede a la URL de la aplicación e introduce su usuario y contraseña. El sistema valida las credenciales y el actor accede al menú CMMI – Tipos de revisiones.

Escenario secundario	-
----------------------	---

Caso de uso	RV_CU_002
Nombre	Alta de un tipo de revisión
Requerimiento relacionado	RF_3.1
Actores	Administrador, Revisor CMMI
Precondición	El caso de uso ha sido llamado por RV_CU_001
Postcondición	Se ha dado de alta un nuevo tipo de revisión
Escenario principal	El actor accede al formulario de alta de tipos de revisiones, completa los campos obligatorios y guarda el registro.
Escenario secundario	El tipo de revisión que quiere dar de alta existe. El sistema le muestra un mensaje de aviso.

Caso de uso	RV_CU_003
Nombre	Baja de tipo de revisión
Requerimiento relacionado	RF_3.1
Actores	Administrador, Revisor CMMI
Precondición	El caso de uso ha sido llamado por RV_CU_001
Postcondición	El tipo de revisión se elimina (lógicamente) del sistema
Escenario principal	EL usuario accede a la lista de tipos de revisiones, selecciona uno y pulsa la opción para darlo de baja.
Escenario secundario	-

Caso de uso	RV_CU_004
Nombre	Modificación de un tipo de revisión
Requerimiento relacionado	RF_3.1
Actores	Administrador, Revisor CMMI
Precondición	El caso de uso ha sido llamado por RV_CU_001
Postcondición	Alguna propiedad del tipo de revisión ha sido modificada.
Escenario principal	EL usuario accede a la lista de tipos de revisión, selecciona uno y pulsa la opción para editarlo. Posteriormente lo guarda
Escenario secundario	-

Diagrama de casos de uso del subsistema de tareas y coberturas

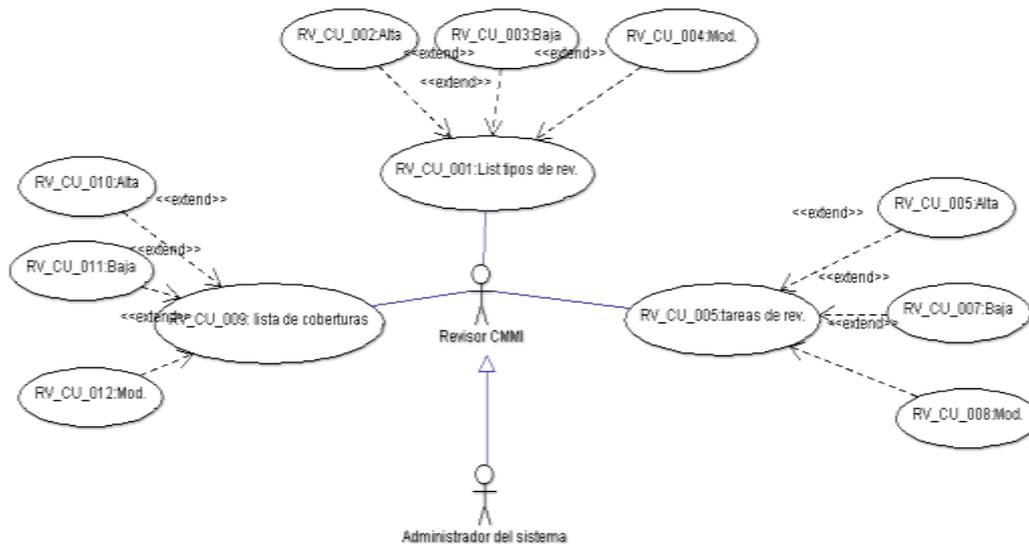


Ilustración 16 Casos de uso tareas y coberturas

## Subsistema de revisiones y evidencias

\* Se describen los más representativos.

Caso de uso	EV_CU_001
Nombre	Matriz de revisiones
Requerimiento relacionado	RF_4.3
Actores	Revisor CMMI
Precondición	Credenciales correctas y acceso al menú Matriz de revisiones
Postcondición	El revisor CMMI ha podido consultar una matriz en la que figuran los diferentes proyectos y las tareas que requieren revisión. Cada una de las tareas está identificada con un color para diferenciar el estado (Pendiente de revisión, Correcta, Pendiente de acción correctiva o No evaluable. Los diferentes registros de revisiones tienen disponible acciones: Asignar/modificar/eliminar una evidencia a la revisión, Generar más revisiones o Cambiar el estado de una revisión.
Escenario principal	El actor accede a la URL de la aplicación e introduce su usuario y contraseña. El sistema valida las credenciales y el actor accede a la matriz de revisiones. En esta matriz, el revisor puede consultar los diferentes proyectos y las tareas que requieren revisión. Cada una de las tareas está identificada con un color para diferenciar el estado (Pendiente de revisión, Correcta, Pendiente de acción correctiva o No evaluable. Los diferentes registros de revisiones tienen disponible acciones: Asignar/modificar/eliminar una evidencia a la revisión, Generar más revisiones o Cambiar el estado de una revisión
Escenario secundario	-

Caso de uso	EV_CU_002
Nombre	Generar revisiones
Requerimiento relacionado	RF_4.3
Actores	Revisor CMMI
Precondición	El caso de uso ha sido llamado por EV_CU_001
Postcondición	Las tareas de revisión han sido generadas.
Escenario principal	<p>El revisor entra en la pantalla de matriz de tareas y selecciona la opción de este caso de uso. A continuación especifica los criterios para los cuales quiere generar la matriz de revisiones. Entre ellos:</p> <ul style="list-style-type: none"> <li>• Intervalo de fechas</li> <li>• Proyectos incluidos</li> <li>• Área de cobertura (por ejemplo, sólo tareas que pertenezcan al área de gestión de proyectos).</li> </ul>
Escenario secundario	

Caso de uso	EV_CU_003
Nombre	Cambiar estado de una revisión
Requerimiento relacionado	RF_4.3
Actores	Revisor CMMI
Precondición	El caso de uso ha sido llamado por EV_CU_001
Postcondición	Se ha cambiado el estado de una revisión
Escenario principal	<p>El revisor accede a la matriz de revisiones. A continuación escoge un registro y selecciona este caso de uso. Se le proporciona una pantalla de tipo detalle con la posibilidad de cambiar el estado a uno de los posibles:</p> <ul style="list-style-type: none"> <li>- Pendiente de revisión (estado inicial)</li> <li>- Pendiente de acción correctiva</li> <li>- Correcta</li> <li>- Incorrecta</li> <li>- No procede revisión</li> </ul>
Escenario secundario	-

Caso de uso	EV_CU_004
Nombre	Añadir una evidencia
Requerimiento relacionado	RF_4.3
Actores	Revisor CMMI
Precondición	El caso de uso ha sido llamado por EV_CU_001
Postcondición	Se ha añadido una evidencia
Escenario principal	<p>El revisor accede a la matriz de revisiones. A continuación escoge un registro y selecciona este caso de uso. Se le proporciona una pantalla de tipo detalle con la posibilidad de añadir los siguientes datos:</p> <ul style="list-style-type: none"> <li>- URL a una dirección que proporcione una visión de la evidencia</li> <li>- Una carpeta dentro de la red donde exista la evidencia</li> <li>- Un comentario del revisor</li> </ul>
Escenario secundario	-

Caso de uso	EV_CU_008
Nombre	Añadir una incidencia y acción correctiva
Requerimiento relacionado	RF_4.3
Actores	Revisor CMMI
Precondición	El caso de uso ha sido llamado por EV_CU_001 o EV_CU_009
Postcondición	Se ha añadido una nueva incidencia asociada a la revisión.
Escenario principal	<p>El revisor accede a la matriz de revisiones o al buzón de incidencias. A continuación escoge un registro y selecciona este caso de uso. Se le proporciona una pantalla de tipo detalle (con la información de la tarea más relevante) con la posibilidad de añadir los siguientes datos:</p> <ul style="list-style-type: none"> <li>- Descripción de la incidencia</li> <li>- Descripción de la acción correctiva</li> <li>- Destinatario de la incidencia (debe ser un miembro del equipo para el cual se está efectuando la revisión)</li> </ul>
Escenario secundario	-

Caso de uso	EV_CU_009
Nombre	Buzón de incidencias
Requerimiento relacionado	RF_4.3
Actores	Revisor CMMI, Miembro de un equipo de proyecto
Precondición	
Postcondición	El usuario consulta, cambia de estado o añade una nueva incidencia o responde a una incidencia creada por el revisor
Escenario principal	<p>Si el revisor accede a esta funcionalidad, se le muestran todas las incidencias que él ha reportado en todos los proyectos. A partir de ahí podrá acceder a añadir nuevas o cambiar el estado.</p> <p>Para un miembro de proyecto, únicamente se le mostrarán las incidencias que hayan sido asignadas a él para que pueda responderlas o reasignarlas a otro miembro de su proyecto.</p>
Escenario secundario	-

Diagrama de casos de uso del subsistema de revisiones y evidencias.

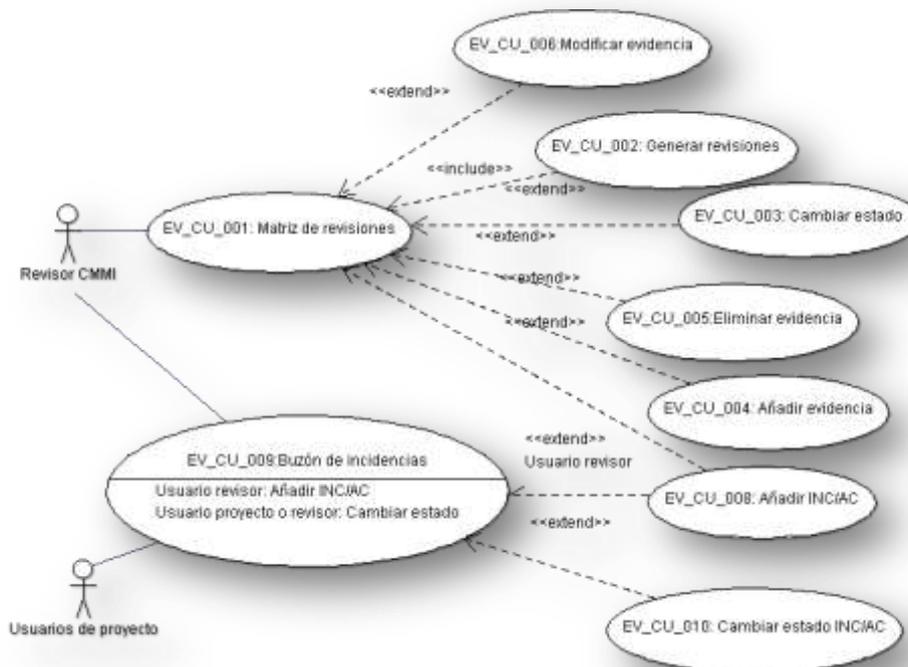


Ilustración 17 Diagrama de casos de uso del subsistema de revisiones y acciones correctivas

### Subsistema de cuadro de mandos e indicadores

Caso de uso	IN_CU_001
Nombre	Dashboard de indicadores
Requerimiento relacionado	RF_4.3
Actores	Líder CMMI, Administrador
Precondición	Disponer de permisos en la opción
Postcondición	El usuario puede visualizar diferentes gráficas sobre el estado de la implantación
Escenario principal	El líder de la implantación CMMI accede a la aplicación y en la primera pantalla se le muestra una serie de gráficas que él mismo ha seleccionado para tener en la pantalla del dashboard. Se trata de un dashboard adaptable mediante portlets. Las graficas iniciales de este proyecto son:

Escenario secundario

- Estado de las revisiones (Gráfico tarta con porcentajes: Revisado, pendientes, En espera)
- Nº de incidencias vs Número de revisiones
- Gráfico de nivel de cobertura en la organización
- Gráfico de cumplimiento de tareas por proyecto

Diagrama de casos de uso del subsistema de cuadro de mandos.

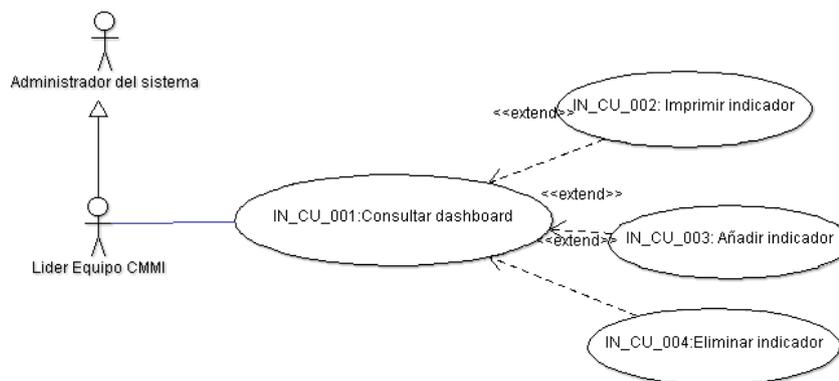


Ilustración 18 Casos de uso Cuadro de mandos

## Modelo estático: Diagrama de clases

### Modelo CMMI DEV 1.3

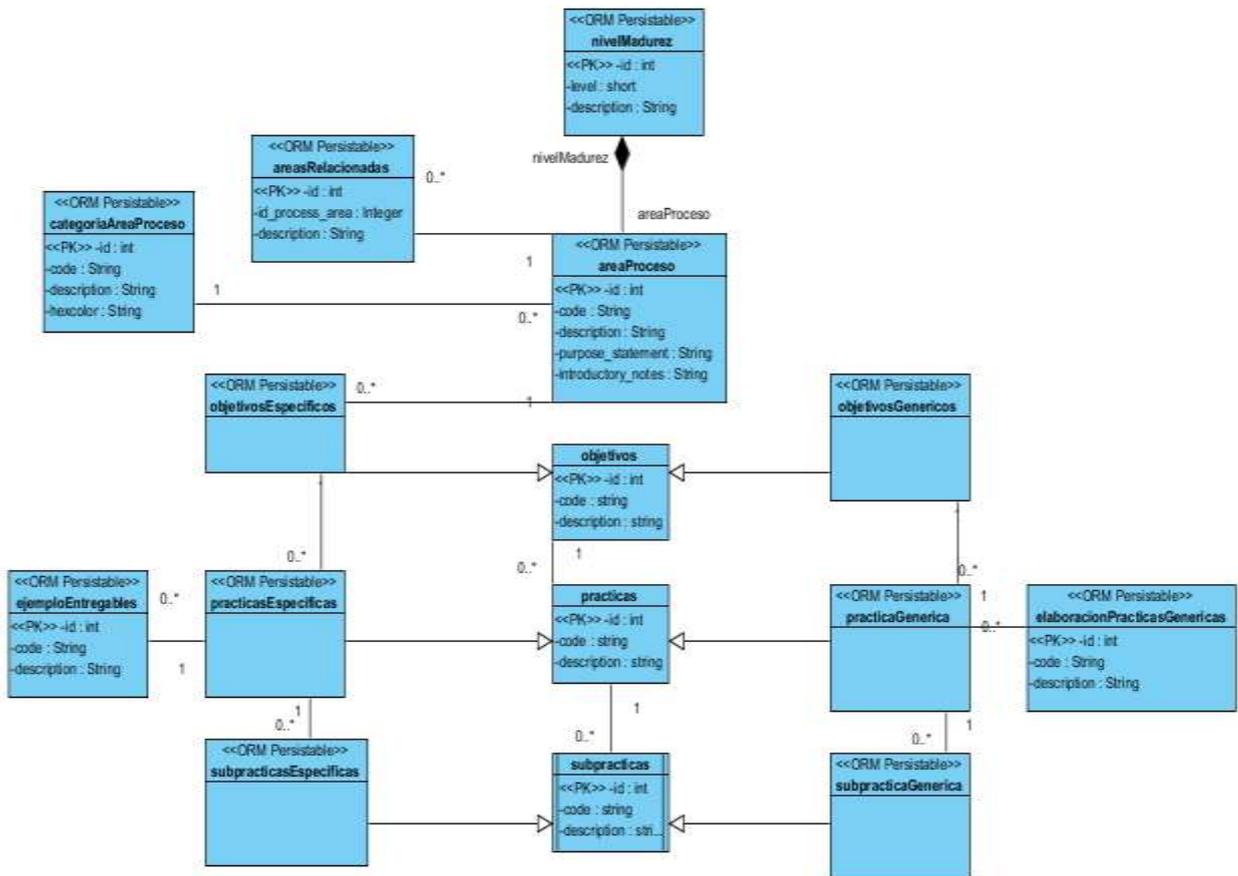


Ilustración 19 Diagrama de clases CMMI

## Documentación de ejemplo

El modelo expuesto anteriormente debe ser capaz de almacenar esta información con la misma estructura.

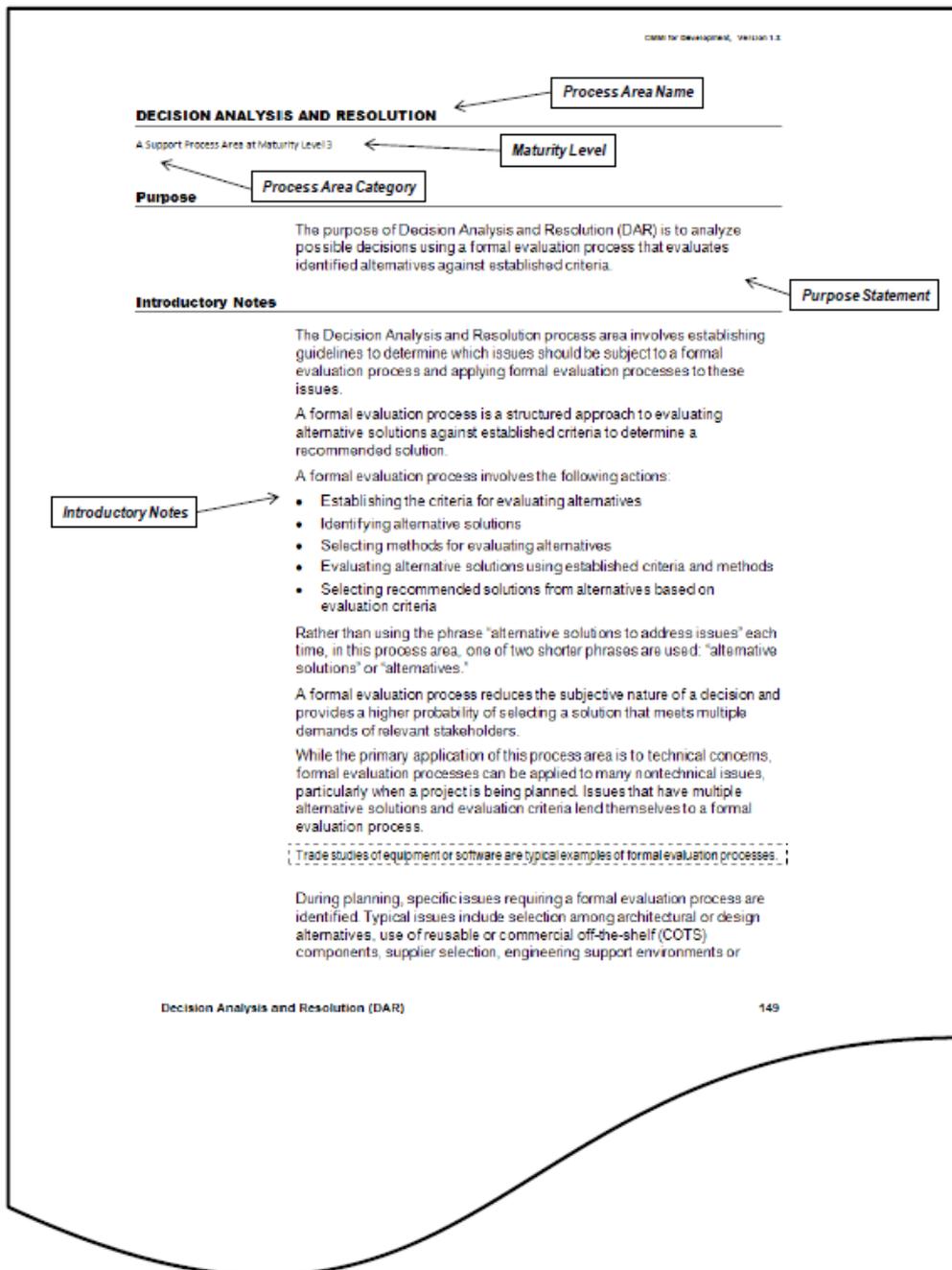


Ilustración 20 Ejemplo información CMMI

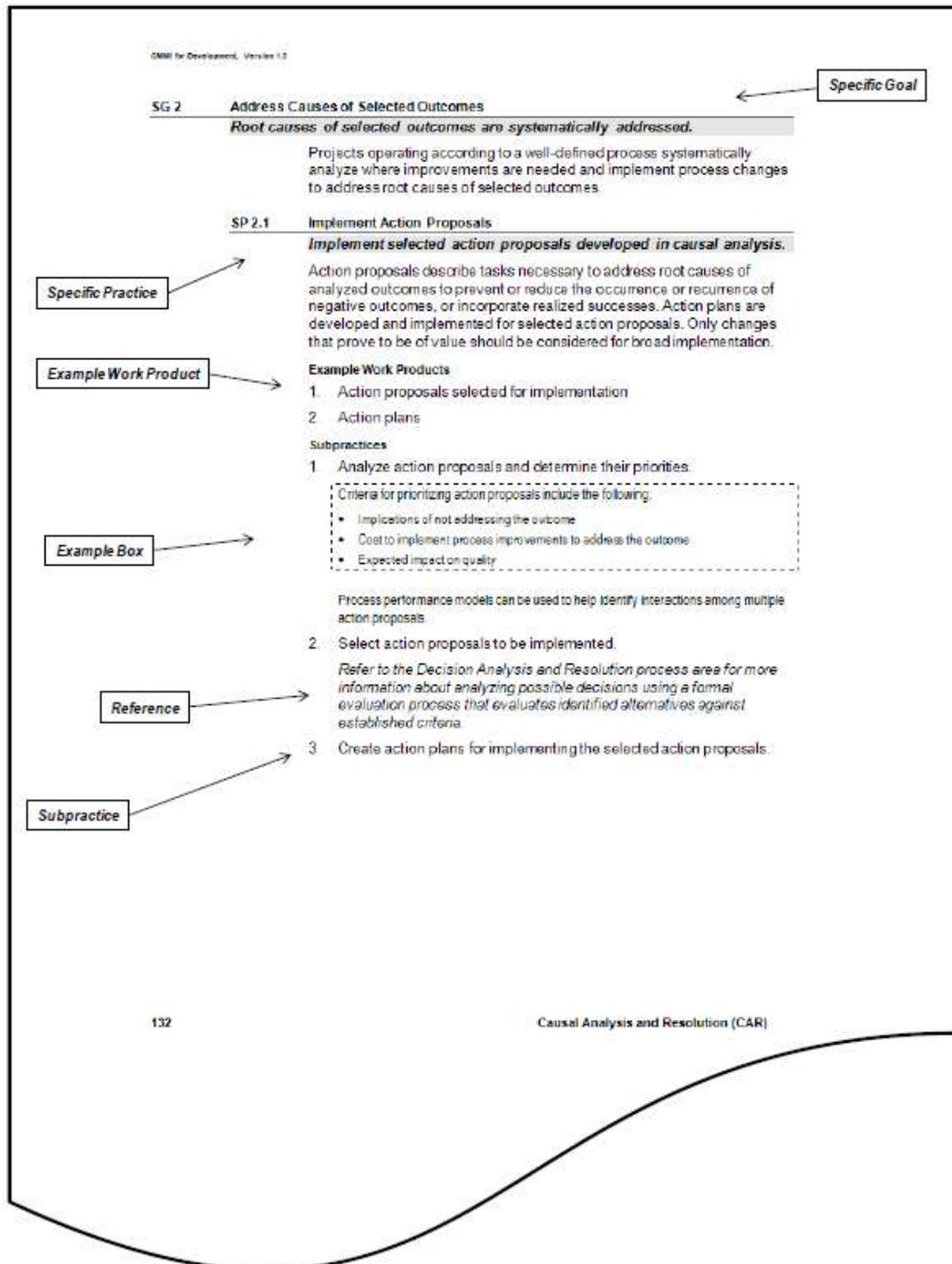


Ilustración 21 Ejemplo Información CMMI (Cont.)

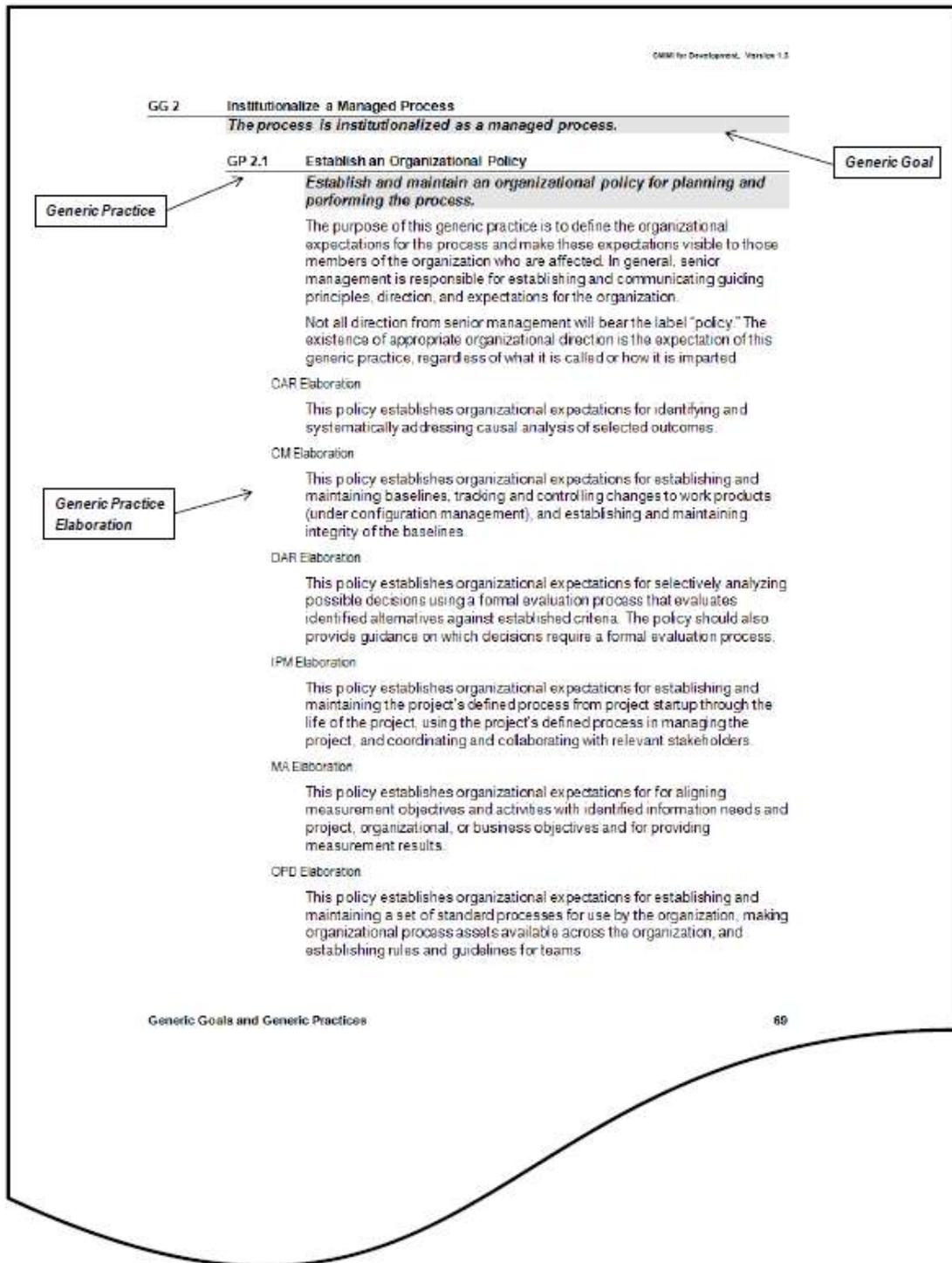


Ilustración 22 Ejemplo Información CMMI (Cont.)

Coberturas, tareas de revisión, proyectos e incidencias

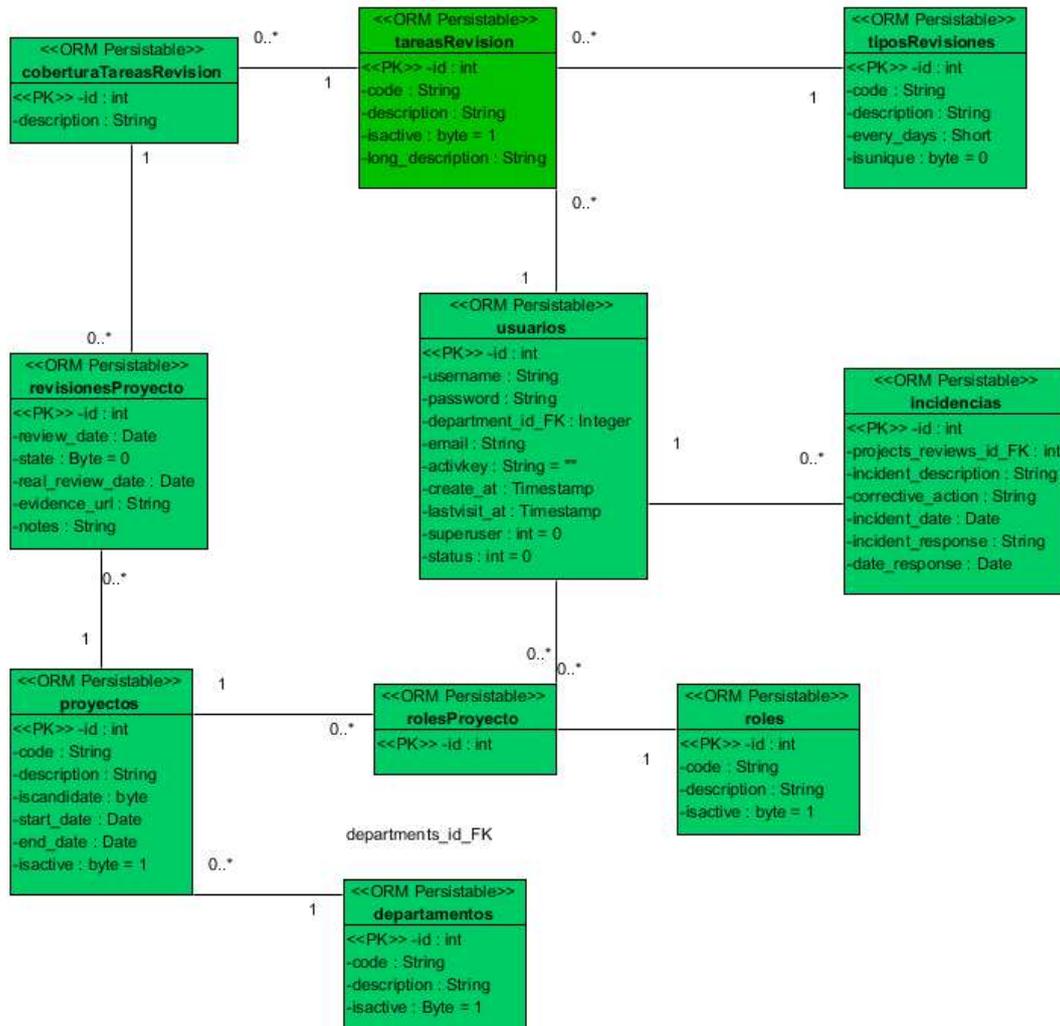


Ilustración 23 Diagrama de clases Revisiones

## Diseño del sistema

---

Una vez nuestro producto ha sido analizado desde el punto de vista de requisitos deberemos marcar las directivas tecnológicas para poder desarrollarlo. A esta etapa en el ciclo de vida se le denomina 'Diseño técnico'.

Hemos visto, en la etapa de análisis, que existen una serie de requisitos no funcionales que de alguna manera ya pueden marcar la tecnología que deberemos usar, las restricciones o límites del sistema, a qué aspecto debemos dar prioridad y a cual podemos dejar en un segundo plano o aspectos de tipo cualitativo que pueden tenerse en cuenta a la hora de tomar las decisiones de arquitectura y tecnológicas. Otro aspecto a considerar y que no está dentro de este análisis es el aspecto económico, pues éste lo trataremos en un apartado diferente.

Vamos a considerar cada una de las partes definidas en la fase anterior y realizar una correspondencia tecnológica. De esta manera enlazaremos 'qué' queremos hacer con el 'cómo' lo haremos'.

A continuación mencionamos una serie de características que hemos considerado en este diseño para que el resultado sea el más óptimo posible:

1. Diseño económico

Como hemos mencionado anteriormente, nuestro diseño no prioriza los aspectos económicos para el estudio de diseño. Sin embargo, nuestro proyecto no presenta ninguna característica diferencial que pueda incidir en costes altos. El uso de tecnología 'libre' que esté al alcance de todos permite tomar decisiones eficaces y con presupuestos totalmente asequibles.

2. Diseño eficaz

Hemos intentado diseñar todas las funcionalidades descritas en la fase de análisis para garantizar el cumplimiento y la eficacia global del sistema. No obstante, por restricciones temporales hemos dejado algunas funcionalidades menos importantes sin diseñar. Esto no supone un problema dado que estas funcionalidades corresponden a mantenimientos similares y deben implementarse de igual manera al resto de mantenimientos.

### 3. Diseño eficiente

Las funcionalidades se diseñarán siguiendo los procedimientos marcados por las buenas prácticas y siguiendo las metodologías de diseño orientada a objetos. De esta manera se garantiza que la respuesta al usuario sea la más correcta posible en términos funcionales y de rendimiento.

### 4. Diseño robusto

La arquitectura escogida, como veremos a continuación, permite dotar al sistema de la robustez suficiente como para soportar la magnitud de usuarios y datos necesarios y en el entorno que se usará.

### 5. Diseño fiable

Siguiendo la metodología de diseño orientado a objetos podemos garantizar que obtendremos un producto con un alto grado de fiabilidad. No obstante, llevando a cabo las pruebas unitarias y las pruebas de casos de uso, podríamos obtener un 100%. Cabe mencionar en este caso, que el marco de nuestro proyecto no contempla el desarrollo ni un plan de test en cualquiera de sus niveles (alto nivel, pruebas unitarias, pruebas de caso de uso, etc.).

### 6. Diseño fácil de mantener

Las buenas prácticas en el desarrollo de software, las metodologías de diseño orientado a objetos y la arquitectura escogida nos permiten garantizar un sistema dotado de un alto grado de facilidad en términos de mantenimiento. Veremos que la arquitectura 3 capas nos permite separar la presentación, la lógica de negocio y la capa de base de datos, y de esta manera facilitar, entre otros, aspectos de mantenimiento.

### 7. Diseño multiplataforma

Las decisiones tecnológicas escogidas, como veremos a continuación, nos permiten adaptar nuestro producto a múltiples plataformas.

### 8. Diseño fácil de usar

Uno de los aspectos claves es la facilidad de uso. La aplicación de directrices de usabilidad que se han estudiado en la carrera son aplicadas en este proyecto.

## Decisiones de arquitectura

Entre las necesidades no funcionales de la solución, junto con algunos aspectos que hemos considerado importantes en esta fase, vamos a destacar algunos:

- Solución WEB.
- Generalmente se usará en un entorno Intranet.
- Puede llegar a integrarse con otras aplicaciones (Aplicaciones Ticketing para incidencias o o LDAP para la gestión de usuarios).
- Fácil de mantener
- Económica, dado que es un coste indirecto en el marco de un proyecto de implantación. Es una herramienta que dará apoyo, no puede ser más cara la herramienta que el propio proyecto.

De entre las soluciones que hemos estudiado, hemos decidido adoptar la siguiente tecnología:

### Patrón de diseño Modelo Vista Controlador (MVC)

El patrón MVC, introducido como parte de la versión SmallTalk-80 del lenguaje de programación SmallTalk, presenta las siguientes ventajas:

- Clara separación entre los componentes permitiendo la implementación por separado.
- La conexión entre el modelo y sus vistas es dinámico, lo cual permite que los cambios se produzcan en tiempo de ejecución, no de compilación.

El patrón MVC, como hemos mencionado, nos permite separar la presentación, la lógica de negocio y los datos en tres componentes distintos. Este componente es adecuado para entornos Web donde la vista es la propia página HTML, el modelo es el Sistema de Gestión de Base de Datos (SGBD) y el controlador es el responsable de recibir los eventos de entrada de la vista.

En más detalle, disponemos de los siguientes componentes:

- **Modelo:** Es la representación específica de la información con la que interactúa el sistema. Gestiona los datos y controla todas sus transformaciones
- **Vista:** El modelo que hemos extraído a través de su controlador, es presentado al usuario a través de este componente.
- **Controlador:** Los eventos del sistema, generalmente iniciados por el usuario, son gestionados por este componente, que a su vez invoca a llamadas al controlador o incluso a la vista.

### Lenguaje de desarrollo: PHP

Uno de los lenguajes de programación más usados para el desarrollo web es el lenguaje PHP.

PHP fue creado originalmente por Rasmus Lerdof en 1995 y dispone de una licencia con su propio nombre. Esta licencia, de acuerdo con la Free Software Foundation, es una licencia de software libre no copyleft y una licencia de código abierto según la Open Source Initiative.

En otros términos, la licencia PHP está diseñada para incentivar la distribución del código fuente. Se permite la redistribución del contenido licenciado en forma código fuente o binaria siempre y cuando se cumplan los siguientes requisitos:

1. Se incluya la declaración de los derechos de autor de la licencia PHP.
2. La palabra PHP no se use en el título de las obras derivadas.
3. Se incluya el siguiente anuncio bajo cualquier forma en la que se redistribuya el código.

```
This product includes PHP software, freely available from <http://www.php.net/software/>
```

Más allá de aspectos legales de distribución y licencia, vamos a describir algunas de sus características más importantes y las cuales encajan perfectamente con los requisitos de nuestra solución.

- Orientado al desarrollo de aplicaciones Web.
- Es considerado un lenguaje fácil de aprender.
- Permite la conexión con una gran variedad de gestores de bases de datos (SGBD's), destacando MySQL y PostgreSQL.
- Escalable por medio de extensiones (ext's).
- Libre y accesible.
- Permite la aplicación de técnicas de programación orientada a objetos.

### Sistema gestor de base de datos: MySQL

Para la persistencia de los datos, hemos escogido un sistema gestor de base de datos relacional denominado MySQL.

MySQL está patrocinado por una empresa privada (Sun Microsystems) que posee el copyright y de la mayor parte del código. Por un lado, se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas organizaciones que necesiten incorporarlo en productos privativos deberán adquirir a la empresa una licencia específica que les permita ese uso.

Pese a que MySQL inicialmente carecía de características que se hacen importantes en los sistemas gestores de bases de datos como pueden ser la gestión de transacciones o la integridad referencial, poco a poco, y en la actualidad podemos destacar, junto a su simplicidad, algunos de los siguientes aspectos:

- Utilización del lenguaje SQL en su inmensa mayoría.
- Disponible en múltiples plataformas y sistemas
- Características para asegurar la integridad referencial (claves foráneas, transacciones).
- Conectividad segura.
- Búsqueda e indexación de campos de texto.

### Integración mediante un Framework de desarrollo: Yii

Hemos mencionado los diferentes componentes que contendrá la arquitectura de nuestro producto CMMI-UP. Ahora bien, en la actualidad, existen numerosos marcos de desarrollo que nos permiten unir los diferentes componentes de la arquitectura, y que nos ayudan a la implementación siguiendo procedimientos y metodología sobre una base conceptual y tecnológica de soporte definido. En realidad, representa la integración de todos los componentes antes mencionados, y modela las relaciones y provee una base sobre la cual implementaremos nuestra solución.

Es el caso del marco de trabajo Yii, el cual nos permitirá integrar los diferentes componentes que hemos mencionado siguiendo una metodología definida de trabajo

Entre las características del framework Yii podemos encontrar las siguientes:

- Diseño con patrones MVC
- Uso de objetos para el acceso a base de datos (DAO)
- Entrada y validación de formularios
- Widgets AJAX para múltiples usos: Autocompletar, treeviews, calendarios, etc.
- ‘Skinning ‘ y ‘theming’ ,permite customizar el interface de usuario
- Web Services para integración con sistemas
- Manejo de errores y excepciones
- Internacionalización i18N para la traducción de literales
- Generación automática de código para CRUD (Create, Update, Delete)
- Incorporación de extensiones de terceros

### Estructura estática de la aplicación

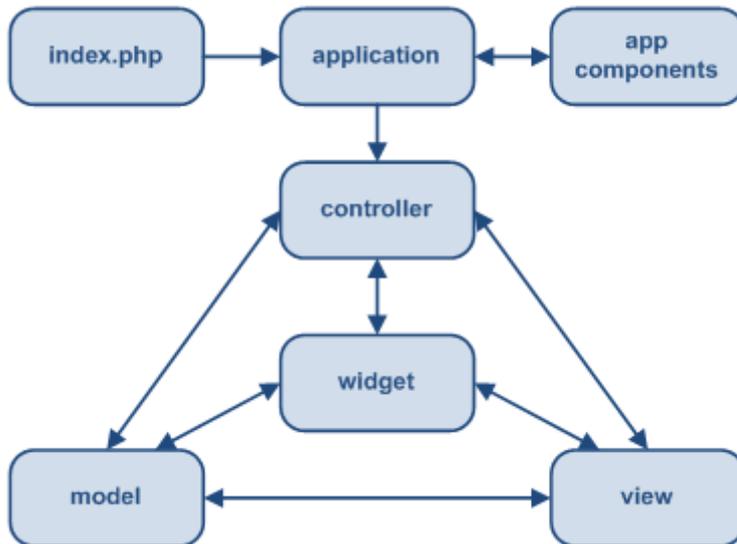


Ilustración 24 Diagrama de la estructura de la aplicación

## Flujo de trabajo genérico de la aplicación

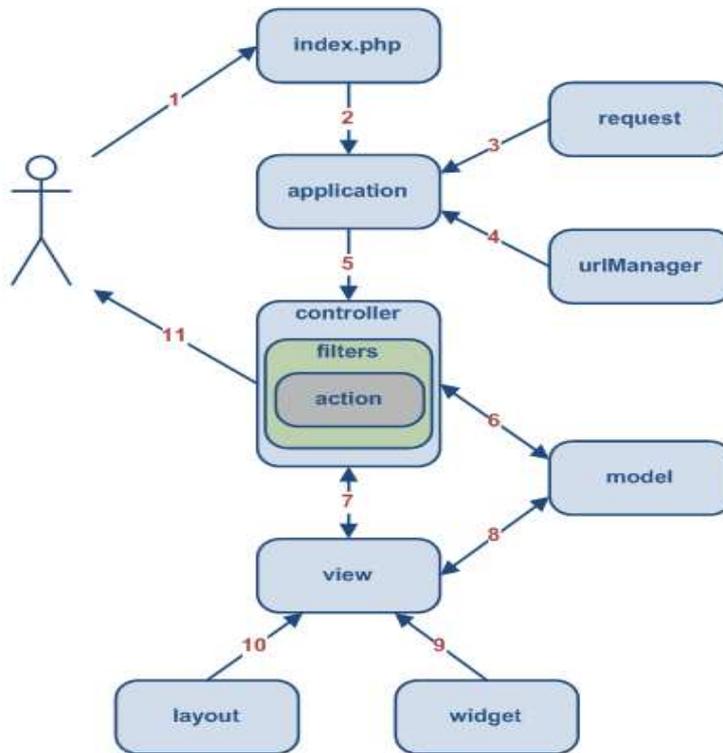


Ilustración 25 Flujo a través de los componentes

1. Un usuario realiza una solicitud a través del explorador, con la siguiente URL `http://www.cmmiup.com/index.php?r=post/show&id=1` y el servidor Web se encarga de la solicitud mediante la ejecución del script de arranque en `index.php`.
2. El script de entrada crea una instancia del componente '[application](#)' y la ejecuta.
3. La aplicación obtiene la información detallada del pedido del usuario del [componente de aplicación](#) `request`.
4. A través del componente '`uriManager`' se determina el [controlador](#) y la [acción](#) pedida .
5. La aplicación crea una instancia del controlador pedido para resolver la solicitud del usuario. El controlador determina que la acción refiere al nombre de método correspondiente en la clase controlador. Entonces crea y ejecuta los filtros asociados con esta acción (ejemplo: control de acceso, benchmarking). La acción es ejecutada si los filtros lo permiten.
6. La acción lee el [modelo](#) y accede a base de datos.
7. La acción realiza la [vista](#) llamada con el modelo.
8. La vista lee y muestra los atributos del modelo.
9. La vista ejecuta algunos [widgets](#).
10. El resultado realizado es embebido en un [esquema \(layout\)](#).
11. La acción completa la vista realizada y se la muestra al usuario.

### Esquema general de la arquitectura CMMI-UP

Una vez hemos identificado los diferentes componentes tecnológicos sobre los cuales diseñaremos e implementaremos la solución, vamos a ver un esquema general de la arquitectura CMMI-UP.

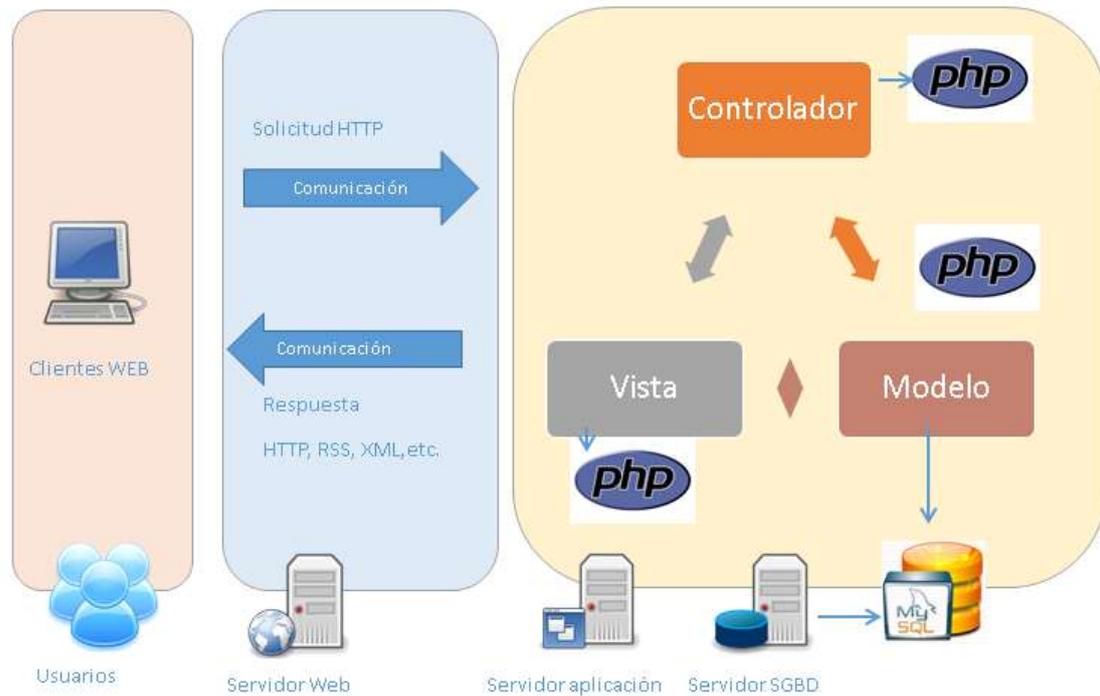


Ilustración 26 Esquema general de la arquitectura

## Modelo estático : Diagrama de clases

### Modelo CMMI DEV 1.3

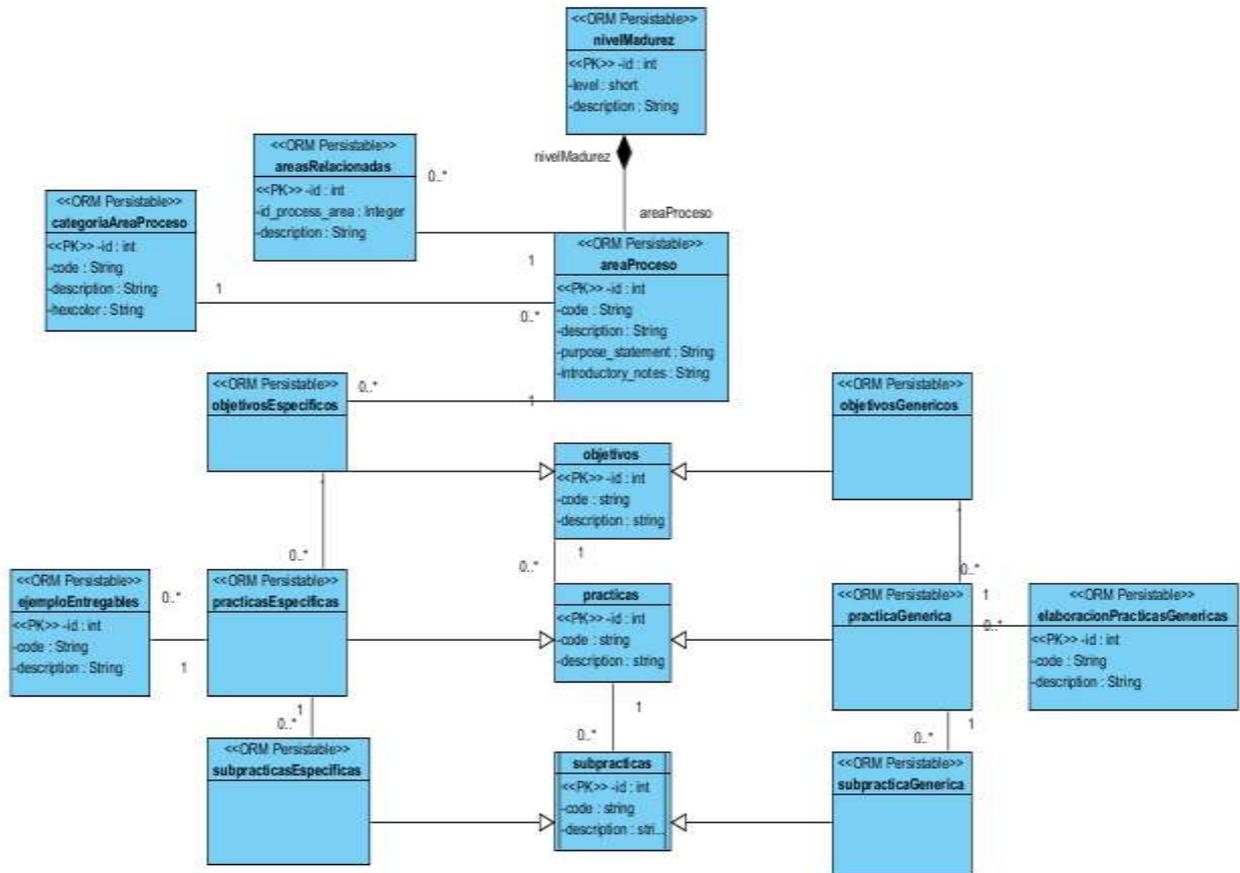


Ilustración 27 Diagrama de clases CMMI

Coberturas, tareas de revisión, proyectos e incidencias

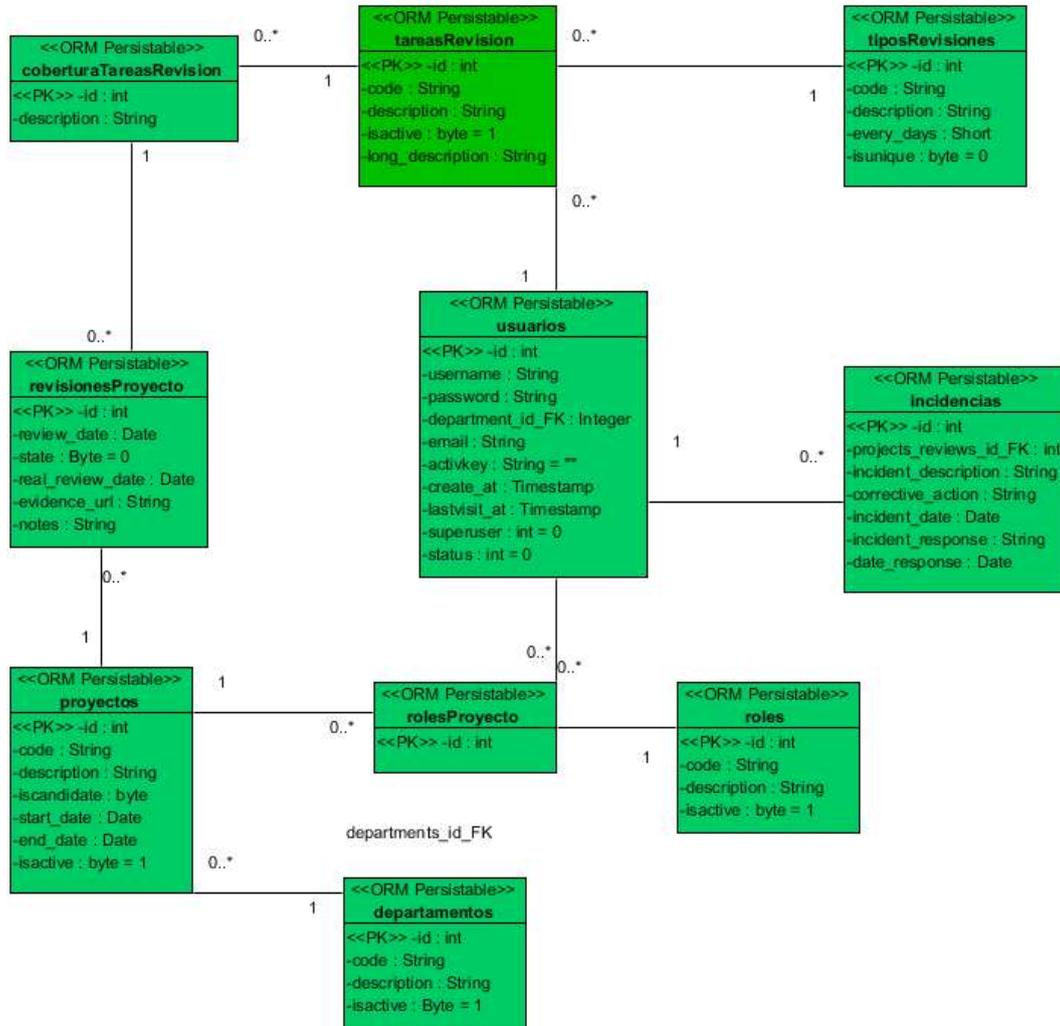
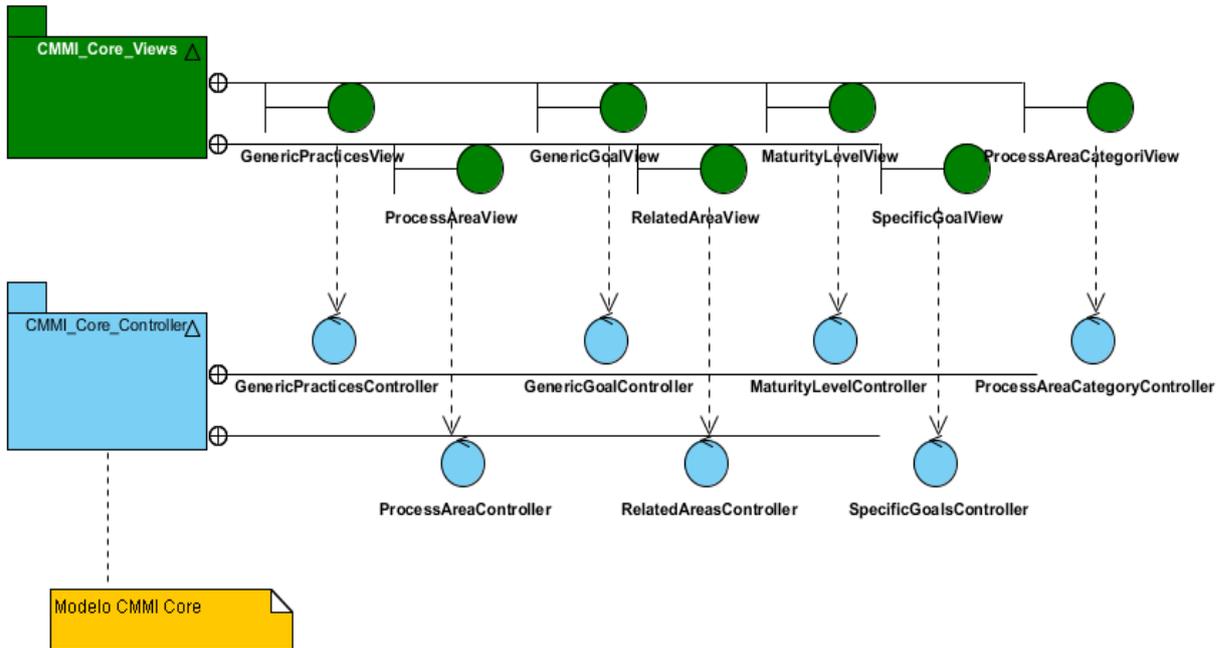


Ilustración 28 Diagrama de clases Revisiones

## Clases frontera y gestoras CMMI Core



Cabe destacar que las clases frontera de tipo CRUD (Create, Update, Delete) o ABM (Alta, Baja,

Ilustración 29 Clases frontera y gestoras CMMI Core

Modificación) se descomponen a su vez en diferentes tipologías de vistas según el propósito.

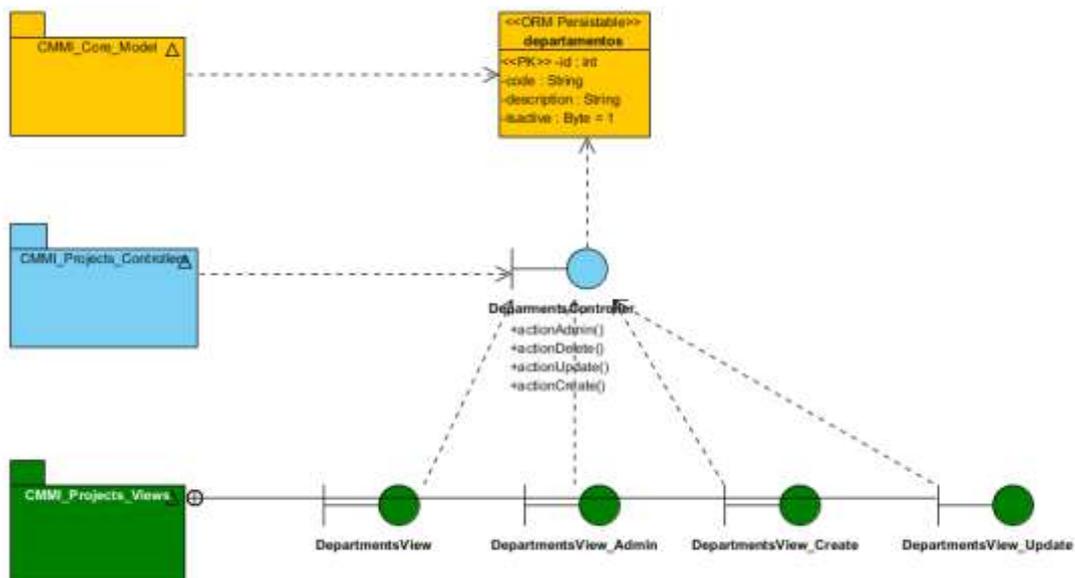


Ilustración 30 Detalle de la relación de componentes modelo, vista, controlador

Clases frontera y gestoras CMMI Proyectos, revisiones incidencias

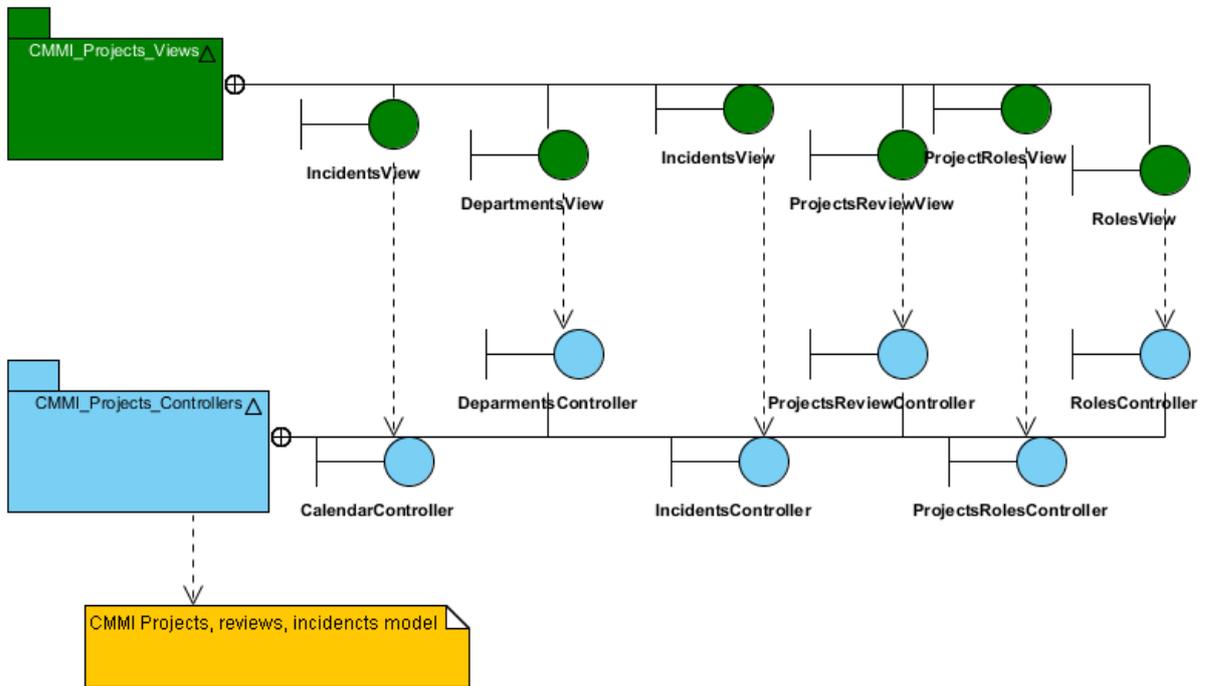


Ilustración 31 Clases frontera y gestoras CMMI Proyectos revisiones e incidencias

## Diseño de casos de uso : Diagramas de actividad de procesos

A continuación realizaremos el diseño de los casos de uso, aportando, en primer lugar unos diagramas de actividad para mostrar el flujo de un proceso entre los diferentes casos de uso.

De esta manera, estamos mostrando el negocio desde el punto de vista del usuario, en el que pueden intervenir diferentes casos de uso, y posteriormente ver cómo están diseñados los casos de uso más relevantes que intervienen en estos procesos.

Subsistema de conexión: Gestión de usuarios

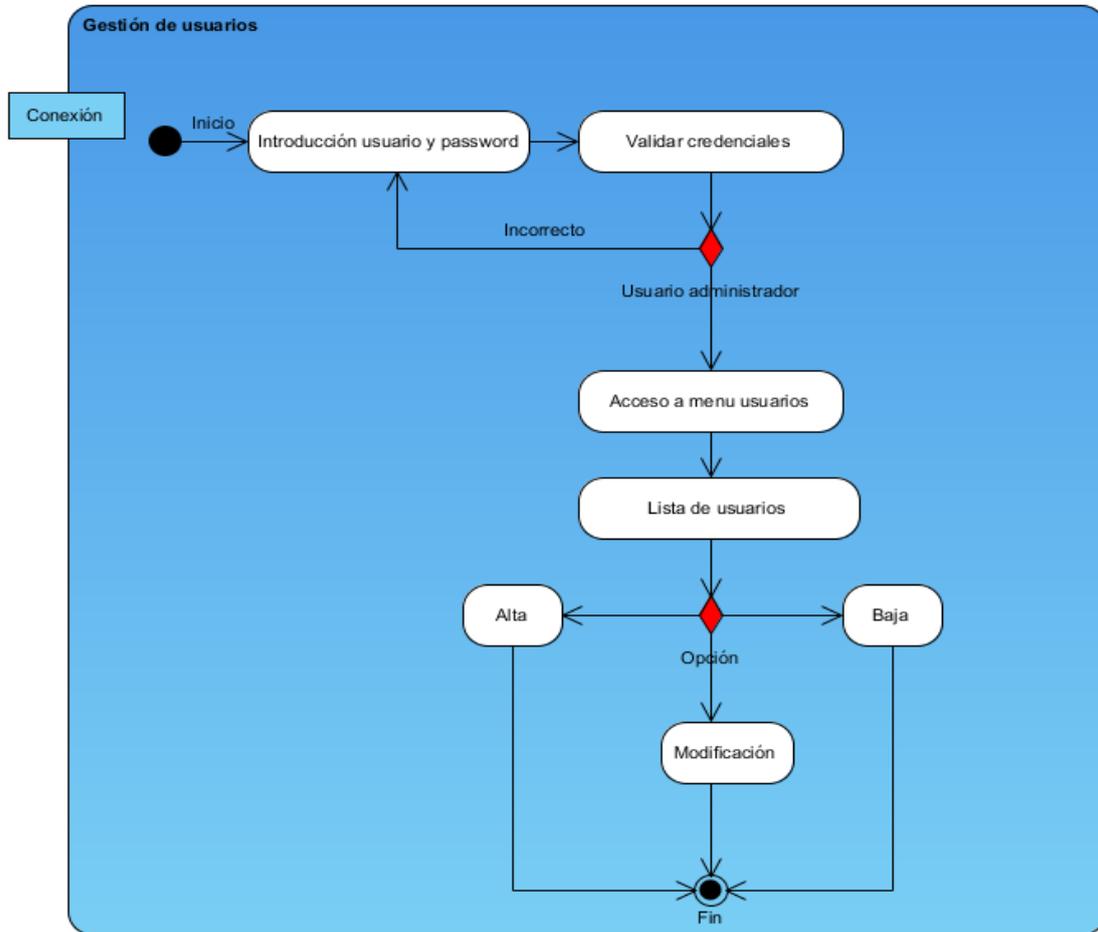


Ilustración 32 Diagrama de actividad para la gestión de usuarios

Acción del diagrama de actividad	Caso de uso relacionado
Inicio: Conexión	GU_CU_001: Conexión
Validar credenciales	GU_CU_002: Validar credenciales
Menú: Lista de usuarios	GU_CU_003: Lista de usuarios
Alta de usuario	GU_CU_004: Alta de usuario
Baja de usuario	GU_CU_005: Baja de usuario
Modificación	GU_CU_006: Modificación
Fin	

Subsistema de revisiones e incidencia: Proceso de revisión de una incidencia

A continuación veremos un diagrama de actividad para ver cómo intervienen los diferentes casos de uso en la revisión de una incidencia. Posteriormente, realizaremos un diagrama de secuencia de los más relevantes.

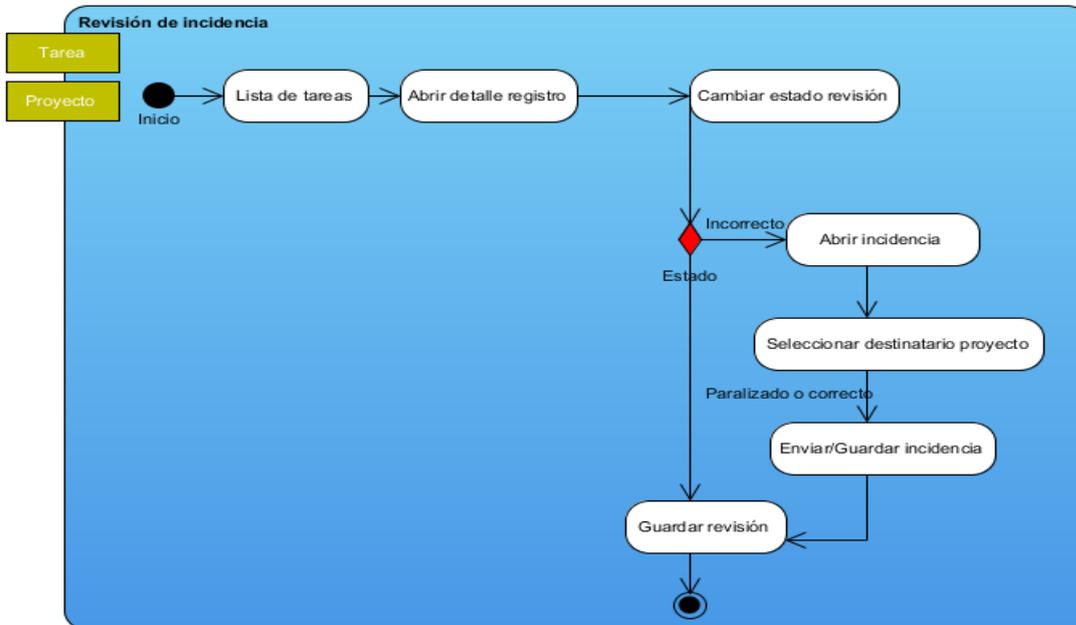


Ilustración 33 Diagrama de actividad para la revisión de una tarea

En el proceso de revisión, el usuario accede a una lista (previamente generada) en la que puede seleccionar un registro correspondiente a una tarea encomendada a un proyecto determinada. El objetivo es poder revisar dicha tarea y posteriormente decidir si ésta ya ha sido completada o bien tiene algún tipo de incidencia.

Como vemos, la relación de casos de uso que intervienen son:

Acción del diagrama de actividad	Caso de uso relacionado
Inicio	EV_CU_001: Matriz de revisiones
Abrir detalle del registro	EV_CU_006: Modificar revisión (corregir diagrama de caso de uso en AF)
Cambiar estado	EV_CU_003: Cambiar estado
Abrir incidencia/Acción correctiva	EV_CU_008: Añadir INC/AC
Seleccionar destinatario incidencia	EV_CU_008: Añadir INC/AC
Enviar guardar incidencia	EV_CU_008: Añadir INC/AC
Guardar revisión	EV_CU_006: Modificar revisión

Subsistema de revisiones e incidencia: Proceso de incorporación de una evidencia

A continuación veremos un diagrama de actividad para ver cómo intervienen los diferentes casos de uso en la revisión de una incidencia. Posteriormente, realizaremos un diagrama de secuencia de los más relevantes

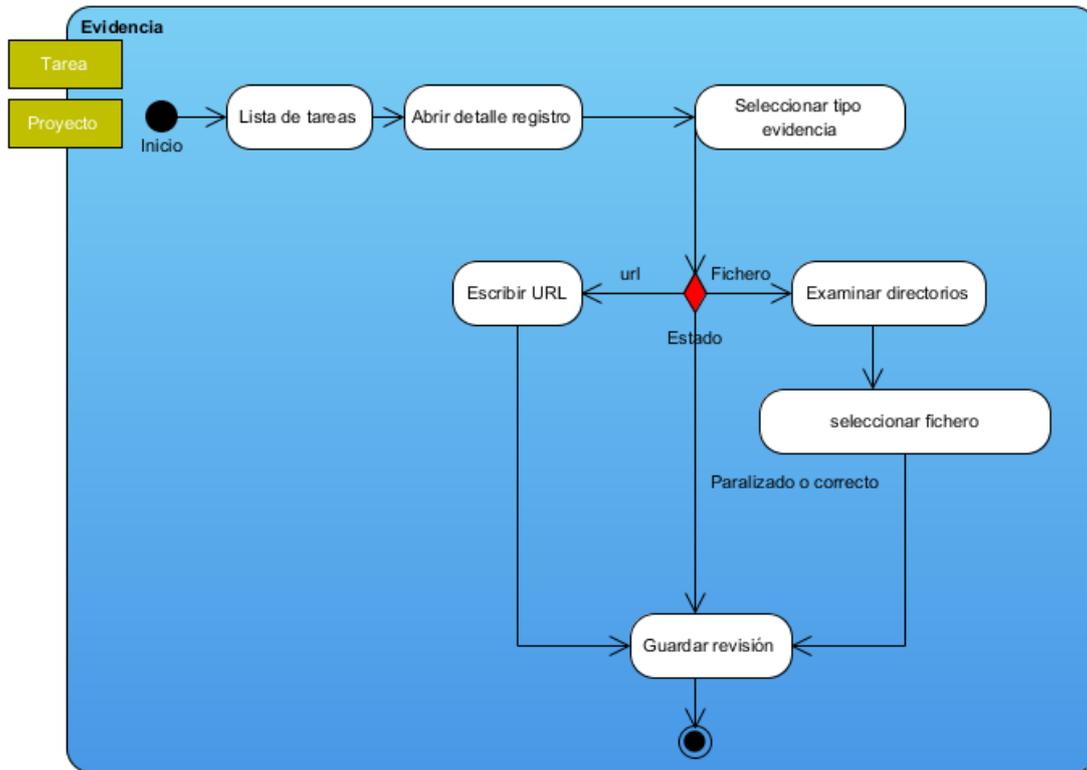


Ilustración 34 Diagrama de actividad para insertar evidencias

Acción del diagrama de actividad	Caso de uso relacionado
Inicio	EV_CU_001: Matriz de revisiones
Abrir detalle del registro	EV_CU_006: Modificar revisión (corregir diagrama de caso de uso en AF)
Seleccionar tipo evidencia	EV_CU_004: Añadir evidencia
Examinar directorios	EV_CU_004: Añadir evidencia
Seleccionar fichero	EV_CU_004: Añadir evidencia
Escribir URL	EV_CU_004: Añadir evidencia
Guardar revisión	EV_CU_006: Modificar revisión

## Diseño de casos de uso : Diagramas de secuencia

En los diagramas de secuencia, se modela la interacción y los mensajes que intercambian los diferentes objetos que intervienen en un proceso o caso de uso. Todo ello, mostrado en una secuencia temporal.

En CMMI-UP existen multitud de mantenimientos de tipo ABC (alta, baja, modificación) o CRUD (Create, Update, Delete). Para este tipo de mantenimientos, mostraremos un diagrama de secuencia genérico, que puede ser aplicado en todos ellos, dado que al estar trabajando con un framework, este tipo de implementaciones se realiza con un mismo procedimiento. En cambio, para aquellos procesos más representativos, mostraremos un diagrama de secuencia concreto.

Diagrama de secuencias para casos de uso tipo 'Listas'

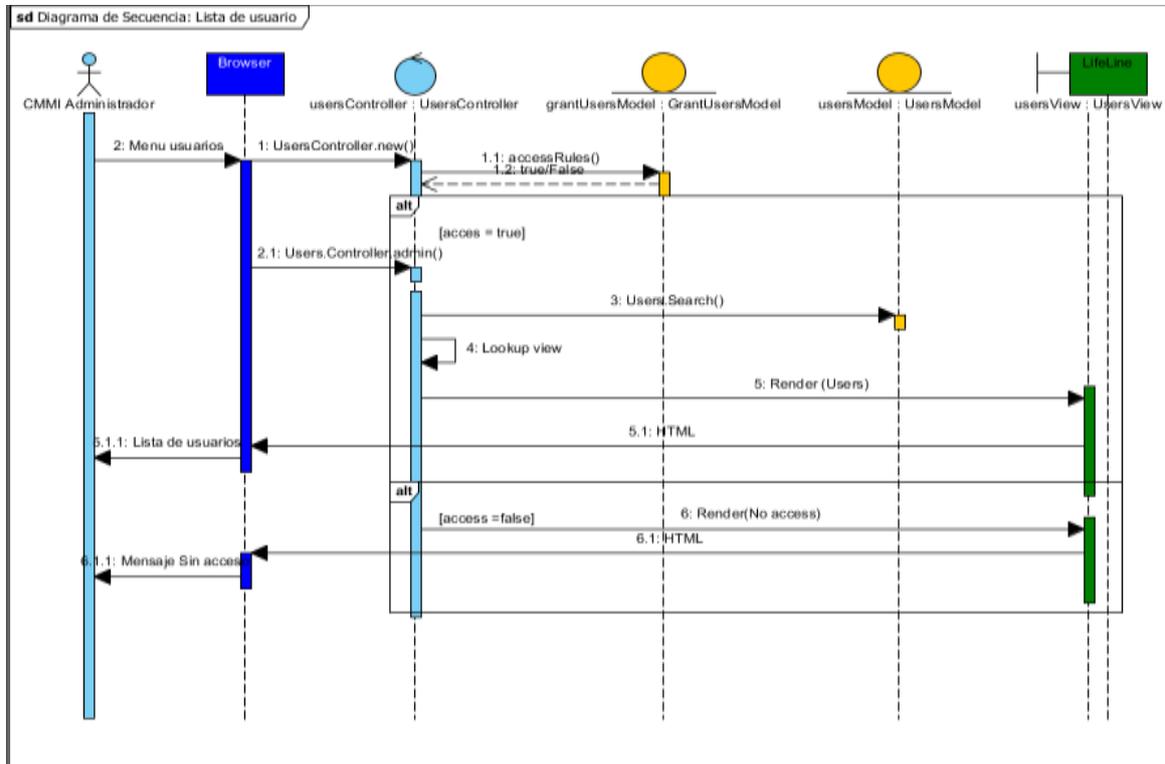


Ilustración 35 Diagrama de secuencia para listas

La secuencia habitual para mostrar la lista es la siguiente:

1. El usuario selecciona la opción correspondiente en el menú de usuarios de la aplicación
2. A través de un componente 'genérico' (omitido para resumir el diagrama) se determina cuál es el controlador asociado al evento disparado por el usuario.
3. A continuación se crea una nueva instancia del controlador.
4. El controlador recupera la lista de permisos para saber si el usuario puede acceder a la opción.
5. En caso afirmativo se accede al método de búsqueda correspondiente en modelo de la entidad (en este caso Users.Search() para recuperar los datos.
6. Al recuperar los datos, el controlador envía el objeto recuperado para renderizarlo a través del componente vista.
7. En caso de que el usuario no tenga permisos para acceder a la opción, se informa al usuario mediante un mensaje genérico

Diagrama de secuencias para casos de uso tipo 'Alta'

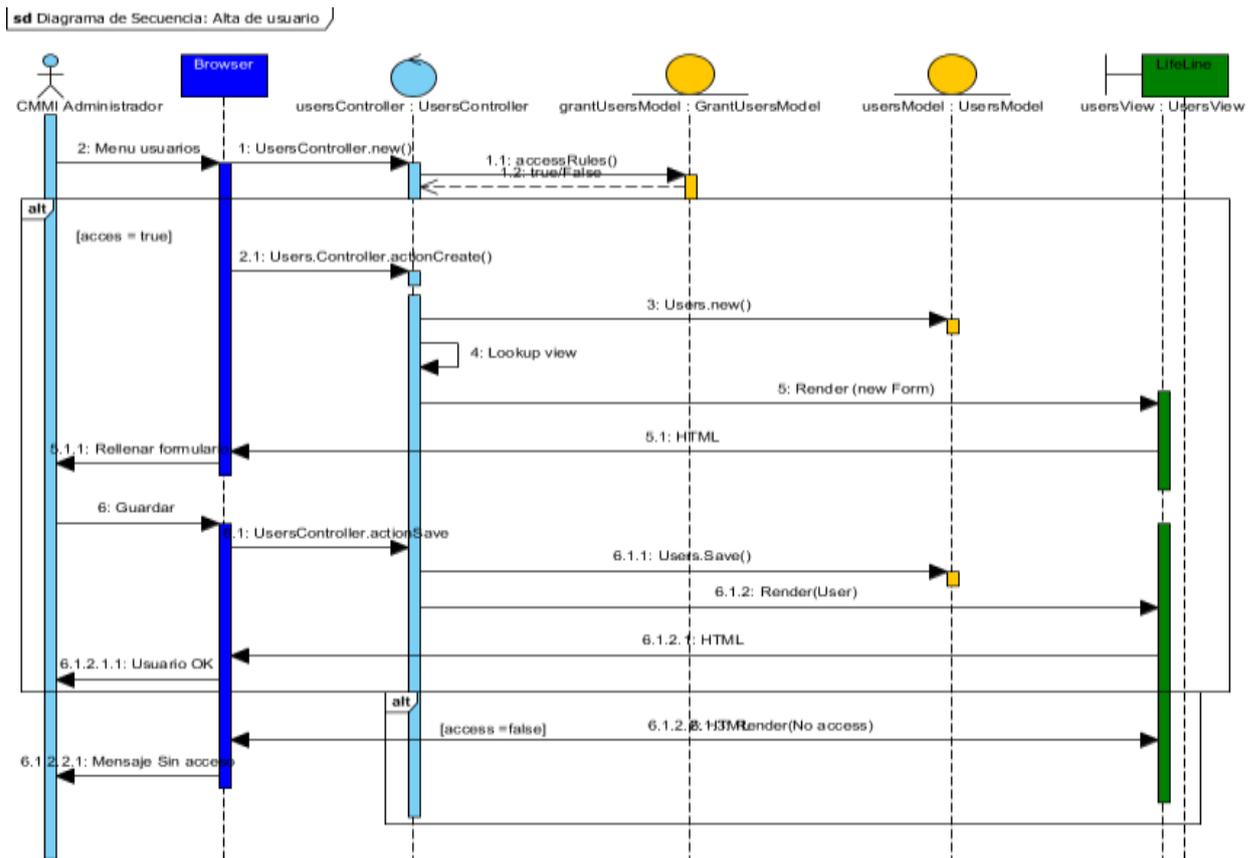


Ilustración 36 Diagrama de secuencia para Altas

La secuencia habitual es parecida a la anterior:

1. El usuario selecciona la opción correspondiente en el menú de usuarios de la aplicación
2. A través de un componente 'genérico' (omitido para resumir el diagrama) se determina cuál es el controlador asociado al evento disparado por el usuario.
3. A continuación se crea una nueva instancia del controlador.
4. El controlador recupera la lista de permisos para saber si el usuario puede acceder a la opción.
5. En caso afirmativo se crea una nueva instancia de la entidad y se renderiza el formulario para inicializar los campos a rellenar por parte de el usuario.
6. El usuario rellena los datos, se validan los campos y sus atributos y pulsa el botón 'Guardar'.
7. El controlador interpreta el evento y lanza un nuevo método en el modelo para almacenar el objeto usuario en la base de datos.

## Diagrama de secuencias para casos de uso tipo 'Baja'

sd Diagrama de Secuencia: Baja de usuario

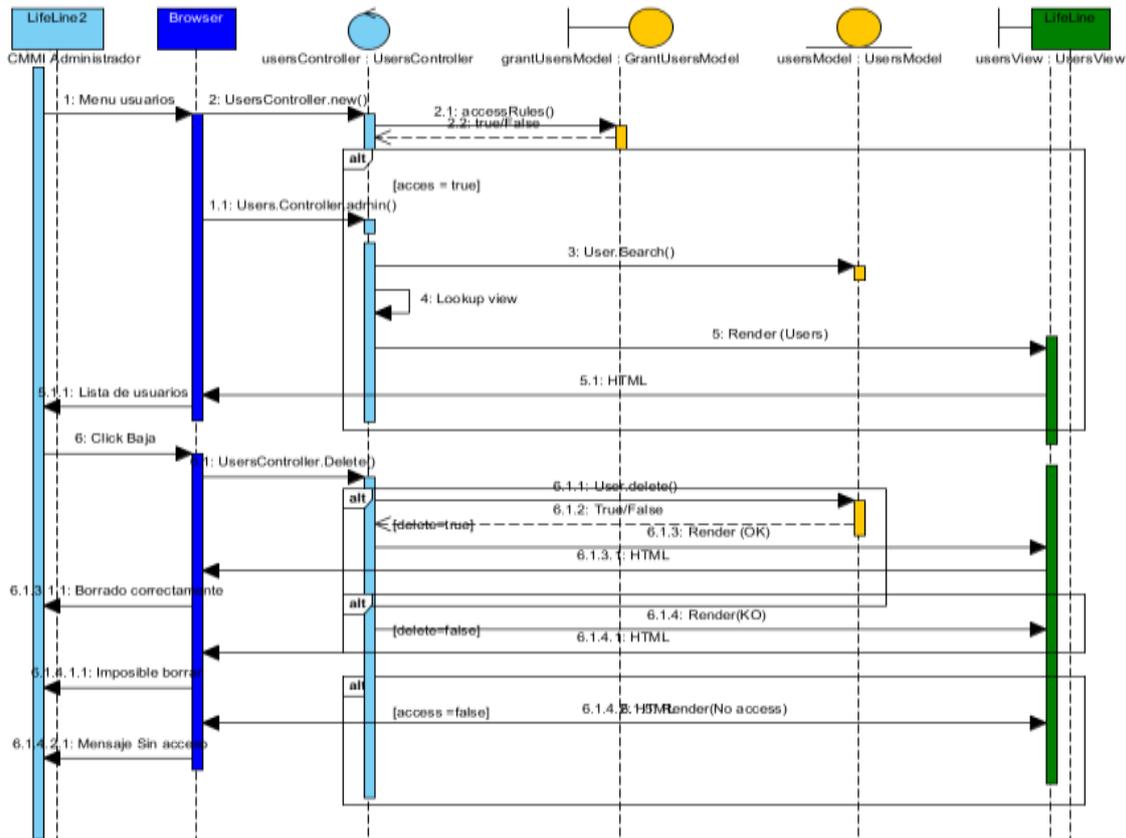


Ilustración 37 Diagrama de secuencias para 'bajas'

La secuencia es la siguiente:

1. El usuario selecciona la opción correspondiente en el menú de usuarios de la aplicación
2. A través de un componente 'genérico' (omitido para resumir el diagrama) se determina cuál es el controlador asociado al evento disparado por el usuario.
3. A continuación se crea una nueva instancia del controlador.
4. El controlador recupera la lista de permisos para saber si el usuario puede acceder a la opción.
5. En caso afirmativo se crea una nueva instancia de la entidad y se renderiza la lista de usuarios con los datos proporcionados por el modelo.
6. El usuario edita el registro y pulsa el botón 'Baja'.
7. El controlador interpreta el evento y lanza la instrucción correspondiente al modelo, el cual, determinará previamente si no afecta a la integridad referencial de los datos. En

caso de que no afecte, borrará el registro. En caso contrario, mostrará un mensaje al usuario.

### Diagrama de secuencias para casos de uso tipo 'Dashboard'

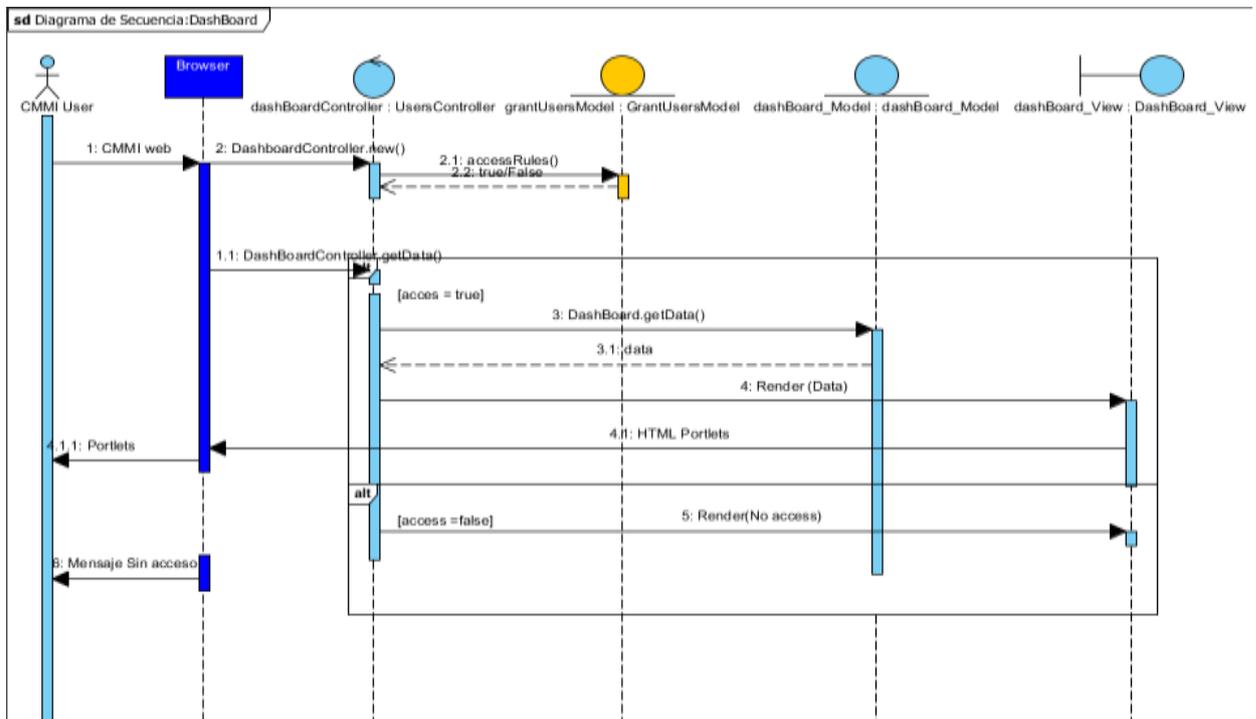


Ilustración 38 Diagrama de secuencia para Dashboard

La secuencia es la siguiente:

1. El usuario accede a la aplicación a través del login.
2. A través de un componente 'genérico' (omitido para resumir el diagrama) se determina cuál es el controlador asociado al evento disparado por el usuario.
3. A continuación se crea una nueva instancia del controlador.
4. El controlador recupera la lista de permisos para saber si el usuario puede visualizar el dashboard.
5. En caso afirmativo se crea una nueva instancia de la entidad (la entidad Dashboard no es persistente) y se llama al método correspondiente para que realiza las diferentes consultas en base de datos.
6. Devuelve los diferentes dato para que sean renderizados por el componente vista.

Diseño E/R CMMI Core

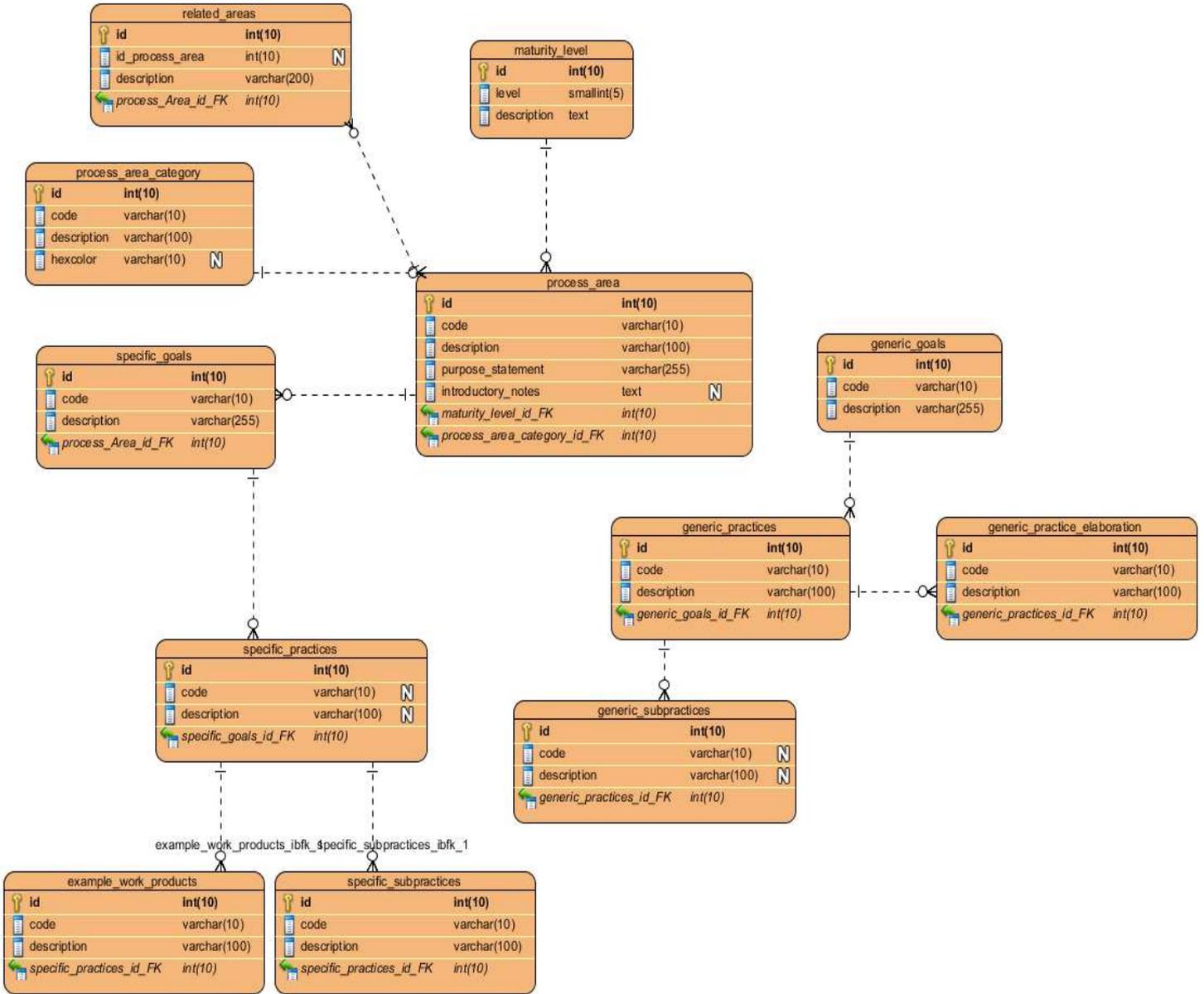


Ilustración 39 E-R CMMI Core

## Diseño de la interfaz gráfica

### Pantalla principal y menú

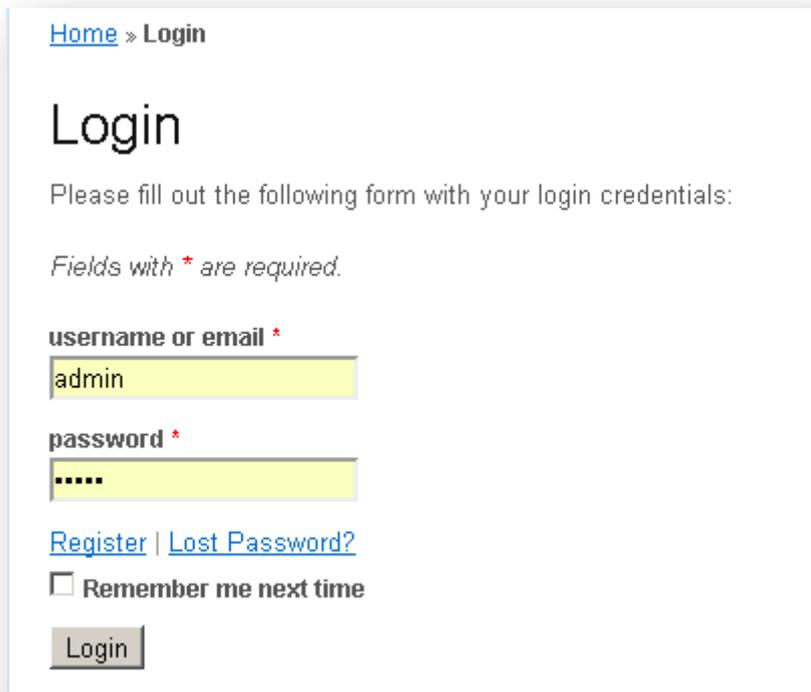


*Ilustración 40 Prototipo pantalla Menú*

La pantalla principal presenta un aspecto típico con un menú de opciones horizontal clásico.

Cabe destacar la posibilidad que nos brinda el framework de desarrollo en adaptar diferentes temáticas (themes) que nos permiten adecuar el look & feel según requerimientos de cliente de una manera fácil.

## Login

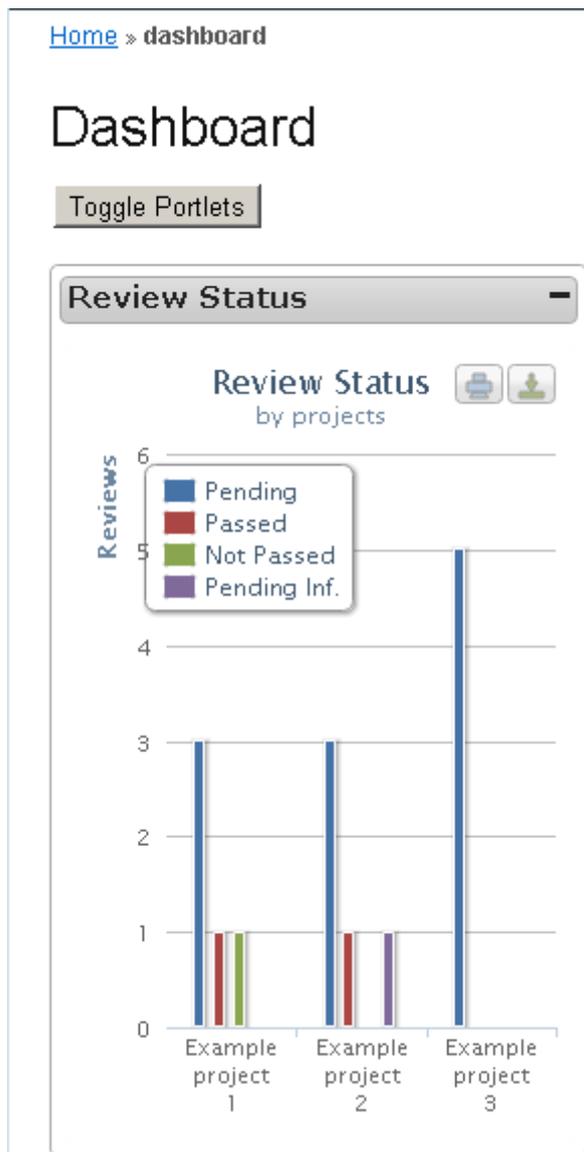


The screenshot shows a web browser window with a login page. At the top left, there is a breadcrumb trail: [Home](#) » [Login](#). The main heading is "Login". Below it, a message says "Please fill out the following form with your login credentials:". A note below that reads "Fields with \* are required.". There are two input fields: the first is labeled "username or email \*" and contains the text "admin"; the second is labeled "password \*" and contains five asterisks. Below the password field are two links: "Register" and "Lost Password?". There is a checkbox labeled "Remember me next time" which is currently unchecked. At the bottom of the form is a "Login" button.

*Ilustración 41 Prototipo Pantalla Login*

La pantalla de acceso a la aplicación permite la entrada y la validación de la contraseña así como las opciones clásicas de recuperación de contraseña o la de habilitar el recordatorio de usuario.

## Dashboard



*Ilustración 42 Prototipo Dashboard*

Una vez el usuario se ha validado, la pantalla principal se sitúa en el caso de uso 'Dashboard', que permitirá tener una visión global de los diferentes gráficos y que, de una manera visual, el usuario puede conocer el estado de la implantación CMMI.

## Pantallas CMMI: Core

[Home](#) » [Process Area Categories](#) » **Manage**

### Manage Process Area Categories

You may optionally enter a comparison operator (<, <=, >, >=, <> or =) at the beginning of each of your search values to specify how the comparison should be done.

[Advanced Search](#)

Displaying 1-5 of 5 result(s).

Code	Category	
PRM	Process Management	
PJM	Project Management	
ENG	Engineering	
SUP	Support	
99	Not defined	

*Ilustración 43 Prototipo lista de Categorías*

Las pantallas de tipo 'listas', en los mantenimientos, siempre presentan un aspecto homogéneo. En este caso, los colores son configurables y permiten diferenciar las áreas de proceso de CMMI.

[Home](#) » [Process Area Categories](#) » Create

## Create ProcessAreaCategory

*Fields with \* are required.*

**Code \***

**Category \***

red ▾

*Ilustración 44 Prototipo Alta de Categoría*

Las pantallas de tipo 'alta' presentan un aspecto similar. Cada uno de los controles será implementado según la información que contendrá (listas desplegables, selectores de fecha, cajas de texto, etc.)

Como podemos observar, también se indicará con un asterisco los campos obligatorios.

[Home](#) » [Maturity Levels](#) » **Manage**

## Manage Maturity Levels

You may optionally enter a comparison operator (<, <=, >, >=, <> or =) at the beginning of each of your search values to specify how the comparison should be done.

[Advanced Search](#)

Displaying 1-5 of 5 result(s).

Level	Maturity Level	
<input type="text"/>	<input type="text"/>	
1	Initial	
2	Managed	
3	Defined	
4	Quantitative Managed	
5	Optimizing	

*Ilustración 45 Prototipo lista niveles de Madurez*

Las listas también presentan la opción de filtro en la primera línea y la capacidad de acceder a los diferentes casos de uso (eliminar, modificar, visualizar) desde la propia línea.

[Home](#) » [Process Areas](#) » [Manage](#)

## Manage Process Areas

You may optionally enter a comparison operator (<, <=, >, >=, <> or =) at the beginning of each of your search values to specify how the comparison should be done.

[Advanced Search](#)

Displaying 1-20 of 23 result(s).

Code	Code	Description	Maturity Level	
PRM	CAR	Causal Analysis and Resolution	Optimizing	
	CM	Configuration Management	Managed	
	DAR	Decision Analysis and Resolution	Defined	
	OPD	Organizational Process Definition	Defined	
	OPM	Organizational Performance Management	Optimizing	
	OPP	Organizational Process Performance	Quantitative Managed	
	OT	Organizational Training	Defined	
	PMC	Project Monitoring and Control	Managed	
	PP	Project Planning	Managed	
	PPQA	Process and Product Quality Assurance	Managed	
	QPM	Quantitative Project Management	Quantitative Managed	
	REQM	Requirements Management	Managed	
	RSKM	Risk Management	Defined	
	SAM	Supplier Agreement Management	Initial	
	TS	Technical Solution	Initial	
	VAL	Validation	Initial	
	VER	Verification	Initial	
	PI	Product Integration	Initial	
	RD	Requeriments development	Initial	
ENG	MA	Measurement and Analysis	Managed	

Go to page: [< Previous](#) [1](#) [2](#) [Next >](#)

Il·lustració 46 Prototipo lista áreas de proceso

En el mantenimiento de áreas de proceso, como podemos observar, se realiza una pantalla de tipo lista de dos niveles, permitiendo agrupar de este modo por categoría.

Obsérvese también que los colores nos indican la categoría a la que pertenece cada área de proceso.

## Pantallas CMMI: Proyectos, Roles

[Home](#) » [Roles](#) » [Manage](#)

### Manage Roles

You may optionally enter a comparison operator (<, <=, >, >=, <> or =) at the beginning of each of your search values to specify how the comparison should be done.

[Advanced Search](#)

Displaying 1-5 of 5 result(s).

Code	Description	Active	
<input type="text"/>	<input type="text"/>	<input type="text"/>	
ANA	Analyst	1	
PRG	Programmer	1	
TD	Technical designer	1	
PM	Project Manager	1	
RWW	Reviewer	1	

Ilustración 47 Prototipo lista Roles

En esta pantalla podemos gestionar los diferentes roles del equipo de proyectos.

[Home](#) » [Projects](#) » [Manage](#)

### Manage Projects

You may optionally enter a comparison operator (<, <=, >, >=, <> or =) at the beginning of each of your search values to specify how the comparison should be done.

[Advanced Search](#)

Displaying 1-3 of 3 result(s).

Code	Description	Description	Candidate	Start Date	End Date	Active	Options
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Example project 1	Laboratory Information Systems	Public & Health	Candidate	2012-07-01	2012-12-31	Active	
Example project 2	Public Health Information Systems	Public & Health	Not Candidate	2012-07-01	2012-07-31	Active	
Example project 3	ESPRICSSHA	Public & Health	Not Candidate	2012-07-01	2012-07-04	Active	

Ilustración 48 Prototipo lista de proyectos

El mantenimiento de proyectos contiene toda la información relevante del proyecto.

[Home](#) » [Projects](#) » [Create](#)

## Create Projects

*Fields with \* are required.*

**Code \***

**Description \***

**Departments Id Fk \***

**Iscandidate \***

**Start Date**

**End Date**

**Isactive \***

Ilustración 49 Prototipo alta de proyectos

## Pantallas CMMI: Revisiones

[Home](#) » [Review Types](#) » [Manage](#)

### Manage Review Types

You may optionally enter a comparison operator (<, <=, >, >=, <> or =) at the beginning of each of your search values to specify how the comparison should be done.

[Advanced Search](#)

Displaying 1-3 of 3 result(s).

Code	Description	Every n Days	Unique	
1	Unique review	0	just one time	 
2	Every 15 days	15	every n days	 
3	Monthly	30	every n days	 

Ilustración 50 Prototipo lista tipos de revisiones

Las revisiones permiten definir el intervalo de días que debe pasar entre una revisión y otra. Posteriormente asociaremos este tipo de revisión a una tarea determinada.

[Home](#) » [Review Tasks](#) » [Manage](#)

### Manage Review Tasks

You may optionally enter a comparison operator (<, <=, >, >=, <> or =) at the beginning of each of your search values to specify how the comparison should be done.

[Advanced Search](#)

Displaying 1-10 of 34 result(s).

Code	Description	Active	Description	Long Description	
PM1	Update Local PMWorkbook	Active	Unique review	Update Local PMWorkbook	 
PM2	Use EET (Planning & Changes)	Active	Every 15 days	Use EET (Planning & Changes)	 
PMC1	Consistent data in EPM/SIH/JIRA	Active	Monthly	Consistent data in EPM/SIH/JIRA	 
PMC2	SIH. Actual work registered & approved weekly.	Active	Unique review	SIH. Actual work registered & approved weekly.	 
PMC3	Jira Project. Actual work & ETC registered (all team) daily. Task status updated online	Active	Unique review	Jira Project. Actual work & ETC registered (all team) daily. Task status updated online	 
PMC4	Check fulfillment Problem Report (include open issues; claims; problems?)	Active	Unique review	Check fulfillment Problem Report (include open issues; claims; problems?)	 
PMC5	Project Status reviewed by xDU Manager	Active	Unique review	Project Status reviewed by xDU Manager	 
PMC6	Show commitment of project team	Active	Unique review	Show commitment of project team	 
PMC7	Show commitment of Customer	Active	Unique review	Show commitment of Customer	 
REQM1	Terms / criteria of project acceptanc	Active	Unique review	Terms / criteria of project acceptanc	 

Go to page: [< Previous](#) [1](#) [2](#) [3](#) [4](#) [Next >](#)

Ilustración 51 Prototipo lista tareas de revisión

Como vemos, aquí ya están asociados los tipos de revisiones a las diferentes tareas.

[Home](#) » [Review Tasks](#) » Create

## Create ReviewTasks

*Fields with \* are required.*

**Code \***

**Description \***

**Active?**

**Review Types \***

**Long Description**

*Ilustración 52 Prototipo Alta de tareas de revisión*

[Home](#) » [Projects Reviews](#) » [Manage](#)

## Manage Projects Reviews

You may optionally enter a comparison operator (<, <=, >, >=, <> or =) at the beginning of each of your search values to specify how the comparison should be done.

[Advanced Search](#)

Displaying 1-10 of 15 result(s).

ID	Review Date	Status	Real Review Date	Evidence Url	Notes	Description	Description	
195	2012-08-04	Passed	2012-08-04	http://www.google.es/		Laboratory Information Systems	Update a Local PM Workbook	
196	2012-08-04	Pending information	2012-08-04			Public Health Information Systems	Update a Local PM Workbook	
197	2012-08-04	Pending	2012-08-04			ESPRICSSHA	Update a Local PM Workbook	
199	2012-08-19	Not Passed	2012-08-19			Laboratory Information Systems	Use EET (Planning & Changes)	
199	2012-08-19	Pending	2012-08-19			Public Health Information Systems	Use EET (Planning & Changes)	
200	2012-08-19	Pending	2012-08-19			ESPRICSSHA	Use EET (Planning & Changes)	
201	2012-09-03	Pending	2012-09-03			Laboratory Information Systems	Consistent data in EPM/SH/JIRA	
202	2012-09-03	Pending	2012-09-03			Public Health Information Systems	Consistent data in EPM/SH/JIRA	

Ilustración 53 Prototipo matriz de revisiones

Esta es la pantalla principal que usará el revisor CMMI para ir revisando las tareas encomendadas a los diferentes proyectos. En él puede observarse la evidencia, el estado de la revisión (con diferentes colores), el proyecto y la tarea que debería haber realizado.

[Home](#) » [Review Task Coverages](#) » [Manage](#)

## Manage Review Task Coverages

You may optionally enter a comparison operator (<, <=, >, >=, <> or =) at the beginning of each of your search values to specify how the comparison should be done.

[Advanced Search](#)

Displaying 1-14 of 14 result(s).

Code	Code	Description	Description	Description	
PRM	REQM	Terms / criteria of project acceptanc	Terms / criteria of project acceptance	Not defined	
PJM	PP	Update Local PMWorkbook	Update a Local PM Workbook	Not defined	
	PMC	SIH. Actual work registered & approved weekly.	SIH. Actual work registered & approved weekly.	Not defined	
	PP	Consistent data in EPM/SIH/JIRA	Consistent data in EPM/SIH/JIRA	Not defined	
	PP	Use EET (Planning & Changes)	Use EET (Planning & Changes)	Not defined	
ENG	VAL	Review performance	Review performance	Not defined	
	VER	Review performance	Review performance	Not defined	
	VAL	Review plan	Review plan	Not defined	
	VER	Review plan	Review plan	Not defined	
	RD	Setup project & use Enterprise Architect	Setup project & use Enterprise Architect	Not defined	
	RD	Setup project & use HPQC	Setup project & use HPQC	Not defined	
SUP	CM	Register PMBook Baseline	Register PMBook Baseline	Not defined	
	CM	Create review and execute CM Plan	Create, review and execute CM Plan	Not defined	
	CM	Move folder structure to EPM Environment and Subversion tool	Move folder structure to EPM Environment and Subversion tool	Not defined	

Ilustración 54 Prototipo lista de cobertura de tareas

En esta pantalla se permite definir la cobertura de cada tarea. De esta manera, a medida que se van realizando las tareas se puede obtener una visión del alcance obtenido en la implantación de una manera real.



Ilustración 55 Prototipo calendario de revisiones

El calendario permite tener una visión general de las tareas que debe revisarse en un intervalo de tiempo. Desde el propio calendario también puede acceder a cambiar el estado de una revisión o asignarle evidencias a una tarea.

## Valoración económica

En este apartado, hemos considerado apropiado realizar una valoración económica del trabajo realizado y una estimación del trabajo pendiente de realizar, como puede ser la implementación del proyecto de software, las pruebas pertinentes, la implantación o la formación.

Existen diversos estudios y modelos entorno a la valoración de la construcción del software. No obstante, varios autores han analizado una aproximación del coste global de desarrollo de un proyecto informático. Estas aproximaciones datan de los años setenta y comienzos de los ochenta por autores como Boehm i Brooks y son aceptadas en la actualidad como referencia.

- Un 40% del coste de desarrollar una aplicación se emplea las etapas de análisis y diseño.
- Un 20% en la etapa de implementación
- El resto, entorno al 40% se emplea en las pruebas.

El costo de mantenimiento que puede representar el doble del coste de implementación, y el del hardware, en este caso no se tiene en cuenta.

Una vez recopilado todos los datos necesarios, procederemos a valorar el coste del proyecto. Para ello nos basaremos en el coste de cada recurso que debiera participar en el proyecto y tomando como referencia los precios estándar del sector y asumiendo una persona por rol y tarea, dado que el tiempo de entrega no es un requisito importante en este caso.

Recurso	Coste/hora	Coste/Jornada
Jefe de proyectos	48€	384€
Analista	36€	288€
Analista programador	24€	192€
Arquitecto	35€	280€

Recordemos la tabla de actividades desglosadas en el plan de proyecto:

- **Nota:** Pese a que la planificación temporal se basa en estimaciones de jornadas de 4 horas, en la estimación económica hemos tenido en cuenta jornadas de 8 horas para adecuarlos a los estándares profesionales. Por lo tanto, hemos considerado adecuado dividir por dos las jornadas detalladas en el planning.

Estimación por actividades					
Actividad	Nombre de actividad: Nivel I	Jornadas de 8 horas	Recurso	Precio Jornada	Total
01	PAC1- Plan de proyecto	4	Jefe de proyecto	384	1536
02	PAC2- Especificación y análisis	13	Analista	288	3744
03	PAC3- Diseño del sistema	12,5	Analista- Programador	192	2400
04	Memoria y presentación	7,5	Jefe de proyecto	384	2880
05	Programación y pruebas unitarias	40	Analista programador	192	7680
06	Pruebas integradas	15,5	Analista	288	4464
07	<sup>1</sup> Formación a usuarios	1	Analista	288	288
08	<sup>2</sup> Adecuación de datos tareas vs cobertura CMMI	1,5	Analista programador	192	288
09	Puesta en producción	1,5	Arquitecto	280	420
10	<sup>3</sup> Final del proyecto	0,5	Jefe de proyecto	384	192
<b>TOTAL</b>		<b>97</b>			<b>23892</b>

<sup>1</sup> Aunque no existe formación explícita, consideramos 1 jornada mínimo para explicar el funcionamiento del sistema a las personas que comiencen a probar

<sup>2</sup> Aunque no existe migración de datos, estimamos 1 jornada para la revisión de los datos de partida (tareas y cobertura de tareas).

<sup>3</sup> El cierre del proyecto requiere recopilar la documentación y formalizar el cierre con el cliente

## Conclusiones

---

El proyecto CMMI UP surge de una experiencia real de una implantación CMMI en una organización, cuya parte de su dedicación recae en el desarrollo de software. En ocasiones, las directrices que marca un modelo se siguen de manera extrema, causando una gran problemática de alineamiento con los procesos ya implantados y que es posible que ya funcionen.

La herramienta que proponemos, no garantiza que una implantación de tal calibre pueda llegar a buen puerto, sino que pretende evitar los costes indirectos que supone la gestión de la información, las revisiones que acarrean y la obtención de una visión clara del estado de la implantación.

No sería descabellado pensar que este tipo de solución que hemos abordado, diera un paso adelante y, abordando el problema desde una perspectiva más abstracta, podamos reutilizarlo para implantaciones de tipo ISO, ITIL, etc. ya que posiblemente lo que cambia es la información, no el modelo.

En aspectos económicos, la solución presenta un coste relativamente bajo, debido a que no posee complejos flujos de trabajo ni complejos algoritmos ni pantallas muy sofisticadas. La simplicidad debe ser un factor clave dado y debe alinearse con las tareas de gestión que habitualmente llevan a cabo las organizaciones y que generalmente realizan combinando diferentes herramientas ofimáticas. El coste de esta solución no debe presentar mayores problemas en incorporarlo en presupuestos globales de implantación.

En definitiva, el proyecto CMMI UP se presenta como una solución que mejorará la productividad de las tareas que conlleva una implantación de mejores prácticas como es la del modelo de madurez CMMI y desarrollado bajo la experiencia de una situación real en la que pueden verse reflejadas multitud de empresas del sector.

## Apéndice A: Glosario

---

**Acción correctiva** (corrective action) Acciones o actos usados para remediar una situación, eliminar un error o ajustar una condición.

**Adaptación** (tailoring) La adaptación de un proceso hace, modifica o adapta la descripción de proceso para un fin particular. Por ejemplo, un proyecto establece su proceso definido adaptándolo a partir del conjunto de procesos estándar de la organización para cumplir los objetivos, las limitaciones y el entorno del proyecto.

**Adecuado** (adequate) Esta palabra se usa para que se puedan interpretar las metas y las prácticas a la luz de los objetivos de negocio de su organización. Cuando se usa cualquier modelo CMMI, se deben interpretar las prácticas de forma que funcionen para su organización. El término se usa en las metas y las prácticas donde ciertas actividades pueden no realizarse siempre (véase también “apropiado” y “según sea necesario”).

**Adquisición** (acquisition) El proceso consistente en obtener productos (bienes y servicios) a través de contrato.

**Alcance de la evaluación** (appraisal scope) La definición de los límites de la evaluación que engloban los límites de la organización y los límites del modelo CMMI, dentro de los cuales operan los procesos que van a ser investigados.

**Análisis de requerimientos** (requirement analysis) La determinación del rendimiento y de las características funcionales específicas del producto, basándose en el análisis de las necesidades, expectativas y restricciones del cliente; en el concepto operativo; en los entornos de uso proyectados para las personas, los productos y los procesos; y en las medidas de eficacia.

**Análisis de riesgos** (risk analysis) La evaluación, clasificación y priorización de los riesgos.

**Análisis funcional** (functional analysis) Examen de una función definida para identificar todas las subfunciones necesarias para realizarla; identificación de las relaciones funcionales e interfaces (internas y externas) y captura de éstas en una arquitectura funcional; y transferir los requerimientos de rendimiento de mayor nivel y asignación de estos requerimientos a subfunciones de menor nivel. (Véase también “arquitectura funcional”).

**Área de proceso** (process area) Un grupo de prácticas relacionadas en un área que, cuando se implementan colectivamente, satisfacen un conjunto de metas consideradas importantes para hacer mejoras en ese área. Todas las áreas de proceso CMMI son comunes tanto a la representación continua como a la representación por etapas.

**Arquitectura del proceso** (process architecture) Las relaciones de orden, las interfaces, las interdependencias y otras relaciones entre los elementos de proceso de un proceso estándar. La arquitectura de proceso también describe las interfaces, las interdependencias y otras relaciones entre los elementos de proceso y los procesos externos (p. ej., la gestión del contrato).

**Arquitectura funcional** (functional architecture) La organización jerárquica de las funciones, de sus interfaces funcionales internos y externos (externos a la propia agregación) e interfaces físicos externos, de sus respectivos requerimientos funcionales y de rendimiento, y de sus restricciones de diseño.

**Arranque del proyecto** (project startup) Cuando un conjunto de recursos interrelacionados se dirigen a desarrollar o entregar uno o más productos destinados a un cliente o usuario final. (Véase también “proyecto”).

**Aseguramiento de la calidad** (quality assurance) Un modo planificado y sistemático de asegurar a la gerencia que se aplican los estándares, prácticas, procedimientos y métodos definidos del proceso.

**Atributos de producto de trabajo y de tarea** (work product and task attributes) Características de los productos, servicios y tareas del proyecto usadas para ayudar en la estimación del trabajo del proyecto. Estas características incluyen elementos tales como el tamaño, la complejidad, el peso, la forma, el ajuste y la función. Se usan normalmente como una entrada para derivar otras estimaciones del proyecto o de recursos (p. ej., esfuerzo, coste y calendario).

**Auditoría** (audit) En las actividades de mejora de procesos de CMMI, un examen objetivo de un producto de trabajo o de un conjunto de productos de trabajo frente a criterios específicos (p. ej., requerimientos).

**Calidad** (quality) La capacidad de un conjunto de características inherentes de un producto, componente de producto o proceso para satisfacer los requerimientos de los clientes.

**Calificación** (rating) (Véase “calificación de la evaluación”).

**Calificación de la evaluación** (appraisal rating) Tal y como se usa en los materiales de evaluación CMMI, el valor asignado por un equipo de evaluación a (a) una meta o área de proceso CMMI (b) el nivel de capacidad de un área de proceso o (c) el nivel de madurez de una unidad de la organización. La calificación se determina aplicando el proceso de calificación definido por el método de evaluación que se está empleando.

**Capacidad de proceso** (process capability) El rango de resultados esperados que pueden lograrse siguiendo un proceso.

**Ciclo de vida del producto** (product lifecycle) El periodo de tiempo, consistente en fases, que empieza cuando se concibe un producto y termina cuando el producto ya no está disponible para su uso. Dado que una organización puede estar produciendo múltiples productos para múltiples clientes, una descripción de un ciclo de vida del producto puede no ser adecuada. Por tanto, la organización puede definir un conjunto de modelos aprobados de ciclo de vida del producto. Estos modelos se encuentran normalmente en la literatura publicada y es probable que se adapten para uso en una organización. Un ciclo de vida del producto podría constar de las siguientes fases: (1) concepto/visión (2) viabilidad (3) diseño/desarrollo (4) producción y (5) retirada.

**Ciente** (customer) La parte (individuo, proyecto u organización) responsable de aceptar el producto o de autorizar el pago. El cliente es externo al proyecto (excepto quizás cuando se usan equipos integrados, como en IPPD), pero no necesariamente externo a la organización.

El cliente puede ser un proyecto de mayor nivel. Los clientes son un subconjunto de las partes interesadas. (Véase también “partes interesadas”). En la mayoría de los casos en los que se usa este término, la definición precedente es la que prevalece. Sin embargo, en algunos contextos, el término “cliente” pretende incluir a otras partes interesadas relevantes (Véase también “Requerimientos de cliente”).

**CMMI UP** Nombre dado al proyecto y al software (producto) en el marco de este proyecto.

**Componente de producto** (product component) En el Conjunto de productos CMMI, un producto de trabajo que es un componente de bajo nivel del producto. Los componentes de producto se integran para producir el producto. Pueden existir varios niveles de componentes de producto. (Véase también “producto” y “producto de trabajo”).

**Componente del modelo CMMI** (CMMI model component) Cualquiera de los principales elementos arquitectónicos que componen el modelo CMMI. Algunos de los principales elementos de un modelo CMMI incluyen prácticas específicas, prácticas genéricas, metas específicas, metas genéricas, áreas de proceso, niveles de capacidad y niveles de madurez.

**Componentes CMMI informativos** (informative CMMI components) Componentes CMMI que ayudan a los usuarios del modelo a comprender los componentes requeridos y esperados de un modelo. Estos componentes pueden contener ejemplos, explicaciones detalladas u otra información útil. Las subprácticas, notas, referencias, títulos de metas, títulos de prácticas, fuentes, productos de trabajo típicos, ampliaciones y elaboraciones de prácticas genéricas son componentes informativos del modelo.

**Componentes CMMI requeridos** (required CMMI components) Componentes CMMI que son esenciales para alcanzar la mejora de procesos en un área de proceso determinada. Estos componentes se usan en las evaluaciones para determinar la capacidad de proceso. Las metas específicas y las metas genéricas son componentes del modelo requeridos.

**Conjunto de procesos estándar de la organización** (organization’s set of standard processes) Una colección de definiciones de los procesos que guían las actividades en una organización. Estas descripciones de procesos cubren los elementos fundamentales de proceso (y las relaciones entre ellos, tales como secuencia e interfaces) que deben incorporarse en los procesos definidos que se implementan en los proyectos de la organización. Un proceso estándar asegura la coherencia de las actividades de desarrollo y de mantenimiento en toda la organización y es esencial para la estabilidad y la mejora a largo plazo. (Véase también “proceso definido” y “elemento de proceso”).

**Control de calidad** (quality control) Las técnicas y las actividades operativas que se usan para satisfacer los requerimientos de calidad. (Véase también “aseguramiento de la calidad”).

**Datos** (data) Información registrada, sin importar la forma o el método de registro, incluyendo datos técnicos, documentos de software de ordenadores, información financiera, información de gestión, representación de hechos, números o datos de cualquier naturaleza que pueden comunicarse, almacenarse y procesarse.

**Definición de proceso** (process definition) El acto de definir y describir un proceso. El resultado de una definición de proceso es una descripción de proceso. (Véase también “descripción de proceso”).

**Desarrollo** (development) En el Conjunto de productos CMMI, no sólo se pueden incluir las actividades de desarrollo, sino también las actividades de mantenimiento. Los proyectos que se benefician de las mejores prácticas CMMI pueden enfocarse en desarrollo, mantenimiento o ambos.

**Desarrollo integrado de producto y de proceso** (integrated product and process development IPPD) Una aproximación sistemática al desarrollo de producto, que logra una colaboración oportuna de las partes interesadas relevantes durante todo el ciclo de vida del producto para satisfacer mejor las necesidades del cliente.

**Descripción de proceso** (process description) Una expresión documentada de un conjunto de actividades realizadas para alcanzar un propósito determinado. Una descripción de proceso proporciona una definición operativa de los principales componentes de un proceso. La descripción específica, de manera completa, precisa y verificable, los requerimientos, el diseño, el comportamiento u otras características de un proceso. Puede también incluir procedimientos para determinar si se han satisfecho estas disposiciones. Se pueden encontrar descripciones de proceso a nivel de actividad, de proyecto o de organización.

**Director** (senior manager) En el Conjunto de productos CMMI, un rol de gestión situado en un nivel lo suficientemente alto en la organización, donde la preocupación principal de la persona que juega este rol es la permanencia de la organización a largo plazo más que las reocupaciones y presiones a corto plazo contractuales y del proyecto. Un director tiene autoridad para dirigir la asignación o reasignación de recursos, para dar soporte a la eficacia de la mejora de procesos de la organización. (Véase también “nivel directivo”). Un director puede ser cualquier gerente que satisface esta descripción, incluyendo el dirigente máximo de la organización. Sinónimos para “director” incluyen “ejecutivo” y “gerente de alto nivel”. Sin embargo, para asegurar la consistencia y la usabilidad, estos sinónimos no se usan en los modelos CMMI.

**Disciplina** (discipline) En el Conjunto de productos CMMI, los corpus de conocimiento disponibles cuando selecciona un modelo CMMI (p. ej., ingeniería de sistemas). El Equipo de Producto CMMI (CMMI Product Team) prevé que otros corpus de conocimiento se integren en el marco CMMI en el futuro.

**Documento** (document) Una colección de datos, sin importar el medio en el que se han registrado, que normalmente permanece y puede ser leído por seres humanos o máquinas. Por tanto, los documentos incluyen tanto los documentos en papel como los electrónicos.

**Ejecutivo** (executive) (Véase “director”) **Elaboración de práctica genérica** (generic practice elaboration) Un componente informativo del modelo, que aparece después de la práctica genérica para proporcionar orientación de cómo debería aplicarse la práctica genérica al área de proceso.

**Elemento de configuración** (configuration item) Una agregación de productos de trabajo que se establece para la gestión de configuración y se trata como una entidad única en el proceso de gestión de configuración. (Véase también “gestión de configuración”).

**Empresa** (enterprise) La composición completa de las compañías. Las compañías pueden consistir en varias organizaciones en varias ubicaciones con distintos clientes (Véase también “organización”).

**Equipo de acción de procesos** (process action team) Un equipo que tiene la responsabilidad de desarrollar e implementar actividades de mejora de procesos para una organización tal como se documentan en un plan de acción de procesos.

**Equipo integrado** (integrated team) Un grupo de personas con habilidades y pericia complementarias, que están comprometidos a entregar los productos de trabajo especificados colaborando de forma oportuna. Los miembros del equipo integrado proporcionan las habilidades y el apoyo apropiados a todas las fases de la vida de los productos de trabajo y son responsables colectivamente de entregar los productos de trabajo según se especificaron. Un equipo integrado debería incluir a los representantes autorizados de las organizaciones, de las disciplinas y de las funciones que tienen un interés en el éxito de los productos de trabajo.

**Estándar** (standard) Cuando vea la palabra *estándar* usada como un nombre en un modelo CMMI, se refiere a los requerimientos formales obligatorios, desarrollados y usados para prescribir aproximaciones coherentes al desarrollo (p. ej., los estándares ISO/IEC, los estándares IEEE y los estándares de la organización). En lugar de usar estándar en su sentido común diario, se usa otro término que significa la misma cosa (p. ej., típico, tradicional, usual o rutinario).

**Estructura de descomposición del trabajo** (WBS) (work breakdown structure) Una disposición de los elementos de trabajo y de su relación entre ellos y con el producto final.

**Evaluación** (appraisal) En el Conjunto de productos CMMI, un examen de uno o más procesos por un equipo de profesionales formados, usando un modelo de evaluación de referencia como base para determinar, como mínimo, fortalezas y debilidades (Véase también “valoración” y “evaluación de capacidad”).

**Evidencia** (evidence) (Véase “evidencia objetiva”).

**Evidencia objetiva** (objective evidence) Tal y como se usa en los materiales de evaluación CMMI, los documentos o resultados de entrevistas usados como indicadores de la implementación o institucionalización de las prácticas del modelo. Fuentes de evidencia objetiva pueden ser instrumentos, presentaciones, documentos y entrevistas.

**Formación** (training) Opciones de aprendizaje formal e informal, que pueden incluir formación en aulas, tutela informal, formación basada en web, autoestudio dirigido, y programas formalizados de formación en el puesto de trabajo. Las opciones de aprendizaje seleccionadas para cada situación se basan en una evaluación de la necesidad de formación y de las carencias de rendimiento a tratarse.

**Gerente** (manager) En el Conjunto de productos CMMI, una persona que proporciona dirección y control técnico y administrativo a aquellos que realizan tareas o actividades dentro del área de responsabilidad del gerente. Las funciones tradicionales de un gerente incluyen el trabajo de planificación, de organización, de dirección y de control en un área de responsabilidad.

**Gestión de cambios** (change management) Uso juicioso de medios para efectuar un cambio, o un cambio propuesto sobre un producto o servicio. (Véase también “gestión de configuración”).

**Gestión de requerimientos** (requirement management) La gestión de todos los requerimientos recibidos o generados por el proyecto, incluyendo tanto los requerimientos técnicos como los no técnicos, así como aquellos requerimientos impuestos al proyecto por la organización.

**Gestión de riesgos** (risk management) Un proceso organizado y analítico para identificar lo que podría causar daño o pérdida (identificar riesgos); para evaluar y cuantificar los riesgos identificados; y para desarrollar y, si es necesario, implementar una aproximación apropiada para prevenir o gestionar las causas de riesgo que podrían dar como resultado daños o pérdidas significativos.

**Grupo de procesos** (process group) Una colección de especialistas que facilita la definición, el mantenimiento y la mejora de los procesos usados por la organización.

**Guías de adaptación** (tailoring guidelines) Las guías de la organización que permiten a los proyectos, a los grupos y a las funciones de la organización, adaptar los procesos estándar apropiadamente para su uso. El conjunto de procesos estándar de la organización se describe a nivel general y puede no ser directamente usable para ejecutar un proceso. Las guías de adaptación ayudan a aquellos que establecen los procesos definidos para los proyectos. Las guías de adaptación cubren (1) la selección de un proceso estándar, (2) la selección de un modelo de ciclo de vida aprobado, y (3) la adaptación del proceso estándar y el modelo de ciclo de vida seleccionados para ajustarse a las necesidades del proyecto. Las guías de adaptación describen qué es lo que puede y no puede modificarse, e identifican los componentes de proceso que son candidatos a modificación.

**Hallazgos** (findings) (Véase “hallazgos de la evaluación”)

**Hallazgos de la evaluación** (appraisal findings) Los resultados de una evaluación que identifican las cuestiones, problemas u oportunidades más importantes para la mejora de procesos, dentro del alcance de la evaluación. Los hallazgos de la evaluación son deducciones sacadas de evidencias objetivas corroboradas.

**Identificación de riesgos** (risk identification) Una aproximación organizada y rigurosa para buscar riesgos probables o realistas en la consecución de los objetivos.

**Ingeniería de hardware** (hardware engineering) La aplicación de una aproximación sistemática, disciplinada y cuantificable, para transformar un conjunto de requerimientos que representan la colección de necesidades, expectativas y restricciones de las partes interesadas, usando técnicas documentadas y tecnología para diseñar, implementar y mantener un producto tangible. (Véase también “ingeniería del software” e “ingeniería de sistemas”). En CMMI, la ingeniería de hardware representa todos los campos técnicos (p. ej., eléctrico o mecánico) que transforman los requerimientos e ideas en productos tangibles y producibles.

**Ingeniería de sistemas** (systems engineering) La aproximación interdisciplinaria que rige el esfuerzo total técnico y de gestión requerido para transformar un conjunto de necesidades, expectativas y restricciones de los clientes en una solución de producto y para dar soporte a esa solución a lo largo de la vida del producto. (Véase también “ingeniería de hardware” e “ingeniería del software”). Esto incluye la definición de las medidas técnicas de rendimiento, la integración de las especialidades de ingeniería para establecer una arquitectura de producto, y la definición de los procesos de soporte del ciclo de vida que aseguran un equilibrio entre los objetivos de coste, de rendimiento y de calendario.

**Ingeniería del software** (software engineering) (1) La aplicación de una aproximación sistemática, disciplinada y cuantificable al desarrollo, a la explotación y al mantenimiento de software. (2) El estudio de las aproximaciones como en (1). (Véase también “ingeniería de hardware” e “ingeniería de sistemas”).

**Institucionalización** (institutionalization) La forma arraigada de funcionamiento que una organización sigue rutinariamente como parte de su cultura corporativa.

**Jefe de proyecto** (project manager) En el Conjunto de productos CMMI, la persona responsable de planificar, dirigir, controlar, estructurar y motivar el proyecto. El jefe de proyecto es responsable de satisfacer al cliente.

**Madurez de la organización** (organizational maturity) El grado en el cual una organización tiene explícita y consistentemente procesos desplegados que están documentados, gestionados, medidos, controlados y mejorados continuamente. La madurez de la organización puede medirse a través de las evaluaciones.

**Marco** (framework) (Véase “marco CMMI”).

**Marco CMMI** (CMMI framework) La estructura base que organiza los componentes CMMI, que incluye elementos comunes de los actuales modelos CMMI, al igual que las reglas y los métodos para generar modelos, métodos de evaluación (incluyendo artefactos asociados) y materiales de formación. El marco permite añadir nuevas disciplinas a CMMI de forma que las nuevas disciplinas se integren con las existentes. (Véase también “modelo CMMI” y “Conjunto de productos CMMI”).

**Medición de proceso** (process measurement) El conjunto de definiciones, métodos y actividades usadas para tomar mediciones de un proceso y de sus productos resultantes, con el propósito de caracterizar y comprender el proceso.

**Mejora de procesos** (process improvement) Un programa de actividades diseñado para mejorar el rendimiento y la madurez de los procesos de la organización y los resultados de dicho programa.

**Mejoras de procesos y de tecnología** (process and technology improvements) Mejoras incrementales e innovadoras a los procesos y a las tecnologías de proceso o de producto.

**Meta** (goal) Un componente CMMI requerido que puede ser una meta genérica o bien una meta específica. Cuando vea la palabra *meta* en un modelo CMMI, se refiere siempre a un componente del modelo (p. ej., meta genérica y meta específica). (Véase también “meta genérica”, “objetivo” y “meta específica”).

**Meta específica** (specific goal) Un componente requerido del modelo que describe las características únicas que deben estar presentes para satisfacer el área de proceso. (Véase también “nivel de capacidad”, “meta genérica”, “objetivos de negocio de la organización” y “área de proceso”).

**Meta genérica** (generic goal) Un componente requerido del modelo, que describe las características que deben estar presentes para institucionalizar los procesos que implementan un área de proceso. (Véase también “institucionalización”).

**Modelo CMMI** (CMMI model) Uno de los modelos posibles de la colección completa que pueden generarse a partir del marco CMMI. Dado que el marco CMMI puede generar distintos modelos basándose en las necesidades de la organización que esté usándolo, existen múltiples modelos CMMI (Véase también “marco CMMI” y “Conjunto de productos CMMI”).

**Modelo de ciclo de vida** (lifecycle model) Una partición en fases de la vida de un producto o proyecto.

**Modelo de madurez y de capacidad** (capability maturity model) un modelo que contiene los elementos esenciales de procesos eficaces para una o más disciplinas, y que describe un camino de mejora evolutiva desde procesos inmaduros ad hoc a procesos maduros disciplinados con eficacia y calidad mejorada.

**Modelo de referencia** (reference model) Un modelo que se usa como punto de referencia para medir algún atributo.

**Modelo de referencia de evaluación** (appraisal reference model) Tal y como se usa en los materiales de evaluación CMMI, el modelo CMMI al cual un equipo de evaluación correlaciona las actividades de proceso implementadas.

**Modelo de rendimiento de proceso** (process performance model) Una descripción de las relaciones entre los atributos de un proceso y sus productos de trabajo, que se desarrolla a partir de los datos históricos de rendimiento de proceso y se calibra usando las medidas recogidas de proceso y de producto del proyecto, y que se usa para predecir los resultados que serán obtenidos siguiendo el proceso.

**Nivel de capacidad** (capability level) Logro de la mejora de procesos dentro de un área de proceso individual. Un nivel de capacidad se define por las prácticas específicas y genéricas apropiadas para un área de proceso (Véase también “meta genérica”, “práctica genérica”, “nivel de madurez”, y “área de proceso”).

**Nivel de madurez** (maturity level) Grado de mejora de procesos a través de un conjunto predefinido de áreas de proceso en las que se alcanzan todas las metas del conjunto. (Véase también “nivel de capacidad” y “área de proceso”).

**Nivel directivo** (higher level management) La persona o personas que proporcionan la política y la orientación global para el proceso, pero no proporcionan la monitorización y el control directo y cotidiano del proceso. Dichas personas pertenecen a un nivel de gestión en la organización por encima del nivel inmediato responsable del proceso y pueden ser (aunque no necesariamente) directores (Véase también “director”).

**Objetivo** (objective) Cuando se usa como un nombre en el Conjunto de productos CMMI, el término *objetivo* (objective) reemplaza a la palabra *meta* (goal), tal y como se usa en su sentido cotidiano corriente, ya que la palabra *meta* (goal) se reserva para usarse cuando se refiere a los componentes del modelo CMMI llamados metas (goals) específicas y metas (goals) genéricas. (Véase también “meta”).

**Objetivo cuantitativo** (quantitative objective) Valor objetivo deseado expresado en medidas cuantitativas. (Véase también “objetivos de mejora de procesos” y “objetivos de calidad y de rendimiento del proceso”).

**Objetivos de calidad y de rendimiento del proceso** (quality and process performance objectives) Objetivos y requerimientos de la calidad del producto, de la calidad del servicio y del rendimiento del proceso. Los objetivos de rendimiento del proceso incluyen la calidad; sin embargo, para enfatizar la importancia de la calidad en el Conjunto de productos CMMI, se usa la frase *objetivos de calidad y de rendimiento del proceso* en lugar de *objetivos de rendimiento del proceso*.

**Objetivos de mejora de proceso** (process improvement objectives) Un conjunto de características objetivo establecidas para guiar el esfuerzo para mejorar un proceso existente, de manera específica y medible, tanto en términos de características del producto resultante (p. ej., calidad, rendimiento y conformidad con los estándares) como en la manera en la que se ejecuta el proceso (p. ej., eliminación de etapas redundantes, combinación de etapas de proceso y mejora del tiempo de ciclo). (Véase también “objetivos de negocio de la organización” y “objetivo cuantitativo”).

**Objetivos de negocio** (business objectives) (Véase “objetivos de negocio de la organización”).

**Objetivos de negocio de la organización** (organization’s business objectives) Estrategias diseñadas por la dirección para asegurar la existencia continuada de la organización y fomentar su rentabilidad, cuota de mercado y otros factores que influyen en el éxito de la organización. (Véase también “objetivos de calidad y de rendimiento del proceso” y “objetivo cuantitativo”). Dichos objetivos pueden incluir la reducción del número de peticiones de cambios durante la fase de integración del sistema, la reducción del tiempo del ciclo de desarrollo, el incremento del número de errores encontrados en la primera o segunda fase de desarrollo del producto y la reducción del número de defectos informados por clientes, cuando se aplica a actividades de ingeniería de sistemas.

**Organización** (organization) Una estructura administrativa en la que la gente gestiona colectivamente uno o más proyectos como un todo, y cuyos proyectos comparten un director y operan bajo las mismas políticas. Sin embargo, la palabra “organización” tal y como se usa en todos los modelos CMMI, también puede aplicarse a una persona que realiza una función en una pequeña organización que podría ser realizada por un grupo de gente en una organización grande. (Véase también “empresa” y “unidad organizativa”).

**Parámetros de rendimiento** (performance parameters) Las mediciones de eficacia y otras medidas clave usadas para guiar y controlar el desarrollo progresivo.

**Parte interesada** (stakeholder) En el Conjunto de productos CMMI, un grupo o individuo que se ve afectado por o es de alguna manera responsable del resultado de un proyecto. Las partes interesadas pueden incluir a los miembros del proyecto, los proveedores, los clientes, los usuarios finales y otros. (Véase también “cliente” y “parte interesada relevante”).

**Participantes en la evaluación** (appraisal participants) Miembros de la unidad de la organización que participan proporcionando información durante la evaluación.

**Perfil** (profile) (Véase “perfil alcanzado” y “perfil objetivo”).

**Perfil alcanzado** (achievement profile) En la representación continua, una lista de áreas de proceso y sus correspondientes niveles de capacidad, que representan el progreso de la organización para cada área de proceso, según avanza a través de los niveles de capacidad. (Véase también “perfil de nivel de capacidad”, “perfil objetivo” y “progresión hacia un nivel objetivo”).

**Perfil objetivo** (target profile) En la representación continua, una lista de áreas de proceso y sus niveles de capacidad correspondientes que representan un objetivo de mejora de procesos. (Véase también “perfil logrado” y “perfil de nivel de capacidad”).

**Plan de proyecto** (Project plan) Un plan que proporciona la base para ejecutar y controlar las actividades del proyecto, las cuales tratan los compromisos con el cliente del proyecto. La planificación del proyecto incluye estimar los atributos de los productos de trabajo y de las tareas, determinar los recursos necesarios, negociar los compromisos, elaborar un calendario, e identificar y analizar los riesgos del proyecto. Puede ser necesaria la iteración entre estas actividades para establecer el plan de proyecto.

**Política** (policy) (Véase “política de la organización”).

**Política de la organización** (organizational policy) Un principio de guía establecido normalmente por la dirección, que se adopta por una organización para influenciar y determinar decisiones.

**Práctica alternativa** (alternative practice) Una práctica que es un sustituto para una o más prácticas genéricas o específicas contenidas en los modelos CMMI, que alcanza un efecto equivalente a satisfacer la meta genérica o específica asociada con las prácticas del modelo. Las prácticas alternativas no son necesariamente reemplazo una-por-una para las prácticas genéricas o específicas.

**Práctica específica** (specific practice) Un componente esperado del modelo que se considera importante para alcanzar la meta específica asociada. Las prácticas específicas describen las actividades esperadas para dar como resultado el logro de las metas específicas de un área de proceso. (Véase también “área de proceso” y “meta específica”).

**Práctica genérica** (generic practice) Un componente esperado del modelo que se considera importante para alcanzar las metas genéricas asociadas. Las prácticas genéricas asociadas con una meta genérica describen las actividades que se espera resulten en el logro de la meta genérica y contribuyan a la institucionalización de los procesos asociados con el área de proceso.

**Procedimiento de prueba** (test procedure) Instrucciones detalladas para el establecimiento, ejecución y evaluación de los resultados de una determinada prueba.

**Proceso** (process) En el Conjunto de productos CMMI, las actividades que pueden reconocerse como implementaciones de las prácticas en un modelo CMMI. Estas actividades pueden corresponderse con una o más prácticas en las áreas de proceso de CMMI para permitir que un modelo sea útil para la mejora de procesos y la evaluación de procesos. Hay un uso especial de la frase “el proceso” en las declaraciones y descripciones de las metas genéricas y las prácticas genéricas. “El proceso”, tal y como se usa en la Parte Dos, es el proceso o procesos que implementan el área de proceso.

**Proceso definido del proyecto** (project’s defined process) El proceso integrado y definido que se adapta a partir del conjunto de procesos estándar de la organización.

**Proceso en optimización** (optimizing process) Un proceso gestionado cuantitativamente que es mejorado basándose en una comprensión de las causas comunes de variación inherentes al proceso. El enfoque de un proceso en optimización está en mejorar continuamente el rango de rendimiento del proceso a través de mejoras tanto incrementales como innovadoras. (Véase también “causa común de variación del proceso”, “proceso definido” y “proceso gestionado cuantitativamente”).

**Proceso estándar** (standard process) Una definición operativa del proceso básico que guía el establecimiento de un proceso común en una organización. Un proceso estándar describe los elementos de proceso fundamentales que son esperados para incorporarse a cualquier proceso definido. También describe las relaciones (p. ej., ordenación e interfaces) entre estos elementos de proceso.

**Producto** (product) En el Conjunto de productos CMMI, un producto de trabajo que está previsto entregar a un cliente o usuario final. La forma de un producto puede variar según el contexto. (Véase también “cliente”, “componente de producto”, “servicio” y “producto de trabajo”).

**Producto de trabajo** (work product) En el Conjunto de productos CMMI, un resultado útil de un proceso. Esto puede incluir ficheros, documentos, productos, partes de un producto, servicios, descripciones de proceso, especificaciones y facturas. Una distinción clave entre un producto de trabajo y un componente de producto es que un producto de trabajo no es necesariamente parte del producto. En los modelos CMMI, se verá la frase *productos de trabajo y servicios*. Aunque la definición de producto de trabajo incluye los servicios, esta frase se usa para enfatizar la inclusión de servicios.

**Productos de trabajo típicos** (typical work products) Un componente informativo del modelo que proporciona ejemplos de resultados de una práctica específica. Estos ejemplos se denominan productos de trabajo típicos porque a menudo hay otros productos de trabajo que son igual de eficaces pero no están enumerados.

**Prototipo** (prototype) Un tipo, forma o instancia preliminar de un producto o componente de producto que sirve como modelo para etapas posteriores o para la versión final y completa del producto. Este modelo (p. ej., físico, electrónico, numérico o analítico) puede usarse entre otros, para los siguientes propósitos:

- Evaluación de la viabilidad de una tecnología nueva o no familiar.
- Evaluación o mitigación de un riesgo técnico.
- Validación de los requerimientos.
- Demostración de características críticas.
- Cualificación de un producto.
- Cualificación de un proceso.
- Caracterización del rendimiento o del producto.
- Explicación de principios físicos.

**Proveedor** (supplier) (1) Una entidad que entrega productos o realiza servicios que han sido adquiridos. (2) Un individuo, sociedad, empresa, corporación, asociación u otros servicios que tienen un acuerdo (contrato) con un comprador para el diseño, el desarrollo, la fabricación, el mantenimiento, la modificación o el suministro de elementos bajo los términos de un acuerdo (contrato).

**Proyecto** (project) En el Conjunto de productos CMMI, un conjunto gestionado de recursos interrelacionados que entrega uno o más productos a un cliente o usuario final. Un proyecto tiene un comienzo concreto (es decir, el arranque del proyecto) y opera normalmente de acuerdo a un plan. Dicho plan está frecuentemente documentado y especifica qué es lo que se va a entregar o implementar, los recursos y los fondos que van a usarse, el trabajo que va a realizarse y el calendario para hacer el trabajo. Un proyecto puede estar compuesto de proyectos.

**Prueba de aceptación** (acceptance testing) Prueba formal llevada a cabo para permitir a un usuario, a un cliente o a otra entidad autorizada determinar si aceptan un producto o componente de producto. (Véase también “prueba unitaria”).

**Prueba unitaria** (unit testing) Pruebas de unidades individuales o de grupos de unidades relacionadas de hardware o de software. (Véase también “prueba de aceptación”).

**Referencia** (reference) Un componente informativo del modelo, que apunta a información adicional o más detallada de áreas de proceso relacionadas.

**Rendimiento de proceso** (process performance) Una medida de los resultados reales alcanzados al seguir un proceso. Se caracteriza tanto por las medidas de proceso (p. ej., esfuerzo, tiempo de ciclo y eficacia en la eliminación de defectos) como por medidas de producto (por ejemplo, fiabilidad, densidad de defectos y tiempo de respuesta).

**Representación** (representation) La organización, uso y presentación de los componentes del CMM. En general, son evidentes dos tipos de aproximación para presentar las mejores prácticas: la representación por etapas y la representación continua.

**Requerimiento** (requirement) (1) Una condición o capacidad necesitada por un usuario para solucionar un problema o lograr un objetivo. (2) Una condición o capacidad que debe cumplir o poseer un producto o componente de producto para satisfacer un contrato, un estándar, una especificación u otros documentos impuestos formalmente. (3) Una representación documentada de una condición o capacidad como en (1) o en (2).

**Requerimientos de cliente** (customer requirements) El resultado de educir, consolidar y resolver los conflictos entre las necesidades, las expectativas, las limitaciones y las interfaces de las partes interesadas relevantes del producto de una forma que sea aceptable para el cliente (Véase también “cliente”).

**Requerimientos de componente de producto** (product component requirements) Una especificación completa de un componente de producto, incluyendo el ajuste, la forma, la función, el rendimiento y cualquier otro requerimiento.

**Requerimientos de producto** (product requirements) Un refinamiento de los requerimientos de cliente en el lenguaje de los desarrolladores, transformando los requerimientos implícitos en requerimientos derivados explícitos. (Véase también “requerimientos derivados” y “requerimientos de componente de producto”). El desarrollador usa los requerimientos de producto para guiar el diseño y la construcción del producto.

**Requerimientos derivados** (derived requirements) Requerimientos que no están indicados explícitamente en los requerimientos de cliente, pero son deducidos (1) de los requerimientos contextuales (p. ej., estándares aplicables, leyes, políticas, prácticas comunes y decisiones de la gerencia) o (2) de los requerimientos necesarios para especificar un componente de producto. Los requerimientos derivados pueden surgir también durante el análisis y el diseño de los componentes del producto o del sistema. (Véase también “requerimientos de producto”).

**Requerimientos no técnicos** (non technical requirements) Estipulaciones, compromisos, condiciones y términos contractuales que afectan a cómo se adquieren los productos o los servicios. Algunos ejemplos son productos que van a ser entregados, derechos de datos para elementos no desarrollados (NDI) de componentes comerciales (COTS) entregados, fechas de entrega e hitos con criterios de salida. Otros requerimientos no técnicos incluyen los requerimientos de formación, los requerimientos del emplazamiento y los calendarios del despliegue.

**Requerimientos técnicos** (technical requirements) Propiedades (atributos) de productos o servicios que van a ser adquiridos o desarrollados.

**Retorno de la inversión** (return on investment) El ratio de retorno entre los ingresos del producto y los costes de producción, que determina si una organización se beneficia de la producción de ese producto.

**Servicio** (service) En el Conjunto de productos CMMI, un servicio es un producto que es intangible y no almacenable. (Véase también “producto”, “cliente” y “producto de trabajo”).

**Subpráctica** (subpractice) Un componente informativo del modelo que proporciona guías para interpretar e implementar una práctica específica o genérica. Las subprácticas pueden redactarse como si fueran obligatorias, pero de hecho sólo pretenden proporcionar ideas que pueden ser útiles para la mejora de procesos.

**Subproceso** (subprocess) Un proceso que es parte de un proceso mayor. Un subproceso puede, a su vez, descomponerse en subprocesos y/o elementos de proceso. (Véase también “proceso”, “descripción de proceso” y “elemento de proceso”).

**Trazabilidad** (traceability) Una asociación discernible entre dos o más entidades lógicas, tales como requerimientos, elementos de sistema, verificaciones o tareas. (Véase también “trazabilidad bidireccional” y trazabilidad de requerimientos”).

**Trazabilidad bidireccional** (bidirectional traceability) Una asociación entre dos o más entidades lógicas que es discernible en ambos sentidos (es decir, hacia y desde una entidad). (Véase también “trazabilidad de requerimientos” y “trazabilidad”).

**Trazabilidad de requerimientos** (requirement traceability) Una asociación discernible entre los requerimientos y los requerimientos relacionados, las implementaciones y las verificaciones. (Véase también “trazabilidad bidireccional” y “trazabilidad”).

**Unidad organizativa** (organizational unit) La parte de una organización que es objeto de una evaluación. Una unidad organizativa despliega uno o más procesos que tienen un contexto de procesos coherente y opera dentro de un conjunto coherente de objetivos de negocio. Una unidad organizativa es normalmente parte de una organización mayor, aunque en una organización pequeña, la unidad organizativa puede ser toda la organización.

**Validación** (validation) Confirmación de que el producto, tal y como se ha proporcionado (o será proporcionado), satisfará su uso previsto. En otras palabras, la validación asegura que “se ha construido el producto correcto”. (Véase también “verificación”).

**Valoración** (assessment) En el Conjunto de productos CMMI, una evaluación que una organización hace internamente con el propósito de mejora de procesos. La palabra *valoración* también se usa en el conjunto de productos CMMI en su sentido cotidiano (p. ej., valoración de riesgos). (Véase también “evaluación” y “evaluación de capacidad”).

**Verificación** (verification) Confirmación de que los productos de trabajo reflejan apropiadamente los requerimientos que se han especificado para ellos. En otras palabras, la verificación asegura que “se construyó correctamente el producto”. (Véase también “validación”).

## Apéndice B: Bibliografía

---

OMG. (s.f.). *Unified Modeling Language 2.4.1*. Obtenido de OMG Specifications: [www.omg.org](http://www.omg.org)

Software Engineering Institute. (2010). *CMMI for development, Version 1.3*.

Software Engineering Institute. (June 1993). *Taxonomy-Based Risk Identification*.

Universitat Oberta de Catalunya. (s.f.). *Gestió i desenvolupament de projectes*.

Universitat Oberta de Catalunya. (s.f.). *Gestión y Organización de Proyectos Informáticos*. UOC.

Universitat Oberta de Catalunya. (s.f.). *Ingeniería de Software*.

Universitat Oberta de Catalunya. (s.f.). *Introducción a la Ingeniería de software OO*.

Universitat Oberta de Catalunya. (s.f.). *Presentació de documents i elaboració de presentacions*.

Universitat Oberta de Catalunya. (s.f.). *Redacció de textos científicotècnics*.

Universitat Oberta de Catalunya. (s.f.). *Treball final de carrera*.