



**IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE COMPETENCIAS PARA  
INSTITUCIONES EDUCATIVAS**

**ALEJANDRA DUQUE CEBALLOS**

**UNIVERSIDAD AUTÓNOMA DE MANIZALES**

**MAESTRIA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE**

**MANIZALES**

**(I COHORTE)**

**2015**

**IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE COMPETENCIAS PARA  
INSTITUCIONES EDUCATIVAS**

**ALEJANDRA DUQUE CEBALLOS**

**Proyecto de Grado para optar al título de Magister en Gestión y Desarrollo de  
Proyectos de Software**

**Asesor Temático y Metodológico**

**Ingeniera de Sistemas Adriana María Giraldo Osorio**

**UNIVERSIDAD AUTÓNOMA DE MANIZALES**

**MAESTRIA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE**

**MANIZALES**

**2015**

## CONTENIDO

INTRODUCCIÓN .....	6
1 REFERENTE CONTEXTUAL .....	8
1.1 Descripción del Área Problemática .....	8
1.2 ANTECEDENTES .....	8
1.2.1 SOFIA PLUS .....	8
1.2.2 COMFAMA .....	9
1.2.3 HUMAN RESOURCE SYSTEMS GROUP .....	9
1.2.4 SENTRICO .....	10
1.2.5 IHRDC .....	10
1.3 JUSTIFICACIÓN .....	11
1.4 FORMULACIÓN DEL PROBLEMA .....	12
1.5 OBJETIVOS .....	13
1.5.1 OBJETIVO GENERAL .....	13
1.5.2 OBJETIVOS ESPECÍFICOS .....	13
1.6 PLANTEAMIENTO DEL PROBLEMA .....	13
1.7 RESULTADOS ESPERADOS .....	14
2 ESTRATÉGIA METODOLÓGICA .....	15
2.1 DISEÑO METODOLÓGICO .....	21
2.1.1 Construcción de la propuesta o del documento proyecto <b>¡Error! Marcador no definido.</b>	
2.2 PRUEBAS .....	23
2.3 PRESUPUESTO .....	23
2.4 CRONOGRAMA .....	23
3 DESARROLLO los inicios de capitulo no van centrados?? .....	25
3.1 REFERENTE TEÓRICO .....	15
3.1.1 COMPETENCIAS .....	15

3.1.2	SISTEMAS DE INFORMACIÓN.....	16
3.1.3	PSP Personal Software Process.....	17
3.1.4	Proceso Unificado UP.....	19
	algo de prototipos funcionales .....	<b>¡Error! Marcador no definido.</b>
3.2	DESARROLLO DEL PROYECTO .....	25
3.2.1	Primera iteración, fase de inicio: .....	25
3.2.2	Segunda iteración, fase de elaboración. ....	36
3.2.3	Tercera iteración, etapa de construcción.....	51
4	CONCLUSIONES es un capitulo se parte pagina .....	66
5	RECOMENDACIONES .....	68
6	Bibliografía.....	70

## LISTA DE FIGURAS

<i>Figura 1, Clasificación operativa y administrativa de los sistemas de información (O'Brien &amp; Marakas, 2006)</i> .....	16
<i>Figura 2 Proceso de PSP (Humphrey, PSP(sm): A self-Improvement Process for Software Engineers, 1995)</i> .....	18
<i>Figura 3 - Elementos de UP</i> .....	19
<i>Figura 4 - Cinco flujos de trabajo: requisitos, análisis, diseño, implementación y prueba, cada una de ellas tienen lugar sobre las cuatro fases: inicio, elaboración, construcción y transición. (BOOCH, 2000)</i> .....	21
<i>Figura 5 - Diagrama de contexto del sistema.</i> .....	26
<i>Figura 6 - Casos de uso de alto nivel.</i> .....	29
<i>Figura 7 – Subsistemas</i> .....	31
<i>Figura 8 - Diagrama de componentes</i> .....	32
<i>Figura 9 - Modelo entidad relación iteración 1</i> .....	33
<i>Figura 10 - CRUD administrar Facultades</i> .....	34
<i>Figura 11 - Diagrama del caso de uso 2, Administrar competencias.</i> .....	36
<i>Figura 12 - Diagrama del caso de uso 3, Valorar competencias.</i> .....	44
<i>Figura 13 - Diagrama entidad relación</i> .....	46
<i>Figura 14 - Asociar subcompetencias con asignaturas y niveles</i> .....	49
<i>Figura 15 - Diagrama del caso de uso 4, Generar consultas y reportes.</i> .....	52
<i>Figura 16 – Diseño reporte Valoración de competencias</i> .....	58
<i>Figura 17 - Consulta por estudiante</i> .....	59
<i>Figura 18 - Reporte por asignatura</i> .....	59

<i>Figura 19 - Consulta por docente .....</i>	<i>60</i>
<i>Figura 20 - Reporte por docente.....</i>	<i>60</i>
<i>Figura 21 - Vista preliminar de valorar competencias.....</i>	<i>61</i>
<i>Figura 22 - Vista preliminar del formulario para cargar información externa .....</i>	<i>61</i>
<i>Figura 23 - Gráfico de tiempo total por fases .....</i>	<i>65</i>

ANEXOS

Anexo A PRESUPUESTO

70

## INTRODUCCIÓN

El sistema de educación superior de Colombia propone realizar los procesos de formación y evaluación basados en competencias. Es por ello que se hace necesario que las instituciones educativas de educación superior (IES) tengan sistemas que permitan de una manera ágil y confiable llevar el registro y seguimiento del desarrollo de las competencias de sus estudiantes.

En los últimos años varias Instituciones educativas han adoptado el modelo de formación por competencias con el fin de alcanzar un conocimiento holístico por parte del estudiante y para formarlo acorde a las necesidades laborales del medio.

Las competencias, son definidas por (Vasco, 2003) como “el conjunto de conocimientos, habilidades, actitudes, comprensiones y disposiciones cognitivas, metacognitivas, socioafectivas y psicomotoras apropiadamente relacionadas entre sí para facilitar el desempeño flexible, eficaz y con sentido de una actividad o de cierto tipo de tareas en contextos nuevos y retadores”. Se plantean también como el desarrollo de competencias generales y específicas que le permiten al estudiante actuar basado en conocimientos propios de su profesión y su perfil de acción.

El presente proyecto surge de la necesidad de diseñar e implementar un sistema de gestión de competencias que articule la implementación, evaluación y el seguimiento de competencias básicas en los estudiantes de las instituciones educativas de educación superior.

El sistema le permitirá a la institución administrar el proceso de desarrollo de las competencias de los estudiantes y apoyará a la administración curricular de dichas instituciones para mejorar los procesos de autoevaluación de programas, que deseen acreditar y a la vez medir la calidad de los mismos utilizando los informes emitidos por el sistema basados en guías coherentes con los criterios, características, variables e indicadores establecidos.



Basándose en la metodología de Proceso Unificado<sup>1</sup>, apoyado en PSP<sup>2</sup> para la medición del esfuerzo invertido, se pretende desarrollar un sistema que permita a los administradores del currículo definir las competencias de cada área del conocimiento administrar las competencias y niveles definidos por la institución, al docente registrar la evolución de las competencias que debe orientar en su materia y al estudiante que le permita conocer los estándares con los cuales será evaluado.

---

<sup>1</sup> Unified Process UP

<sup>2</sup> Personal Software Process

## **1 REFERENTE CONTEXTUAL**

### **1.1 DESCRIPCIÓN DEL ÁREA PROBLÉMICA**

La Educación Superior es un proceso permanente que posibilita el desarrollo de las potencialidades del ser humano de una manera integral, se realiza con posterioridad a la educación media o secundaria y tiene por objeto el pleno desarrollo de los alumnos y su formación académica o profesional. (Ley 30, Art. 1)

Desde esta perspectiva, el concepto de formación integral hace referencia a un saber, saber hacer y saber ser, que se constituye en la base de la formación de un profesional y un ciudadano idóneo y competente. Esto implica, integrar conocimientos, habilidades, motivos y valores, que se construyen en el proceso de interacción social y que expresan la autodeterminación de la persona en el ejercicio eficiente y responsable de su profesión (L, 2006).

Cuando las instituciones adoptan el modelo de formación por competencias, por lo general no poseen un sistema adecuado para su seguimiento por parte de educadores y administrativos y para la verificación de los avances por parte del estudiante.

Es el caso de la Universidad Autónoma de Manizales que no tiene un sistema para llevar registro de evolución de competencias de los estudiantes de manera cualitativa, por lo cual se busca diseñar un mecanismo de registro y seguimiento de estas competencias para facilitar la consulta y seguimiento de las mismas.

### **1.2 ANTECEDENTES**

A continuación se presentan algunos antecedentes de sistemas de información relacionados con el objeto de este proyecto.

#### **1.2.1 SOFIA PLUS**

El Servicio Nacional de Aprendizaje SENA, posee un Sistema de información denominado SOFIA Plus, el cual es una herramienta de gestión académica que se usa para la ejecución de la formación, labores de administración educativa y para administrar la oferta educativa del SENA. (SENA, 2008)

SOFIA Plus es un sistema que permite al aspirante consultar ofertas educativas, realiza la programación de las actividades de formación de un aprendiz, ya que puede visualizar su ruta de aprendizaje o avances alcanzados en su etapa de formación. También sirve como fuente de consulta para el empresario, ya que le permite verificar información sobre solicitudes y programas de certificaciones existentes.

Este sistema es importante como referencia porque permite calificar cualitativamente las competencias de los estudiantes y a éstos les permite consultar sus resultados y hacer seguimiento a su aprendizaje.

### **1.2.2 COMFAMA**

También la Caja de Compensación Familiar de Antioquia, en convenio con Corpoeducación<sup>3</sup> realizó un proyecto empresarial de competencias con el ánimo de mejorar el desempeño de los trabajadores. Este sistema computacional consta de varias partes, una prueba empresarial de competencia: matemática, lectura y escritura que se diseñó con el fin de medir los niveles de los empleados y que pueden ser aplicadas para selección de personal ó para identificar necesidades de formación de los trabajadores; consta también de un curso de formación en competencias laborales generales con el fin de que los trabajadores puedan mejorar sus habilidades como empleados basados en los principios de la formación por competencias; y por último consta de una evaluación de competencias laborales generales que responde a las necesidades detectadas en las organizaciones para poder evaluar el desempeño del trabajador siguiendo un enfoque de competencias.

### **1.2.3 HUMAN RESOURCE SYSTEMS GROUP**

Es una compañía que proporciona soluciones basadas en competencias de gestión del talento humano que aborda la mejora del rendimiento, retención del conocimiento y el proceso de sucesión de cargos.

---

<sup>3</sup> Corporación para el desarrollo de la educación básica

Ofrecen un sistema SAAS<sup>4</sup> llamado I-SkillSuite que reúne las buenas prácticas aplicadas a la empresa, perfiles de la totalidad de los cargos, competencias a desarrollar y registro de los problemas presentados con su respectiva solución para realizar una gestión apropiada del conocimiento.

Sirve como apoyo para la selección de personal y permite crear planes de capacitaciones al personal idóneo.

Esta referencia es relevante ya que permite crear perfiles del talento humano, dependiendo de las competencias que posea. (Human Resource Systems Group, 2012)

#### **1.2.4 SENTRICO**

Se dedica a la gestión de las competencias de tipo técnico, permiten ver por medio de indicadores el porcentaje de desarrollo del empleado en cierta competencia, ya sea básica, de soporte o general, genera reportes y compara áreas de la empresa.

Es importante para este proyecto ya que analiza las competencias del área técnica que la persona posee o necesita adquirir. (Sentrico, 2012)

#### **1.2.5 IHRDC**

Son empresas que se dedican al desarrollo de sistemas de gestión de competencias enfocados a las empresas, ya que con estos mecanismos pueden realizar un control de las funciones desempeñadas por los empleados y tener unos criterios de selección de personal más adecuados y pertinentes a los requerimientos de la empresa.

También tienen poderosas herramientas para identificar los vacíos existentes en las funciones de los empleados y ayudan a retener el patrimonio intangible que es el conocimiento.

---

<sup>4</sup> Software as a Service

Las herramientas existentes serán de ayuda a la hora de asociar las competencias a los estudiantes en el sistema propuesto. (IHRDC, 2012)

### **1.3 JUSTIFICACIÓN**

De acuerdo con los planteamientos de (RUIZ DE VARGAS, 2005), en la llamada sociedad del conocimiento, “el principal motor del desarrollo económico de organizaciones y naciones es la capacidad de producir y gestionar el saber, donde la universidad asume un papel protagónico que le exige replantear sus relaciones con la sociedad” y sus dinámicas internas de trabajo.

Ante estos cambios inminentes y las nuevas condiciones sociales, económicas y culturales, las organizaciones deben competir con el más grande recurso con que cuentan, y este recurso lo constituyen sus miembros, que deben ser capaces de generar resultados, que puedan adquirir conocimientos relevantes y útiles para el buen desempeño de su labor.

En ese sentido, las exigencias educativas actuales subrayan la necesidad de generar transformaciones en las estructuras curriculares, centradas en el diseño de ambientes pedagógicos y didácticos que propicien experiencias de aprendizaje, que estimulen el desarrollo de las potencialidades del ser humano, por eso es prioridad para las instituciones educativas de educación superior ofrecer una formación de calidad y con total transparencia de los contenidos y de la gestión y evaluación de los mismos, pero es problemático para la administración curricular de dichas instituciones tanto el diagnosticar el desarrollo de competencias en los estudiantes, como la evaluación, seguimiento y mejoramiento de las mismas.

Un sistema de gestión de competencias del estudiante debe ser capaz de organizar, sistematizar, dar a conocer a los diferentes actores los principios, metodologías y resultados de un proceso de evaluación por competencias.

Algunos tipos de plataformas que se adquieren externamente no facilitan la adaptación al entorno educativo de una institución, debido a que son estándares, o de tipo comercial que impiden cualquier tipo de modificación.

Por esto, se necesita desarrollar un sistema para gestionar las competencias que sea fácil de administrar, manejar y modificar si es necesario, como también que sea agradable para estudiantes y docentes, con las herramientas necesarias para cumplir con los objetivos trazados.

Teniendo en cuenta principalmente las necesidades del docente y de la institución, además el uso de un sistema de gestión de competencias también puede orientarse hacia estudiantes permitiéndoles realizar seguimiento de sus avances y el desarrollo de sus competencias.

Por tal motivo con este proyecto se busca desarrollar un sistema de gestión de competencias que presente características ajustadas a los requerimientos de una institución educativa en especial de la UAM®, que brinde funcionalidades como: Administración de competencias, consultas y seguimiento a los estudiantes.

Para promover y seguir los lineamiento que permitan la calidad del SW se busca aplicar métodos efectivos como el conjunto de métricas y procesos PSP, desarrollado por Watts Humphrey en la Carnegie Mellon University, que se diseñaron para ayudar a los desarrolladores a planear, hacer seguimiento al trabajo y establecer metas que se puedan medir y cuantificar. Y de igual forma para aportar un proyecto que contenga métricas que le permitan al grupo de investigación de la Universidad Autónoma de Manizales UAM® en ingeniería de software, aumentar su base de proyectos.

También, como ejercicio académico de la maestría, este trabajo busca hacer énfasis en el desarrollo de un sistema de información, teniendo en cuenta que la formación de pregrado de la desarrolladora de este trabajo de grado no es en el área de las ciencias computacionales y por lo tanto el proyecto le aportará las competencias necesarias para obtener el grado de magister, además de ampliar su perfil profesional de ingeniera electrónica.

#### **1.4 FORMULACIÓN DEL PROBLEMA**

A partir de un sistema transaccional se pueden registrar y hacer seguimiento a las competencias que la UAM® defina para sus programas académicos.

## **1.5 OBJETIVOS**

### **1.5.1 OBJETIVO GENERAL**

Desarrollar un prototipo de un sistema de información para gestionar competencias, que le permitan a la UAM® administrar sus lineamientos de competencias y el seguimiento a la evolución de sus estudiantes en este aspecto, siguiendo la metodología UP.

### **1.5.2 OBJETIVOS ESPECÍFICOS**

- Realizar el análisis y diseño del sistema computacional parametrizado de acuerdo con los lineamientos de competencias definidos por la UAM®.
- Identificar los lineamientos que la UAM® ha definido para la construcción de un modelo de competencias e incorporarlo en el diseño del sistema a desarrollar.
- Determinar el esfuerzo invertido por el desarrollador y el tamaño de software elaborado, aplicando métricas de desarrollo de SW siguiendo el modelo PSP.
- Implementar un prototipo parametrizado de gestión de competencias para la UAM® siguiendo los lineamientos de la metodología UP.

## **1.6 PLANTEAMIENTO DEL PROBLEMA**

Los directivos y docentes de las instituciones educativas de nivel superior de la región encuentran dificultades en la implementación y evaluación de competencias ya que no cuentan con un sistema efectivo que les permita el ingreso, seguimiento, administración y valoración de las mismas.

Actualmente, en la Universidad Autónoma de Manizales UAM®, se requiere de una herramienta computacional por la cual se puedan gestionar las competencias que los estudiantes desarrollan en su proceso de formación, ya que es necesario para un docente realizar el seguimiento en la evolución y desarrollo de dichas competencias.

## 1.7 RESULTADOS ESPERADOS

Resultado / Producto esperado	Indicador	Descripción
<b>Desarrollo fase 1</b>	Software	Primera versión de la aplicación con funcionalidad para el registro y administración de los datos de las facultades, los programas, las asignaturas y niveles de competencia.
<b>Desarrollo fase 2</b>	Software	Módulo de administración de competencias y subcompetencias, y asociación con las asignaturas y niveles de competencia.
<b>Desarrollo fase 3</b>	Software	Módulo de valoración de competencias, carga de información desde registro académico y generación de reportes.
<b>Manual del usuario</b>	Documento	Dirigido a quienes van a usar el sistema. Dar a conocer a los usuarios finales las características y las formas de funcionamiento del software.
<b>Manual técnico</b>	Documento	Explica la instalación y configuración del sistema.
<b>Documento de tesis</b>	Documento	Documento que contiene toda la información relacionada con el desarrollo del proyecto.



## **2 ESTRATÉGIA METODOLÓGICA**

A continuación se expone la metodología, en la cual se plantean la realización de las diferentes etapas y actividades, y algunas bases teóricas necesarias para el desarrollo del proyecto.

### **2.1 REFERENTE TEÓRICO**

En el desarrollo de todo proyecto se necesitan definir las diferentes actividades de las cuales se compone, desde la idea inicial hasta su implementación y puesta a punto, se deben tener en cuenta los conceptos esenciales de los temas concernientes de cada etapa del proyecto.

#### **2.1.1 COMPETENCIAS**

De acuerdo a (Proyecto Tuning América Latina, 2004) la competencia no se puede reducir al simple desempeño laboral, tampoco a la sola apropiación de conocimientos para saber hacer, sino que abarca todo un conjunto de capacidades, que se desarrollan a través de procesos que conducen a la persona responsable a ser competente para realizar múltiples acciones (sociales, cognitivas, culturales, afectivas, laborales, productivas), por las cuales proyecta y evidencia su capacidad de resolver un problema dado, dentro de un contexto específico y cambiante.

La competencia, al igual que la inteligencia, no es una capacidad innata, sino que, por el contrario, es susceptible de ser desarrollada y construida a partir de las motivaciones internas de cada quien. La integración de estas dos áreas conforma la opción de vida, para el desarrollo cotidiano, el académico y el científico. Así, al fusionarlos, plantea la formación integral que abarca conocimientos (capacidad cognoscitiva), habilidades (capacidades sensorio-motrices), destrezas, actitudes y valores. En otras palabras: saber, saber hacer en la vida y para la vida, saber ser, saber emprender.

En la actualidad, el desempeño profesional eficiente en una sociedad globalizada y del conocimiento exige, además de las competencias específicas propias del ejercicio de una determinada profesión, competencias genéricas o transversales, que son comunes y necesarias para todo profesional.

Según el proyecto Tuning (Proyecto Tuning América Latina, 2004), las competencias genéricas son las compartidas, que puedan generarse en cualquier titulación y que son consideradas importantes por ciertos grupos sociales. Hay ciertas competencias, como la capacidad de aprender y actualizarse permanentemente, la capacidad de abstracción, análisis y síntesis, etc., que son comunes a todas o casi todas las titulaciones. En una sociedad en transformación, donde las demandas se están reformulando constantemente, estas competencias genéricas se vuelven muy importantes. Siguiendo la línea del proyecto mencionado, las competencias específicas están vinculadas con una disciplina y son las que confieren identidad y consistencia a un programa específico.

### 2.1.2 SISTEMAS DE INFORMACIÓN

Un SI<sup>5</sup> puede ser cualquier combinación organizada de personas, hardware, software, redes de comunicación y recursos de información que almacene, recupere, transforme y disemine información en una organización. (O'BRIEN, 2006)

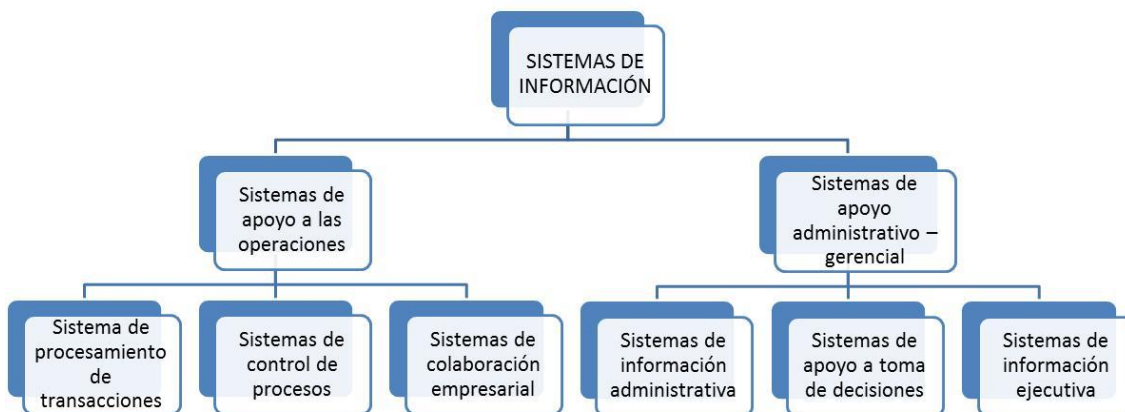


Figura 1, Clasificación operativa y administrativa de los sistemas de información (O'Brien & Marakas, 2006)

---

<sup>5</sup> Sistema de Información

### **2.1.2.1 Sistemas Transaccionales**

Es un tipo de sistema de información diseñado para recolectar, almacenar, modificar y recuperar todo tipo de información que es generada por las transacciones en una organización. Una transacción es un evento o proceso que genera o modifica la información que se encuentran eventualmente almacenada en un sistema de información.

Estos sistemas suelen usarse para proveer de información a las empresas como son operaciones básicas, facturación, consulta de clientes, etc.

(CORTÉS MORALES & CORTÉS MORALES, 1998)

### **2.1.3 PSP Personal Software Process**

(POMEROY-HUFF, 2009) Afirman que el PSP es un conjunto ordenado de procesos definidos que orientan a los ingenieros de software y a programadores a medir, evaluar y monitorear la manera de hacer sus tareas.

Los principales objetivos del PSP son:

1. Mejorar las estimaciones
2. Mejorar la planeación y acompañamiento de cronogramas
3. Proteger contra el exceso de compromisos
4. Crear un compromiso personal con la calidad
5. Compromiso del desarrollador en la mejora continua del proceso de desarrollo.
6. Aumento de la calidad a través de la reducción de la incidencia de errores
7. Mayor precisión en las estimaciones de tamaño del software y tiempo de desarrollo.

Los ingenieros de software normalmente desarrollan productos a partir de sus propios métodos y técnicas o a partir de los ejemplos de los ingenieros más experimentados. PSP ofrece una forma de mejorar la calidad, la predicción y la productividad del trabajo. A medida que el trabajo mejora, también tiende a mejorar la calidad de los proyectos.

#### **2.1.3.1 Estrategia de PSP**

El modelo PSP está dividido en niveles, implantados de manera incremental. Los niveles superiores adicionan características a los niveles ya implantados lo que minimiza el

impacto de los cambios en los hábitos del desarrollador. Este deberá tan sólo adaptar nuevas técnicas a las ya existentes y conocidas.

Lo más importante en el proceso de aprendizaje son los datos recogidos después de cada fase, pues con base en los resultados obtenidos en la fase actual se propone mejorar el desempeño personal para la siguiente fase.

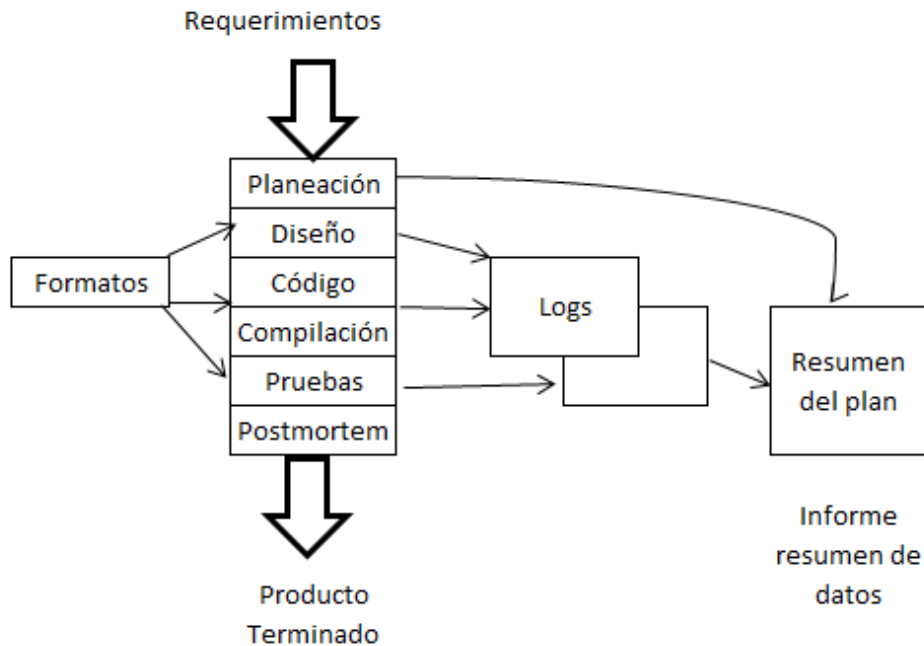


Figura 2 Proceso de PSP (Humphrey, PSP(sm): A self-Improvement Process for Software Engineers, 1995)

Los siguientes son los niveles de mejoramiento PSP:

NIVEL	NOMBRE	ACTIVIDADES
<b>PSP0</b>	Medición Personal	<ul style="list-style-type: none"> <li>• Registro de tiempo</li> <li>• Registro de defectos</li> </ul>
<b>PSP0.1</b>	Registro de defectos	<ul style="list-style-type: none"> <li>• Patrón de tipos de defectos.</li> <li>• Patrón de codificación.</li> <li>• Medida de tamaño.</li> <li>• Propuesta de mejoramiento de proceso.</li> </ul>
<b>PSP1</b> <b>PSP1.1</b>	Planeación personal	<ul style="list-style-type: none"> <li>• Estimación de tamaño.</li> <li>• Informe de pruebas.</li> <li>• Planeamiento de tareas.</li> </ul>

		<ul style="list-style-type: none"> <li>• Cronogramas.</li> </ul>
<b>PSP2</b> <b>PSP2.1</b>	Gerenciamiento de la calidad personal	<ul style="list-style-type: none"> <li>• Revisiones de código.</li> <li>• Revisiones de proyecto.</li> <li>• Patrones del Proyecto.</li> </ul>
<b>PSP3</b>	Proceso personal cíclico	<ul style="list-style-type: none"> <li>• Desarrollo cíclico</li> </ul>

Tabla 1 Niveles de mejoramiento PSP (Humphrey, Introduction to the team software process, 2000)

Se ha escogido PSP0.1 porque brinda una introducción para desarrollar la disciplina necesaria para registrar tiempos y obtener medidas frente a las líneas de código y defectos, además sirven para crear una línea base o de referencia para futuros proyectos.

#### 2.1.4 Proceso Unificado UP

De acuerdo a Booch, G, et al, (2000) es un conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema software, sirve como marco genérico que puede especializarse para una variedad de tipos de sistemas, diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y diferentes tamaños de proyectos. Se compone de:

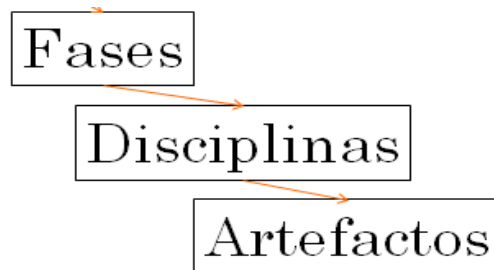


Figura 3 - Elementos de UP

Se caracteriza por:

- UP se basa en componentes interconectados a través de interfaces, está dirigido por casos de uso, centrado en la arquitectura y es iterativo e incremental. Es práctico dividir el esfuerzo de desarrollo de un proyecto de software en partes más pequeñas o mini proyectos, cada uno de ellos es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos a crecimientos en el producto. Las iteraciones deben estar

controladas. Esto significa que deben seleccionarse y ejecutarse de una forma planificada.

- Posee un desarrollo dirigido por el riesgo: Se identifican y definen estrategias para enfrentar los riesgos, resolviendo los puntos más difíciles, elaborando planes de contingencia y tratando de anticipar las dificultades.
- Basado en el análisis de requisitos, casos de uso, crea un esquema en borrador de la arquitectura comenzando por la parte no específica de los casos de uso (por ejemplo la plataforma) pero con una comprensión general de los casos de uso fundamentales. Cada caso de uso es especificado en detalle y realizado en términos de subsistemas, clases, y componentes. A medida que los casos de uso se especifican y maduran, se descubre más de la arquitectura, y esto a su vez lleva a la maduración de más casos de uso.

Para el proceso unificado es práctico dividir el trabajo en partes más pequeñas o miniproyectos, cada uno de ellos es una iteración que resulta en un incremento y estos a su vez al crecimiento del producto. Para una mayor efectividad, estas iteraciones deben estar controladas; lo que significa que deben ejecutarse y planearse de forma rigurosa.

El proceso unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema, cada ciclo concluye con una versión del producto.

Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición, cada una de estas a su vez se divide en iteraciones que a su vez se pueden descomponer en iteraciones más pequeñas.

Cada fase termina con un hito, y cada uno de éstos se determina por la disponibilidad de un conjunto de artefactos.

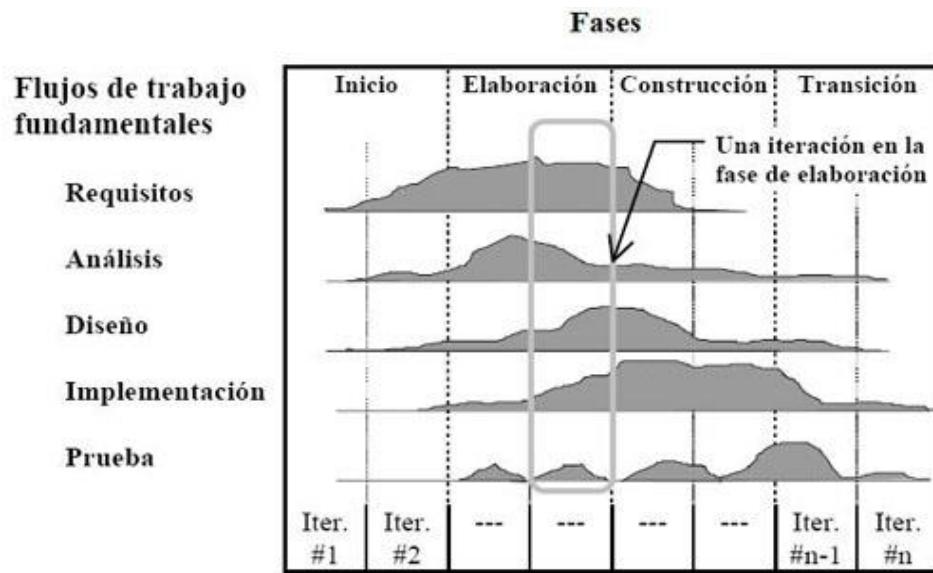


Figura 4 - Cinco flujos de trabajo: requisitos, análisis, diseño, implementación y prueba, cada una de ellas tienen lugar sobre las cuatro fases: inicio, elaboración, construcción y transición. (BOOCH, 2000)

## 2.2 DISEÑO METODOLÓGICO

La primera etapa es la construcción de la propuesta siguiendo los lineamientos de la maestría, a continuación se describen los pasos a seguir en el desarrollo del proyecto

### 2.2.1 Metodología

Siguiendo el esquema del Proceso Unificado se diseñó el proceso a través de iteraciones, como se describe a continuación:

**Inicio, Primera iteración:** fases de inicio, en la cual se realiza la planeación del proyecto, definición de metodología a seguir, se esboza un esquema general del sistema, se planea la cantidad de iteraciones posibles, búsqueda bibliográfica y de referencias, la identificación de conocimientos iniciales, productos a entregar, y definición de formatos de PSP a diligenciar.

También hace parte de esta etapa identificar el software en el cual se desarrollará la aplicación y algunos temas adicionales relacionados para iniciar el aprendizaje y contextualización de los mismos.

Como parte fundamental se realiza una revisión de los temas referentes al desarrollo de software para utilizar las herramientas pertinentes en el análisis y diseño.

Ya que UP es un método iterativo de desarrollo que ayuda a construir un programa de gran tamaño utilizando una secuencia de pasos más pequeños incrementales, se recopilan datos para aplicar un guion de PSP0.1 por cada uno de los ciclos definidos en la fase de inicio, al interior de cada ciclo se aplican plantillas de PSP0.1 para toma de tiempos de cada sesión de trabajo y registro de problemas encontrados, de esta manera se puede determinar el esfuerzo, tiempo invertido y perfil de defectos.

También se definen requerimientos funcionales y no funcionales del sistema, así como los casos de uso de alto nivel.

Al terminar esta iteración se realiza el diligenciamiento del formato resumen de PSP0.1 con los datos de tiempos invertidos en las fases correspondientes.

**Elaboración, Segunda iteración:** se complementa la planeación, análisis y definición de requerimientos inicial del sistema y definición de indicadores del proyecto que permitirán su evaluación permanente.

Se definen los detalles de los primeros casos de uso de la iteración anterior y se realizan pruebas de instalación y configuración de las herramientas de desarrollo del proyecto (software de servidor web, motor de bases de datos y librerías de terceros, software de versionamiento, software para seguimiento del proceso de desarrollo).

Se implementan los casos de uso ya especificados, también se realiza registro de tiempos y registro de defectos de PSP en cada sesión de trabajo.



Al terminar esta iteración se realiza el diligenciamiento del formato resumen de PSP0.1 con los datos de tiempos invertidos en las fases correspondientes.

**Construcción, Tercera iteración:** Se detallan e implementan los demás casos de uso y se realizan pruebas de cada uno de ellos por separado, a medida que se desarrollan se integran para garantizar su funcionamiento en conjunto mediante pruebas de integración.

Se utiliza el Framework Yii del lenguaje PHP para facilitar el desarrollo, implementación y posterior integración con el sistema de la UAM®.

Se realiza registro de tiempos y registro de defectos de PSP.

Al terminar esta iteración se realiza el diligenciamiento del formato resumen de PSP0.1 con los datos de tiempos invertidos en las fases correspondientes.

Al terminar cada iteración se utilizan los artefactos generados como insumos de la siguiente etapa y se realiza la planeación de la misma.

Se elabora el manual técnico y de usuario, en los cuales se describe la instalación de los programas necesarios para el funcionamiento del sistema, así como cada característica funcional del sistema.

### 2.3 PRUEBAS

Al final de cada iteración se realizan pruebas unitarias, también pruebas con el usuario y pruebas de integración con cada uno de los casos de uso implementados.

### 2.4 PRESUPUESTO

El desarrollo del proyecto se realizará con recursos propios del estudiante y se contará con la asesoría del director asignado por la maestría. Ver anexo A.

### 2.5 CRONOGRAMA

Actividad	Fecha inicio	Cronograma
Elaboración de propuesta		junio – octubre 2012

<b>Fase inicial:</b> 1. Primera iteración: Revisión de literatura y actualización del estado del arte, proceso unificado, PSP. Definición de plantillas que se van a utilizar.	Etapa 1	diciembre 2012 – marzo 2013
2. Revisión de sistemas de gestión existentes.		
3. Segunda iteración: Definición de requerimientos y casos de uso, cargar datos base al sistema, creación del módulo de Administración de competencias, pruebas unitarias y con el usuario.	Etapa 2	marzo 2014 – junio 2014
<b>Fase de Desarrollo:</b> 4. Tercera iteración: Creación del módulo Valorar competencias, pruebas unitarias y con el usuario, pruebas de integración.	Etapa 3	junio 2014 – febrero 2015
5. Elaboración de informe	Etapa 1-6	diciembre de 2012
6. Revisiones		febrero 2015
7. Entrega final	Etapa 6	Junio 2015

Tabla 2 – Cronograma

La tabla anterior muestra el cronograma dividido en etapas para facilitar su ejecución.

### **3 DESARROLLO**

En el siguiente capítulo se encuentra el proceso seguido para la elaboración del proyecto, también se ubica al lector mediante el referente teórico, el cual incluye los conceptos básicos para el entendimiento del proyecto.

#### **3.1 DESARROLLO DEL PROYECTO**

En este apartado se muestra la ejecución de lo planteado en la metodología y lo planeado en el cronograma y se plantea según las etapas de desarrollo de UP.

Se considera que tres es una cantidad de iteraciones adecuadas para la complejidad del proyecto y el tiempo disponible para su desarrollo.

Cada una de las iteraciones corresponde a una fase definida por la metodología UP: Inicio, elaboración y construcción, la fase de transición no se tiene en cuenta ya que la implantación del sistema está fuera del alcance de este proyecto.

##### **3.1.1 Primera iteración, fase de inicio:**

###### **3.1.1.1 Actividades de análisis**

Se ha realizado la revisión de la literatura y actualización del estado del arte, al empezar el proyecto se necesita efectuar una revisión bibliográfica y consultas de las definiciones de competencias existentes y cuáles de estas definiciones se adecúan a los requisitos de la Universidad Autónoma.

Luego se ha llevado a cabo una revisión bibliográfica de los temas concernientes al desarrollo de software como son: Sistemas de información, gerenciales y transaccionales.

- Proceso unificado
- UML

- Personal Software Process, plantillas y formatos a utilizar.

También se especifican los diagramas UML a diseñar y se perfila la cantidad de iteraciones que tendrá el desarrollo con UP.

Para la recolección de información referente al sistema a desarrollar se ha realizado una reunión con la Coordinadora de Acreditación Institucional Adriana María Giraldo Osorio el 6 de marzo del 2012, en la cual se discute el escenario actual en el ámbito de competencias de la UAM®. Se definen la estructura del modelo el cual consiste en competencias divididas en subcompetencias, las cuales se asocian a asignaturas en particular asignándoles un nivel. Los niveles definidos son básico, intermedio y avanzado.

También se obtienen parte de los datos necesarios para determinar los requerimientos funcionales y no funcionales del sistema, además de los casos de uso de alto nivel, así como el tipo de PSP que se utiliza, el cual es PSP 0.1.

La medición de tiempos y cantidad de líneas de código servirán para sentar un precedente, estándares y proponer mejora para el desarrollo de futuros proyectos.

Mediante el Diagrama de origen de datos del sistema. que se muestra a continuación, se ayuda a determinar el origen de los datos que el sistema utilizará, como son: los datos de usuarios, programas, asignaturas, grupos y demás.

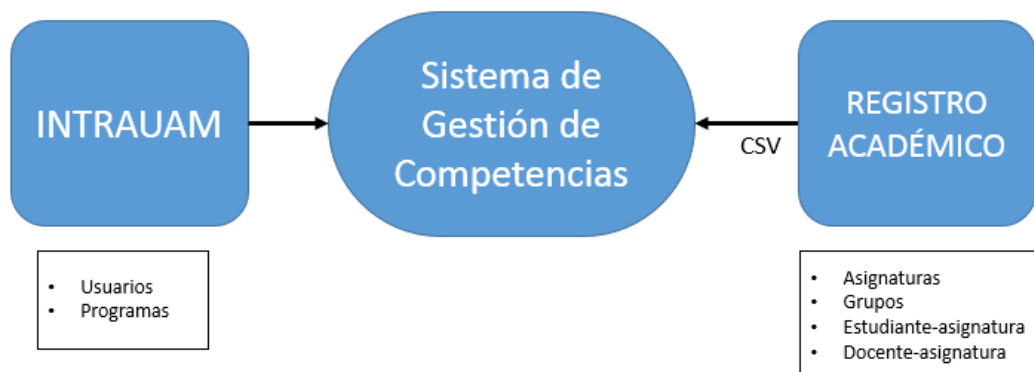


Figura 5 Diagrama de origen de datos del sistema.

### 3.1.1.1.1 Requerimientos No Funcionales

<b>ID</b>	RNF-01
<b>Tipo</b>	Portabilidad, estándares, implementación.
<b>Nombre</b>	Plataforma
<b>Descripción</b>	El sistema de gestión de competencias debe ser desarrollado en un entorno WEB tipo cliente/Servidor, para ser instalado en la Intranet de la UAM®.

<b>ID</b>	RNF-02
<b>Tipo</b>	Estándares, implementación.
<b>Nombre</b>	Entorno de Ejecución y librerías.
<b>Descripción</b>	El sistema de gestión de competencias debe ser desarrollado utilizando el lenguaje de programación PHP, el Framework de desarrollo Yii y el motor de base de datos MySQL, para facilitar su integración con el ecosistema tecnológico de la UAM®.

<b>ID</b>	RNF-03
<b>Nombre</b>	Entorno Visual
<b>Tipo</b>	Usabilidad, estándares.
<b>Descripción</b>	El sistema de gestión de competencias va a utilizar los colores institucionales de la Universidad Autónoma de Manizales o en su defecto de las otras aplicaciones institucionales para tener uniformidad en cuanto a la vista que se presenta a los usuarios. El sistema, va a utilizar una plantilla que permita tener en cada una de las páginas que el usuario visite, el mismo modelo, por ejemplo, un menú principal, y una sección de contenido siguiendo los lineamientos institucionales

<b>ID</b>	RNF-04
<b>Tipo</b>	Interoperabilidad
<b>Nombre</b>	Acoplamiento con Registro académico
<b>Descripción</b>	El sistema debe soportar la importación de información en formato plano (CSV) desde la base de datos de Registro Académico.

<b>ID</b>	RNF-05
<b>Tipo</b>	Interoperabilidad
<b>Nombre</b>	Acoplamiento con IntraUAM
<b>Descripción</b>	El sistema debe conectarse con la base de datos de IntraUAM para obtener información de los usuarios.

### **3.1.1.1.2 Requerimientos de dominios**

<b>ID</b>	RD-01
<b>Nombre</b>	Historial de valoraciones
<b>Descripción</b>	El sistema debe guardar registros de las valoraciones realizadas a los estudiantes para permitir que los mismos estudiantes o interesados puedan consultar el historial de competencias valoradas.

### 3.1.1.1.3 Diagrama de casos de uso de alto nivel

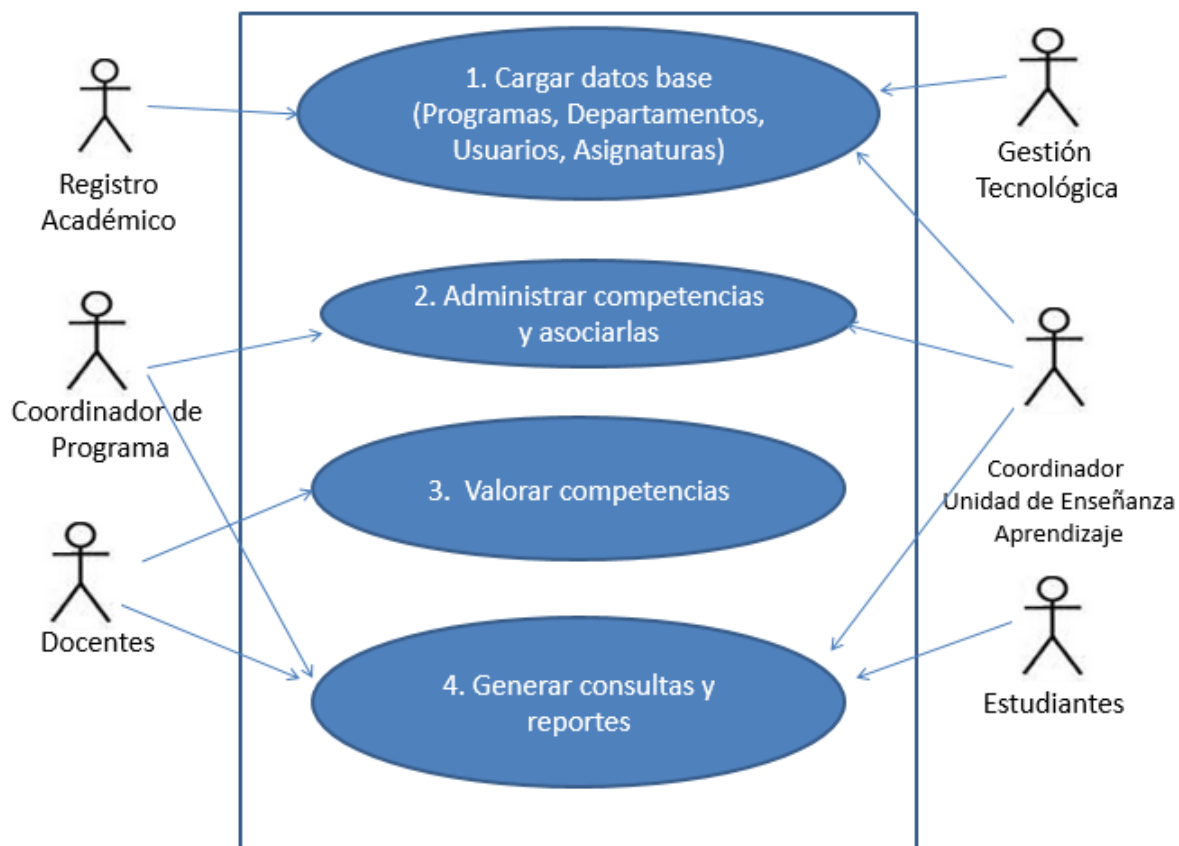


Figura 6 - Casos de uso de alto nivel

#### **Caso: 1. Cargar datos base.**

Tipo: Primario.

Actores: Registro académico, Gestión tecnológica, Coordinador de la Unidad de enseñanza aprendizaje.

Descripción: Este caso permite cargar la información de usuarios, programas y asignaturas al sistema.

Dicha información proviene del sistema Universitas XXI utilizado por Registro Académico y de Intrauam.

#### **Caso: 2. Administrar competencias y asociarlas.**

Tipo: Primario.

Actores: Coordinador de programa, Gestión tecnológica, Coordinador de la Unidad de enseñanza aprendizaje.

Descripción: Este caso permite ingresar las competencias, asociarlas a los diferentes programas y asignaturas y administrar sus niveles, permite que los datos ingresados sean parametrizables.

### **Caso: 3. Valorar competencias.**

Tipo: Primario.

Actores: Docente.

Descripción: Este caso permite al docente calificar las competencias de los estudiantes de forma cualitativa.

### **Caso: 4. Generar consultas y reportes.**

Tipo: Primario.

Actores: Coordinador de programa, Gestión tecnológica, Coordinador de la Unidad de enseñanza aprendizaje, docente, estudiantes.

Descripción: Este caso permite realizar consultar y reportes de las asignaturas, competencias asociadas y sus respectivas calificaciones.

#### **3.1.1.2 Actividades de diseño**

##### **3.1.1.2.1 Objetivos del diseño**

A continuación se plantean los objetivos que se busca que le diseño del sistema cumpla asociados a algunos de los requerimientos no funcionales.

- Diseñar la estructura del software utilizando una arquitectura web, lenguaje de programación PHP, Framework de desarrollo Yii y el motor de base de bases de datos MySQL (RNF-01, RNF-02).
- Emplear un sistema de plantillas estándar que permita la incorporación de los colores institucionales de la UAM® (RNF-03).
- Incluir una interfaz basada en el formato CSV para la carga de información desde el sistema de registro académico de la UAM® (RNF-04).



- Incluir en el diseño de la persistencia los elementos que se acceden en la base de datos de IntraUAM (RNF-05).

### **3.1.1.2.2 Arquitectura del sistema**

#### **3.1.1.2.2.1 Configuración del hardware**

Para el presente proyecto se utiliza MySQL 5.1.37 y PHP 5.3.0 instalados en el paquete XAMPP 1.7.2 para Windows en un equipo para el desarrollo que simule las condiciones de la infraestructura tecnológica de la UAM cuyos servidores utiliza las mismas herramientas en versiones similares.

#### **3.1.1.2.2.2 Definición de subsistemas**

Se propone dividir el sistema en dos subsistemas. Esta división se representa en un diagrama de componentes mostrado en la Figura 7 – Subsistemas:

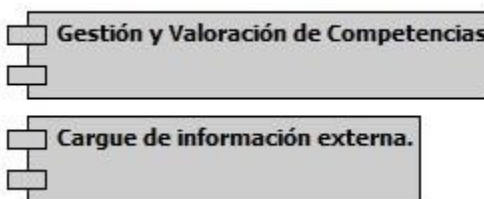


Figura 7 – Subsistemas

Estos subsistemas se definen así:

- **Cargue de información externa:** toma la información del sistema de registro académico de la UAM en archivos CSV.
- **Gestión y valoración de competencias:** permite el ingreso de las competencias, asociación a asignaturas y a niveles y asignación de juicio valorativo.

#### **3.1.1.2.2.3 Mapeo de los subsistemas a los nodos**

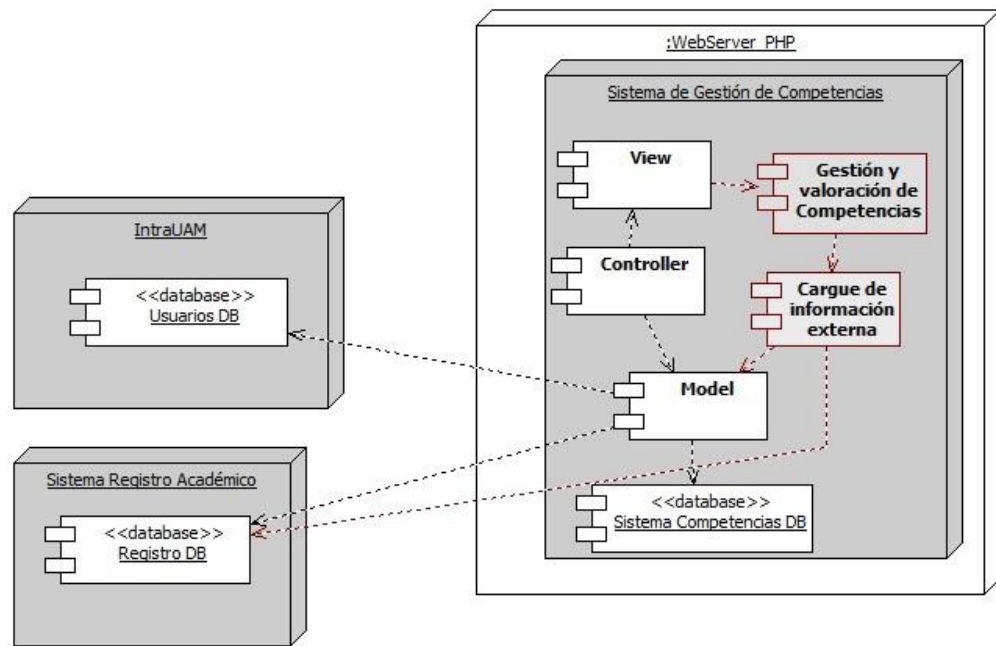


Figura 8 - Diagrama de componentes

Como se muestra en la Figura 8, el sistema de gestión de competencias utiliza la información que actualmente posee el sistema de registro académico y el sistema IntraUAM, por eso se ha elegido una plataforma compatible con estos dos sistemas, que facilite el proceso de integración y poder compartir dichos datos.

### 3.1.1.2.3 *Uso frameworks y librerías:*

Debido a que se propone utilizar la arquitectura MVC se elige el Framework de desarrollo Yii, que implementa este patrón y es ampliamente utilizado por los sistemas de información de la UAM.

Este framework permite crear aplicaciones web partiendo de tablas previamente creadas, utiliza plantillas lo que facilita la generación de código, además posee varias clases de librerías que implementan funciones comunes a todas las aplicaciones web basadas en

MVC<sup>6</sup> como los modelos, los controladores, sus acciones, vistas, autenticación, autorización, internacionalización, entre otros. Cuenta con documentación completa de su API<sup>7</sup>, un amplio conjunto extensiones aportadas por usuarios y una creciente comunidad de usuarios y desarrolladores.

#### 3.1.1.2.4 Diseño de la persistencia

Se realiza una primera versión de diagrama entidad relación utilizando la herramienta MySQL WorkBench. En la Figura 9 se muestran las tablas, las de nombre con fondo blanco simbolizan las que se encuentran definidas en IntraUAM, las demás son específicas de este sistema.

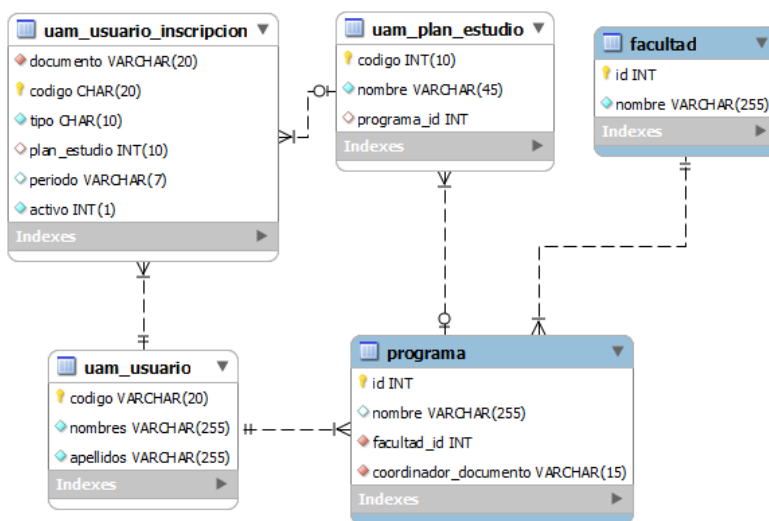


Figura 9 - Modelo entidad relación iteración 1

#### 3.1.1.3 Actividades de implementación

A partir de las entidades diseñadas y utilizando la herramienta de generación de código de Yii, se crean los modelos, controladores y vistas para la administración de las entidades de facultad, programas y nivel (CRUD).

<sup>6</sup> Modelo – vista – controlador.

<sup>7</sup> <http://www.yiiframework.com/doc/api/>



Figura 10 - CRUD administrar Facultades

### 3.1.1.4 Actividades de postmortem

Durante esta iteración se tomó el tiempo dedicado a cada actividad de desarrollo y se totalizaron, además se aplicó la plantilla para registrar el resumen del proyecto de PSP0.1, también se utilizó el Process Dashboard para el conteo de líneas de código.

Tiempos de desarrollo en la primera iteración

Fase	Tiempo	Porcentaje
<b>Requerimientos</b>	05:48:00	17%
<b>Análisis</b>	09:23:00	28%
<b>Diseño</b>	11:24:00	34%
<b>Implementación</b>	06:09:00	18%
<b>Pruebas</b>	0:43:00	2%
<b>Total Iteración 1</b>	33:27:00	100%
<b>Tiempo en Días</b>	1 días 9 horas	

Tabla 3 - Total de tiempos medidos según la fase.

Tipo de líneas	Cantidad
<b>Base:</b>	0 (1461)
<b>Borradas:</b>	0 (13)
<b>Modificadas:</b>	50
<b>Agregada:</b>	105
<b>Nuevas &amp; Cambiadas:</b>	155
<b>Total:</b>	155 (1553)

Tabla 4 - Cantidad de líneas de código para la primera iteración.

Durante esta etapa se establecen algunas conclusiones respecto a la iteración según los tiempos referenciados en la Tabla 3.

- Los interesados en el proyecto tenían claras las necesidades y por ello fue más fácil elicitar los requerimientos.
- El tiempo empleado en la etapa de requerimientos fue relativamente corto en comparación con el común de la industria.
- La etapa de diseño tomó tiempo ya que se experimentaron con herramientas diferentes a las utilizadas en el proyecto como fueron Ruby on Rails.
- A pesar de la cantidad de líneas totales de código el tiempo de implementación fue corto gracias al uso de la herramienta de generación de código de Yii.
- La implementación de las funciones de importación de CSV se posponen para una iteración posterior debido a que se requiere mayor conocimiento del lenguaje de programación y del Framework que en esta iteración no se ha alcanzado.

En la Tabla 4 se aprecian las líneas de código contadas utilizando el Process Dashboard para la primera iteración. Dicho programa ha tomado como líneas base las que el previamente se habían generado utilizando Yii Framework (cuyo valor se muestra entre paréntesis), durante la creación de los CRUD de facultad, programa, nivel y asignatura. Sin embargo, hay que considerar que la cantidad de líneas base es cero, ya que no corresponden a código elaborado por el programador de manera previa.

En ese orden de ideas la cantidad de líneas borradas debe establecerse también en 0 ya que las mismas fueron generadas por la herramienta.

Se dejan en el registro los valores contados por el software (entre paréntesis) y también los reales, ya que esto permitirá estimar el tamaño derivado del trabajo del programador y distinguirlo de la herramienta de generación de código.

### 3.1.2 Segunda iteración, fase de elaboración.

Una vez terminada la fase inicial se continúa con el desarrollo del sistema frente a un segundo grupo de requerimientos, además, con los insumos de la iteración anterior se empiezan a detallar y a implementar los casos de uso 2 “Administrar Competencias y Asociarlas”, se realiza registro de tiempos y defectos con los formatos de PSP para su posterior análisis al final de la iteración.

#### 3.1.2.1 Actividades del análisis

##### 3.1.2.1.1 Casos De Uso Detallados:

- 2. Administrar competencias y asociarlas
  - 2.1 Administrar competencias.
  - 2.2 Asociar competencia a programa.
  - 2.3 Asociar competencia a asignatura.
  - 2.4 Administrar niveles.

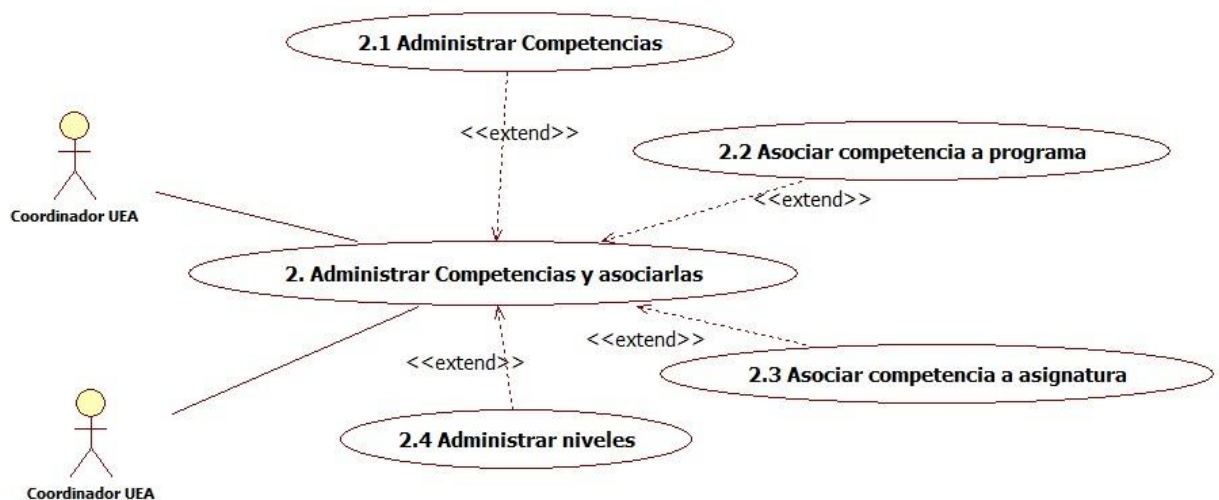


Figura 11 - Diagrama del caso de uso 2, Administrar competencias.

## Caso: 2.1. Administrar competencias

Tipo: Primario

Actores: Coordinador de programa, Coordinador de la Unidad de enseñanza aprendizaje.

Descripción: Este caso permite ingresar las competencias, editarlas y desactivarlas del sistema

Curso normal de eventos:

ACTOR	SISTEMA
1. El usuario ingresa a la opción Administrar competencia.	2. El sistema muestra las opciones: - Registrar Competencia - Actualizar competencia - Inactivar competencia
3. El usuario elige "Registrar Competencia".	4. Se muestran los campos para ingresar el nombre y la descripción de la competencia.
5. Digita el nombre de la competencia.	6. Verifica que el nombre de la competencia no se encuentre registrado ya en el sistema y se permite la digitación de la descripción de la competencia (opcional).
7. Digita la descripción de la competencia y Se envía la información para almacenar en el sistema.	8. Se almacena la información de la competencia en el sistema y se reporta el éxito de la operación al usuario.

Excepciones:

Paso 6: Si el nombre de la competencia ya existe, genera un mensaje "nombre existe en el sistema" y regresa al paso 4.

Paso 9: Si no se envía la información al sistema termina el caso de uso.

Curso alterno:

ACTOR	SISTEMA
3. El usuario elije "Actualizar Competencia"	
4. Busca la competencia de acuerdo a una palabra clave.	5. Lista las coincidencias.
6. Selecciona cual desea actualizar y presiona el botón correspondiente.	7. Se muestran los campos para ingresar el nombre y la descripción de la competencia.
8. Actualiza la información requerida	9. Se almacena la información de la competencia en el sistema y se reporta el éxito de la operación al usuario.

Excepciones:

Paso 4. Si la palabra clave no registra coincidencias, regresa al paso 3.

Paso 10. Si no se envía la información al sistema termina el caso de uso.

Curso alterno:

ACTOR	SISTEMA
3. El usuario elije "Inactivar Competencia".	
4. Busca la competencia de acuerdo a una palabra clave.	5. Lista las coincidencias.
6. Selecciona cual desea inactivar y presiona el botón correspondiente.	7. solicita confirmación del usuario.
8. El usuario confirma.	9. Si se confirma, se cambia el estado de la competencia a inactivo, se registra la fecha de inactivación y se guarda en el sistema.

Excepciones:

Paso 4. Si la palabra clave no registra coincidencias, regresa al paso 3.

Paso 8. Si no se confirma, termina el caso de uso y regresa al caso anterior.

Paso 10. Si no se envía la información al sistema termina el caso de uso.



## Caso: 2.2. Asociar competencia a programa

Tipo: Primario

Actores: Coordinador de programa, Coordinador de la Unidad de enseñanza aprendizaje.

Descripción: Este caso permite asociar las competencias a los programas o quitar competencias a un programa

Curso normal de eventos:

ACTOR	SISTEMA
0. Se requiere asociar o desvincular una competencia a un programa académico de la universidad.	1. Se pregunta si se quiere “asociar competencia” o “Desvincular competencia” a un programa.
2. Se selecciona asociar	3. El sistema muestra los programas a los cuales se les puede asociar las competencias.
4. Selecciona el programa al cual se va a asociar.	5. Se solicita la competencia a asociar
5. Busca la competencia de acuerdo a una palabra clave.	6. Lista las coincidencias.
7. Se selecciona	8. Se guarda la asociación en el sistema.

NOTA: Si se desea se pueden seguir asociando competencias (inicia paso 4)

Excepciones:

Paso 5. Si la palabra clave no registra coincidencias, regresa al paso 3.

Curso alternativo:

ACTOR	SISTEMA
2. Se selecciona “Desvincular competencia”	3. El sistema muestra los programas a los cuales se les puede desvincular las competencias.
4. Selecciona el programa.	5. Lista las competencias asociadas a dicho programa.
6. Busca la competencia de acuerdo a una palabra clave.	7. Lista las coincidencias.
8. selecciona la opción desvincular.	9. Se guardan los cambios en el sistema.

### Caso: 2.3. Asociar competencias a asignaturas

Tipo: Primario

Actores: Coordinador de programa, Coordinador de la Unidad de enseñanza aprendizaje.

Descripción: Este caso permite asociar o desvincular las competencias a las asignaturas de cada programa.

Curso normal de eventos:

ACTOR	SISTEMA
0. Se requiere asociar o desvincular una competencia a una asignatura.	1. Pregunta si se quiere “asociar competencia a asignatura” o “Desvincular competencia a una asignatura”.
2. Se selecciona “asociar competencia a asignatura”	3. El sistema muestra los programas.
4. Selecciona el programa al cual pertenece la asignatura.	5. El sistema muestra las asignaturas del programa seleccionado.

6. Busca las asignaturas de acuerdo a una palabra clave o código.	7. Lista las coincidencias.
8. Se selecciona la asignatura.	9. Lista las competencias asociadas al programa.
10. Selecciona una competencia para asociar a la asignatura.	11. Lista niveles.
12. selecciona nivel de la competencia seleccionada.	13. Se guarda la asociación en el sistema.

NOTA: Si se desea se pueden seguir asociando competencias a las asignaturas (inicia paso 10)

Excepciones:

Paso 6. Si la palabra clave o código no registra coincidencias, regresa al paso 5.

Curso alterno:

ACTOR	SISTEMA
2. Se selecciona "Desvincular competencia a una asignatura".	3. El sistema muestra los programas.
4. Selecciona el programa al cual pertenece la asignatura.	5. El sistema muestra las asignaturas del programa seleccionado.
6. Busca las asignaturas de acuerdo a una palabra clave o código.	7. Lista las coincidencias.
8. Se selecciona la asignatura.	9. Lista las competencias asociadas a la asignatura.
10. selecciona la opción desvincular.	11. Se guardan los cambios en el sistema.

NOTA: Si se desea se pueden seguir desvinculando competencias a las asignaturas (inicia paso 9)

Excepciones:

Paso 6. Si la palabra clave o código no registra coincidencias, regresa al paso 5.

### **Caso: 2.4. Administrar niveles**

Tipo: Primario

Actores: Coordinador de programa, Coordinador de la Unidad de enseñanza aprendizaje.

Descripción: Este caso permite crear los niveles que tendrán las competencias.

Curso normal de eventos:

ACTOR	SISTEMA
1. El usuario ingresa a la opción "Administrar niveles de competencia".	2. El sistema muestra las opciones: - Registrar Nivel de Competencia. - Actualizar Nivel de Competencia. - Inactivar Nivel de Competencia.
3. El usuario elige "Registrar Nivel de Competencia".	4. Se muestran los campos para ingresar el nombre y la descripción del nivel.
5. Digita el nombre del nivel.	6. Verifica que el nombre del nivel no se encuentre registrado ya en el sistema y se permite la digitación de la descripción del nivel (opcional).

7. Digita la descripción del nivel y se envía la información para almacenar en el sistema.	8. Se almacena la información del nivel en el sistema y se reporta el éxito de la operación al usuario.
--	---

Excepciones:

Paso 6: Si el nombre del nivel de competencia ya existe, genera un mensaje “nombre existe en el sistema” y regresa al paso 4.

Paso 9: Si no se envía la información al sistema termina el caso de uso.

Curso alternativo:

ACTOR	SISTEMA
3. El usuario elige “Actualizar Nivel de Competencia”	4. Lista los niveles de competencias.
5. Selecciona cual desea actualizar y presiona el botón correspondiente.	6. Se muestran los campos para ingresar el nombre y la descripción del nivel de competencia.
7. Actualiza la información requerida	8. Se almacena la información en el sistema y se reporta el éxito de la operación al usuario.

Excepciones:

Paso 9. Si no se envía la información al sistema termina el caso de uso.

Curso alternativo:

ACTOR	SISTEMA
3. El usuario elige "Inactivar Nivel de Competencia"	4. Lista los niveles de competencias.
5. Selecciona cual desea inactivar y presiona el botón correspondiente.	6. solicita confirmación del usuario.
7. El usuario confirma.	8. Si se confirma, se cambia el estado del nivel a inactivo, se registra la fecha de inactivación y se guarda en el sistema.

Excepciones:

Paso 9. Si no se envía la información al sistema termina el caso de uso.

### Caso: 3. Valorar competencias.

Tipo: Primario.

Actores: Docente.

Descripción: Este caso permite al docente calificar las competencias de los estudiantes.

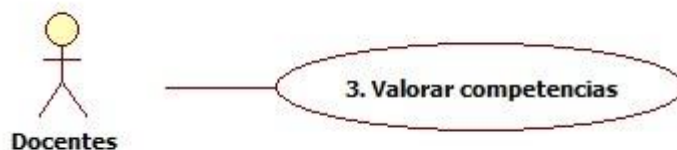


Figura 12 - Diagrama del caso de uso 3, Valorar competencias.

Curso normal de eventos:

ACTOR	SISTEMA
-------	---------

1. El docente ingresa a la opción Valorar competencia.	2. El sistema muestra los cursos asociados al docente.
3. El docente elije el curso a valorar.	4. Se muestra el primer estudiante inscrito con las competencias asociadas.
5. Digita juicio de valor para cada competencia.	
6. Elige la opción de “guardar”	7. Se almacena en el sistema la información de la valoración y se reporta el éxito de la operación al usuario.
8. Da clic en el botón de “siguiente estudiante”.	9. Se muestra el siguiente estudiante con sus competencias a validar.
10. el usuario valora y guarda las competencias del último estudiante.	11. El sistema reporta la calificación del grupo completo al usuario.

Excepciones:

Paso 5: el usuario deja vacío alguno de los juicios de valor de las competencias asociadas a los estudiantes, el sistema mostrará un aviso de error y no permitirá continuar con el siguiente estudiante.

Paso 12: Si no se envía la información al sistema termina el caso de uso.

### 3.1.2.2 Actividades del diseño

#### Diseño de la persistencia

Se realiza una nueva versión del diagrama entidad relación que refleja la integración del sistema de gestión de competencias con el actual sistema de registro académico de la UAM

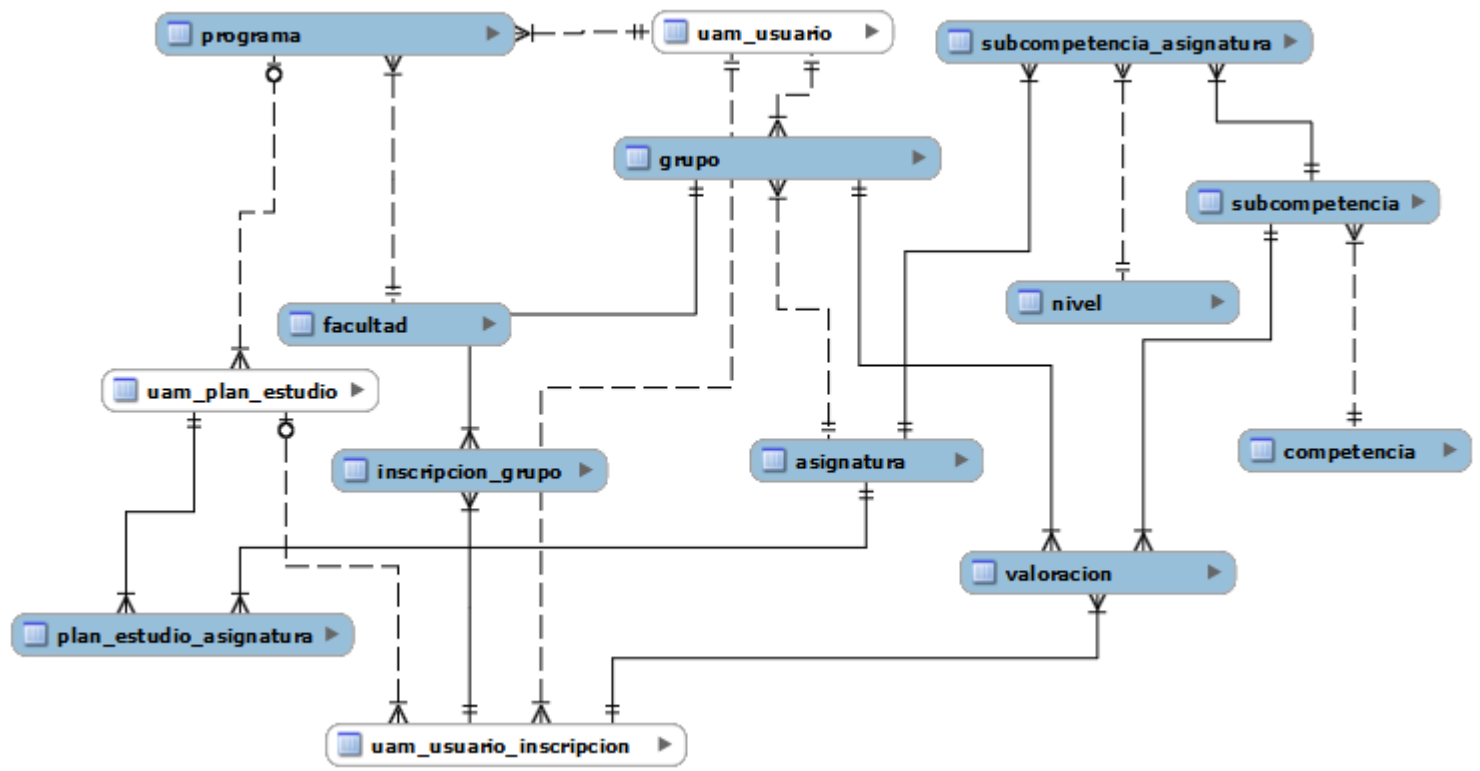


Figura 13 - Diagrama entidad relación



En el diagrama anterior se aprecia de color blanco las entidades que pertenecen al sistema de Intrauam<sup>8</sup>.

<b>Tabla</b>	<b>Descripción</b>
uam_usuario	registra todos los docentes, estudiantes, personal administrativo u otros que hagan uso del sistema de información de Intrauam
uam_plan_estudio	almacena la información de los planes de estudio a los que pueden estar vinculados los estudiantes
uam_usuario_inscripcion	Vincula un estudiante a un plan de estudios especificando el código de su expediente. (Utilizado extensamente en el sistema de registro académico).

Las demás entidades son las propias del Sistema de Gestión de Competencias.

<b>Tabla</b>	<b>Descripción</b>
facultad	Unidad académica de la universidad que agrupa los programas de pregrado y postgrado
programa	programa curricular de pregrado y postgrado que agrupa la universidad

---

<sup>8</sup> Sistema de información creado, soportado y utilizado por la UAM para ofrecer a sus empleados aplicaciones para desarrollar algunas de sus funciones, a los estudiantes y docentes aplicaciones con información académica, generar listas consultar horarios, etc.

plan_estudio_asignatura	asocia un programa con sus respectivos planes de estudio (un programa puede tener varios planes de estudio)
asignatura	lista los nombres y códigos de todas las asignaturas ofrecidas por la universidad
grupo	se registran el listado de grupos existentes por cada asignatura en cada periodo académico
inscripcion_grupo	identifica el grupo de una asignatura al que pertenece un estudiante en un periodo específico
competencia	lista todas las competencias definidas en el sistema
subcompetencia	almacena las subcompetencias relacionadas con cada competencia
subcompetencia_asignatura	se guardan las asociaciones entre las subcompetencias y asignaturas
nivel	clasificación de las subcompetencias en básico, intermedio, avanzado
valoración	guarda el juicio de valor de una subcompetencia para un estudiante en un grupo específico de una asignatura

### Diseño de interfaz

Se crea una interfaz a partir de la plantilla incorporada de la versión 1.1 de Yii framework, conservando una barra de menú en la parte superior con las opciones generales del sistema como se muestra en la Figura 14 Menú del sistema.



Figura 14 Menú del sistema

### 3.1.2.3 Actividades de implementación

Utilizando las herramientas de generación de código de Yii se crearon los modelos, controladores y vistas para la administración de competencias y subcompetencias.

Partiendo del código anterior se realizaron las modificaciones necesarias para que el programa permitiera asociar las competencias con sus subcompetencias. De manera similar se adicionó código para realizar la asociación entre subcompetencias, niveles y asignaturas.



Figura 15 - Asociar subcompetencias con asignaturas y niveles

### 3.1.2.4 Actividades de postmortem

Para el postmortem de esta iteración se aplicaron las mismas herramientas para las mediciones utilizadas en la anterior iteración.

Fase	Tiempo	Porcentaje
<b>Requerimientos</b>	06:41:00	6%
<b>Análisis</b>	36:35:00	33%
<b>Diseño</b>	5:09:00	5%
<b>Implementación</b>	59:40:00	55%
<b>Pruebas</b>	01:09:00	1%
<b>Total Iteración 1</b>	109:14:00	100%
<b>Tiempo en Días</b>	4 días 13 horas	

Tabla 5 - Total de tiempos medidos según la fase

En la tabla anterior se observa que la mayor parte del tiempo del desarrollo del sistema para la segunda iteración, se invierte en análisis e implementación. El resto del tiempo se divide entre diseño y requerimientos, las pruebas realizadas fueron para verificar el funcionamiento de los módulos de crear competencias y asociar subcompetencias de forma independiente.

Tipo de líneas	Cantidad
<b>Base:</b>	155 (3376)
<b>Borradas:</b>	12
<b>Modificadas:</b>	221
<b>Agregada:</b>	112
<b>Nuevas &amp; Cambiadas:</b>	333
<b>Total:</b>	255 (3476)

Tabla 6 - Cantidad de líneas de código para la segunda iteración

En la Tabla 6 se muestra el conteo de líneas de código para la segunda iteración. Entre paréntesis se muestran las cantidades con líneas de código generadas, es decir, aquellas que no fueron escritas por el programador, sino producto del Framework.

Durante esta etapa se establecen algunas conclusiones respecto a la iteración:

- A diferencia de la primera iteración, en esta se invierten menos tiempo en la elicitación de requerimientos y en el diseño del sistema y más en la implementación y análisis de otros casos de uso.
- Las líneas de código modificadas exceden a las agregadas debido a la utilización de código generado por el Framework Yii.
- Durante esta iteración algunos aspectos que se clasificaron inicialmente como defectos fueron descartados debido a que después de un análisis más detallado se determinó que no correspondían a un defecto del software desarrollado sino a condiciones o limitaciones de las librerías utilizadas.
- Las líneas de código que se muestran en la tabla anterior son las generadas en esta iteración, es decir, no incluyen a las presentadas en la primera iteración.
- Del tiempo dedicado al análisis gran parte corresponde a la realización de la documentación del proceso de desarrollo, como lo es la especificación de requerimientos, los diagramas de casos de uso y su especificación.
- En esta iteración se han implementado los casos de uso de Administrar Competencias y Asociarlas.

### **3.1.3 Tercera iteración, etapa de construcción**

En esta fase se ha detallado e implementado en el sistema el caso de uso Generar consultas y reportes.

Así como en las iteraciones anteriores se han tomado los tiempos correspondientes y se han diligenciado las plantillas de registro, defectos y de resumen de PSP0.1.

#### **3.1.3.1 Actividades de análisis**

##### **Caso: 4. Generar consultas y reportes.**

Tipo: Primario.

Actores: Coordinador de programa, Gestión tecnológica, Coordinador de la Unidad de enseñanza aprendizaje, docente, estudiantes.

Descripción: Este caso permite realizar consultar y reportes de las asignaturas, competencias asociadas y sus respectivas calificaciones.



Figura 16 - Diagrama del caso de uso 4, Generar consultas y reportes.

**Caso: 4.1. Consultar competencias por programa**

Tipo: Primario

Actores: Coordinador de programa, Coordinador de la Unidad de enseñanza aprendizaje, Docente.

Descripción: Este caso permite consultar las competencias y subcompetencias asociadas a cada programa de la universidad.

Curso normal de eventos:

ACTOR	SISTEMA
1. El usuario ingresa a la opción Consultar competencia por programa.	2. El sistema muestra los diferentes programas de la universidad.

3. El usuario selecciona el programa al cual desea consultar las competencias.	4. El sistema muestra las competencias asociadas al programa seleccionado.
5. El usuario selecciona una competencia.	6. el sistema muestra las subcompetencias asociadas a la competencia seleccionada.

### Caso: 4.2. Consultar Niveles

Tipo: Primario

Actores: Coordinador de programa, Coordinador de la Unidad de enseñanza aprendizaje, Docente.

Descripción: Este caso permite consultar los niveles de las subcompetencias designados por la universidad.

Curso normal de eventos:

ACTOR	SISTEMA
1. El usuario ingresa a la opción Consultar niveles de competencias.	2. El sistema muestra los niveles: básico, intermedio, avanzado.

### Caso: 4.3. Consultar competencias

Tipo: Primario

Actores: Coordinador de programa, Coordinador de la Unidad de enseñanza aprendizaje, Docente.

Descripción: Este caso permite consultar las competencias y subcompetencias en general

Curso normal de eventos:

ACTOR	SISTEMA
-------	---------

1. El usuario ingresa a la opción Consultar competencias.	2. El sistema muestra las opciones: Consultar Competencia por nombre. Consultar Competencia por asignatura. Consultar Competencia por docente. Consultar Competencia por estudiante.
3. El usuario selecciona "Consultar Competencia por nombre" y digita el nombre de la competencia.	4. El sistema muestra las competencias que cumplen ese criterio.
5. El usuario elige de cual competencia desea ver las subcompetencias.	6. El sistema muestra las subcompetencias asociadas a dicha competencia.

Excepciones:

Paso 4: el sistema no encuentra las competencias que cumple con el criterio y le informa al usuario.

Curso alternativo:

Actores: Coordinador de programa, Coordinador de la Unidad de enseñanza aprendizaje, Docente, Estudiante.

ACTOR	SISTEMA
1. El usuario selecciona "Consultar Competencia por asignatura"	2. El sistema solicita ingresar el código o nombre de la asignatura.
3. El usuario ingresa la información solicitada.	4. El sistema muestra las asignaturas que cumplan con el criterio ingresado.
5. El usuario selecciona la asignatura y el código de la cual desea consultar las competencias.	6. El sistema muestra las competencias y subcompetencias que tiene asociadas a esa asignatura.

Excepciones:



Paso 4: El sistema no encuentra el criterio ingresado para la búsqueda de la asignatura y se lo informa al usuario.

Curso alterno:

Actores: Coordinador de programa, Coordinador de la Unidad de enseñanza aprendizaje, Docente.

ACTOR	SISTEMA
1. El usuario selecciona "Consultar Competencia por docente"	2. El sistema solicita ingresar el número del documento del docente.
3. El usuario ingresa la información solicitada.	4. El sistema muestra el docente que cumpla con el criterio ingresado.
5. El usuario selecciona el docente.	6. El sistema muestra las competencias y subcompetencias que tiene asociadas ese docente en sus diferentes asignaturas.

Excepciones:

Paso 4: El sistema no encuentra el criterio ingresado para la búsqueda del docente y se lo informa al usuario.

Curso alterno:

Actores: Coordinador de programa, Coordinador de la Unidad de enseñanza aprendizaje, Docente, estudiante.

ACTOR	SISTEMA
1. El usuario selecciona "Consultar Competencia por estudiante"	2. El sistema solicita ingresar el código del estudiante.
3. El usuario ingresa la información solicitada.	4. El sistema muestra el estudiante que cumpla con el criterio ingresado.
5. El usuario selecciona el docente.	6. El sistema muestra las competencias y subcompetencias que tiene asociadas ese estudiante en sus diferentes asignaturas.

Excepciones:

Paso 4: El sistema no encuentra el criterio ingresado para la búsqueda del estudiante y se lo informa al usuario.

Paralelo a esto se llevaron registros de tiempos con los formatos de PSP y se utilizó el formato de informe resumen de PSP0.1 ya que se busca medir el tamaño del programa producido.

### 3.1.3.2 Actividades de diseño

En esta iteración se avanza en el diseño del sistema construyendo la matriz de control de acceso, que muestra a los diferentes actores con sus funciones.

#### 3.1.3.2.1 Matriz de control de acceso

En la siguiente figura se observa la matriz que permite definir los roles y funciones a los diferentes actores identificados en el sistema.

Función\Usuario	Estudiante	Docente	Gestión tecnológica	Coordinador UEA	Coordinador de programa	Registro académico
Cargar asignaturas			X			X
Cargar grupos			X			X
Administrar competencias				X		
Administrar subcompetencias				X		
Administrar niveles				X		
Asociar subcompetencias con asignaturas y niveles					X	
Valorar subcompetencias		X				
Consulta valoraciones	X	X			X	
Consultar competencias por estudiante	X			X	X	
Consultar competencias por asignatura		X		X	X	
Consultar competencias por docente		X		X	X	
Consultar competencias por nombre				X	X	

#### 3.1.3.2.2 Diseño de interfaces externas:

Para cumplir con el requerimiento no funcional RNF-04, se diseña la estructura de cuatro archivos en formato CSV<sup>9</sup>, para la carga de información generada por el sistema de registro académico de la UAM:

- Asignatura:
  - Código: código único de la asignatura (número de 6 dígitos).
  - Nombre: nombre de la asignatura.
- Grupo:
  - Código asignatura
  - Periodo: periodo académico (P.E "2014-03").
  - Número de grupo: se refiere al número del grupo según el periodo académico (número entero).
  - Documento docente: número de identificación del docente.
- Inscripción grupo:
  - Código asignatura
  - Periodo
  - Número grupo
  - Código del estudiante: código único que se le asigna a cada estudiante al ingresar al programa académico (conjunto alfanumérico máximo 12 caracteres).
- Plan estudio asignatura:
  - Código plan: código del plan de estudios (número de 4 dígitos).
  - Código asignatura.

### **3.1.3.2.3 Diseño de reportes**

---

<sup>9</sup> Coma Separated Values

Utilizando el pencil sketching<sup>10</sup> como un programa agregado del navegador, se ha diseñado una muestra del reporte que obtendrá el estudiante al seleccionar la opción correspondiente.

En la Figura 17 – Diseño reporte Valoración de competencias, se observa el diseño elaborado del reporte de valoración de competencias que presenta la asignatura y su código, el nombre de la competencia y de su subcompetencia asociada, el nivel determinado y si aprobó o no, esta vista puede ser consultada por los estudiantes y por los coordinadores de programa.

## UNIVERSIDAD AUTÓNOMA DE MANIZALES

### REPORTE VALORACIÓN DE COMPETENCIAS

Programa

Nombre Estudiante

Código

Asignatura	Código materia	Competencia	Subcompetencia	Nivel	Valoración
Matemáticas 1	10360	Competencia 1	Subcompetencia 1.1	Básico	<input checked="" type="checkbox"/>
Lógica de programación	10360	Competencia 2	Subcompetencia 2.1	Intermedio	<input type="checkbox"/>

Reporte generado el:

Fecha

Figura 17 – Diseño reporte Valoración de competencias

---

<sup>10</sup> <http://pencil.evolus.vn/>

## UNIVERSIDAD AUTÓNOMA DE MANIZALES

### Consulta de valoración por estudiante

<b>Código Estudiante</b>	<input type="text"/>
<b>Documento Estudiante</b>	<input type="text"/>
<b>Nombre Estudiante</b>	<input type="text"/>
<b>Apellidos Estudiante</b>	<input type="text"/>
<input type="button" value="Buscar"/>	

Figura 18 - Consulta por estudiante

En la Figura 18 - Consulta por estudiante se muestra el diseño de la consulta por estudiante, una vez implementado este diseño permitirá las búsquedas por cualquiera de los criterios mostrados.

## UNIVERSIDAD AUTÓNOMA DE MANIZALES

### REPORTE VALORACIÓN DE COMPETENCIAS POR ASIGNATURA

<b>ASIGNATURA</b>	<input type="text"/>	<b>Código</b>	<input type="text"/>	<b>Periodo</b>	<input type="text"/>
<b>DOCENTE</b>	<input type="text"/>	<b>Grupo</b>	<input type="text"/>		

Estudiante	Subcompetencia 1	Subcompetencia 2	Subcompetencia
Estudiante 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Estudiante 2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Reporte generado el:

<input type="text" value="Fecha"/>
------------------------------------

Figura 19 - Reporte por asignatura

En la Figura 19 se observa el reporte que le permitirá al docente, al coordinador de programa y al coordinador de la unidad de enseñanza aprendizaje, obtener las valoraciones de las asignaturas consultadas.

## UNIVERSIDAD AUTÓNOMA DE MANIZALES

### Consulta de valoración por docente

Documento Docente	<input type="text"/>
Nombre Docente	<input type="text"/>
Apellidos Docente	<input type="text"/>
Código Asignatura	<input type="text"/>
Nombre Asignatura	<input type="text"/>
	<input type="button" value="Buscar"/>

Figura 20 - Consulta por docente

En la Figura 20 se muestra el diseño de la pantalla que utilizaría el docente o el coordinador de programa para consultar las valoraciones que el docente debe realizar.

## UNIVERSIDAD AUTÓNOMA DE MANIZALES

### REPORTE VALORACIÓN DE COMPETENCIAS POR ASIGNATURA

Docente	Identificación	Código asignatura	Nombre Asignatura	Periodo	
Docente 1	104455677	10360	Lógica de Programación	2014 - 02	<input type="button" value="Ver"/>
Docente 2	5653434567	10360	Lógica de Programación	2014 - 03	<input type="button" value="Ver"/>

Figura 21 - Reporte por docente

En la Figura 21 se muestra el diseño del resultado de la búsqueda de la Figura 20, en el cual se puede ver los datos del docente y de las asignaturas que orienta.

### 3.1.3.3 Actividades de implementación:

#### 3.1.3.3.1 Valorar Competencias

Se implementaron controlador y vistas para la valoración de competencias

Inicio Facultad Competencias Niveles Asignaturas Valorar Competencias Salir (30392367)

[Home](#) » [Grupos](#) » [Lógica de Programación](#) » Valorar

## Valorar Competencias

Nombre	Lógica de Programación	
Numero	1	

Competencia Competencia Ciudadana Subcompetencia Competencia ciudadana 1.2 Listar

Nombre	Valoración	Estado
Apellido 1 Estudiante 1	<input type="checkbox"/>	Sin valorar
Apellido 2 Estudiante 2	<input type="checkbox"/>	Sin valorar
Apellido 3 Estudiante 3	<input type="checkbox"/>	Sin valorar

Guardar

Figura 22 - Vista preliminar de valorar competencias.

Una vez elegida la asignatura el docente puede seleccionar una competencia y subcompetencia para valorar. El sistema despliega una lista de los estudiantes que cursan la asignatura, una casilla para marcar si el estudiante alcanza la competencia.

### 3.1.3.3.2 Cargue de información

Se implementó un formulario que permite cargar en el sistema los archivos definidos previamente en el diseño de las interfaces externas.

sistema de gestión de competencias UAM  
UNIVERSIDAD AUTÓNOMA DE MANIZALES

Inicio Facultad Competencias Niveles Asignaturas Valorar Competencias Salir (30392367)

### Subir información

Campos con \* son requeridos.

Datos a cargar \*  
Asignaturas

Archivo de datos  
Browse... No file selected.

Cargar

Figura 23 - Vista preliminar del formulario para cargar información externa

### 3.1.3.4 Actividades de prueba

Se realizaron pruebas de tipo funcional a algunos de los controladores generados, a los módulos de Asignaturas y valorar competencias, debido a que son fundamentales en el funcionamiento del programa. Ver anexo casos de pruebas.

### 3.2.4.5 Actividades de postmortem

Se aplicaron las mismas herramientas de medición que en las iteraciones anteriores.

Fase	Tiempo	Porcentaje
Requerimientos	0:00:00	0,0%
Análisis	0:00:00	0,0%
Diseño	0:00:00	0,0%
Implementación	77:40:00	88,9%
Pruebas	10:30:00	12%
<b>Tiempo Total IT 3</b>	88:10:00	
<b>Tiempo en Días</b>	3 días 16 horas	

Tabla 7 - Total de tiempos según la fase

En la tabla anterior se observa que la cantidad de tiempo invertido en las fases de requerimientos, análisis y diseño ya es mínima, contrario de las fases de implementación y pruebas, las cuales toman la mayor parte del tiempo total.

Tipo de líneas	Cantidad
<b>Base:</b>	488 (2927)
<b>Borradas:</b>	528
<b>Modificadas:</b>	301
<b>Agregada:</b>	80
<b>Nuevas &amp; Cambiadas:</b>	381
<b>Total:</b>	2479

Tabla 8 - Cantidad de líneas de código para la tercera iteración

La tabla anterior toma los datos de las modificaciones realizadas durante la tercera iteración.



En las base se muestran entre paréntesis las líneas que arroja la herramienta utilizada para el conteo de líneas de código, 488 es la cantidad de líneas lógicas que fueron agregadas o modificadas por el desarrollador durante la primera y segunda iteración.

Tamaño Programa LOC	A la fecha
<b>Total LOC (T)</b>	7508
<b>Base (B)</b>	7764
<b>Borradas (D)</b>	553
<b>Modificadas (M)</b>	572
<b>Reutilizadas (R)</b>	0
<b>Agregadas (A)</b>	297
<b>Agregadas y modificadas (A&amp;M)</b>	869

Tabla 9 - Líneas de código para total del proyecto

En la tabla anterior se observa la cantidad de líneas de las 3 iteraciones, se han sumado las LOC de cada iteración y que han sido generadas por el programa Process Dashboard.

Fase	Tiempo
<b>Requerimientos</b>	12:29:00
<b>Análisis</b>	45:58:00
<b>Diseño</b>	16:33:00
<b>Implementación</b>	143:29:00
<b>Pruebas</b>	12:22:00
<b>Total tiempo desarrollo del proyecto</b>	230:51:00
<b>Total tiempo en días</b>	9 días 14 horas

Tabla 10 - Tiempo total del proyecto por fases

En la tabla anterior se encuentra el tiempo invertido para cada una de las fases de proyecto en horas, minutos, segundos. También se observa el tiempo total invertido en horas y en días.

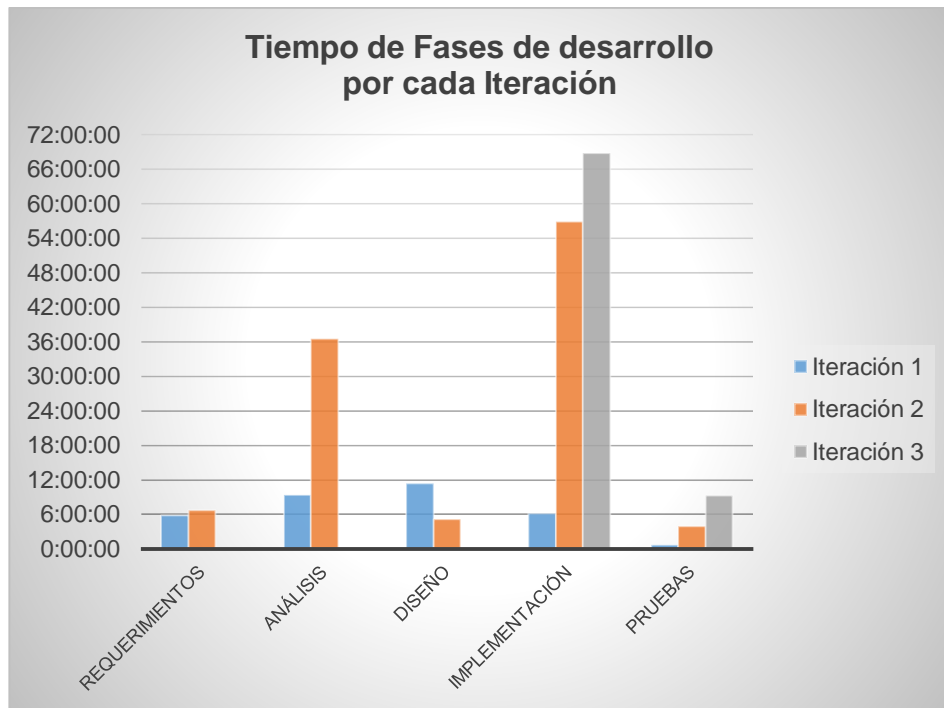


Figura 24 - Fases vs Tiempo invertido

En la Figura 24 se muestran las fases con su correspondiente tiempo invertido, se observa que para la iteración 1 e iteración 2 se invirtió tiempo en cada una de los flujos de trabajo, para la iteración 3 sólo se invirtieron tiempo en implementación y pruebas; implementación es la parte a la cual se le invierte más tiempo de forma general en cada una de las iteraciones, pruebas tiene participación en cada una de las iteraciones.

Requerimientos y análisis no han tenido participación de la iteración 3 debido a que ya estaban definidas en su mayoría desde el principio del proyecto y no se realizaron cambios por parte del cliente.

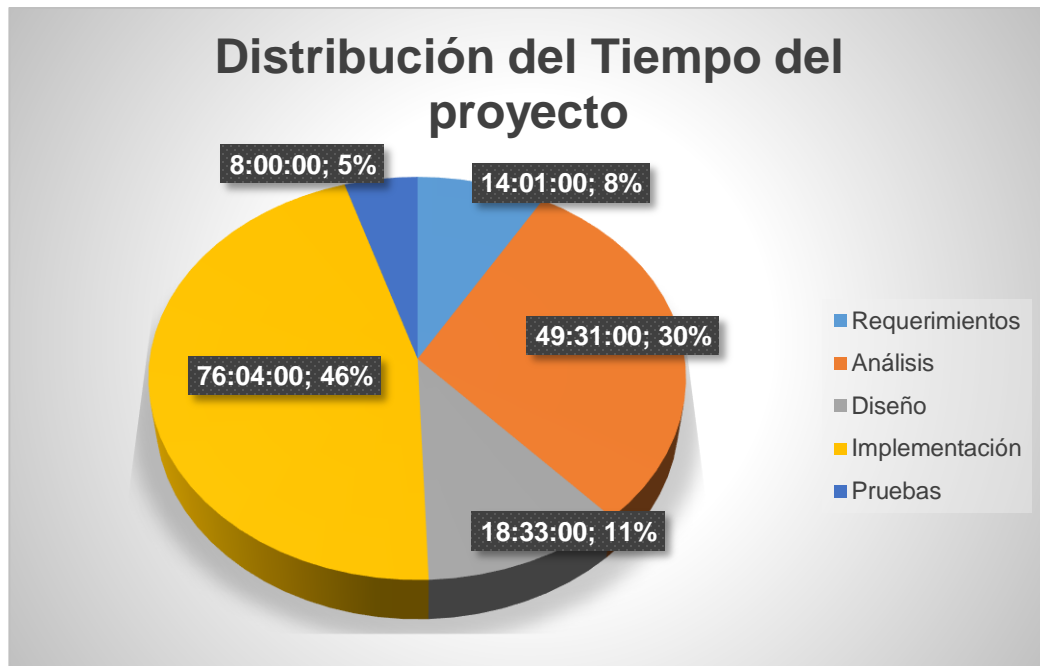


Figura 25 - Gráfico de tiempo total por fases

En la Figura 25, se muestra el tiempo en horas y el porcentaje correspondiente a cada fase del proyecto, mostrando su proporción respecto al tiempo total del proyecto.

En la fase de implementación se invirtió la mayor parte del tiempo, seguida en orden descendente de las de análisis, diseño, requerimientos y pruebas. En las pruebas se ha invertido la menor cantidad de tiempo ya que sólo se realizan las de tipo funcional y las de aceptación están por fuera del alcance del proyecto.

## 4 CONCLUSIONES

- El sistema implementado permite a los usuarios la definición y administración de competencias, subcompetencias, niveles y valoraciones en el contexto de las asignaturas orientadas en los programas académicos de la UAM®.
- Por medio del registro de los tiempos invertidos en cada fase e iteración en el desarrollo del software, cantidad de líneas de código y algunos defectos encontrados, se pudo determinar el esfuerzo del desarrollador y el tamaño de Software. Estos datos constituyen una línea base para futuros proyectos, permitiendo una mejor estimación de recursos, incrementando la eficacia del proceso de desarrollo personal.
- Al aplicar PSP se recopila valiosa información para el desarrollador, datos que servirán para la planeación y desarrollo de futuros proyectos para reducir la cantidad de defectos y el tiempo que se tarda en encontrarlos y repararlos, además se adquieren herramientas para mejorar la estimación de esfuerzos en proyectos similares y medir el tamaño del programa producido.
- Gracias a la adopción de un proceso de desarrollo de software formal y a la planeación del proyecto por iteraciones, se facilitó el desarrollo del mismo porque se definieron de forma clara las fases a ejecutar, los artefactos a construir y los entregables correspondientes a presentar.
- La aplicación de prácticas y de modelos de desarrollos de software como lo es UP, promueven la mejora del producto y facilitan la construcción de un software de buena calidad, haciendo énfasis en agotar cada una de las etapas del proceso de desarrollo de software. En este caso UP ha servido como guía para cada una de las actividades planteadas al inicio del proyecto, el uso de esta metodología facilitó la planeación del proyecto mitigando la falta de experiencia en procesos de desarrollos de software.

- Al realizar el diseño de un sistema parametrizado se le permite al cliente, en este caso, registro académico, coordinador Unidad de Enseñanza Aprendizaje y coordinadores de programa, realizar modificaciones al digitar las competencias, cambiar sus asociaciones y sus niveles. De esta forma se adapta a los eventuales cambios que realice la UAM® en lo que a competencias se refiere.
- El tiempo total empleado en horas fue de 230 aproximadamente, del cual la fase de implementación fue a la que más tiempo se le invirtió, la cantidad de líneas de código generadas fueron 7508, de las cuales 1024 fueron escritas por el programador. Dado que no se contaba con experiencia en la herramienta de desarrollo hubo una curva de aprendizaje de la configuración, API y el proceso de generación automática de código.
- Teniendo en cuenta lo anterior y basados en los objetivos específicos planteados, se evidencia el logro del proyecto de entregar un prototipo funcional acorde con los lineamientos de la UAM®, basándose en la metodología de Proceso Unificado, siguiendo las etapas de desarrollo de software y aplicando métricas con PSP0.1.

## 5 RECOMENDACIONES

- Para obtener datos ciertos se necesita llevar una estricta administración de los tiempos en PSP, ya que es común olvidar realizar los registros de pausas, de inicio o de fin, se sugiere utilizar un software especializado para realizar dichos procesos como el Process Dashboard, que es una herramienta que permita a un usuario registrar las etapas o fases de su proyecto, e iniciar y detener un temporizador que lleva registro del tiempo invertido en cada actividad.
- El proyecto que se presenta es un prototipo funcional, se recomienda a la UAM® considerar su validación y uso en el momento en el cual la universidad decida adoptar un sistema de formación basado en competencias y se pueda integrar con los sistemas que actualmente se manejan en ella.
- Se recomienda realizar un registro de los defectos encontrados y corregidos de forma sistemática, además tener clara su tipología antes de iniciar los flujos de desarrollo.
- Debido a la falta de experiencia en procesos de software, el tiempo empleado en el desarrollo fue mayor al planeado. Se considera que una opción de mejora para el currículo del programa es la realización de cursos nivelatorios antes de iniciar el plan de estudios de la maestría, además incluir más actividades tomadas del mundo real en cuanto a temas relacionados con elicitación de requerimientos, diseño de aplicaciones y herramientas de programación, sobre todo para estudiantes que no se desempeñan en las áreas relacionadas.
- Para implantar el software es necesario instalarlo en un servidor de la universidad, configurar su base de datos y la conexión con la base de datos de IntraUAM. También es necesario realizar una validación del sistema con los usuarios para determinar si se ajusta a los requerimientos de la institución y realizar cualquier cambio pertinente.
- El modelo de formación por competencias se aplica en algunas de las instituciones educativas con el ánimo de integrar conocimientos, habilidades, aptitudes y actitudes

de los estudiantes, pero necesita un cambio cultural para su aplicación, ya que requiere cambiar también el esquema de formación tradicional que se utiliza en este momento. Este proyecto permitiría una mejor adaptación a la nueva modalidad a los docentes y directivos en instituciones educativas que deseen cambiar el sistema de formación, de tradicional al basado en competencias como lo han hecho algunas instituciones a nivel nacional e internacional.

## 6 BIBLIOGRAFÍA

- BOOCH, G. e. (2000). *El Proceso Unificado de Desarrollo de Software*. Madrid: Pearson education.
- CORTÉS MORALES, R., & CORTÉS MORALES, R. (1998). *Introducción al análisis de sistemas y la ingeniería de software*. México: Editorial Universidad Estatal a Distancia.
- FERNÁNDEZ ALARCÓN, V. (2006). *Desarrollo de sistemas de información: una metodología basada en el modelado*. Barcelona: Ediciones UPC.
- Human Resource Systems Group. (2012). *Success through Competency-based Talent Management*, . <http://www.competencycore.com/>.
- Humphrey, W. (1995). *PSP(sm): A self-Improvement Process for Software Engineers*.
- Humphrey, W. (2000). *Introduction to the team software process*.
- IHRDC. (2012). *International Human Resources Development Corporation*. [www.ihrdc.com/](http://www.ihrdc.com/).
- L, G. (2006). *Currículo Universitario Basado En Competencias* . Colombia: Universidad del Norte.
- L., G. (2006). *Currículo Universitario Basado En Competencias*. Colombia: Universidad del Norte.
- O'BRIEN, J. M. (2006). *Sistemas de Información Gerencial*. México: McGraw Hill.
- O'Brien, & Marakas. (2006).
- POMEROY-HUFF, M. e. (2009). *The Personal Software Process, Body of Knowledge, Version 2.0*. Carnegie Mellon University.



Proyecto Tuning América Latina. (2004). Reflexiones y perspectivas en la Educación Superior. 2004 – 2007. *Proyecto Tuning América Latina*, 35 – 36.

RUIZ DE VARGAS, M. e. (2005). Competencias Laborales y la formación universitaria. En M. e. RUIZ DE VARGAS, *Sicología desde el Caribe* (págs. 64 – 99.).

SENA, s. N. (2008). *Dirección General*. Bogotá - Colombia: Sena Sofía Plus: <http://portal.senasofiaplus.edu.co>.

Sentrico. (2012). *Competency Management*. [www.sentricocompetencymanagement.com/](http://www.sentricocompetencymanagement.com/).

*The Fast, Secure and Professional PHP Framework*. (2014). Obtenido de <http://www.yiiframework.com/>

**ANEXO A**  
**PRESUPUESTO**

Estos fueron los recursos utilizados durante el desarrollo del proyecto.

<b>RECURSOS</b>			
<b>Materiales</b>	<b>Cantidad</b>	<b>Valor Unitario (\$)</b>	<b>Total (\$)</b>
Computador Portátil	1	\$ 1.800.000	\$ 1.800.000
Fotocopias	500	\$ 50	\$ 25.000
Hojas digitadas	200	\$ 1.000	\$ 200.000
Uso computador (horas)	250	\$ 1.000	\$ 250.000
Uso Internet (horas)	250	\$ 2.000	\$ 500.000
Empastado	4	\$ 25.000	\$ 100.000
Argollados	1	\$ 15.000	\$ 15.000
Papelería y útiles de oficina	15	\$ 500	\$ 7.500
CD's	5	\$ 1.000	\$ 5.000
<b>TOTAL</b>			<b>\$ 2.902.500</b>