

UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

"Propuesta de un modelo dinámico para la educación de la materia de Ingeniería de Software que reduzca el vacío entre la Universidad y la

Industria."

TESIS PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

PRESENTA:

Noé Coronel García

DIRECTOR DE TESIS:

Dr. Iván Antonio García Pacheco

Huajuapán de León, Oaxaca 15 de marzo de 2013

Tesis presentada el 22 de Marzo del 2013,
Ante los siguientes sinodales:

M.I.S. Hugo Enrique Martínez Cortés
Ing. David Martínez Torres
Ing. Wendy Yaneth García Martínez

Director de Tesis:

Dr. Iván Antonio García Pacheco

Dedicatoria

A la memoria de mi hermana Eva.

A mi madre porque a pesar de los sacrificios y carencias nunca dudo en darme lo más valioso: mis estudios.

A mis hermanos para que sigan adelante y logren sus metas.

Agradecimientos

De una manera especial al Dr. Iván Antonio García Pacheco, por la paciencia de todas esas fechas postergadas y la oportunidad de trabajar con él. Por todas las aportaciones, sus valiosas sugerencias e interés que otorgo al presente trabajo. Gracias por todo.

A los sinodales M.I.S. Hugo Enrique Martínez Cortés, Ing. David Martínez Torres e Ing. Wendy Yaneth García Martínez, por las sugerencias y haber dedicado parte de su tiempo durante la revisión del presente trabajo.

A mi madre, por su incansable esfuerzo e inmenso apoyo. Pero principalmente por darme la seguridad de que puedo terminar lo que comienzo, que mediante sus acciones me inculco la responsabilidad y la humildad.

A mis hermanos Mary, Hugo, Gris y Zoila por tener confianza en mí y haberme apoyado incondicionalmente para poder llegar hasta aquí.

A mis sobrinos Dylan y Hanna por esas grandes sonrisas que alegran a la familia, y que cuando llegue el tiempo el presente trabajo sea un aliciente para ellos.

A mi familia y amigos, gracias totales.

Resumen

En la actualidad, el avance en los campos de la educación, la investigación y el desarrollo de software crecen de manera considerable, pero cada parte lo hace de manera independiente, siguiendo su propia línea. Esto ocasiona que las empresas de software inviertan más tiempo y dinero en la capacitación del personal joven que recluta, esto a consecuencia de que las Instituciones no incluyan en sus planes de estudio cursos sobre lo que las empresas realmente necesitan. Es bien sabido que la Ingeniería de Software es una disciplina que pertenece a las Tecnologías de Información y Comunicación (TIC) y trata diversas áreas de las Ciencias de la Computación aplicables a un amplio espectro de campos, tales como los negocios, la investigación científica, la medicina, la producción, entre muchas otras.

Sin embargo existen diversos problemas en el proceso de desarrollo de software, algunos de ellos tienen que ver con el uso de las técnicas y herramientas, otros con las formas de enseñanza y aprendizaje, y unos más están relacionados con la divergencia de objetivos entre Universidades e Industria. Esto genera que el software que se construye generalmente no satisfaga los requerimientos ni las necesidades del cliente, además de que se exceden los presupuestos y tiempos establecidos. Debido a la complejidad en general de este proceso, también las personas implicadas en éste tienen varias dificultades. Y los nuevos profesionistas contratados por la Industria, y reclutados de las diferentes Universidades del país, las ven acrecentadas aun más.

En la actualidad existe un gran interés a nivel global sobre el desarrollo de modelos educativos orientados a la Ingeniería de Software, enfocados en la adquisición de experiencias y habilidades en proyectos reales. Algunos países como Brasil, India, Estados Unidos, Japón, Tailandia, Australia, están creando iniciativas en este sentido y han desarrollado modelos y metodologías de aprendizaje para la educación de la Ingeniería de Software. Los resultados de estos trabajos les permitirán encontrar soluciones futuras para el fomento y desarrollo de su industria de software.

En este contexto, la industria de software en México es demasiado joven aún, sin embargo es necesario establecer cambios en la currícula de materias y formas de enseñanza que se adecuen a la demanda actual. Esto hará que la industria crezca y camine de la mano junto con las Universidades, las cuales son su principal fuente de reclutamiento.

Por todo lo anterior, este trabajo de tesis pretende establecer las bases de un modelo dinámico que sería el primer paso al intentar alinear los objetivos del gobierno estatal y órganos colegiados quienes buscan fortalecer la madurez de la industria software desde sus etapas iniciales.

Índice general

| | |
|---|-----------|
| 1. Introducción. | 1 |
| 1.1. Introducción y motivaciones. | 1 |
| 1.2. Importancia del problema. | 4 |
| 1.3. Limitaciones del trabajo. | 7 |
| 1.4. Hipótesis del trabajo. | 7 |
| 1.5. Objetivos del trabajo. | 8 |
| 1.6. Aproximación a la solución | 8 |
| 1.7. Resultados generados. | 10 |
| 2. Marco Teórico sobre la Educación de la Ingeniería de Software. | 13 |
| 2.1. Aplicando PBL en la Educación de la Ingeniería de Software | 14 |
| 2.1.1. Objetivo | 14 |
| 2.1.2. Descripción | 14 |
| 2.1.3. Resultados | 18 |
| 2.2. IT Spiral: Un caso de estudio sobre la educación escalable de la Ingeniería de Software. | 19 |
| 2.2.1. Objetivo | 19 |
| 2.2.2. Descripción | 20 |
| 2.2.3. Resultados | 22 |
| 2.3. Modelo Global basado en la Competencia para fomentar las habilidades técnicas en la Educación de la Ingeniería de Software | 24 |
| 2.3.1. Objetivo | 24 |
| 2.3.2. Descripción | 24 |
| 2.3.3. Resultados | 28 |
| 2.4. Diseño e Implementación de un Curso Práctico de Ingeniería de Software | 31 |
| 2.4.1. Objetivo | 31 |
| 2.4.2. Descripción | 32 |
| 2.4.3. Resultados | 35 |
| 2.5. Comparativa empírica sobre las propuestas relacionadas con la vinculación Academia-Industria | 36 |
| 3. Introducción a la Solución del Problema. | 41 |
| 3.1. Bases del modelo dinámico para la enseñanza de la Ingeniería de Software | 41 |
| 3.2. Análisis del modelo dinámico en el contexto actual | 43 |

| | |
|---|-----------|
| 4. Diseño y desarrollo de VinculAE. | 61 |
| 4.1. Diseño Centrado en el Usuario Extendida | 62 |
| 4.1.1. Etapa 1: Entendimiento | 65 |
| 4.1.2. Etapa 2: Estudio | 65 |
| 4.1.3. Etapa 3: Diseño | 66 |
| 4.1.4. Etapa 4: Construcción | 66 |
| 4.1.5. Etapa 5: Evaluación | 67 |
| 4.2. Aplicación de DCU Extendida al problema identificado | 67 |
| 4.2.1. Análisis de usuario | 67 |
| 4.2.1.1. Perfiles de grupos de usuarios en DCU | 67 |
| 4.2.2. Problemática | 69 |
| 4.2.3. Perfil de los usuarios | 69 |
| 4.2.3.1. Grupo de usuarios: Alumnos | 69 |
| 4.2.3.1.1. Contexto | 69 |
| | |
| 4.2.3.1.2. Factores demográficos | 69 |
| | |
| 4.2.3.1.3. Entorno | 75 |
| | |
| 4.2.3.1.4. Experiencia en trabajos reales con empresas | 76 |
| | |
| 4.2.3.1.5. Experiencia en proyectos escolares | 76 |
| | |
| 4.2.3.1.6. Experiencia en el uso de sistemas vía web | 77 |
| | |
| 4.2.3.1.7. Frecuencia de uso: internet | 77 |
| | |
| 4.2.3.2. Grupo de usuarios: Profesores | 80 |
| 4.2.3.2.1. Contexto | 80 |
| | |
| 4.2.3.2.2. Factores demográficos | 81 |
| | |
| 4.2.3.2.3. Entorno | 85 |
| | |
| 4.2.3.2.4. Experiencia en trabajos reales con empresas | 86 |
| | |
| 4.2.3.2.5. Experiencia en proyectos escolares | 86 |
| | |
| 4.2.3.2.6. Experiencia en el uso de sistemas vía web | 86 |
| | |
| 4.2.3.2.7. Frecuencia de uso: Internet | 86 |
| | |
| 4.2.3.3. Grupo de usuarios: Empresarios | 86 |

| | | |
|------------|--|------------|
| 4.2.3.3.1. | Contexto | 86 |
| 4.2.3.3.2. | Factores demográficos | 86 |
| 4.2.3.3.3. | Entorno | 88 |
| 4.2.3.3.4. | Experiencia en trabajos reales con Instituciones educativas | 89 |
| 4.2.3.3.5. | Experiencia en proyectos escolares | 89 |
| 4.2.4. | Perfil de Usuario | 89 |
| 4.2.5. | Objetivos | 90 |
| 4.2.6. | Prototipos | 91 |
| 4.2.7. | Pruebas de Usabilidad | 97 |
| 4.2.7.1. | Roles en las pruebas | 98 |
| 4.2.7.2. | Resultados de las pruebas | 100 |
| 4.3. | Diseño de la Base de Datos | 102 |
| 4.3.1. | Diseño de tablas y atributos | 104 |
| 4.3.1.1. | Tabla “alumnos” | 104 |
| 4.3.1.2. | Tabla “profesores” | 106 |
| 4.3.1.3. | Tabla “empresarios” | 107 |
| 4.3.1.4. | Tabla “áreas” | 108 |
| 4.3.1.5. | Tabla “propuestas” | 109 |
| 4.3.1.6. | Tabla “proyectos” | 110 |
| 4.3.1.7. | Tabla “universidades” | 113 |
| 5. | Resultados. | 115 |
| 5.1. | Preparación del caso de estudio | 116 |
| 5.2. | Diseño y desarrollo del sistema para la gestión de carteras de proyectos | 118 |
| 5.3. | Hallazgos alcanzados | 124 |
| 5.3.1. | Problemas culturales | 124 |
| 5.3.2. | Problemas de negocio | 125 |
| 5.4. | Comprobación de la hipótesis | 125 |
| 6. | Conclusiones. | 127 |
| A. | Acrónimos | 135 |
| B. | Cuestionarios | 137 |
| B.1. | Cuestionario Empresarios | 137 |
| B.2. | Cuestionario Profesores | 138 |
| B.3. | Cuestionario Alumnos | 139 |

| | |
|--|------------|
| C. Registro de usuario | 141 |
| D. Tareas | 143 |
| E. Hoja de resultados de las tareas | 145 |
| F. Publicaciones | 147 |
| G. Carta de colaboración | 149 |

Índice de figuras

| | | |
|-------|---|----|
| 1.1. | Asociación entre los componentes básicos para mejorar la industria software. . . | 3 |
| 1.2. | Propuesta de modelo alternativo para reducir la brecha entre industria y academia. | 9 |
| 2.1. | El modelo IT Spiral. | 20 |
| 2.2. | Sesión IT Spiral. | 22 |
| 2.3. | El Modelo GSD en el 2008. | 28 |
| 2.4. | Monitor de estrés del estudiante. | 29 |
| 2.5. | Características comunes de las propuestas analizadas. | 36 |
| 2.6. | Diagrama de bloques de la solución planteada. | 38 |
| 3.1. | Módulos de capacitación para el modelo propuesto. | 42 |
| 3.2. | Diagrama a bloques de la solución planteada. | 59 |
| 4.1. | Las 4 etapas del modelo de investigación / diseño centrado en el usuario convencional. | 63 |
| 4.2. | Metodología DCU extendida. La nueva etapa implica el análisis conceptual o “entendimiento”. | 64 |
| 4.3. | Censo nacional de INEGI 2010. | 70 |
| 4.4. | Población escolar por régimen, 2008-2009. | 71 |
| 4.5. | Matrícula total por áreas de estudio. | 72 |
| 4.6. | Población escolar por áreas de estudio y régimen. | 73 |
| 4.7. | Distribución de matrícula por género de los estudiantes en el SUNEEO. | 74 |
| 4.8. | Distribución de estudiantes en las diferentes áreas del SUNEEO. | 74 |
| 4.9. | Egresados por carrera de las diferentes universidades del SUNEEO. | 75 |
| 4.10. | Frecuencia de uso de Internet. | 79 |
| 4.11. | Usuarios de internet. | 79 |
| 4.12. | Resumen nacional de datos de docentes. | 82 |
| 4.13. | Distribución de la educación superior. | 83 |
| 4.14. | Distribución del personal académico en el SUNEEO. | 84 |
| 4.15. | Distribución del personal académico del SUNEEO por áreas. | 85 |
| 4.16. | Datos del Estado de Oaxaca dentro de las entidades económicas del País. | 87 |
| 4.17. | Unidades económicas en el país. | 88 |
| 4.18. | Pantalla inicial del primer prototipo. | 92 |
| 4.19. | Pantalla general del primer prototipo. | 92 |
| 4.20. | Pantalla inicial del segundo prototipo. | 93 |

| | |
|--|-----|
| 4.21. Pantalla general del segundo prototipo. | 93 |
| 4.22. Pantalla de captura para dar de alta un proyecto. | 94 |
| 4.23. Pantalla inicial del tercer prototipo. | 95 |
| 4.24. Pantalla general del tercer prototipo. | 95 |
| 4.25. Pantalla de búsqueda del tercer prototipo. | 96 |
| 4.26. Pantalla de captura de proyecto para el usuario empresario. | 96 |
| 4.27. Pantalla que muestra los datos de un proyecto. | 97 |
| 4.28. UsaLab UTM. | 99 |
| 4.29. Usuarios probando las interfaces de VinculAE. | 100 |
| 4.30. Resumen de la aplicación de UCDE | 102 |
| 4.31. Modelo relacional de VinculAE. | 103 |
| 4.32. Tabla alumnos. | 105 |
| 4.33. Tabla profesores. | 106 |
| 4.34. Tabla empresarios. | 108 |
| 4.35. Tabla áreas. | 108 |
| 4.36. Tabla propuestas. | 110 |
| 4.37. Tabla proyectos. | 112 |
| 4.38. Tabla universidades. | 114 |
| 5.1. Propuestas registradas en VinculAE. | 116 |
| 5.2. Creación del equipo y asignación de proyecto. | 117 |
| 5.3. Arquitectura del sistema para la gestión de carteras de proyectos informáticos. | 119 |
| 5.4. Diagrama general de casos de uso de la herramienta propuesta. | 120 |
| 5.5. Compartición de documentos entregables para revisión. | 121 |
| 5.6. Alta de un proyecto en herramienta creada. | 122 |
| 5.7. Cuantificación de necesidades en herramienta creada. | 122 |
| 5.8. Selección del método para la evaluación económica en herramienta creada. | 123 |
| 5.9. Gestión de la cartera de proyectos en herramienta creada. | 124 |

Índice de tablas

| | | |
|-------|---|-----|
| 2.1. | Asignación de tiempo por disciplina. | 17 |
| 2.2. | Resultados de la encuesta IT Spiral. | 23 |
| 2.3. | Evolución del modelo GSD en 4 años de colaboración. | 25 |
| 2.4. | Múltiples percepciones sobre la calidad. | 29 |
| 2.5. | Comunicación entre roles(donde + representa intensidad). | 30 |
| 2.6. | Características para la enseñanza de la Ingeniería de Software. | 37 |
| | | |
| 3.1. | Descripción de la Ingeniería de Software en la UNAM. | 45 |
| 3.2. | Descripción de la Ingeniería de Software en la UAM. | 46 |
| 3.3. | Descripción de la Ingeniería de Software en el IPN. | 46 |
| 3.4. | Descripción de la Ingeniería de Software en la UV. | 47 |
| 3.5. | Descripción de la Ingeniería de Software en la UTM. | 47 |
| 3.6. | Módulo de repositorios de datos. | 48 |
| 3.7. | Módulo para el entorno de negocio. | 49 |
| 3.8. | Módulos de modelos de proceso. | 50 |
| 3.9. | Módulo de programación y aplicación del conocimiento. | 51 |
| 3.10. | Módulo de programación y aplicación del conocimiento (continuación). | 52 |
| 3.11. | Módulo de programación y aplicación del conocimiento (continuación). | 53 |
| 3.12. | Módulo de programación y aplicación del conocimiento (continuación). | 54 |
| 3.13. | Módulo de programación y aplicación del conocimiento (continuación). | 55 |
| 3.14. | Módulo de programación y aplicación del conocimiento (continuación). | 56 |
| 3.15. | Módulo de programación y aplicación del conocimiento (continuación). | 57 |
| 3.16. | Módulo de arquitectura de software. | 58 |
| | | |
| 4.1. | Población Escolar de Oaxaca en nivel superior, datos nacionales ANUIES. | 71 |
| 4.2. | Tabla por áreas de estudio del Estado de Oaxaca. | 73 |
| 4.3. | Formato de las tablas. | 104 |
| 4.4. | Descripción de campos de alumnos. | 104 |
| 4.5. | Descripción de campos de profesores. | 106 |
| 4.6. | Descripción de campos de empresarios. | 107 |
| 4.7. | Descripción de campos de áreas. | 108 |
| 4.8. | Descripción de campos de propuestas. | 109 |
| 4.9. | Descripción de campos de proyectos. | 111 |
| 4.10. | Descripción de campos de universidades. | 113 |

Capítulo 1

Introducción.

1.1. Introducción y motivaciones.

El término Ingeniería de Software fue usado ocasionalmente durante el final de la década de los 50's e inicios de los 60's. Se adjudica a Fritz Bauer, debido a que lo utilizó por primera vez en la primera conferencia sobre desarrollo de software patrocinada por el Comité de Ciencia de la OTAN, celebrada en Garmish, Alemania, y realizada en octubre de 1968. En esa época, el crecimiento espectacular de la demanda de sistemas cada vez más y más complejos, asociado a la inmadurez del sector informático y a la falta de métodos y recursos, provocó lo que se llamó en palabras de Edsger Dijkstra *“la crisis del software”*. Esta situación originó que se buscaran soluciones para escribir programas libres de defectos, fácilmente comprensibles y verificables. Así, se pretendía que con el surgimiento de la Ingeniería de Software se daría fin a la crisis¹.

Sin embargo, sin tener en cuenta cómo se defina al conjunto de problemas que originaron el surgimiento de la Ingeniería de Software, el proceso de desarrollo de software requiere, por un lado, un conjunto de conceptos además de una metodología y un lenguaje propio. El proceso de desarrollo de software no se limita al producto mismo, sino también a los problemas asociados a cómo desarrollarlo, cómo dar mantenimiento al software que ya existe y cómo mantener satisfecha la demanda creciente.

De acuerdo a Jacobson, el proceso de Ingeniería de Software se define entonces como *“un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad”* [35]. Por otro lado, el proceso de desarrollo de software se define como *“aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo”*. Concretamente este proceso *“define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto*

¹De acuerdo al diccionario Webster, una crisis se define como: *“un punto decisivo en el curso de algo, momento, etapa o evento decisivo o crucial”*. No obstante en el transcurso del desarrollo de software no ha habido ningún punto crucial, ningún momento decisivo, solo un lento cambio evolutivo. Según Daniel Tiechrow, profesor de la Universidad de Michigan, sería mucho más preciso definir esta situación como *“una aflicción crónica, algo muy duradero o que reaparece con frecuencia y continúa de forma indefinida”*

objetivo". A este proceso también se le llama el *ciclo de vida del software* y comprende cuatro grandes fases: concepción, elaboración, construcción y transición.

Ahora bien la producción de software es una actividad económica que se caracteriza por generar un alto valor agregado y aportar a la economía productos y servicios esenciales para su modernización. Esta industria se basa en el conocimiento, desarrolla habilidades más allá de la manufactura, propicia la innovación tecnológica y genera empleos bien remunerados, no contamina y requiere de relativamente poco capital para iniciar. La industria software forma parte del grupo de actividades económicas que componen a las Tecnologías de Información (TI). Éstas se integran además por la industria de hardware y los servicios, y junto con las Comunicaciones componen lo que se conoce como TIC. La incorporación de las TIC en los procesos de producción, comercialización, de servicios, de educación y de administración pública es un factor clave para mejorar la competitividad de las organizaciones y los países.

Durante la década de los noventa la mayor parte de los países, aún los que enfrentaron crisis financieras y recesiones económicas, incrementaron su gasto en TIC [54]. Si bien los países desarrollados continúan siendo líderes en esta materia, la demanda creciente no puede ser satisfecha con su oferta interna².

En este sentido, es importante mencionar que una proporción creciente de la producción mundial de software se realiza en países "en vías de desarrollo". India e Irlanda representan dos casos exitosos de creación y crecimiento de industrias nacionales basadas en la exportación. India, por ejemplo, ha maquilado software para Estados Unidos durante 15 años ¿Cómo lo ha hecho? de acuerdo a [36],[41] y [11], el gobierno de la India ha invertido varios billones de dólares en apoyo a la capacitación, certificación, promoción e infraestructura. Esto la llevó a alcanzar exportaciones anuales de software de aproximadamente 5,000 MDD en el año 2000. Al parecer y de acuerdo a experiencias en estos países, la combinación industria-universidades-educación es la clave del éxito (ver Figura 1.1).

Entre los factores que contribuyeron a que India empezara a producir software se encuentra la relación de los ingenieros hindúes residentes en Estados Unidos con las grandes empresas compradoras. Pero la principal clave es que abogó por dos estrategias base:

- Utilizar capital humano bien capacitado y con baja remuneración para desarrollar proyectos en su territorio.
- Desarrollar un método efectivo de entrenamiento a mano de obra funcional y avanzada y lo convirtió en un modelo mundialmente conocido, el modelo NITT [40],[43].

Irlanda por otro lado se especializó en el desarrollo de aplicaciones y basó el desarrollo de su industria software en la atracción de empresas extranjeras por medio de programas de apoyo sumamente agresivos. Logró un rápido crecimiento, teniendo una tasa de crecimiento anual promedio de 19.3 % (1993-2001) pero enfrentó la limitante de una mano de obra ya escasa.

²Por ejemplo, de acuerdo a Estados Unidos existía una brecha entre oferta y demanda de más de 800,000 desarrolladores de software que, simplemente por el tamaño relativamente reducido de la población joven de ese país, no podría eliminarse antes del año 2010.

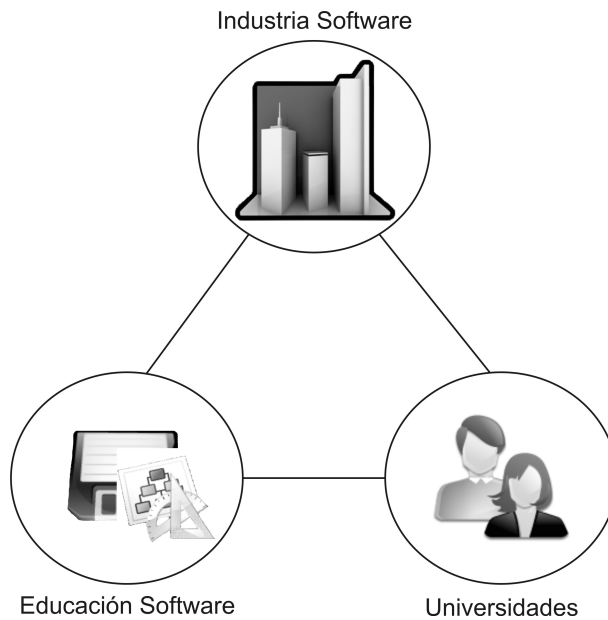


Figura 1.1: Asociación entre los componentes básicos para mejorar la industria software.

Sin embargo, para el año 2008, y de acuerdo a datos de la Autoridad del Desarrollo Industrial (IDA, *Industrial Development Authority*) [34], 9 de las 10 compañías en el ranking mundial tienen operaciones significativas en Irlanda, haciéndolo el país con mayor exportación de software en el mundo [65].

De acuerdo con la Conferencia de las Naciones Unidas sobre Comercio y Desarrollo (UNCTAD, *United Nations Conference on Trade and Development*) [67], el desarrollo de la industria software en los países “en vías de desarrollo” se puede llevar a cabo a través de dos alternativas de política estratégica:

- La primera estrategia: atención de demanda interna, se basa en el uso de software libre como una herramienta para el fortalecimiento de la infraestructura económica a fin de soportar el desarrollo de otros sectores de la economía.
- Y la segunda estrategia: generación de un proveedor internacional, se orienta a considerar a la industria del software como sector prioritario por la generación de nuevos empleos y el crecimiento industrial basado en la creación de oferta de software para exportación.

Sin embargo, el contexto de México se está centrando en la segunda estrategia con el fortalecimiento de la industria de TI anunciado en el Plan Nacional de Desarrollo del 2006. Esto quiere decir que una opción alternativa para el desarrollo de la industria software consistiría en integrar ambas estrategias bajo un orden secuencial; el uso de software libre como herramienta para el fortalecimiento de la infraestructura económica sería reemplazada gradualmente por la estrategia de desarrollo de software para exportación.

Pero, al igual que la India o Irlanda, los altos índices de exportación serán consecuencia de una industria software madura, con personal calificado y excelentemente bien dirigido.

Pero de ninguna forma México es igual que India o Irlanda, por lo que las condiciones específicas deben ser analizadas³. Muchos organismos internacionales han estudiado la situación actual de México en diversos aspectos de las TIC, tales como: el Connectivity Scorecard 2009, patrocinado por Nokia Siemens, y desarrollado por el profesor Waverman (miembro de la Escuela de Negocios de Londres) y con el apoyo de la firma de consultoría LECG. En este estudio se midió el grado en el cual los gobiernos, los negocios y los consumidores de 50 países hacen uso de tecnologías de la conectividad para aumentar su bienestar económico y social. De acuerdo a este estudio, México ocupa el quinto lugar en el uso de TIC de entre las 25 economías emergentes que integran este grupo [71].

Por otro lado, el foro económico mundial publicó el 21 de junio del 2009, el Reporte de Competitividad de México 2009⁴, realizado en colaboración con la Universidad de Harvard. De acuerdo al reporte, México ocupa el lugar 60 en el índice de competitividad global 2008-2009. Así mismo, cuenta con grandes fortalezas como: el tamaño de su mercado, el sector privado sofisticado y la estructura de producción diversificada y las bases macroeconómicas sólidas que posee. Sin embargo tiene una serie de debilidades que aún persisten como: la calidad de sus instituciones públicas, el mercado de bienes de consumo y el mercado laboral, el nivel de educación, y el potencial de innovación.

De manera similar, en el 2007 la Unidad de Inteligencia de *The Economist* realizó el estudio “Índice de Competitividad en TI” para la Business Software Alliance (BSA). El estudio evalúa a 66 países y compara los entornos de la industria de TI para determinar el grado en el que se propicia la competitividad del sector a través del análisis de seis áreas sustantivas para la industria. En este estudio México ocupó el lugar número 44 a nivel mundial en el año 2008 y mantuvo la misma posición que alcanzó en el año 2007. Sin embargo existen áreas relevantes en las cuales hubo retrocesos, mismas que consideran retos fundamentales para el país ya que lo ubican con los indicadores más bajos en la escala global.

1.2. Importancia del problema.

Dadas las razones anteriores, México se encuentra rezagado como competidor a nivel internacional y en consecuencia ha implementado varios programas para elevar y extender la competitividad, mediante la estrategia de promover el uso y aprovechamiento de la TI. Entre ellos se encuentran los siguientes:

³Simplemente el concepto promovido en Irlanda e India de “reevaluar el rol económico de la educación considerando a esta como una inversión en vez de un gasto” nos hace diferentes.

⁴El Reporte de Competitividad ofrece una visión general sobre el panorama actual del país sobre este campo, centrándose en las fortalezas y en las áreas más problemáticas.

- El 9 de octubre del 2002 la Secretaría de Economía lanzó el Programa para el Desarrollo de la Industria del Software (PROSOFT) con el objetivo de crear las condiciones necesarias para que México cuente con una industria de software competitiva a nivel internacional y asegurar su crecimiento a largo plazo. El resultado directo de esta iniciativa residió en el surgimiento de un Modelo mexicano denominado MoProSoft [50].
- Otro programa lanzado en el 2008, el Programa de Desarrollo de la Industria de Medios Interactivos (PROMEDIA), tiene el objetivo de crear las condiciones necesarias para asegurar el crecimiento y la consolidación de la industria de medios interactivos en México, así como elevar su competitividad internacional mediante la TI.
- El sector empresarial en México ha llevado a cabo proyectos y estrategias enfocadas a desarrollar a las empresas de TIC, interviniendo diversos organismos como la Cámara Nacional de la Industria Electrónica, de Telecomunicaciones y Tecnologías de Información (CANIETI), la Asociación Mexicana de la Industria de Tecnologías de Información (AMITI), la Asociación Mexicana de Internet (AMIPCI), la Asociación Nacional de Distribuidores de Tecnología Informática y Comunicaciones (ANADIC), la Asociación Mexicana Empresarial de Software Libre (AMESOL) y el Instituto Mexicano de Teleservicios (IMT).
- La Secretaría de Economía y el Consejo Nacional de Ciencia y Tecnología (CONACYT) suscribieron un convenio de colaboración con el objeto de constituir un fideicomiso con recursos concurrentes al que se le denomina “Fondo de Innovación Tecnológica Secretaría de Economía-CONACYT” y apoyar proyectos de innovación tecnológica que permitan mejorar la competitividad de las empresas mexicanas. Una de las líneas estratégicas del plan son las TI.

Sin embargo, y a pesar de estos esfuerzos, existe una problemática poco atendida: el capital humano. Actualmente los planes y programas de estudio de las Universidades están desvinculados del sector empresarial, existen pocos esfuerzos involucrados para desarrollar el capital humano que se requiere en esta industria. En este contexto, la Asociación Nacional de Instituciones de Educación en Informática (ANIEI) ha realizado algunos proyectos para mejorar el perfil de los egresados del sistema educativo, sin embargo el problema aun persiste.

Los diseñadores de los cursos de Ingeniería de Software con frecuencia encuentran dificultades para definir los objetivos de aprendizaje de acuerdo a las necesidades de la industria y por consiguiente diseñar evaluaciones que los satisfagan. Los cursos están orientados hacia el aprendizaje de herramientas y métodos en casos de estudio que poco o nada tienen que ver con la realidad.

Actualmente existen iniciativas en algunos países como Brasil, India, Estados Unidos, Japón, Tailandia y Holanda para desarrollar modelos y metodologías de aprendizaje en la educación y entrenamiento de la Ingeniería de Software y lograr que los egresados de las Universidades adquieran las habilidades y experiencias en proyectos reales, por ejemplo:

- Universidades de E.E.U.U. y Asia han estado utilizando durante cuatro años un modelo llamado GSD, el cual es un proyecto de Desarrollo de Software Global [25]. Este modelo se basa en la competencia y tiene como uno de sus objetivos lograr la sinergia entre estudiantes e Industria. Este modelo se inicia con los aspectos básicos del desarrollo de software para que los estudiantes establezcan una relación de confianza con los profesores y posteriormente logren establecer relaciones de soporte entre ellos y avanzar así a las siguientes fases.
- El Centro de Sistemas de Estudios Avanzados de Recife, Brasil; ha desarrollado un método de aprendizaje basado en problemas reales apoyado en la metodología PBL. Éste se basa en un aprendizaje unilateral entre profesor y estudiante. Se debe crear un ambiente donde los estudiantes estén inmersos en la práctica, y puedan recibir guía y soporte de sus amigos y compañeros. El aprendizaje es funcional basado en la resolución de problemas reales.
- En Kansai, Japón se desarrolla un proyecto que usa aprendizaje avanzado basado en la realidad, en este colaboran 9 Universidades y 4 empresas. Este modelo se llama Espiral IT.

El Gobierno del Estado de Oaxaca y el CONACYT, con fundamento en lo dispuesto en la Ley de Ciencia y Tecnología (LCYT), han constituido un fideicomiso con recursos concurrentes denominado “Fondo Mixto de Fomento a la Investigación Científica y Tecnológica CONACYT – Gobierno del Estado de Oaxaca”, con el propósito de apoyar proyectos de investigación científica y tecnológica que generen el conocimiento requerido para resolver problemas, necesidades u oportunidades del Estado de Oaxaca; fortalezcan sus capacidades científicas y tecnológicas, y competitividad de sus sectores productivos, con la finalidad de contribuir al desarrollo económico y social del Estado de Oaxaca. La edición 2009 de esta convocatoria convocó a Instituciones, Universidades públicas y particulares, Centros de Investigación, Laboratorios y personas físicas y morales a la presentación de propuestas de investigación científica y tecnológica que respondieran a demandas establecidas en cuatro áreas específicas: competitividad, desarrollo regional, minería, tecnologías de información y comunicaciones.

Específicamente la “Demanda 4.1 Fortalecimiento de la industria de tecnologías de información y comunicaciones de Oaxaca mediante la certificación TSP/PSP” de esta convocatoria argumenta la “necesidad de certificar en los modelos People Software Process (PSP) y Team Software Process (TSP) a un miembro del Clúster Oaxaca TI y a una persona de una Institución de Educación Superior, con el objeto de convertirse en instancias que proporcionen la capacitación y consultoría a las empresas de tecnologías de información y comunicaciones en la entidad en dichos modelos”.

Sin embargo y de acuerdo al creador de los modelos PSP [29] y TSP [30], Watts Humphrey, la pieza fundamental para que esto ocurra en México son las Universidades [31]. Las principales recomendaciones de Humphrey son:

- Es necesario incluir más cursos de Ingeniería de Software en los planes de estudio.
- Los catedráticos deben poseer la facultad para aprender y para enseñar los métodos PSP y TSP.

- Los universitarios deben aprender a trabajar en equipos TSP.
- Debe impulsarse la obtención de certificaciones de ingenieros en PSP y organizaciones en TSP.

A pesar de esto, los esfuerzos por impulsar los tres primeros puntos de Humphrey (relacionados con la educación de la Ingeniería de Software) son nulos dado que los modelos educativos que se enfocan en la enseñanza de esta disciplina no están basados en las necesidades actuales de la industria. Por lo cual, se pretende realizar un modelo dinámico en el que los estudiantes establezcan relaciones de trabajo y competencia entre ellos, para la resolución de casos reales y reducir así el tiempo de adaptación o integración al sector industria.

1.3. Limitaciones del trabajo.

El desarrollo del modelo educativo en el aprendizaje de Ingeniería de Software tendrá las limitantes de implementarse en un ambiente restrictivo, debido a algunos factores que se listan a continuación:

- Escasez de proyectos paralelos de desarrollo de métodos educativos en el aprendizaje de Ingeniería de Software y que pudieran servir de retroalimentación.
- Existe un número reducido de empresas u organismos dedicados al desarrollo de software a nivel regional. Y los que existen se encuentran desvinculados de las universidades.
- Falta de espacios, equipos tecnológicos y herramientas de software para el desarrollo.

La aplicación de un modelo educativo requiere de aprendizaje, requiere tiempo para evaluar sus resultados y realizar los cambios y ajustes correspondientes para avanzar a las siguientes fases. Diferentes universidades en otros países, han utilizado estos modelos durante años para irlos perfeccionando y ajustarlos a los requerimientos que la industria del software requiere. Pero la identificación de necesidades en modelos TSP y PSP que se complementen con MoProSoft es un excelente punto de partida.

1.4. Hipótesis del trabajo.

La principal hipótesis de este trabajo se puede formular de la siguiente forma:

"El establecimiento de un modelo dinámico para la educación de la Ingeniería de Software permitirá reducir las diferencias entre la oferta académica de las Universidades Regionales y las necesidades reales de la industria de software".

1.5. Objetivos del trabajo.

El objetivo general de este trabajo de tesis se puede resumir como:

“Crear un modelo dinámico para la educación de la materia de Ingeniería de Software que proporcione a los estudiantes las habilidades y experiencias en la resolución de casos reales, y que les permitan adaptarse fácilmente al ámbito industrial del país”

Este objetivo estará apoyado por varios objetivos específicos:

- Realizar un análisis exploratorio sobre las principales propuestas relacionadas con el tema de tesis e identificar aspectos comunes para adaptarlos a la condición específica de México.
- Analizar y adaptar la metodología PBL aplicada en [14], como soporte al modelo propuesto en esta tesis.
- Establecer los principios y bases de un modelo alternativo adecuado, en primera instancia, a la Universidad Tecnológica de la Mixteca.
- Desarrollar una herramienta de software que permita utilizar el modelo propuesto.
- Validar la herramienta de software para obtener resultados preliminares sobre los cursos de Ingeniería de Software impartidos en la Universidad Tecnológica de la Mixteca.

Las aportaciones de este trabajo se pueden dividir de la siguiente manera:

- Aportaciones teóricas y de investigación de base: Se aplicarán métodos de enseñanza-aprendizaje que permitan fomentar un aprendizaje unilateral de profesores y estudiantes.
- Aportaciones metodológicas: Los resultados obtenidos servirán como base para encontrar soluciones futuras en el desarrollo de software, en la aplicación de herramientas de investigación y propuestas de evaluación de planes y programas de estudio.
- Aportaciones prácticas: Se realizarán proyectos prácticos basados en casos reales de tipo industria, que permitan a los egresados crear lazos de soporte entre sus compañeros y profesores y vincular su trabajo con la solución.

1.6. Aproximación a la solución

Tomando en cuenta el surgimiento de un Modelo de Procesos mexicano, MoProSoft, y los comentarios de Humphrey orientados a cultivar el mejoramiento de la calidad educativa relacionada con la Ingeniería de Software mediante cursos teóricos y prácticos de PSP y TSP, se propone la creación de un modelo alternativo compuesto por cinco módulos. Se pretende que estos módulos ayuden a los académicos universitarios en la provisión del conocimiento necesario dentro del contexto actual de la industria software del país. En la Figura 1.2 se muestran estos módulos desde un punto de vista muy general, sin embargo será necesaria la descomposición en áreas o niveles de mayor granularidad.

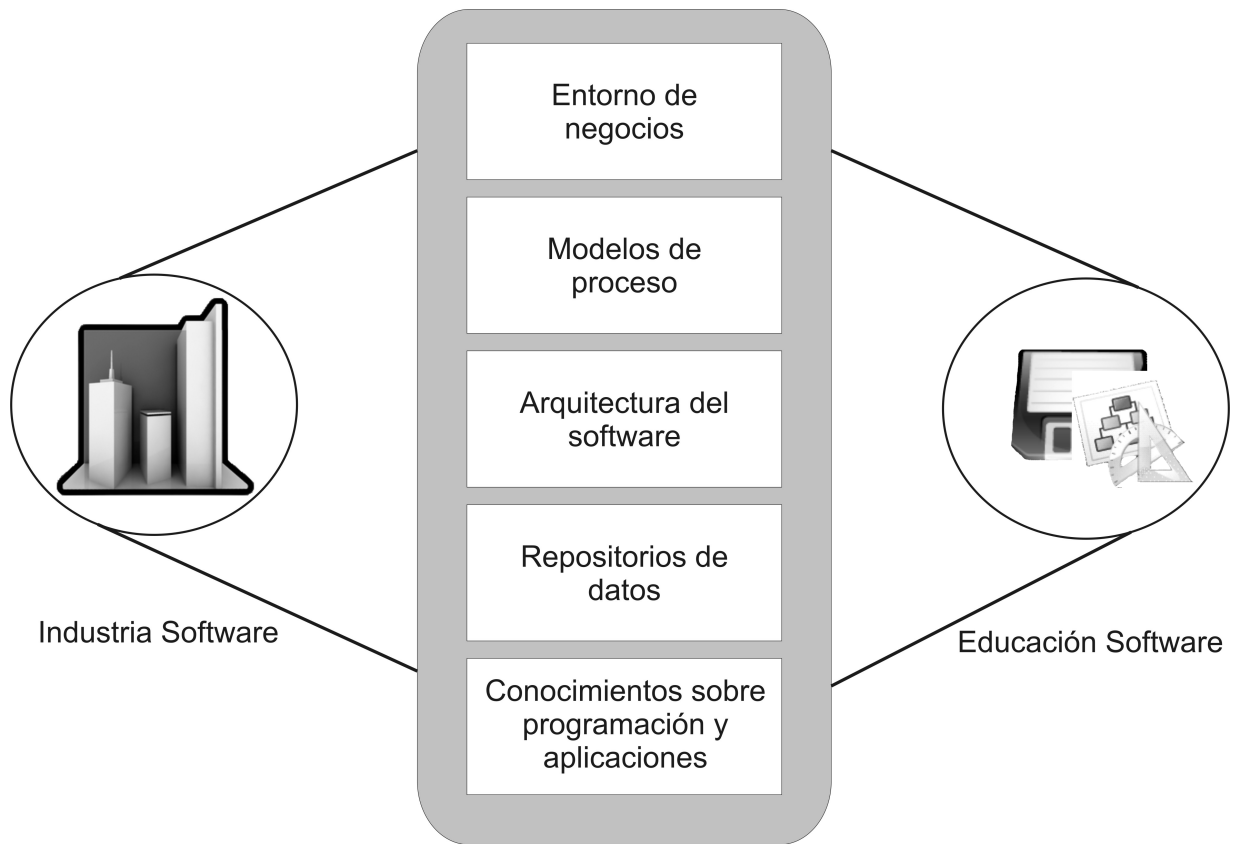


Figura 1.2: Propuesta de modelo alternativo para reducir la brecha entre industria y academia.

Los módulos del modelo pueden ser descritos brevemente de la siguiente forma:

- **Módulo de entorno de negocios**

El módulo del entorno empresarial proporciona el conocimiento relacionado con la implementación de soluciones de TI en un dominio dado de negocios. En cualquier dominio de negocios, los factores más importantes son las reglas de negocio, los procesos de negocio, el rendimiento y la seguridad. Debido a esto, la característica distintiva de este módulo será la combinación de educación en las clases y cooperación con los negocios.

- **Módulo de modelos de proceso**

Las metodologías tradicionales de ciclos de vida para el desarrollo de software están lejos de ser útiles en la industria actual. Hoy en día, las empresas controlan los cambios a los requisitos, exigen soluciones en el menor tiempo posible y con un alto nivel de calidad. De acuerdo a las necesidades mexicanas actuales de la industria software, los modelos como MoProSoft y TSP son cruciales para lograr su impulso. El módulo de modelos de proceso proporcionará el conocimiento relacionado con el aprendizaje y la práctica de modelos específicos del proceso software, incluido el CMMI [32] [33]. El modelo alternativo será apoyado por las herramientas desarrolladas por los mismos estudiantes. La educación se basará en la Arquitectura de Aprendizaje Centrada en el Modelo, o MCLA, enfocándose

en la formación de capacidades con el objeto de obtener entrenamiento eficaz. Los estudiantes universitarios, una vez que están graduados, sabrán desempeñar roles importantes en los negocios y la industria.

- **Módulo de arquitectura del software**

El módulo de arquitectura proporcionará el conocimiento sobre los estilos arquitectónicos del software. Es bien sabido que de acuerdo a cómo aumentan las demandas del negocio, la arquitectura necesita ser el punto central para las muchas soluciones del negocio. Los estudiantes aprenderán que varios factores afectan a la arquitectura. Este módulo incluirá el estilo de arquitecturas orientadas al proceso y orientadas a mensajes, orientados al objeto y orientadas al servicio, arquitecturas específicas del dominio y arquitectura de la empresa, las cuales son aplicadas de acuerdo a las necesidades futuras del negocio.

- **Módulo de repositorio de datos**

Cualquier dato que ayude a un negocio en la toma de decisiones y apoye las habilidades de los estudiantes es parte del módulo del depósito de datos. Este módulo proporcionará conocimientos sobre las tendencias para las operaciones del negocio y de la academia usando el almacenamiento, los reportes y el análisis y la gestión del rendimiento. Todas las referencias, textos y materiales didácticos serán cargados en el módulo y serán creados a partir de las necesidades de las empresas y tendencias de tecnología, y serán actualizados de acuerdo al progreso en la tecnología y el desarrollo de la industria mexicana.

- **Módulo de conocimientos sobre programación y aplicaciones**

Este módulo se refleja en cada curso que incluya tecnología práctica donde cada tópico sigue la huella del uso tecnológico diario. El módulo se puede categorizar en dos grupos: cursos fundamentales del software; y cursos de desarrollo y programación de software según los cambios de las demandas mexicanas de la industria software.

El módulo de conocimientos sobre programación y aplicaciones combina la educación con el desarrollo práctico para mejorar las habilidades del estudiante. Todos aquellos que están relacionados con la formación en TI e Ingeniería de Software pueden decir fácilmente que el plan de estudios está enfocado a la práctica directa.

En este sentido, ¿cuál será el soporte del modelo? Combinar teoría y práctica con desarrollos reales en un entorno real de trabajo, con estudiantes de Licenciatura y Maestría.

1.7. Resultados generados.

A continuación se enlistan algunas de las publicaciones que se generaron durante el desarrollo del presente trabajo.

Autores: García, I., Pacheco, C. & Coronel, N.
Título: “Aprender a través de la Práctica: Definiendo un Modelo Alternativo para la Educación de la Ingeniería de Software en las Universidades Mexicanas como medio para reducir el vacío entre la Industria y la Academia”
Congreso: Coloquio Nacional de Investigación en Ingeniería de Software y Vinculación Academia-Industria, CONIIS’10.
Publicación: Memorias de CONIIS 2010
ISBN: 978-607-7753-65-0.
Páginas: 93-99.
México, 2010.
Lugar: León Guanajuato, México.
Año: 2010.

Autores: García, I., Pacheco, C. & Coronel, N.
Título: “Learn from Practice: Defining an Alternative Model for Software Engineering Education in Mexican Universities for reducing the breach between Industry and Academia”
Revista: WSEAS Applied Computer Science
Publicación: ISSN: 1792-4863.
Páginas: 120-124.
Bulgaria, 2010.
Lugar: Malta.
Año: 2010.

Capítulo 2

Marco Teórico sobre la Educación de la Ingeniería de Software.

De acuerdo con la Norma IEEE del Glosario de Terminología de la Ingeniería de Software, esta se define como: (1) *La aplicación de un enfoque sistemático, disciplinado y un enfoque cuantificable al desarrollo, operación y mantenimiento de software;* (2) *El estudio de enfoques como en (1).* Así, al igual que en otras áreas de la ingeniería, la Ingeniería de Software está preocupada por la interacción de las personas, la tecnología, el dominio y aspectos que le permitan mejorar su forma de enseñanza. Sin embargo, cuando se habla de “enseñanza” se toca un tema sensible, pero muy importante en el proceso de la educación. El mecanismo de enseñanza ha estado atrayendo la atención de pensadores de la Filosofía, la Psicología, la Educación, y de las Ciencias de la Computación. Los conductistas ven a la enseñanza como un cambio relativamente permanente en el comportamiento debido a la experiencia y control del ambiente externo. Los psicólogos cognitivos perciben a las asociaciones mentales como un cambio relativo debido a la experiencia y creen que los seres humanos son capaces de discernir, percibir y atribuir un significado. Los psicólogos sociales lo ven como una empresa social, dependiendo de la interacción entre el alumno y su entorno socio-cultural. Los humanistas enfatizan en el desarrollo integral de la persona y hacen hincapié en el dominio afectivo. El constructivismo destaca que todo conocimiento está vinculado al contexto y que los individuos toman un significado personal de sus experiencias de aprendizaje a través de la construcción interna de la realidad.

Actualmente la industria de las TI se enfrenta a un mercado de constantes cambios y desafíos, lo cual hace obsoletos en un corto tiempo ciertos conocimientos. Dichas características demandan a los profesionales de las TI tener una visión amplia del problema, y no sólo el conocimiento de la tecnología. Así, las TI enfrentan un mercado voluble, en el cual, los profesionales del área a menudo tienen que adaptar conceptos, métodos y técnicas para situaciones específicas de los problemas del mercado, en general, influenciada por las variaciones de alcance, costos, plazos y riesgos de los proyectos de software.

Por otra parte, la evolución del mercado de las TI, se muestra como una ventaja competitiva de las organizaciones, donde se demanda el fomento de manera diferente e innovadoras para el desarrollo de software con más calidad, flexibilidad y rendimiento que se manifieste en la pro-

ductividad. Estas características del mercado de las TI promulgan una demanda de profesionales, particularmente de Ingenieros de Software, que tengan no sólo el conocimiento sobre tecnologías, sino también una amplia visión de los problemas a resolver, comprensión de negocios, espíritu empresarial y habilidades interpersonales, todo relacionado con la experiencia práctica del trabajo.

Por lo tanto, la industria y el mundo moderno dependen cada vez más de complejos sistemas de software. Sin embargo, los problemas con estos sistemas a menudo son más cuestiones sociales que dificultades técnicas. Este tipo de incidentes se observan en sistemas bancarios, bolsa de valores, sistemas de control de tránsito y otras áreas. Pero existe aún un problema más grave: la oferta insuficiente de Ingenieros de Software.

De esta manera, diversas instituciones han desarrollado iniciativas centradas a mejorar las habilidades de los aprendices de la Ingeniería de Software mediante su participación en proyectos reales controlados por la industria. En este sentido, estas iniciativas pretenden establecer un vínculo de colaboración que permita introducir a los estudiantes en entornos reales de trabajo. Con el objetivo de establecer las bases de un modelo dinámico para la educación de la Ingeniería de Software, a continuación serán analizadas algunas propuestas relacionadas con la reducción de la brecha entre Universidad e Industria.

2.1. Aplicando PBL en la Educación de la Ingeniería de Software

En el contexto de esta tesis, la metodología pedagógica apoyada en el Aprendizaje Basado en Problemas (PBL, *Problem-Based Learning*) [60] para la educación de la Ingeniería de Software presenta algunos resultados importantes, obtenidos de su aplicación en un curso de posgrado en Ingeniería de Software con un énfasis profesional. PBL se utiliza para mejorar la eficacia del aprendizaje, promoviendo la capacidad de los estudiantes para trabajar en equipo y resolver problemas que fomenten el desarrollo de sus habilidades y aptitudes; incluyendo al trabajo en equipo y habilidades de aprendizaje auto-dirigido, a la cooperación, a la ética y al respeto del punto de vista de los demás [57],[66].

2.1.1. Objetivo

Dos Santos et al. proponen en [14] la aplicación de una metodología pedagógica innovadora basada en PBL para mejorar la efectividad del aprendizaje en la Ingeniería de Software, lo cual se logra mediante la implementación de *fábricas de software*, donde los estudiantes puedan trabajar en conjunto para resolver problemas reales.

2.1.2. Descripción

En [53], Peterson nombra tres características importantes que promueven el aprendizaje óptimo en el enfoque de PBL:

- Un entorno donde el alumno está inmerso en la práctica;
- La orientación y el apoyo que el estudiante recibe de sus amigos y compañeros; y
- El aprendizaje funcional, basado en la solución de un problema real.

Teniendo como referencia los criterios definidos por Peterson, la propuesta de Do santos et al. proporciona un entorno basado en fábricas de software en el que los estudiantes están inmersos en la práctica de proyectos de desarrollo de software tomados de clientes reales, con el apoyo de procesos, roles y métricas para controlar los resultados obtenidos. Sin embargo, la adopción de PBL no es una tarea fácil y con frecuencia resulta sorprendente tanto para los estudiantes como para los profesores, debido a su paradigma que difiere totalmente de la enseñanza convencional, que se basa en el aprendizaje unilateral maestro-estudiante.

Así, en este contexto la metodología propone el uso de la fábrica de software como el entorno adecuado para desarrollar prácticas para los estudiantes de Ingeniería de Software, de acuerdo con la característica (1) de Peterson. El modelo de fábrica de software utilizado por la metodología tiene por objeto la normalización de las buenas prácticas, con el fin de mejorar gradualmente las herramientas, las técnicas, y establecer un control adecuado mediante la definición de métricas y evaluaciones [4]. Para apoyar este ciclo de vida, se requiere de algunos contenidos, guías y procesos. De esta forma, es necesario identificar a los interesados y sus funciones, para asignarles un rol en cada paso. Dado lo anterior, la metodología se apoya en cuatro funciones que interactúan entre sí para ejecutarse a lo largo del semestre:

- El equipo de la fábrica de software (Equipo SWF , *Software Factory Team*), que está diseñado para explorar las diferentes capacidades en el desarrollo de software. Está compuesto por estudiantes, quienes desempeñan todas las funciones en el desarrollo de software como: jefe del proyecto, analista de requisitos, arquitecto de software e ingeniero de calidad y pruebas. Se recomiendan equipos de 5 a 7 miembros, por lo que cada alumno realiza más de una función en el equipo. Es importante destacar que todos los estudiantes se involucran en cada uno de los pasos del proceso de desarrollo de software.
- El tutor, quien es el responsable de enseñar el programa previsto para cada disciplina, coordinar las actividades académicas y actuar como consultor en la disciplina especificada. Además de lo anterior, debe evaluar sistemáticamente a los estudiantes y proporcionar a la fábrica de software monitores de información sobre las actividades prácticas que sean necesarias. Por lo tanto, además de la experiencia académica y de investigación, es esencial que el tutor tenga suficiente habilidad en el área de conocimiento de sus clases, y es deseable que cuente con una participación exitosa y relevante en proyectos de desarrollo de software (ya sea de tipo escolar o a nivel empresarial).
- El monitor de la fábrica de software (Monitor SWF), es responsable de dar seguimiento al desarrollo de las fábricas, generar informes, resultados y el estado sobre las mismas, así como evaluar el desempeño y compromiso de cada miembro del equipo.
- El cliente, al igual que en un entorno real, solicita los proyectos a través de una solicitud de

propuesta (RFP, *Request For Proposal*). Este documento tiene una breve explicación de las necesidades del cliente y debe estar disponible para interactuar y realizar un seguimiento en cada fábrica, de acuerdo al desarrollo del proyecto. El cliente también es el responsable de responder la “evaluación de satisfacción del cliente”, que forma parte del método de evaluación aplicado a las fábricas.

El enfoque pedagógico propuesto incluye un proceso de construcción de las fábricas de software como parte de la metodología [4]. Este proceso se organiza en siete pasos :

1. *Integración de equipos de desarrollo de software*: La clase se divide en equipos, considerando aspectos de afinidad y habilidades individuales de cada estudiante. Esto se logra a través de un cuestionario de análisis de perfiles y entrevistas, con el fin de evitar la concentración y/o insuficiencia de capacidades en un equipo. Además, el Indicador de Tipo de Myers-Briggs (MBTI, *Myers Briggs Type Indicator*) [46], se puede aplicar para identificar las características y preferencias de cada alumno.
2. *Estudio y selección de un modelo*: Los estudiantes comprenden y analizan los procesos más importantes del desarrollo de software, los cuales ayudarán a elegir y especificar formalmente el que será aplicado en su fábrica de software.
3. *Definición de la fábrica de software*: Se definen tanto el detalle del proceso de desarrollo, como el flujo de actividades, las funciones del equipo, los artefactos y las plantillas.
4. *Definición de la infraestructura de la fábrica de software*: Los estudiantes definen la infraestructura necesaria para el funcionamiento de las fábricas, tales como: el sitio web, el repositorio de plantillas y artefactos, los contratos de Acuerdos de Nivel de Servicio (SLA, *Service Level Agreement*), las métricas de desarrollo de software, el entorno de desarrollo de software y la garantía de la calidad de los productos y de procesos.
5. *Definición del modelo de negocios de la fábrica de software*: Los estudiantes definen las licencias de software, la generación de políticas sobre los derechos de autor para los productos, la creación del modelo de comunicación con los clientes, la realización de los modelos de contratos y las plantillas de propuestas comerciales y técnicas.
6. *Definición de las políticas internas de evaluación*: Esta etapa contempla la evaluación de los procesos y del capital humano como ejecutor de este proceso. Además, se definen, el contrato y la política de renovación del capital humano.
7. *Definición de políticas de evaluación del cliente*: En este paso cada fábrica tendrá que definir los criterios de satisfacción del cliente y las estrategias para mejorar la evaluación de los productos y de los procesos.

Es importante mencionar que este proceso se está ejecutando desde el año 2003 y que ha resultado eficaz como mecanismo de organización de los equipos, definición de un proceso de desarrollo de software y en la gestión de proyectos asociados a estos equipos [66]. Es evidente que para cada nuevo proyecto es necesario un nuevo sitio web, lo cual proporciona una identidad propia al proyecto, con el objetivo de controlar las actividades de desarrollo de software tanto

por los miembros del equipo como por otras partes interesadas.

Teniendo en cuenta el objetivo de “*promover una visión coherente de la Ingeniería de Software en todo el mundo*” y “*para caracterizar el contenido de la disciplina de Ingeniería de Software*” la metodología también recomienda un plan de estudios basado en el Cuerpo de Conocimientos en Ingeniería de Software de la IEEE (SWEBOK, *Software Engineering Body of Knowledge*) [3]. Dentro de este contexto, el plan de estudios está organizado en siete disciplinas basadas en SWEBOK, y donde cada una se divide en tres módulos: *básico, intermedio y profesional*. Estas disciplinas se estructuran con un contenido para apoyar la resolución de problemas de las fábricas de software, y sus módulos se intercalan para promover un desarrollo iterativo e incremental. Las disciplinas propuestas por la metodología se muestran en la Tabla 2.1.

| Disciplina | Horas-teoría | Horas-práctica | Forma de evaluación |
|---|--------------|----------------|-------------------------------|
| Las fabricas de software | 15 hrs/sem | 12 hrs/sem | Evaluación individual semanal |
| Aplicaciones y requisitos de la interfaz de usuario | 15 hrs/sem | 12 hrs/sem | |
| Gestión de proyectos | 15 hrs/sem | 12 hrs/sem | |
| Arquitectura orientada al desarrollo | 15 hrs/sem | 12 hrs/sem | |
| Ingeniería de reuso | 15 hrs/sem | 12 hrs/sem | |
| Tecnología de la interoperabilidad del sistema | 15 hrs/sem | 12 hrs/sem | |
| Verificación y validación de sistemas | 15 hrs/sem | 12 hrs/sem | |

Tabla 2.1: Asignación de tiempo por disciplina.

El programa en general se ha diseñado con módulos semanales de cada disciplina, donde se destinan 15 horas de clases, 12 horas de prácticas guiadas y una evaluación individual al final de la semana. Las horas-práctica se utilizan para ayudar a los estudiantes a poner en práctica la teoría presentada durante las clases, resolviendo los problemas reales que se presentan en las fábricas de software. Durante todo el proceso existen tutores para dar apoyo en la ejecución de estas actividades.

Los proyectos de software desarrollados bajo este modelo deben ser demandados por un cliente real. Por lo que la metodología define los siguientes criterios para apoyar en la definición de este tipo de proyectos:

- Los problemas deben tener conceptos y principios relacionados con el dominio de contenido. Así, el proceso comienza con la identificación de conceptos primarios o principios básicos que un estudiante debe aprender.
- Los problemas deben ser reales, con sus respectivas preocupaciones y solicitudes. Cada proyecto es presentado por los clientes y debe ser compatible con una serie de requisitos,

relacionados con los temas: innovación, importancia para la industria, modelo de negocios y procesos de desarrollo.

- Los clientes necesitan motivar a los estudiantes e involucrarlos en el problema. Es importante destacar que, si los estudiantes están comprometidos con un problema real, entonces el problema existe. Los alumnos perciben al problema como un caso de gran relevancia [60].

Dentro de este ámbito, el seguimiento de las fábricas de software, esta planeado por el Monitor SWF con el apoyo de Moodle [45] (que es un recurso gratuito de código abierto sobre sistemas de gestión, diseñado para apoyar a los educadores). Es importante destacar que la metodología incluye evaluaciones individuales y colectivas. Las primeras se centran en los conceptos y fundamentos aprendidos de los contenidos y disciplinas llevadas a cabo por el tutor, mientras que las colectivas se enfocan a obtener el resultado global de cada fábrica de software y es llevada a cabo por el Monitor SWF.

2.1.3. Resultados

Esta metodología se está aplicando en un curso de Maestría en Ingeniería de Software administrado por el Centro de Estudios y Sistemas Avanzados de Recife (CESAR, *Centro de Estudos e Sistemas Avancados do Recife*), un Instituto Brasileño de Investigación con experiencia en el desarrollo innovador de software desde agosto del 2007. El objetivo principal de este curso es formar ingenieros de software con una amplia visión sobre el ciclo de vida del software y todos los recursos involucrados (personas, procesos e infraestructura). Además, el curso ayuda a entender cómo aplicar los conocimientos adquiridos para resolver problemas reales y complejos utilizando formas innovadoras. El curso dura un año, donde los primeros seis meses de clases se centran en las disciplinas y el trabajo de las fábricas de software, los siguientes seis meses se dedican a la investigación y desarrollo de proyectos, con un informe basado en las contribuciones individuales de cada estudiante.

El primer curso en el que se probó estaba compuesto por 19 estudiantes, con una edad promedio de 27 años y profesionales en la mayoría de los casos: ingenieros de pruebas e ingenieros de calidad. Esta evaluación comprendió de agosto del 2007 a febrero del 2008, durante la primera etapa del programa de la Maestría. En esta clase, se crearon cuatro fábricas de software. La segunda clase fue más pequeña que la primera, ya que estaba compuesta por 10 estudiantes, con una edad promedio de 30 años donde sus habilidades estaban relacionadas principalmente con las de ingenieros de software y jefes de proyecto. El segundo grupo tenía un mayor nivel de madurez profesional que el primero por lo que los estudiantes fueron distribuidos en dos fábricas de software. Esta evaluación se realizó de marzo a septiembre del 2008.

De acuerdo con Do Santos et al., durante la ejecución de las fábricas se presentaron algunas dificultades, en particular con el primer curso. Algunas se generan por las características del proceso de desarrollo de software relacionadas con una extensa y descriptiva producción de artefactos que se concentra en las primeras etapas del ciclo de vida (como el plan del proyecto,

documentos de requisitos, diseño de proyectos, etc.), dado que éstas causaron cierto temor en los estudiantes que deseaban ver un sistema funcionando lo más rápido posible. Para minimizar este problema, fue modificada la secuencia de los módulos de las disciplinas para el primer curso, incluyendo módulos más relacionados con el paso de codificación, tales como la Arquitectura Orientada al Desarrollo e Ingeniería de Reuso.

Por otro lado, a diferencia de la primera clase, la segunda clase realmente incluyó al cliente en el proceso de desarrollo de software. Más allá de las especificaciones del proyecto, cada fábrica mantuvo reuniones frecuentes con los clientes para validar los requisitos, las interfaces y los cambios en los proyectos. Sin embargo, en las primeras etapas de las fábricas, se detectaron algunos otros problemas relacionados especialmente con la fase de requisitos, en la que los estudiantes suelen tener una visión superficial del problema y se centran en las posibles soluciones en lugar de entender las implicaciones de muchos de los requisitos del problema. En este contexto, Do Santos et al. recomiendan que la metodología sea enfocada a estudiantes de posgrado, dada su experiencia en el desarrollo de software (de 1-2 años en general) y su nivel de madurez profesional en el trabajo colaborativo.

2.2. IT Spiral: Un caso de estudio sobre la educación escalable de la Ingeniería de Software.

De acuerdo a Barker e Inoue [6], la educación escalable de la Ingeniería de Software puede utilizar métodos desarrollados en el banco de pruebas de la Iniciativa del Programa de Especialistas en TI para el Aprendizaje Avanzado basado en la Realidad (IT Spiral, *IT Specialist Program Initiative for Reality-based Advanced Learning*). Bajo esta premisa, el uso de una biblioteca DVD de conferencias y presentaciones para los tópicos básicos y avanzados de la Ingeniería de Software permitiría un aumento rápido y eficiente del número de estudiantes. Las sesiones prácticas intensivas impulsan la participación de los socios de la industria para permitir el trabajo eficaz con los estudiantes de tal manera que se aumente rápidamente en escala.

2.2.1. Objetivo

IT Spiral es un proyecto de colaboración que se realiza entre 9 Universidades y 4 empresas en Japón, con el propósito de desarrollar un diseño curricular común para la enseñanza de la Ingeniería de Software utilizando las fortalezas de las Universidades y la colaboración de las empresas. Esta colaboración combina prácticas individuales en Universidades con sesiones intensivas proporcionadas por la gente de la industria. El objetivo es desarrollar técnicas avanzadas de TI en estudiantes de nivel superior. El proyecto combina temas avanzados y les da un enfoque práctico, que sea útil en el desarrollo de ingenieros de software de clase mundial. El personal involucrado en la educación de la Ingeniería de Software puede utilizar métodos desarrollados en un banco de pruebas, realizado por especialistas y basados en la realidad del aprendizaje avanzado que contempla IT Spiral.

2.2.2. Descripción

IT Spiral es un nuevo enfoque para la enseñanza de la Ingeniería de Software en Japón, que utiliza la combinación de la experiencia y el conocimiento de las Universidades y la industria a través de videoconferencias proporcionadas por las Universidades, ejercicios en equipo y la instrucción práctica bajo la dirección de las empresas participantes, lo cual refuerza el conocimiento de la vida laboral aplicado a la enseñanza. Los objetivos del proyecto incluyen el desarrollo de técnicas avanzadas de TI por los alumnos seleccionados; las Universidades apoyan con material educativo sobre la Ingeniería de Software y la cooperación de la industria proporciona un enfoque práctico a la educación en Ingeniería de Software a nivel Maestría.

La iniciativa del Ministerio de Educación, Cultura, Deportes, Ciencia y Tecnología de Japón (MEXT, por sus siglas en inglés *Ministry of Education, Culture, Sports, Science and Technology, Japan*) consistió en solicitar proyectos bajo el nombre de “Programa para especialistas líderes en TI” para ampliar la práctica en la educación de la Ingeniería de Software en el Posgrado de las Universidades en Japón y así ayudar a reducir la brecha entre la creciente demanda y la insuficiencia de elementos de calidad [5]. Un requisito fundamental para los proyectos fue la colaboración entre Universidades y la industria (ver Figura 2.1). Durante cuatro años, MEXT ha seleccionado y financiado seis proyectos de un total de 26 propuestas de diversas universidades.

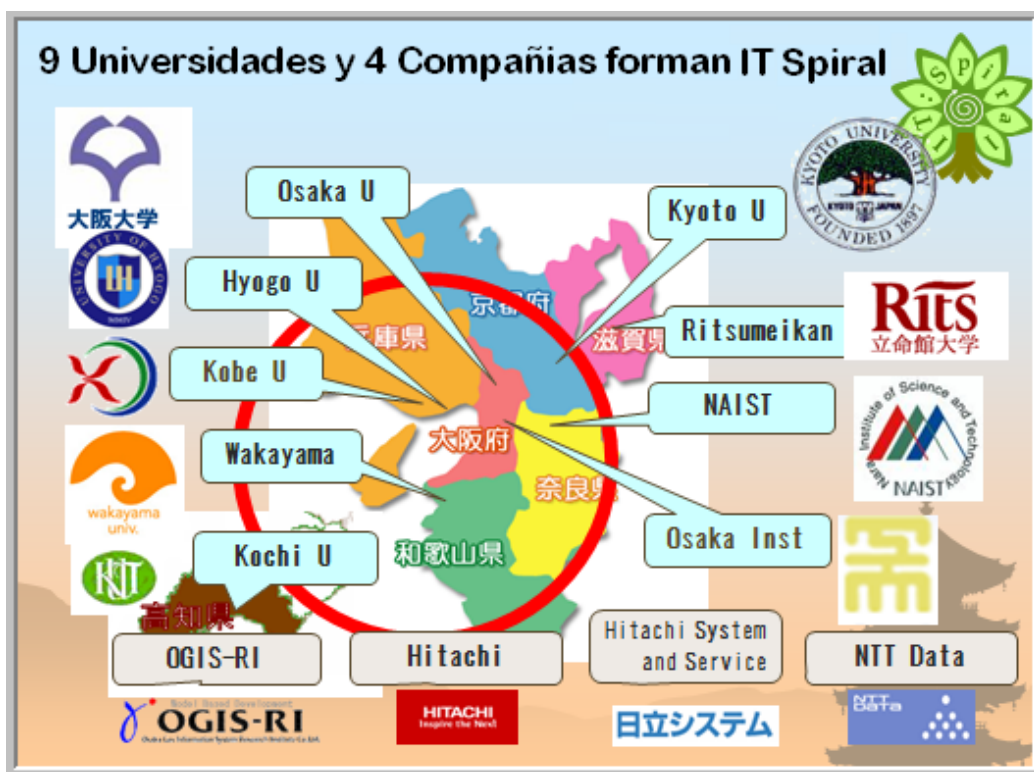


Figura 2.1: El modelo IT Spiral.

En el área de Kansai, MEXT ha seleccionado a la Universidad de Osaka por su singular

colaboración con universidades, profesionales e investigadores en Ingeniería de Software, para encabezar el proyecto. Así, a través de IT Spiral se plantea el desarrollo de ingenieros de software del más alto nivel y con una formación técnica a largo plazo, mediante la selección de estudiantes con talento y el fomento de la actividad de desarrollo de software de clase mundial, ofreciendo educación sistemática de Ingeniería de Software. Sin embargo, uno de los aspectos clave en el desarrollo colaborativo del plan de estudios fue que no es suficiente que los estudiantes sepan qué método o herramienta utilizar para el desarrollo de software, si no que IT Spiral los motive a preguntar ¿por qué utilizar ese método?, ¿por qué funciona? y si ¿existe alguna otra forma de hacerlo?.

En este cambio no ha sido fácil transferir a los nuevos dominios del pensamiento, el conocimiento, la comprensión y la aplicación (que son los niveles más bajos de la taxonomía de Bloom), el análisis, la evaluación, la síntesis y la creatividad. Sin embargo, todo esto se considera una parte necesaria del nuevo plan de estudios. De acuerdo con Barker e Inoue [6], los elementos que han ayudado a desarrollar el talento buscado por IT Spiral son:

- El proyecto incluye a los mejores educadores, investigadores y líderes de la industria en el área de Kansai.
- El proyecto es selectivo con 40 estudiantes de alta calidad cada año.
- El plan de estudios y el material fue desarrollado considerando los recursos de las nueve universidades y las cuatro empresas.

El plan de estudios se basa en tres conceptos fundamentales: inicia de forma sistemática con la enseñanza de temas fundamentales relacionados con el desarrollo de software y las habilidades básicas de la Ingeniería de Software; también contempla las nuevas tendencias, técnicas e investigación de las teorías para fomentar la capacidad de adaptación y aprendizaje continuo; por último, el plan de estudios proporciona oportunidades prácticas de desarrollo, fomentando las habilidades prácticas del estudiante. Las clases también reflejan esta división. Cada escuela de posgrado asigna e imparte temas fundamentales de Ingeniería de Software a los estudiantes en su universidad. Las facultades de todas las universidades han desarrollado una biblioteca compartida en DVD's de conferencias, material y ejercicios asociados con temas avanzados de la Ingeniería de Software, con el objetivo de que cada universidad puede utilizarla y contribuir a mejorar la enseñanza de la Ingeniería de Software. Durante el primer año se grabaron 60 videos de conferencias, de una hora cada uno, organizados en 12 volúmenes con diversos temas avanzados, tales como: Diseño UML, Metodologías ágiles, Minería de datos, Patrones de software, Software en tiempo real, Software basado en Web, Metamodelados, Repositorios de software y componentes.

Mediante esta dinámica, algunos conferencistas de diversas empresas asisten a trabajar con el personal docente universitario para colaborar con temas prácticos de la Ingeniería de Software. Estas sesiones son especialmente importantes dado que trasladan el mundo real de la Ingeniería de Software a los programas académicos, logrando motivar a los alumnos con nuevos retos [28]. IT Spiral hace uso del SWEBOK [3], ampliamente utilizado durante el diseño del plan de estudios.

Los estudiantes de las nueve Universidades asisten al Centro de Desarrollo de Nakanoshima para recibir seis horas de trabajo intenso un día por semana (ver Figura 2.2).



Figura 2.2: Sesión IT Spiral.

Las tres ramas del plan curricular de IT Spiral fomentan la participación de los estudiantes en el proceso educativo y la aplicación del conocimiento en diversas situaciones prácticas (aprendizaje activo y colaborativo). Los expertos de la industria, diversos ayudantes y profesores interactúan con los estudiantes en el rol de mentores, una interacción profesor-estudiante más cercana. Así, las variadas oportunidades de aprendizaje mejoran claramente el entorno educativo de IT Spiral enriqueciendo con ello la experiencia educativa.

2.2.3. Resultados

La medida más importante sobre la eficacia de IT Spiral es el fomento de ingenieros de software del más alto nivel que demuestran pleno dominio sobre la práctica de las técnicas de la Ingeniería de Software. Un resultado relacionado, es la promoción de la cooperación entre las Universidades y las empresas para el desarrollo de ingenieros de software de alto nivel.

De acuerdo con Barker e Inoue [6] se ha logrado:

- Desarrollar un plan curricular común de un año para los estudiantes de Maestría en Ingeniería de Software en el área de Kansai, en el cual se hace énfasis en habilidades básicas, temas avanzados y capacidades prácticas.
- Proporcionar un modelo para la Universidad y la colaboración de la industria en este tipo de educación, combinando clases locales, una biblioteca compartida con los DVD's de conferencias y sesiones comunes de prácticas intensivas dos veces por semana.
- Plantear una serie de preguntas sobre la mejor manera de compartir material, proyectos, tareas, exámenes y experiencias educativas.

| | |
|-------------|--|
| Propósito | <ul style="list-style-type: none"> ■ El objetivo principal es la formación práctica del desarrollo de software. ■ Las empresas se benefician en desarrollar y obtener talento en TI para apoyar a la industria Japonesa. ■ Los estudiantes obtienen habilidades básicas de TI, incluyendo a las tecnologías de las comunicaciones y la comprensión de la gestión. ■ Las universidades mejoran su enfoque educativo mediante la cooperación y la experimentación con nuevos métodos educativos. |
| Diferencias | <ul style="list-style-type: none"> ■ Los estudiantes pueden conocer gente de otras universidades y de la industria en sí. ■ Los conferencistas de las empresas aportan experiencia en el proceso real de desarrollo. ■ El desarrollo de software está cubierto desde las primeras necesidades hasta las pruebas. ■ Destaca la importancia de la estimación y gestión de proyectos. ■ El uso de material de video con ejercicios en equipo. |
| Resultados | <ul style="list-style-type: none"> ■ Mejora las habilidades de comunicación, el desarrollo de las prácticas de software y la actitud entre los estudiantes. ■ Una biblioteca con material de video, conferencias y cursos enfocados a proyectos reales. ■ Alto grado de satisfacción de los estudiantes, el crecimiento de la cooperación entre universitarios y la industria obtiene graduados de alta calidad. |

Tabla 2.2: Resultados de la encuesta IT Spiral.

La Tabla 2.2 muestra algunos de los resultados de una encuesta realizada a estudiantes y

profesores, así como a líderes de la industria que participan en el proyecto, donde se les preguntó sobre algunos aspectos clave del proyecto IT Spiral.

2.3. Modelo Global basado en la Competencia para fomentar las habilidades técnicas en la Educación de la Ingeniería de Software

La aparición del Desarrollo Global de Software (GSD, *Global Software Development*) puede tener un impacto diferente en la educación de la Ciencia de la Computación, dependiendo de la perspectiva en que se vea. En los E.E.U.U., por ejemplo, esta perspectiva se asocia con una menor matrícula en Ciencias de la Computación debido al temor de los estudiantes a no encontrar empleo después de graduarse. Sin embargo, en países como India y China la matrícula en Ciencias de la Computación se ha disparado y las empresas han tenido problemas para encontrar candidatos debidamente calificados.

Al mismo tiempo, sólo algunos esfuerzos educativos en todo el mundo se enfocan en exponer a los estudiantes las realidades de GSD con el objetivo de proveerles las habilidades técnicas necesarias para ser competitivos en dicho entorno. Las iniciativas existentes se emprenden en su mayoría en cursos finales de la Ingeniería de Software (p.ej., [10], [18], [21], y [56]) e intentan introducir proyectos con clientes reales o distribuir a los equipos de estudiantes en base a su localización, hora y cultura, para replicar algunos de los desafíos de GSD (p.ej., [9], [24], [26], [44], y [51]).

2.3.1. Objetivo

La propuesta aborda algunas de las dificultades que se han experimentado con los estudiantes en el pasado y que se relacionan con la entrega de productos software con calidad al final de un curso semestral en el cual la Ingeniería de Software es tomada por primera vez, mientras que un proyecto final se desarrolla de forma concurrente. De acuerdo con Gotel et al. [38], esta iniciativa es única dado que combina los conocimientos de estudiantes de Pregrado, Posgrado y aquellos que ya trabajan en la industria de una forma sinérgica para capitalizar las necesidades individuales de aprendizaje y el establecimiento de habilidades.

2.3.2. Descripción

De acuerdo con [8], la calidad se define comúnmente como “*la conformidad con los requisitos especificados y percibidos*”. Con los sistemas software, esta conformidad se establece en base a la aceptación o rechazo final del cliente. En un intento por ajustar las demandas de habilidades del estudiante en una sola experiencia relacionada con la Ingeniería de Software. La calidad puede ser el único atributo del proyecto que no obtenga el escrutinio que podría experimentarse en un entorno industrial real. Este es el caso, particularmente, cuando los mismos estudiantes son sus propios clientes de su trabajo y no existe expectativa alguna para entregar un sistema en funcionamiento. La iniciativa GSD propuesta por Gotel et al., intenta subsanar

dichas deficiencias proporcionando un entorno desafiante, donde los clientes y los desarrolladores se encuentran dispersos en todo el mundo y el objetivo es desarrollar software en un país en desarrollo. Así, durante cuatro años de colaboración se han propuesto, evaluado y refinado diferentes modelos de trabajo entre estudiantes e instructores para promover una mejor calidad. La apreciación de la calidad inicia con el aprendizaje de los estudiantes para diseñar sistemas de terceros en lugar de sistemas propios, por lo que las suposiciones relacionadas con los requisitos son analizadas minuciosamente y otras personas son las que toman las decisiones sobre la calidad. Este entorno de trabajo establece grandes exigencias a los estudiantes quienes perfeccionan sus habilidades en Ingeniería de Requisitos, y además adquieren las competencias sociales necesarias para obtener, comunicar y negociar los requisitos a través de la distancia, la lengua y la cultura. Además, establece demandas para la adquisición de las habilidades técnicas necesarias para diseñar, desarrollar, y cambiar los requisitos, desarrollando así el proceso de comunicación.

La Tabla 2.3 muestra la evolución del modelo durante los cuatro años con respecto al enfoque, el número de instructores y estudiantes involucrados, el número de países e instituciones involucradas, los roles de los estudiantes y el apoyo de las herramientas.

| | | 2005 | 2006 | 2007 | 2008 |
|---|----------------------|---|---|--|---|
| Enfoque | | Globalización | 2005 + Procesos de ingeniería de software, suministro en cadena | 2006 + Calidad, RFP, Tecnología, Integración, espíritu empresarial | 2007 + Implementación, Competición, Socialización |
| Número de Instituciones / Países | | 2 / 2 | 3 / 3 | 3 / 3 | 5 / 4 |
| Número de Instructores / Estudiantes Número de Desarrolladores de Sistemas de Software | | 3 / 32 2 versiones del mismo software, 3 versiones de otro software (sin competencia) | 4 / 33 3 diferentes software (sin competencia) | 4 / 34 2 versiones del mismo software (sin competencia) | 6 / 60 5 versiones del mismo software (competencia) |
| Clientes Desarrolladores | | Camboya EE.UU (UG) | Camboya EE.UU (UG) India (G) | Camboya EE.UU (UG) India (G) | Camboya Camboya (G); India (G); EE.UU (UG) (2 equipos); Tailandia (UG) |
| Subcontratistas | | | India (G) | India (G) | |
| Entrenadores del Cliente Entrenadores de Desarrollador | | | | EE.UU (G) Solo para desarrolladores de EE.UU | EE.UU (G) – para todos los equipos de desarrollo |
| Auditores | | | | EE.UU (G) Solo para desarrolladores de EE.UU | EE.UU (G / industria) – para todos los equipo de desarrollo Tailandia (UG) (alumnos) |
| Equipo de Socialización | | | | | Camboya (UG) |
| Tecnologías | Técnico | Eclipse IDE con JUnit, MySQL | Eclipse IDE con JUnit y CVS, MySQL | Eclipse IDE con JUnit, MySQL y java.net para el seguimiento de problemas | Eclipse IDE con JUnit y subversiones (Netbeans - India, Visual Studio - Tailandia), DB designer para bases de datos, MySQL, java.net para el seguimiento de problemas |
| | Comunicación | Yahoo IM, grupos Yahoo, sitios web, blogs | Yahoo IM, grupos Yahoo, sitios web, blogs | Yahoo IM, grupos Yahoo, sitios web, blogs, videos | Yahoo IM, grupos de Yahoo, wikis, guión para la captura de pantalla de video |
| | Socialización | Yahoo IM, intercambio de regalos | Yahoo IM, intercambio de regalos | Yahoo IM, intercambio de regalos, videos | Yahoo IM, ejercicios de socialización, Second Life |
| | Gestión de proyectos | Sitios web, blogs | Sitios web, blogs | Wikis, blogs | Wikis, Google calendarios, software de zona horaria |

Tabla 2.3: Evolución del modelo GSD en 4 años de colaboración.

La visión de este proyecto siempre ha sido la de desarrollar un modelo mediante el cual los estudiantes con habilidades complementarias y necesidades de aprendizaje se reúnan para tra-

bajar y apoyarse mutuamente para hacer esto posible. Los proyectos GSD regularmente tienen una duración de 14 a 15 semanas y se realizan cada primavera. La configuración del proyecto se determina en base a la investigación y los objetivos específicos, destacando las necesidades de la industria y los problemas con proyectos de estudiantes anteriores, en conjunto con el número y el nivel de estudiantes que participarán y sus necesidades de aprendizaje.

El Patrón de la Tabla 2.3 permite observar que la cantidad de trabajo requerido por los instructores participantes para organizar y ejecutar los proyectos GSD puede ser sustancial. Es decir, si el número de proyectos se incrementa, la carga de trabajo aumenta también. Gotel et al., han determinado que es más productivo y gratificante concentrarse en realizar un proyecto, en lugar de hacer malabares con una serie de proyectos y obtener resultados parciales. En el modelo del 2007 se examinó la forma de integrar a los estudiantes de Pregrado y Posgrado para el trabajo conjunto. Esta versión se enfocó en proporcionar tutores para los equipos de desarrollo de E.E.U.U., pero se descuidaron aspectos como: la necesidad de apoyar a los clientes en sus actividades relacionadas con los requisitos, el factor del tiempo necesario para responder a los comentarios de la auditoría en el cronograma del proyecto, y el establecimiento de uno de los equipos en un rol de subcontratación que condujo a una percepción de “no pertenencia”.

Estos problemas se resolvieron en la versión 2008 del modelo que fue diseñada para motivar a los estudiantes a distinguir cuatro puntos de vista diferentes de calidad:

- La visión subjetiva de los propios desarrolladores,
- El punto de vista de un entrenador que se especializa en prácticas de calidad y que actúa como observador interno de calidad en el equipo de desarrollo,
- La visión más objetiva de los auditores de la calidad que actúan de forma externa, y
- La percepción más crítica del cliente.

La configuración del modelo GSD para el año 2008 se ilustra en la Figura 2.3, y las funciones y responsabilidades son descritas a continuación:

- ***Equipo del Cliente - Estudiantes de Pregrado del Instituto Tecnológico de Camboya.*** Un equipo de cinco estudiantes de tiempo completo de Ciencias de la Computación es propietario de los requisitos, además de gestionarlos. Este equipo solicita sistemas candidatos para cubrir sus requisitos con cinco equipos de desarrollo, prueba cada sistema, selecciona el de la calidad más alta y lo entrega.
- ***Equipos de Desarrollo - Cinco equipos de cuatro países diferentes.*** Los equipos están formados de cuatro a seis estudiantes de tiempo completo de la India, Tailandia, Camboya y dos sitios en los E.E.U.U. (Nueva York y Pleasantville, específicamente). Todos los equipos están formados por estudiantes de Pregrado de nivel junior/senior, a excepción de los estudiantes de la India (Maestría en Aplicaciones Computacionales). Cada equipo de desarrollo es patrocinado por un cliente y apoyado por un entrenador de calidad y un equipo de auditores.

- ***Entrenadores de Calidad – Estudiantes de Posgrado de la Universidad de Pace, E.E.U.U.***
Cinco estudiantes de tiempo parcial controlan al equipo del cliente, ayudándole con la línea base de requisitos, la gestión de los cambios, la confianza de que funcionan como responsables del proyecto, la formulación de los planes de prueba y la guía en el proceso de selección. Cinco estudiantes de tiempo parcial también controlan a los equipos de desarrollo para inyectar calidad en su trabajo y asegurar que ocurre la comunicación cliente/desarrollador. Los entrenadores son una mezcla de estudiantes de Ciencias de la Computación, Diseño de Software e Ingeniería de Software.
- ***Audidores de Calidad – Estudiantes de Posgrado de la Universidad de Pace, E.E.U.U.***
Quince auditores (cinco equipos de tres elementos) revisan de forma independiente los procesos y los productos de cada equipo de desarrollo, proporcionando información para asegurar y mejorar la calidad. Estos también actúan como auditores del cliente, revisando el documento de los requerimientos en períodos regulares. Uno de los estudiantes funge como responsable del Aseguramiento de la Calidad del Proceso y del Producto para coordinar esfuerzos. Todos estos estudiantes por un periodo de tiempo se especializan en Diseño de Software e Ingeniería.
- ***Equipo de Socialización - Estudiantes de Posgrado de la Universidad Real de Phnom Penh, Camboya.*** Un equipo de dos estudiantes de Ciencias de la Computación es el encargado de hacer todo lo posible para introducir a uno de los equipos de E.E.U.U. a la vida y cultura de Camboya, para examinar el impacto del aumento de la socialización en las relaciones cliente/desarrollador.
- ***Alumnos para el Aseguramiento de Calidad – Estudiantes de Pregrado de la Universidad de Mahidol, Tailandia.*** Cuatro estudiantes aprenden el rol y el valor de la calidad trabajando con entrenadores y auditores de Tailandia.

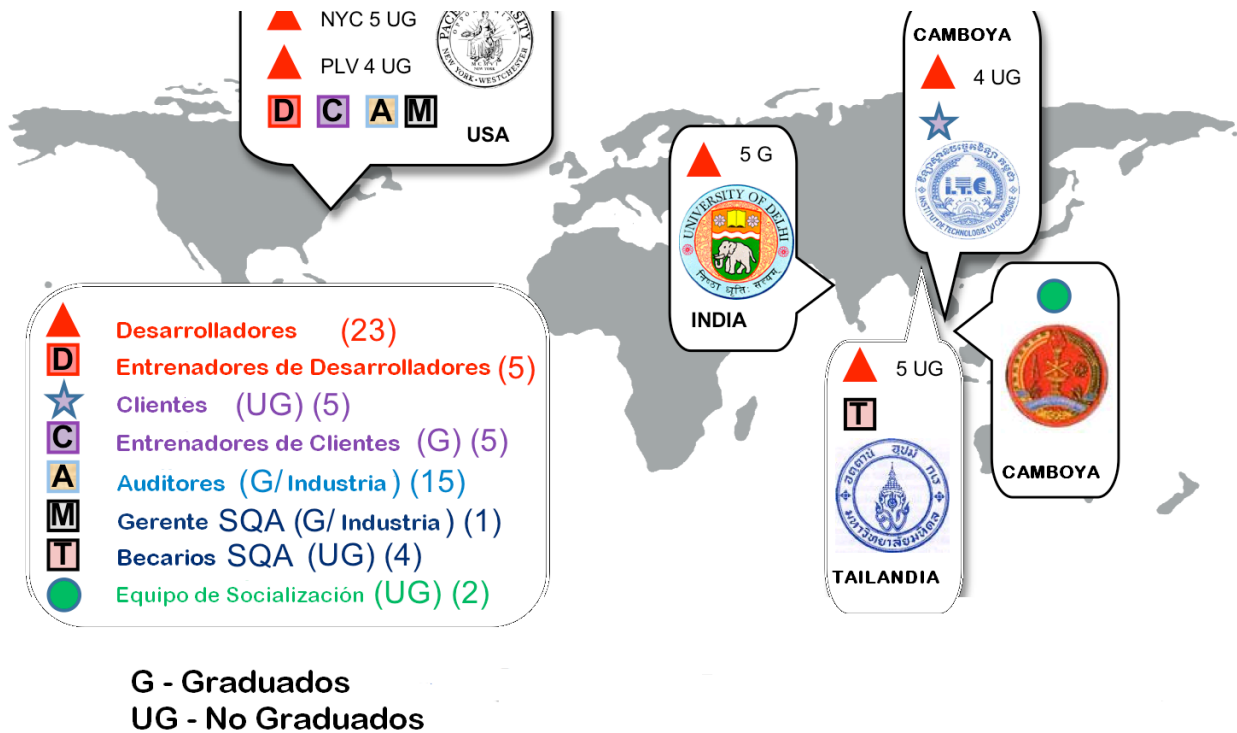


Figura 2.3: El Modelo GSD en el 2008.

Al final del período de desarrollo de 14 semanas, el equipo del lado del cliente selecciona el software ganador mediante el conteo de requisitos comprobables, que cumplan con las especificaciones originales de acuerdo a una escala de calificación. El software seleccionado es entregado al equipo de Camboya para su implementación. A lo largo del proyecto, los datos fueron recogidos mediante encuestas semanales para controlar el nivel de estrés de los estudiantes y su equipo de comunicación. También se completan cuestionarios post-proyecto.

2.3.3. Resultados

Los clientes de Camboya aceptaron tres de los sistemas de software desarrollados bajo el modelo GSD en su versión 2008. Uno de estos sistemas es el denominado MultiLib, un software de librería para el Departamento de Ciencias de la Computación del Instituto Tecnológico de Camboya. El factor decisivo fueron los requisitos no funcionales que los clientes consideraban más importantes para apoyar la implementación y el mantenimiento. La continua interacción cliente-desarrollador para cada equipo de desarrollo, llevó a la aclaración y reformulación de los requisitos. Con ello existe un gran beneficio desde múltiples perspectivas. El control continuo al que fue expuesto el trabajo de los desarrolladores y los múltiples niveles de pruebas, llevó a que los productos finales cumplieran con lo especificado.

La versión final del documento sobre las necesidades de MultiLib contenía 71 requisitos: 22 con prioridad baja, 21 con prioridad media y 28 con prioridad alta. En función de la importancia para el cliente, se definió una escala de puntuaciones (entre 1 y 10) asociada con cada uno de los requisitos y que fue utilizada para calificar cada uno de los sistemas de software desarrollados. La Tabla 2.4 muestra el número de requisitos completado como una forma de medir la multiplicidad de percepciones de la calidad en este proyecto. Además, muestra el número de requisitos que los desarrolladores intentaron implementar por iteración, junto con su evaluación en cuanto a estado después de la prueba. También se muestra el estado de los requisitos desarrollados según lo determinado por el entrenador del equipo de desarrollo, sus auditores y el cliente.

| | | Camboya | India | Tailandia | EE.UU. NYC | EE.UU. PLV |
|---|--|---|--|-----------------------------------|---|---|
| Requisitos dirigidos por iteración | It. 1 | 4 | 11 | 27 | 51 | 29 |
| | It. 2 | 42 | 38 | 11 + 27 (4 its.) | 5 | 4 |
| | It. 3 | 25 | 17 | 6 | 8 | 2 |
| | Total | 71 | 66 | 71 | 64 | 35 |
| Estado según la percepción de los desarrolladores en la entrega final (las cifras no siempre coinciden con el número de la demanda dirigida) | | 24 Completado 43 Parcialmente 2 No Completado | 66 Completado 2 Parcialmente 3 No Completado | 68 Completado 3 No Completado | 64 Completado 7 No Completado | 32 Completado 1 Parcialmente 2 No Completado |
| Pruebas y estado según la percepción de otras partes (de nuevo tome en cuenta que las cifras no siempre están de acuerdo a todas las partes) | Por Cliente | 8 Completado 63 No Completado | 60 Completado 11 No Completado | 66 Completado 5 No Completado | 52 Completado 19 No Completado | 31 Completado 40 No Completado |
| | Por Entrenadores de Clientes | 13 Completado 58 No Completado | 62 Completado 2 Parcialmente 7 No Completado | 65 Completado 3 No Completado | 58 Completado 1 Parcialmente 12 No Completado | 26 Completado 2 Parcialmente 43 No Completado |
| | Por Entrenadores de Desarrolladores | 23 Parcialmente | 69 Completado 2 No Completado | 69 Completado | 45 Completado 5 Parcialmente 7 No Completado | 30 Completado 29 No Completado |
| | Por Auditores | Características 9 Completado 15 Parcialmente 47 No Completado | 60 Completado 5 Parcialmente 1 No Completado | 60 Completado 10 No Completado | 50 Completado 15 Parcialmente 5 No Completado | 32 Completado 9 Parcialmente 30 No Completado |
| | Estado | ROJO | VERDE | VERDE | VERDE | AMARILLO |

Tabla 2.4: Múltiples percepciones sobre la calidad.

Gotel et al. comprobaron que colocar a estudiantes con antecedentes diferentes en una situación de competencia tiene sus ventajas y desventajas. Los estudiantes más fuertes son motivados a ganar y pueden sobresalir, mientras que los estudiantes más débiles pueden sentirse abrumados ante la primera señal de problemas. Este proyecto examinó también los niveles de estrés reportados por los estudiantes. Los resultados obtenidos se presentan en la Figura 2.4.



Figura 2.4: Monitor de estrés del estudiante.

En general, los equipos con mayor carga de estrés fueron los dos equipos de desarrollo de la Universidad de Pace. Se determinó que los roles del instructor y de los equipos de apoyo son fundamentales para alertar sobre los problemas y así proporcionar la asistencia necesaria. Por otro lado, otro factor monitoreado fue la comunicación entre las distintas partes involucradas en el modelo (ver Tabla 2.5).

| | Clientes | Ent. Cliente | Des. Camboya | Des. India | Des. Tailandia | Des. NY | Des. PLV | Ent. Desarrolladores | Aud. | Equipo Socialización | Profes. |
|------------------------------|----------|--------------|--------------|------------|----------------|---------|----------|----------------------|------|----------------------|---------|
| Clientes | ++++ | ++++ | +++ | +++ | +++ | +++ | ++++ | + | + | N/A | + |
| Entrenadores Cliente | | ++++ | + | +++ | + | + | + | + | + | N/A | + |
| Desarrolladores Camboya | | | ++++ | N/A | N/A | N/A | N/A | +++ | +++ | N/A | + |
| Desarrolladores India | | | | ++++ | N/A | N/A | N/A | ++ | ++++ | N/A | + |
| Desarrolladores Tailandia | | | | | ++++ | N/A | N/A | ++++ | ++ | N/A | + |
| Desarrolladores NY | | | | | | ++++ | N/A | +++ | + | N/A | + |
| Desarrolladores PLV | | | | | | | +++ | ++++ | + | ++++ | + |
| Entrenadores Desarrolladores | | | | | | | | + | ++ | N/A | + |
| Auditores | | | | | | | | | ++++ | N/A | + |
| Equipo Socialización | | | | | | | | | | N/A | N/A |
| Profesores | | | | | | | | | | N/A | +++ |

Tabla 2.5: Comunicación entre roles(donde + representa intensidad).

Es claro que la calidad de los requisitos fue más imponente que en cualquier proyecto anterior de la iniciativa GSD y, aunque no es posible afirmar que la correlación de los niveles de comunicación entre los clientes y sus entrenadores es causal, Gotel et al. sugieren un indicativo (+++++) para calificarla. La única dificultad importante fue la determinación de los requisitos no funcionales, un tema que salió a la luz durante la selección. Del mismo modo, los entrenadores ayudaron a sus equipos con la planificación de la calidad, las actividades de las pruebas, etc., e incluso proporcionando tutoriales sobre temas que se analizaron en sesiones de clase. El principal problema técnico que enfrentaron tanto los entrenadores de desarrollo como los auditores fue hacer que los estudiantes establecieran trazabilidad para apoyar en la gestión del cambio.

Después de la selección se presentaron pocos problemas durante la fase de implementación del proyecto. Los principales problemas estuvieron relacionados con la incompatibilidad que surgió al usar Windows para el desarrollo y la solicitud de usar Linux para la entrega, así como la falta de comentarios en el código, lo que retrasó la corrección de errores. Algunos errores que se encontraron en el software estaban relacionados con el idioma y con supuestas fallas erróneas, como el uso de los códigos postales en los datos de las cuentas (estos no existen en Camboya) y el uso de imágenes que no son de Camboya.

2.4. Diseño e Implementación de un Curso Práctico de Ingeniería de Software

Un curso de Ingeniería de Software regularmente forma parte de las materias de tercer o cuarto año que conforman el programa de estudios de cualquier Universidad relacionada con carreras enfocadas a las Ciencias de la Computación. La Ingeniería de Software es una disciplina de la Ingeniería, cuyo principio general recae sobre las tecnologías para el desarrollo y mantenimiento de software [39],[59]. Así, el contenido de los cursos para la enseñanza de la teoría de los Sistemas de Información se centra en los principios, conceptos y métodos de la Ingeniería de Software, incluyendo muchos aspectos de los métodos, la tecnología, las herramientas y la gestión.

Sin embargo, únicamente con el estudio teórico, los estudiantes no pueden entender y captar la esencia principal de un curso y por ende aplicar sus conocimientos al proceso de desarrollo de software. Es decir, la eficacia de la enseñanza se ha debilitado. De acuerdo con [12], las Directrices para los Programas de Pregrado en Ingeniería de Software hacen hincapié en que la educación de todos los estudiantes de la Ingeniería de Software debe incluir las experiencias de los estudiantes con la práctica profesional. Por lo tanto, es necesario diseñar e implementar un curso que permita el seguimiento práctico después de los cursos teóricos de Ingeniería de Software.

2.4.1. Objetivo

La propuesta de Wang et al., [70] presenta los pasos necesarios para el diseño e implementación de este curso práctico sobre Ingeniería de Software. El objetivo se enfoca en mejorar el efecto de la enseñanza combinando la teoría de la Ingeniería de Software con las prácticas de campo. De esta forma los estudiantes son provistos de experiencias reales sobre cómo desarrollar

los proyectos de software y cómo mejorar la calidad de los mismos.

2.4.2. Descripción

El curso práctico está diseñado en base a proyectos y se compone de tres partes interrelacionadas:

- Explicar el proceso de software, su gestión y el aseguramiento de la calidad,
- Planificar el proyecto y asignar los roles, e
- Implementar el curso

En la primer parte los profesores resumen el Proceso Unificado de Rational (RUP, *Rational Unified Process*) con el objetivo de que los estudiantes entiendan los conceptos de la gestión del proceso de desarrollo. Cabe mencionar que RUP es un estándar comprensible para el proceso de desarrollo, sin embargo también es complejo. Las organizaciones pueden adaptar los pasos del proceso de desarrollo de acuerdo a sus diferentes proyectos. Sin embargo, es importante que cada una establezca correctamente su proceso de desarrollo. En este sentido, los estudiantes no son capaces de llevar a cabo el desarrollo iterativo debido a la limitación del tiempo para realizar las prácticas de un curso. Sin embargo, la comprensión y el dominio de RUP es muy útil para orientarlos a organizar y gestionar eficazmente el proceso de desarrollo de software. Durante el curso propuesto, son explicados los métodos para el Aseguramiento de la Calidad del Software y la Gestión de la Configuración [52]. Además, también se proporciona una breve introducción al Lenguaje Unificado de Modelado (UML, *Unified Modeling Language*), herramientas UML, y otras herramientas tales como MS Project, Visual SourceSafe y métodos como el Análisis Orientado a Objetos y los Métodos de Diseño, que han sido cubiertos y revisados en un curso teórico de Ingeniería de Software. Se requiere de 28 horas de enseñanza para cubrir estos conocimientos:

- Introducción a RUP (4 horas),
- Obtención de los requisitos, especificación y casos de uso (5 horas),
- Métodos y pruebas para el aseguramiento de la calidad del software (7 horas),
- Diseño orientado a objetos (4 horas),
- Introducción a UML, uso de herramientas (Rational Rose, MS Project, VSS) (6 horas),
- Gestión de la configuración (2 horas),

La segunda parte del curso, relacionada con la planificación del proyecto y la asignación de roles, asigna un proyecto pequeño a los estudiantes. La carga de trabajo del proyecto debe coincidir con el límite de tiempo del período de prácticas del curso. Algunos proyectos asignados son: un sistema de registro para las conferencias, sistema de simulación de cajeros automáticos, sistema de seguimiento de medallas olímpicas, y sistema de selección de cursos. Después de establecer un proyecto, a los estudiantes se les permite organizarse libremente por equipos de entre 5 y 7 alumnos. La capacidad de colaboración de trabajo en equipo de los estudiantes es inspeccionada por cada grupo. Los miembros de cada equipo deben incluir jefes de proyecto, analistas de

sistemas, diseñadores, implementadores, testers, personal para el control de la calidad, gestor de la configuración, y diseñadores de interfaz. Un estudiante puede tener múltiples roles (siempre y cuando sean compatibles), pero es posible que dos roles no puedan ser desempeñados al mismo tiempo (p.e., un estudiante no puede tener dos roles incompatibles, uno como implementador y otro como tester). Cada rol tiene su propia responsabilidad:

- El jefe del proyecto está a cargo de la gestión diaria del proyecto y de la planificación del mismo.
- El analista de sistemas es responsable de documentar la especificación de los requisitos del software, la cual incluye la especificación de los casos de uso y las especificaciones suplementarias.
- El diseñador es responsable de construir los modelos de análisis y diseño que incluyen a los subsistemas, clases, interfaces e implementación de los casos de uso.
- El implementador es responsable de la codificación del programa.
- El tester es responsable de organizar los planes de las pruebas, diseñar los casos de prueba, ejecutar las pruebas y evaluar los resultados de las pruebas.
- El gestor de la configuración es responsable de desarrollar los planes de gestión de la configuración, gestionar la base de datos de configuración y construir la línea base.
- El diseñador de interfaces es responsable del desarrollo de las interfaces y manuales de usuario.

La última parte, o bien la implementación del curso, se divide en tres pasos:

1. La obtención de los requisitos: Cada equipo toma una hora para discutir con los usuarios y los clientes (los maestros interpretan los roles de clientes). Los estudiantes formulan las preguntas de muchas maneras para entender lo que los usuarios y clientes quieren y necesitan [39], [27]. En base a los resultados de la discusión, la definición de los requisitos y las especificaciones son documentados.
2. Entrega de documentos: Los estudiantes tienen cuatro semanas para desarrollar el producto. Algunos documentos y artefactos que son requeridos cada semana son revisados por los maestros de la siguiente forma:
 - a. El jefe del producto regularmente está muy ocupado durante la primera semana. Él debe establecer el plan y el calendario del proyecto, y comenzar a seguir el progreso del proyecto de acuerdo con este calendario. Los miembros del equipo también pueden aprender estas tareas de forma paralela. El analista de sistemas documenta la especificación de los requisitos del software. Los documentos presentados son los siguientes:
 - a) Plan del proyecto: que incluye la definición del proyecto, un plan de asignación de personal, un plan para la gestión de riesgos y un plan de medición.

- b) Cronograma del proyecto: que incluye tareas y subtareas, las responsabilidades de los miembros del equipo, los tiempos estimados, la interdependencia de las tareas, y las fechas de inicio y finalización.
- c) Especificación de requisitos: que incluye los casos de uso y las especificaciones suplementarias.

b. Otros miembros del equipo deben de comenzar a trabajar en la segunda semana de acuerdo con el plan y el calendario del proyecto. Los documentos presentados son los siguientes:

- a) Plan de configuración: que incluye la estructura del directorio de productos, el método de identificación de la configuración y la línea base.
- b) Documentos del análisis y el diseño: que incluye los subsistemas, los diagramas de clase y los diagramas de secuencia.
- c) Plan de pruebas.
- d) Interfaz de usuario.

c. al llegar a la tercera semana el desarrollo del proyecto se encuentra en una etapa intermedia. En esta semana serán requeridos los documentos para las pruebas. Los documentos presentados son los siguientes:

- a) Casos de prueba: que incluye la descripción de los casos de prueba, las entradas y las salidas esperadas.

d. El proyecto se encuentra a punto de culminar en la cuarta semana. El reporte de pruebas deberá ser entregado. Los documentos presentados son los siguientes:

- a) Registro de clientes: que incluye problemas a discusión y decisiones resueltas.
- b) Informe de seguimiento de defectos: que incluye la descripción de los errores, la razón y la solución propuesta, así como el estado final.
- c) Manual de usuario
- d) Base de datos de la configuración.

3. Demostración del producto: El producto se exhibe por los estudiantes al final de la cuarta semana. Los maestros comprueban si éste coincide con los requisitos de los usuarios y de los clientes. Se demuestran también la fiabilidad y facilidad de uso del producto. En este paso, se requiere también que los estudiantes entreguen los informes finales que respondan a las siguientes preguntas:

- a. ¿Qué problemas encontraron durante el proyecto?, ¿cómo los resolvieron?, ¿utilizaron los conocimientos aprendidos en clase?
- b. ¿Consideran que su proyecto será un éxito o un fracaso?, ¿por qué?

- c. ¿Qué opinan de la calidad de la versión final de su software?, ¿por qué?
- d. ¿Les gusta el estilo de este curso? ¿creen que se le puede añadir algo a este curso? ¿por qué?

A través del informe final es posible obtener retroalimentación, buenas ideas y consejos de los estudiantes. Al mismo tiempo, se motiva a los estudiantes a resumir los resultados del curso. Los documentos juegan roles importantes en el desarrollo del software, dado que garantizan efectivamente la consistencia y el mantenimiento del software; por lo que es necesario mantener la coherencia entre los documentos en las diferentes fases del ciclo de vida.

2.4.3. Resultados

El objetivo de este curso práctico no consiste solamente en el desarrollo de un pequeño sistema de software de acuerdo a un ciclo de vida determinado, sino también en la orientación de los estudiantes para entender las responsabilidades de los diferentes roles y los pasos básicos en el aseguramiento de la calidad del software dentro del proceso de desarrollo. De acuerdo a [12], el SE2004 ha logrado describir 10 áreas que conforman el conocimiento para la Educación de la Ingeniería de Software: Fundamentos de la Computación, Matemáticas y Fundamentos de la Ingeniería, Prácticas Profesionales, Análisis de Software y Modelado, Diseño de Software, Verificación y Validación de Software, Evolución del Software, Proceso de Software, Calidad del Software, y Gestión del Software. Cada área de conocimiento se divide entre dos y siete unidades e indica el número mínimo de horas recomendadas para cada área y unidad. Algunas áreas del conocimiento clave y unidades referenciadas en el SE2004 han sido incluidas en el curso propuesto por Wang et al., [70], y se espera ampliar esta cobertura en cursos futuros. De acuerdo a estos investigadores, mediante este curso los estudiantes pueden obtener los siguientes resultados:

- Familiarizarse con RUP y UML,
- Entender el ciclo de vida de la Ingeniería de Software,
- Ser capaz de analizar y documentar los sistemas de software,
- Estar familiarizado con las tecnologías de análisis y obtención de requisitos,
- Ser capaz de aplicar los casos de uso,
- Entender y aplicar el análisis y diseño orientado a objetos,
- Ser capaz de diseñar e implementar sistemas de software,
- Ser capaz de diseñar e implementar bases de datos,
- Documentar un sistema de software,
- Comprender y aplicar las técnicas básicas de pruebas del software,
- Ser capaz de trabajar como individuo y como parte de un equipo para desarrollar y entregar un producto software de calidad,
- Estar familiarizado con el método de configuración.

2.5. Comparativa empírica sobre las propuestas relacionadas con la vinculación Academia-Industria

Como se puede observar, las cuatro propuestas analizadas para mejorar la enseñanza de la Ingeniería de Software están enfocadas a fortalecer los conocimientos adquiridos en las aulas a través de la vinculación entre la Universidad y la Industria. Por lo tanto, estas propuestas permiten la identificación y el establecimiento de características comunes que podrían considerarse para nuestro entorno Universitario. La comparativa empírica consiste en analizar la información recogida, separándola en dos categorías:

- La primera, relacionada con la identificación de aquellas características que son compartidas por las propuestas analizadas, y que podrían convertirse en los requerimientos base de una herramienta software que apoye a alcanzar los objetivos establecidos en esta tesis, y
- La segunda, relacionada con la definición de problemas presentes en la adopción de este tipo de propuestas en nuestro entorno académico, y que extenderían los requerimientos base con necesidades más específicas.

La Figura 2.5 muestra las características compartidas por las propuestas de Cavalcanti et al., [14], Barker e Inoue [6], Gotel et al., [38], y Huo et al., [70], analizadas en la sección anterior. Sin embargo, además de las características resaltadas en la Figura, es posible identificar particularidades que son comunes entre las propuestas, pero que se enfocan específicamente a los requerimientos de infraestructura en las Universidades.

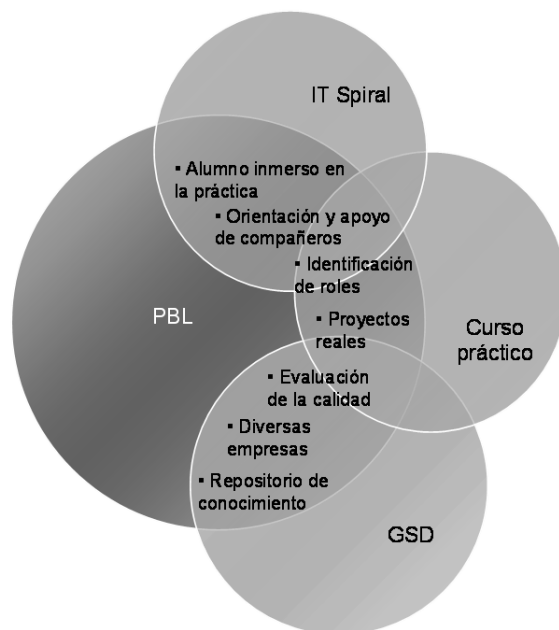


Figura 2.5: Características comunes de las propuestas analizadas.

La Tabla 2.6 resume el total de características que pueden utilizarse para diseñar una solución acorde a las condiciones de nuestro entorno Universitario.

| |
|---|
| Relacionadas con el aprendizaje: |
| <ul style="list-style-type: none"> ■ Alumnos inmersos en la práctica constante [14]. ■ Aprendizaje basado en proyectos reales con clientes reales [14]. ■ Orientación y apoyo de otros compañeros [14], aprendizaje colaborativo [6]. ■ Repositorio de conocimientos (con presentaciones de conferencias, artículos de divulgación, ejercicios, temarios, etc.) [6]. ■ Mezcla de conocimientos de diferentes niveles (Pregrado, Posgrado y trabajadores de la industria) [38]. |
| Relacionadas con el proceso software: |
| <ul style="list-style-type: none"> ■ Definición del proceso, roles, métricas para controlar los resultados [14]. ■ Definición, planificación y gestión, e implementación del proceso [70]. ■ Gestión y distribución de roles [14], [70]. ■ Enfoque en los requisitos y la calidad [38]. ■ Selección del ciclo de vida adecuado [70]. |
| Relacionadas con la interacción: |
| <ul style="list-style-type: none"> ■ Normalización de buenas prácticas de trabajo [14]. ■ Equipos de entre 5 a 7 miembros [14]. ■ Incorporación de diversas Universidades y empresas [6]. ■ Apoyo constante al cliente [38]. ■ Comunicación entre roles [14], [70]. |
| Relacionadas con la evaluación: |
| <ul style="list-style-type: none"> ■ Evaluaciones de los alumnos en base a la selección del software ganador considerando el conteo de requisitos comprobables [38]. ■ Evaluaciones de los clientes basadas en la calidad del producto [70]. |

Tabla 2.6: Características para la enseñanza de la Ingeniería de Software.

Estas características han sido clasificadas en las categorías de:

- Relacionadas con el aprendizaje, que engloba aquellos factores vinculados con la mejora

en la impartición de la Ingeniería de Software y que representa un elemento crucial para el modelo educativo.

- Relacionadas con el proceso software, que busca homogenizar el conocimiento teórico aprendido en las aulas en relación al establecimiento de un proceso eficiente para el desarrollo de software.
- Relacionadas con la interacción, que busca establecer canales efectivos de comunicación entre el equipo de desarrollo y los demás interesados.
- Relacionadas con la evaluación, que pretende determinar la efectividad en el aprendizaje a través de auditorías de calidad al interior del equipo de desarrollo, y evaluaciones de satisfacción al exterior del equipo con los clientes reales.

Así, se hace evidente la necesidad de un modelo dinámico que permita capacitar mejor a los estudiantes de la Ingeniería de Software y que considere, por lo menos, los elementos identificados en el análisis anterior. Sin embargo, un modelo teórico que establezca qué estudiar y qué no estudiar, que indique el orden de la currícula para lograr mejores resultados, o que sea modificado continuamente para atender el avance tecnológico actual no sería de mucha ayuda.

De acuerdo a la Figura 2.6, la solución planteada en esta tesis pretende establecer un modelo dinámico que sea soportado por una herramienta de software que se adapte a la currícula del Sistema de Universidades del Estado de Oaxaca (SUNEO) y que busque una homogenización en la formación de Ingenieros de Software con las habilidades que la industria en realidad necesita. Esta solución será planteada y desarrollada formalmente en el Capítulo 3 de esta tesis.

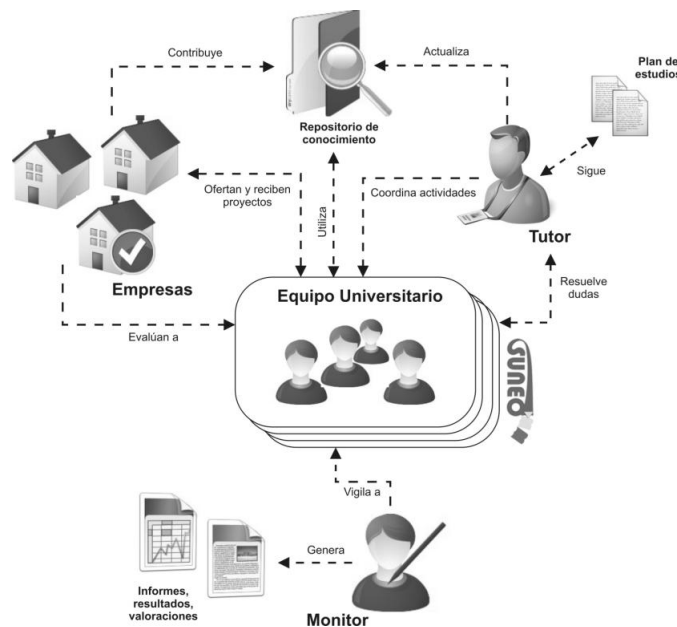


Figura 2.6: Diagrama de bloques de la solución planteada.

De esta forma, y utilizando los criterios identificados en la Tabla 2.6, los estudiantes combinarán la teoría y la práctica y sus capacidades serán calificadas por la práctica en el mundo

real. Este sistema establecerá una forma de educación basada en la Arquitectura de Aprendizaje centrada en el Modelo MCLA (Model Centered Learning Architecture), enfocándose en la formación de habilidades como objetivo final del aprendizaje vinculado (Universidad-Industria). Los estudiantes, una vez que se gradúen, sabrán desempeñar papeles críticos en el negocio y la industria.

Este modelo dinámico permitirá que las empresas establezcan necesidades de proyectos que deberán ser cubiertas por equipos universitarios, quienes harán uso de un repositorio de activos definido para consultar artículos, presentaciones de congresos, ejercicios y demás. Este repositorio será actualizado por el Tutor y recibirá contribuciones por parte de los especialistas de las empresas. En este sentido, el papel del Tutor es muy importante puesto que es este quien coordinará las actividades del equipo y resolverá dudas del proyecto. Por último, el Monitor será el responsable de vigilar el desempeño del equipo y recibirá las evaluaciones de las empresas en relación a la calidad de los proyectos.

Capítulo 3

Introducción a la Solución del Problema.

3.1. Bases del modelo dinámico para la enseñanza de la Ingeniería de Software

La introducción de un modelo mexicano de procesos, MoProSoft, en la industria software desde el 2005 tiene que cambiar la forma en que las Universidades educan a los jóvenes ingenieros de software. Los modelos como CMMI v1.2 [33], ISO 9000:2000 [58], ISO/IEC 12207 [2], e ISO/IEC 15504 [1] se han utilizado en la industria mexicana del software; destacándose el CMMI v1.1 [32] con un número considerable de empresas grandes certificadas. Sin embargo, un gran porcentaje de las empresas software en México son pequeñas empresas en las cuales la adopción de este modelo es una tarea inalcanzable (dados los recursos económicos y de personal necesarios).

Así, MoProSoft fue desarrollado para cubrir las necesidades y las expectativas de estas pequeñas empresas software y éstas están esperando contratar a ingenieros jóvenes con los conocimientos necesarios para impulsar sus objetivos de negocio.

En este contexto, se propone un modelo alternativo que se enfoca en cinco módulos que brindarán apoyo a los académicos para proporcionar el conocimiento necesario de acuerdo a la realidad de la industria mexicana de software. La Figura 3.1 muestra que estos módulos se encuentran en áreas de alto nivel que necesitan ser descompuestos a un nivel granular mayor.

Se intenta que estos módulos funcionen como herramientas comunicativas, utilicen puntos de referencia y promuevan un proceso extenso de razonamiento. De manera similar que el modelo NIIT de la India, se pretende que la combinación de la educación con la práctica sea la llave del éxito de dicho modelo.

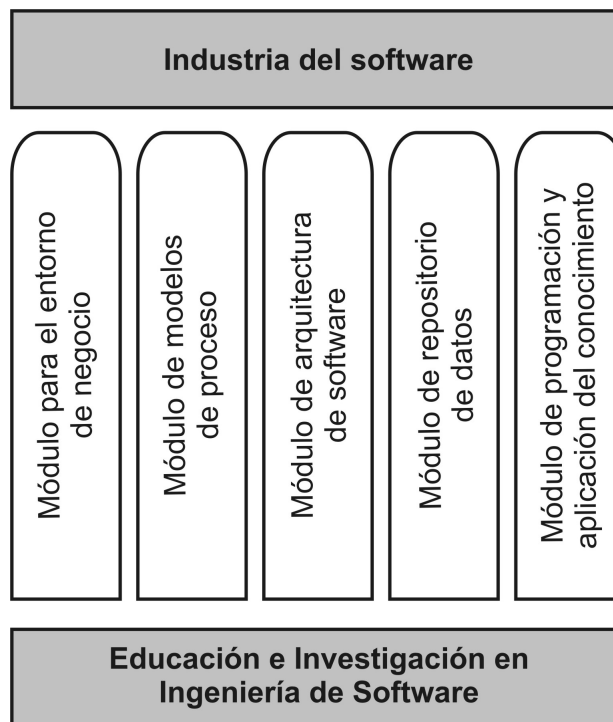


Figura 3.1: Módulos de capacitación para el modelo propuesto.

Los módulos propuestos se resumen en:

- *Módulo para el entorno de negocio*, que deberá proporcionar el conocimiento relacionado con la implantación de soluciones TI en un determinado dominio de negocio. En cualquier dominio de negocio, los factores más importantes son las reglas de negocio, los procesos de negocio, el rendimiento y la seguridad. Debido a esto, la característica más distintiva de este módulo es la combinación de la educación en clase y la cooperación con el negocio (o empresas incorporadas en el modelo).
- *Módulo de modelos de proceso*, las metodologías tradicionales de ciclos de vida para el desarrollo de software están lejos de ser útiles en la industria por si solas. Hoy en día, los negocios cambian continuamente los requisitos y exigen soluciones lo más pronto posible y con un alto nivel de calidad. De acuerdo a las necesidades de la industria Mexicana de software, los modelos como MoProSoft y TSP son cruciales para su promoción. El módulo de modelos de proceso proporciona el conocimiento relacionado con el aprendizaje y la práctica de los modelos de proceso específicos del software, incluido el CMMI.
- *Módulo de arquitectura de software*, que proporciona el conocimiento relacionado con los tipos de arquitectura del software. Es bien sabido que de acuerdo al aumento de las demandas de negocio, la arquitectura necesita ser el punto central para las muchas soluciones del mismo. Este módulo incluye desde la arquitectura orientada al proceso a la orientada a mensajes, de la orientada a objetos a la orientada al servicio, de la arquitectura específica del dominio a la arquitectura de la propia empresa; estilos que se están siguiendo en la

industria para las necesidades futuras de negocio.

- *Módulo de repositorios de datos*, cualquier dato que ayude al negocio en la toma de decisiones y apoye las habilidades de los estudiantes es parte del módulo de repositorios de datos. Este módulo proporciona conocimiento sobre las tendencias en las operaciones del negocio y de la academia usando el almacenamiento, informes y análisis, así como gestión del rendimiento de los datos. Todas las referencias, textos y materiales didácticos se deberán cargar en un módulo electrónico y son creados de acuerdo al desarrollo de las necesidades de negocio y las tendencias tecnológicas, y se actualizarán con el progreso de la tecnología y el desarrollo de la industria mexicana.
- *Módulo de programación y aplicación del conocimiento*, este módulo se refleja en cada curso que incluya práctica con tecnología y cada tema sigue la huella del uso tecnológico diario. El módulo se puede categorizar en dos grupos: cursos de software fundamental; y cursos de desarrollo de software y programación de aplicaciones de acuerdo a las demandas de la industria mexicana del software. El módulo de programación y aplicación del conocimiento combina la educación con el desarrollo práctico para mejorar las habilidades del estudiante.

3.2. Análisis del modelo dinámico en el contexto actual

De acuerdo a Ford y Gibbs [20], en 1989 *“la industria estaba preocupada de que los Colegios y Universidades no estaban produciendo estudiantes que pudieran trabajar productivamente en proyectos de la industria de software.”* A pesar de que es una declaración bastante antigua, considerando que México está intentando desarrollar su industria de software desde el 2005, es perfectamente aplicable a nuestro contexto tecnológico. Los graduados de las carreras en Ingeniería en Sistemas Computacionales o Licenciaturas en Informática comprenden la codificación, compilación, operación de sistemas y demás; pero no trabajan en equipo bajo limitaciones reales. De hecho, no es parte de su formación.

Como parte de esta tesis se ha formulado una investigación paralela sobre la currícula utilizada para la enseñanza de la Ingeniería de Software alrededor del mundo, destacando lo siguiente:

- La investigación de Kitchenham et al., [37] adaptó un instrumento de evaluación desarrollado por Timothy Lethbridge para evaluar el grado de conformidad de la educación proporcionada por las Universidades del Reino Unido con los requisitos de la industria de software. A pesar de que los resultados no fueron satisfactorios debido a la informalidad del instrumento de evaluación, Kitchenham et al. pudieron identificar diferentes temas de importancia para la enseñanza de la Ingeniería de Software, concretamente: modelos de negocio, modelos de calidad, aseguramiento de la calidad, arquitectura del software, y gestión de proyectos.
- En el 2005, Liew [42] rompe el esquema tradicional en la enseñanza de la Ingeniería de Software que indica que ésta debe ser enseñada en niveles avanzados de los cursos de pregrado; y propone un modelo diferente para estudiantes de segundo año (tercer o cuarto semestre) de carrera Universitaria. Liew identifica dos elementos claves para el éxito de

este enfoque: (1) desarrollar más proyectos reales y (2) reducir o eliminar el tiempo invertido en la enseñanza de habilidades de desarrollo de forma teórica. Esta propuesta se enfoca a desarrollar el trabajo en equipo, alcanzar la comprensión de los requisitos y las especificaciones, e implementar programas (sistemas) de complejidad media.

- Fenwick y Kurtz establecieron la Metodología intra-curricular [19] para la enseñanza de la Ingeniería de Software en la Universidad del Estado de Appalachian en E.E.U.U. Este enfoque incorporó los siguientes objetivos a la enseñanza tradicional: aumentar la comprensión del estudiante de los problemas de la gestión de un proyecto, aumentar la comprensión y apreciación del estudiante sobre el diseño de un sistema, incrementar la dedicación del estudiante a los problemas de calidad del software, minimizar la interferencia sobre otros cursos de soporte (materias paralelas). Se agregaron temas especializados sobre: lenguajes de programación, herramientas de soporte y trabajo en equipo.
- El Reporte Introductorio para la currícula de la Computación, presentado en [64], incorporó volúmenes para disciplinas específicas adicionales: Ingeniería en Computación, Sistemas de Información e Ingeniería de Software. Particularmente para esta última disciplina, se recomendaban tópicos como: modelos de negocio y arquitecturas empresariales, calidad del producto, y gestión de proyectos.
- El estudio presentado en [13] exhibe una realidad más cercana al caso de México. Turquía desarrolla algunos productos software de calidad para algunas áreas específicas, pero no puede alcanzar los niveles de éxito de países como la India y algunos otros pequeños países, específicamente para el desarrollo en masa. En este contexto, la Universidad Técnica del Medio Este (METU) propone una currícula que perfeccione las habilidades técnicas de sus egresados a través de los siguientes temas: validación y verificación, aseguramiento de la calidad, arquitecturas de software, técnicas formales de especificación, modelado de procesos, gestión de proyectos, y métodos y técnicas para la construcción de software; entre otras.

Como se puede observar, estas investigaciones se enfocan en buscar formas alternativas de enseñanza para potenciar las habilidades de los estudiantes en el área de la Ingeniería de Software pero, particularmente, centrándose en lo que la industria necesita en ese momento específico. Algo que no está pasando en México como se demostrará más adelante.

Para analizar el contexto educativo de la Ingeniería de Software en nuestro país, se realizó un estudio sobre los programas existentes en las Universidades de mayor renombre (a nivel nacional e internacional). Dada la constante variación en el campo, no es fácil intentar definir un plan de estudios estandarizado para el nivel de pregrado. Sin embargo, no es difícil obtener una comprensión razonablemente común que permite identificar lo siguiente:

- La disciplina de Ciencias de la Computación es la preferida por los estudiantes jóvenes, y la Ingeniería de Software destaca en las preferencias de esta mayoría. Sin embargo, es asombroso encontrar que ya existen muchas materias para incorporar nuevas en un plan de estudios de 10 semestres que es ofrecido en un corto periodo de tiempo.
- Todos los programas revisados reflejan la necesidad de la especialización. Es común observar que las mismas Universidades emplean a un promedio de 25 egresados en puestos técnicos y de especialistas, y el resto de éstos está trabajando en áreas no relacionadas con

la computación.

- La dificultad más grande se encuentra relacionada con la oferta de una variedad de cursos cuando se cuenta con limitación de profesores especialistas. Afortunadamente, y de acuerdo con la Universidad Nacional Autónoma de México (UNAM), el Instituto Politécnico Nacional (IPN), y la Universidad Autónoma Metropolitana (UAM) “*existen instructores bien capacitados en la industria que podrían estar dispuestos a compartir la carga*”.
- A pesar de que se considera un alto dinamismo en el área, los planes de estudios y el conjunto de departamentos que son prominentes y que ofrecen carreras relacionadas con la Computación (y que obviamente incluyen a la Ingeniería de Software) no realizaron cambios drásticos en los últimos cinco años.

Al hacer el análisis comparativo de diversos objetivos del área de Ingeniería de Software de instituciones que la contemplan, se estará en condiciones de analizar críticamente los planes y programas desde una perspectiva amplia. Por lo tanto el planteamiento que se propone para el análisis y la identificación de los problemas educativos, y la solución que se les puede dar, está relacionada con la concepción que se tiene acerca de cómo se produce el conocimiento y con los principios lógicos en que se apoya.

Así, a pesar de que no existe un programa de estudios homogéneo se pueden identificar elementos comunes que pueden compararse con los módulos definidos en esta tesis. Por ejemplo, la Tabla 3.1 resume el objetivo de la materia de Ingeniería de Software en los dos programas de licenciatura ofertados por la UNAM – Facultad de Ciencias.

| <i>Universidad Nacional Autónoma de México</i> | |
|---|-----------|
| Materia: Ingeniería de Software | Semestre: |
| Nombre del programa: Licenciatura en Ciencias de la Computación | |
| Este curso presenta principalmente un estudio profundo de muchos de los temas de Ingeniería de Software, especificación de requisitos, administración de proyectos de software, diseño funcional y orientado a objetos, verificación y mantenimiento de software. Se discuten ambientes y herramientas de software y se introduce a los estudiantes a su uso. Se consideran también implicaciones sociales tales como el costo de las fallas y las responsabilidades profesionales. El curso también comprende una introducción a la Interacción Humano-Computadora y una discusión de algunas interfases de software las cuales son analizadas desde el punto de vista de Ingeniería de Software. Este curso requiere de la participación de los estudiantes en proyectos por equipos. | |
| Nombre del programa: Licenciatura de Ingeniería en Computación | |
| El alumno aplicará en un proyecto procesos y herramientas mediante las cuales se analiza, diseña e implementa un proyecto de software. | |

Tabla 3.1: Descripción de la Ingeniería de Software en la UNAM.

De forma similar, la Tabla 3.2 muestra el objetivo de la UAM – División de Ciencias Naturales e Ingeniería al incorporar la Ingeniería de Software como materia optativa.

| | |
|--|-----------|
| Universidad Autónoma Metropolitana | |
| Materia: Ingeniería de Software | Semestre: |
| Nombre del programa: Ingeniería en Computación | |
| Este curso pretende familiarizar al alumno con los métodos de la Ingeniería de Software aplicables en la solución de problemas complejos. Además se pretende desarrollar la capacidad del alumno para organizar grupos de trabajo multidisciplinarios a través de la aplicación de metodologías de diseño, desarrollo y mantenimiento de proyectos de software. Por último, se proporcionarán conocimientos necesarios para aplicar las metodologías de planeación, evaluación y control de proyectos a casos concretos. | |

Tabla 3.2: Descripción de la Ingeniería de Software en la UAM.

El IPN oferta dos programas de estudio relacionados con la Ingeniería en Computación que incluyen a la Ingeniería de Software como materia obligatoria: la Ingeniería en Computación en la Escuela Superior de Ingeniería Mecánica y Eléctrica, y la Ingeniería en Sistemas Computacionales en la Escuela Superior de Cómputo (ver Tabla 3.3).

| | |
|---|-----------------|
| Instituto Politécnico Nacional | |
| Materia: Ingeniería de Software | Semestre: Sexto |
| Nombre del programa: Ingeniería en Computación | |
| A nivel nacional e internacional, el ambiente laboral y de negocio se caracteriza por una gran dependencia de las TIC's (Tecnologías de la Información y Comunicaciones). El estudio, manejo y dominio de la computación, forma parte central de los modernos sistemas de producción e informática, representando el quehacer diario del personal directivo, administrativo y docente quienes tienen por objetivo aplicar sus conocimientos y talento en la solución de problemas de la Ingeniería de Software de la industria de la investigación. | |
| Nombre del programa: Ingeniería en Sistemas Computacionales | |
| Es la disciplina a través de la cual un profesionalista adquiere una visión innovadora que se anticipa a los cambios tecnológicos para crear y proveer soluciones de software e infraestructura computacional de la era digital. Sus principales herramientas: bases sólidas en ingeniería, ciencias computacionales, desarrollo de software, sistemas de información, infraestructura computacional, administración de proyectos y valores universales sensibles a las necesidades sociales. | |

Tabla 3.3: Descripción de la Ingeniería de Software en el IPN.

La Universidad Veracruzana (UV) es una de las pocas instituciones educativas a nivel nacional que ofertan dos cursos de Ingeniería de Software para una formación disciplinaria en el programa de la Licenciatura en Informática (ver Tabla 3.4).

| | |
|--|---|
| Universidad Veracruzana | |
| Materia: Ingeniería de Software I | Semestre: Por selección de materias |
| Nombre del programa: Licenciatura en Informática | |
| Este curso taller, ubicado en el área de formación disciplinaria, desarrolla aspectos centrales de las competencias de un informático. En éste, el estudiante conoce diversos procesos de desarrollo, métodos de analizar y diseñar sistemas, así como de asegurar y medir la calidad de los productos a desarrollar o seleccionar y además realiza prácticas específicas de algunos métodos relacionados con los temas anteriores. La realización de prácticas y el desarrollo se evalúan a lo largo del curso, de modo que el estudiante evidencie su desarrollo mediante reportes de modelado y prácticas realizadas, atendiendo a criterios básicos de pertinencia, claridad y coherencia. En éste, el estudiante conoce diversos aspectos importantes del desarrollo, especialmente lo relativo a los requerimientos del usuario y a los criterios de usabilidad que deben aplicarse. Al mismo tiempo se introduce a las metodologías de análisis y diseño orientadas a objetos usando UML. | |
| Materia: Ingeniería de Software II | Semestre: Seriado con el curso anterior |
| En este curso, el estudiante conoce diversos aspectos importantes del desarrollo, especialmente lo relativo a los requerimientos del usuario y a los criterios de usabilidad que deben aplicarse. Al mismo tiempo se introduce a las metodologías de análisis y diseño orientadas a objetos usando UML. | |

Tabla 3.4: Descripción de la Ingeniería de Software en la UV.

Por último, la Tabla 3.5 resume el objetivo de la materia de Ingeniería de Software dentro del programa de Ingeniería en Computación en la Universidad Tecnológica de la Mixteca (UTM).

| | |
|--|------------------|
| Universidad Tecnológica de la Mixteca | |
| Materia: Ingeniería de Software | Semestre: Cuarto |
| Nombre del programa: Ingeniería en Computación | |
| Proporcionar al estudiante el conocimiento necesario para analizar, diseñar e implementar un sistema de computación nuevo o mejorar uno existente. Asimismo el estudiante conocerá las técnicas y metodologías de la Ingeniería de Software para desarrollar productos de calidad. | |

Tabla 3.5: Descripción de la Ingeniería de Software en la UTM.

Es fácil distinguir que, a pesar de las marcadas diferencias, es posible identificar aspectos comunes en los planes de estudio mencionados. Por obvias razones es imposible analizar cada plan de estudios a detalle, pero consultando la información proporcionada por las Universidades se puede reconocer el interés por:

- Gestión de los proyectos de software (planeación, evaluación y control).
- Procesos de desarrollo.
- Diseño y modelación de sistemas.
- Aseguramiento de la calidad.

- Prácticas de desarrollo por equipo.
- Arquitecturas del software.

A pesar de que actualmente no existen cursos concretos que empaten con el módulo de arquitectura de software, el plan de estudios actual cubre en un 90 % los conceptos mínimos identificados para establecer un aprendizaje proactivo de la Ingeniería de Software. Después de un análisis sobre el plan de estudios y temarios existentes, fue posible considerar temas específicos para cubrir las habilidades básicas relacionadas con la preparación de un arquitecto de software. De esta forma sólo se agregaron temas específicos para aquellos alumnos que participaron en la validación del modelo. Estos temas básicamente fortalecen las habilidades del estudiante para el trabajo en equipo dado que no es posible cometer errores cuando se interactúa con una empresa real.

En nuestro contexto, las Tablas 3.6 a 3.16 muestran el mapeo entre los módulos propuestos al inicio de este Capítulo y el temario establecido para la Ingeniería de Software en el Plan de Estudios actual para la Ingeniería en Computación.

| Módulo propuesto | Concordancia con plan de estudios actual |
|---------------------------------|--|
| | Temas y subtemas |
| | Curso: <i>Bases de datos</i> |
| Módulo de repositorios de datos | <p>1. Introducción a las bases de datos</p> <p>1.1 Tipos de Datos Estructurados, Semi-estructurados, No estructurados</p> <p>1.2 Niveles de abstracción</p> <p>1.3 Funciones de un DBMS</p> <p>1.4 Componentes de un DBMS</p> <p>1.5 Modelos de Datos</p> <p>2. Modelo entidad-relación</p> <p>2.1 Conceptos básicos</p> <p>2.2 Tipos de Llaves</p> <p>2.3 Diagrama</p> <p>2.4 Características del modelo entidad-relación extendido</p> <p>3. Modelo relacional</p> <p>3.1 Transformación del modelo E-R al modelo relacional</p> <p>3.2 Álgebra relacional (QBE)</p> <p>3.3 Cálculo relacional de tuplas</p> <p>3.4 Cálculo relacional de predicados</p> <p>3.5 Reglas de integridad</p> <p>4. Diseño de base de datos relacionales</p> <p>4.1 Definición del problema</p> <p>4.2 Normalización: 1NF, 2NF, 3NF, BCNF</p> <p>4.3 Criterios para normalizar</p> <p>5. Lenguaje SQL</p> <p>5.1. Lenguaje de definición de datos</p> <p>5.2. Lenguaje de manipulación de datos</p> <p>5.3. SQL incluyendo: procedimientos almacenados, disparadores, cursores</p> |

Tabla 3.6: Módulo de repositorios de datos.

| | |
|-----------------------------------|--|
| | Curso: <i>Administración y dirección de empresas</i> |
| Módulo para el entorno de negocio | <p>1. Proceso Administrativo</p> <p>1.1 Planeación</p> <p>1.2 Organización</p> <p>1.3 Dirección</p> <p>1.4 Control</p> <p>2. Cultura organizacional</p> <p>2.1 Concepto de cultura organizacional y su importancia</p> <p>2.2 Análisis de los elementos que integran la cultura organizacional</p> <p>3. Introducción a la función de dirección</p> <p>3.1 Definición de la función de dirección</p> <p>3.2 Tareas de la alta dirección</p> <p>3.3 Estilos de dirección</p> <p>3.4 Modelos gerenciales</p> <p>4. El medio ambiente de las organizaciones y la función estratégica de la dirección</p> <p>4.1 Factores que determinan el ambiente</p> <p>4.2 La planeación estratégica y el compromiso de la alta dirección</p> <p>4.3 Estrategias directivas</p> <p>4.4 Estrategia competitiva</p> <p>4.5 Ética y responsabilidad social en las organizaciones</p> <p>5. La empresa como proyecto</p> <p>5.1 Diseño y estructura administrativa de la empresa</p> |
| | Curso: <i>Administración de Infraestructura de Tecnologías de Información</i> |
| | <p>1. Las tecnologías de información y las organizaciones</p> <p>1.1 Conceptos y características de las tecnologías de información</p> <p>1.2 Impacto y repercusiones de los sistemas de información en las organizaciones</p> <p>1.3 Usuarios de las tecnologías de información</p> <p>1.4 Infraestructura de TI</p> <p>1.5 Características y requerimientos de software, hardware y personal humano</p> <p>2. Administración de Recursos e Infraestructura</p> <p>2.1 Recursos Físicos</p> <p>2.2 Personal de TI</p> <p>2.3 Hardware</p> <p>2.4 Software y regulación</p> <p>3. Caso de estudio</p> <p>3.1 Modelo de administración de TI</p> <p>3.2 Caso práctico</p> |

Tabla 3.7: Módulo para el entorno de negocio.

| | Curso: <i>Procesos de calidad</i> |
|-------------------------------|---|
| Módulos de modelos de proceso | <p>1. Introducción a los procesos de calidad</p> <p>1.1 ¿Qué es Calidad?</p> <p>1.2 Conceptos de Calidad</p> <p>2. Procesos</p> <p>2.1 Importancia en la definición de procesos</p> <p>2.2 Definición de procesos (entradas, salidas y actividades)</p> <p>2.3 Procesos en el desarrollo de Software</p> <p>2.3.1 Administración de Proyectos</p> <p>2.3.2 Administración de Información</p> <p>2.3.3 Administración de riesgos</p> <p>2.3.4 Proceso de desarrollo de software en equipo (TSP)</p> <p>3. Roles comunes</p> <p>3.1 Líder de Proyecto</p> <p>3.2 Líder técnico</p> <p>3.3 Administrador de la configuración</p> <p>3.4 Ingeniero de calidad</p> <p>3.5 Ingeniero de software</p> <p>3.6 Tester</p> <p>4. Modelos</p> <p>4.1 Norma ISO 15504 (MetaNorma de los modelos de calidad)</p> <p>4.2 CMMI</p> <p>4.3 SPICE</p> <p>4.4 MoProSoft</p> <p>5. Mejora continua de los procesos</p> <p>5.1 Mejora de procesos</p> <p>5.1.1 Iniciar</p> <p>5.1.2 Diagnosticar</p> <p>5.1.3 Establecer</p> <p>5.1.4 Actuar</p> <p>5.1.5 Difundir</p> |

Tabla 3.8: Módulos de modelos de proceso.

| | |
|--|--|
| | Curso: <i>Programación estructurada</i> |
| Módulo de programación y aplicación del conocimiento | <p>1. El lenguaje de Programación ANSI C</p> <p>1.1 Introducción a la programación estructurada</p> <p>1.2. Estructura de un programa</p> <p>1.3 Tipos de datos</p> <p>1.4 Operadores</p> <p>1.5 Ejemplos de instrucciones de entrada y salida</p> <p>2. Estructuras de Control</p> <p>2.1 Secuencial</p> <p>2.2 Selectivas</p> <p>2.3 Iterativas.</p> <p>3.Funciones</p> <p>3.1 Definición de funciones</p> <p>3.2 Funciones con paso de parámetros por valor</p> <p>3.3 Funciones con paso de parámetros por referencia</p> <p>3.4 Funciones recursivas</p> <p>4. Arreglos</p> <p>4.1 Arreglos unidimensionales</p> <p>4.2 Arreglos bidimensionales</p> <p>4.3 Arreglos multidimensionales</p> <p>4.4 Arreglos como parámetros</p> <p>5. Apuntadores</p> <p>5.1. Apuntadores</p> <p>5.2 Aritmética de apuntadores</p> <p>5.3 Memoria estática y dinámica</p> <p>5.4 Arreglos dinámicos de una y dos dimensiones</p> <p>5.5 Funciones como parámetros</p> <p>6. Tipos de datos definidos por el programador</p> <p>6.1 Introducción al tipo de dato abstracto</p> <p>6.2 Implementación de un tipo de dato abstracto</p> <p>6.3 Tipos de datos abstractos como parámetros</p> <p>6.4 Arreglos de tipos de datos abstractos</p> <p>7.Archivos</p> <p>7.1 Operaciones básicas</p> <p>7.2 Texto y binarios</p> <p>7.3 Acceso secuencial y directo</p> |

Tabla 3.9: Módulo de programación y aplicación del conocimiento.

| | |
|--|---|
| | Curso: Programación orientada a objetos I |
| | <p>1. Introducción a la programación orientada a objetos</p> <p>1.1 Paradigmas de programación</p> <p>1.2 El paradigma orientado a objetos</p> <p>1.3 Lenguajes de programación orientados a objetos</p> <p>2. Conceptos básicos de la programación orientada a objetos</p> <p>2.1 Abstracción y programación</p> <p>2.2 Tipos abstractos de datos</p> <p>2.3 Clase</p> <p>2.4 Objetos y mensajes</p> <p>2.5 Herencia</p> <p>2.6 Asociación y agregación</p> <p>2.7 Polimorfismo</p> <p>3. Definición de clases y objetos</p> <p>3.1 Abstracción y ocultamiento de información</p> <p>3.2 Métodos de acceso y de utilidad</p> <p>3.3 Constructores y destructores</p> <p>3.4 Sobrecarga de operaciones</p> <p>4. Relaciones entre clases: Herencia</p> <p>4.1 Clase base y clase derivada</p> <p>4.2 Clases bases directas e indirectas</p> <p>4.3 Redefinición de los métodos de una clase base en una clase derivada</p> <p>4.4 Constructores y destructores en clases derivadas</p> <p>4.5 Conversión implícita de objetos de una clase derivada a objetos de una clase base</p> <p>4.6 Conversión explícita de objetos de una clase base a objetos de una clase derivada</p> <p>4.7 Herencia múltiple</p> <p>4.8 Interfaces</p> <p>5. Relaciones entre clases: Asociación y agregación</p> <p>5.1 Implementación de asociación y agregación</p> <p>5.2 Asociación/agregación reflexiva</p> <p>5.3 Multiplicidad en asociación/agregación</p> <p>6. Polimorfismo</p> <p>6.1 Clases bases abstractas</p> <p>6.2 Implementación de polimorfismo</p> <p>7. Programación genérica</p> <p>7.1 Implementación de programación genérica</p> <p>7.2 Uso de bibliotecas estándar de programación genérica</p> <p>8. Manejo de Excepciones</p> <p>8.1 Concepto de manejo de excepciones</p> <p>8.2 Implementación de manejo de excepciones</p> <p>8.3 Jerarquía de excepciones</p> |

Tabla 3.10: Módulo de programación y aplicación del conocimiento (continuación).

| | |
|--|--|
| | Curso: <i>Programación orientada a objetos II</i> |
| | <p>1. Conceptos básicos para el modelado de software</p> <p>1.1 Antecedentes de UML</p> <p>1.2 Modelado visual con UML</p> <p>1.3 Procesos actuales para el desarrollo de software orientado a objetos</p> <p>2. Requerimientos</p> <p>2.1 Introducción</p> <p>2.2 Artefactos involucrados</p> <p>2.3 Personas involucradas y sus actividades</p> <p>2.4 Modelado de requerimientos</p> <p>3. Análisis orientado a objetos</p> <p>3.1 Introducción</p> <p>3.2 Artefactos involucrados</p> <p>3.3 Personas involucradas y sus actividades</p> <p>3.4 Modelado de análisis</p> <p>4. Diseño orientado a objetos</p> <p>4.1 Introducción</p> <p>4.2 Diferencia entre análisis y diseño</p> <p>4.3 Artefactos involucrados</p> <p>4.4 Personas involucradas y sus actividades</p> <p>4.5 Introducción a los patrones de diseño</p> <p>4.6 Modelado de diseño</p> <p>5. Implementación</p> <p>5.1 Introducción</p> <p>5.2 Artefactos involucrados</p> <p>5.3 Personas involucradas y sus actividades</p> <p>5.4 Modelado de implementación</p> <p>6. Pruebas</p> <p>6.1 Introducción</p> <p>6.2 Artefactos involucrados</p> <p>6.3 Personas involucradas y sus actividades</p> <p>6.4 Modelado de pruebas</p> |

Tabla 3.11: Módulo de programación y aplicación del conocimiento (continuación).

| | |
|--|---|
| | Curso: <i>Programación en Web</i> |
| | <p>1. Introducción al WWW</p> <p>1.1 Origen y evolución de Internet</p> <p>1.2 Lenguaje de marcas (SGML)</p> <p>1.3 HTML y XML</p> <p>1.4 Tendencias de HTML</p> <p>2. Aplicaciones Web</p> <p>2.1 Metodologías para el desarrollo de aplicaciones Web</p> <p>2.2 Arquitectura de las aplicaciones Web</p> <p>2.3 Consideraciones de seguridad</p> <p>2.4 Diferencias en la interpretación y despliegue</p> <p>3. Lenguaje de marcas hipertexto (HTML)</p> <p>3.1 Formato de lenguaje y formalidades</p> <p>3.2 Elementos básicos del lenguaje</p> <p>3.3 Tablas, listas y estructuras de datos</p> <p>3.4 Marcas de formularios</p> <p>3.5 Integración de elementos multimedia</p> <p>4. Hojas de estilo en cascada (CSS)</p> <p>4.1 Sintaxis y organización</p> <p>4.2 Parámetros de estilo</p> <p>4.3 Características avanzadas para múltiples dispositivos</p> <p>5. DHTML y lenguajes script</p> <p>5.1 Formato de lenguaje (Javascript)</p> <p>5.2 Modos de integración en páginas</p> <p>5.3 Capacidades y alcances</p> <p>5.4 Modelo de objetos del lenguaje</p> <p>5.5 Manejo de eventos</p> <p>5.6 Interacción con CSS y HTML</p> <p>6. Componentes basados en servidores</p> <p>6.1 Gestión del lado del servidor</p> <p>6.2 Lenguajes de procesamiento en el servidor (CGI, PHP, JSP, ASP)</p> <p>6.3 Procesado de formularios y sesiones</p> <p>6.4 Interacción entre servidores (bases de datos, servicios Web)</p> <p>7. Interacción web avanzada</p> <p>7.1 Actualización y transferencias de contenido (AJAX, JSON, SOAP)</p> <p>7.2 Plataformas de desarrollo web avanzados (Ruby on Rails)</p> <p>7.3 Tendencias actuales y futuras (RSS, ATOM)</p> |

Tabla 3.12: Módulo de programación y aplicación del conocimiento (continuación).

| | |
|--|--|
| | Curso: <i>Ingeniería de Software I</i> |
| | <p>1. Introducción a la ingeniería de software</p> <p>1.1 Importancia del Software</p> <p>1.2 Evolución y características del software</p> <p>1.3 Tipos de software</p> <p>1.4 La crisis del software</p> <p>1.5 Definición de Ingeniería de Software</p> <p>1.6 Paradigmas de ciclos de vida de la Ingeniería de software</p> <p>1.7 Herramientas CASE</p> <p>2. Administración de proyectos de software</p> <p>2.1 Gráficas PERT, GANTT</p> <p>2.2 Métricas del proyecto</p> <p>2.3 Mediciones del software</p> <p>2.4 Métricas orientadas al tamaño(LDC)</p> <p>2.5 Modelo de estimación de costos COCOMO</p> <p>2.6 Métricas orientadas a los puntos de función</p> <p>2.7 Análisis de riesgo</p> <p>3. Análisis de requerimientos</p> <p>3.1 Técnicas de recolección de información</p> <p>3.2 Identificación de requerimientos</p> <p>3.3 Análisis de requisitos basados en el estándar 830-1993 IEEE</p> <p>3.4 Introducción y aplicación a los métodos estructurados</p> <p>3.5 Introducción del método orientado a objetos en el análisis</p> <p>3.6 Validación de requerimientos</p> <p>4. Diseño e implementación del Software</p> <p>4.1 Niveles de diseño</p> <p>4.2 Aplicación del diseño estructurado</p> <p>4.3 Introducción al diseño orientado a objetos</p> <p>5. Métodos de prueba del software</p> <p>5.1 El proceso de prueba</p> <p>5.2 Métodos de prueba</p> <p>5.3 Diseño de pruebas</p> <p>5.4 Caso de estudio: estándar de prueba</p> <p>6. Administración de la calidad del software</p> <p>6.1 El concepto de calidad del software</p> <p>6.2 Estándares de calidad existentes</p> <p>6.3 Métricas de calidad del software</p> <p>7. Mantenimiento de software</p> <p>7.1 Aspectos generales</p> <p>7.2 Características</p> <p>7.3 Categorías de mantenimiento: correctivo, adaptativo, preventivo, perfectivo</p> <p>7.4 Reingeniería</p> |

Tabla 3.13: Módulo de programación y aplicación del conocimiento (continuación).

| | |
|--|--|
| | Curso: <i>Ingeniería de Software II</i> |
| | <p>1. Ingeniería de Requerimientos</p> <p>1.1 Importancia de los requerimientos en el éxito del desarrollo del software</p> <p>1.2 Técnicas de obtención de requerimientos:</p> <p>1.2.1 Entrevistas y cuestionarios</p> <p>1.2.2 Taller de requerimientos</p> <p>1.2.3 Lluvia y reducción de ideas</p> <p>1.2.4. Storyboards</p> <p>1.2.5 Casos de uso</p> <p>1.2.6 Juego de roles</p> <p>1.2.7 Prototipos</p> <p>1.3 Validación de requerimientos</p> <p>1.4 La Ingeniería de Requerimientos bajo diferentes procesos de software.</p> <p>2. Tendencias actuales en metodologías de desarrollo de software</p> <p>2.1 Un panorama de las metodologías actuales</p> <p>2.2 Métodos ágiles y/o RAD</p> <p>2.3 Métodos tradicionales</p> <p>2.4 Técnicas efectivas para el desarrollo de software</p> <p>3. Reingeniería del Software</p> <p>3.1 Introducción a la reingeniería de software</p> <p>3.2 Cuándo aplicar reingeniería en el proceso del software</p> <p>3.3 Métodos y modelos de la reingeniería de software</p> <p>3.4 Recursos para la reingeniería de software</p> <p>3.5 Ingeniería inversa</p> <p>4. Especificación y verificación formal de software</p> <p>4.1 Justificación e Introducción a los métodos formales</p> <p>4.2 Especificación formal de software</p> <p>4.3 Verificación formal de software</p> <p>4.4 Herramientas para la especificación y verificación formal de software</p> |

Tabla 3.14: Módulo de programación y aplicación del conocimiento (continuación).

| | |
|--|---|
| | Curso: <i>Administración de proyectos</i> |
| | <p>1. Conceptos básicos de la administración de proyectos</p> <p>1.1 Significado e importancia de la administración de proyectos</p> <p>1.2 Beneficios de la administración de proyectos</p> <p>1.3 Importancia de la comunicación en la administración de proyectos</p> <p>1.4 Ciclo de vida de los proyectos</p> <p>1.5 Procesos en la dirección y administración de proyectos</p> <p>2. Planeación del proyecto</p> <p>2.1 Asignación de recursos</p> <p>2.2 Diagramas PERT-CPM</p> <p>2.3 Puntos de control</p> <p>2.4 Establecimiento de una línea base</p> <p>2.5 Desarrollo de actividades</p> <p>2.6 Finalización del proyecto</p> <p>2.7 Entrega del proyecto</p> <p>3. Viabilidad del proyecto</p> <p>3.1 Estimación de costos y presupuestos</p> <p>3.2 Costos de capital y financiamiento</p> <p>3.3 Punto de equilibrio (inversión/financiamiento)</p> <p>3.4 Evaluación económica</p> <p>3.5 Impacto del proyecto</p> <p>3.6 Análisis y administración de riesgos</p> <p>4. Administración</p> <p>4.1 Responsabilidades y habilidades del líder</p> <p>4.2 Integrantes del equipo</p> <p>4.3 Asignación de roles de trabajo</p> <p>4.4 Organización del equipo de trabajo</p> <p>5. Software en la administración de proyectos</p> <p>5.1 Herramientas de software para la administración de proyectos</p> <p>5.2 Introducción al uso de alguna de las herramientas (Project)</p> <p>5.3 Uso del Internet en la administración de proyectos</p> <p>6. Desarrollo y control del proyecto</p> <p>6.1 Actividades durante el seguimiento y monitoreo del proyecto</p> <p>6.2 Control de calidad</p> <p>6.3 Detección de errores y el manejo de conflictos</p> <p>7. La fase final del proyecto</p> <p>7.1 Tipos de prueba</p> <p>7.2 Planeación de pruebas</p> <p>7.3 Entrega y “lecciones aprendidas”</p> <p>8. Documentación del proyecto</p> <p>8.1 Informe técnico</p> <p>8.2 Informe administrativo</p> <p>8.3 Informe financiero</p> <p>8.4 Otros informes (desempeño, costo del proyecto, cambios, calidad)</p> |

Tabla 3.15: Módulo de programación y aplicación del conocimiento (continuación).

| | |
|------------------------------------|--|
| | Curso: <i>Programación orientada a objetos II</i> |
| Módulo de arquitectura de software | 1. Conceptos básicos para el modelado de software 1.1 Antecedentes de UML 1.2 Modelado visual con UML 1.3 Procesos actuales para el desarrollo de software orientado a objetos 2. Diseño orientado a objetos 2.1 Introducción 2.2 Diferencia entre análisis y diseño 2.3 Artefactos involucrados 2.4 Personas involucradas y sus actividades 2.5 Introducción a los patrones de diseño 2.6 Modelado de diseño |
| | Curso: <i>Sistemas distribuidos y paralelos</i> |
| | 1. Conceptos de sistemas distribuidos y paralelos 1.1 Escalabilidad 1.2 Transparencia 1.3 Consistencia y control de datos 1.4 Control de concurrencia 1.5 Tolerancia a fallas y recuperación 2. Paradigmas y plataformas de desarrollo distribuido y paralelo 2.1. Sockets 2.2. RPC 2.3. Plataformas de programación distribuida 2.4. Plataformas de programación paralela 2.5. Herramientas de monitoreo y medición de rendimiento |
| | Curso: <i>Bases de datos</i> |
| | 1. Modelo entidad-relación 1.1 Conceptos básicos 1.2 Tipos de Llaves 1.3 Diagrama 1.4 Características del modelo entidad-relación extendido 2. Modelo relacional 2.1 Conceptos básicos 2.2 Transformación del modelo E-R al modelo relacional 2.3 Álgebra relacional (QBE) 2.4 Cálculo relacional de tuplas 2.5 Cálculo relacional de predicados 2.6 Reglas de integridad 3. Diseño de base de datos relacionales 3.1 Definición del problema 3.2 Normalización: 1NF, 2NF, 3NF, BCNF 3.3 Criterios para normalizar |

Tabla 3.16: Módulo de arquitectura de software.

De esta forma, la solución propuesta en esta tesis primero establece las bases de un modelo educativo para la Ingeniería de Software en base al plan de estudios actual de la Ingeniería en Computación de la Universidad Tecnológica de la Mixteca. Sin embargo, y dado que se pretende incorporar dinamismo al aprendizaje de esta área, es necesario crear una herramienta web que establezca un aprendizaje proactivo permitiendo la interacción entre la empresa y la universidad. La Figura 3.2 muestra que estos dos elementos son cruciales para el establecimiento de un modelo que permita incrementar la participación de los estudiantes en proyectos reales y reducir así el vacío que existe entre lo que las empresas necesitan de un egresado y lo que éste realmente aprende en clases.

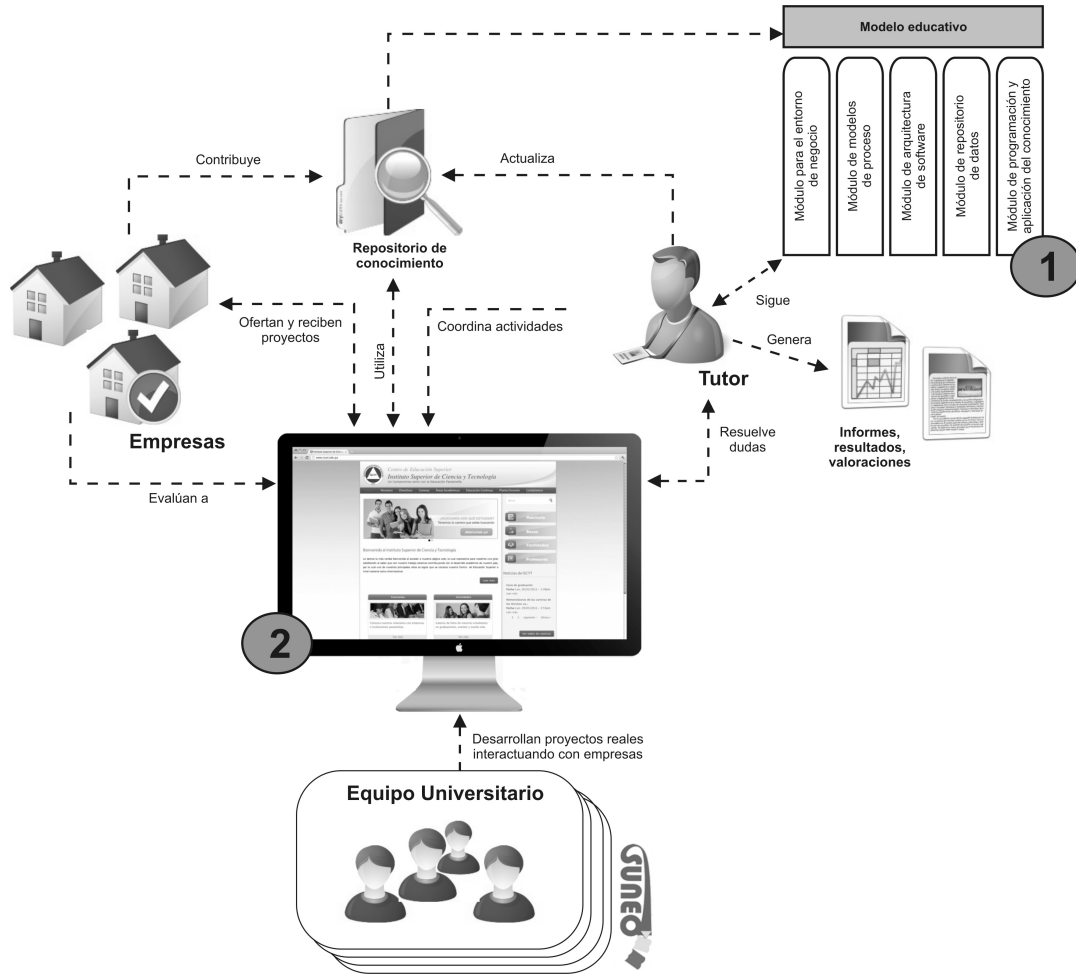


Figura 3.2: Diagrama a bloques de la solución planteada.

Por último, dado que el objetivo de esta tesis no radica en proponer un nuevo plan de estudios específico de la Ingeniería de Software, sino apoyar el aprendizaje de la misma con el plan de estudios actual (orientado a la Ingeniería en Computación); el siguiente Capítulo se centrará en el diseño y desarrollo de una herramienta software que permita establecer un aprendizaje vinculado y que sirva para implementar los cinco módulos propuestos en el modelo dinámico; la herramienta VinculAE (Vinculación Academia Empresa).

Capítulo 4

Diseño y desarrollo de VinculAE.

En la actualidad, al trabajar con desarrollos que interactúan con el usuario vía web se necesita trabajar con estándares que involucren directamente al usuario final. Con esto se ayuda a que éstos utilicen la web de manera más rápida y útil aprovechando de mayor manera la información ahí presentada.

Cuando se habla de usabilidad, se crea controversia, pero la clave de ello es lograr algo funcional y que satisfaga al usuario final. Cabe mencionar que la palabra “usabilidad” es muy amplia y que se aplica no solamente al ámbito de Internet, sino a cualquier cosa que pueda ser “usada” por alguien. En el contexto de esta tesis, todo sitio web que tenga que ver con la educación necesita proporcionar espacios que faciliten el trabajo operativo en el proceso de enseñanza-aprendizaje, por eso es muy importante diseñar sitios usables para la academia, y sobre todo si ésta se vincula con la industria, esto con el fin de que sea más fácil para ambos alcanzar sus objetivos y que no tengan que preocuparse de la herramienta a utilizar.

El movimiento acelerado en el desarrollo tecnológico y en diferentes áreas que tienen que ver con la educación tales como el e-learning y los modelos de enseñanza apoyados en sistemas multimedia, han hecho que el diseño y modelado hayan adquirido especial relevancia en el escenario educativo, por lo que en esta tesis se ha contemplado trabajar con el Diseño Centrado en el Usuario en su versión Extendida [22].

Dando pie a esto y analizando al usuario final, por ejemplo cuando utiliza o visita un sitio web, lo primero que hace es un barrido visual, examinando el contenido en la pantalla, eliminando visualmente la información que no le interesa y centrando su atención en la que si, siendo esta una práctica muy común en la mayoría de usuarios. Por lo tanto, un buen diseño de la información tomándose desde el punto de vista organizativo y de su usabilidad, es aquel que ayuda al usuario a encontrar la información que busca de la forma más fácil, rápida y cómoda posible. Por lo que se debe evitar la sobrecarga informativa, esto es demasiada información textual y visual.

De acuerdo con Bevan, Kirakowski y Maissel [7], la usabilidad es un anglicismo que significa "facilidad de uso" y que parece tener su origen en la expresión "user friendly", que es reemplazada por sus connotaciones vagas y subjetivas. Múltiples autores han propuesto diversas definiciones de usabilidad, normalmente a través de la enumeración de los diferentes atributos o factores mediante los que puede ser evaluada, aunque al final cada definición depende del enfo-

que con el que ésta se mida [16].

En el presente trabajo y con fines prácticos tomaremos la definición más extendida, que es la ofrecida por el estándar internacional ISO 9241-11 (1998): y que define usabilidad como: “*El grado en que un producto puede ser usado por determinados usuarios para lograr sus propósitos con eficacia, eficiencia y satisfacción en un contexto de uso específico*”.

A continuación se exponen los términos usados en la definición:

- Eficacia: precisión con la que los usuarios alcanzan las metas específicas.
- Eficiencia: normalmente, la eficiencia suele medirse en términos del tiempo que les lleva a los usuarios realizar dichas tareas.
- Satisfacción: percepción de agrado y actitud positiva hacia el uso del producto.

Según la definición de la ISO, la usabilidad posee atributos cuantificables de forma objetiva (eficacia y eficiencia) y atributos cuantificables de forma subjetiva (satisfacción) [23].

4.1. Diseño Centrado en el Usuario Extendida

El Diseño Centrado en el Usuario (DCU) o UCD del inglés (*User-Centered Design*) ha cobrado popularidad en los últimos años como proceso encaminado al diseño de productos, objetos o servicios, que respondan a las necesidades reales de sus usuarios finales.

El primer uso más o menos conocido del concepto de DCU fue realizado en el libro de 1986 “*User Centered System Design; New Perspectives on Human-Computer Interaction*” [49] el cual es una recopilación de artículos de diferentes autores sobre el diseño de sistemas informáticos desde el punto de vista de sus usuarios. Donald Norman lo popularizó más tarde utilizándolo como título del capítulo 7 en su libro *La psicología de los objetos cotidianos* traducción del título original “*The Design of Everyday Things*” [48], en el que comenta del diseño centrado en el usuario, una teoría basada en las necesidades y los intereses del usuario, con especial hincapié en hacer que los productos sean utilizables y comprensibles. Las obras bibliográficas de Norman son referenciadas y utilizadas como material base en cursos de interacción humano – computadora, diseño y computación.

En la metodología de Diseño Centrado en el Usuario las necesidades de los usuarios se hacen presentes a lo largo de todo el proceso de desarrollo. A diferencia de otras metodologías, en el DCU el usuario es parte esencial de todas y cada una de las fases del proceso de desarrollo.

La principal fortaleza del DCU se basa en la evaluación y la investigación, cada una de las hipótesis propuestas durante el desarrollo de un proyecto se somete a pruebas con usuarios y se califican de acuerdo con criterios de usabilidad.

El DCU ha desarrollado una serie de metodologías propias a fin de poder conocer adecuadamente a los usuarios incluso desde la fase de análisis.

Por lo regular en la mayoría de los procesos de diseño centrado en el usuario, la primera fase es el estudio o análisis.

Entre las técnicas de DCU en esta fase se encuentran las entrevistas, los estudios de campo y

las sesiones de grupo, que sirven para saber más sobre el usuario final y cómo interactúa con los productos. Así mismo; estas técnicas sirven también para saber sobre los problemas de accesibilidad que presentan los mismos.

El DCU representa una alternativa a los sistemas tradicionales de diseño que marchan en dirección de funcionalidad o tecnología, que son llevados a cabo por expertos que se basan en sus conocimientos, donde las necesidades de los usuarios finales quedan en un plano secundario, y generalmente resultan en productos difíciles de entender y/o manejar por los usuarios finales; caso muy recurrente en el desarrollo de aplicaciones software.

De acuerdo a [22], el DCU sigue normalmente un ciclo iterativo, que comprende cuatro procesos fundamentales: estudio, diseño, construcción y evaluación (ver Figura 4.1). Pueden utilizarse diferentes términos, pero fundamentalmente las cuatro etapas implican el mismo tipo de actividades.

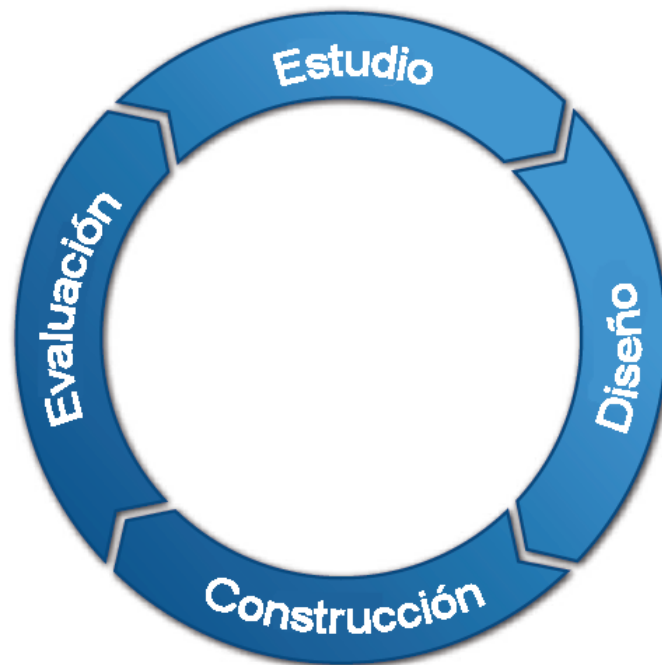


Figura 4.1: Las 4 etapas del modelo de investigación / diseño centrado en el usuario convencional.

En marzo de 2007, durante el foro titulado HCI 2020: Valores Humanos en la Era Digital, celebrado en San Lúcar la Mayor, España. Donde el propósito fue reunir luminarias en informática, diseño, ciencias sociales y filosofía científica para discutir, debatir y ayudar a formular una agenda para la interacción humano – computadora HCI (*Human Computer Interaction*, Interacción Humano Computadora por sus siglas en Inglés) durante la próxima década y más allá.

En este evento facilitado por Microsoft y convocado por Richard Harper y Abigail Sellen de Microsoft Research de Cambridge, Tom Rodden de la Universidad de Nottingham del Reino Unido e Yvonne Rogers de la Universidad Abierta, dio lugar a un informe detallado, lanzado el 2 de abril del 2008, llamado Being Human : Human-Computer Interaction in the Year 2020.

Proponen en [22] que un nuevo programa de HCI debe extender este modelo de diseño, mediante la adición de una nueva etapa, que implica el análisis conceptual (ver Figura 4.2).

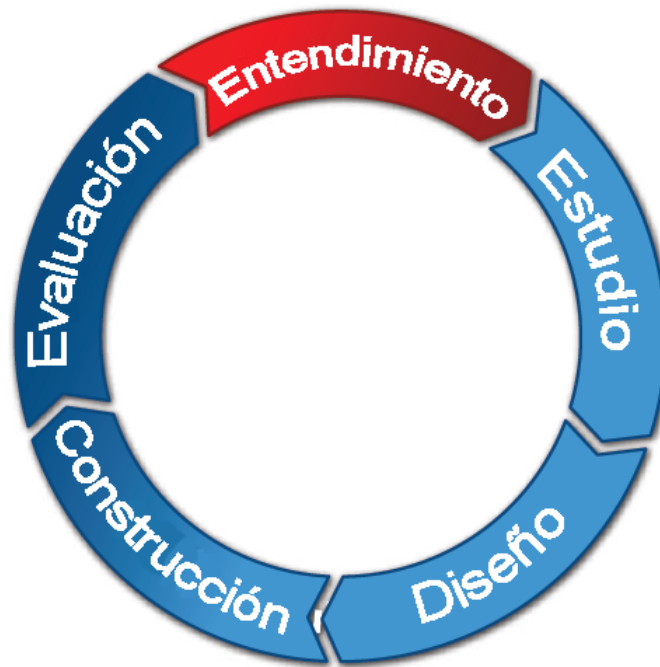


Figura 4.2: Metodología DCU extendida. La nueva etapa implica el análisis conceptual o “entendimiento”.

Al agregar una etapa de entendimiento, a sabiendas que la comprensión de un problema siempre ha sido parte de la fase de estudio, se pretende establecer un proceso más explícito, como lo proponen en [22], en donde se tomen en cuenta los diversos valores humanos y las ventajas y desventajas examinadas de manera sistemática. Esto conlleva a la participación de varias áreas del saber.

En esta nueva etapa se necesita saber más acerca de las experiencias de la gente, los usuarios finales, es aquí donde los estudios etnográficos, registros de interacción con el usuario y las encuestas se usan comúnmente.

En base a los resultados obtenidos, se comienza a pensar por qué, qué y cómo diseñar algo mejor. Continuando con el proceso, se identifican los objetivos de usabilidad y experiencia de

usuario para desarrollar modelos conceptuales. Los prototipos se construyen, y son evaluados reiteradamente, hasta que se alcancen los objetivos establecidos por los usuarios o la nueva experiencia del grupo objetivo sea agradable, placentera y valiosa.

El ciclo, como es normalmente el caso en HCI, se puede introducir en cualquier momento, y por lo general se repite en el transcurso de cualquier proyecto de investigación o de diseño.

4.1.1. Etapa 1: Entendimiento

La primera y nueva etapa sugiere centrarse en los valores humanos y precisar qué y para qué se desea diseñar o investigar. Para ello será necesario el pensamiento reflexivo, el análisis conceptual entre otras disciplinas, tales como la filosofía, psicología, arte, literatura, estudios culturales, antropología, sociología o diseño. También significa hablar con las partes interesadas, incluidos los usuarios, así como los que participan en la elaboración o diseño de la tecnología en cuestión (si este es el objetivo).

Por lo tanto esta nueva etapa del ciclo, da lugar a la toma de decisiones. También implica especificar qué tipo de personas son el centro del proyecto, en particular sus actividades, entorno o cultura, e información fundamental para llevar a cabo la etapa 2.

4.1.2. Etapa 2: Estudio

Consiste en desarrollar un conocimiento más profundo de los factores que están en juego, mientras que la Etapa 1 proporciona un marco para orientar el diseño y la investigación, esta etapa consiste en dar contenido a los detalles de cómo los individuos y los grupos sociales persiguen y alcanzan aspiraciones particulares.

Se parte de considerar los detalles de determinadas tareas o prácticas, pero se pregunta cómo los mecanismos de interacción ayudan a las personas a obtener un valor duradero a través y más allá de la interacción.

Este tipo de análisis no sólo toma en cuenta la interacción de las personas con la tecnología informática, sino también su interacción con el mundo cotidiano de una manera más amplia, como por ejemplo, el medio ambiente, los objetos cotidianos, con otras personas, así como con los elementos de alta tecnología a su alrededor. Se trata de un análisis más complejo, una vez más se necesita la participación de disciplinas fuera de HCI, centrándose en las fuerzas que llevan a los usuarios a comprometerse con la tecnología y la forma en que ésta está integrada en su mundo.

Esta etapa generalmente implica la realización de un estudio o análisis de usuarios de uno u otro tipo, personas en contextos particulares.

Por lo tanto, la Etapa 2 proporciona una comprensión de cómo los valores humanos de interés se producen a través de la interacción, teniendo en cuenta factores sociales, ambientales, etc. En esencia, esta etapa ofrece una rica mezcla de perspectivas y puntos de vista por la que se puede empezar a imaginar y esbozar distintas posibilidades tecnológicas.

4.1.3. Etapa 3: Diseño

La tercera etapa es principalmente el diseño o la fase creativa e implica reflexionar sobre cuáles son los objetivos que el diseño debe tener. Podría ser que se pretenda generar, apoyar o ampliar los valores humanos analizados en las etapas anteriores. También, podría ser que el objetivo del diseño sea profundizar el conocimiento de un conjunto de valores, un grupo de personas, o un dominio.

La etapa de diseño debe tener en cuenta la cultura y el lugar en el que se ubicará el diseño, sobre todo si se trata de ecosistemas físicos y sociales tan diferentes, como escuelas, estaciones, iglesias y plazas cívicas. ¿Qué significará para los diferentes habitantes? ¿Cómo puede el dispositivo o tecnología interactuar con otras tecnologías o dispositivos existentes?

Para esta etapa se deben tomar en cuenta el alcance, las implicaciones, los valores y las personas.

4.1.4. Etapa 4: Construcción

Esta fase puede incluir cualquier cosa, desde métodos de baja tecnología, prototipos en papel y bocetos, hasta sistemas robustos y de la más alta tecnología listos para pruebas de campo. Considerando que antes, gran parte de la construcción dentro de HCI ha sido esencialmente basada en software, por lo que implica el desarrollo, por ejemplo, de una interfaz para un sistema de escritorio o en un dispositivo móvil. A medida que se avanza hacia el año 2020, los sistemas pueden ser híbridos y requerir tanto el desarrollo de una interfaz de software como algo nuevo en hardware.

La complejidad de tales sistemas híbridos parece implicar que su construcción sería un proceso lento y laborioso. Sin duda sería si al hacerlo se requiere calidad en el diseño, pero para objeto de pruebas se puede trabajar con un prototipo de baja fidelidad que después será pulido. Actualmente existen varias herramientas y tecnologías que permiten a los investigadores llevar a cabo muchos tipos de tareas de construcción de forma rápida y sencilla.

Se sabe que la fase de construcción en cualquier programa de investigación no siempre será fácil y rápida; claramente dependerá de la meta. Además, siempre se pueden utilizar técnicas más clásicas de construcción, como simulaciones virtuales e incluso técnicas de Mago de Oz cuando no sea práctico construir un sistema completamente funcional. Con este enfoque, una función del sistema puede ser realizada por medio de un asistente que es invisible para el usuario.

Cualquiera que sea la técnica o la tecnología utilizada en esta etapa, el objetivo es construir algo, que permita de cualquier manera a los investigadores realizar pruebas acerca de la experiencia que están tratando de habilitar. Sólo entonces se podrá avanzar a la siguiente etapa, la evaluación.

4.1.5. Etapa 5: Evaluación

La quinta etapa consiste en la evaluación de lo que se ha construido. El trabajo de diseño no puede sólo representar una mejor respuesta al tipo de solución lograda en base a la investigación y los objetivos de diseño.

En esta etapa es mejor probar que suponer. Aquí, existen varias metodologías de HCI que pueden ser utilizadas, desde un focus group hasta pruebas en laboratorio. Una orientación en cuanto a qué tipo de técnica de evaluación es la adecuada, parte del objetivo de diseño o investigación.

Los resultados de la evaluación permitirán una mayor explicación acerca de los dispositivos o servicios diseñados o desarrollados para que estos funcionen en un contexto más amplio. Cualquiera que sea la consideración puede hacer que los investigadores revisen su diseño original. Es por esto que el proceso iterativo es muy importante.

4.2. Aplicación de DCU Extendida al problema identificado

4.2.1. Análisis de usuario

La fase de análisis de los usuarios es de gran relevancia dado que es en esta parte donde se proporcionan detalles sobre quién utilizará el producto. En esta fase se identifican roles y se definen las características del usuario, como son: el nivel de conocimiento, experiencia y habilidad con productos similares, su entorno, frecuencia de uso y dependiendo del tipo de producto, su hardware, software y tecnologías de apoyo que utilizará.

Cuando se trabaja en el análisis sin utilizar un proceso formal en el que se tenga en cuenta a los demás, se suele diseñar para uno mismo, es entonces que muchos proyectos dan origen a productos que están diseñados con base a las preferencias, habilidades y el entorno del propio diseñador.

4.2.1.1. Perfiles de grupos de usuarios en DCU

Dos de los aspectos más importantes de la usabilidad son las tareas de usuario y sus características y diferencias individuales [47]. En los perfiles de grupos de usuarios se describen las características de los usuarios del producto, es decir, de las personas que utilizan ese producto.

En el caso de un producto web, para un entorno de oficina, las características definidas para el grupo o los grupos de usuarios incluirían aspectos tales como:

- Responsabilidades y tareas laborales.
- Frecuencia de uso (diaria, una vez a la semana, una vez al mes o una vez al año).
- Factores demográficos.
- Hardware (portátil o de escritorio, resolución del monitor, etc.).
- Entorno (oficina individual, oficina compartida, equipo público compartido o si el usuario está en su casa).

- Software (sistema operativo, versión del navegador).
- Experiencia informática.
- Experiencia en aplicaciones web (si reconocen elementos en los que se puede hacer clic).
- Conocimiento de la tarea (saber, hasta qué punto comprenden la tarea que están realizando en contraposición a cómo comprenden la propia aplicación).

Cuando se desarrollan perfiles de grupos de usuarios, las fuentes de información que se utilizan para el desarrollo de un producto nuevo son, entre otras: el estudio general de mercado, los usuarios de productos de la competencia, las sesiones de grupo y las observaciones y entrevistas con posibles usuarios. Para introducir mejoras en un producto ya existente, los perfiles de grupos de usuarios pueden basarse en encuestas, sesiones de grupo, entrevistas contextuales y pruebas de usabilidad con la versión actual del producto [15].

Un primer punto al momento de desarrollar perfiles de grupos de usuarios, es identificar y definir los grupos, poniendo especial interés en esta parte, ya que puede haber varias categorías de usuarios para un mismo proceso o producto. Especialmente cuando se trabaja una aplicación vía web.

Cuando se trabaja con DCU, por lo general se crean varios perfiles de grupos de usuarios, pero se entiende que se delimita el proceso ya que no se desarrolla para todos los grupos, sólo para los que sean considerados principales en el proyecto en que se trabaja.

Para entender a los usuarios es necesario entender los procesos, capacidades y predilecciones que pueden asociarse a las tareas que desempeñan. Esto involucra un entendimiento y conocimiento de cosas como la memoria, visión, cognición, oído, tacto y habilidades motrices.

De acuerdo a Christine Faulkner [17], *“El sistema computacional necesita ser entendido en términos de lo que puede hacer por los usuarios y cómo podría comunicarse mejor con ellos.”*

De acuerdo con Jenny Preece [55], *“Un affordance es un término técnico acuñado por Donald Norman que se refiere a las propiedades de los objetos que indican qué clase de operaciones y manipulaciones se pueden llevar a cabo con dicho objeto.”*

Nuestro entorno de trabajo se definió en un inicio como la Universidad Tecnológica de la Mixteca (UTM) y se contempló enseguida como el Sistema de Universidades del Estado de Oaxaca (SUNEO). El área de estudio incluye a la gente involucrada con las áreas de Informática y Computación. La delimitación del espacio de trabajo permitió identificar tres grupos: Empresarios, Profesores y Alumnos.

Al haberse definido el grupo de usuarios con el que se trabajaría se realizó un sondeo para conocer sus inquietudes principales, las cuales nos servirían para crear la pregunta principal de la encuesta a aplicar, quedando de la siguiente manera:

Empresarios: ¿Cómo les gustaría trabajar con la academia y qué beneficios ofrece a los estudiantes?

Alumnos: ¿Te gustaría aprender trabajando en proyectos reales?

Academia: ¿Le gustaría evaluar a sus alumnos a través de un proyecto real?

El contenido de los cuestionarios aplicados puede verse en el Apéndice B.

4.2.2. Problemática

La falta de vinculación entre los sectores Académicos y Empresariales es uno de los factores que ocasiona que no existan lazos que permitan a ambos sectores realizar proyectos en conjunto con beneficios mutuos.

La información que se tiene en los sectores de la Industria es escasa, mientras que del lado de la Academia, se establece un vínculo de trabajo limitado a estancias profesionales, veranos de investigación, prácticas profesionales o bien para bolsas de trabajo.

La falta de comunicación entre estos sectores ocasiona una mala objetividad en la aplicación de los conocimientos, que al ser aplicados a un entorno real de trabajo, restan calidad al trabajo realizado, la cual se tiene que recuperar con iniciativas autodidactas o bien con cursos de capacitación por parte de las Empresas.

El tiempo de preparación, enseñanza, aprendizaje, la falta de proyectos reales, así como los espacios de aplicación hacen complejo el desarrollo y la integración de los sectores del ámbito académico y el empresarial.

4.2.3. Perfil de los usuarios

Se identificaron 3 grupos de usuarios:

- Empresarios, que vinculan y representan al sector empresarial,
- Profesores y Alumnos, que representan el sector de la academia.

Mediante el análisis del entorno de cada grupo es posible identificar los factores a considerar.

4.2.3.1. Grupo de usuarios: Alumnos

El primer grupo analizado corresponde a los alumnos, quienes participaron en entrevistas directas y respondieron cuestionarios que pueden consultarse en el apéndice B.

4.2.3.1.1. Contexto

VinculAE será un sistema que permitirá establecer un enlace para la cooperación en proyectos reales entre la Industria y la Academia. El diseño del sitio web deberá considerar la practicidad del manejo y presentación de la información que permita conocer a detalle las propuestas presentadas por el sector empresarial y que los participantes del sector académico, profesores, líderes de proyecto y los demás participantes comprendan y participen de ellas. Esto con el fin de darle una presentación que sea del agrado de la mayoría y a la vez sea funcional. El grupo de usuarios de alumnos incluye a un líder de proyecto, como usuario final de VinculAE, y está formado por universitarios con edades comprendidas entre los 18 y 25 años.

4.2.3.1.2. Factores demográficos

Dada la importancia que tiene la propuesta en nuestro sistema y acorde a los retos que plantea la dinámica global, los factores demográficos son un punto de gran importancia por los datos que se muestran a nivel nacional de nuestro Estado en comparación con el resto del país. Dada la

ubicación geográfica de Oaxaca, su nivel socioeconómico y de estudios, entre otros factores, se considera que VinculAE debería estar al alcance del mayor número posible de personas de todos los sectores de nuestra sociedad. A pesar de que solamente se considera trabajar con estudiantes de nivel superior para la resolución de los proyectos, la idea es poner al alcance de todos un instrumento de consulta rápida de proyectos, que encamine a los estudiantes y empresarios de manera fácil a la esencia de la propuesta.

En este sentido, de acuerdo con datos del censo del INEGI en el 2010 nuestro grupo de estudio, queda comprendido en el porcentaje de población de 15 a 29 años, que corresponde al 26.2 % del total de la población del Estado y al 26.8 % de la población nacional (ver Figura 4.3).



Figura 4.3: Censo nacional de INEGI 2010.

De forma adicional a esta información, los datos obtenidos por Universia México [68] confirman que:

- 31,6 % de la población mexicana es estudiante.
- 8 % de los estudiantes de México, pertenecen a la educación superior.
- 6,8 % de los estudiantes en educación superior cursa un posgrado.
- 51 % de los estudiantes de educación superior son mujeres .

De acuerdo con cifras proporcionadas por la Asociación Nacional de Universidades e Instituciones de Educación Superior (ANUIES), la población escolar total del Estado de Oaxaca en

2008-2009 correspondía a 40,199 (ver Tabla 4.1).

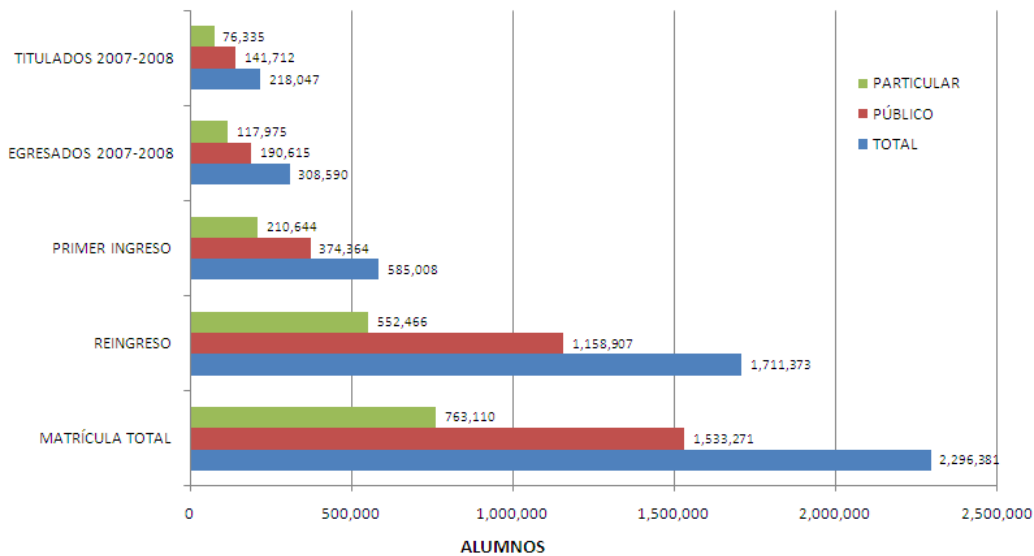
**POBLACIÓN ESCOLAR TOTAL, 2008-2009
(INGRESO, REINGRESO, EGRESADOS Y TITULADOS)**

| ENTIDAD FEDERATIVA | CIENCIAS AGROPECUARIAS | | CIENCIAS DE LA SALUD | | CIENCIAS NATURALES Y EXACTAS | | CIENCIAS SOCIALES Y ADMINISTRATIVAS | | EDUCACIÓN Y HUMANIDADES | | INGENIERÍA Y TECNOLOGÍA | | TOTAL | |
|--------------------|------------------------|-----|----------------------|------|------------------------------|-----|-------------------------------------|------|-------------------------|-----|-------------------------|------|-----------|-------|
| | ABS. | % | ABS. | % | ABS. | % | ABS. | % | ABS. | % | ABS. | % | ABS. | % |
| OAXACA | 2,362 | 5.9 | 4,265 | 10.6 | 1,473 | 3.7 | 15,876 | 39.5 | 1,489 | 3.7 | 14,734 | 36.7 | 40,199 | 2.6 |
| TOTAL NACIONAL | 51,857 | 3.4 | 179,346 | 11.7 | 44,063 | 2.9 | 572,828 | 37.4 | 88,282 | 5.8 | 596,895 | 38.9 | 1,533,271 | 100.0 |

Tabla 4.1: Población Escolar de Oaxaca en nivel superior, datos nacionales ANUIES.

De acuerdo con cifras proporcionadas por la ANUIES, la población escolar del Estado de Oaxaca, dividida por régimen en 2008-2009 correspondía a los datos que se muestran en la Figura 4.4.

Población escolar por régimen

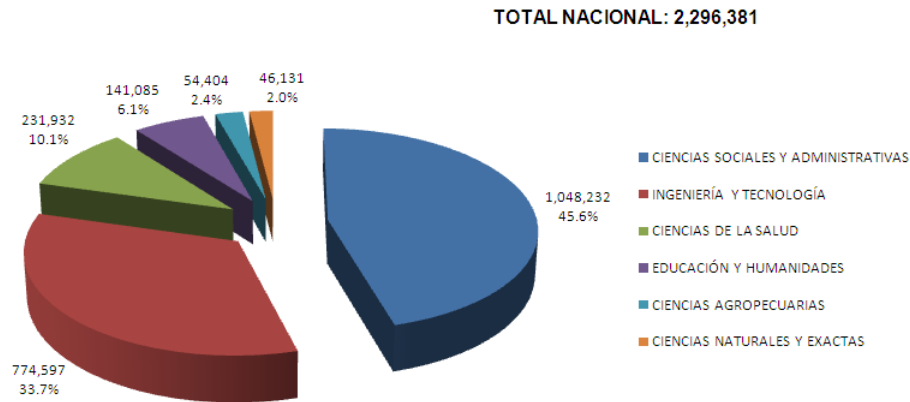


Fuente: elaboración propia con datos del Formato 911.9A. Ciclo escolar 2008-2009. La información corresponde exclusivamente a la modalidad escolarizada y a la Licenciatura Universitaria y Tecnológica.

Figura 4.4: Población escolar por régimen, 2008-2009.

La Figura 4.5 muestra la matrícula del total nacional de estudiantes de nivel superior segmentados por áreas de estudio, lo cual nos permite observar la inclinación de los estudios del sector universitario, dato importante al estudiar los factores demográficos.

Matrícula total por áreas de estudio



Fuente: elaboración propia con datos del Formato 911.9A. Ciclo escolar 2008-2009.
La información corresponde exclusivamente a la modalidad escolarizada y a la Licenciatura Universitaria y Tecnológica.

Figura 4.5: Matrícula total por áreas de estudio.

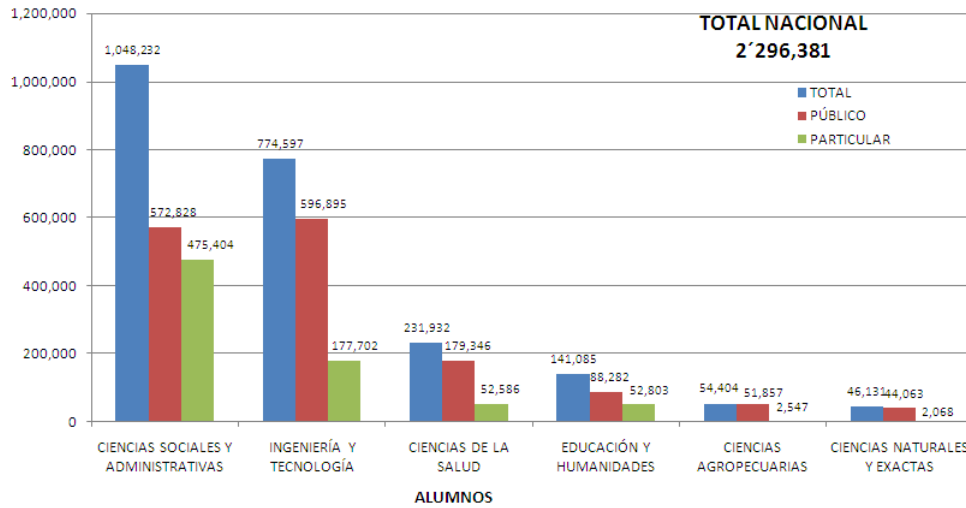
En base a estos datos de carácter nacional más adelante se formula el contexto que se refleja en nuestro Estado, aquí cabe mencionar algunos datos proporcionados actualmente por la Secretaría de Educación Pública (SEP) en su portal de Internet [62].

- Según la SEP 5 de cada 10 egresados de la universidad tienen trabajo en un área diferente a la que estudiaron.
- Sólo 10 % de las universidades en México incluyen en sus programas de estudio un plan de entrenamiento y destrezas específicas de la profesión, por lo que los egresados presentan dificultades para incorporarse al campo laboral.

En el informe que puede consultarse en [63], se destaca que tanto en el nivel medio superior como superior, la educación de calidad requiere estar vinculada con el sector productivo, por lo que se debe avanzar en este propósito que, dicen, traerá grandes beneficios para la juventud, dado que contará en su aprendizaje con la experiencia laboral en su formación. Se agrega que este vínculo educación-empresa no sólo se convierte en una plataforma para detonar el desarrollo de algunos sectores sino del país.

La gráfica de la Figura 4.6 muestra el total de la población escolar a nivel superior por régimen particular y público, al mismo tiempo se pueden observar las áreas con mayor concentración de estudiantes. Es notable que el área de Ciencias Sociales y Administrativas es la que más eligen los estudiantes universitarios en México.

Población escolar por áreas de estudio y régimen



Fuente: elaboración propia con datos del Formato 911.9A ciclo escolar 2008-2009. La información corresponde exclusivamente a la modalidad escolarizada y a la Licenciatura Universitaria y Tecnológica.

Figura 4.6: Población escolar por áreas de estudio y régimen.

La parte que corresponde al Estado de Oaxaca se resume en la Tabla 4.2.

MATRÍCULA DEL RÉGIMEN PÚBLICO POR ÁREAS DE ESTUDIO, 2008-2009

| ENTIDAD FEDERATIVA | CIENCIAS AGROPECUARIAS | | CIENCIAS DE LA SALUD | | CIENCIAS NATURALES Y EXACTAS | | CIENCIAS SOCIALES Y ADMINISTRATIVAS | | EDUCACIÓN Y HUMANIDADES | | INGENIERÍA Y TECNOLOGÍA | | TOTAL | |
|--------------------|------------------------|-----|----------------------|------|------------------------------|-----|-------------------------------------|------|-------------------------|-----|-------------------------|------|-----------|-------|
| | ABS. | % | ABS. | % | ABS. | % | ABS. | % | ABS. | % | ABS. | % | ABS. | % |
| OAXACA | 2,362 | 5.9 | 4,265 | 10.6 | 1,473 | 3.7 | 15,876 | 39.5 | 1,489 | 3.7 | 14,734 | 36.7 | 40,199 | 2.6 |
| TOTAL NACIONAL | 51,857 | 3.4 | 179,346 | 11.7 | 44,063 | 2.9 | 572,828 | 37.4 | 88,282 | 5.8 | 596,895 | 38.9 | 1,533,271 | 100.0 |

Tabla 4.2: Tabla por áreas de estudio del Estado de Oaxaca.

Considerando la información anterior, se procedió al análisis del objetivo de este trabajo que comprende a la población estudiantil del SUNEIO, el cual está conformado por la UTM, UMAR, UNISTMO, UNPA, UNSIS, UNSIJ y NovaUniversitas.

La Figura 4.7 muestra la distribución de la población académica del SUNEIO.

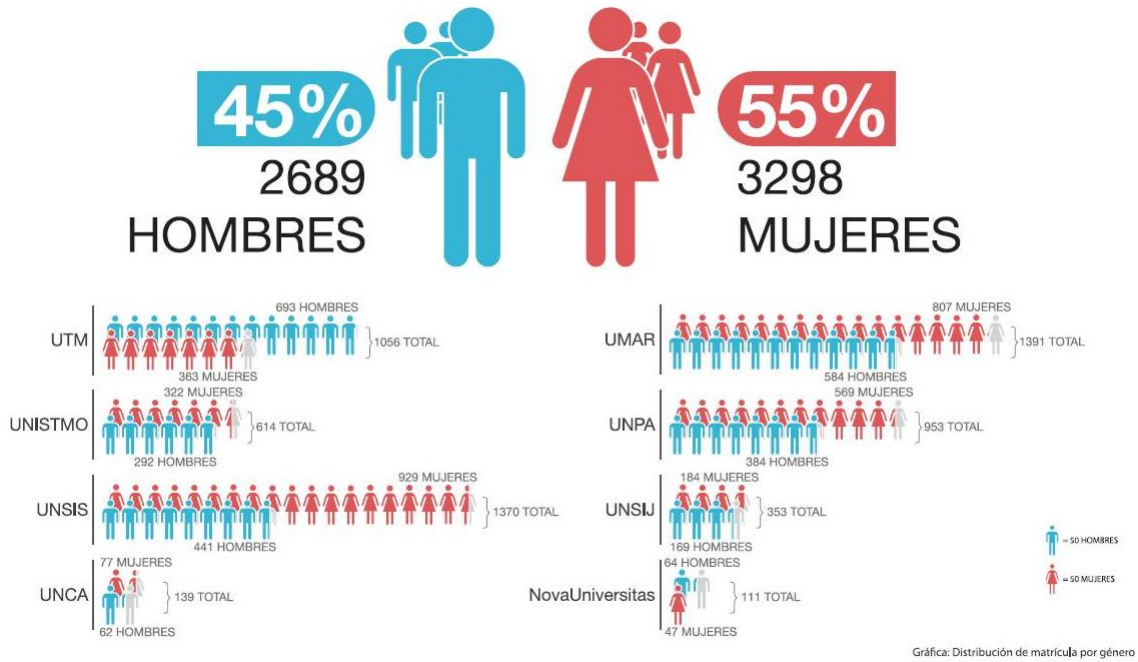


Figura 4.7: Distribución de matrícula por género de los estudiantes en el SUNEQ.

De este total, los alumnos en el SUNEQ se encuentran distribuidos en las diferentes carreras que se ofrecen (ver Figura 4.8).

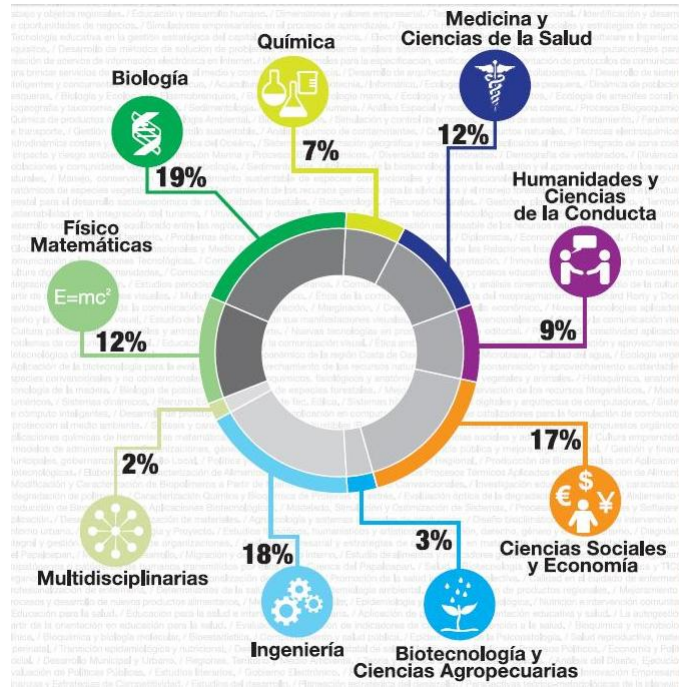


Figura 4.8: Distribución de estudiantes en las diferentes áreas del SUNEQ.

Datos estadísticos del SUNEIO en el 2010 muestran el porcentaje de egresados que lo conforman. Para los fines de esta tesis únicamente se contemplan aquellas universidades que incluyen la carrera de Computación o Informática. Dicha información se concentra en la Figura 4.9.



Figura 4.9: Egresados por carrera de las diferentes universidades del SUNEIO.

4.2.3.1.3. Entorno

Se realizó una encuesta entre los alumnos de séptimo a décimo semestre de la Universidad Tecnológica de la Mixteca para conocer sus inquietudes acerca de las materias que han cursado y su utilidad en proyectos reales después de realizar 2 periodos de prácticas profesionales en diferentes empresas y centros de investigación de nuestro país. Esta encuesta fue realizada a las generaciones 2010, 2011 y 2012. En dicha encuesta se obtuvieron los siguientes datos:

- El 89 % de los estudiantes encuestados afirma que deberían de realizarse practicas en clases que les ayuden a enfrentar las necesidades reales de las empresas.
- El 76 % de los estudiantes mencionó que debería existir un enlace entre la universidad y algunas empresas que les permitieran trabajar en proyectos durante el semestre y no esperar hasta la temporada de prácticas.
- El 91 % mencionó que de existir un sistema que les permita una vinculación entre la academia y la empresa, esta pueda utilizarse tanto en la Escuela como desde sus hogares.

Estas cuestiones permiten considerar el uso de los sistemas y del entorno estudiantil para manejar cuestiones más prácticas o con materias que demanda el sector empresarial; y que al mismo tiempo permita que los alumnos estén enterados de ofertas de proyectos para poder participar, y de la facilidad de uso y acceso del sistema que realice la vinculación.

4.2.3.1.4. Experiencia en trabajos reales con empresas

Para que un trabajo académico sea real debe tenerse un cliente real, ya que cada que se inicia un nuevo ciclo académico, se cuenta con varios proyectos con los que se trata de vincular a los alumnos de las diferentes carreras con destacadas necesidades de empresas reales, esto tanto en el campo de lo comercial como de índole cultural y social por las características de nuestro sistema universitario. Sin embargo, esto no siempre se logra por las condiciones que se tienen en la universidad, ateniéndose a recursos, tiempos y cargas de trabajo.

En nuestro sistema universitario se contemplan 2 temporadas de prácticas en el verano, las cuales representan el único acercamiento real de los estudiantes con las empresas dado que se pone en práctica el conocimiento adquirido en las aulas, y se relacionan con las exigencias del mundo real empresarial, del tipo de conocimiento que se demanda y de las características técnicas, intelectuales y demás que se requieren para sobrevivir en un mundo altamente competitivo que requiere estar al día con los avances tecnológicos.

4.2.3.1.5. Experiencia en proyectos escolares

En el salón de clases es común encontrar algunas dificultades para abarcar determinados ejes temáticos por, entre otras, diversas situaciones ajenas a este trabajo de investigación. Por lo tanto encontrar la mejor forma de enseñar un tema particular, a veces resulta algo complicado para el docente, sobre todo si se habla de un trabajo práctico, es entonces cuando se vuelve complicado analizar de qué manera los alumnos captarán mejor el tema, para tratar de abordar la brecha que existe entre la teoría y la práctica.

Los proyectos escolares que se realizan durante el semestre, la mayor parte de las veces son lo más cercano a un proyecto real de una empresa. Por lo tanto, este trabajo se convierte en el único medio para que los alumnos obtengan conocimiento de desarrollos reales basados en demandas de sus profesores (quienes representan al cliente) que miden su aprendizaje en determinada área a través de una calificación para la materia o el semestre, contando con la revisión y aprobación de

sólo la parte académica. Por lo tanto, se busca que el aprendizaje vaya más allá de los proyectos escolares.

4.2.3.1.6. Experiencia en el uso de sistemas vía web

Internet es utilizada diariamente por millones de personas en todos los ámbitos, sin embargo el interés de este trabajo se enfoca a los sectores de profesores, estudiantes universitarios y las empresas. De acuerdo al análisis del perfil de los estudiantes, éstos utilizan Internet en su hogar, en la escuela y en sitios públicos que les permiten el acceso.

Los estudiantes pasan un tercio del día donde obtienen información y datos de primera mano, a través de Internet, la cual pueden utilizar para sus clases. Por lo tanto, es necesario innovar el proceso de aprendizaje – enseñanza, debido a que actualmente los jóvenes universitarios se encuentran viviendo una etapa donde la tecnología está presente en el quehacer diario de todas las áreas.

Gracias a las nuevas tecnologías de la información y la comunicación (TIC), la mayoría de los estudiantes pueden comunicarse y colaborar con sus compañeros de clase, líderes de proyecto, profesores, tutores, etc., de una manera presencial, online o en una reunión virtual evitando de esta manera la situación espacio y tiempo.

Las practicas educativas no solo en nuestro medio sino alrededor del mundo han innovado dada la relación que existe entre la enseñanza – aprendizaje y el uso de Internet, lo que ha llevado a rediseñar los espacios y las formas de aprendizaje y enseñanza para ser utilizados de una manera coherente con la educación que se transmite, y utilizar todos los medios que se tengan al alcance como elementos vitales del proceso educativo, que permitan a los profesores darle mas herramientas a los alumnos antes de salir al campo laboral.

La relación actual de los jóvenes y el uso de internet hace cuestionar los roles, de los jóvenes alumnos y de los profesores, así como de los modelos de aprendizaje y el uso de tecnologías de información y de comunicaciones con las que se está acostumbrado a trabajar. Por lo tanto modificar el modelo educativo se vuelve una necesidad.

Los sistemas o aplicaciones vía web son utilizados comúnmente en la actualidad debido a lo práctico de los navegadores web como clientes ligeros, también la independencia del sistema operativo, así como la facilidad para realizar cambios y mantener las aplicaciones sin tener que distribuir o instalar software de actualización a los usuarios.

Del total de estudiantes encuestados, el 97 % ha tenido experiencia en el uso de sistemas vía web.

4.2.3.1.7. Frecuencia de uso: internet

En este momento las TIC modernas constituyen para todos los que tienen acceso a ellas, una herramienta privilegiada para el desarrollo, ya que contribuyen no sólo al mejoramiento social sino también al crecimiento económico de una nación. La parte del sector educativo de la cual partimos juega un papel muy importante, es desde donde se pretende crear la vinculación de los sectores involucrados para contribuir a ese proceso de crecimiento.

En este sentido, los datos estadísticos presentados por el INEGI en marzo de 2008, una cuarta parte de la población de seis años o más en México se declaró usuaria de Internet. Concretamente:

- El 77 % de los cibernautas mexicanos tienen menos de 35 años.
- La Tasa Media de Crecimiento Anual (TMCA) de usuarios de Internet es de 17.8 puntos porcentuales.
- Los hogares con Internet y los hogares con computadora aumentan a un ritmo anual del 14.5 %.
- Los datos presentados para estas fechas indicaban que en México había 34.9 millones de usuarios de Internet, y se esperaba que dicha cifra aumentara cada año entre 16 y 20 %.

Otros datos estadísticos del INEGI relacionados con los resultados de la Encuesta en Hogares sobre Disponibilidad y Uso de las Tecnologías de la Información, detallan que hasta el mes de abril del 2011, la cifra de usuarios de Internet era del 14.7 % mayor al dato reportado un año antes.

En relación al uso que se hace de Internet, el INEGI precisa que 61.9 % lo emplea para obtener información de carácter general, 60.9 % para realizar actividades de comunicación como mensajería, recibir o enviar correos electrónicos y 31.6 % la usa como apoyo a las actividades escolares.

En el 2012, durante la celebración del día mundial de Internet, el Comité de Investigación de la Asociación Mexicana de Internet (AMIPCI) presentó datos más completos sobre los hábitos de consumo de Internet en México.

El cálculo de los universos para presentar las estadísticas, se realizó a través de una metodología diseñada por este mismo organismo, que incluye datos estadísticos provenientes de distintas fuentes, entre las que se encuentran:

- El Consejo Nacional de Población (CONAPO)
- El Instituto Nacional de Estadística y Geografía (INEGI)
- La Comisión Federal de Telecomunicaciones (COFETEL) y
- El Departamento de Investigación Online de la Empresa (ELOGIA)

De acuerdo a la información presentada se contemplan más de 40 millones de usuarios de internet en nuestro país.

Tomando en cuenta estos datos estadísticos, la Figura 4.10 muestra en porcentajes la frecuencia del uso de Internet en México.

La Figura 4.11 ilustra la proporción de usuarios de computadora en cada Estado de la República Mexicana en relación a su población total: en ambos casos, usuarios y población se refiere a individuos de seis años o mayores.

En la estructura de la gráfica se agrupan de modo directo a las entidades en dos categorías respecto al promedio nacional (40 %). En el segmento externo se encuentran las 15 entidades con la mayor proporción de usuarios de computadora; sin embargo, incluso entre éstas, existe una diferencia significativa de casi 15 puntos porcentuales entre los valores extremos: Morelos y Sonora.

Por otra parte, el Estado de Oaxaca se encuentra dentro de las entidades con la menor participación en el indicador, compitiendo con San Luis Potosí, Veracruz, Michoacán, Guerrero y Chiapas, donde menos de un tercio de su población hace uso de una computadora.

Dentro de los datos obtenidos en nuestras encuestas, los estudiantes del SUNEО tienen una frecuencia de uso diaria de un 90 % y un acceso a internet del 100 % en sus universidades.

4.2.3.2. Grupo de usuarios: Profesores

4.2.3.2.1. Contexto

Este grupo de usuarios corresponde a los profesores del SUNEО; como usuario final de VinculAE, son profesores de las diferentes áreas académicas y de investigación que conforman el personal académico de las universidades. De acuerdo a datos del Departamento de Recursos Humanos, el promedio de edad oscila entre los 28 y 65 años.

La importancia de los profesores como grupo de usuarios es grande, por la misión que realizan para el presente y futuro de nuestro país. En pocos años los jóvenes que cursan una carrera en la universidad egresarán como profesionistas, y la formación teórica y práctica que tengan a través de la carrera será vital para su ingreso en el sector de la industria, de ahí la importancia de los profesores, quienes en nuestro contexto serán los que fomenten y realicen la vinculación de sus alumnos con la industria.

Actualmente son algunos profesores o los alumnos los encargados de emprender proyectos en forma independiente o subordinada con algunas empresas. Por lo que se trata de proponer soluciones a problemas más reales aplicados a las diferentes áreas y carreras de las instituciones, basados en un estilo de trabajo de metodología científica que permita planear y diseñar programas estratégicos con el fin de desarrollar sistemas que promuevan la calidad y productividad en los alumnos, para prepararlos en su encuentro con la industria; y son los profesores quienes ayudarán en la toma de decisiones, cuando decidan resolver conjuntamente con sus alumnos un problema real planteado por alguna empresa contribuyendo a su análisis y evaluación de proyectos educativos en los diferentes proyectos que desarrollen conjuntamente la academia y la industria.

Los profesores que forman nuestro grupo de usuarios trabajan en las Universidades Tecnológicas, estas son impulsadas por la federación desde 1991. Dichas universidades actualmente atienden al 88.7 % de los jóvenes que cursan el nivel Técnico Superior. Las Universidades Tecnológicas operan bajo un esquema de financiamiento compartido entre la federación y los gobiernos estatales. En nuestro Estado y en el sistema de nuestro contexto tienen una carga horaria de 8 horas diarias que se distribuyen en horas clase e investigación.

4.2.3.2.2. Factores demográficos

Dada la importancia que tiene la propuesta para el desarrollo de la educación en nuestro Estado, en nuestro sistema y acorde a los retos que plantea la dinámica global, los factores demográficos son un punto de gran importancia por los datos que se muestran a nivel nacional de nuestro Estado en comparación con el resto del país, nuestra ubicación geográfica, el nivel socioeconómico, de estudios, entre otros. Es este sentido, se considera que la educación debería de estar al alcance del mayor número de personas posible de todos los sectores de nuestra sociedad. A pesar de que esta tesis solamente se enfoca en trabajar con estudiantes de nivel superior, por el momento, la idea es poner al alcance de todos un instrumento de consulta rápida de proyectos, que encamine a los estudiantes y empresarios de manera fácil a transformar propuestas en proyectos.

La SEP, como responsable de la planeación y evaluación del Sistema Educativo Nacional (SEN), integra entre otros datos las estadísticas básicas que describen la situación actual y los avances en la prestación de los servicios educativos. Esta información se recaba directamente de cada una de las escuelas del país, tanto públicas como privadas. A través de la Unidad de Planeación y Evaluación de Políticas Educativas, mediante la Dirección General de Planeación, se ha realizado un proceso de integración de la información estadística, misma que permite elaborar el documento: *Sistema Educativo de los Estados Unidos Mexicanos, principales cifras, ciclo escolar 2010-2011*, donde se incluyen los datos estadísticos más significativos del Sistema Educativo Nacional [61]. Los datos generados constituyen la estadística oficial del sector federal y de los gobiernos estatales y es la base para llevar a cabo los procesos de planeación, programación, presupuestación y asignación de recursos, evaluación y rendición de cuentas del sector, entre otras actividades.

De aquí se obtiene lo siguiente:

- La matrícula escolarizada para el ciclo escolar 2010-2011 fue de 2,773,088 alumnos, que equivale al 27.9 % de la población de 19 a 23 años de edad (excluyendo al posgrado).
- En este ciclo escolar ingresó a la educación superior el 83 % de los egresados del ciclo anterior de la educación media superior.
- A su vez, la matrícula escolarizada y mixta para el mismo ciclo fue de 2,904,829 estudiantes que equivalen al 29.2 %. A esta cifra deben agregársele 166,814 estudiantes registrados en la modalidad no escolarizada.
- Con esta adición, la cobertura en la educación superior se elevó a 30.9 %.
- La educación superior, con casi 3 millones de alumnos, abarca el 8.7 % de la matrícula total del sistema educativo nacional.

Estos datos son el preámbulo para determinar que para la educación superior se cuenta con 308,061 docentes en el total nacional y, particularmente, para el Estado de Oaxaca 5,335 (ver Figura 4.12).

RESUMEN DEL SISTEMA EDUCATIVO NACIONAL

ALUMNOS, DOCENTES Y ESCUELAS
TOTAL

Concluye

| ENTIDAD FEDERATIVA | CAPACITACIÓN PARA EL TRABAJO* | | | EDUCACIÓN MEDIA SUPERIOR | | | EDUCACIÓN SUPERIOR | | |
|-----------------------|-------------------------------|---------------|--------------|--------------------------|----------------|---------------|--------------------|----------------|--------------|
| | Alumnos | Docentes | Escuelas | Alumnos | Docentes | Escuelas | Alumnos | Docentes | Escuelas |
| Aguascalientes | 21 586 | 533 | 66 | 44 733 | 3 239 | 160 | 36 729 | 4 505 | 43 |
| Baja California | 21 990 | 766 | 133 | 117 878 | 6 957 | 313 | 85 391 | 9 806 | 126 |
| Baja California Sur | 7 834 | 231 | 30 | 25 039 | 1 700 | 80 | 16 942 | 2 081 | 33 |
| Campeche | 30 517 | 472 | 63 | 29 972 | 2 130 | 112 | 24 307 | 2 114 | 65 |
| Coahuila | 34 008 | 852 | 123 | 96 895 | 7 426 | 416 | 81 779 | 8 714 | 131 |
| Colima | 8 056 | 586 | 158 | 25 532 | 1 726 | 97 | 19 074 | 2 261 | 50 |
| Chiapas | 43 709 | 1 504 | 454 | 190 591 | 9 915 | 723 | 69 989 | 7 220 | 166 |
| Chihuahua | 36 122 | 806 | 126 | 128 494 | 7 402 | 506 | 98 689 | 9 363 | 113 |
| Distrito Federal | 188 209 | 4 300 | 523 | 435 581 | 34 572 | 683 | 462 581 | 63 070 | 454 |
| Durango | 20 103 | 394 | 45 | 66 505 | 4 239 | 218 | 38 785 | 4 232 | 72 |
| Guanajuato | 96 329 | 689 | 114 | 174 384 | 11 291 | 783 | 100 323 | 12 309 | 226 |
| Guerrero | 14 532 | 726 | 168 | 114 005 | 6 256 | 341 | 54 769 | 4 058 | 130 |
| Hidalgo | 76 504 | 1 556 | 123 | 108 571 | 6 134 | 293 | 66 275 | 6 419 | 104 |
| Jalisco | 79 748 | 5 354 | 625 | 252 221 | 15 877 | 823 | 193 416 | 18 897 | 251 |
| México | 110 789 | 2 230 | 346 | 514 099 | 37 411 | 1 416 | 314 472 | 32 140 | 383 |
| Michoacán | 52 715 | 1 465 | 302 | 134 963 | 9 541 | 626 | 87 014 | 7 921 | 176 |
| Morelos | 30 767 | 613 | 57 | 69 668 | 5 379 | 314 | 43 907 | 5 667 | 102 |
| Nayarit | 15 393 | 386 | 82 | 42 438 | 2 795 | 272 | 30 468 | 2 753 | 46 |
| Nuevo León | 96 173 | 1 964 | 299 | 147 813 | 9 970 | 485 | 159 845 | 12 986 | 204 |
| Oaxaca | 53 746 | 1 650 | 173 | 136 582 | 7 323 | 633 | 59 512 | 5 335 | 109 |
| Puebla | 50 719 | 1 700 | 255 | 233 599 | 15 667 | 1 441 | 179 033 | 16 502 | 426 |
| Querétaro | 29 588 | 572 | 76 | 64 950 | 4 438 | 219 | 50 101 | 6 120 | 80 |
| Quintana Roo | 20 873 | 480 | 34 | 45 502 | 3 106 | 137 | 24 576 | 3 055 | 42 |
| San Luis Potosí | 36 830 | 1 191 | 106 | 93 965 | 6 275 | 433 | 60 004 | 5 689 | 87 |
| Sinaloa | 126 948 | 1 710 | 128 | 121 828 | 8 005 | 350 | 84 961 | 7 236 | 115 |
| Sonora | 42 595 | 747 | 90 | 106 065 | 6 466 | 316 | 89 205 | 7 772 | 125 |
| Tabasco | 23 856 | 759 | 188 | 97 320 | 5 409 | 278 | 65 796 | 5 230 | 65 |
| Tamaulipas | 31 822 | 677 | 112 | 119 042 | 7 393 | 356 | 106 683 | 9 474 | 179 |
| Tlaxcala | 11 291 | 450 | 91 | 47 600 | 3 198 | 188 | 26 220 | 2 625 | 52 |
| Veracruz | 108 029 | 3 292 | 544 | 275 008 | 18 891 | 1 665 | 157 410 | 13 118 | 342 |
| Yucatán | 14 916 | 903 | 172 | 73 525 | 5 070 | 247 | 57 487 | 5 370 | 118 |
| Zacatecas | 13 382 | 370 | 82 | 53 160 | 3 068 | 186 | 35 570 | 4 019 | 74 |
| TOTAL NACIONAL | 1 549 679 | 39 928 | 5 888 | 4 187 528 | 278 269 | 15 110 | 2 981 313 | 308 061 | 4 689 |

* Cifras estimadas

Figura 4.12: Resumen nacional de datos de docentes.

La educación superior en el país se distribuye de la siguiente manera:

- Profesional Asociado o Técnico Superior, que constituye el 3.8 %,
- Licenciatura, con el 89.2 % y
- Posgrado, que representa el 7.0 %

Sin una educación superior de calidad no será fácil ofrecer a los jóvenes más y mejores oportunidades. Una oferta educativa de calidad es también un medio indispensable para lograr una inserción ventajosa de México en la economía del conocimiento y en las cadenas de valor de la competitividad mundial.

Los esfuerzos dirigidos a mejorar la calidad de los servicios que brindan las instituciones educativas se han venido apuntalando desde hace al menos dos lustros. Entre las diversas medidas instrumentadas sobresalen los programas de fortalecimiento institucional, de profesionalización del personal académico, de formación y fortalecimiento de cuerpos académicos y la integración de redes de investigación. Los resultados de esos esfuerzos son alentadores como se describe en los siguientes puntos:

- Existen casi 330 mil docentes de educación superior en el país (16.3 % más que en 2006). De ese total, uno de cada cuatro (81,550) son de tiempo completo.

- La gran mayoría de los Profesores de Tiempo Completo (PTC) trabaja en instituciones públicas (87 %, que es un porcentaje similar al de 2006).
- Alrededor de 70 de cada 100 PTC cuentan con estudios de posgrado en las instituciones públicas; entre las particulares, la relación es de 65 de cada 100.
- El Sistema Nacional de Investigadores (SNI) está formado por casi 16,600 académicos (37.2 % más que en 2006).
- Hay poco más de 16,600 profesores con perfil PROMEP (75 % más que en 2006).

No sólo aumentó el número de los cuerpos académicos, sino que ahora es mayor el peso de aquellos ya consolidados o en proceso de consolidación (de 24.2 % a 46.6 %).

La Figura 4.13 muestra que el mayor porcentaje se observa en la licenciatura, que es nuestro grupo de estudio, donde los profesores tienen un rol muy importante con la universidad para llevar a la práctica el conocimiento que adquieren los jóvenes, ya que con la información recabada es posible determinar que la gran mayoría de los empresarios afirman que a los jóvenes universitarios en nuestro país no se les prepara lo suficientemente bien para el mundo laboral, que es cuestión de tiempo que la Formación Profesional impulse el desarrollo de nuestro Estado y país, a un ritmo mucho más rápido.

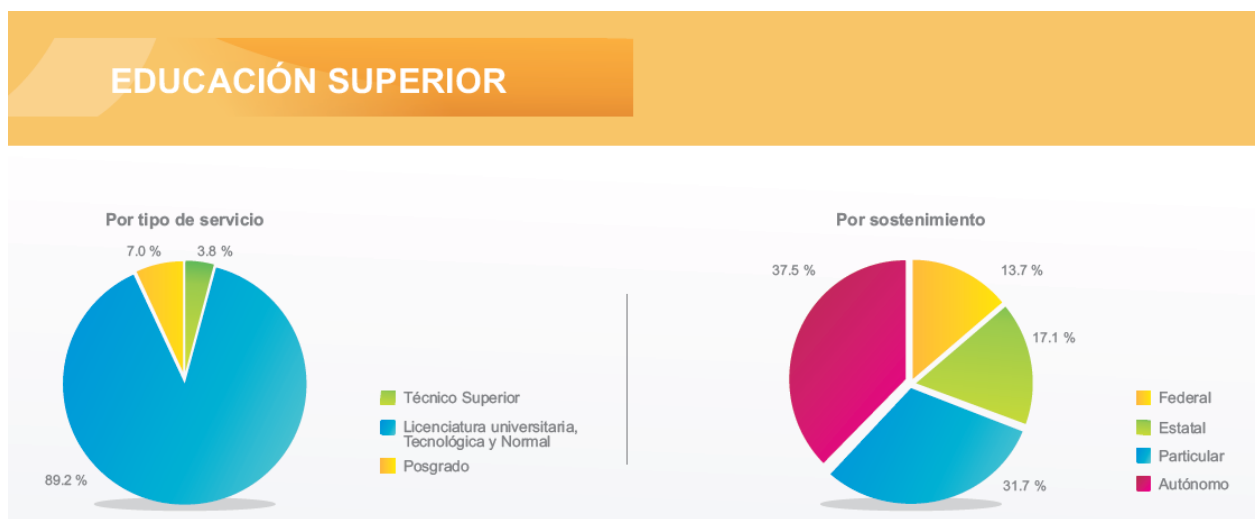


Figura 4.13: Distribución de la educación superior.

De acuerdo con datos obtenidos por el Consejo Nacional de Población, organismo que depende de la Secretaría de Gobernación:

- En 1978, México tenía 623 mil maestros, cifra que tres décadas después aumentó a un millón 730 mil docentes.
- El 61 % de los profesores son de sexo femenino (un millón 60 mil),
- El 39 % de los profesores (670 mil) son de sexo masculino.
- El 22.8 % de los maestros mexicanos son menores de 30 años,

- El 45.4 % que representa la mayor proporción se ubica entre los 30 y 44 años.
- El grupo de 45 a 59 años representa el 28.4 %.
- Los profesores de más de 60 años representan el restante 3.4 % del total.
- Los profesores de educación superior (universidad y posgrado) representan el 16.4 % del total de la planta educativa nacional.
- Al 2009 en México había 5.2 maestros por cada 100 alumnos, aunque en entidades como el Distrito Federal, Colima, Nayarit y Yucatán la relación se encuentra entre 6.5 y 5.8, en Chiapas, Querétaro, Guanajuato y Tabasco la proporción es de 4.6 por cada cien alumnos.

La Figura 4.14 muestra la distribución del personal académico por áreas de adscripción. Con la información presentada se nota que más del 90 % de los profesores se encuentran en institutos de investigación, y es ahí donde se busca respaldar los proyectos de colaboración.



Figura 4.14: Distribución del personal académico en el SUNEI.

La Figura 4.15 muestra la distribución de los profesores de acuerdo a las áreas de investigación con las que cuenta el SUNEI.

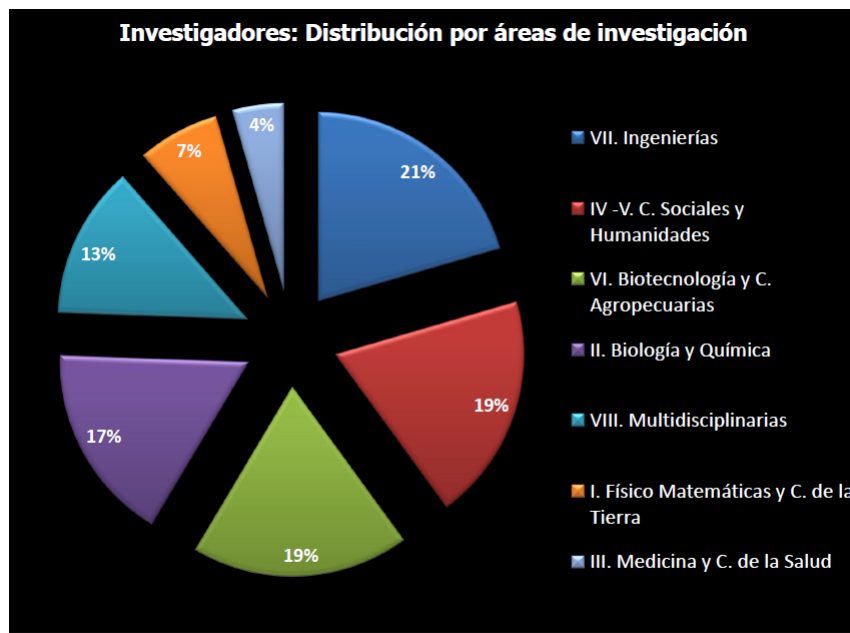


Figura 4.15: Distribución del personal académico del SUNEО por áreas.

4.2.3.2.3. Entorno

Se realizó una encuesta entre el personal académico de las universidades que comprenden el SUNEО, principalmente de las áreas de Ingeniería y Ciencias Sociales y Humanidades, para conocer las inquietudes que presentaban sus alumnos después de realizar sus prácticas profesionales y que son su primer acercamiento al área laboral en un entorno real, considerando fundamentalmente el modelo de enseñanza aprendizaje que se utiliza en el sistema. Esta encuesta fue realizada en el periodo de Octubre de 2011 a Septiembre 2012 y se obtuvieron los siguientes datos:

- El 93 % de los profesores encuestados afirma que sus alumnos han realizado prácticas profesionales, por lo tanto han tenido ya un acercamiento al campo laboral real, donde han tenido la oportunidad de poner en práctica los conocimientos adquiridos en las clases.
- El 81 % afirma que sus alumnos después de regresar de las prácticas profesionales, manifiestan sus inquietudes sobre los requerimientos en el campo profesional.
- El 91 % afirma que le gustaría colaborar en algún proyecto con sus alumnos que se vincule con una práctica real en alguna empresa.
- El 23 % conoce algún programa de vinculación entre la academia y el sector empresarial.
- El 63 % considera que el modelo de aprendizaje es el adecuado para los estudiantes.
- Al 94 % de los encuestados les gustaría evaluar a sus alumnos mediante un proyecto real.

Los resultados obtenidos nos hacen considerar que debe de existir una vinculación más directa entre el entorno estudiantil y el sector industrial.

4.2.3.2.4. Experiencia en trabajos reales con empresas

De forma similar que para el perfil de estudiantes, las 2 temporadas de prácticas en el verano representan el único medio real para que los estudiantes apliquen los conocimientos que los profesores transmiten en sus clases. También se observó que la mayoría de los profesores no tienen experiencia en el sector industrial y se integran directamente a una línea de investigación o académica. En menor proporción, algunos profesores que han tenido experiencia en el sector de la industria o han formado algunas empresas se han integrado para incursionar en la vida académica.

4.2.3.2.5. Experiencia en proyectos escolares

En los proyectos escolares que se realizan como parte de la evaluación práctica de un curso, los maestros generan equipos e identifican alumnos como líderes de proyectos, con los cuales generan propuestas de proyectos. Sin embargo, la mayoría de las veces éstas solamente quedan como propuestas académicas que el profesor evalúa y aprueba para la calificación práctica del curso.

En los proyectos de fin de semestre el profesor evalúa a los alumnos, tanto en trabajo en equipo como de manera individual, en esta parte del proceso el académico funge como Asesor, Revisor, Analista y evaluador de dichos proyectos.

4.2.3.2.6. Experiencia en el uso de sistemas vía web

En esta práctica de la docencia y para motivar e innovar en el proceso de aprendizaje – enseñanza los académicos interactúan con sistemas web de diversa índole, ya sea para consultas, o para alimentar información a varios sistemas a los que tienen acceso.

4.2.3.2.7. Frecuencia de uso: Internet

En la parte académica del SUNEО los profesores – investigadores cuentan con un equipo de computo y acceso a internet, lo cual deriva en una frecuencia de uso constante.

4.2.3.3. Grupo de usuarios: Empresarios

4.2.3.3.1. Contexto

El grupo de usuarios identificado como empresarios, son personas que tienen un puesto directivo en alguna empresa ya sea propia o para la cual trabajan como parte directiva de la misma. La edad de este grupo de usuario oscila entre los 30 y 50 años.

4.2.3.3.2. Factores demográficos

Las micro, pequeñas y medianas empresas constituyen la columna vertebral de la economía de nuestro país tanto por su alto impacto en la generación de empleos como en la producción nacional.

De acuerdo con datos del INEGI, en México existen aproximadamente 4 millones 15 mil unidades empresariales, de las cuales 99.8 % son medianas empresas y son las que generan el 52 % del Producto Interno Bruto (PIB) y el 72 % del empleo en el país. De ahí la importancia de estas entidades y de que exista un enlace con la educación. Por lo tanto es importante generar vínculos para mejorar el entorno económico y generar un trabajo conjunto en el cual sean las empresas quienes apoyen directamente las propuestas académicas. Lo anterior con el propósito de crear condiciones que contribuyan al desarrollo y practicas académicas así como al refinamiento y consolidación de las mismas.

En cuanto a datos necesarios para el análisis del contexto explorado en esta tesis, específicamente en nuestro Estado, es posible utilizar los datos que contemplan información de la industria Oaxaqueña y que provienen de CANACINTRA y del mismo INEGI.

La Figura 4.16 muestra las entidades económicas consideradas por el INEGI en el censo económico del 2009, se toman en cuenta 236,465 unidades para el caso particular del Estado de Oaxaca.

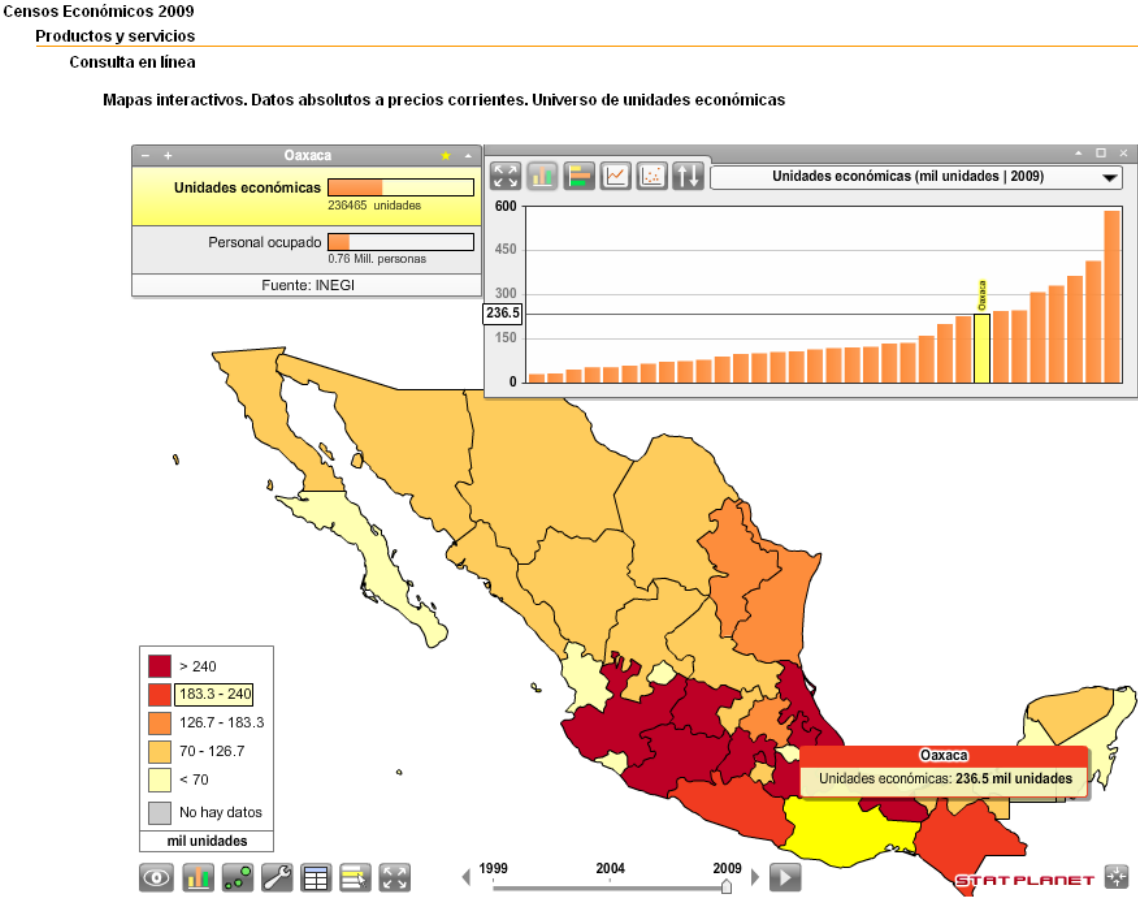


Figura 4.16: Datos del Estado de Oaxaca dentro de las entidades económicas del País.

Por otro lado, la Figura 4.17 resume las entidades económicas consideradas a nivel nacional distribuidas en los diferentes sectores. Para el análisis en nuestro contexto, es posible observar que el Estado de Oaxaca cuenta con un total de 7.7 % del total de unidades económicas en la República.

Censos Económicos 2009

Estructura porcentual de las unidades económicas, personal ocupado total y producción bruta total en los sectores Minería, Manufacturas, Comercio y Servicios, según la integración en empresas o en establecimientos por entidad federativa 2008

| Entidad federativa | Unidades económicas | | Personal ocupado total | | Producción bruta total | |
|---------------------------------|--------------------------------|------------------------|-------------------------------|-----------------------|-------------------------------|-----------------------|
| | Integradas por establecimiento | Integradas por empresa | Integrado por establecimiento | Integrado por empresa | Integrada por establecimiento | Integrada por empresa |
| Total nacional | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Aguascalientes | 1.1 | 1.1 | 1.2 | 1.1 | 1.3 | 0.7 |
| Baja California | 2.1 | 2.1 | 3.7 | 3.3 | 2.5 | 1.9 |
| Baja California Sur | 0.6 | 0.6 | 0.7 | 0.6 | 0.4 | 0.3 |
| Campeche | 0.8 | 0.8 | 0.8 | 0.7 | 7.8 | 7.7 |
| Coahuila de Zaragoza | 2.2 | 2.1 | 3.1 | 2.8 | 4.6 | 3.0 |
| Colima | 0.7 | 0.7 | 0.6 | 0.5 | 0.3 | 0.2 |
| Chiapas | 3.5 | 3.5 | 2.1 | 1.9 | 2.5 | 1.5 |
| Chihuahua | 2.4 | 2.4 | 4.1 | 3.9 | 2.4 | 2.2 |
| Distrito Federal | 10.3 | 10.3 | 14.7 | 19.2 | 13.6 | 33.6 |
| Durango | 1.2 | 1.2 | 1.2 | 1.2 | 0.8 | 1.0 |
| Guanajuato | 4.9 | 4.9 | 4.7 | 4.4 | 4.4 | 2.5 |
| Guerrero | 3.4 | 3.5 | 2.1 | 1.9 | 0.5 | 0.4 |
| Hidalgo | 2.2 | 2.2 | 1.7 | 1.6 | 2.3 | 0.6 |
| Jalisco | 7.1 | 7.1 | 7.6 | 7.5 | 5.5 | 4.7 |
| México | 12.4 | 12.6 | 10.3 | 9.6 | 9.0 | 7.9 |
| Michoacán de Ocampo | 4.8 | 4.8 | 3.1 | 2.9 | 1.5 | 1.3 |
| Morelos | 2.1 | 2.2 | 1.6 | 1.5 | 0.9 | 1.2 |
| Nayarit | 1.0 | 1.0 | 0.8 | 0.7 | 0.3 | 0.2 |
| Nuevo León | 3.4 | 3.4 | 6.0 | 7.2 | 8.2 | 9.2 |
| Oaxaca | 3.8 | 3.9 | 2.1 | 1.9 | 1.9 | 0.4 |
| Puebla | 5.8 | 5.9 | 4.4 | 4.0 | 3.3 | 2.7 |
| Querétaro | 1.5 | 1.5 | 1.9 | 1.7 | 2.3 | 1.8 |
| Quintana Roo | 1.0 | 1.0 | 1.6 | 1.3 | 0.9 | 0.7 |
| San Luis potosí | 2.1 | 2.1 | 2.0 | 1.8 | 2.0 | 1.4 |
| Sinaloa | 2.1 | 2.1 | 2.1 | 2.2 | 1.1 | 1.0 |
| Sonora | 2.2 | 2.2 | 2.8 | 2.5 | 2.8 | 1.8 |
| Tabasco | 1.4 | 1.3 | 1.2 | 1.0 | 4.9 | 3.5 |
| Tamaulipas | 2.7 | 2.7 | 3.4 | 3.1 | 4.0 | 2.3 |
| Tlaxcala | 1.3 | 1.4 | 0.9 | 0.8 | 0.6 | 0.3 |
| Veracruz de Ignacio de la Llave | 6.0 | 6.1 | 4.7 | 4.3 | 5.9 | 2.8 |
| Yucatán | 2.3 | 2.3 | 2.0 | 1.8 | 0.9 | 0.7 |
| Zacatecas | 1.3 | 1.3 | 0.9 | 0.8 | 0.6 | 0.5 |

INEGI. Las empresas en los Estados Unidos Mexicanos. censos económicos 2009. 2012

Figura 4.17: Unidades económicas en el país.

En información más detallada proporcionada por CANACINTRA en el 2009, en el Estado de Oaxaca se encontraban afiliadas 394 empresas, de las cuales el 90 % tenía su localización en los valles centrales.

4.2.3.3.3. Entorno

Se realizó una encuesta entre algunos empresarios al interior del Estado y de diferentes sectores, con el fin de conocer sus inquietudes en relación a dos temas: (1) su impresión de las áreas académicas y de investigación que las universidades del Estado ofertan y (2) su interés por cooperar en el desarrollo conjunto de proyectos (industria/universidad) y su posible apoyo en proyectos vinculados a las áreas académicas. Esta encuesta fue realizada en el periodo de diciembre del 2011 a septiembre del 2012. En dicha encuesta se obtuvieron los siguientes resul-

tados:

- El 65 % afirma que su empresa recibe alumnos para realizar practicas profesionales.
- El 86 % considera que sí trabajaría en algún proyecto vinculado con alguna entidad educativa.
- El 76 % afirma que su empresa no trabaja en ningún programa de vinculación con el sector educativo.
- El 73 % considera que los conocimientos de los estudiantes en sus empresas es adecuado para desempeñar un trabajo.

4.2.3.3.4. Experiencia en trabajos reales con Instituciones educativas

Un punto importante a destacar en este trabajo es que los directivos de la empresa argumentan que regularmente no tienen contemplado dedicar tiempo a proyectos vinculados con el sector académico. Consideran a los estudiantes de universidad o egresados como mano de obra barata y los altos mandos delegan a los mandos medios los aspectos relacionados con la vinculación; este grupo pocas veces tiene capacidad de decisión para realizar dichas actividades.

De acuerdo a las empresas encuestadas, esta forma de trabajo que es altamente cuestionada, puede afectar la impresión que las universidades y/o los alumnos puedan tener del sector empresarial.

4.2.3.3.5. Experiencia en proyectos escolares

Algunas de las actividades que se han realizado con algunas empresas son de carácter escolar y están relacionadas con un proyecto durante el curso o proyectos de fin de semestre. Esta práctica se da en su mayoría en áreas administrativas o relacionadas con las Ciencias Sociales y Humanidades. Resultando en menor parte la inclusión de las áreas de Ingeniería.

4.2.4. Perfil de Usuario

- En relación a la academia, las personas que utilizan este tipo de herramientas por lo general necesitan trabajar en proyectos reales, diferentes a los ya comunes en las instituciones educativas. En el caso de las empresas relacionadas con la industria, es necesario encontrar personal que entienda su filosofía de trabajo o evaluar los conocimientos prácticos del grupo de personas que potencialmente trabajarán para ellos.
- La manera en que estas dos entidades se relacionan, es frecuentando las bolsas de trabajo de las universidades o consultando sitios o blogs de tecnología, donde se formulan proyectos en busca de soluciones, o bien solicitando colaboraciones directas con las instituciones o vía profesores que trabajan en el área de interés y que tienen un acercamiento con las empresas.
- Las empresas extranjeras ofrecen estancias de trabajo mientras que los centros de investigación ofrecen estancias de verano o trabajar en algún proyecto con investigadores, aunque

en su gran mayoría es investigación y no desarrollo de proyectos reales en donde se pongan en práctica los conocimientos técnicos y logren desarrollar las habilidades del estudiante.

- Las empresas nacionales consolidadas, pocas veces trabajan con las universidades, a menos que éstas tengan un prestigio reconocido en el área de trabajo que requieran. Regularmente esto por problemas de tiempo, recursos económicos y el trabajo a distancia, que normalmente bloquea en parte el acceso a la práctica.
- En el área de Computación los conocimientos básicos pocas veces se requieren para trabajar en proyectos reales, regularmente se demanda la experiencia en el desarrollo de soluciones complejas que requiere la industria, la cual está en constante evolución, debido al mundo complejo en el que tecnológicamente vivimos.
- El conocimiento del idioma es importante para trabajar en áreas computacionales, principalmente en el área de requisitos, ya que un mal entendimiento de las propuestas origina errores en la solución final. Lo contrario ocurre en el área de desarrollo dado que la mayoría de las herramientas están desarrolladas para el idioma inglés y el estudiante se relaciona más rápidamente por su constante uso.
- En relación al nivel de entendimiento en el área de la Computación, el 100 % de los involucrados en los proyectos, tanto del área académica como del área empresarial afirman conocer el significado de las palabras utilizadas en el argot computacional para describir proyectos.
- La academia y la industria utilizan básicamente la misma tecnología, computadoras, teléfonos móviles, acceso a Internet. La única diferencia reside en la capacidad de las mismas y el modo de acceso, mientras que algunos lo hacen en equipos portátiles medianos y con una conexión limitada a Internet, los otros tienen grandes recursos informáticos a su disposición.

4.2.5. Objetivos

Los objetivos identificados se resumen como:

- Crear un vínculo entre la academia y la industria que ayude a fortalecer la preparación de los estudiantes antes de salir al campo laboral.
- Con el apoyo de la tecnología eliminar las barreras que existen entre la academia y la industria para mejorar la experiencia de los estudiantes al trabajar en proyectos reales.
- Realizar una conexión entre los estudiantes y profesores con las empresas e industrias del ámbito computacional, en especial del desarrollo de software que ayude a impulsar el desarrollo de la economía de nuestro Estado y país, con la nueva generación de empleos.
- Ganarse la confianza de los empresarios a través de un trabajo conjunto y bien realizado que cumpla con las expectativas generadas para que el empresario logre mejorar su experiencia de trabajo con el sector académico.
- Motivar el crecimiento, diversificación y la competitividad de los servicios académicos ofrecidos en las áreas tecnológicas en el Estado de Oaxaca.

- Generar interés en las empresas para que ofrezcan proyectos a las universidades, cumpliendo con las reglas establecidas para un proyecto vinculado.

4.2.6. Prototipos

Los prototipos son una representación limitada de un producto o sistema, permite tanto a los desarrolladores como a los usuarios probarlo en situaciones reales o explorar su uso, creando así un proceso de diseño de iteración que genera calidad. Son útiles para comunicar, analizar y definir ideas entre ambas partes. Un prototipo puede ser cualquier cosa, desde un trozo de papel con dibujos sencillos hasta un complejo sistema de software.

Son particularmente útiles en las fases iniciales del desarrollo, durante el diseño conceptual apoyan la evaluación del producto o sistema, clarifican requisitos de usuario y definen alternativas. Es común que los usuarios no sepan lo que quieren, pero cuando ven algo y utilizan prototipos, pronto saben lo que no quieren.

Pero, ¿Porqué desarrollar prototipos? El fin es obtener una retroalimentación del diseño, además ahorra tiempo, dinero y esfuerzo, permite experimentar con diseños, corrige problemas antes que se codifique y la parte principal mantiene el diseño centrado al usuario.

La construcción de prototipos se realizó mediante el desarrollo de versiones preliminares de lo que sería la parte principal del portal. De manera intencional se inició con una versión incompleta la cual se fue aumentando de acuerdo a las entradas obtenidas de la realimentación con los usuarios.

Los prototipos presentados fueron de alta fidelidad por lo que su nivel de detalle se asemeja al producto final. De esta forma, las distintas versiones se iban haciendo más detalladas, más completas y correctas que la anterior.

Una característica principal de estos prototipos es que fueron desarrollados en un tiempo medio que no demandó demasiados recursos en tiempo y de sistema; de esta forma fue posible obtener otros requerimientos que fueron considerados en la versión final para hacerla más funcional y flexible. Esto permitió considerar uno de los factores importantes en la etapa de desarrollo: minimizar el costo de modificación.

Así, una ventaja de los prototipos es que permiten el desarrollo gradual de un sistema.

A continuación se presentan los prototipos presentados durante el desarrollo.

El primero de los prototipos tenía un diseño parecido al de los portales actuales del gobierno, característica que pronto fue identificada por los usuarios durante las pruebas.

La Figura 4.18 muestra la imagen que el usuario ve al ingresar al sitio web donde podrá iniciar sesión o autenticarse con los datos asignados.



Figura 4.18: Pantalla inicial del primer prototipo.

Después de ingresar al sistema VinculAE se muestra la pantalla principal como en la Figura 4.19, donde se observan opciones de consulta y en el marco principal algunos proyectos vigentes que pueden ser de interés al usuario.



Figura 4.19: Pantalla general del primer prototipo.

El segundo de los prototipos muestra como inicio una pantalla en la cual se solicita al usuario su Identificación y contraseña para ingresar al sistema como se observa en la Figura 4.20. En esta opción se agrega un botón de cancelar no considerada en el primer prototipo. Este diseño consiste en una apariencia que mezcla lo industrial con lo institucional.

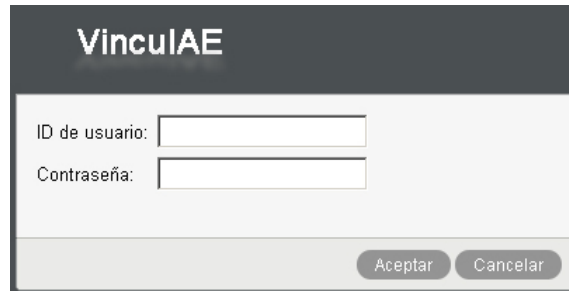


Figura 4.20: Pantalla inicial del segundo prototipo.

Al ingresar al sistema se muestra la pantalla principal como se observa en la Figura 4.21 cuya apariencia es totalmente diferente a la del primer prototipo, en este nuevo prototipo se presenta una breve descripción de lo que es el sistema VinculAE y las diferentes opciones de consulta que tienen los usuarios.



Figura 4.21: Pantalla general del segundo prototipo.

La Figura 4.22 muestra en el marco principal la pantalla de captura que se despliega al momento en que un empresario desea darse de alta en el sistema para poder ingresar los datos de una propuesta de proyecto que desee realizar de manera conjunta con el sector académico, quien tiene acceso a la información almacenada en VinculAE.

Este prototipo es el primero en incluir una pantalla de captura de datos similar al que ocupan algunos sistemas comerciales para agregar los datos al sistema, en la pantalla presentada se permite al usuario agregar los datos de una empresa.

La pantalla mostrada se conforma por campos de texto, donde el proceso para introducir los datos es básicamente muy simple. Se considera que la mayoría de los usuarios se encuentra familiarizado con la interfaz del sistema.

En la interfaz del prototipo, quizá los elementos más importantes a tomar en cuenta son los botones de ayuda, regresar, siguiente y cancelar.



Figura 4.22: Pantalla de captura para dar de alta un proyecto.

La pantalla de inicio del tercer prototipo solicita la identificación del usuario y su password para poder ingresar al sistema, tal y como se observa en la Figura 4.23. El prototipo ya presenta una mejora considerable, la cual fue realizada en base a los comentarios y sugerencias de usuarios previos.



Figura 4.23: Pantalla inicial del tercer prototipo.

La Figura 4.24 muestra la pantalla principal del sistema, donde se observa en la parte superior un menú de opciones que el usuario puede explorar, en esta pantalla el usuario puede realizar una consulta o búsqueda de proyectos.



Figura 4.24: Pantalla general del tercer prototipo.

En la pantalla que muestra la Figura 4.25 se observa lo que el usuario visualiza al realizar una búsqueda, una caja de texto donde puede teclear lo que desea buscar, esta opción de búsqueda se

mantiene de manera constante y funciona para todos los perfiles de usuario.

Usuario usuario123@vinculae.utm.mx
Contraseña ***** Ingresar
Regístrate ahora ¿Olvidaste tu contraseña?

Busqueda Buscar

Inicio Quienes Somos Que es VinculAE Consultas Servicio Contacto Sitios de Interés

Universidad Tecnológica de la Mixteca, Carretera a Acatlilma, Huajuapán de León, C.P. 69000 Oaxaca, México.
Tel. 9535320214 Lada: 01-800-0000-UTM webmaster@mixteco.utm.mx

Figura 4.25: Pantalla de búsqueda del tercer prototipo.

La Figura 4.26 muestra la pantalla que observa el usuario empresario al momento de dar de alta los datos de un proyecto, se pueden ver los diferentes campos que tiene que rellenar y al final una opción de guardar para que los datos se almacenen en el sistema.

Usuario usuario123@vinculae.utm.mx
Contraseña ***** Ingresar
Regístrate ahora ¿Olvidaste tu contraseña?

Busqueda Buscar

Inicio Quienes Somos Que es VinculAE Consultas Servicio Contacto Sitios de Interés

Empresa
Duración
Carreras
Recursos

Nombre del Proyecto
Descripción del Proyecto

Guardar

Universidad Tecnológica de la Mixteca, Carretera a Acatlilma, Huajuapán de León, C.P. 69000 Oaxaca, México.
Tel. 9535320214 Lada: 01-800-0000-UTM webmaster@mixteco.utm.mx

Figura 4.26: Pantalla de captura de proyecto para el usuario empresario.

En la Figura 4.27 se muestran en pantalla los datos de un proyecto, los cuales se pueden consultar por todos los perfiles de usuario: los empresarios para modificarlos, los profesores y alumnos para consultarlos y poder aplicar para el proyecto.

The screenshot displays the VinculAE website interface. At the top left is the VinculAE logo, a blue flower-like shape with letters A, V, E, and F. To its right is a login section with fields for 'Usuario' (containing 'usuario123@vinculae.utm.mx') and 'Contraseña' (containing '*****'), with buttons for 'Ingresar', 'Regístrate ahora', and '¿Olvidaste tu contraseña?'. Below the login section is a search bar labeled 'Busqueda' with a 'Buscar' button. A green navigation bar contains links: 'Inicio', 'Quienes Somos', 'Que es VinculAE', 'Consultas', 'Servicio', 'Contacto', and 'Sitios de Interés'. The main content area is light blue and shows project details for 'DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA GESTOR DE VIDEOS'. On the left, a sidebar lists: 'Empresa: NOCOGA PRO', 'Duración: 3 MESES', 'Carreras: COMPUTACIÓN, DISEÑO, EMPRESARIALES', and 'Recursos: 30 MIL PESOS'. The main area shows 'Nombre del Proyecto: DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA GESTOR DE VIDEOS' and 'Descripción del Proyecto: Implementar un sistema con procedimientos efectivos de gestión de videos que nos permita contabilizar, ubicar, registrar, integrar, documentar, verificar y analizar las diferentes videos que se tienen y que se percibirán en un futuro, con el propósito de generar reportes oportunos referentes a los videos, con estadísticas completas, saldos correctos de cada cliente y cada material almacenado, con referencia a su antigüedad y monto que sirvan como base para efectuar trabajos posteriores.' An 'Aplicar' button is at the bottom right of the main area. At the bottom, a blue footer bar contains contact information: 'Universidad Tecnológica de la Mixteca, Carretera a Acatlima, Huajuapán de León, C.P. 69000 Oaxaca, México. Tel: 9535320214 Lada: 01-800-0000-UTM webmaster@mixteco.utm.mx'.

Figura 4.27: Pantalla que muestra los datos de un proyecto.

4.2.7. Pruebas de Usabilidad

Cuando se trabaja en el desarrollo de un sistema, producto o servicio, una de las formas que más se utiliza para evaluar su facilidad de uso son las pruebas de usabilidad. Estas pruebas le presentan al usuario una serie de tareas que tiene que resolver con el prototipo realizado. Dichas pruebas deben realizarse en un lugar similar a donde funcionará el sistema, producto o servicio, o bien en un laboratorio acondicionado para ello. Todas las reacciones que se tomen del usuario deberán considerarse para el proceso siguiente.

El tomar en cuenta de manera temprana los criterios de usabilidad ofrece una clara visión de aspectos de funcionamiento, aceptación y calidad, para los usuarios y desarrolladores. Las pruebas de usabilidad que se realizan durante el desarrollo del producto han ganado aceptación como estrategia para mejorar la calidad del producto o sistema a evaluar.

De acuerdo al UPA Website [69] algunos beneficios de la usabilidad incluyen:

- Mejora en la productividad

- Reducción en costos de capacitación y soporte
- Mejora en ventas y ganancias
- Reducción de tiempos y costos de desarrollo
- Reducción en costos de mantenimiento
- Mejora en la satisfacción de los clientes
- Ventaja competitiva

4.2.7.1. Roles en las pruebas

Al realizar las pruebas de usabilidad de algún producto o sistema, es necesario definir diferentes roles que las personas deben desempeñar con el fin de obtener resultados congruentes. Cuando las pruebas se realizan en un laboratorio especializado se requiere de tres personas con los siguientes roles:

- **Facilitador:** La persona que funge de facilitador es la encargada de guiar al usuario en la prueba. No es necesario que conozca el desarrollo del producto o sistema, pero si debe de tener ya una idea general del mismo para poder ayudar al usuario si fuera necesario. Antes de realizar la prueba, los desarrolladores deben de hablar con el facilitador para explicarle el funcionamiento del sistema de una manera general. De esta forma él se encargará de comentar e indicar las tareas que el usuario debe realizar.
- **Observador:** El papel de observador lo puede desempeñar una o más personas que conozcan el producto o sistema que está bajo prueba dado que deberán tener una idea más completa de las tareas. Dentro de las actividades realizadas por este rol se encuentran las siguientes: Observar las reacciones, gestos, emociones y demás acciones que los usuarios realicen en cada una de las tareas durante la prueba de usabilidad; Realizar anotaciones y formular comentarios acerca de las observaciones en las pruebas; El observador o grupo de observadores deberá colocarse en el área designada de observación del laboratorio, detrás del vidrio espejo.
- **Usuarios:** El grupo de usuarios es el más importante durante las pruebas, dado que éstos son los que van a probar el producto o sistema. Con el fin de no obtener datos erróneos o una cierta tendencia al producto o servicio, deberán ser usuarios reales del producto o sistema en evaluación.

Para realizar dichas pruebas se requiere de un espacio creado para este fin. En la Universidad Tecnológica de la Mixteca se cuenta con un laboratorio de usabilidad (UsaLab) (ver Figura 4.28), espacio en el cual se trabajó con los usuarios, equipo de cómputo, el producto o sistema a evaluar, herramientas para grabación del audio y video, y todo lo necesario para obtener material que ayudó a los análisis posteriores a las pruebas.



Figura 4.28: UsaLab UTM.

En el UsaLab se trabajó con un grupo de 20 usuarios reales, compuesto por estudiantes, profesores y empresarios.

Al llegar el usuario para realizar la prueba se le recibió en el espacio del UsaLab y se le explicó en qué consistía y las actividades a realizar; además se le entregó una hoja con la descripción que puede consultarse en el apéndice C.

Con el fin de evaluar el desenvolvimiento del usuario al utilizar las interfaces del prototipo del sistema se propuso realizar 4 tareas:

1. Abrir y entrar al sistema
2. Realizar una búsqueda de proyectos
3. Consultar un proyecto
4. Dar de alta una propuesta de proyecto

La descripción de las tareas puede consultarse en el apéndice D.

El objetivo de mostrar las diferentes interfaces creadas con los prototipos fue generar una experiencia real del usuario a través de una experiencia agradable. Dichas pruebas se realizaron con usuarios que no estaban implicados en el desarrollo del sistema, ya que, como éstos serán los usuarios finales, su percepción del producto es la que cuenta (ver Figura 4.29). Esto se realizó para no cometer el error de diseñar bajo nuestro concepto únicamente.



Figura 4.29: Usuarios probando las interfaces de VinculAE.

4.2.7.2. Resultados de las pruebas

- El 100 % de los usuarios reportaron haber tenido contacto con interfaces web.

- El 90 % de los usuarios declararon que las interfaces eran prácticas y fáciles de usar.
- El 85 % de los usuarios ejecutaron las tareas sin ningún problema.
- El 95 % de los usuarios considera que la información presentada en las interfaces es la necesaria.
- El 95 % de los usuarios considera que la aplicación de VinculAE ayudará a una mayor relación entre las empresas y la universidad.
- El 100 % de los usuarios mostró interés en el sistema, ya que consideran de gran importancia la vinculación entre la academia y la empresa.
- El 89 % de las opciones mostradas en las interfaces fueron interpretados correctamente por los usuarios.
- El 95 % de los usuarios considera que al ser vía web la aplicación de VinculAE ayudará a una mayor relación entre las empresas y la universidad. Considerando que la mayoría de las empresas se encuentran muy distantes de nuestra ubicación física.

Al finalizar las pruebas y en base a la ayuda de los observadores, y a los comentarios y sugerencias que realizaron los usuarios participantes en las pruebas, se observaron carencias y detectaron deficiencias en el sistema, las cuales se enlistan a continuación:

- La mayoría de los usuarios coincide en que la opción “proyectos” debe tener una sección aparte.
- Los usuarios recomiendan los colores azul, verde, rojo y blanco.
- Algunos usuarios coincidieron en apuntar que en la pantalla principal debe indicarse el significado de VinculAE.
- El 75 % de los usuarios comentaron que deseaban ver la página principal sin tener que teclear un usuario y contraseña.
- La mayoría de los usuarios coincidió en que debe mostrarse la opción proyectos en el menú principal para no confundir al usuario.
- Algunos usuarios consideran que la puesta en marcha de VinculAE impulsará el desarrollo económico y social de la entidad.
- El hecho de tener diferentes opciones en los prototipos causó confusión en los usuarios, pero al identificarlos los pudieron manejar sin problemas.
- Algunos usuarios comentan que el cuadro de diálogo “búsqueda”, debería mostrarse en todas las pantallas.
- Algunos usuarios consideran necesario incluir una sección de oferta laboral al estar relacionados en el sistema el sector empresarial y el académico por parte de los alumnos.
- En el menú de opciones del tercer prototipo sugieren que se tengan submenús de opciones para la posibilidad de visualizar los registros del usuario.
- Los prototipos 2 y 3 fueron los preferidos por los usuarios, pero la gran mayoría considera una versión que haga una mezcla entre los dos prototipos.

A modo de resumen, la Figura 4.30 describe el proceso de la aplicación de DCU Extendida al problema.



Figura 4.30: Resumen de la aplicación de UCDE

4.3. Diseño de la Base de Datos

Un modelo de datos es una colección de herramientas conceptuales para la descripción de datos, relaciones entre ellos, semántica de los datos y restricciones de consistencia. En esta parte se estudiarán dos modelos de datos; el modelo entidad-relación y el modelo relacional.

El modelo entidad-relación (E-R) es un modelo de datos de alto nivel. Está basado en la percepción del mundo real que consiste en una colección de objetos básicos, denominados entidades y de relaciones entre estos objetos.

El modelo relacional es un modelo de menor nivel, usa una colección de tablas para representar tanto los datos como las relaciones entre los datos. Su simplicidad conceptual ha conducido a su adopción general; actualmente, una vasta cantidad de productos de bases de datos se basan en el modelo relacional. Los diseñadores formulan generalmente el diseño del esquema de la base de datos modelando primero los datos en alto nivel, usando el modelo E-R, y después traduciéndolo al modelo relacional. La Figura 4.31 muestra el modelo relacional que corresponde a VinculAE.

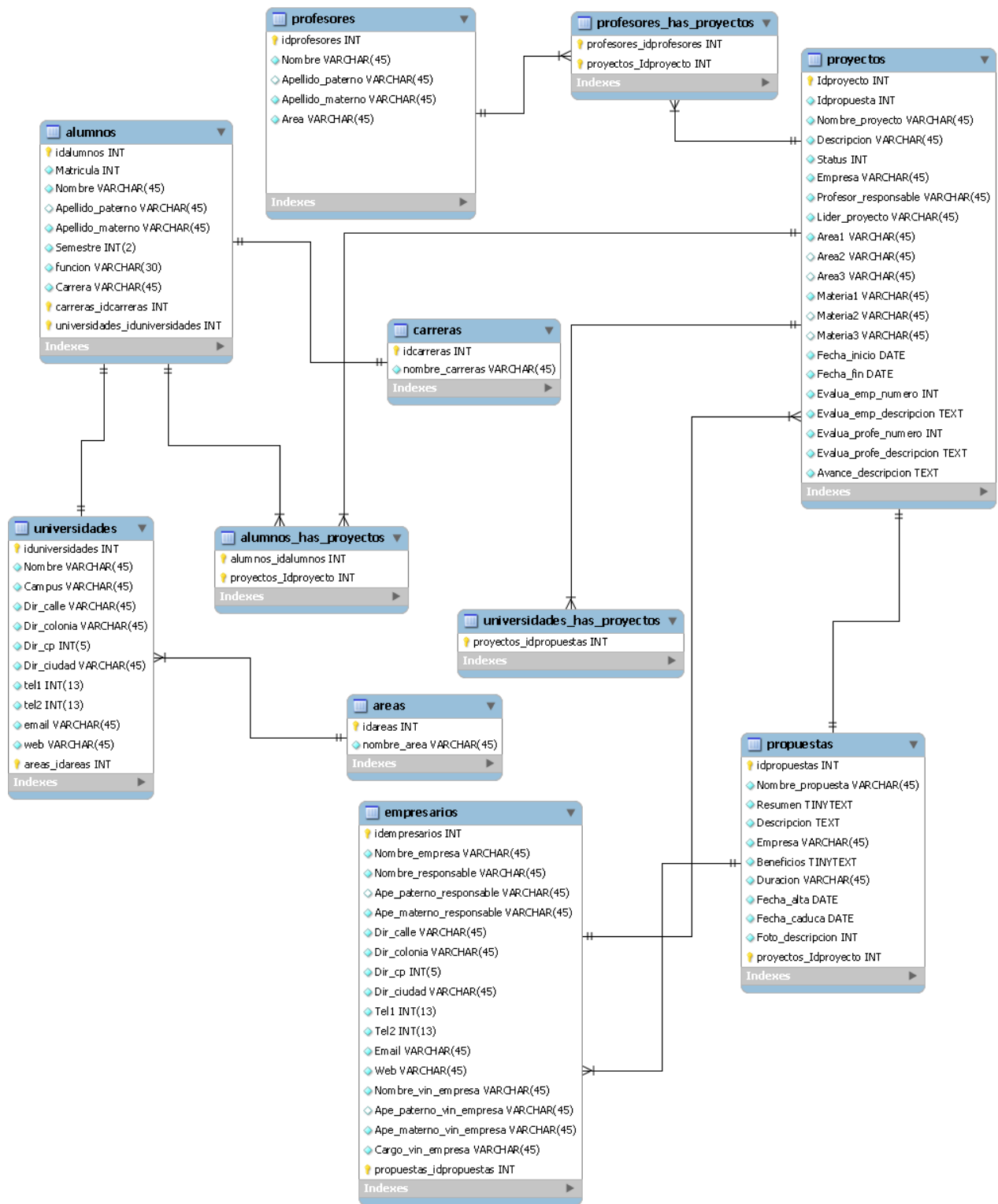


Figura 4.31: Modelo relacional de VinculAE.

4.3.1. Diseño de tablas y atributos

En este apartado se describen las diferentes tablas que componen el sistema una por una, sus relaciones, atributos y toda la información generada en el proceso de modelación de la Base de Datos.

A continuación se detallan las diversas tablas que componen la aplicación. Para cada una de ellas se hará una descripción de cada uno de sus campos, así como las claves propias y ajenas que la componen. En algunos casos, se mostrará una representación de la tabla. También se incluirá el código SQL que se crea en la base de datos.

Para cada una de las tablas del sistema se incluirá una tabla con el formato mostrado en la Tabla 4.3.

| Campo | Tipo | Nulo | Pk | Fk | Descripción |
|-------|------|------|----|----|-------------|
|-------|------|------|----|----|-------------|

Tabla 4.3: Formato de las tablas.

Esta tabla incluirá el nombre del campo, su tipo, si está permitido o no un valor nulo, si es clave primaria (Primary Key), si es clave ajena (Foreign Key) y finalmente una descripción del campo.

También se incluirá un breve dibujo con la tabla en cuestión. En este dibujo se resaltarán los campos que la componen, así como las claves primarias y ajenas.

El formato de esta figura corresponde con la salida de la herramienta gráfica que se ha usado para diseñar la base de datos.

4.3.1.1. Tabla “alumnos”

En esta Tabla se almacenarán los datos de los alumnos que participarán en los proyectos, los cuales se pueden consultar a través del sistema y durante la gestión del proyecto. Se describen los campos que la componen en la Tabla 4.4 y la estructura correspondiente en la Figura 4.32.

| Campo | Tipo | Nulo | Pk | Fk | Descripción |
|------------------|-------------|------|----|----|---|
| Idalumnos | Int | No | Si | No | Identificativo del alumno |
| Matricula | Int | No | No | No | Matrícula del alumno |
| Nombre | Varchar(45) | No | No | No | Nombre del alumno |
| Apellido_paterno | Varchar(45) | Si | No | No | Apellido paterno del alumno |
| Apellido_materno | Varchar(45) | No | No | No | Apellido materno del alumno |
| Semestre | Int | No | No | No | Semestre que cursa el alumno |
| Funcion | Varchar(30) | No | No | No | Función o cargo del alumno en el proyecto |
| Carrera | Varchar(45) | No | No | No | Carrera que estudia el alumno |

Tabla 4.4: Descripción de campos de alumnos.

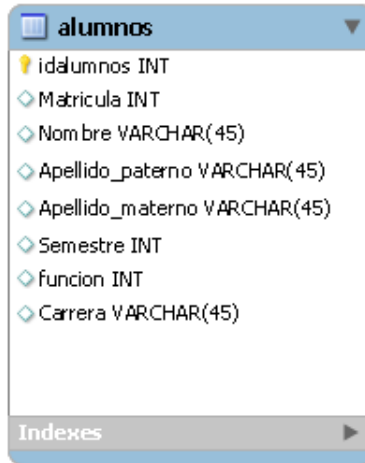


Figura 4.32: Tabla alumnos.

El código relacionado con la Tabla anterior es:

```
CREATE TABLE IF NOT EXISTS 'vinculae'. 'alumnos' (
  'idalumnos' INT NOT NULL AUTO_INCREMENT ,
  'Matricula' INT NOT NULL ,
  'Nombre' VARCHAR(45) NOT NULL ,
  'Apellido_paterno' VARCHAR(45) NULL ,
  'Apellido_materno' VARCHAR(45) NOT NULL ,
  'Semestre' INT(2) NOT NULL ,
  'funcion' VARCHAR(30) NOT NULL ,
  'Carrera' VARCHAR(45) NOT NULL ,
  'carreras_idcarreras' INT NOT NULL ,
  'universidades_iduniversidades' INT NOT NULL ,
  PRIMARY KEY ('idalumnos', 'carreras_idcarreras', 'universidades_iduniversidades') ,
  INDEX 'fk_alumnos_carreras1' ('carreras_idcarreras' ASC) ,
  INDEX 'fk_alumnos_universidades1' ('universidades_iduniversidades' ASC) ,
  CONSTRAINT 'fk_alumnos_carreras1'
  FOREIGN KEY ('carreras_idcarreras' )
  REFERENCES 'vinculae'. 'carreras' ('idcarreras' )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT 'fk_alumnos_universidades1'
  FOREIGN KEY ('universidades_iduniversidades' )
  REFERENCES 'vinculae'. 'universidades' ('iduniversidades' )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
```

4.3.1.2. Tabla “profesores”

En esta Tabla se almacenarán los datos de los profesores que participarán en los proyectos, los cuales se pueden consultar a través del sistema y durante la gestión del proyecto. Se describen los campos que la componen en la Tabla 4.5 y la estructura correspondiente en la Figura 4.33.

| Campo | Tipo | Nulo | Pk | Fk | Descripción |
|------------------|-------------|------|----|----|--|
| Idprofesores | Int | No | Si | No | Identificativo del profesor |
| Nombre | Varchar(45) | No | No | No | Nombre del profesor |
| Apellido_paterno | Varchar(45) | Si | No | No | Apellido paterno del profesor |
| Apellido_materno | Varchar(45) | No | No | No | Apellido materno del profesor |
| Área | Varchar(45) | No | No | No | Área a la que pertenece el profesor |
| Iduniversidad | Varchar(45) | No | No | Si | Universidad a la que pertenece el profesor |

Tabla 4.5: Descripción de campos de profesores.

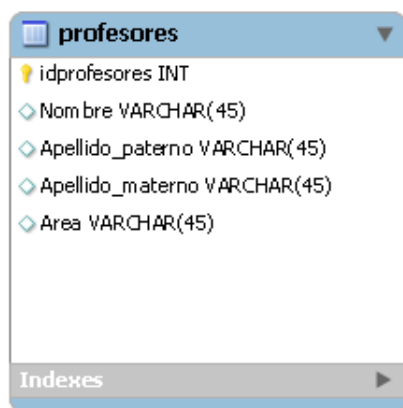


Figura 4.33: Tabla profesores.

El código relacionado con la Tabla anterior es:

```
CREATE TABLE IF NOT EXISTS 'vinculae'.'profesores' (
  'idprofesores' INT NOT NULL AUTO_INCREMENT ,
  'Nombre' VARCHAR(45) NOT NULL ,
  'Apellido_paterno' VARCHAR(45) NULL ,
  'Apellido_materno' VARCHAR(45) NOT NULL ,
  'Area' VARCHAR(45) NOT NULL ,
  PRIMARY KEY ('idprofesores')
ENGINE = InnoDB
COMMENT = 'Tabla que contiene los datos de los profesores'
```

4.3.1.3. Tabla “empresarios”

En esta Tabla se almacenarán los datos de los empresarios que participarán en los proyectos, los cuales se pueden consultar a través del sistema y durante la gestión del proyecto. Se describen los campos que la componen en la Tabla 4.6 y la estructura correspondiente en la Figura 4.34.

| Campo | Tipo | Nulo | Pk | Fk | Descripción |
|-------------------------|-------------|------|----|----|--|
| Idempresarios | Int | No | Si | No | Identificativo del empresario |
| Nombre_empresa | Varchar(45) | No | No | No | Nombre de la Empresa |
| Nombre_responsable | Varchar(45) | No | No | No | Nombre del responsable de la empresa que propone el proyecto |
| Ape_paterno_responsable | Varchar(45) | Si | No | No | Apellido paterno del responsable |
| Ape_materno_responsable | Varchar(45) | No | No | No | Apellido materno del responsable |
| Dir_calle | Varchar(45) | No | No | No | Calle de la empresa |
| Dir_colonia | Varchar(45) | No | No | No | Colonia de la empresa |
| Dir_cp | Int(5) | No | No | No | Codigo postal de la empresa |
| Dir_ciudad | Varchar(45) | No | No | No | Ciudad de la empresa |
| Tel1 | Int(13) | No | No | No | Número telefónico 1 de la empresa |
| Tel2 | Int(13) | No | No | No | Número telefónico 2 de la empresa |
| Email | Varchar(45) | No | No | No | Email de la empresa |
| Web | Varchar(45) | No | No | No | Sitio web de la empresa |
| Nombre_vin_empresa | Varchar(45) | No | No | No | Nombre del responsable de la vinculación por parte de la empresa |
| Ape_paterno_vin_empresa | Varchar(45) | Si | No | No | Apellido paterno del responsable |
| Ape_materno_vin_empresa | Varchar(45) | No | No | No | Apellido materno del responsable |
| Cargo_vin_empresa | Varchar(45) | No | No | No | Cargo del responsable de vinculación |
| Email_vin_empresa | Varchar(45) | No | No | No | Email del responsable de vinculación |

Tabla 4.6: Descripción de campos de empresarios.

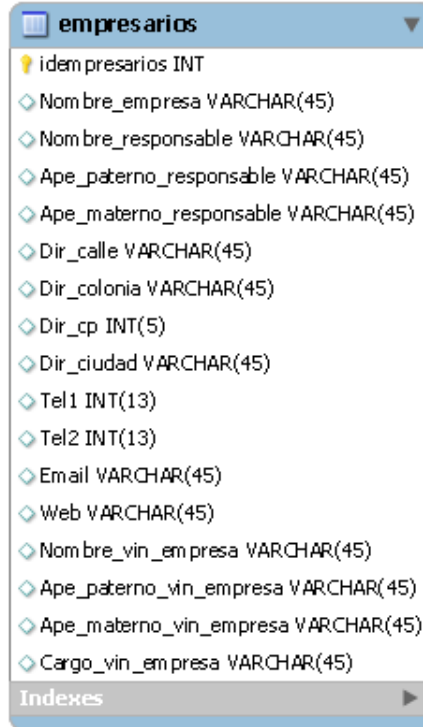


Figura 4.34: Tabla empresarios.

4.3.1.4. Tabla “áreas”

En esta Tabla se almacenarán las áreas de las universidades participantes en los proyectos, las cuales se pueden consultar a través del sistema y durante la gestión del proyecto. Se describen los campos que la componen en la Tabla 4.7 y la estructura correspondiente en la Figura 4.35.

| Campo | Tipo | Nulo | Pk | Fk | Descripción |
|-------------|-------------|------|----|----|--------------------------------|
| Idareas | Int | No | Si | No | Identificativo de áreas |
| Nombre_area | Varchar(45) | No | No | No | Nombre del área o departamento |

Tabla 4.7: Descripción de campos de áreas.

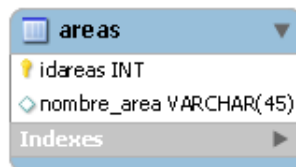


Figura 4.35: Tabla áreas.

```

CREATE TABLE IF NOT EXISTS 'vinculae'. 'areas' (
'idareas' INT NOT NULL AUTO_INCREMENT ,
'nombre_area' VARCHAR(45) NOT NULL ,
PRIMARY KEY ('idareas' )
ENGINE = InnoDB

```

4.3.1.5. Tabla “propuestas”

En esta Tabla se almacenarán los datos de las propuestas realizadas por los empresarios a través del sistema, las cuales se pueden consultar a través del mismo. Los datos almacenados se muestran a profesores y alumnos, y que al ser seleccionados para su aplicación pasaran a ser un proyecto activo. Se describen los campos que la componen en la Tabla 4.8 y la estructura correspondiente en la Figura 4.36.

| Campo | Tipo | Nulo | Pk | Fk | Descripción |
|------------------|-------------|------|----|----|--|
| Idpropuestas | Int | No | Si | No | Identificativo de propuesta |
| Nombre_propuesta | Varchar(45) | No | No | No | Nombre de la propuesta |
| Resumen | Tinytext | No | No | No | Resumen de la propuesta |
| Descripción | Text | No | No | No | Descripción general que contiene los datos de la propuesta |
| Empresa | Varchar(45) | No | No | No | Nombre de la empresa que propone |
| Beneficios | Tinytext | No | No | No | Beneficios que ofrece la empresa |
| Duracion | Varchar(45) | No | No | No | Tiempo de duración estimada de la propuesta de proyecto |
| Fecha_alta | Date | No | No | No | Fecha en que fue dada de alta la propuesta |
| Fecha_caduca | Date | No | No | No | Fecha hasta la cual podrá estar vigente la propuesta en el sistema |
| Foto_descripcion | Int | No | No | No | Imagen o fotografía que describe la propuesta |

Tabla 4.8: Descripción de campos de propuestas.

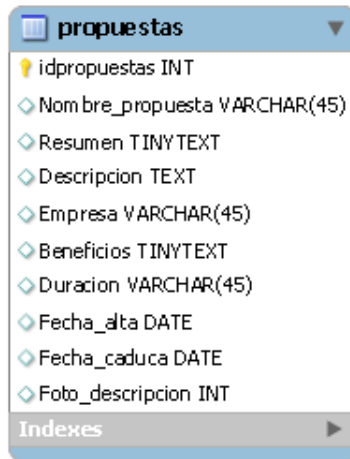


Figura 4.36: Tabla propuestas.

```

CREATE TABLE IF NOT EXISTS 'vinculae'.'propuestas' (
'idpropuestas' INT NOT NULL AUTO_INCREMENT ,
'Nombre_propuesta' VARCHAR(45) NOT NULL ,
'Resumen' TINYTEXT NOT NULL ,
'Descripcion' TEXT NOT NULL ,
'Empresa' VARCHAR(45) NOT NULL ,
'Beneficios' TINYTEXT NOT NULL ,
'Duracion' VARCHAR(45) NOT NULL ,
'Fecha_alta' DATE NOT NULL ,
'Fecha_caduca' DATE NOT NULL ,
'Foto_descripcion' INT NOT NULL ,
'proyectos_Idproyecto' INT NOT NULL ,
PRIMARY KEY ('idpropuestas', 'proyectos_Idproyecto') ,
INDEX 'fk_propuestas_proyectos1' ('proyectos_Idproyecto' ASC) ,
CONSTRAINT 'fk_propuestas_proyectos1'
FOREIGN KEY ('proyectos_Idproyecto' )
REFERENCES 'vinculae'.'proyectos' ('Idproyecto' )
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB

```

4.3.1.6. Tabla “proyectos”

En esta Tabla se almacenarán los datos de los proyectos que se encuentran en proceso de desarrollo, los cuales pueden ser consultados a través del sistema y durante la gestión del proyecto. Se describen los campos que la componen en la Tabla 4.9 y la estructura correspondiente en la Figura 4.37.

| Campo | Tipo | Nulo | Pk | Fk | Descripción |
|----------------------|-------------|------|----|----|---|
| Idproyecto | Int | No | Si | No | Identificativo del proyecto |
| Idpropuesta | Int | No | No | Si | Identificativo de propuesta |
| Nombre_proyecto | Varchar(45) | No | No | No | Nombre del proyecto |
| Descripción | Varchar(45) | No | No | No | Descripción del proyecto |
| Status | Int | No | No | No | Estado de avance del proyecto especificado en porcentaje |
| Empresa | Varchar(45) | No | No | No | Empresa responsable del proyecto |
| Profesor_responsable | Varchar(45) | No | No | No | Profesor responsable del proyecto |
| Lider_proyecto | Varchar(45) | No | No | No | Alumno Lider del proyecto |
| Area1 | Varchar(45) | No | No | No | Area 1 participante del proyecto |
| Area2 | Varchar(45) | Si | No | No | Area 2 participante del proyecto |
| Area3 | Varchar(45) | Si | No | No | Area 3 participante del proyecto |
| Materia1 | Varchar(45) | No | No | No | Materia 1 de evaluación por el proyecto |
| Materia2 | Varchar(45) | Si | No | No | Materia 2 de evaluación por el proyecto |
| Materia3 | Varchar(45) | Si | No | No | Materia 3 de evaluación por el proyecto |
| Fecha_inicio | Date | No | No | No | Fecha de inicio del proyecto |
| Fecha_fin | Date | No | No | No | Fecha de fin del proyecto |
| Evalua_emp_numero | Int | No | No | No | Evaluación numérica otorgada al proyecto por el empresario |
| Evalua_emp_descri | Text | No | No | No | Justificación de la calificación otorgada por el empresario |
| Evalua_profe_numero | Int | No | No | No | Evaluación numérica otorgada por el profesor al proyecto |
| Evalua_profe_descri | Text | No | No | No | Descripción de la calificación otorgada por el profesor |
| Avance_descripcion | Text | Si | No | No | Reporte de avances del proyecto |

Tabla 4.9: Descripción de campos de proyectos.

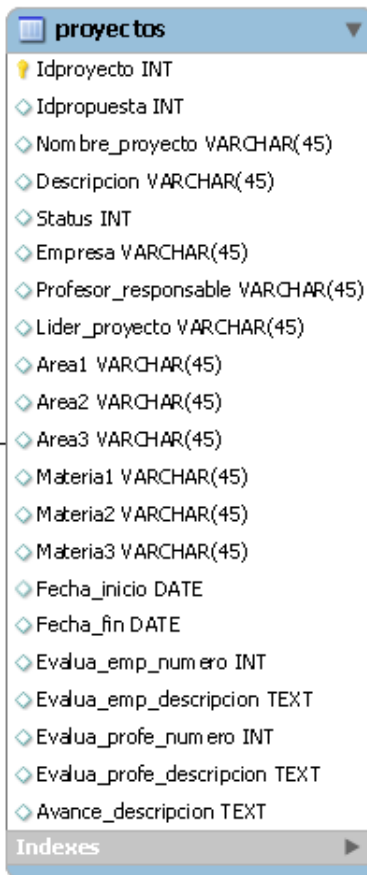


Figura 4.37: Tabla proyectos.

```
CREATE TABLE IF NOT EXISTS 'vinculae'. 'proyectos' (
  'Idproyecto' INT NOT NULL AUTO_INCREMENT ,
  'Idpropuesta' INT NOT NULL ,
  'Nombre_proyecto' VARCHAR(45) NOT NULL ,
  'Descripcion' VARCHAR(45) NOT NULL ,
  'Status' INT NOT NULL ,
  'Empresa' VARCHAR(45) NOT NULL ,
  'Profesor_responsable' VARCHAR(45) NOT NULL ,
  'Lider_proyecto' VARCHAR(45) NOT NULL ,
  'Area1' VARCHAR(45) NOT NULL ,
  'Area2' VARCHAR(45) NULL ,
  'Area3' VARCHAR(45) NULL ,
  'Material' VARCHAR(45) NOT NULL ,
  'Materia2' VARCHAR(45) NULL ,
  'Materia3' VARCHAR(45) NULL ,
  'Fecha_inicio' DATE NOT NULL ,
  'Fecha_fin' DATE NOT NULL ,
```

```

'Evalua_emp_numero' INT NOT NULL ,
'Evalua_emp_descripcion' TEXT NOT NULL ,
'Evalua_profe_numero' INT NOT NULL ,
'Evalua_profe_descripcion' TEXT NOT NULL ,
'Avance_descripcion' TEXT NOT NULL ,
PRIMARY KEY ('Idproyecto') ,
INDEX 'Idempresa' () ,
CONSTRAINT 'Idempresa'
FOREIGN KEY ()
REFERENCES 'vinculae'. 'empresarios' ()
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB

```

4.3.1.7. Tabla “universidades”

En esta Tabla se almacenarán los datos de las universidades participantes dentro del SUNEО, las cuales pueden ser consultadas a través del sistema. Se describen los campos que la componen en la Tabla 4.10 y la estructura correspondiente en la Figura 4.38.

| Campo | Tipo | Nulo | Pk | Fk | Descripción |
|---------------|-------------|------|----|----|---------------------------------|
| Iduniversidad | Int | No | Si | No | Identificativo de universidades |
| Nombre | Varchar(45) | No | No | No | Nombre de la universidad |
| Campus | Varchar(45) | No | No | No | Campus de la universidad |
| Dir_calle | Varchar(45) | No | No | No | Calle de la universidad |
| Dir_colonia | Varchar(45) | No | No | No | Colonia de la universidad |
| Dir_cp | Int(5) | No | No | No | Código postal de la universidad |
| Dir_ciudad | Varchar(45) | No | No | No | Ciudad o población |
| Tel1 | Int(13) | No | No | No | Teléfono 1 |
| Tel2 | Int(13) | No | No | No | Teléfono 2 |
| Email | Varchar(45) | No | No | No | Email de la universidad |
| Web | Varchar(45) | No | No | No | Sitio web de la universidad |

Tabla 4.10: Descripción de campos de universidades.

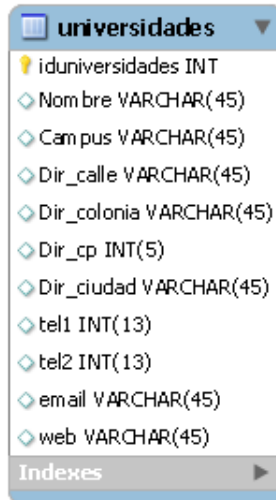


Figura 4.38: Tabla universidades.

```

CREATE TABLE IF NOT EXISTS 'vinculae'.'universidades' (
  'iduniversidades' INT NOT NULL AUTO_INCREMENT ,
  'Nombre' VARCHAR(45) NOT NULL ,
  'Campus' VARCHAR(45) NOT NULL ,
  'Dir_calle' VARCHAR(45) NOT NULL ,
  'Dir_colonia' VARCHAR(45) NOT NULL ,
  'Dir_cp' INT(5) NOT NULL ,
  'Dir_ciudad' VARCHAR(45) NOT NULL ,
  'tel1' INT(13) NOT NULL ,
  'tel2' INT(13) NOT NULL ,
  'email' VARCHAR(45) NOT NULL ,
  'web' VARCHAR(45) NOT NULL ,
  'areas_idareas' INT NOT NULL ,
  PRIMARY KEY ('iduniversidades', 'areas_idareas'),
  INDEX 'fk_universidades_areas1' ('areas_idareas' ASC) ,
  CONSTRAINT 'fk_universidades_areas1'
  FOREIGN KEY ('areas_idareas')
  REFERENCES 'vinculae'.'areas' ('idareas')
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB

```

Capítulo 5

Resultados.

Una vez que se ha mostrado el diseño de un modelo dinámico para la enseñanza de la Ingeniería de Software, el cual se basa en (1) la identificación y el establecimiento de 5 módulos de aprendizaje compuestos por materias del plan de estudios actual para la Ingeniería en Computación de la UTM, y (2) el diseño de la herramienta VinculAE para habilitar la participación de los estudiantes en proyectos reales de empresas registradas en la red de vinculación; a continuación se muestra su aplicación en el desarrollo de un sistema real basado en las necesidades de una empresa Oaxaqueña.

En este sentido, actualmente la red de VinculAE cuenta con el registro de dos empresas del Estado de Oaxaca a quienes se les ha presentado la idea y están dispuestas a participar activamente en la formación de los estudiantes a través de asesorías y consejos durante el desarrollo de los proyectos: TICDES S.A de C.V y RagaSoft S.A de C.V. Ambas empresas son del tipo PyME desarrolladoras de software, por lo que los estudiantes recibirán de primera mano la guía y tutoría de profesionistas que fortalecerán los conceptos teóricos aprendidos en las aulas.

Para iniciar la fase de evaluación, se le pidió a ambas empresas que subieran propuestas de proyectos al sistema VinculAE para que pudieran ser analizadas junto con los estudiantes (equipo de desarrollo) y se platicara sobre las posibles soluciones. Cabe recordar que de ninguna manera se intenta cambiar la forma de desarrollar software que los estudiantes hayan aprendido en cursos anteriores, simplemente, y durante el desarrollo de la solución, se enseña a aplicar los conceptos teóricos aprendidos y se brinda asesoría sobre métodos, técnicas o estándares que las empresas requieran. Por su parte, los representantes de las empresas dirigen a los estudiantes en el trabajo por equipo, la planificación de los proyectos y los planes de calidad. En el caso de las dos empresas, ambas trabajan de forma ágil siguiendo SCRUM, por lo que fue necesario que el director de la tesis ofreciera previamente un curso de formación que enseñara a los estudiantes a requisitar las plantillas requeridas por la empresa. Estos conocimientos se fortalecieron con la práctica constante en el desarrollo de un proyecto.


De esta forma, la herramienta de vinculación permite que todos los artefactos (documentos de diseño, planes, calendarios, modelos, etc.) generados durante el desarrollo de software se compartan y expliquen a los estudiantes durante la práctica, para proporcionales así conocimiento práctico a través de especialistas en la materia. La metodología que los estudiantes utilicen para desarrollar los proyectos, dependerá de la exigencia de las empresas, el profesor (especialista de

la academia) deberá guiar y controlar el trabajo de los equipos.

Por lo tanto, los resultados que se presentan a continuación ejemplifican y muestran el uso de VinculAE a través del desarrollo de una herramienta creada para la empresa RagaSoft S.A de C.V., cumpliendo así el objetivo principal del presente trabajo.

5.1. Preparación del caso de estudio

El caso de estudio inició con la presentación del proyecto VinculAE a la empresa RagaSoft S.A. de C.V., resumiendo los objetivos de la vinculación academia-industria y los posibles beneficios que se alcanzarían si se logra consolidar como una herramienta de soporte en el desarrollo de software. RagaSoft S.A. de C.V. es una pequeña empresa desarrolladora de software ubicada en el Estado de Oaxaca desde comienzos del 2006. La empresa cuenta con una plantilla de 15 desarrolladores de software, 2 secretarias, 1 contador y 1 director general. Después de platicar y definir el alcance de la colaboración para la prueba de VinculAE, RagaSoft decidió definir 4 proyectos para iniciar la experimentación sobre la vinculación. La Figura 5.1 muestra que RagaSoft registró cuatro iniciativas de proyectos que quedaron a disposición de los estudiantes para iniciar su participación en proyectos reales.



Propuesta No.

| | |
|------------------------------|--|
| Nombre de la Empresa: | RagaSoft S.A. de C.V |
| Nombre de la propuesta: | Diseñar y construir una herramienta de software para evaluar proyectos. |
| Resumen de la propuesta: | Diseño y construcción de una herramienta de software que permita evaluar la viabilidad de los proyectos informáticos y la gestión de una cartera de proyectos. |
| Descripción de la propuesta: | Debido al crecimiento de la empresa se propone la realización de un proyecto que consiste en el desarrollo de un sistema de información o herramienta de software que nos permita realizar la evaluación de proyectos informáticos que nos presenten y que de igual manera nos permita gestionar una cartera de proyectos. |

| Estudiantes | Docentes | Universidades | Empresas |
|-----------------|------------|---------------|------------|
| Proyectos | Biblioteca | Oferta SUNE0 | Proyectos |
| Becas | Docentes | Noticias | Noticias |
| Practicas | Evaluación | Universidades | Clientes |
| Servicio social | | Vinculación | Evaluación |

Figura 5.1: Propuestas registradas en VinculAE.

Para el desarrollo de la solución se formó un equipo con 3 estudiantes de servicio social que cursaban el décimo semestre de la carrera de Ingeniería en Computación y así como el tesista, todos alumnos de la Universidad Tecnológica de la Mixteca. Dado que era necesario probar la efectividad de VinculAE con proyectos que no afectaran el desarrollo de un curso verdadero, se decidió probar con estos estudiantes para que una vez que las ventajas y desventajas fueran plenamente identificadas sea posible integrar la herramienta como soporte a la enseñanza de los estudiantes. El director de esta tesis fungió como tutor de la materia “Ingeniería de Software I”, mientras que el tesista desempeñó el papel de líder del proyecto. De forma adicional, el responsable de la empresa fungió como consultor especialista y asumió la responsabilidad de guiar y dirigir a los alumnos en el desarrollo del producto. La Figura 5.2 muestra la creación del equipó y la selección del proyecto a realizar. Una vez que el proyecto fue seleccionado, se realizó el lanzamiento de su desarrollo de forma colaborativa. El papel del profesor es guiar a los alumnos desde la selección del ciclo de vida hasta la entrega y liberación del sistema. El consultor especializado marca los tiempos de desarrollo, asesora, solicita información y/o documentación, participa activamente en el diseño del sistema, prueba los prototipos, acepta el sistema y evalúa el desempeño de los alumnos. Las secciones posteriores explicarán el desarrollo de la solución a una de las propuestas sometidas por la empresa.

The screenshot displays the VinculAE web application interface. At the top, there are navigation tabs: Inicio, VinculAE, Proyectos, Trabajo, and Contacto. The main content area is a form for creating a project. It includes fields for 'Proyecto No.' (2) and 'Propuesta No.' (3). The 'Nombre de la Empresa' is 'RagaSoft S.A. de C.V.', and the 'Nombre de la propuesta' is 'Diseñar y construir una herramienta de software para evaluar proyectos.' The 'Resumen de la propuesta' is 'Diseño y construcción de una herramienta de software que permita evaluar la viabilidad de los proyectos informáticos y la gestión de una cartera de proyectos.' Below this, there are dropdown menus for 'Area 1' (Computación), 'Materia 1' (Ingeniería de Software I), 'Area 2', 'Materia 2', 'Area 3', and 'Materia 3'. The 'Profesor responsable' is 'Ivan Antonio Garcia Pacheco' and the 'Lider de proyecto' is 'Noe Coronel Garcia'. At the bottom, there are date pickers for 'Fecha inicio' (12 Julio 11) and 'Fecha fin' (07 Dic 11). A footer menu contains links for Estudiantes, Docentes, Universidades, and Empresas, each with a list of sub-links.

| Estudiantes | Docentes | Universidades | Empresas |
|-----------------|------------|---------------|------------|
| Proyectos | Biblioteca | Oferta SUNEO | Proyectos |
| Becas | Docentes | Noticias | Noticias |
| Practicas | Evaluación | Universidades | Clientes |
| Servicio social | | Vinculación | Evaluación |

Figura 5.2: Creación del equipo y asignación de proyecto.

5.2. Diseño y desarrollo del sistema para la gestión de carteras de proyectos

Una de las propuestas seleccionadas para el desarrollo vinculado consistió en el diseño y construcción de una herramienta software que permitiera la evaluación de la viabilidad de los proyectos informáticos y la gestión de una cartera de proyectos. De acuerdo a la Universidad de Melbourne, una cartera de proyectos es un sistema de información que contiene los atributos clave de las aplicaciones implementadas en una empresa. Se utiliza como herramienta para gestionar el valor de negocio de una aplicación a través de su ciclo de vida.

De forma práctica, el consultor describió a los alumnos dos funcionalidades principales para la herramienta:

- La primera consistía en facilitar la evaluación de los proyectos antes de pasar a la cartera,
- La segunda se refería a la gestión de la cartera de proyectos, donde se podrían visualizar las principales características de cada proyecto y se podrían comparar los proyectos de la cartera en base a distintos parámetros.

Una vez que el consultor definió la funcionalidad general de la herramienta, el tutor dirigió a los estudiantes en el desarrollo de una solución utilizando el modelo incremental para desarrollo de software, el cual es un paradigma evolutivo e iterativo de la Ingeniería de Software que permite llevar a cabo el diseño y el desarrollo del proceso del software de forma más dinámica.

Así, se presentó una arquitectura de solución al consultor, basada en el modelo incremental, que distinguía cuatro capas en su diseño y desarrollo: la capa lógica, la capa de análisis técnico-operativo, la capa de evaluación económica, y la capa de proyecto definitivo y gestión de cartera (ver Figura 5.3).

Después de establecer ajustes y correcciones a cada capa, ambas partes llegaron al acuerdo de definir el contenido de cada capa como se resume a continuación:

- Capa lógica. Permite la gestión de usuarios (crear, eliminar, editar, buscar, listar) y la gestión de proyectos (crear, abrir, eliminar), además incluye la creación y gestión de la base de datos del sistema.
- Análisis técnico operativo. Una vez que se ha creado o abierto un proyecto permite llevar a cabo las primeras dos fases de la evaluación de la factibilidad de los proyectos informáticos.

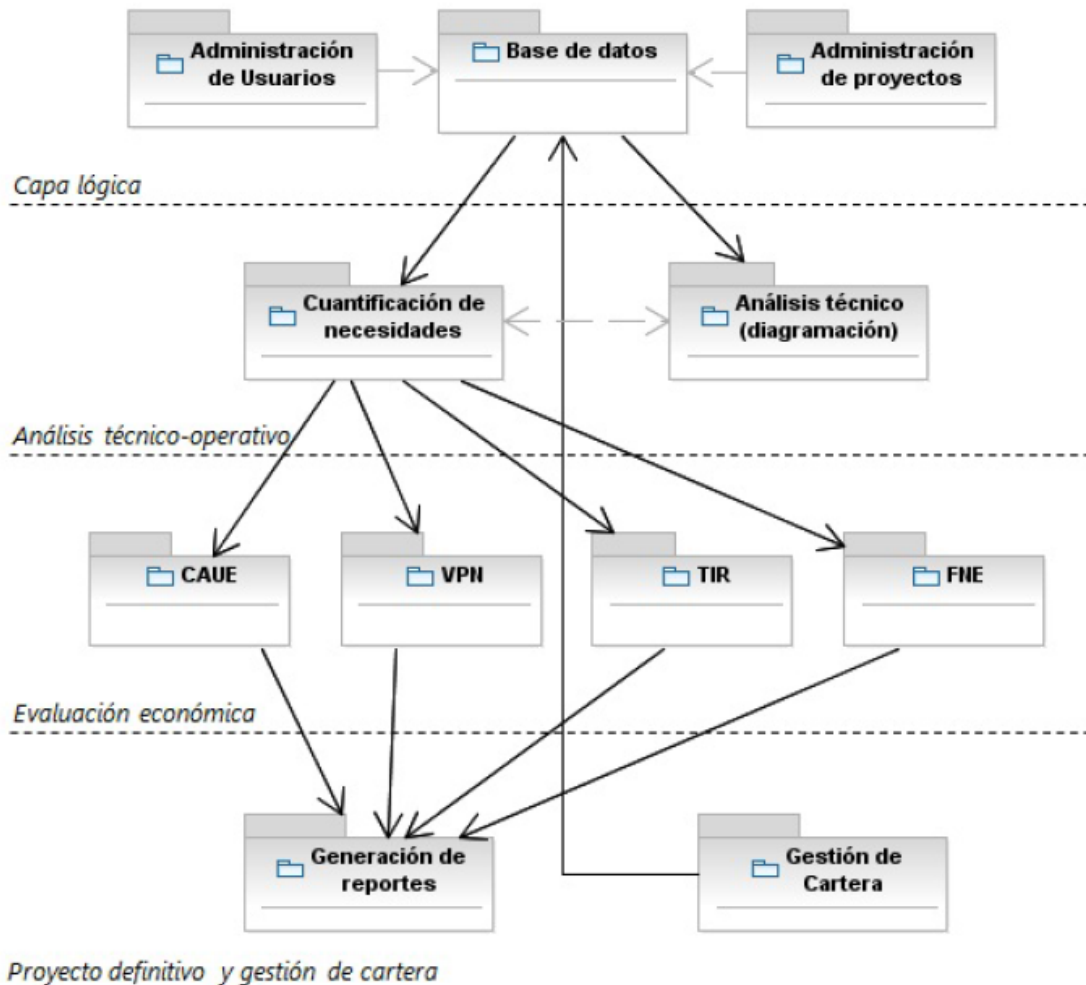


Figura 5.3: Arquitectura del sistema para la gestión de carteras de proyectos informáticos.

- Evaluación económica. Ingresar los datos necesarios y realizar el tipo de evaluación económica que convenga al proyecto, CAUE, VPN, TIR o FNE.
- Proyecto definitivo y gestión de cartera. Permite la generación de reportes a partir del resultado de la evaluación de factibilidad. Incluye las funcionalidades referentes a la gestión de cartera (ver proyectos de la cartera, comparar, aplicar filtros de visualización, ver detalle de un proyecto, eliminar proyecto).

La Figura 5.4 muestra que la arquitectura de la herramienta consideró dos actores principales que serían quienes interactuarían con las funcionalidades del sistema: el administrador y el analista.

El analista de proyectos sería el usuario principal de la herramienta, su función consistiría en administrar los proyectos del sistema y llevar a cabo paso a paso la evaluación de la viabilidad de los mismos.

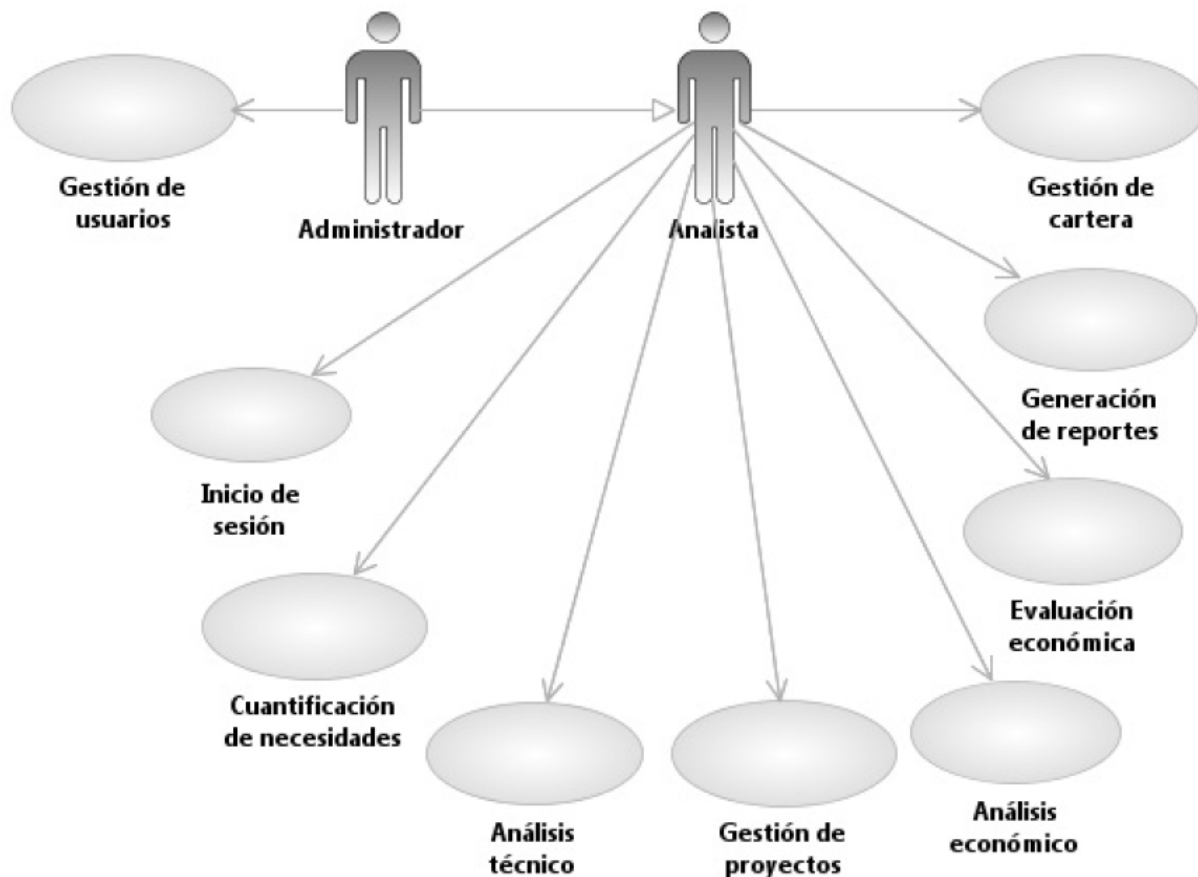


Figura 5.4: Diagrama general de casos de uso de la herramienta propuesta.

Por otro lado, el usuario administrador podría realizar las funciones del analista agregando a sus responsabilidades la gestión de los usuarios del sistema. Cada funcionalidad se agrupó de acuerdo a las capas previamente descritas de la siguiente forma:

- La capa lógica estuvo conformada por la gestión de usuarios, el inicio de sesión y la gestión de proyectos
- A la capa técnico-operativa le pertenecieron la cuantificación de necesidades y el análisis técnico
- El análisis económico y la evaluación económica conformaron la capa económica o de evaluación económica
- Y la capa de proyecto definitivo y gestión de cartera la integraron la generación de reportes y la gestión de cartera de proyectos.

Para hacer posible cada una de estas funcionalidades se requirió de una base de datos que almacena la información tanto de los proyectos ingresados al sistema como de los usuarios que acceden al mismo. En total, la base de datos de la herramienta contó con 14 tablas que buscan optimizar la organización de la información.

A través del trabajo vinculado fue posible acercar a los estudiantes con las exigencias y

demandas de una empresa real con necesidades verdaderas. El consultor ofreció guía y asesoría de primera mano en la forma en que se entregaban la especificación de requerimientos o el diseño de bajo nivel, por ejemplo. La Figura 5.5 muestra que desde el inicio del proyecto fue posible liberar los entregables para su revisión y aceptación dentro de los tiempos establecidos por el consultor. Cuando se presentaron retrasos, ambas partes (consultor y tutor) hicieron las acciones correctivas necesarias para que los estudiantes aprendieran sobre las consecuencias de los retrasos.

Universidad Tecnológica de la Mixteca

VinculAE

Inicio VinculAE Proyectos Trabajo Contacto

Proyecto No. Avance del proyecto: Fecha:

Nombre de la Empresa:

Nombre del proyecto:

Reporte de avance:

Seleccionar archivo:

Profesor responsable:

Lider de proyecto:

[Guardar »](#)

| Estudiantes | Docentes | Universidades | Empresas |
|-----------------|------------|---------------|------------|
| Proyectos | Biblioteca | Oferta SUNE0 | Proyectos |
| Becas | Docentes | Noticias | Noticias |
| Practicas | Evaluación | Universidades | Clientes |
| Servicio social | | Vinculación | Evaluación |

Figura 5.5: Compartición de documentos entregables para revisión.

Finalmente, la herramienta construida refleja el conocimiento transmitido por la empresa a los estudiantes en el sentido de que es posible diferenciar entre los tipos de software que ésta desarrolla. Los estudiantes aprendieron, con este proyecto en particular, que no todos los proyectos de software son idénticos y que es necesario tratar a cada uno en base a características muy concretas (ver Figura 5.6). En este sentido, la Figura 5.7 muestra que también la herramienta entregada cumple con las especificaciones sometidas por la empresa en cuanto a la cuantificación de necesidades.

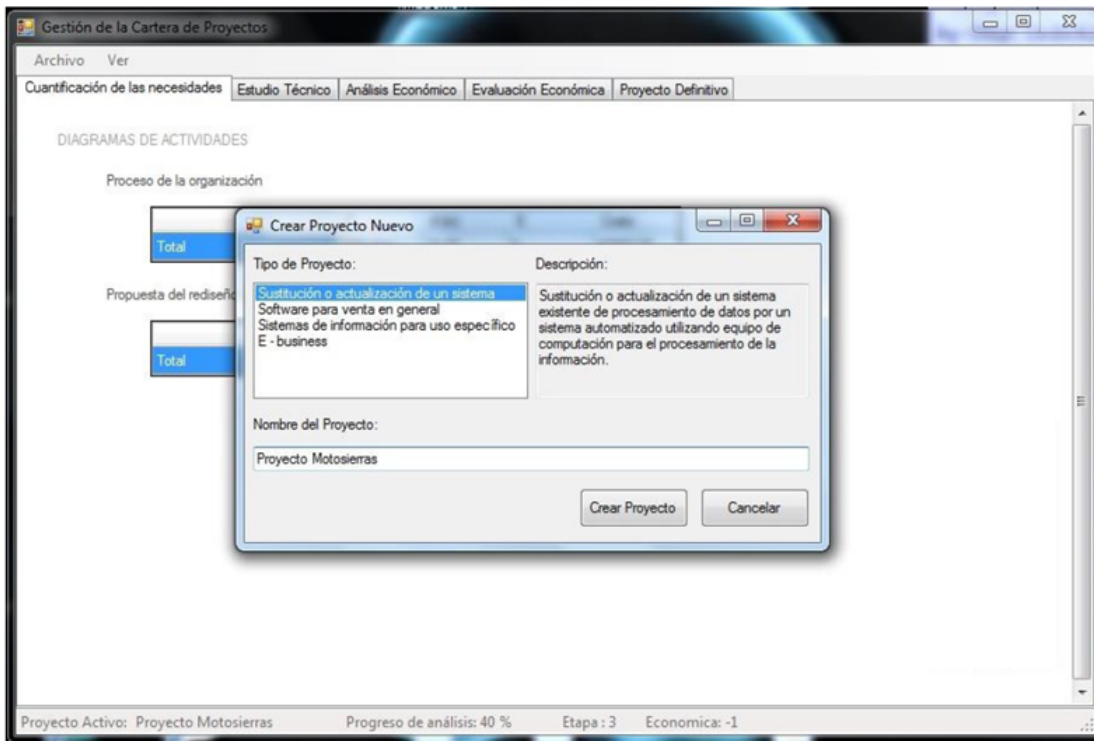


Figura 5.6: Alta de un proyecto en herramienta creada.

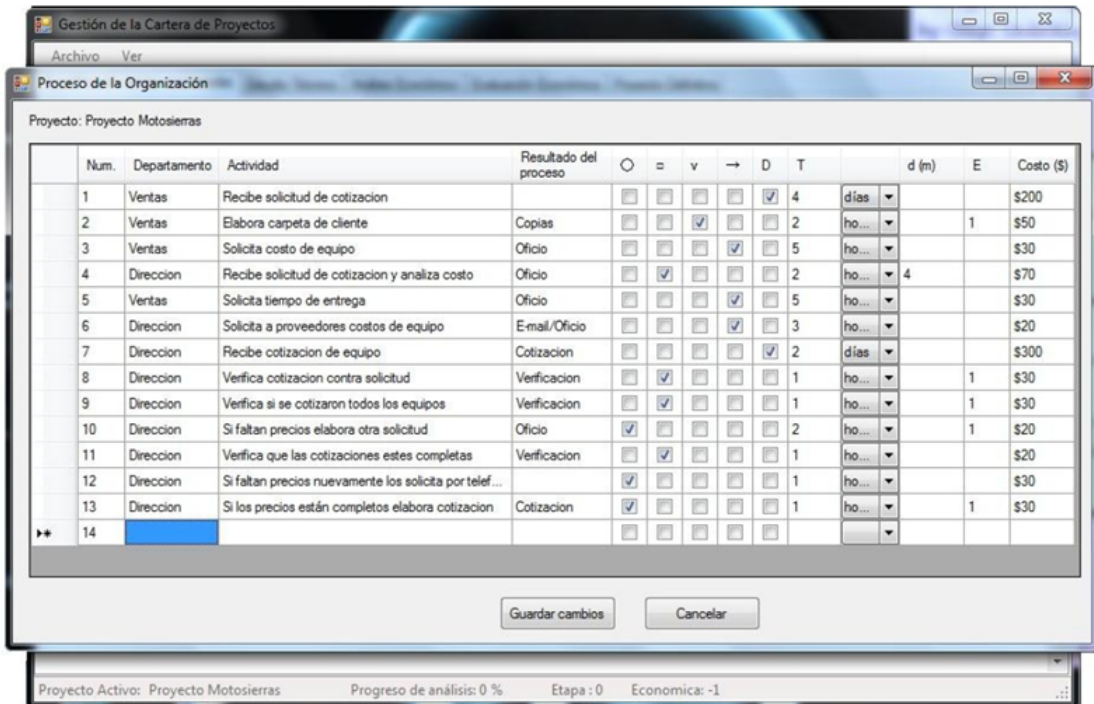


Figura 5.7: Cuantificación de necesidades en herramienta creada.

Como se había mencionado anteriormente, y dadas las necesidades del proyecto relacionadas con la evaluación económica de proyectos informáticos, fue necesario recurrir a otras materias y otros especialistas en matemáticas financieras de la Universidad para automatizar el proceso de evaluación utilizando los cuatro métodos solicitados por la empresa (ver Figura 5.8).

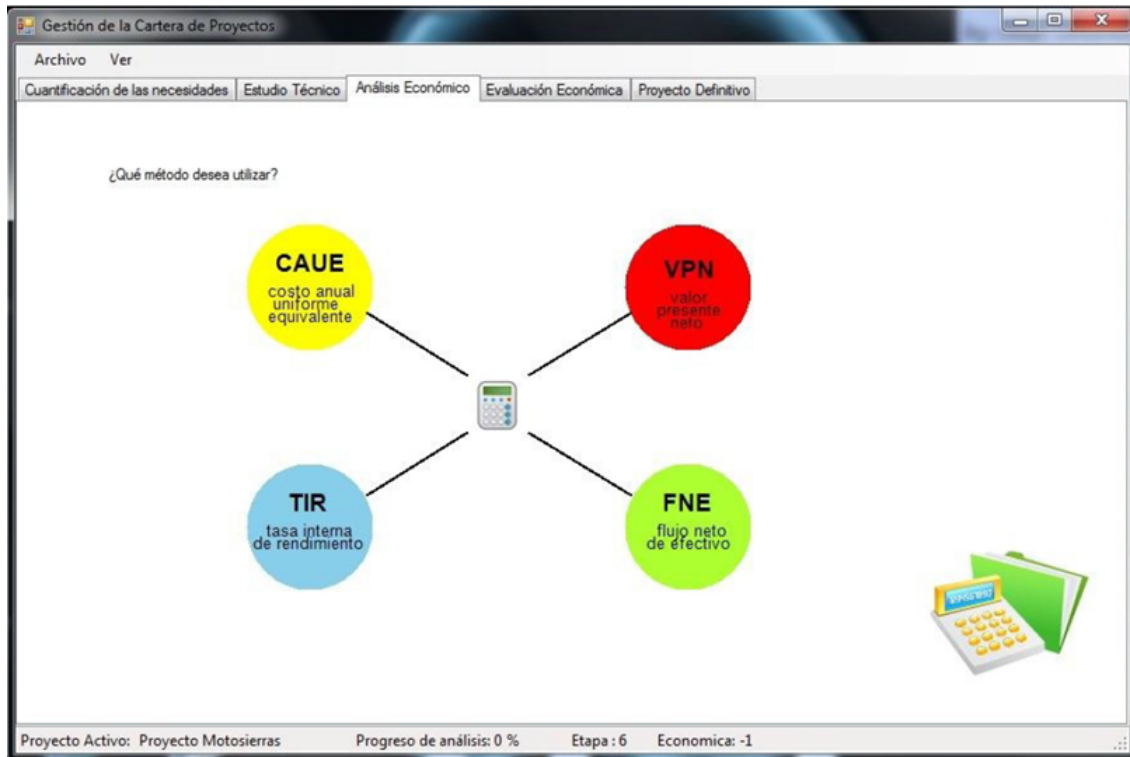


Figura 5.8: Selección del método para la evaluación económica en herramienta creada.

Por último, la Figura 5.9 muestra que la gestión de la cartera se basó en los parámetros especificados por la empresa y al igual que las herramientas comerciales se incluyó una alarma tipo semáforo que le indica a la empresa cuando es conveniente realizar un proyecto (color verde), cuando es conveniente realizar asumiendo riesgos y estableciendo controles (color amarillo), y cuando no es conveniente realizarlo (color rojo).



Figura 5.9: Gestión de la cartera de proyectos en herramienta creada.

5.3. Hallazgos alcanzados

Después de desarrollar tres proyectos en vinculación con la empresa RagaSoft con diferentes equipos, se identificaron algunos problemas que deben ser abordados cuidadosamente cuando se pretenda trabajar bajo este mismo enfoque. Éstos se resumen a continuación.

5.3.1. Problemas culturales

Los equipos formados para la implementación de cada proyecto asumían un alto grado de aceptación de sus soluciones en áreas donde se proporcionaba una mejor funcionalidad que el antiguo sistema basado en hojas y cálculos manuales. Sin embargo, algunas funciones y procesos no obtuvieron el pleno reconocimiento de la empresa por las características de los sistemas heredados con los que se contaban. Los equipos decidieron resolver este problema incluyendo mejoras que permitiera sustituir los sistemas viejos, rompiendo así la tradicional segregación de departamentos. El plan original trazado por la empresa se modificó en un intento por abordar los cambios culturales y de capacitación que acarrearía la nueva herramienta. Esta capacitación tomó la forma de seminarios organizados, que se dividieron en dos grupos de usuarios: analistas y directivo. La capacitación restante para los usuarios finales se llevó a cabo internamente en colaboración con el consultor.

En relación a la forma de trabajo, los estudiantes no tuvieron problemas al trabajar bajo fechas

previamente establecidas y regularmente eran felicitados por el consultor por terminar a tiempo el desarrollo de una tarea. Creemos que este resultado tiene relación directa con el ritmo de trabajo que se establece en la Universidad desde el inicio de la carrera. Sin embargo, se observaron problemas de entendimiento de necesidades en los 3 proyectos realizados. A los estudiantes les costó mucho traducir los requerimientos del consultor en un diseño formal que pudieran ser reutilizados en proyectos futuros; de hecho, se observó una mala costumbre por programar sin diseñar o elaborar arquitecturas. El desarrollo de proyectos con una empresa real permitió introducir a los estudiantes a un proceso real en donde fue necesario generar documentación que podía ser aceptada o rechazada. La empresa proporcionó sus plantillas basadas en el estándar 830-1998 de la IEEE para la especificación de requerimientos, las cuales se combinaron con el diseño basado en casos de uso que aprendieron en la materia de “Programación orientada a objetos II”.

5.3.2. Problemas de negocio

La empresa participante entendió rápidamente que los estudiantes han asimilado en su formación una forma desorganizada de trabajo sin planes, calendarios o estimaciones de esfuerzo/tiempo y fue necesario que se ajustaran estas prácticas para obtener buenos resultados. En relación a esto fue necesario dar un curso básico a los estudiantes en el manejo de las plantillas de TSP (Team Software Process) para respetar los tiempos y estimaciones de tamaño/esfuerzo establecidas por la empresa. Cabe mencionar que los 4 estudiantes utilizados como equipo de desarrollo cursaron la carrera con el plan de estudios antiguo, y el problema podría solventarse con alumnos que cursan el nuevo plan que ya incluye cursos como PSP y modelos de calidad.

Por otro lado, a pesar que desde un inicio la empresa se mostró interesada en participar activamente en la formación de los estudiantes participantes, ésta mostró cierto grado de incertidumbre dado que nunca había participado en un proyecto de este tipo. En este sentido, se presentaron ciertos problemas relacionados con la incompatibilidad de horarios dado que era imposible que el consultor estuviera disponible en las horas programadas, por lo que se decidió establecer una interfaz basada en la actualización de archivos para obtener revisiones de acuerdo al tiempo disponible del evaluador. Por último, al final del desarrollo de los dos proyectos, el consultor expresó su interés en seguir participando con más proyectos de diferentes complejidades y tamaños.

5.4. Comprobación de la hipótesis

A partir de los resultados logrados, en esta sección se comprueba o rechaza la hipótesis planteada en el Capítulo 1 del presente trabajo de investigación, Sección 1.4. A continuación se resume la comprobación de la hipótesis.

- Hipótesis: *“El establecimiento de un modelo dinámico para la educación de la Ingeniería de Software permitirá reducir las diferencias entre la oferta académica de las Universidades Regionales y las necesidades reales de la industria de software”.*

Esta hipótesis se comprobó dada la información obtenida con la aplicación de VinculAE a proyectos reales y con la incorporación de dos empresas a la red de trabajo. Esta información puede analizarse de la siguiente forma:

A través del establecimiento de un modelo dinámico de enseñanza, apoyado por un plan curricular (sección 3.2) y una herramienta computarizada para vincular proyectos reales con la industria (Capítulo 4); fue posible trabajar cercanamente con las necesidades reales de las empresas en cuanto a habilidades específicas que éstas demandan de los estudiantes. En este sentido, el desarrollo de una herramienta para la gestión de carteras de proyectos para la empresa RagaSoft S.A. de C.V. (Capítulo 5), permitió que los estudiantes se acercaran más a la aplicación del conocimiento aprendido en las aulas y recibieran capacitación sobre aquellos aspectos que posiblemente no se incluyen en el plan de estudios actual. Se considera que dados los buenos comentarios obtenidos por parte de la empresa vinculada (Apéndice G), esta tesis podría ser el primer paso en la consolidación de un acercamiento que permita reducir las diferencias entre la Universidad y la industria local.

Capítulo 6

Conclusiones.

Una de las motivaciones iniciales de este trabajo consistió en ofrecer una investigación que mostrara las inquietudes y perspectivas que tienen los estudiantes sobre el sector académico y el empresarial, y desarrollar una solución que ayudara a los alumnos a tener una vinculación con el sector laboral al cual se integrarán al terminar sus estudios.

El objetivo principal se enfocó en proponer un modelo dinámico para la educación de la materia de Ingeniería de Software que reduzca la brecha entre la Universidad y la Industria, que permita que los alumnos fortalezcan los conocimientos teóricos adquiridos en clase con experiencias de los profesionales de la Industria. De esta forma, el desarrollo del presente trabajo cumplió con los objetivos planeados al inicio de la tesis, dado que la información recogida al final del desarrollo de un sistema permitió comprobar satisfactoriamente la hipótesis establecida para guiar el desarrollo del trabajo. En este sentido, la solución planteada en esta tesis se compone de dos elementos principales: (1) un plan curricular que permite establecer las bases teóricas del modelo en base al plan de estudios actual de la UTM, este modelo respeta el contenido temático de cada materia y solamente sugiere la formación en algunos temas específicos que posiblemente no sean incluidos en la educación del estudiante, pero que la empresa puede requerir para el desarrollo de los proyectos, y (2) la herramienta VinculAE, que se diseñó en base a la Metodología Extendida de Diseño Centrado en el Usuario, y que permite establecer un vínculo entre la Universidad y el sector empresarial que ayuda en la formación profesional de los estudiantes.

Así, VinculAE representa una herramienta útil para establecer vinculación entre los estudiantes, el sector laboral y los profesores. Durante el proceso de desarrollo de VinculAE se observó que es necesario considerar a los usuarios, sus actividades, necesidades, lo anterior con el objetivo de diseñar una herramienta lo más cercana posible a las necesidades detectadas, lo cual puede ahorrar tiempo, esfuerzo y dinero a gran escala dependiendo del tamaño del desarrollo.

Cabe mencionar que este trabajo deja en evidencia que es necesario también conocer las tecnologías, herramientas, procesos que se utilizan actualmente en la industria como sector laboral, para que al dejar la parte académica la integración del estudiante sea lo más rápida posible.

Por último, se considera que la solución propuesta establece las bases de un puente de vinculación entre alumnos, profesores, universidades con empresas, el mercado laboral, el mundo

real, que establezca un mayor acercamiento entre las empresas que demandan una necesidad de carácter real, con equipos de trabajo que pueden solucionar dichas necesidades. A través del enfoque presentado, todos los actores involucrados en los proyectos pueden darle seguimiento a los mismos a través del sistema.

Para concluir, se resumen las principales aportaciones realizadas en esta tesis:

- La implementación de un sistema que permita la vinculación entre la academia y la empresa.
- El sistema reúne en un mismo punto, necesidades y soluciones, conecta soluciones reales a problemas reales.
- La investigación realizada para fundamentar esta tesis permitirá que trabajos posteriores puedan apoyarse en ésta para que sea aplicada en otras investigaciones.
- VinculAE puede integrar a todas las universidades del SUNEIO para trabajar de manera conjunta en un proyecto a lo largo de todas sus fases.

Al final de la tesis es posible concluir que los estudiantes que han tenido contacto real con las empresas dentro del ambiente académico, y no solo por el generado principalmente por el espacio de prácticas profesionales, pueden integrarse con menos problemas al ámbito laboral, mediante el fortalecimiento de las habilidades obtenidas durante su formación académica.

Bibliografía

- [1] ISO/IEC. (2004). *ISO/IEC 15504-2:2003/Cor.1:2004(E): Information Technology - Process Assessment - Part 2: Performing an Assessment. International Organization for Standardization*. Geneva, 2004. 3.1
- [2] ISO/IEC 12207: 2008. *Systems and software engineering - Software life cycle processes*. Geneva, 2008. 3.1
- [3] P. Bourque A. Abran, J. W. Moore and R. Dupuis. *Guide to the Software Engineering Body of Knowledge: 2004 Edition - SWEBOOK*. IEEE Computer Society, April 2005. 2.1.2, 2.2.2
- [4] S. R. L. Meira A. K.O. Moraes and J. O. Albuquerque. Open source software factory - step by step: A case report. *In: First International Conference on Open Source Collaborative Development Platforms,, 2006*. 2.1.2
- [5] D. Henry K. Lasfer L. Bernstein. A.Pyster, R. Turner. The current state of software engineering masters degree programs. *Proceedings, 21st Conference on Software Engineering Education and Training*, pages 103–109, April 2008. 2.2.2
- [6] M. Barker and K. Inoue. It spiral : A case study in scalable software engineering education. *22nd Conference on Software Engineering Education and Training*, pages 53–60, 2009. 2.2, 2.2.2, 2.2.3, 2.5, 2.5
- [7] N. Bevan and J. Kirakowski, J. Maissel. What is usability? *Proceedings of the 4th International Conference on HCI, Stuttgart,, Elsevier., September 1991*. 4
- [8] P.B. Crosby. *Quality Is Free: The Art of Making Quality Certain*. McGraw Hill, 1979. 2.3.2
- [9] D. Damian and D. Zowghi. Requirements engineering challenges in multi-site software development organizations. *Requirements Engineering Journal*, 8(1):149–160, 2003. 2.3
- [10] D. Damian, A. Hadwin, and B. Al-Ani. Instructional design and assessment strategies for teaching global software development: A framework. *Proceedings, 28th Intl. Conf. Software Engineering (ICSE 06), Shanghai, China*, pages 20–28, May 2006. 2.3
- [11] S. Deshpande and I. Richardson. Management at the outsourcing destination global software development in india. pages 217–225. IEEE Computer Society, 2009. 1.1
- [12] J.L Diaz-Herrera and T.A. Hilburn. Software engineering 2004: Curriculum guidelines for undergraduate degree programs in software engineering. 2004. URL <http://sites.computer.org/ccse/SE2004Volume.pdf>. 2.4, 2.4.3
- [13] A. Dogru and M. Turhan-Yondem. Software engineering field and curriculum adaptation in turkey. *Proceedings of the 2008 International Workshop on Software Engineering in East*

and South Europe, art. ACM Publisher, 2008. 3.2

- [14] S.C. dos Santos, A.P.C. da Conceicao Moraes Batista, M. Cavalcanti, J.O. Albuquerque, and S.R.L. Meira. Applying pbl in software engineering education. *Proceedings 22nd Conference on Software Engineering Education and Training CSEET 09, Hyderabad, Andhra Pradesh*, pages 182–189, Feb 17-20 2009. 1.5, 2.1.1, 2.5, 2.5
- [15] J.F. Dumas and J.C. Redish. *A Practical Guide to Usability Testing*. Intellect Books, 1999. 4.2.1.1
- [16] J. Bosch E. Folmer. Architecting for usability: a survey. *Journal of Systems and Software*, 70:61–78, Febrero 2004. 4
- [17] C. Faulkner. *The Essence of Human-Computer Interaction*, volume 1. The Prentice Hall, 1 edition, November 1997. HCI. 4.2.1.1
- [18] J. Favela and F. Peña-Mora. An experience in collaborative software engineering education. *IEEE Software*, 18(2):47–53, March-April 2001. 2.3
- [19] J. Fenwick and B. Kurtz. Intra-curriculum software engineering education. *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, art. ACM Publisher, 2005. 3.2
- [20] G. Ford and N. Gibbs. A master of software engineering curriculum - recommendations from the software engineering institute. *IEEE Computer*, 22(9):59–71, September 1989. 3.2
- [21] D. Petkovic G. Thompson and R. Todtenhoefer. Assessment and comparison of local and global software engineering practices in a classroom setting. *Proceedings, 13th Conf. on ITiCSE, Madrid, Spain*, June 30 - July 02 2008. 2.3
- [22] R. Harper, T. Rodden, Y. Rogers, and A. Sellen. *Being Human: Human-Computer Interaction in the year 2020*. Microsoft Research Ltd, 2008. 4, 4.1, 4.1, 4.1
- [23] Y. Hassan, F. Martín, D. Hassan, and O. Martín. Arquitectura de la información en los entornos virtuales de aprendizaje. aplicación de la técnica de card sorting y análisis cuantitativo de los resultados. *El Profesional de la Información*, 13(2):93–99, 2004. 4
- [24] M.J. Hawthorne and D.E. Perry. Software engineering education in the era of outsourcing, distributed development, and open source software: Challenges and opportunities. *Proceedings, 27th Intl. Conf. on Software Engineering (ICSE 05), St. Louis, Missouri, USA*, May 15-21 2005. 2.3
- [25] J. Herbsleb and D. Moitra. Global software development. *Journal.*, 18(2):16–20, March-April 2001. 1.2
- [26] J.D. Herbsleb. Global software engineering: The future of socio-technical coordination. *Proceedings, 29th Intl. Conf. on Software Engineering - The Future of Software Engineering (ICSE-FASE 07), Minneapolis, USA*, pages 188–198, May 20-26 2007. 2.3
- [27] A.M. Hickey and A. Davis. A unified model of requirements elicitation. *Journal of Management Information Systems*, 20(4):65–84, Spring 2004. 1
- [28] W. L. Honig. Teaching successful real-world software engineering to the net generation:

Process and quality win. *Proceedings, 21st Conference on Software Engineering Education and Training*, pages 25–32, April 2008. 2.2.2

- [29] W. Humphrey. *PSP: A Self-improvement Process for Software Engineers. The SEI Series in Software Engineering*. Addison-Wesley Professional, 2005. 1.2
- [30] W. Humphrey. *TSP Leading a Development Team. The SEI Series in Software Engineering*. Addison-Wesley Professional, 2005. 1.2
- [31] W. Humphrey. *Software: The competitive edge*. Carnegie Mellon University, 2008. 1.2
- [32] Software Engineering Institute. *CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SW/IPPD/SS, V1.1)*. CMU/SEI-2002-TR-011., 2002. 1.6, 3.1
- [33] Software Engineering Institute. *CMMI for Development (CMMI-DEV, V1.2) CMU/SEI-2006 TR-008*. Carnegie Mellon University, 2006. 1.6, 3.1
- [34] Ireland, 2009. URL <http://www.idaireland.com/>. 1.1
- [35] I. Jacobson, M. Christerson, P. Jonsson, and G. Overgaard. *Object-Oriented Software Engineering, a use case driven approach*. Addison-Wesley, 1992. 1.1
- [36] N. Khan, W. Currie, V. Weerakkody, and B. Desai. Evaluating offshore it outsourcing in india: Supplier and customer scenarios. *Proceedings of the 36th Hawaii International Conference on Systems Sciences (HICSS 03)*, pages 239–249, 2003. 1.1
- [37] B. Kitchenham, D. Budgen, P. Brereton, and P. Woodall. An investigation of software engineering curricula. *Journal of Systems and Software*, 74(3):325–335, 2004. 3.2
- [38] V. Kulkarni, O. Gotel, L. Neak, and C. Scharff. Working across borders: Overcoming culturally-based technology challenges in student global software development. *Proceedings, 21st CSEE&T 08, Charleston, S. Carolina, USA*, April 14-17 2008. 2.3.1, 2.5, 2.5
- [39] S. Lawrence. *Software Engineering Theory and Practice*. Upper Saddle River, 3rd edition edition, 2005. 2.4, 1
- [40] L. Li and S. Wu. *The Thought of Two Years Advanced Software Developer Training according to NIIT Model*. Higher Education Explorer, 2004. 1.1
- [41] L. Li, S. Wu, and J. Luo. Learning from practicing: Adaptation of niit(india) model for software developer training to chinese higher vocational and technical colleges. *Proceedings of the 9th International Conference for Young Computer Scientists*, pages 2454–2458, 2008. 1.1
- [42] C. W. Liew. Teaching software development skills early in the curriculum through software engineering. *Proceedings of the 10th annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE 05)*, art. ACM Publisher, 2005. 3.2
- [43] C. R. Mitra. The niit academy: A unique university in the making. 18(3):17–23, September 2008. 1.1
- [44] N.B. Moe and Smite D. Understanding lacking trust in global software teams: A multi-case study. *Proceedings Conf. on Product Focused Software Process Improvement (PROFES 07), Riga, Latvia*, 4589:20–34, 2007. 2.3

- [45] MOODLE, 2009. URL <http://moodle.org/>. 2.1.2
- [46] I. B. Myers. *Gifts Differing: Understanding Personality Type*. CA: Davies-Black Publishing,, 1980. 1
- [47] J. Nielsen. *Usability Engineering*. 1993. 4.2.1.1
- [48] D. Norman. *The Design of Everyday Things*. Basic Books, Inc., 2002. 4.1
- [49] D. Norman and S. Draper. *User Centered System Design; New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1986. 4.1
- [50] H. Oktaba. Moprosoft: A software process model for small enterprises. Special Report CMU/SEI-2006-SR-001, Carnegie Mellon University., 2006. 1.2
- [51] J.S. Olson and G.M. Olson. Culture surprises in remote software development teams. *ACM Queue*, 1(9):52, Dec-Jan 2003-2004. 2.3
- [52] D. Parnas and M. Lawford. The role of inspection in software quality assurance. *Journal, IEEE Transactions on Software Engineering*, 29(8):674–676, August 2003. 2.4.2
- [53] M. Peterson. Skills to enhance problem-based learning. *Medical Education Online*, 2(3), 1997. 2.1.2
- [54] C. K. Prahalad and M. S. Krishnan. The dynamic synchronization of strategy and information technology. 43(4):24–33, 2002. 1.1
- [55] J. Preece. *Human Computer Interaction (ICS)*, volume 1. Addison Wesley, 1 edition, Mayo 1994. 4.2.1.1
- [56] M. Purvis, M. Purvis, and S. Cranefield. Educational experiences from a global software engineering gse project. *Proceedings, 6th Conf. on Australasian Computing Education ACE04 Dunedin New Zealand*, pages 269–275, 2004. 2.3
- [57] L. R. C. Ribeiro and M.G. Mizukami. An experiment with pbl in higher education as appraised by the teacher and students. *Interface - Comunic*, 9(17):357–68, mar-ago 2005. 2.1
- [58] S. Russell. Iso 9000:2000 and the efqm excellence model: competition or cooperation?. *Total Quality Management*, 11(4,5 & 6):657–665, 2000. 3.1
- [59] J. Satzinger, R. Jackson, and S. Burd. *System Analysis and Design*. 2000. 2.4
- [60] J.R. Savery and T.M. Duffy. Problem based learning: An instructional model and its constructivist framework. *Constructivist Learning Environments: case studies in instructional desing*, 1(1):239, Enero 1996. 2.1, 2.1.2
- [61] SEP. Principales cifras 2010 - 2011. URL http://www.sep.gob.mx/work/models/sep1/Resource/1899/2/images/principales_cifras_2010_2011.pdf. 4.2.3.2.2
- [62] SEP. Secretaria de educación pública, Noviembre 2012. URL <http://www.sep.gob.mx/>. 4.2.3.1.2
- [63] SEP. Secretaria de educación pública: Comunicado 176: Cimentar en el talento y la energía de los jóvenes, el desarrollo y competitividad de México., Noviembre 2012. URL <http://www.sep.gob.mx/es/sep1/C1761112>. 4.2.3.1.2

- [64] R. Shackelford, A. McGettrick, R. Sloan, H. Topi, G. Davies, R. Kamali, J. Cross, J. Impagliazzo, R. LeBlanc, and B. Lunt. Computing curricula 2005: The overview report. *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*, art. ACM Publisher, 2006. 3.2
- [65] S. Skandan and M. Sidhardhan. Software engineering education at tcs introduction an experience report. pages 16–19, 2009. 1.1
- [66] P. Tynala. Towards expert knowledge, a comparison between a constructivist and a traditional learning environment in the university. *International Journal of Educational Research*, 31(5):357–442, 1999. 2.1, 2.1.2
- [67] UNCTAD, 2009. URL <http://www.unctad.org/>. 1.1
- [68] UNIVERSIA. Universia, Octubre 2012. URL <http://www.universia.net.mx/>. 4.2.3.1.2
- [69] UPA. Upa usability professional’s association. URL http://www.usabilityprofessionals.org/usability_resources/. 4.2.7
- [70] H. Wang, Y. Huo, and C. Jin. Design and implementation of a software engineering practice course. *2010 Second International Workshop on Education Technology and Computer Science*, pages 563–566, 2010. 2.4.1, 2.4.3, 2.5, 2.5
- [71] L. Waverman and K. Dasgupta. Scorecard 2009. Technical report, Nokia Siemens Networks, 2009. 1.1

Apéndice A

Acrónimos

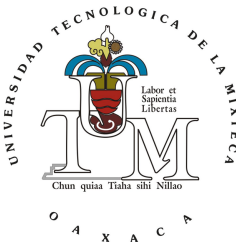
- AMESOL** Asociación Mexicana Empresarial de Software Libre
- AMIPCI** Asociación Mexicana de Internet
- ANIEI** Asociación Nacional de Instituciones de Educación en Informática
- ANUIES** Asociación Nacional de Universidades e Instituciones de Educación Superior
- CANIETI** Cámara Nacional de la Industria Electrónica, de Telecomunicaciones y Tecnologías de Información
- CESAR** Centro de Estudios y Sistemas Avanzados de Recife
- COFETEL** Comisión Federal de Telecomunicaciones
- CONACYT** Consejo Nacional de Ciencia y Tecnología
- CONAPO** Consejo Nacional de Población
- DCU** Diseño Centrado en el Usuario
- DCUE** Diseño Centrado en el Usuario Extendida
- GSD** Global Software Development
- HCI** Human Computer Interaction
- INEGI** Instituto Nacional de Estadística y Geografía
- MoProSoft** Modelo Mexicano de procesos
- PROMEDIA** Programa de Desarrollo de la Industria de Medios Interactivos
- PSP** Personal Software Process
- PYME** Pequeña y Mediana Empresa
- SUNEO** Sistema de Universidades del Estado de Oaxaca
- TIC** Tecnologías de Información y Comunicación
- TMCA** Tasa Media de Crecimiento Anual
- TSP** Team Software Process
- UMAR** Universidad del Mar

UNISTMO Universidad del Istmo
UNPA Universidad del Papaloapan
UNSIJ Universidad de la Sierra Juárez
UNSI Universidad de la Sierra Sur
UTM Universidad Tecnológica de la Mixteca

Apéndice B

Cuestionarios

B.1. Cuestionario Empresarios



Ingeniería en Computación

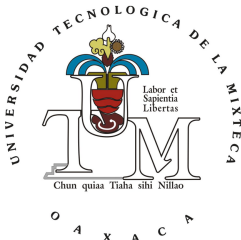
Empresa: _____

Fecha: ____/____/____

1. ¿Su empresa recibe alumnos para realizar prácticas profesionales?
SI ____ NO _____
2. ¿Los alumnos durante las prácticas profesionales, manifiestan alguna inquietud sobre los requerimientos del sector profesional?
SI ____ NO _____ ¿Cuáles? _____
3. ¿Le gustaría trabajar algún proyecto de su empresa con alguna institución educativa?
SI ____ NO _____
4. ¿Conoce qué áreas de conocimientos tienen mayor demanda en la industria del software?
5. ¿Su empresa participa en algún programa de vinculación entre la universidad y el sector empresarial en el área de desarrollo, implementación o capacitación de software?
SI ____ NO _____ ¿Cuál? _____
6. ¿Considera que el conocimiento de los estudiantes practicantes en su empresa es el adecuado para enfrentar el área laboral al término de sus estudios?
SI ____ NO _____ ¿Por qué? _____

7. ¿Qué beneficios ofrecería a alumnos que trabajen a través de un proyecto real con su empresa?
8. Mencione 3 requisitos que considera debe cumplir un sistema de vinculación Academia – Empresa.

B.2. Cuestionario Profesores



Ingeniería en Computación

Área / Departamento: _____

Fecha: ____/____/____

1. ¿Sus alumnos han hecho prácticas profesionales?
SI ____ NO _____
2. ¿Sus alumnos después de estar en prácticas profesionales, manifiestan alguna inquietud sobre los requerimientos del sector profesional?
SI ____ NO _____ ¿Cuáles? _____
3. ¿Le gustaría trabajar algún proyecto real en el desarrollo de software para alguna Empresa o Centro de Investigación con un equipo de sus alumnos?
SI ____ NO _____
4. ¿Conoce algún programa de vinculación entre la universidad y el sector empresarial en el área de desarrollo, implementación o capacitación de software?
SI ____ NO _____ ¿Cuál? _____
5. ¿Considera que el modelo de aprendizaje es el adecuado para los estudiantes, para enfrentar el área laboral al término de sus estudios?
SI ____ NO _____ ¿Por qué? _____
6. ¿Cuáles son sus temas de interés en computación, especialmente en el área de desarrollo de software?
7. ¿Le gustaría evaluar a sus alumnos a través de un proyecto real?

8. Mencione 3 requisitos que considera debe cumplir un sistema de vinculación Academia – Empresa.

B.3. Cuestionario Alumnos



Ingeniería en Computación

Licenciatura en Informática

Semestre: ___ Fecha: ___/___/___

1. ¿Has hecho prácticas profesionales?
SI ___ NO _____
2. ¿Cuál consideras es el área fuerte de tus conocimientos en computación?
3. ¿Te gustaría trabajar en el desarrollo de software para alguna Empresa o Centro de Investigación?
SI ___ NO _____
4. Menciona algunas Empresas o Centros de Investigación que se dediquen al desarrollo de Software.
5. ¿Sabes de los conocimientos que requiere y demanda la industria del software?
SI ___ NO _____
6. ¿Conoces algún programa de vinculación entre la universidad y el sector empresarial en el área de desarrollo, implementación o capacitación de software?
SI ___ NO _____ ¿Cuál? _____
7. ¿Consideras que tu modelo de aprendizaje es el adecuado para enfrentar el área laboral, al término de tus estudios?
SI ___ NO _____ ¿Por qué? _____
8. ¿Cuáles son tus temas de interés en computación, especialmente en el área de desarrollo de software?
9. ¿Te gustaría aprender trabajando en proyectos reales?
SI ___ NO _____ ¿Por qué? _____

10. Menciona 3 requisitos que consideras debe cumplir un sistema de vinculación Academia – Empresa.

11. ¿Tienes acceso a Internet?

SI ___ NO ___ frecuencia de uso _____

Apéndice C

Registro de usuario

Nombre del usuario: _____

Fecha: _____/_____/_____

Estimado usuario:

Agradecemos su presencia y disposición para realizar las pruebas, en estos momentos se requiere probar el grado de aceptación y funcionamiento de las interfaces del sistema VinculAE, el cual se refiere a un sistema de vinculación entre la Academia y la Empresa.

Se pretende ver si las interfaces son fáciles de entender y si la información presentada corresponde a las necesidades de usted como usuario del sistema.

Le pedimos realizar las tareas con las interfaces del sistema, mismas que se le entregaran en una hoja por separado.

Le recomendamos leer cada tarea atentamente antes de empezar. Al terminar las tareas, le pedimos por favor responder un cuestionario, sus respuestas y comentarios nos brindaran información importante con respecto al sistema. El facilitador le brindará apoyo cuando se considere necesario.

Sus observaciones ayudarán a conocer o encontrar deficiencias en el sistema y en base a ello realizar mejoras en el sistema. Nos interesan sus impresiones por lo que no dude en manifestar sus inquietudes durante el transcurso de la prueba así como observaciones y comentarios en la hoja del cuestionario. Si usted no tiene ninguna pregunta, duda o comentario, podemos dar inicio a la prueba.

Gracias por su colaboración y apoyo.

Apéndice D

Tareas

Tarea 1

Abrir y entrar al sistema VinculAE

- a) Ubicar y seleccionar VinculAE
- b) Teclear nombre de usuario
- c) Teclear contraseña
- d) Salir del sistema

Tarea 2

Realizar una búsqueda de un proyecto en VinculAE

- a) Ubicar la opción de búsqueda en VinculAE
- b) Escribir el texto de búsqueda
- c) Realizar la búsqueda
- d) Salir del sistema

Tarea 3

Consultar proyectos en VinculAE

- a) Ubicar la sección de proyectos en VinculAE
- b) Seleccionar proyectos
- c) Seleccionar el proyecto de su interés
- d) Salir del sistema

Tarea 4

Dar de alta una propuesta de proyecto en VinculAE

- a) Ubicar sección de propuestas en VinculAE
- b) Seleccionar dar de alta
- c) Teclear los datos seleccionados
- d) Guardar los datos tecleados
- e) Salir de VinculAE

Apéndice E

Hoja de resultados de las tareas

1 ¿Las tareas fueron fáciles de realizar?

Totalmente de acuerdo

De acuerdo en gran parte

Algo de acuerdo

Algo en desacuerdo

Totalmente en desacuerdo

2 La explicación en la interfaz del sistema fue:

Buena Mala

Difícil Sencilla

Clara Confusa

Comprensible Incomprensible

3 ¿en algún momento no entendió alguna indicación en las tareas?

Frecuentemente

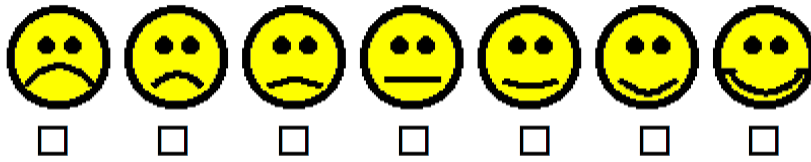
Algunas veces

Raramente

Muy raramente

Nunca

4 ¿Qué tan satisfecho estuvo después de terminar las tareas? ¿Qué cara corresponde a su estado de ánimo?



5 ¿Ha tenido problemas con las tareas? ¿Tiene alguna propuesta?

Apéndice F

Publicaciones

Coloquio Nacional de Investigación en Ingeniería de Software y Vinculación Academia-Industria

Aprender a través de la Práctica: Definiendo un Modelo Alternativo para la Educación de la Ingeniería de Software en las Universidades Mexicanas como medio para reducir el vacío entre la Industria y la Academia

García, I., Pacheco, C.
División de Estudios de Posgrado
Universidad Tecnológica de la Mixteca
Oaxaca, Mexico
{ivan,leninca}@mixteco.utm.mx

Coronel, N.
Media Lab, División de Estudios de Posgrado
Universidad Tecnológica de la Mixteca
Oaxaca, Mexico
coronel@mixteco.utm.mx

Abstract—En la actualidad, los avances en la educación, investigación y desarrollo del software se están incrementando en una forma considerable; el problema es que se está haciendo de forma separada. Esto genera que las empresas de software inviertan grandes cantidades de dinero en la formación de su personal joven, dado que las Universidades no incluyen en su currícula temas relacionados con las necesidades reales de esta industria. En este momento, el interés a nivel global se centra en desarrollar modelos educativos alternativos enfocados a la Ingeniería de Software, y mejorar así la adquisición de experiencias y habilidades dentro de proyectos reales. Países como Brasil, India, Estados Unidos de Norteamérica, Japón, Australia y Tailandia han creado iniciativas dentro de este contexto. Los resultados de sus investigaciones les han permitido encontrar soluciones futuras para fomentar y mejorar la industria de software. Este artículo pretende mostrar las bases de un modelo dinámico que permita alinear los objetivos y las necesidades de la industria Mexicana de software de acuerdo con la oferta actual de las Universidades que imparten esta disciplina. A largo plazo, esto asegurará que la industria obtenga el talento adecuado (de acuerdo a habilidades y capacidades apropiadas) y que los estudiantes obtengan oportunidades para trabajar en la industria de software del país.

Keywords- modelo dinámico, educación, ingeniería de software, industria de software

I. INTRODUCCIÓN

De acuerdo con Jacobson [6], la Ingeniería de Software (IS) se define como "un conjunto de etapas parcialmente ordenadas con el propósito de alcanzar un objetivo, en este caso, de obtener un producto software de calidad" y el proceso de desarrollo software es "el proceso donde las necesidades del usuario son traducidas en requisitos de software; estos requisitos son transformados en un diseño; este diseño es codificado; el código obtenido es probado, documentado y validado en su uso operativo". Concretamente, "este proceso define quién hace qué, cuándo, y cómo para alcanzar tal objetivo". Históricamente es bien sabido que la IS surgió como

una medida para evitar la crisis del software identificada en 1969, sin embargo esto no ha sucedido todavía. En este contexto, el diccionario Webster define al término "crisis" como: "un punto decisivo en el curso de algo; un momento, etapa o evento crucial" [13]. No obstante, en el desarrollo de la IS no ha habido un punto crucial, ningún momento decisivo, sólo un cambio evolutivo retardado. De acuerdo con Daniel Tichrow, sería más preciso definir esta situación como: "una aflicción crónica, algo que perdura o reaparece frecuentemente y continua de una forma indefinida".

Pero de hecho, con crisis o sin ella, la demanda de software está en aumento por lo que ha dejado de ser un simple producto o proceso para convertirse en una industria completa. La mayoría de los países ven a la producción de software como una actividad económica que se caracteriza por generar alto valor agregado y contribuir a la economía con productos y servicios esenciales para su modernización. Esta industria posee las siguientes características: está basada en el conocimiento que desarrolla capacidades más allá de la manufactura; propone que la innovación tecnológica genere trabajos bien remunerados; no contamina, y requiere relativamente poco capital para comenzar. Así, la industria de software es parte del grupo de actividades que conforman a las Tecnologías de la Información (TI). La incorporación de la TI en la comercialización productiva, servicios, educación y procesos de la Administración Pública, es un factor clave para mejorar la competitividad de las organizaciones y de los países.

Aquí, el punto interesante es el hecho de que una proporción creciente de la producción de software en todo el mundo se lleva a cabo en los denominados "países en vías de desarrollo". India e Irlanda representan dos casos exitosos de creación y crecimiento de industrias nacionales basadas en la exportación. India, por ejemplo, ha estado desarrollando software para los Estados Unidos de Norteamérica por 15 años. ¿Cómo lo hace? De acuerdo con [12], [1] y [14], el éxito de India está relacionado con las enormes inversiones (billones de

Learn from Practice: Defining an Alternative Model for Software Engineering Education in Mexican Universities for Reducing the breach between Industry and Academia

Garcia, I., Pacheco, C.

Computer Science Faculty, Postgraduate Department
Technological University of the Mixtec Region
Oaxaca, Mexico
ivan@mixteco.utm.mx

Coronel, N.

Media Lab, Postgraduate Department
Technological University of the Mixtec Region
Oaxaca, Mexico
coronel@mixteco.utm.mx

Abstract—Nowadays, advancements in education, research and software development are increasing in a considerable way, but each part is doing it independently. This problem generates that software enterprises spend a lot of money in the formation of young personnel, because Universities do not include in their curriculums topics related to Industry necessities. At this moment, there is an increasing interest at global level focused on developing alternative educative models oriented to Software Engineering and improving the acquisition of experiences and abilities within real projects. Brazil, India, United States of America, Japan, Australia and Thailand are creating initiatives in this context. The results of their research will enable them to find future solutions for fomenting and improving the software industry. This paper aims to show the basis of a dynamic model to align the objectives and necessities of Mexican software industry with the real offer from Universities who teach the discipline. This will make sure that industry gets the suitable talent (appropriate skills and capabilities) and students get suitable opportunities to work within the software industry.

Keywords- dynamic model, software engineering education, software industry,

I. INTRODUCTION

According to Jacobson [5], the Software Engineering area is defined as “a set of stages partially arranged with the aim to achieve an objective, in this case, to obtain a quality software product” and the software development process is “the process where user necessities are translated into software requirements, these requirements are transformed into a design, the design is codified, the obtained code is tested, documented and validated for operative use” Concretely, “this process defines who is doing what, when, and how to reach the objective” Historically, it is well known that Software Engineering arose as a way to avoid the software crisis identified in 1969, however such thing has not happened yet. In this context, the Webster dictionary defines the “crisis” term as: “a decisive point in the course of something; a crucial moment, stage or event” [16]. Nevertheless, in the passing of Software Engineering there was no crucial point, none decisive moment, only a slowed evolutionary change. According to Daniel

Tiechrow¹, it would be more precise define this situation as: “a chronic affliction, something that endures or reappears frequently and continues in an undefined way”.

But in fact, with crisis or without it, the software demand is increasing and has changed from being a single product or process to a whole industry. Most of the countries see the software production as an economic activity which is characterized for generating a high added value and contributing to the economy with essential products and services for its modernization. This industry is based on knowledge which develops abilities beyond manufacturing, and proposes that technological innovation generates well paid jobs; it does not contaminate, and requires relatively little capital to start. The software industry is part of the economic activities group which conform to Information Technologies (IT). The incorporation of IT in productive commercialization, services, education and public administration processes, is a key factor for improving the organization’ and countries’ competitiveness.

But, the interesting point is the fact that one increasing proportion of the worldwide software production is performed in “developing countries”. India and Ireland represent two successful cases of creation and growing of national industries based on exportation. India, for example, has been developed software for the United States of America for 15 years. How it does it? According to [11], [1] and [13] India’s success is related to huge inversions (billions of dollars) in formation (training), certification, promotion and infrastructure. But, one of the key causes for the extraordinary growth of India’s software industry is that India has developed an effective method of training an advanced and functional software workforce and turned into a globally recognized model of such training, the NIIT model. On the other hand, Ireland specialized in developing customized applications and focused the development of its software industry to attract foreign enterprises. The Industrial Development Authority states that

¹ This terminology was suggested by Professor Daniel Tiechrow of the University of Michigan in a talk presented in Geneva, Switzerland, April 1989.

Apéndice G

Carta de colaboración



www.gruporagasoft.com
soluciones@gruporagasoft.com

Oaxaca de Juárez, Oaxaca a 01 de Octubre de 2012


A quien corresponda:

A través de la presente confirmo que el C. Noé Coronel García ha recogido información relacionada con proyectos de software que han sido desarrollados de forma vinculada con la empresa que dirijo desde el 2006, como parte de su trabajo de tesis denominado "Propuesta de un modelo dinámico para la educación de la materia de Ingeniería de Software que reduzca la brecha entre la Universidad y la Industria". Específicamente, reconozco nuestra participación y guía a los estudiantes en el desarrollo del "Sistema para gestionar carteras de proyectos informáticos en pequeñas empresas".

Por último, en RagaSoft consideramos que esta tesis no brindará beneficios únicamente en el contexto académico puesto que puede ser un esfuerzo importante por ampliar los horizontes en cuanto a la formación de recursos humanos de calidad que requiere la industria actualmente.

En caso de requerir más información, por favor no duden en contactarme.

Quedo de Uds.


Ing. Alberto Martínez Ramírez
Director General
RagaSoft S.A. de C.V.

RagaSoft
Desarrollamos Soluciones Educativas
RFC: RAG101110AM0
Calle Privada de Hacedorito Alcazar
1201 A, Dize Oaxaca
Oaxaca de Juárez,
Oaxaca 68840