



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

---

---

**FACULTAD DE CIENCIAS**

**DESARROLLO DE SOFTWARE GUIADO POR LA  
NORMA ISO/IEC 29110 Y SCRUM: SIDEV V.2.0**

**T E S I S**

**QUE PARA OBTENER EL TÍTULO DE:**

**LICENCIADO EN CIENCIAS DE LA  
COMPUTACIÓN**

**P R E S E N T A:**

**ERICK ORLANDO MATLA CRUZ**



**DIRECTOR DE TESIS:  
M. EN I. MIGUEL EHÉCATL MORALES  
TRUJILLO  
2014**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**1. Datos del alumno:****Matla****Cruz****Erick Orlando****55931431****Universidad Nacional Autónoma de México****Facultad de Ciencias****Ciencias de la Computación****304025318****2. Datos del tutor****M. en I.****Miguel Ehécatl****Morales****Trujillo****3. Datos del sinodal 1****Dra.****Amparo****López****Gaona****4. Datos del sinodal 2****Dra.****Hanna****Oktaba****5. Datos del sinodal 3****M. en C.****Gustavo Arturo****Márquez****Flores****6. Datos del sinodal 4****M. en C.****María Guadalupe Elena****Ibargüengoitia****González****7. Datos de la Tesis****Desarrollo de Software Guiado por la Norma****ISO/IEC 29110 y SCRUM: SIDEP V.2.0****229p****2014**

---

# Agradecimientos

Cuando una persona termina un trabajo tan arduo y complicado como lo es el desarrollo de una tesis, es inevitable que el egocentrismo lo asalte a uno. Sin embargo, un análisis poco profundo permite darse cuenta que este logro jamás habría sido posible sin el aporte y participación de personas e instituciones quienes han facilitado las cosas para que este trabajo llegue a un feliz término.

Debo comenzar por agradecer a mis padres Margarito Matla y Natividad Cruz, quienes han sabido formarme con buenos sentimientos, hábitos y valores, lo cual me ha permitido salir adelante en momentos difíciles y quienes también depositaron toda su confianza en mí.

A mi hermana que siempre ha estado junto a mí, brindándome su apoyo, muchas veces poniéndose en el papel de madre.

A mis abuelas, quienes siempre estuvieron al pendiente de mi andar escolar y mismas que en alguna ocasión me dijeron que la mejor herencia que me podrían dejar sería un título.

A mi tío Jesús, que siempre ha estado junto a mí, brindándome su apoyo, muchas veces poniéndose en el papel de padre.

A mis tíos Daniel y Guadalupe quienes han sido unos segundos padres para mí y a sus hijos Daniel y Marisol quienes me han brindado apoyo y consejo como si fueran mis hermanos.

A mi familia en general, porque me han dado su confianza y han compartido conmigo buenos y malos momentos.

En especial debo agradecer a mi asesor Miguel Morales, quien desde que me conoció en la clase de Ingeniería de Software depositó en mí su confianza y me apoyó en todo momento para culminar esta tesis.

A cada uno de mis sinodales quienes me apoyaron revisando este trabajo y sus comentarios permitieron obtener un mejor producto final.

A David Velázquez quien me dio todas las facilidades para llevar a cabo este proyecto y por ser no un jefe sino un amigo.

A René Villeda quien fue un gran apoyo durante mi estancia en la Facultad y de quien aprendí muchas cosas para el mundo laboral y algunas más para la vida.

A la UNAM que me permitió llevar a cabo mis estudios superiores y me permitió conocer y convivir con excelentes personas, incluso hasta me permitió ganar torneos de soccer en sus instalaciones.

A todos los amigos que conocí durante mi estancia en la Facultad, porque estuvieron en las buenas y en las malas, porque incluso parte de esta tesis es gracias a su apoyo, ¡ficha!

Y a cada una de las personas con las que me he cruzado durante mis diferentes etapas escolares, cada una influyó con sus lecciones y experiencias en formarme como una persona de bien y preparada para los retos que pone la vida, a todos y cada uno de ellos les dedico cada una de estas páginas.

*¿Sabes cuál es el problema?*

*Mujeres, alcohol... me ven a mí y creen que es fácil, pero no saben todos los años de trabajo duro y dedicación que me permiten ser como soy.*

*Charlie Harper*

---

# Índice general

<b>Agradecimientos</b>	<b>3</b>
<b>1. Introducción</b>	<b>11</b>
<b>2. Conceptos Básicos</b>	<b>13</b>
2.1. Ingeniería de Software . . . . .	14
2.2. Metodologías para el desarrollo de software . . . . .	16
2.3. Bases de datos . . . . .	22
<b>3. El Proceso Unificado</b>	<b>24</b>
3.1. Lenguaje de Modelado Unificado (UML) . . . . .	24
3.2. Casos de Uso . . . . .	26
3.3. Etapas del desarrollo del Proceso Unificado . . . . .	29
3.4. ISO/IEC 29110 y Proceso Unificado . . . . .	31
<b>4. Planteamiento del Problema</b>	<b>36</b>
4.1. Sidep v.1.0. . . . .	37
4.2. Análisis a Sidep v.1.0. . . . .	38
4.3. Conclusión del análisis a Sidep.v.1.0 . . . . .	42
<b>5. Planeación del Proyecto</b>	<b>44</b>
5.1. AP.1.1 Revisar el <i>Enunciado de trabajo</i> . . . . .	47
5.2. AP.1.2 Definir con el cliente las <i>Instrucciones de entrega</i> para cada uno de los entregables especificados en el <i>Enunciado de trabajo</i> . . . . .	47

---

5.3.	AP.1.3 Identificar las tareas específicas a realizar para producir los entregables y sus componentes. . . . .	51
5.4.	AP.1.4 Establecer la <i>Duración estimada</i> para realizar cada tarea. . . . .	60
5.5.	AP.1.5 Identificar y documentar los recursos requeridos para el desarrollo del proyecto. . . . .	64
5.6.	AP.1.6 Establecer la <i>Estructura del Equipo de Trabajo</i> . . . . .	67
5.7.	AP.1.7 Asignar la estimación inicial y las fechas determinadas para cada tarea. . . . .	68
5.8.	AP.1.8 Calcular y Documentar la <i>Estimación del Esfuerzo y Costo</i> . . . . .	72
5.9.	AP.1.9 Identificar y documentar los riesgos que pueden afectar al proyecto. . . . .	73
5.10.	AP.1.10 Documentar la <i>Estrategia de Control de Versiones</i> . . . . .	75
5.11.	AP.1.11 Generar el <i>Plan del Proyecto</i> integrando los elementos previamente identificados y documentados. . . . .	77
5.12.	AP.1.12 Incluir la descripción del producto. . . . .	77
5.13.	AP.1.13 Verificación del proyecto / AP.1.14 Revisar y obtener la aprobación del <i>Plan del Proyecto</i> . . . . .	79
5.14.	AP.1.15 Establecer el repositorio del proyecto usando la <i>Estrategia de Control de Versiones</i> . . . . .	80
<b>6.</b>	<b>Inicio de la Implementación de Software</b>	<b>82</b>
6.1.	I.S.1.1 Revisar el <i>Plan del Proyecto</i> / I.S.1.2 Establecer el ambiente de implementación. . . . .	83
<b>7.</b>	<b>Análisis de Requerimientos de Software.</b>	<b>87</b>
7.1.	I.S.2.1 Asignar tareas a los miembros del Equipo de Trabajo de acuerdo a cada rol, basadas en el <i>Plan del Proyecto</i> actual. . . . .	88
7.2.	I.S.2.2 Documentar o actualizar la <i>Especificación de Requerimientos</i> . . . . .	88
7.3.	I.S.2.3 Verificar la <i>Especificación de Requerimientos</i> y proponer una <i>Solicitud de Cambio</i> . . . . .	100
7.4.	I.S.2.4 Validar y obtener la aprobación de la <i>Especificación de Requerimientos</i> . . . . .	103

---

7.5.	I.S.2.5 Documentar la versión preliminar del <i>Manual de Usuario</i> o actualizar la versión existente. . . . .	103
7.6.	I.S.2.6 Verificar y obtener la aprobación del <i>Manual de Usuario</i> . . . . .	105
7.7.	I.S.2.7 Preliminar de Manual de Usuario verificado . . . . .	105
<b>8.</b>	<b>Arquitectura y Diseño Detallado del Software.</b>	<b>107</b>
8.1.	I.S.3.1. Asignar tareas a los miembros del Equipo de Trabajo de acuerdo a su rol para el <i>Plan del Proyecto Actual</i> . . . . .	108
8.2.	I.S.3.2. Comprender la Especificación de Requerimientos. . . . .	108
8.3.	I.S.3.3 Documentar o actualizar el <i>Diseño de Software</i> . . . . .	110
8.4.	I.S.3.4 Verificar y obtener la aprobación del <i>Diseño de Software</i> . . . . .	129
8.5.	I.S.3.5 Establecer los <i>Casos y Procedimientos de Prueba</i> para pruebas de integración basados en la <i>Especificación de Requerimientos y el Diseño de Software</i> . . . . .	129
8.6.	I.S.3.6 Verificar y obtener la aprobación de los <i>Casos y Procedimientos de Prueba</i> . . . . .	133
8.7.	I.S. 3.7 Actualizar el <i>Registro de Rastreo</i> / I.S.3.8 Incorporar el <i>Diseño de Software</i> . . . . .	133
<b>9.</b>	<b>Construcción del Software</b>	<b>134</b>
9.1.	I.S.4.1 Asignar tareas a los miembros del Equipo de Trabajo de acuerdo al <i>Plan del Proyecto Actual</i> . . . . .	134
9.2.	I.S.4.2 Entender el <i>Diseño de Software</i> . . . . .	135
9.3.	I.S.4.3 Construir los <i>Componentes de Software</i> basados en la parte detallada del <i>Diseño de Software</i> . . . . .	135
9.4.	I.S.4.4 Diseñar o actualizar los casos de pruebas unitarias y aplicarlos para verificar que los <i>Componentes de Software</i> implementan la parte detallada del <i>Diseño de Software</i> . . . . .	136
9.5.	I.S.4.5 Corregir los defectos encontrados hasta lograr una prueba exitosa.	137
9.6.	I.S.4.6 Actualizar el <i>Registro de Rastreo</i> incorporando <i>Componentes de Software</i> construidos o modificados. . . . .	137
9.7.	I.S.4.7 Incorporar <i>Componentes de Software y Registro de Rastreo</i> . . . . .	139

---

<b>10. Integración y Pruebas del Software</b>	<b>140</b>
10.1. I.S.5.1 Asignar tareas a los miembros del Equipo de Trabajo. . . . .	141
10.2. I.S.5.2 Entender los <i>Casos y Procedimientos de Prueba</i> . . . . .	141
10.3. I.S.5.3 Integrar el <i>Software</i> usando los <i>Casos y Procedimientos de Prueba</i> para las pruebas de integración. . . . .	142
10.4. I.S.5.4 Realizar pruebas de <i>Software</i> usando <i>Casos y Procedimientos de Prueba</i> . . . . .	143
10.5. I.S.5.5 Corregir los defectos encontrados. . . . .	146
10.6. I.S.5.6 Actualizar el <i>Registro de Rastreo</i> en caso de ser necesario. . . .	146
10.7. I.S.5.7 Documentar el <i>Manual de Operación</i> o actualizar el manual actual, en caso de ser apropiado. . . . .	146
10.8. I.S.5.8 Verificar y obtener la aprobación del <i>Manual de Operación</i> , en caso de ser necesario. Verificar la consistencia del <i>Manual de Operación</i> con el <i>Software</i> . . . . .	148
10.9. I.S.5.9 Documentar el <i>Manual de Usuario</i> o actualizar el actual. . . .	148
10.10. I.S.5.10 Verificar y obtener la aprobación del <i>Manual de Usuario</i> . . .	149
10.11. I.S.5.11 Incorporar los <i>Casos y Procedimientos de Prueba</i> , <i>Reporte de Pruebas</i> , <i>Manual de Operación</i> y <i>Manual de Usuario</i> como parte de la línea base. . . . .	149
<b>11. Entrega de Productos</b>	<b>150</b>
11.1. I.S.6.1 Asignar tareas a los miembros del equipo de trabajo. . . . .	150
11.2. I.S.6.2 Comprender la <i>Configuración de Software</i> . . . . .	151
11.3. I.S.6.3 Documentar el <i>Manual de Mantenimiento</i> . . . . .	151
11.4. I.S.6.4 Verificar y obtener la aprobación del <i>Manual de Mantenimiento</i> . 152	
11.5. I.S.6.5 Incorporar el <i>Manual de Mantenimiento</i> a la línea base de la <i>Configuración de Software</i> . . . . .	152
11.6. I.S.6.6 Llevar a cabo la entrega de acuerdo a las <i>Instrucciones de Entrega</i> . 152	
<b>12. Ejecución del Plan de Proyecto</b>	<b>156</b>
12.1. A.P.2.1 Monitorear la ejecución del Plan del Proyecto y registrar la información actual en el Reporte de Avance. . . . .	156

---

12.2. A.P.2.2	Analizar y evaluar el impacto en costo, tiempo e impacto técnico de la <i>Solicitud de Cambio</i> . . . . .	157
12.3. A.P.2.3	Conducir reuniones de revisión con el Equipo de Trabajo. . . . .	158
12.4. A.P.2.4	Realizar reuniones con el Cliente. . . . .	158
12.5. A.P.2.5	Realizar el <i>Respaldo del Repositorio del Proyecto</i> de acuerdo a la Estrategia de Control de Versiones. . . . .	160
12.6. A.P.2.6	Realizar la recuperación del <i>Repositorio del Proyecto</i> utilizando el <i>Respaldo del Repositorio del Proyecto</i> . . . . .	160
<b>13.</b>	<b>Evaluación y Control del Proyecto</b>	<b>162</b>
13.1. A.P.3.1	Evaluar el progreso del proyecto con respecto al Plan del Proyecto. . . . .	162
13.2. A.P.3.2	Establecer acciones para corregir desviaciones o problemas e identificar riesgos que amenacen el cumplimiento del plan. . . . .	163
13.3. A.P.3.3	Identificar cambios a requerimientos y/o al <i>Plan del Proyecto</i> . . . . .	164
<b>14.</b>	<b>Cierre del Proyecto</b>	<b>165</b>
14.1. A.P.4.1	Formalizar la conclusión del proyecto de acuerdo a las <i>Instrucciones de entrega</i> establecidas en el <i>Plan del Proyecto</i> . . . . .	165
14.2. A.P.4.2	Actualizar el <i>Repositorio del Proyecto</i> . . . . .	166
<b>15.</b>	<b>Resultados</b>	<b>167</b>
<b>16.</b>	<b>Conclusiones</b>	<b>170</b>
<b>A.</b>	<b>Administración del Proyecto</b>	<b>171</b>
<b>B.</b>	<b>Implementación del Software</b>	<b>178</b>
<b>C.</b>	<b>Informe del Estado de Configuración</b>	<b>191</b>
C.1.	Documento de Aceptación. . . . .	199
C.2.	Solicitud de Cambio. . . . .	199
C.3.	Registro de Defectos. . . . .	200
C.4.	Manual de Mantenimiento. . . . .	200
C.5.	Minuta. . . . .	201

---

C.6. Manual de Operación. . . . .	201
C.7. Reporte de Avance. . . . .	202
C.8. Plan del Proyecto. . . . .	203
C.9. Repositorio del Proyecto. . . . .	205
C.10.Respaldo del Repositorio del Proyecto. . . . .	216
C.11.Especificación de Requerimientos. . . . .	216
C.12.Software. . . . .	218
C.13.Componente de Software. . . . .	218
C.14.Configuración de Software. . . . .	218
C.15.Diseño de Software. . . . .	219
C.16.Manual de Usuario. . . . .	220
C.17.Enunciado de Trabajo. . . . .	221
C.18.Casos y Procedimientos de Prueba. . . . .	222
C.19.Reporte de Pruebas. . . . .	222
C.20.Rastreo. . . . .	223
C.21.Listas de Verificación. . . . .	224
C.22.Listas de Validación . . . . .	224
<b>Bibliografía</b>	<b>228</b>

---

# Capítulo 1

## Introducción

La División de Estudios de Posgrado de la Facultad de Medicina es el área encargada de administrar las Especialidades Médicas y Altas Especialidades Médicas que forman parte de su plan de estudios. Esta división depende de la Facultad de Medicina de la Universidad Nacional Autónoma de México.

Actualmente la División cuenta con un sistema de administración específico para los cursos de Especialidad y Alta Especialidad Médica. Este sistema es conocido como Sidep v.1.0 y ha sido utilizado por 4 años.

Sidep v.1.0 ha presentado diversos fallos desde su liberación, los más evidentes son los relacionados a la integridad y consistencia de datos. Estos fallos han causado distintos inconvenientes entre sus usuarios y administradores, tales como pérdida de información de profesores en sedes y cursos para un ciclo escolar específico, recaptura de información que el sistema no almacenó y la necesidad de manipular directamente la base de datos utilizando sentencias SQL.

Bajo este contexto, el alcance del presente trabajo es analizar los fallos que presenta Sidep v.1.0 y proponer una solución a éstos, ya sea a través de una actualización del sistema o bien mediante la creación de un sistema nuevo. El proceso para llevar a cabo este proyecto se guiará por metodologías ágiles y tradicionales, basándose principalmente en SCRUM y la Norma ISO/IEC 29110 para cada una de las metodologías correspondientes, generando así un proceso de desarrollo de software que involucre prácticas de ambos tipos de metodologías y permita generar una solución que atienda las necesidades presentes en la División.

---

Este trabajo se organiza de la siguiente manera, en el capítulo 2 se presentan los conceptos básicos en los que se fundamenta este proyecto. El capítulo 3 presenta el marco de trabajo utilizado como base para la realización de Sidep v.2.0, en este caso el Proceso Unificado. El planteamiento del problema se detalla en el capítulo 4. Posteriormente, se describen las actividades planteadas para dar solución al problema planteado. Dichas actividades se agruparon en dos procesos: el primero de ellos agrupa los capítulos 5, 12, 13 y 14 y presenta las actividades relativas a la administración del proyecto. Mientras que el segundo proceso, referente a la implementación del sistema, está compuesto por los capítulos 6, 7, 8, 9, 10 y 11. Finalmente el capítulo 15 presenta los resultados obtenidos y las conclusiones.

---

## Capítulo 2

# Conceptos Básicos

Los sistemas de consulta de información han cobrado gran importancia a lo largo de los años. El beneficio de automatizar el manejo de los datos para desde una pantalla realizar inserciones, actualizaciones, borrados y consultas se ha extendido a tal grado de que el día de hoy las empresas dependen de ellos para poder llevar a cabo su trabajo de manera eficiente.

A pesar de ello, la existencia de un sistema para el manejo de la información, no asegura que se tenga un manejo adecuado de ésta. Más aún, es recurrente el encontrarse con sistemas no documentados, lo que los vuelve poco mantenibles y les augura poca vida en operación. Además, se puede agregar a esta problemática el hecho de que los sistemas tengan problemas de diseño, consistencia e integridad de datos.

El fin de documentar un sistema, como lo veremos más adelante, es dejar una referencia de éste respondiendo a preguntas como:

- ¿Para qué se necesita?.
- ¿Cuáles son las funciones que están implementadas?.
- ¿Por qué se implementó de esa manera?.

El objetivo principal de documentar un sistema será crear una guía para entenderlo, poder corregir fallas y añadir módulos. Además sirve como una referencia para el futuro encargado del mantenimiento del sistema ya que le permitirá entender la forma en que trabaja el mismo.

Si bien es cierto que incluso un sistema bien documentado puede no ser correcto, será más simple llevarlo a este punto realizando las modificaciones pertinentes así como registrando cada uno de los cambios que se lleven a cabo.

Construir, operar, mantener y documentar un sistema de software son algunas de las actividades realizadas por una disciplina en particular, la Ingeniería de Software. En la siguiente sección se profundizará sobre esta disciplina.

## 2.1. Ingeniería de Software

Tomando en cuenta las siguientes definiciones de Ingeniería de Software:

- La ingeniería de software es la rama de la ingeniería que aplica principios de las ciencias de la computación y matemáticas para lograr soluciones rentables a los problemas de software [9].
- Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software [12].
- La Ingeniería de Software es la disciplina de la ingeniería que estudia la naturaleza, los enfoques y metodologías de desarrollo a gran escala del software. Además de las teorías y leyes detrás de las prácticas ingenieriles y el comportamiento del software [14].

Podemos construir nuestra propia definición:

- Llamaremos Ingeniería de Software a aquel proceso que involucra métodos, herramientas y técnicas que nos guiarán en el proceso de desarrollo de software y que tiene como fin la construcción de software correcto.

De manera genérica, todo proceso de Ingeniería de Software involucra las siguientes etapas de desarrollo:

- Análisis de requerimientos.

La captura, análisis y especificación de requerimientos es la primer etapa para la creación de software, se conocen los requerimientos que debe cumplir el sistema a través de la retroalimentación de parte del cliente y el conocimiento de las reglas de negocio.

- **Diseño.**

Tras pasar el análisis, en esta etapa se proponen los módulos a desarrollar y se deberán detallar cada uno de ellos. Es en esta fase cuando se definen aspectos como la arquitectura del sistema, la forma en que se desarrollarán las aplicaciones, el diseño de la base de datos y aquellas herramientas que se utilizarán para el desarrollo, pruebas y liberación.

- **Construcción.**

Se resume esta etapa en una palabra: codificación, se procede a codificar cada uno de los módulos planteados en la etapa de diseño, respetando estándares de codificación y de nombrado. Dentro de esta etapa, el equipo de desarrollo deberá probar cada uno de los componentes que conforman el sistema, realizando pruebas de caja negra y caja blanca.

- **Integración.**

En esta fase se unen los módulos creados en la etapa de construcción de acuerdo a las especificaciones obtenidas en la etapa de diseño, en cada integración de módulos se comienza en parte con la etapa de pruebas, ya que cada integración conlleva la necesidad de comprobar que en conjunto los módulos funcionan correctamente.

- **Pruebas.**

Las pruebas pueden dividirse en dos partes, primeramente el equipo de desarrollo deberá verificar que las funcionalidades implementadas trabajan de manera correcta. Posteriormente, al haberse asegurado del buen funcionamiento del sistema, se deberán realizar pruebas con usuarios finales del sistema para validar que éste cumple con los requerimientos que inicialmente se establecieron. Durante esta fase se recabarán sus comentarios sobre lo que se debe mejorar y las fallas que se detectaron en el uso del sistema.

- **Liberación.**

Al concluir la fase de pruebas, el sistema se encuentra listo para su uso. Es por ello que se entrega junto con la documentación al cliente, misma que servirá de referencia para los usuarios y el equipo encargado de su administración, operación y mantenimiento.

## 2.2. Metodologías para el desarrollo de software

¿Qué es una metodología para el desarrollo de software? Una metodología de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información [10]. Existen varias metodologías para el desarrollo de software, entre ellas podemos destacar:

- Proceso Unificado.

Es un proceso modular o basado en componentes, esto es, la construcción depende de módulos los cuales se encuentran conectados por interfaces bien definidas.

Guiado por casos de uso que son documentos que permiten capturar requerimientos funcionales y en los cuales profundizaremos en el capítulo 3, centrado en la arquitectura al considerar que afecta al funcionamiento del sistema por factores como son la plataforma, los componentes reutilizables y los requerimientos no funcionales [8] e interactivo e incremental que permite identificar y especificar los casos de uso a realizar para poder generar un producto funcional en cada entrega para ir anexándole funcionalidades en incrementos de manera gradual.

Es la metodología que se aplicará durante el desarrollo de este trabajo y se explicará a detalle en el capítulo 3.

- Personal Software Process.

En esta metodología, el programador deberá realizar las tareas definidas en documentos conocidos como scripts. Estos scripts no sólo definen las tareas a realizar sino que deben respetarse de manera disciplinada.

Los productos generados de estos scripts generarán una estadística, la cual el desarrollador de software tomará en cuenta para identificar fortalezas y debilidades de su desarrollo lo que le permitirá realizar un autoanálisis que más adelante lo llevará a realizar un proceso de autoaprendizaje [5].

- ISO/IEC 29110.

Es un estándar internacional que se ha desarrollado para entidades muy pequeñas (VSE - Very Small Entities) encargadas de construir software. Una VSE se define como una entidad que tiene menos de 25 personas. Esta norma ISO,

se divide en dos partes:

- ⊗ Implementación del Software.  
Describe conceptos de procesos, ciclo de vida y estandarización para la construcción del software.
- ⊗ Administración del proyecto.  
Describe el desarrollo de software de una sola aplicación por un solo equipo de proyecto sin riesgos o factores situacionales especiales [7].

- KANBAN.

El sistema Kanban para el software, derivado del Sistema de Producción de Toyota (TPS), en un enfoque sin iteraciones para organizar el trabajo. En vez de usar iteraciones de tiempo fijo y reuniones de planificación, el equipo de desarrollo toma tareas sólo cuando completó su trabajo anterior.

Ligado al uso de un planeador estructurado por columnas, identifica cada uno de los estados por los que una tarea puede pasar:

- ⊗ Solicitada.
- ⊗ Asignada.
- ⊗ En desarrollo.
- ⊗ En pruebas.
- ⊗ Completada.

Además se incluye la medición del tiempo invertido en cada tarea de la siguiente manera:

- ⊗ Fecha y hora de inicio.
- ⊗ Fecha y hora de fin.
- ⊗ Tiempo real invertido.

Un aspecto fundamental de esta metodología es mantener acotada una tarea bajo los siguientes dos conceptos:

- ⊗ No podemos comenzar una nueva tarea hasta que se haya terminado la anterior.

- ⊗ El número máximo de trabajo en curso tiene un límite, que es nuestra capacidad por ciclo o iteración [1].

- SCRUM.

Proceso iterativo e incremental, definido por un conjunto de prácticas y roles que sirven como base para la planeación del desarrollo del proyecto.

Los roles principales son:

- ⊗ *Product Owner*.

Representa la voz del cliente. Se asegura de que el equipo Scrum trabaje de forma adecuada desde la perspectiva del negocio.

- ⊗ *Scrum Master*.

Su trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El *ScrumMaster* no es el líder del equipo (ya que es auto-organizado), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga.

- ⊗ *Stakeholders*.

Será la gente que hace posible el proyecto y para quienes el proyecto producirá el beneficio acordado que justifica su producción. Sólo participan directamente durante las revisiones del sprint.

- ⊗ *Team*.

Tiene la responsabilidad de entregar el producto, integrado por 3 a 9 personas con las habilidades para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación, etc).

La manera de trabajar en esta metodología es mediante periodos de tiempo conocidos como *Sprints*, tras cada uno de estos periodos el equipo de desarrollo entrega un producto funcional. Las características que debe cumplir el entregable se encuentran detalladas en un documento nombrado: *Product Backlog*, dicho documento es elaborado durante la reunión de *Sprint Planning* en la cual el *Product Owner* identifica todos los elementos del *Product Backlog* y se lo extiende a los miembros del equipo, además existen reuniones diarias *Daily Scrum* o *Stand-up meeting* en las cuales los integrantes del equipo de desarrollo

hacen del conocimiento del *Scrum Master* los problemas encontrados o bien los avances que se van dando en el desarrollo del proyecto, se deben responder 3 preguntas específicas:

- ⊗ ¿Qué has hecho desde ayer?.
- ⊗ ¿Qué es lo que harás hasta la reunión de mañana?.
- ⊗ ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo?.

Otra situaciones a considerar para cada una de las *Daily Scrum* o *Stand-up meeting* son:

- ⊗ La reunión comienza puntualmente. Se acuerdan multas para quien llegue tarde.
- ⊗ Duración de la reunión de sólo 15 minutos.
- ⊗ **La reunión debe ocurrir en la misma ubicación y a la misma hora todos los días.**

Además del Daily Scrum, esta metodología define algunas otras reuniones con distintos objetivos. A continuación se describen brevemente:

- ⊗ Scrum de Scrums.

Todos los días tras la *Daily Scrum* o *Stand-up meeting* se reúnen una persona por equipo de trabajo, la reunión se hace en base a las mismas preguntas que los *Daily Scrum* o *Stand-up meeting* además de las siguientes 4 preguntas:

- ◇ ¿Qué ha hecho tu equipo desde nuestra última reunión?.
- ◇ ¿Qué hará tu equipo antes que nos volvamos a reunir?.
- ◇ ¿Hay algo que demora o interfiere a tu equipo?.
- ◇ ¿Estás a punto de poner algo en el camino del otro equipo?.

- ⊗ Sprint Planning Meeting.

Se realizan cada cierto periodo de tiempo definido al inicio del proyecto, normalmente 2 o 4 semanas, se definen las tareas por hacer identificando que tareas son posibles de realizar durante la duración del sprint, se detalla

con el equipo responsable de la tarea el *Sprint Backlog* en donde se indica el tiempo que tomará realizar la tarea.

⊗ Sprint Review Meeting.

Reunión en donde se revisan las tareas completadas y las no completadas, se presentan los avances funcionales a los *Stakeholders*.

⊗ Sprint Retrospective. Los integrantes del equipo de desarrollo realizan una sesión de intercambio de impresiones sobre el sprint que ha pasado, la idea es aprender y mejorar en el proceso de desarrollo [11].

En las metodologías ágiles, como SCRUM, no se manejan Casos de Uso, el detalle de las funcionalidades del sistema se hace mediante Historias de Usuario. De acuerdo con Mike Cohn [2] las Historias de Usuario tienen las siguientes características:

⊗ Independencia.

Las Historias de Usuario deben ser independientes, esto es, deben detallar una sola funcionalidad del sistema.

⊗ Negociables.

Las Historias de Usuario son detalles de funcionalidades explicadas en palabras del Usuario, dado que se detallan en lenguaje común no se pueden considerar tan importantes como un contrato, por lo mismo el Equipo de Desarrollo debe sostener discusiones con el Usuario con el fin de establecer el alcance de la función y se debe dejar de manera explícita para la posterior prueba de validación de la función por parte del Usuario.

⊗ Valor del Cliente y Usuario.

Los intereses tanto del Cliente como del Usuario tienden a ser siempre diferentes, por lo cual la Historia de Usuario debe ser importante para ambos más que para el Equipo de Desarrollo.

⊗ Estimables.

Tras la definición de la Historia de Usuario por parte de los Clientes y Usuarios ésta debe ponerse a discusión con el Equipo de Desarrollo para poder despejar dudas sobre los requerimientos funcionales lo que permitirá estimar el tiempo total de desarrollo.

⊗ Cortas.

Las Historias de Usuario grandes o que abarcan muchas funcionalidades son difíciles de estimar, por lo mismo se pide que cada función del sistema quede detallada siempre en una Historia de Usuario, así se permite un desarrollo iterativo-incremental que es más fácil de estimar.

⊗ Verificables.

Dado que cada Historia de Usuario representa una funcionalidad del sistema generalmente implica que ésta se puede verificar en cada entrega del sistema, la forma de verificación o validación por parte del usuario se establece durante las discusiones en que se esclarecen dudas y se estiman costos.

El manejar la especificación de requerimientos mediante Historias de Usuario colleva las siguientes ventajas:

- ⊗ Al pedir que las realicen los Usuarios, el detalle se describe en lengua casual por lo que permiten un mejor entendimiento de las funcionalidades requeridas.
- ⊗ Al ser cortas y representar en la mayoría de los casos una sola funcionalidad, la implementación se lleva a cabo de manera rápida (días o semanas).
- ⊗ Permiten involucrar al Cliente en el desarrollo, manteniendo una relación cercana con éste.
- ⊗ Se puede estimar el esfuerzo requerido de manera simple.

Como principales desventajas, podemos mencionar las siguientes:

- ⊗ Al no tener Pruebas de Validación de tipo Caja Negra y Caja Blanca, se podrían fugar errores en la programación y funcionalidad del sistema.
- ⊗ Se requiere contacto directo e involucramiento del Cliente y Usuarios en el desarrollo del proyecto, lo que puede resultar muy complicado.
- ⊗ Al ser las Historias de Usuario escritas por los Usuarios, se debe ser muy cuidadoso con los alcances y estimaciones de esfuerzo del proyecto, ya que para poder utilizarlas como base de un contrato éstas pueden quedar abiertas a distintas interpretaciones.

## 2.3. Bases de datos

El día de hoy las bases de datos se han vuelto una herramienta muy importante, no sólo por el hecho de permitir el almacenamiento de grandes cantidades de datos, sino también por que facilitan la obtención de información para la toma de decisiones. Diremos que una base de datos es un conjunto de datos los cuales mantienen una relación entre si al pertenecer a un mismo contexto, estos datos son administrados a través de un Sistema Manejador de Bases de Datos (DBMS por sus siglas en inglés). Al ser ésta el “almacén” de los datos del sistema, se convierte en el cimiento del mismo. Es por ello que se debe contar con un buen diseño de la misma para que sobre ésta se construyan los módulos del software.

- Existen varios tipos de bases de datos, entre los que podemos destacar:

- ⊗ Estáticas.

Son bases de datos que se utilizan sólo para lectura, conocidas como “Data warehouse” o cubos de datos, su función principal es ser la fuente de datos para la generación de reportes, toma de decisiones, análisis empresariales y minería de datos (extracción de conocimiento) [13].

- ⊗ Dinámicas.

Son la mayoría de las bases que existen en la actualidad, permiten la modificación de datos mediante inserciones, actualizaciones y borrados además de consulta de datos [13].

- Modelos de bases de datos Entre los modelos de bases de datos podemos destacar:

- ⊗ Bases de datos jerárquicas.

La estructura de estas bases de datos consiste de un nodo padre el cual puede tener varios hijos. Optimizadas por índices, este modelo es eficiente para el manejo de grandes cantidades de datos, en su contra está la limitación para eficientar la redundancia de datos [13].

- ⊗ Bases de datos transaccionales.

Permiten la consulta y modificación de datos a gran velocidad, al ser

transaccionales sus operaciones son atómicas; para comprender más este concepto el ejemplo común es realizar un traspaso de la cuenta A a la cuenta B, se realizan dos operaciones una que decrementará el saldo en A y otra que incrementará el saldo en B, de esta manera o bien se llevan a cabo ambas operaciones o no se lleva a cabo ninguna [13].

⊗ Bases de datos relacionales.

Los datos son almacenados en entidades las cuales se encuentran relacionadas, las entidades se encuentran compuestas de atributos y almacenan tuplas las cuales contienen los datos que estamos almacenando. Para manipular la información se cuenta con un lenguaje formal llamado álgebra relacional que sienta las bases para construir las consultas las cuales se realizan en lenguaje estructurado de consultas (SQL por sus siglas en inglés). Durante el diseño de la base en este modelo se pasa por varias fases, como son: creación de diagramas entidad-relación, normalización de relaciones y creación de diagramas de clases UML [13].

Para este proyecto se ha elegido utilizar una base de datos relacional debido a la necesidad de representar entidades con múltiples características. Estas entidades se encuentran relacionadas entre sí, además de la facilidad para modelar los supuestos que se obtienen tras el análisis de la lógica que deberá llevar el sistema.

---

# Capítulo 3

## El Proceso Unificado

Esta metodología se encuentra dirigida por casos de uso, es iterativa e incremental. Denominado también Proceso Unificado de Rational (RUP por sus siglas en inglés), desarrollado por la empresa “Rational Software” propiedad de IBM [6] y siendo “El proceso unificado” es la versión genérica y de dominio público de la metodología. Es un proceso basado en componentes, esto es, el software que se encuentra en construcción está formado por componentes los cuales se encuentran interconectados por interfaces bien definidas, utiliza lenguaje unificado de modelado (UML por sus siglas en inglés) para detallar todos los esquemas del software a desarrollar.

### 3.1. Lenguaje de Modelado Unificado (UML)

Es un lenguaje de modelado de sistemas orientado a objetos que permite especificar, construir, visualizar y documentar los módulos de un software, creado por Grady Booch y Jim Rumbaugh en 1995 [8]. Se centra en la descripción de métodos y procesos con lo cual permite definir de igual manera las interacciones que se llevan a cabo de parte de los programadores y los usuarios con el software que se está desarrollando, esto se hace a través de diferentes tipos de diagramas:

- De Estructura.

- ⊗ Digrama de clases.

- Representan elementos que son estáticos como lo son clases(entidades),

las cuales representan un conjunto de objetos que mantienen una estructura, comportamiento y relaciones con mismas características; sus contenidos (atributos), los cuales representan las propiedades de un objeto perteneciente a una clase y las relaciones las cuales se obtienen de la interacción que existe entre clases.

⊗ Diagrama de objetos.

Se muestran instancias específicas de clases (objetos) en un momento particular del sistema. Los diagramas de objetos utilizan un subconjunto de los elementos de un diagrama de clase.

⊗ Diagrama de componentes.

Representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre ellos.

⊗ Diagrama de paquetes.

Muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones.

• De Comportamiento.

⊗ Diagrama de estado.

Representan la secuencia de estados por los que un objeto o interacción pasa durante su ciclo de vida identificando las transiciones que se llevan a cabo para pasar de un estado a otro.

⊗ Casos de uso.

Ilustran los requerimientos del sistema, muestran la manera en que reacciona el sistema a eventos que ocurren en el mismo, la relación que existe entre caso de uso y actor pueden ser:

- ◇ Un actor se comunica con un caso de uso.
- ◇ Un caso de uso extiende a otro caso de uso.
- ◇ Un caso de uso utiliza otro caso de uso.

• De Interacción.

⊗ Diagrama de secuencia.

Representa el período durante el que un objeto está desarrollando una

acción directamente o bien a través de un método o función, estos objetos pueden ser creados durante la interacción o bien destruidos en la misma.

⊗ Diagrama de colaboración.

Muestra interacciones organizadas alrededor de los roles.

## 3.2. Casos de Uso

Como ya hemos mencionado, los casos de uso regirán la manera en que se desarrollará el software, por esto se vuelven muy importantes para un correcto desarrollo en busca de un mejor producto. En ellos se especifican los requisitos, condiciones y características funcionales que debe cubrir el sistema. Se deberá crear un formato estándar bajo el cual se harán los casos de uso, éste básicamente deberá estar compuesto por:

- Un identificador único para cada caso de uso.
- Un nombre representativo a la acción que describirá el documento.
- Un fragmento de la funcionalidad del caso de uso descrito.
- Especificar los actores que pueden llevar a cabo dicho caso de uso.
- Se deberán representar los requisitos funcionales del fragmento seleccionado a manera de precondiciones así como las dependencias de otros casos de uso en caso de existir.
- Todos los casos de uso juntos describen la funcionalidad completa del sistema.

Tras esto podemos decir que los casos de uso no sólo especificarán los requisitos detallados del sistema, sino que estos nos guiarán en el proceso de desarrollo de software. Un modelo de casos de uso estará compuesto por: actores, casos de uso, instancias de caso de uso y relaciones entre estos.

- Actor.

Será todo aquel ente que interaccione con el sistema (usuarios, clientes, administradores, otros sistemas, dispositivos), cada uno de ellos será representado como un actor; un actor será entonces un tercero externo al sistema que interactúa con él.



Figura 3.1: Representación de un actor en UML.

- Narrativa.

Cada característica o funcionalidad que los usuarios requieran del sistema será representada como un caso de uso. Un caso de uso detalla acciones que el sistema llevará a cabo tras la interacción con los actores. Estas interacciones incluyen flujos alternos, los cuales pueden estar descritos en el mismo caso o bien plasmarse en otro caso de uso, la forma de representar una acción/caso de uso en UML se muestra en la figura 3.2.

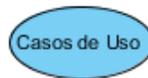


Figura 3.2: Representación de un caso de uso en UML.

- Instancia de un caso de uso.

Será el caso de uso en ejecución, será entonces la interacción que se lleva a cabo entre actor-caso de uso. Claro está que la interacción puede ser no sólo de un actor sino de varios con un caso de uso, se deberá detallar la secuencia de acciones que se llevan a cabo en la interacción. Los pasos que se deberán detallar son los siguientes:

- ⊗ La instancia de caso de uso inicia.
- ⊗ El caso es invocado por un mensaje de un actor.
- ⊗ Se transita entre estados dependiendo del mensaje emitido o la acción llevada a cabo por el actor, durante esta transición se deberá tener en

cuenta los mensajes a desplegar al actor así como las posibles operaciones internas que implica la transición y las excepciones que puedan ocurrir.

- Se llega a un estado final el cual implica un cambio de caso de uso o se llega a un estado de espera el cual podrá verse afectado por una acción del actor sobre el sistema.

Un ejemplo de formato para un caso de uso instanciado se podrá ver en la figura 3.3.



UNIVERSIDAD NACIONAL  
AV. PÍEZSA DE MÉXICO

SIDEP v.2.0



Facultad de Medicina

---

Fecha:

Detallado de Casos de Uso

**ID:** identificador de caso de uso  
**Caso de Uso:** nombre del caso de uso  
**Actor:** se especifica el o los actores que pueden interactuar en el caso de uso  
**Descripción:** pequeña descripción, se genera una idea o acción principal que explica la función del caso de uso  
**Precondiciones:**

- Se enumeran las precondiciones necesarias para que el caso de uso se pueda llevar a cabo
- Precondición 1
- Precondición 2
- 

**Id + caso de uso**



Actor	Sistema	Excepciones
Acción del actor sobre el sistema	Acción del sistema cambio de estado, respuesta del sistema, procesamiento de datos, realización de operaciones.	Identificador de excepción

**Excepciones.**

ID	Nombre	Acción
Identificador de excepción	Nombre de la excepción	Como se manejará la excepción

Figura 3.3: Ejemplo de una plantilla para instancia de caso de uso.

- Relaciones.

Las relaciones estarán dadas entre actores y casos de uso, esto es, representarán la interacción de los actores con el sistema o bien la dependencia de casos de uso que extienden a otros casos de uso.

### 3.3. Etapas del desarrollo del Proceso Unificado

El desarrollo de software bajo el Proceso Unificado involucra las siguientes etapas:

- Planteamiento del problema.

Es la etapa inicial del proceso de construcción de software. En donde se deberá definir de manera clara el problema, resolviendo toda duda que se llegue a generar entre el equipo de desarrollo y el cliente. Para esta etapa deberán tenerse varias reuniones con los clientes, preguntando sobre lo que necesitan y desean que haga el sistema. Se deberá poner atención incluso a como le gustaría al cliente que funcione y luzca el sistema.

- Análisis.

Tras recabar la información necesaria con el cliente y/o usuario, se deberá proceder a plantear entre el equipo de desarrollo todos los “supuestos” obtenidos en la fase anterior. Se analiza toda la lógica que deberá implementar el sistema, así como la manera en que se resolverá el problema planteado.

De esta fase depende la funcionalidad del proyecto así como la aprobación del sistema final, se realiza una propuesta inicial de interfaz de usuario la cual deberá ser presentada al cliente/usuario para su aprobación.

- Diseño.

Se realiza el diseño de la base de datos “los cimientos del sistema”, si bien lo importante es el producto final completo, justo aquí podemos resolver muchos de los problemas a los que nos enfrentamos en el mantenimiento del sistemas si aplicamos una correcta abstracción de entidades, atributos y relaciones, además de una buena normalización de relaciones.

Así pues el diseño de la base de datos se vuelve parte fundamental para el buen funcionamiento del sistema así como para la creación de un correcto

software. En esta etapa entran los casos de uso, creando casos de uso para ubicar dependencias e identificar la manera en que se deberán desarrollar los diferentes módulos diseñados durante esta etapa.

- Codificación.

Consiste en preparar el ambiente de implementación y comenzar con la codificación de los componentes. Todo esto deberá seguir lo definido en la fase de diseño.

- Pruebas.

En esta etapa se consideran 2 tipos de pruebas:

- Pruebas de integración.

Son pruebas que se realizan para cada componente al integrarlo con algún otro componente creado.

- Pruebas de sistema.

Son pruebas para verificar que el sistema funciona correctamente como un sólo módulo. En estas pruebas es deseable incluir las pruebas con usuarios finales del sistema para su aceptación.

- Liberación.

En algunos casos representa la última fase de desarrollo, se entrega el sistema al cliente una vez que se ha detectado que éste funciona correctamente. Para ello se aconseja planear un proceso de transición, el cual involucre la adaptación de los clientes/usuarios al nuevo sistema.

- Mantenimiento.

En caso de requerirlo, el cliente solicitará que tras la liberación del producto el mantenimiento al sistema, esto incluye, actualizaciones, creación de nuevos módulos y administración general del sistema.

Por esta razón cobra gran importancia la documentación creada durante el desarrollo, ya que para esta etapa se asignará sólo a parte del equipo de desarrollo o en otros casos a personal que no tiene conocimiento alguno del sistema. Por lo que deberán conocer el sistema completo aún cuando ellos sólo hayan desarrollado ciertos módulos.

### 3.4. ISO/IEC 29110 y Proceso Unificado

Para este proyecto se ha decidido utilizar la norma ISO/IEC 29110 basándonos en la metodología del Proceso Unificado y en su estructura de documentación. De acuerdo con la norma ISO/IEC 29110 se definen dos procesos: el de Administración del Proyecto y el de Implementación de Software, ver figura 3.4.

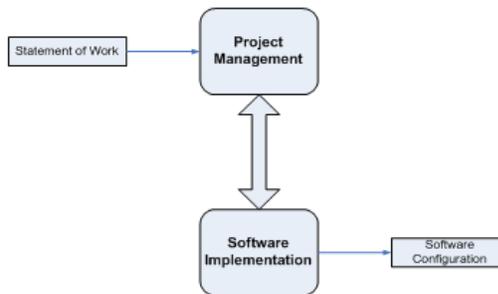


Figura 3.4: Procesos guía del perfil básico

- El Proceso de Administración del Proyecto (AP).

Tiene como propósito establecer y llevar a cabo de manera sistemática las tareas de un proyecto de implementación de software, que permitan cumplir con los objetivos del proyecto en la calidad, tiempo y costos esperados. Está definido por las siguientes fases [7]:

- ⊗ Planeación de Proyecto.
- ⊗ Ejecución del Plan de Proyecto.
- ⊗ Evaluación y Control del Proyecto.
- ⊗ Cierre de proyecto.

Ver figura 3.5

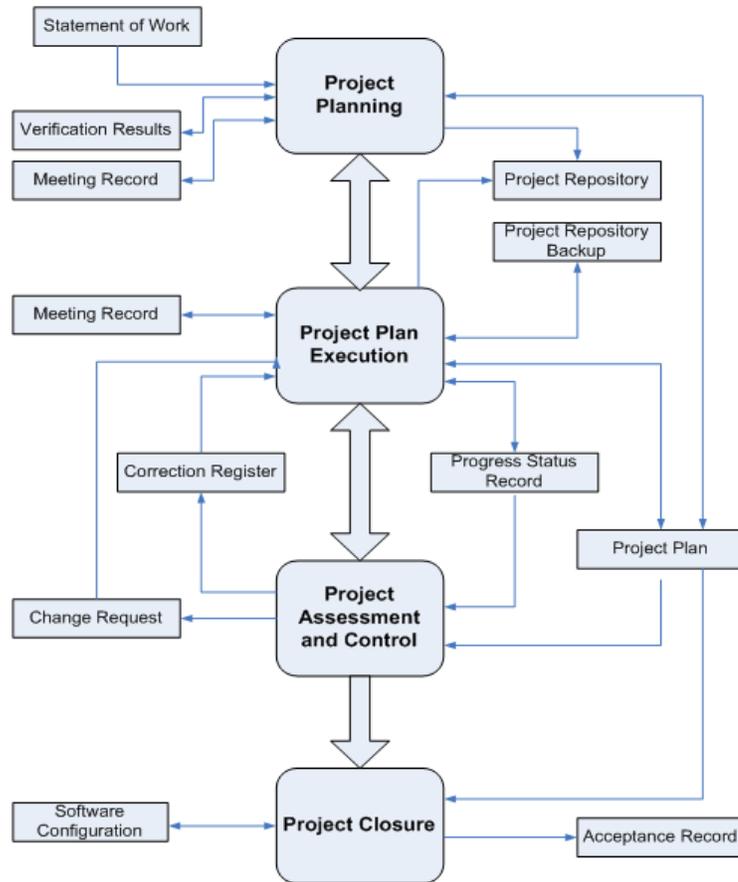


Figura 3.5: Diagrama del proceso de Administración del Proyecto

- El proceso de Implementación de Software (IS).

Tiene como propósito la realización sistemática de las actividades de análisis, diseño, construcción, de integración y pruebas para productos de software, nuevos o modificados, de acuerdo a los requerimientos especificados. Está definido por las siguientes etapas [7]:

- ⊗ Inicio de la Implementación de Software.
- ⊗ Análisis de Requerimientos de Software.
- ⊗ Arquitectura y Diseño de Detallado del Software.
- ⊗ Construcción del Software.

- ⊗ Integración y Pruebas de Software.
- ⊗ Entrega de Productos.

Ver figura 3.6

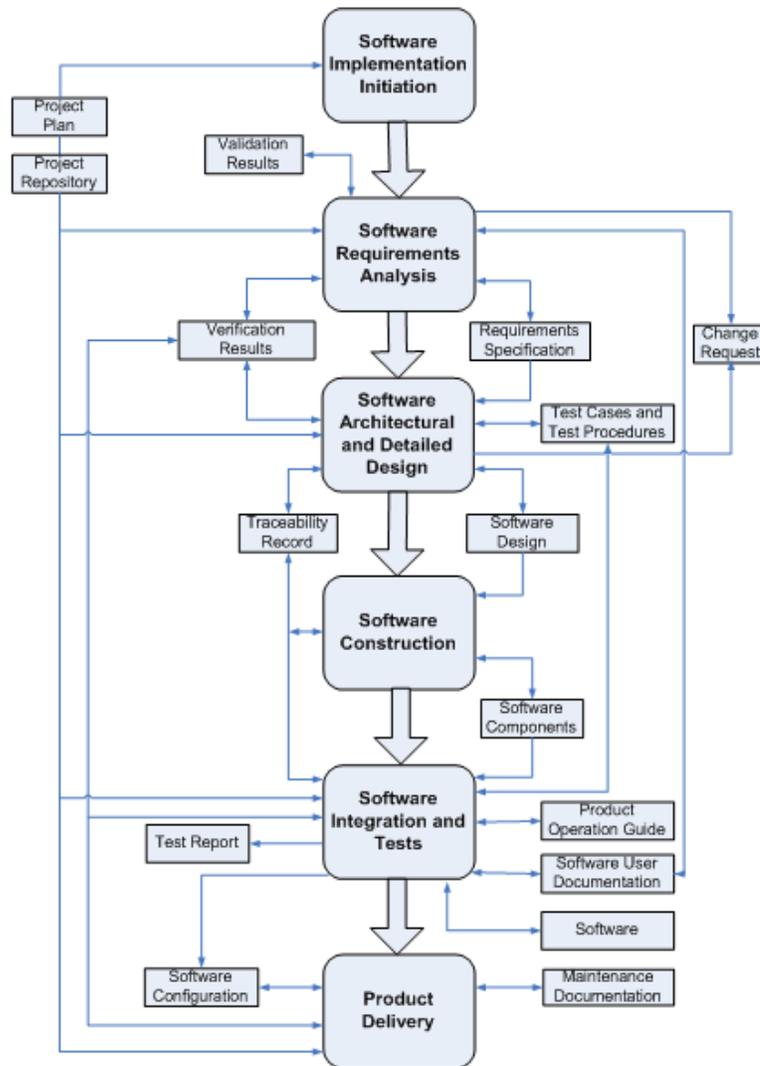


Figura 3.6: Diagrama del proceso de Implementación de Software.

- Manuales a entregar:

Un manual es el documento que contiene la descripción de actividades que se llevaron a cabo durante el desarrollo del proyecto, además incluye los roles que intervinieron en el desarrollo de las tareas. Para fines de esta tesis se generarán los siguientes manuales:

⊗ Manual de Administración del Proyecto.

Contiene de manera detallada la documentación que respalda la forma en cómo se creó el proyecto, desde definiciones de estándares hasta asignaciones de tiempo y tareas a los roles definidos. De acuerdo al Proceso Unificado, se construye en base a las etapas de desarrollo siguientes:

- ◇ Lanzamiento.
- ◇ Estrategia.
- ◇ Planeación.
- ◇ Seguimiento.
- ◇ Cierre (Lecciones aprendidas y sugerencias de mejora, Informe de mediciones)

⊗ Manual de Implementación de Software.

Especifica de manera detallada el cómo se construyeron los módulos de software, esto permitirá a gente ajena al equipo de trabajo mantener o ampliar el alcance del sistema. Tomando en cuenta lo definido por el Proceso Unificado, este manual se construye en base a las etapas de desarrollo siguientes:

- ◇ Especificación de requerimientos.
- ◇ Diseño.
- ◇ Construcción.
- ◇ Pruebas.
- ◇ Cierre

⊗ Manual de Usuario.

Permitirá conocer el funcionamiento del sistema para que los usuarios se puedan adaptar a él, se incluyen capturas de pantalla y descripciones de las actividades que deben seguirse para realizar funciones de los módulos presentados en el sistema. Puede generarse más de un manual de usuario,

dependerá de la cantidad de *Tipos de Usuario* que se detecten, creando un *Manual de Usuario* para cada uno de ellos.

Para este proyecto seguiremos la norma ISO/IEC 29110, apoyándonos en el Proceso Unificado para llevarlo a cabo, desarrollando las siguientes actividades en las fases de cada una de las dos etapas de la norma:

- Etapa AP.

Se analiza el problema planteado por el cliente y su lógica de negocio. Además se involucra la creación y asignación de roles para cada uno de los miembros del equipo de desarrollo, listas de verificación, documentación en minutas, casos de uso, diagramas de secuencia, diagramas de estado. Para la base de datos se incluyen diagramas entidad-relación y de clases; se crea un repositorio del proyecto accesible a todos los miembros del equipo de desarrollo y se establece un estándar para la creación de respaldos del repositorio del proyecto administrado mediante un control de versiones.

- Etapa IS.

Basados en el desarrollo iterativo e incremental, las tareas deberán completarse de manera correcta para poder continuar con el resto de ellas, en caso de presentar fallas, estas deberán corregirse para poder continuar con el resto de las tareas; documentación en minutas sobre cambios en los repositorios de acuerdo a los avances de codificación que se van efectuando; correcciones o modificaciones se harán sólo bajo la creación de documentos de solicitudes de cambio; reportes de avances de las tareas indicando estado(sin comenzar, en proceso, finalizada); juntas semanales para saber el avance de los miembros del equipo y verificar el avance general; pequeñas entregas de avances semanales durante las reuniones de equipo.

Ejecución de pruebas de caja blanca y caja negra por parte de los miembros del equipo de desarrollo y los clientes, todas ellas documentadas; autoevaluación y evaluación de los integrantes del equipo de desarrollo; creación de documentos de lecciones aprendidas durante el desarrollo para mejoras futuras.

Finalmente se configurará el software con el cliente liberándose una versión funcional del sistema.

---

## Capítulo 4

# Planteamiento del Problema

La División de Estudios de Posgrado de la Facultad de Medicina [3] es la dependencia de la Universidad Nacional Autónoma de México encargada de la administración de cursos de especialidad y alta especialidad médica. Estos cursos se imparten en diversas sedes a lo largo de la República Mexicana.

Para administrar cada uno de estos cursos, las sedes y los profesores que los imparten, la División cuenta con el sistema Sidep v.1.0 desde el año 2008. Dicho sistema ha presentado diferentes fallas desde su liberación, por esta razón, se ha decidido que el sistema sea analizado para saber si es funcional, si se debe modificar, actualizar o rehacer por completo.

Los supuestos bajo los que se construyó el sistema y debería manejar de manera correcta son:

- Despliegue de información de sedes en las que se imparte un curso en específico.
- Despliegue de información sobre los profesores(titulares y adjuntos) asignados a un curso en una sede específica.
- Despliegue de información de reportes como: totales de profesores, cursos y sedes, profesores que se encuentran en actividad durante un periodo escolar específico.
- Permitir operaciones sobre la información de profesores, cursos y sedes como inserciones, actualizaciones y borrados de registros.

Además, se tienen identificados 3 tipos de usuarios que tienen interacción con el sistema, uno de ellos “Jefe de Enseñanza” es externo a la división, según la importancia de los usuarios se dividen en:

- Administrador de sistemas de la División de Estudios de Posgrado de la Facultad de Medicina.

Es el encargado dentro de la División del desarrollo, administración y mantenimiento de los distintos sistemas y bases de datos con que cuenta la dependencia.

- Administrador de cursos de la División de Estudios de Posgrado de la Facultad de Medicina.

Es el encargado, junto con su personal de apoyo, dentro de la Unidad de Posgrado de realizar cambios en la información del sistema, es el único rol que tiene acceso a todas las funciones del sistema.

- Jefes de enseñanza.

Existe uno por sede, sólo podrán ser jefes de una unidad; podrán realizar cambios en la asignación de profesores a cursos que se imparten en su sede únicamente. Además informarán al Administrador de cursos de la División de Estudios de Posgrado de la Facultad de Medicina la baja de cursos o bien la baja de la sede del padrón.

- Usuarios de consulta.

Sólo podrán entrar a consultar datos, esto es, tendrán acceso a toda la parte de despliegue de consultas y reportes.

## 4.1. Sidep v.1.0.

Este sistema tiene 5 años funcionando en la División, se creó como solución al manejo de la información presentada anteriormente, dicho sistema está creado con PHP versión 5.2.0, con una base de datos creada en Postgresql versión 8.4 y se encuentra montado en un servidor Apache.

Durante los últimos 3 años ha presentado más errores de los tolerables, entre los principales problemas comentados por los usuarios e identificados por el administrador del sistema se encuentran:

- Inconsistencia de datos, desde “pérdida o desaparición” de registros hasta “creación” de registros basura.
- Problemas con la consulta de datos.
- Enlaces rotos a páginas de consulta o reportes.
- Cambios en los registros que debe hacer el administrador “a pie” ya que el sistema no lo permite.

Tras estas observaciones el Jefe del Departamento de Cómputo de la División solicitó hacer un análisis profundo del sistema Sidep v.1.0 para localizar las fallas y de acuerdo a éstas definir el futuro del sistema.

## 4.2. Análisis a Sidep v.1.0.

De acuerdo al informe entregado por los usuarios, la mayor parte de los errores se encuentran en problemas de inconsistencia de datos, además por parte del Jefe del Departamento de Cómputo se ha solicitado el cambio de interfaz por una que respete criterios ergonómicos de usabilidad y que les permita explotar de mejor manera los datos almacenados.

- Análisis de la base de datos Especialidades.
  - ⊗ La base de datos actual de Sidep se encuentra administrada mediante el DBMS Postgresql versión 8.4, dicha base consta de 63 tablas. Ver figura 4.1.
  - ⊗ El modelo de la base de datos pretendía ser una base de datos relacional, sin embargo, la base no tiene definidas llaves primarias ni foráneas por lo cual se viola por completo el principio del modelo relacional.
  - ⊗ Otro aspecto identificado es la existencia de tablas repetidas: Las tablas repetidas nos provocan redundancia de datos, además de confusión sobre cual es la tabla real en caso de que sólo una se vea afectada con las transacciones de la base, tenemos que la tabla *academicodatospersonale* y la tabla *academicodatospersonales* son la misma y están almacenando

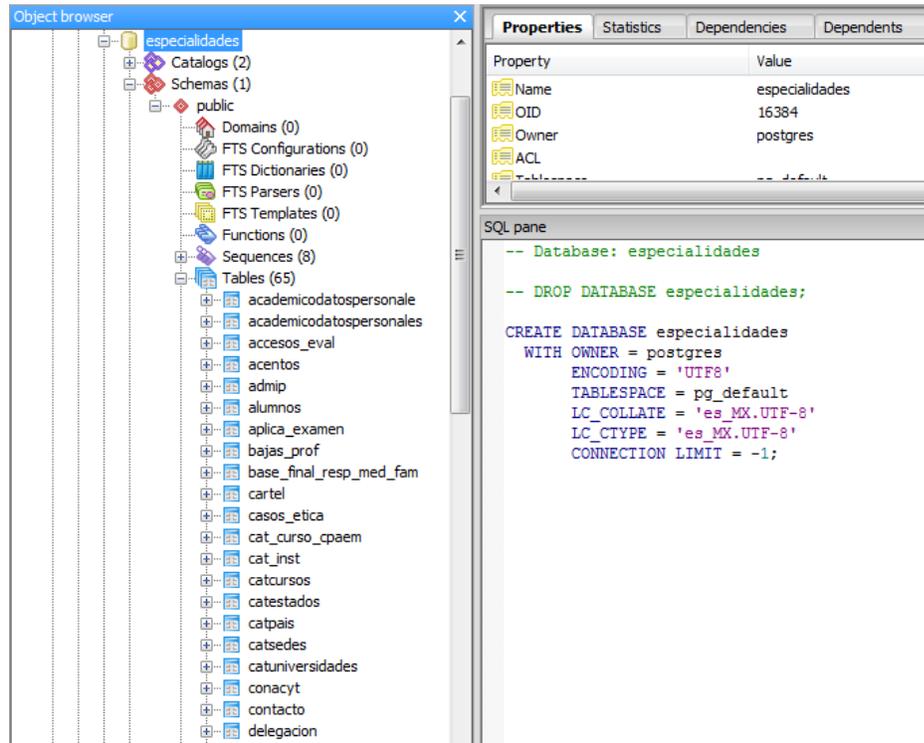


Figura 4.1: Estructura de la base de datos Especialidades del sistema Sidep v.1.0.

justo la misma información, lo mejor aquí sería sólo dejar una; otro caso es con la tabla *entidad* y la tabla *catestados*, la idea de ambas es almacenar la información de los estados de la República Mexicana por lo cual lo mejor sería eliminar alguna de las dos.

- ⊗ Tablas inútiles, es decir tablas vacías, incompletas o que contienen registros incongruentes:
  - ◇ *admip*: contiene solo la ip del administrador del sistema, dicha tabla esta compuesta de 5 atributos los cuales almacenan las diferentes partes de la ip.
  - ◇ *contacto*: tabla con 25 atributos, completamente vacía.
  - ◇ *delegacion*: tabla que intentó ser un catálogo de delegaciones, sin embargo nunca se utilizó, contiene solo 2 registros.
  - ◇ *entidad*: pretendía ser un catálogo de estados, es una tabla incompleta en cuanto a registros, sólo contiene 7 estados y es una tabla repetida

ya que la base ya cuenta con la tabla catestados que cumple la misma función.

- ◇ *ex\_3440\_2009\_sem*: tabla con 34 atributos, completamente vacía.
- ◇ *examen*: tabla con 33 atributos, completamente vacía.
- ◇ *file\_carpdf*: tabla que contiene registros basura, pretendía tener un atributo cartel que fungiría como id y uno nombre\_archivo que almacenaría la ruta de un archivo pdf, sin embargo se tiene solo la secuencia de los id's de carteles y no se tiene ningún registro válido con información en nombre\_archivo ya que todos los registros contienen la cadena “ ”.
- ◇ *impresion\_sede*: tabla con 2 atributos(sedes,sedes\_baja), registros basura, ya que existen registros nulos o bien incompletos, no existe registro alguno con ambos atributos ocupados, pretendía almacenar el id de la sede que se daba de baja y el motivo por el cual lo hacía.
- ◇ *impresion\_sede\_baja*: tabla con un atributo, sólo almacena el id de la sede.
- ◇ *jefesxsede*: pretendía asociar un jefe de enseñanza con la sede asignada sin embargo es una tabla que sólo contiene 2 registros, dichos registros asocian dos sedes al mismo jefe lo cual viola la lógica del negocio.
- ◇ *la\_agenda*: pretendía ser una agenda para contacto con usuarios al almacenar nombre, correo electrónico y teléfono fijo, sin embargo sólo contiene 2 registros de prueba.
- ◇ *prueba*: tabla con 2 atributos, completamente vacía.
- ◇ *t\_nopropuesta*: tabla con 4 atributos, pretendía mantener la relación entre curso, sede y un registro default que indicaría que alguno de los puestos de profesor se encuentra libre, completamente vacía.
- ◇ *t\_resumen*: tabla con 5 atributos, pretendía almacenar la información de trabajos de alumnos, sin embargo sólo existen registros con el atributo cuenta ocupado, este atributo fungiría como llave, el resto de los atributos se encuentran vacíos en todos los registros existentes.
- ◇ *tipo\_contacto*: tabla que pretendía ser un catálogo para definir un tipo de contacto en la tabla la\_agenda, sin embargo jamás se hace

referencia a esta tabla.

⊙ Tablas restantes.

La mayor parte de las tablas de la base de datos tienen errores de consistencia de datos ya que la mayoría de los atributos de éstas se encuentran o vacíos o con información inconsistente. Esto se debe a que la base de datos no tiene implementadas restricciones de integridad de datos o bien en la interfaz de usuario no se maneja ningún tipo de validación, un ejemplo de este problema lo tenemos en la figura 4.2.

Como ejemplos de atributos innecesarios tenemos:

- ◇ Mantener 4 atributos para correos electrónicos distintos, cuando la mayoría de los registros sólo cuentan con un correo electrónico.
- ◇ Manejo de 3 direcciones diferentes, cuando la mayoría de los registros no tienen dirección alguna registrada.
- ◇ Manejo de atributos nulos, esto es, ningún registro cuenta con algún dato en atributos como comite,direc.

nombres	grado	genero	fechanac	curp	rfc	calle_p	colonia_p	delegacio	municipio_p
character va									
GABRIELA	DR.	''	//	MEGG680907M	MEGG680	''	''	''	''
JORGE ROMÁN	DR.			AUSJ620824H	AUSJ620				
MARÍA DEL C	DRA.		//	CABC581006M	CABC581	CALLE 34 X	COL. INDU	''	MÉRIDA
ALFREDO	DR.	M	''	''	''	''	''	''	''
LUIS FRANCI	''	''	''	PIGL780323H	PIGL780	''	''	''	''
ARTURO	DR.	M	5/3/1968 00	AADA680305H	AADA680	''	''	''	''
FRANCISCO	DR.	M	25/3/1960 0	AOJF600325H	AOJF600	''	''	''	''
MIGUEL ÁNGE	DR.	M	10/9/1961 0	AAPM610910H	AAPM610	GUANABANA	HOGAR Y S	AZCAPOTZ	''
ARMIDA	DR.		//	GARA751209M	GARA751	AVE. SAN B	BELISARIO	''	DELEG. T
MARÍA DEL P	DRA.	F	''	CUDP690105M	CUDP690	''	''	''	''
JOSÉ ALEJAN	DR.	M	''	''	''	''	''	''	''
LUIS RODOLF	DR.	M	//	ROVL690513H	ROVL690	CARLOS LAZ	TACUBAYA	''	DELEG. M
ARTURO	''	''	''	OEA790701H	OEA790	''	''	''	''

Figura 4.2: Ejemplo de errores de integridad de datos

- Análisis a la interfaz de Sidep.v.1.0 (PHP)

La interfaz como tal no presenta errores graves pero tiene errores de ligas,

consultas incompletas y validación de datos, lo cual entorpece la realización de las tareas de los usuarios. Ver figura 4.3

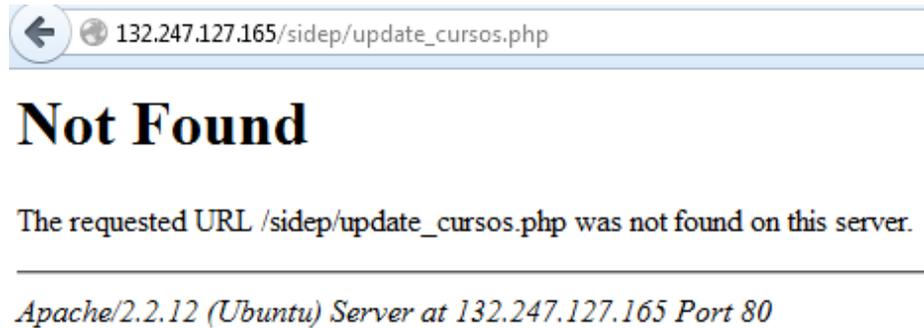


Figura 4.3: Ejemplo de enlace roto en Sidep.v.1.0.

### 4.3. Conclusión del análisis a Sidep.v.1.0

- Dada la cantidad de defectos graves encontrados durante la operación del sistema Sidep v.1.0., el Departamento de Cómputo decidió migrar el sistema actual de PHP a Java. Dicha migración se realizará basándose en la interfaz actual añadiendo las respectivas validaciones para evitar inconsistencias de datos y arreglando enlaces rotos, todo esto como un nuevo desarrollo apoyándonos de las herramientas de Java WEB.
- Del lado de la base de datos se concluye que se necesita una nueva base de datos, la cual sea relacional y se encuentre normalizada; se mantendrá la integridad de los datos con validaciones en la interfaz de usuario y con restricciones en la base de datos.

Además se necesitan optimizar las operaciones que se llevan a cabo en la base como lo son: inserciones, actualizaciones, recuperación de datos para asociar a

formularios y consultas, por lo cual se propone añadir procedimientos almacenados y vistas dentro del nuevo esquema de base de datos.

---

# Capítulo 5

## Planeación del Proyecto

Durante este capítulo se mostrará la manera en que se planifica el desarrollo de un proyecto de software, basándose como ya se ha comentado en la norma ISO/IEC 29110 y en las metodologías del Proceso Unificado y SCRUM.

De acuerdo a la norma ISO/IEC 29110 durante esta etapa se documentan los detalles de la planeación necesarios para llevar a cabo una adecuada administración del proyecto.

- La actividad de Planeación del Proyecto proveerá de:
  - ⊗ El **enunciado de trabajo** validado y las tareas necesarias para proveer los entregables acordados para satisfacer los requerimientos del cliente.
  - ⊗ El **ciclo de vida del proyecto**, incluyendo dependencias entre tareas y duración de las mismas.
  - ⊗ La **estrategia de aseguramiento de la calidad del proyecto**, a través de la verificación y validación de los productos/entregables, así como las revisiones del equipo de desarrollo y del cliente.
  - ⊗ Los **roles y responsabilidades** del equipo de desarrollo y del cliente.
  - ⊗ Los **recursos y necesidades de capacitación** para el proyecto.
  - ⊗ La **estimación** del esfuerzo, costo y tiempo.
  - ⊗ La **identificación de los riesgos** del proyecto.
  - ⊗ Una **estrategia para el control de versiones** y la línea base del proyecto.

- ⊗ Un **repositorio del proyecto** para almacenar, administrar y entregar de manera controlada los productos, versiones de documentos y líneas base citeISO.
- Los roles definidos para esta actividad son los siguientes:
  - ⊗ Cliente (CL).

Es aquella persona responsable de llevar a cabo el buen desempeño del proyecto, por parte de la empresa que contrata el desarrollo. Debe asumir los deberes de la empresa contratante ante el equipo de desarrollo. Deberá estar presente en todas las fases de desarrollo del producto y realizar sus actividades entre las que están las siguientes:

    - ◇ Servir de conexión entre la empresa solicitante del software y el equipo de desarrollo.
    - ◇ Conocer las etapas y roles en la construcción de software.
    - ◇ Definir y priorizar requerimientos.
    - ◇ Revisar y aprobar documentos.
    - ◇ Entregar los recursos necesarios para llevar a cabo el producto.
    - ◇ Participar en la elaboración del **Manual de Usuario**.
  - ⊗ Administrador de Proyecto (AP).

Es aquella persona que administra y controla los recursos asignados (humanos, económicos, tecnológicos, espacio físico, etc.) a un proyecto con el propósito de que se cumplan correctamente los planes definidos. El foco de una buena administración debe estar en el control y coordinación de los diferentes eventos y actividades de un proyecto. Los objetivos de un AP son:

    - ◇ Tener el producto “a tiempo”, “bajo presupuesto” y con los requerimientos de calidad definidos.
    - ◇ Terminar el proyecto con los recursos asignados logrando el mejor uso de estos.
    - ◇ Apoyar a los integrantes de su equipo de desarrollo a cumplir los objetivos trazados para así poder cumplir el objetivo general.
    - ◇ Cumplir con éxito cada una de las fases de un proyecto.

- ◇ Cumplir con las expectativas del cliente.
- ⊗ Líder Técnico (LT).  
Persona que cuenta con conocimiento y experiencia en el dominio del proceso de software, deberá elegirse con cuidado correspondiendo su conocimiento con la metodología elegida.
- ⊗ Equipo de Trabajo (ET).  
Conformado por personal especializado en su área, creación de nuevos roles que se profundizarán en la etapa de Implementación del Software como:
  - ◇ Analista (AN).
  - ◇ Diseñador (DIS).
  - ◇ Programador (PR).
- A partir de este capítulo se mostraran tablas de actividades tomadas de la Norma ISO/IEC 29110 [7] las cuales estarán coloreadas de acuerdo a la siguiente especificación:

- ⊗ Para las actividades desarrolladas de manera **ágil** se utilizará el color:



Figura 5.1: Color ágil.

- ⊗ Para las actividades desarrolladas de manera **tradicional** se utilizará el color:



Figura 5.2: Color tradicional.

- ⊗ Para las actividades desarrolladas de manera **híbrida** se utilizará el color:



Figura 5.3: Color híbrido.

La primer lista de tareas tomada de la norma ISO/IEC 29110 se presenta en las figuras A.2 y A.3 que se pueden encontrar en el Apéndice Administración de Proyecto.

En las siguientes secciones se desarrollarán cada una de las tareas requeridas por la *Planeación del Proyecto*. A su vez, en el Apéndice Administración del Proyecto, se incluyen los productos generados y que conformarán los distintos manuales solicitados por la norma. Para facilitar la identificación de los productos de trabajo en el texto, éstos serán escritos en itálicas.

### 5.1. AP.1.1 Revisar el *Enunciado de trabajo*.

Se tomó como *producto de entrada* el capítulo 3 el cual muestra el problema planteado desde el punto de vista del cliente. Por otra parte, el **producto de salida** que tendremos será el enunciado de trabajo [revisado] por el equipo de desarrollo, lo que implica mostrar en palabras del mismo equipo el enunciado de trabajo:

*La División de Estudios de Posgrado de la Facultad de Medicina, perteneciente a la Universidad Nacional Autónoma de México, a través de su Departamento de Cómputo y Sistemas ha solicitado la creación de un nuevo sistema para la administración de cursos, sedes y profesores de especialidades y altas especialidades que esta dependencia administra. La solución deberá basarse en el sistema con que se cuenta actualmente (Sidep.v.1.0) y deberá concentrarse en la corrección de los errores que éste presenta.*

### 5.2. AP.1.2 Definir con el cliente las *Instrucciones de entrega* para cada uno de los entregables especificados en el *Enunciado de trabajo*

Para poder llevar a cabo la tarea del proyecto es necesario definir un estándar para nombrar cada uno de los productos que se generen durante el proceso. Se define el estándar para nombrado y documentación:

- Estándar de nombrado.

Todo documento generado durante el proyecto será nombrado de acuerdo a la especificación de nombrado de la norma ISO/IEC 29110, para los documentos creados en base al Proceso Unificado se les nombrará de acuerdo a la especificación de dicha metodología de la siguiente manera:

*nombre\_version.tipo*

El estándar para la asignación de versiones se detalla en el siguiente punto; los tipos se asignarán de la siguiente forma:

- ⊗ .docx para versiones entregables pero que **NO** son finales.
- ⊗ .pdf para versiones entregables finales construido apartir del entregable homónimo pero de tipo .docx.

- Definición del control de versiones.

Dado que el proyecto será dividido en iteraciones o sprints de una semana, se tomará cada fin de sprint como el punto en el tiempo para establecer la versión de cada producto de trabajo.

- ⊗ Sólo se asignará un control de versión a un entregable, esto es, una tarea completada.
- ⊗ El primer número de versión a asignar siempre será 1.0.
- ⊗ De acuerdo a los cambios que se vayan realizando el nuevo número de versión se asignará de la siguiente manera:

- ◇ El primer número(a la izquierda del punto) se incrementará en 1 en cada cambio que se realice al entregable.

*1 para la primer versión, 2 para la versión que sufrió el primer cambio, etc.*

Ej:

*enunciadodetrabajo\_2.1f.pdf* significa que el producto *enunciadodetrabajo* ha sufrido un cambio respecto a su versión inicial.

- ◇ El segundo número(a la derecha del punto) indicará el número de la semana en que se llevó a cabo la modificación del entregable.

Ej:

Retomando el ejemplo del punto anterior *enunciadodetrabajo\_2.1f.pdf*

indicará no sólo que es la segunda versión del producto sino que el cambio se llevó a cabo en la semana número uno del proyecto.

- ◇ Para el caso de las versiones finales, se indicará dicha situación añadiendo una f al después del segundo número (a la derecha del punto), quedando:

*nombre\_versionf.tipo*, considerando el mismo ejemplo de los puntos anteriores tenemos el ejemplo:

*enunciadodetrabajo\_2.1f.pdf* que indica que la versión generada es la final.

- Definición del estándar de documentación.

Todo documento que se genere durante el proyecto se deberá respetar el siguiente formato de entrega:

- ⊗ Deberán ser creados en el editor de textos Word versión 2010.
- ⊗ El tamaño definido para todos los documentos será A4, fuente tipo Calibri y tamaño de la fuente 11 en caso de que no se especifique otro tamaño.
- ⊗ Deberá incluir los logotipos de la Universidad Nacional Autónoma de México (parte superior izquierda de la hoja) y de la Facultad de Medicina (parte superior derecha de la hoja).
- ⊗ El nombre del sistema se ubicará entre ambos logos definidos en el punto anterior (SIDEPE v.2.0) en tamaño de fuente 16.
- ⊗ Línea de separación en color azul entre la cabecera del documento definida en los puntos anteriores y el resto del documento.
- ⊗ Fecha del documento ubicada en la parte izquierda bajo el logo de la Universidad Nacional Autónoma de México y la línea que azul que especifica la cabecera del documento con el formato:  
*dia de mes del año*
- ⊗ Título del documento generado bajo el logo de la Facultad de Medicina y la línea que azul que especifica la cabecera del documento.
- ⊗ Versión del documento, ubicada bajo la fecha del documento.
- ⊗ La numeración de páginas se ubicará en la parte inferior del documento en un círculo color azul y con fuente color blanco.

- ⊗ Contenido de acuerdo a las plantillas planteadas por la norma 29110, para el caso de los documentos generados en base al Proceso Unificado se deberá presentar y aprobar una plantilla para poder crear el documento de manera correcta.
- ⊗ En caso de requerir tablas los bordes serán en color azul, los títulos de columna de éstas deberán tener fondo azul y fuente color blanco, el contenido deberá estar en fuente color negro.

Siguiendo los lineamientos antes mencionados, en la figura 5.4 se tiene definida la plantilla general a utilizar en este proyecto.

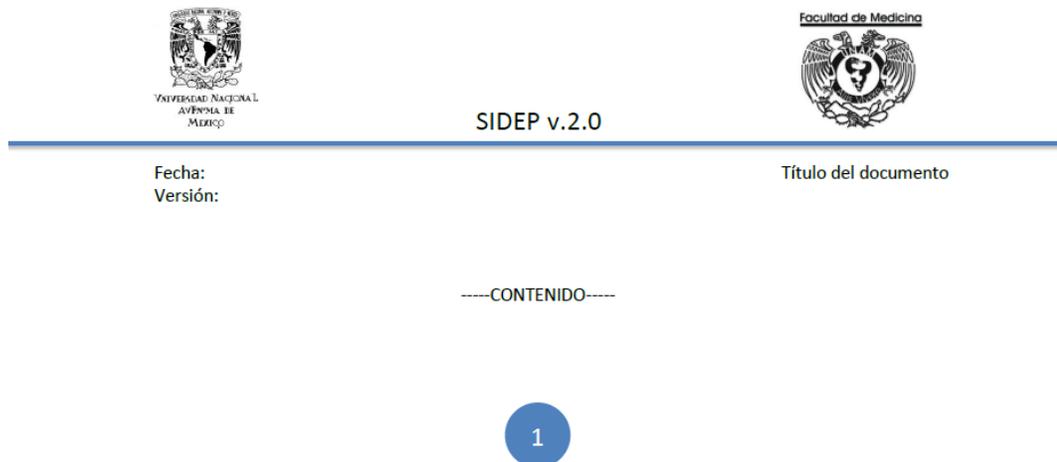


Figura 5.4: Diseño general de plantilla (encabezado y numeración de páginas).

En base a los puntos definidos anteriormente, podemos mostrar el primer documento generado, ver figura 5.5



La División de Estudios de Posgrado de la Facultad de Medicina, perteneciente a la Universidad Nacional Autónoma de México, a través de su Departamento de Cómputo y Sistemas ha solicitado la creación de un nuevo sistema para la administración de cursos, sedes y profesores de especialidades y altas especialidades que esta dependencia administra. La solución deberá basarse en el sistema con que se cuenta actualmente (Sidep.v.1.0) deberá concentrarse en la corrección de los errores que presenta Sidep.v.1.0.

1

Figura 5.5: Documento Enunciado de Trabajo.

### 5.3. AP.1.3 Identificar las tareas específicas a realizar para producir los entregables y sus componentes.

Como ya habíamos mencionado, se tomará la idea de los distintos tipos de reuniones de la metodología SCRUM para el desarrollo de este proyecto. Los elementos tomados de SCRUM se definen a continuación:

- *sprint*.

Para este proyecto los *sprints* tendrán una duración de una semana por lo que cada lunes se cerrará el *sprint* con la reunión *Revisión del sprint* y comenzará el

siguiente *sprint* con el *sprint Planning* correspondiente.

- Reuniones *sprint Planning*.

Se planean de acuerdo a la duración de los *sprints* definidos en el punto anterior, por lo cual en este proyecto se llevarán a cabo cada día lunes a las 11:00 hrs. en la oficina del Jefe del Departamento de Cómputo de la División de Estudios de Posgrado de la Facultad de Medicina. Los productos que se definan como entregables durante esta reunión se encontrarán detallados en el *Product Backlog*. El cual deberá cumplir con el estándar de documentación definido (ver figura 5.6) además de los datos mencionados en el encabezado general, incluirá el número de *sprint* al que pertenece debajo del nombre del documento.



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

SIDEP v.2.0



Facultad de Medicina

---

Fecha:

Versión:

Product Backlog

Sprint:

Producto	Características de entrega
<b>Producto 1</b>	Características que deberá cumplir el Producto 1 para su entrega
<b>Producto 2</b>	Características que deberá cumplir el Producto 2 para su entrega
<b>Producto 3</b>	Características que deberá cumplir el Producto 3 para su entrega
...	...
...	...
...	...
<b>Producto n</b>	Características que deberá cumplir el Producto n para su entrega

Figura 5.6: Plantilla para Product Backlog.

- Reuniones *sprint Planning Meeting*.

Para el proyecto se llevarán a cabo cada 7 días y más que planear los productos a efectuar durante un *sprint*, se plantearán los problemas a los que se ha enfrentado o bien ha detectado el equipo de trabajo, además de hacer una revisión general del estado en el que se encuentra el desarrollo del proyecto. Aquí añadiremos un nuevo documento para recabar la información durante la reunión, ver figura 5.7, se añadirá bajo el nombre del documento el número de reunión de tipo *sprint Planning Meeting* que se está realizando.



UNIVERSIDAD NACIONAL  
ANDRÉS BELLO

SIDEP v.2.0



FACULTAD DE MEDICINA

---

Fecha:

Versión:

Planning Meeting Backlog

Reunión:

Problemas	Estado general del proyecto
1.Problema	Situación del producto x1
2.Problema	Situación del producto x2
3.Problema	Situación del producto x3
4.Problema	Situación del producto x4
...	...
...	...
...	...
...	...
...	...
...	...
N.Problema	Situación del producto xn

Figura 5.7: Plantilla para documentar la información recabada en las reuniones *sprint Planning Meeting*.

- Reuniones *Daily Scrum*

Debido a que el tamaño del equipo es reducido, se manejarán reuniones tipo *Daily Scrum* en las cuales participará el único integrante del equipo de desarrollo y el Jefe del Departamento de Cómputo de la División, cuando sea requerido se contará con la participación del cliente.

- Lista de Tareas

Las tareas como lo especifica SCRUM, son acciones que derivan en funcionalidades del sistema, está compuesta bajo los siguientes **requerimientos** o *funcionalidades* que deberá implementar el sistema:

- ⊗ Permitirá el **registro de nuevos Profesores** y permitirá la **actualización o eliminación de cualquier Profesor existente**, así como la **asociación de estos a un curso y sede** manteniendo los datos para consultas posteriores.
- ⊗ Permitirá el **registro de nuevas sedes y nuevos cursos**.
- ⊗ Permitirá la **baja de sedes** eliminando también toda relación con profesores y cursos que ahí se imparte manteniendo los datos para consultas posteriores.
- ⊗ De manera similar al punto anterior permitirá la **baja de cursos** eliminando también toda relación con profesores y sedes en las que se encuentra activo el curso manteniendo los datos para consultas posteriores.
- ⊗ Permitirá el **registro de nuevos Jefes de Enseñanza** en cualquiera de las sedes disponibles permitiendo la eliminación del Jefe de Enseñanza de dicha sede en caso de existir manteniendo los datos para consultas posteriores, además de permitir la **eliminación o actualización de cualquier Jefe de Enseñanza existientem** manteniendo los datos para consultas posteriores.
- ⊗ Permitirá consultar los datos actuales y los de años anteriores, las consultas que debe manejar el sistema son:
  - ◇ Mostrar todos los Profesores registrados en un ciclo específico.
  - ◇ Mostrar los Profesores por Curso específico en un ciclo específico.
  - ◇ Mostrar los Profesores por Sede específica en un ciclo específico.

- ◊ Mostrar los Profesores por Sede y Curso específicos en un ciclo específico.
- ◊ Mostrar todos los Jefes de Enseñanza en un ciclo específico.
- ◊ Mostrar el Jefes de Enseñanza por Sede específica en un ciclo específico.
- ◊ Mostrar todos los Cursos disponibles en un ciclo específico.
- ◊ Mostrar todas las Sedes disponibles en un ciclo específico.
- ◊ Mostrar las Sedes de acuerdo a un Curso específico en un ciclo específico.
- ◊ Mostrar los Cursos de acuerdo a una Sede específica en un ciclo específico.

Si se diseña y construye un buen sistema no tendremos que preocuparnos por los errores encontrados en la versión anterior del sistema (Sidep.v.1.0) ya que el nuevo sistema se construirá bajo los mismos requerimientos.

La plantilla para el detallado de las tareas se muestra en la figura 5.8, en dicha plantilla deberá especificarse la tarea a realizar, la fecha de inicio y fin de la tarea, el tiempo estimado deberá ser medido en horas e inmediatamente deberá especificarse el tiempo invertido(serán las horas reales), por último una columna para observaciones o dependencia de tareas.



Ministerio de Salud  
Instituto de Servicios de Salud

SIDEP v.2.0



Facultad de Medicina

---

Fecha:  
Versión:

Tareas

Tarea	Fecha Inicio	Fecha Fin	Tiempo Estimado = Tiempo Invertido	Observaciones o dependencias
Tarea1	dd/mm/aaaa	dd/mm/aaaa	x hrs-x hrs.	Sin observaciones
Tarea2	dd/mm/aaaa	dd/mm/aaaa	x hrs-y hrs.	Depende de la tarea 1
Tarea3	dd/mm/aaaa	dd/mm/aaaa	y hrs-x hrs.	Sin observaciones
...	...	...	...	...
...	...	...	...	...
TareaN	dd/mm/aaaa	dd/mm/aaaa	z hrs-z hrs.	Depende de la tarea 3

Figura 5.8: Plantilla para documentar las tareas identificadas a desarrollar.

De acuerdo a las especificaciones de documentación y requerimientos mencionados y siguiendo la norma ISO/IEC 29110, hasta este punto tendremos las tareas listadas de manera general, de cierto modo, tendremos bloques de tareas con un fin en común, ver en las figuras 5.9, 5.10 y 5.11, los colores ayudan a diferenciar en que etapa se llevará a cabo la actividad.



SIDEP v.2.0

Fecha: 6 de junio del 2012  
 Versión: 1.0f

Tareas

Tarea	Fecha Inicio	Fecha Fin	Tiempo Estimado – Tiempo Invertido	Observaciones o dependencias
Análisis de requerimientos	//	//		
Prototipo de interfaz	//	//		
Diseño de base de datos	//	//		
Codificación de base de datos	//	//		
<b>Codificación de módulos de interfaz, control y conexión</b>				
Codificación de interfaces de inserciones	//	//		
Codificación de interfaces de eliminaciones	//	//		
Codificación de interfaces de actualizaciones	//	//		
Codificación de interfaces de consultas	//	//		
Codificación de clase de control	//	//		
Codificación de clase de conexión	//	//		
Codificación de lógica de interfaces de inserciones	//	//		
Codificación de lógica de interfaces de eliminaciones	//	//		

Figura 5.9: Bloque de tareas identificadas a desarrollar.



SIDEP v.2.0

Fecha: 6 de junio del 2012

Tareas

Versión: 1.0f

Codificación de lógica de interfaces de actualizaciones	//	//
Codificación de lógica de interfaces de consultas	//	//
<b>Integración de sistema</b>		
Integración de clase de conexión con clases de control	//	//
Integración de clases de control con interfaces de inserciones	//	//
Integración de clases de control con interfaces de eliminaciones	//	//
Integración de clases de control con interfaces de actualizaciones	//	10/10/12
Integración de clases de control con interfaces de consulta	//	//
<b>Pruebas</b>		
Pruebas de caja blanca de inserciones	//	//
Pruebas de caja blanca de eliminaciones	//	//
Pruebas de caja blanca de actualizaciones	//	//
Pruebas de caja blanca de	//	//

Figura 5.10: Bloque de tareas identificadas a desarrollar.



SIDEP v.2.0

Fecha: 6 de junio del 2012  
 Versión: 1.0f

Tareas

consultas		
Pruebas de caja negra de inserciones	//	//
Pruebas de caja negra de eliminaciones	//	//
Pruebas de caja negra de actualizaciones	//	//
Pruebas de caja negra de consultas	//	//
<b>Migración de Sidep v.1.0 a Sidep.v.2.0</b>		
Limpieza de datos de base de datos original para migración	//	//
Respaldo de datos(limpios) de la base de datos original	//	//
Inserción de datos en nueva base de datos	//	//
Adaptación de usuarios a nuevo sistema	//	//
Entrega del producto final	//	//

Figura 5.11: Bloque de tareas identificadas a desarrollar.

## 5.4. AP.1.4 Establecer la *Duración estimada* para realizar cada tarea.

De acuerdo a la lista presentada en el punto anterior, estimaremos el tiempo (en horas) que llevará realizar cada tarea definida, dado que nos apoyamos también en la metodología SCRUM, una tarea no puede durar más de 16 horas, así que los tiempos estimados estarán definidos de acuerdo a la restricción que nos indica la metodología. Tenemos entonces la nueva lista de tareas presentada en las figuras 5.12, 5.13 y 5.14, los colores ayudan a diferenciar en que etapa se llevará a cabo la actividad.



Facultad de Medicina



SIDEP v.2.0

Fecha: 6 de junio del 2012  
 Versión: 1.0f

Tareas

Tarea	Fecha Inicio	Fecha Fin	Tiempo Estimado – Tiempo Invertido	Observaciones o dependencias
Análisis de requerimientos	//	//	12 hrs-	
Prototipo de interfaz	//	//	16hrs-	
Diseño de base de datos	//	//	16hrs-	
Codificación de base de datos	//	//	16hrs-	
<b>Codificación de módulos de interfaz, control y conexión</b>				
Codificación de interfaces de inserciones	//	//	16hrs-	
Codificación de interfaces de eliminaciones	//	//	16hrs-	
Codificación de interfaces de actualizaciones	//	//	16hrs-	
Codificación de interfaces de consultas	//	//	16hrs-	
Codificación de clase de control	//	//	16hrs-	
Codificación de clase de conexión	//	//	16hrs-	
Codificación de lógica de interfaces de inserciones	//	//	16hrs-	
Codificación de lógica de interfaces de eliminaciones	//	//	16hrs-	

Figura 5.12: Tiempos estimados para los bloques de tareas.



SIDEP v.2.0

Fecha: 6 de junio del 2012  
 Versión: 1.0f

Tareas

Codificación de lógica de interfaces de actualizaciones	//	//	16hrs-
Codificación de lógica de interfaces de consultas	//	//	16hrs-
<b>Integración de sistema</b>			
Integración de clase de conexión con clases de control	//	//	16hrs-
Integración de clases de control con interfaces de inserciones	//	//	16hrs-
Integración de clases de control con interfaces de eliminaciones	//	//	16hrs-
Integración de clases de control con interfaces de actualizaciones	//	10/10/12	16hrs-
Integración de clases de control con interfaces de consulta	//	//	16hrs-
<b>Pruebas</b>			
Pruebas de caja blanca de inserciones	//	//	16hrs-
Pruebas de caja blanca de eliminaciones	//	//	16hrs-
Pruebas de caja blanca de actualizaciones	//	//	16hrs-
Pruebas de caja blanca de	//	//	16hrs-

Figura 5.13: Tiempos estimados para los bloques de tareas.



SIDEP v.2.0

Fecha: 6 de junio del 2012

Tareas

Versión: 1.0f

consultas			
Pruebas de caja negra de inserciones	//	//	16hrs-
Pruebas de caja negra de eliminaciones	//	//	16hrs-
Pruebas de caja negra de actualizaciones	//	//	16hrs-
Pruebas de caja negra de consultas	//	//	16hrs-
<b>Migración de Sidep v.1.0 a Sidep.v.2.0</b>			
Limpieza de datos de base de datos original para migración	//	//	16hrs-
Respaldo de datos(limpios) de la base de datos original	//	//	10hrs-
Inserción de datos en nueva base de datos	//	//	10hrs-
Adaptación de usuarios a nuevo sistema	//	//	16hrs-
Entrega del producto final	//	//	2hrs-

Figura 5.14: Tiempos estimados para los bloques de tareas.

## 5.5. AP.1.5 Identificar y documentar los recursos requeridos para el desarrollo del proyecto.

De acuerdo a la norma ISO/IEC 29110, debemos *Identificar y documentar los recursos: humanos, materiales, equipo y herramientas, estándares, incluyendo la capacitación requerida para que el Equipo de Trabajo pueda realizar el proyecto. Así como incluir las fechas en el calendario cuando sean requeridos los recursos y la capacitación.*

La documentación con estas especificaciones se pueden ver en las figuras 5.15, 5.16, se debe recordar que durante este capítulo se ha definido ya el estándar de documentación y control de versiones.



SIDEP v.2.0

Fecha: 6 de junio del 2012  
Versión: 1.0f

Material y herramientas.

El material y las herramientas empleados durante el desarrollo de este proyecto han sido definidas de acuerdo a los requerimientos del cliente así como las especificaciones de desarrollo planteadas por el Jefe del Departamento de Cómputo de la División de Estudios de Posgrado de la Universidad Nacional Autónoma de México.

Material	Herramientas
1.Equipo de escritorio proporcionado por el Departamento de Cómputo de la División de Estudios de Posgrado.	1.Entorno de desarrollo NetBeans versión 6.9.1
2.Servidor Dell con Windows Server 2008 en el cual se instalará la aplicación para su uso por la División de Estudios de Posgrado.	2.Servidor web GlassFish versión 2.1.1
	3.Dreamweaver versión CS5
	4.PostgreSQL versión 8.4

Figura 5.15: Material y herramientas requeridos para el desarrollo del proyecto.



SIDEP v.2.0

Fecha: 6 de junio del 2012  
 Versión: 1.0f

Estándar de Codificación

Seguiremos las convenciones de código para el lenguaje Java definidas por Sun Microsystems Inc. y de las cuales se puede encontrar referencia en:

<http://www.um.es/docencia/vjimenez/ficheros/practicas/ConvencionesCodigoJava.pdf>.

Indicaremos las más importantes además de las convenciones para codificación de la Base de Datos.

Tipo	Estándar	Ejemplo
Tablas	Las tablas se nombran con sustantivos, siempre en singular y comenzado siempre por Mayúscula.	<code>create table Curso{</code>
Atributos	Los atributos se nombran con sustantivos, siempre en singular y comenzando siempre por una letra minúscula que indica el tipo de dato del atributo, serán mayúsculas cuando se indique el comienzo de un sustantivo nuevo dentro del nombrado.	<code>nIdCurso integer;</code>
Clases	Las clases se nombran con sustantivos comenzando siempre cada uno de ellos con Mayúscula.	<code>class Curso{</code>
Métodos	Los métodos se nombran con verbos en minúscula, en caso de añadir un sustantivo este comenzará con Mayúscula.	<code>asignaCurso{</code>
Variables	Se nombran en minúsculas y utilizando las últimas letras del abecedario.	<code>int x; double y; String z;</code>
Constantes	Se nombran en minúsculas y utilizando las primeras letras del abecedario.	<code>int a = 5; double b = 1.5; String c = "2010-2011";</code>

Figura 5.16: Estándares para el desarrollo del proyecto.

## 5.6. AP.1.6 Establecer la *Estructura del Equipo de Trabajo*

De la norma ISO/IEC 29110 se debe *Establecer la Estructura del Equipo de Trabajo, asignando los roles y responsabilidades acordes a los Recursos*. De acuerdo a esto, la composición del Equipo de Trabajo queda definida en el documento que se muestra en la figura 5.17.




SIDEP v.2.0

---

Fecha: 6 de junio del 2012  
Versión: 1.0f

Equipo de Trabajo

El equipo de trabajo para el desarrollo del proyecto estará compuesto por:

Nombre	Rol
Dr. Agles Cruz Avelar	Cliente
M. en I. David Velázquez Portilla	Administrador de Proyecto
Erick Orlando Matia Cruz	Equipo de desarrollo

Figura 5.17: Equipo de trabajo para el desarrollo del proyecto.

## 5.7. AP.1.7 Asignar la estimación inicial y las fechas determinadas para cada tarea.

El fin de esta tarea es crear el *Calendario de Tareas del Proyecto* considerando los recursos asignados, la secuencia y dependencia de tareas.

Asignaremos rangos de fechas de acuerdo a su tiempo estimado de trabajo, considerando que hay 3 semanas de vacaciones (11 de julio del 2012 al 29 de julio del 2012), ver las figuras 5.18, 5.19 y 5.20, además generamos el diagrama de gantt del proyecto el cual nos permitirá plasmar de manera gráfica la duración de **los bloques de tareas** del proyecto, los colores ayudan a diferenciar en que etapa se llevará a cabo la actividad.



SIDEP v.2.0

Fecha: 6 de junio del 2012  
 Versión: 1.0f

Tareas

Tarea	Fecha Inicio	Fecha Fin	Tiempo Estimado – Tiempo Invertido	Observaciones o dependencias
Análisis de requerimientos	11/06/12	29/06/12	12 hrs-	
Prototipo de interfaz	23/07/12	27/07/12	16hrs-	
Diseño de base de datos	30/07/12	06/08/12	16hrs-	
Codificación de base de datos	07/08/12	09/08/12	16hrs-	
<b>Codificación de módulos de interfaz, control y conexión</b>				
Codificación de interfaces de inserciones	13/08/12	17/08/12	16hrs-	
Codificación de interfaces de eliminaciones	10/09/12	14/09/12	16hrs-	
Codificación de interfaces de actualizaciones	03/09/12	07/09/12	16hrs-	
Codificación de interfaces de consultas	17/09/12	21/09/12	16hrs-	
Codificación de clase de conexión	10/08/12	10/08/12	16hrs-	
Codificación de lógica de interfaces de inserciones	20/08/12	24/08/12	16hrs-	
Codificación de lógica de interfaces de eliminaciones	10/09/12	14/09/12	16hrs-	
Codificación de lógica de interfaces de actualizaciones	17/09/12	21/09/12	16hrs-	

Figura 5.18: Fechas especificadas para el desarrollo de las tareas del proyecto.



SIDEP v.2.0

Fecha: 6 de junio del 2012

Tareas

Versión:1.0f

Codificación de lógica de interfaces de consultas	24/09/12	28/09/12	16hrs-
Codificación de Validaciones	01/10/12	05/10/12	16hrs-
<b>Integración de sistema</b>			
Integración de clase de conexión con clases de control	08/10/12	10/10/12	16hrs-
Integración de clases de control con interfaces de inserciones	11/10/12	15/10/12	16hrs-
Integración de clases de control con interfaces de eliminaciones	16/10/12	18/10/12	16hrs-
Integración de clases de control con interfaces de actualizaciones	19/10/12	23/10/12	16hrs-
Integración de clases de control con interfaces de consulta	24/10/12	26/10/12	16hrs-
<b>Pruebas</b>			
Pruebas unitarias de inserciones	20/08/12	24/08/12	16hrs-
Pruebas unitarias de eliminaciones	10/09/12	14/09/12	16hrs-
Pruebas unitarias de actualizaciones	03/09/12	07/09/12	16hrs-
Pruebas unitarias de consultas	17/09/12	21/09/12	16hrs-
Pruebas Historias de Usuario de inserciones	27/08/12	31/08/12	16hrs-

Figura 5.19: Fechas especificadas para el desarrollo de las tareas del proyecto.



SIDEP v.2.0

Fecha: 6 de junio del 2012

Tareas

Versión:1.0f

Pruebas Historias de Usuario de eliminaciones	10/09/12	14/09/12	16hrs-
Pruebas Historias de Usuario de actualizaciones	03/09/12	07/09/12	16hrs-
Pruebas Historias de Usuario de consultas	24/09/12	28/11/12	16hrs-
<b>Migración de Sidep v.1.0 a Sidep.v.2.0</b>			
Limpieza de datos de base de datos original para migración	12/11/12	21/11/12	16hrs-
Respaldo de datos(limpios) de la base de datos original	22/11/12	22/11/12	10hrs-
Inserción de datos en nueva base de datos	23/11/12	23/11/12	10hrs-
Adaptación de usuarios a nuevo sistema	26/11/12	06/12/12	16hrs-
Entrega del producto final	07/12/12	07/12/12	2hrs-



Figura 5.20: Fechas especificadas para el desarrollo de las tareas del proyecto.

## 5.8. AP.1.8 Calcular y Documentar la *Estimación del Esfuerzo y Costo.*

De acuerdo a la proyección que se realizó en la sección anterior, mostramos en la figura 5.21 el *Diagrama de Gantt* correspondiente a este proyecto.

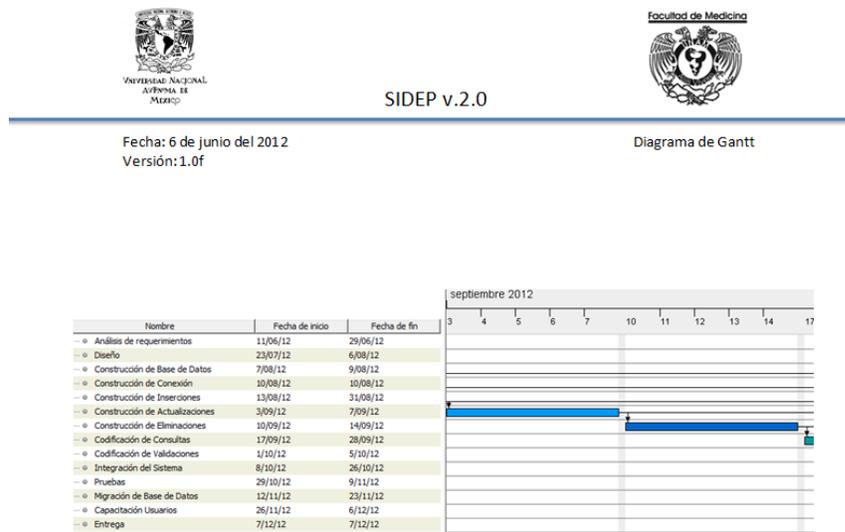


Figura 5.21: Diagrama de Gantt de las tareas del proyecto.

## **5.9. AP.1.9 Identificar y documentar los riesgos que pueden afectar al proyecto.**

En la figura 5.22 se especifican los riesgos identificados que podrían afectar al desarrollo del proyecto, indicando el impacto que causarían al desarrollo (bajo, medio, alto), que tan probable es que ocurra dicho riesgo (bajo, medio, alto) y una estrategia para el manejo en caso de ocurrir.



SIDEP v.2.0

Fecha: 6 de junio del 2012  
 Versión: 1.0f

Riesgos

Riesgo	Frecuencia	Impacto	Estrategia de manejo
Falla en el suministro eléctrico.	Bajo	Medio	Esperar a que se reestablezca el servicio y reajustar los horas de trabajo de acuerdo a las fechas planteadas.
Enfermedad de alguno de los miembros del equipo.	Bajo	Alto	Repartir el trabajo entre el resto del equipo o bien reajustar el trabajo de acuerdo a las fechas planteadas.
Problemas con el repositorio o los archivos fuentes.	Bajo	Alto	Mantener respaldos diarios del proyecto.
Fallas en la estimación del tiempo requerido.	Bajo	Alto	Generar una buena estimación de los tiempos necesarios para cumplir con el proyecto.

Figura 5.22: Registro de riesgos posibles.

## **5.10. AP.1.10 Documentar la *Estrategia de Control de Versiones.***

Anteriormente se estableció el control de versiones para la documentación. En esta sección se definirá el control de versiones para los entregables de codificación del proyecto, generando así el control de versiones para todos los productos a entregar del proyecto.



## SIDEP v.2.0

Fecha: 6 de junio del 2012  
Versión: 1.0f

Control de Versiones

Seguiremos el estándar definido para la documentación, mezclando en este caso los entregables codificados y los directorios del repositorio.

- Sólo se le asignará un control de versión a un entregable, esto es, una tarea completada (documento o archivos de código).
- El primer número de versión a asignar siempre será 1.0.
- De acuerdo a los cambios que se vayan realizando (documentos, archivos de código) el nuevo número de versión se asignará de la siguiente manera:
  - El primer número (a la izquierda del punto) se incrementará en 1 en cada cambio que se realice al entregable.  
*1 para la primera versión, 2 para la versión que sufrió el primer cambio, etc.*
  - El segundo número (a la derecha del punto) indicará el número de la semana en que se llevó a cabo la modificación del entregable.
  - Para las versiones finales, se indicará dicha situación añadiendo una f después del segundo número (a la derecha del punto), quedando:  
*nombre\_versionf.tipo*

Dado que el código se desarrollará en NetBeans, se creará un repositorio (detallado en el documento Repositorio) en el cual se creará una copia de cada entregable que se vaya generando en NetBeans y ahí se le asignará el control de versión.

Para el manejo de directorios en el repositorio la asignación de versiones será de la misma manera, en este caso se le cambiará la versión al directorio **CADA QUE ALGUNO DE LOS ARCHIVOS QUE CONTenga SUFRa ALGÚN CAMBIO.**

## **5.11. AP.1.11 Generar el *Plan del Proyecto* integrando los elementos previamente identificados y documentados.**

Para generar el *Plan del Proyecto* se hará un compendio y se ordenarán los documentos ya creados. El Plan de Proyecto completo puede ser consultado en el Apéndice A1.

## **5.12. AP.1.12 Incluir la descripción del producto.**

Esta sección incluye el alcance del producto, objetivos del proyecto y del equipo de desarrollo así como la licencia del software a desarrollar.

- Alcance del producto.

El alcance del producto estará guiado por el *Enunciado de Trabajo*, ya que el objetivo de este proyecto es atenderlo totalmente. En la figura 5.24 se muestran los elementos relacionados u afectados con el proyecto.



SIDEP v.2.0

Fecha: 6 de junio del 2012  
Versión: 1.0f

Alcance

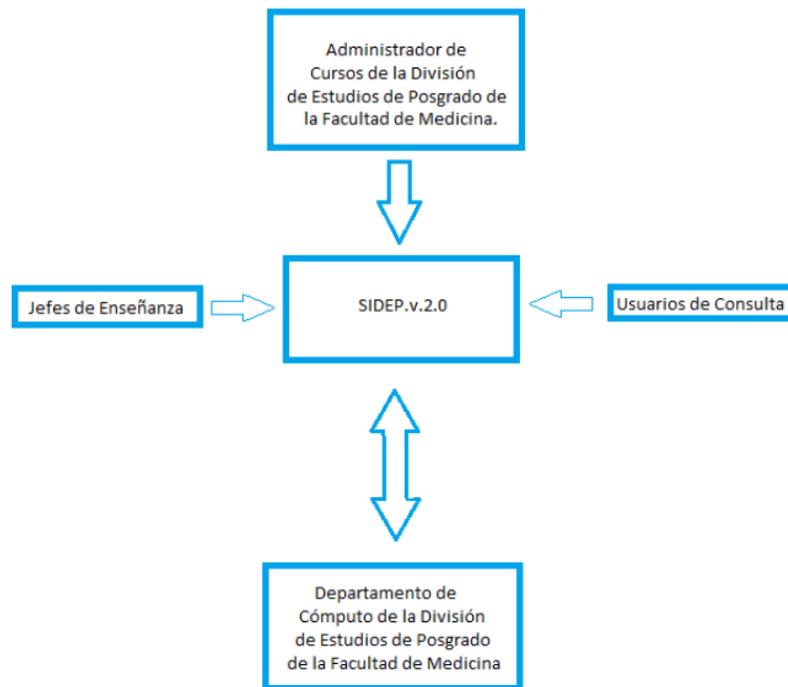


Figura 5.24: Alcance del Proyecto.

- Objetivo del Proyecto.

Construir un producto que cumpla de manera correcta con los requerimientos planteados en el capítulo anterior de una manera simple e intuitiva para el usuario. Con esto se pretende una migración y cambio de sistema que impacte lo menos posible en las actividades diarias de los usuarios del sistema de la División de Estudios de Posgrado de la Facultad de Medicina.

- Objetivo del Equipo de Desarrollo.

Mantener un buen ambiente de trabajo con el cliente basado en la comunicación y trabajo continuo, cumpliendo con las tareas planeadas durante cada uno de los *sprints* que tenga de duración el proyecto para así generar un software de calidad.

- Licencia

El software desarrollado y aquí mencionado es propiedad de la División de Estudios de Posgrado de la Facultad de Medicina de la Universidad Nacional Autónoma de México [3] y quedará a resguardo del Jefe de Cómputo de la División.

### **5.13. AP.1.13 Verificación del proyecto / AP.1.14 Revisar y obtener la aprobación del *Plan del Proyecto*.**

De acuerdo a lo presentado en las secciones anteriores, se conformó el *Plan del Proyecto*, mismo que fué presentado en la junta del día 8 de junio del 2012 al Jefe del Departamento de Cómputo de la División de Estudios de Posgrado de la Facultad de Medicina, quien aprobó el proyecto.

### **5.14. AP.1.15 Establecer el repositorio del proyecto usando la *Estrategia de Control de Versiones*.**

El repositorio del proyecto contendrá todos los elementos y componentes generados, especificando su versión de acuerdo al documento *versiones\_1.0f.pdf*, ver figura 5.25.



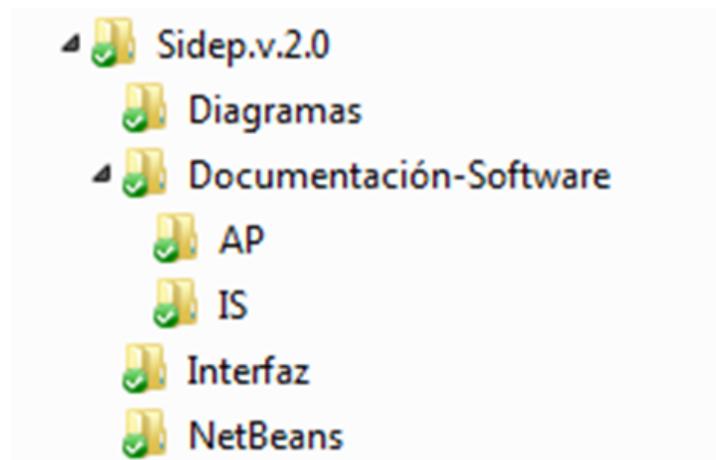
SIDEP v.2.0



Fecha: 6 de junio del 2012  
Versión: 1.0f

Repositorio

La estructura del repositorio será la siguiente:



En los directorios mostrados la información se almacenará de la siguiente manera:

- Directorio Sidep.v.2.0 -> Directorio general del proyecto.
- Diagramas -> Almacenará todos los diagramas generados durante el proyecto.
- Documentación Software -> Almacenará la documentación y código generados durante el proyecto y que se creó siguiendo la norma ISO/IEC 29110, los subdirectorios AP e IS representan la separación de la documentación de las dos etapas especificadas por la norma.
- Interfaz -> Almacenará los archivos de interfaz del sistema, aquí se realizará la copia de los códigos generados en el proyecto de NetBeans asignándoles un número de versión.
- NetBeans -> Será el directorio creado por el entorno de desarrollo y sobre el cual se codificará.

Figura 5.25: Especificación del Repositorio del Proyecto.

---

## Capítulo 6

# Inicio de la Implementación de Software

De acuerdo a la norma ISO/IEC 29110, esta actividad asegura que el *Plan del Proyecto* establecido en la actividad Planeación del Proyecto sea llevado a cabo por el Equipo de Trabajo [7]. Esta actividad provee:

- Una revisión del *Plan del Proyecto* por parte del Equipo de Trabajo para determinar la asignación de las tareas.
- Un compromiso por parte del Equipo de Trabajo y del Administrador de Proyecto con el *Plan del Proyecto*.
- El establecimiento de un ambiente para la implementación[7].

Para ésta y las subsecuentes actividades definimos los roles que estarán involucrados y que se sumarán a los ya definidos en la Actividad Planeación del Capítulo 4 de acuerdo a la norma ISO/IEC 29110[7]:

- Analista (AN).

Miembro del Equipo de Trabajo que deberá contar con:

- ⊗ Conocimiento y experiencia que permita deducir, especificar y analizar los requerimientos.
- ⊗ Conocimiento en diseño de interfaces de usuario y criterios ergonómicos.

- ⊗ Conocimiento de técnicas de revisión.
- ⊗ Conocimiento de técnicas de edición.
- ⊗ Experiencia en desarrollo y mantenimiento de software.
- Diseñador (DIS).  
Miembro del Equipo de Trabajo que deberá contar con:
  - ⊗ Conocimiento y experiencia en los componentes y diseño de arquitectura.
  - ⊗ Conocimiento de técnicas de revisión.
  - ⊗ Conocimiento y experiencia en la planeación y ejecución de pruebas de integración.
  - ⊗ Conocimiento de técnicas de edición.
  - ⊗ Experiencia en desarrollo y mantenimiento de software.
- Programador (PR).  
Miembro del Equipo de Trabajo que deberá contar con:
  - ⊗ Conocimiento y/o experiencia en programación, integración y pruebas unitarias.
  - ⊗ Conocimiento de técnicas de revisión.
  - ⊗ Conocimiento de técnicas de edición.
  - ⊗ Experiencia en desarrollo y mantenimiento de software.

La norma ISO/IEC 29110 nos provee la lista de tareas [7] la cual se encuentra en el Apéndice Implementación del Software en la figura B.2.

## 6.1. I.S.1.1 Revisar el *Plan del Proyecto* / I.S.1.2 Establecer el ambiente de implementación.

De acuerdo a lo definido en la tabla de tareas, se generó un Plan del Proyecto entre Administrador del Proyecto y Equipo de Trabajo el cual incluye juntas diarias y semanales tomadas de la metodología ágil SCRUM, en ellas se verá el avance del proyecto de acuerdo a 3 preguntas:

- ¿Qué has hecho desde ayer? - ¿Qué has hecho desde la semana pasada?
- ¿Qué es lo que harás hasta la reunión de mañana? - ¿Qué es lo que harás hasta la reunión de la próxima semana?
- ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo? [11]

Además el Equipo de Trabajo generó el documento de la figura 6.1 aprobado por el Cliente y el Administrador del Proyecto que especifica el ambiente con el que se debe de contar para poder llevar a cabo el proyecto; la asignación de tareas en nuestro caso serán distribuidas en su totalidad al *Equipo de Trabajo* sin distinción de roles internos debido al tamaño reducido del equipo. La lista de tareas se tomará del documento *diagramagantt\_1.0f.pdf* y el documento final se muestra en la figura 6.2.



Para llevar a cabo el Plan de Trabajo que se presenta en el repositorio como *plantrabajo\_1.0f.pdf*, se requiere que el Equipo de Trabajo cuente con:

- Equipo de cómputo destinado **EXCLUSIVAMENTE** para el uso del *Equipo de Trabajo* y proporcionado por el Departamento de Cómputo de la División de Estudios de Posgrado de la Facultad de Medicina.
- El *Equipo de Trabajo* se dedicará en su totalidad al desarrollo del proyecto, en caso de requerir que este realice otras actividades se podría ver afectada la calendarización presentada en *gantt\_1.0f.pdf* lo que podría afectar a la entrega final del proyecto.
- El *Plan de Trabajo* presentado en *plantrabajo\_1.0f.pdf* ha sido aprobado y aceptado por el Cliente, el Administrador del Proyecto y el Equipo de Trabajo considerando que las fechas son las adecuadas para el desarrollo del sistema, por lo cual no se permiten modificaciones con el fin de reducir tiempos ya que esto podría afectar la relación entre los involucrados en el proyecto.

Figura 6.1: Ambiente establecido de trabajo.



SIDEP v.2.0

Fecha: 6 de junio del 2012  
 Versión: 1.0f

Diagrama de Gantt

Todas las Tareas son Asignadas al Equipo de Trabajo conformado por una única persona.

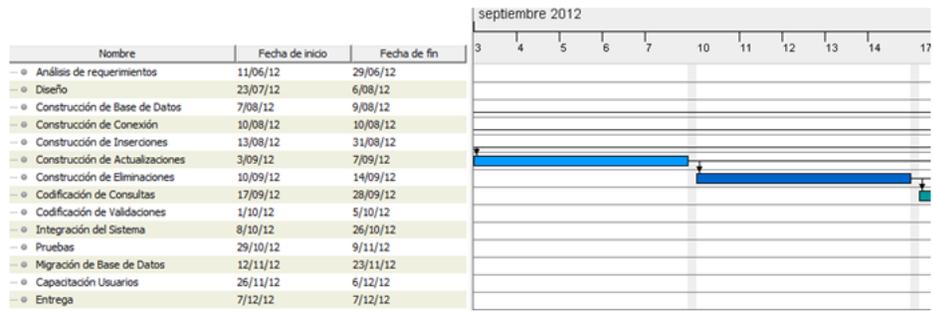


Figura 6.2: Tareas asignadas.

---

## Capítulo 7

# Análisis de Requerimientos de Software.

De acuerdo a la norma ISO/IEC 29110, esta actividad examina las necesidades del cliente y establece los requerimientos del proyecto. Esta actividad provee:

- Una revisión del *Plan del Proyecto* por parte del Equipo de Trabajo para determinar la asignación de las tareas.
- El análisis y especificación de los requerimientos del cliente.
- Un acuerdo sobre los requerimientos con el cliente.
- Una verificación y validación de los requerimientos.
- Un control de versiones de los requerimientos de producto de software.

En la tabla 16 tomada de la Norma ISO/IEC 29110 [7] y que se encuentra en el Apéndice Implementación del Software, se listan las tareas de esta actividad y sus respectivos roles y productos de trabajo las cuales serán una guía para el desarrollo de la etapa.

## 7.1. I.S.2.1 Asignar tareas a los miembros del Equipo de Trabajo de acuerdo a cada rol, basadas en el *Plan del Proyecto* actual.

De acuerdo a SCRUM se definen tres roles fundamentales para la Metodología: Scrum Master, Product Owner y Equipo de Trabajo.

Por otro lado, la norma ISO/IEC 29110 define a: Cliente, Administrador de Proyecto y varios roles más que conforman al Equipo de Desarrollo.

Debido al tamaño del Equipo de Trabajo, que es una persona, se tomó la decisión de fusionar los roles de la siguiente manera:

- Producto Owner - Cliente.  
Administrador de Cursos de la División de Estudios de Posgrado de la Facultad de Medicina asignado a Dr. Agles Avelar Cruz.
- Scrum Master - Administrador de Proyecto.  
Jefe del Departamento de Cómputo y Sistemas de la División de Estudios de Posgrado de la Facultad de Medicina asignado al Fis. David Velázquez Portilla
- Equipo de Desarrollo - Equipo de Trabajo.  
Asignado a Erick Orlando Matla Cruz.

## 7.2. I.S.2.2 Documentar o actualizar la *Especificación de Requerimientos*.

El documento que refleja las actualizaciones sobre la Especificación de Requerimientos en forma de lista se puede encontrar en el Apéndice A A.8, además se modelaron los requerimientos del sistema empleando una documentación híbrida: Casos de Uso e Historias de Usuario.

Mediante casos de uso se modelaron las funcionalidades de:

- Ingresar al Sistema.
- Salir del Sistema.

Mientras que utilizando Historias de Usuario se modelaron las funcionalidades de:

- Ciclo
  - ⊗ Alta
  - ⊗ Eliminación
  
- Currícula
  - ⊗ Alta
  - ⊗ Actualización
  - ⊗ Eliminación
  - ⊗ Consulta
  
- Curso
  - ⊗ Alta
  - ⊗ Actualización
  - ⊗ Eliminación
  - ⊗ Consulta
  
- Jefe de Enseñanza
  - ⊗ Alta
  - ⊗ Actualización
  - ⊗ Eliminación
  - ⊗ Consulta
  
- Nombramiento
  - ⊗ Alta
  - ⊗ Actualización
  - ⊗ Eliminación
  - ⊗ Consulta

- Profesor

- Alta
- Actualización
- Eliminación
- Consulta

- Sede

- Alta
- Actualización
- Eliminación
- Consulta

- Usuario

- Alta
- Actualización
- Eliminación
- Consulta

En la figura 7.1 se muestra el Diagrama General de Casos de Uso que incluye todas las funcionalidades del sistema. En verde se indican aquellas cuyo detalle se obtuvo a través de Casos de Uso y en azul aquellas en las que se utilizaron Historias de Usuario.





SIDEP v.2.0

Fecha: 22 de junio del 2012  
 Versión: 5.6f

Caso de Uso 1: Ingreso al Sistema

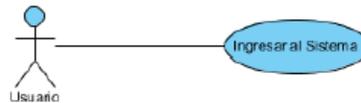
**Descripción:** Permitirá al usuario acceder al sistema mediante un usuario y contraseña previamente proporcionados. El sistema obtendrá los permisos del usuario y verificará si se trata de un Jefe de Enseñanza o no.

**Precondiciones:** El usuario cuenta con usuario y contraseña e ingresó a la dirección [www.sidep.fmposgrado.unam.mx:8080/Sidep.v.2.0/](http://www.sidep.fmposgrado.unam.mx:8080/Sidep.v.2.0/)

Datos requeridos:

Dato	Descripción	Requerido
Usuario	Cadena de caracteres alfanuméricos que representa el nombre de usuario.	Si
Contraseña	Cadena de caracteres alfanuméricos que representa la clave de acceso del usuario.	Si

**Postcondiciones:** El usuario será redirigido a menú.jsp con sus permisos asociados y su estado de Jefe de Enseñanza o no.



Usuario		Sistema		Excepción
Paso	Acción	Paso	Acción	
1	Ingresar usuario y contraseña previamente proporcionados.			
2	Da click en el botón <i>Ingresar</i> .	3	Recibe los parámetros usuario y contraseña ingresados por el usuario y va a <i>index_proceso.jsp</i>	<b>E1</b>

1

Figura 7.2: Caso de Uso 1 - Ingresar al Sistema

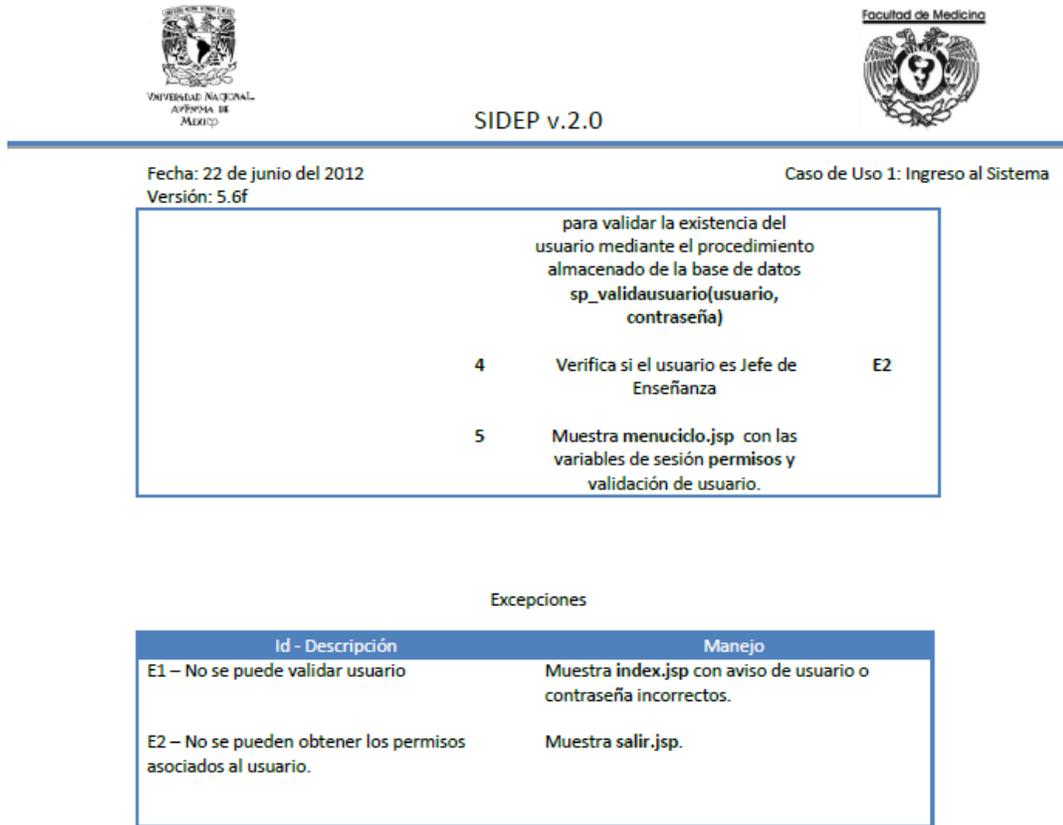


Figura 7.3: Caso de Uso 1 - Ingresar al Sistema

En la figura 7.4 se muestra el detallado del Caso de Uso - Salir del Sistema.



SIDEP v.2.0

Fecha: 22 de junio del 2012  
Versión: 1.0f

Caso de Uso 9: Salir del Sistema

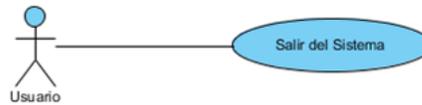
**Descripción:** Permitirá al usuario cerrar su sesión para salir del sistema.

**Precondiciones:** El usuario se encuentra logueado en el sistema.

Datos requeridos:

Dato	Descripción	Requerido
	Ninguno	

**Postcondiciones:** El usuario será redirigido a index.jsp.



Usuario		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Da click en el botón <i>Salir</i> .	2	Recupera todas las variables de sesión y las finaliza.	E1
		3	Muestra <i>index.jsp</i>	

Excepciones

Id - Descripción	Manejo
E1 – No se pueden recuperar todas las variables de sesión.	Muestra mensaje de error indicando que no se pudo terminar la sesión

Figura 7.4: Caso de Uso 9 - Salir del Sistema

A continuación en la figura 7.5 se muestra la Historia de Usuario - Alta de Profesor. Cabe mencionar que dada la naturaleza de las Historias de Usuario, cuyo detalle se presentó en el capítulo 2, se utilizaron entre dos y cuatro Historias por cada Caso de Uso presentado en la figura 7.1.

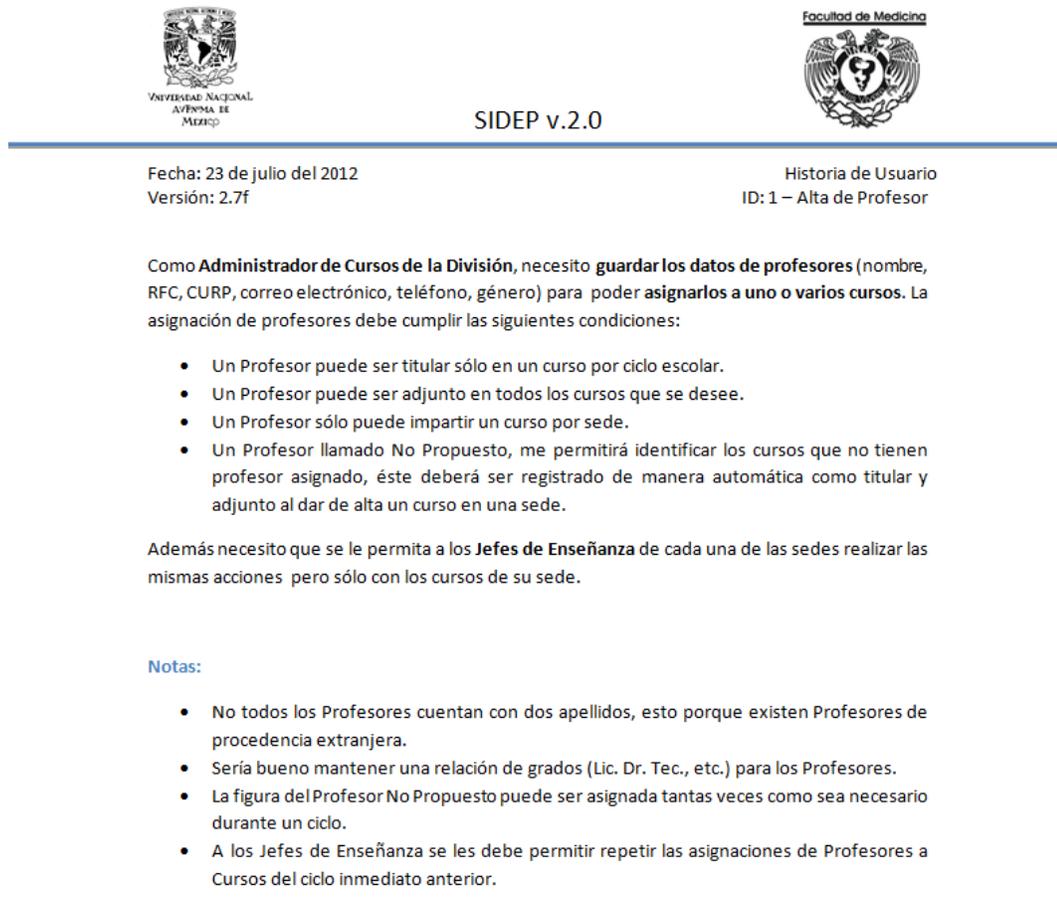
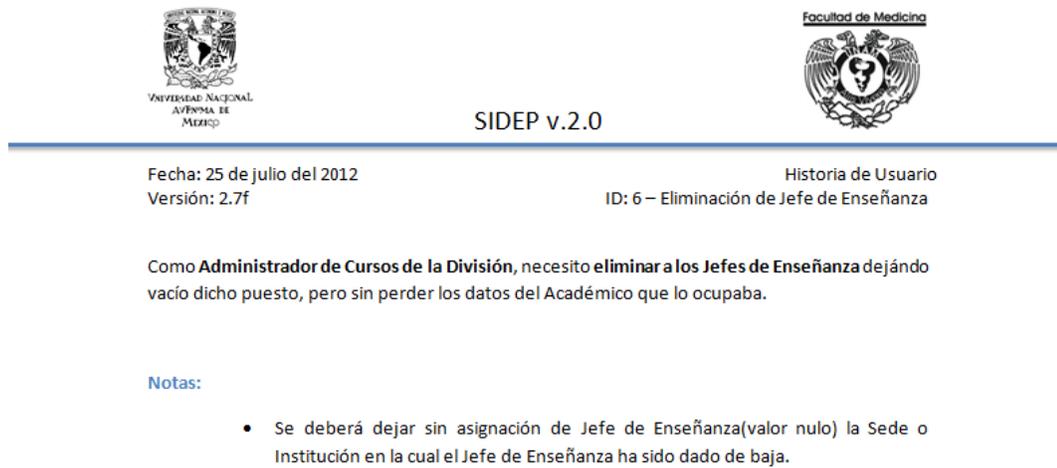


Figura 7.5: Historia de Usuario - Alta de Profesor.

En la figura 7.6 se muestra la Historia de Usuario - Baja de Profesor.



SIDEP v.2.0

---

Fecha: 25 de julio del 2012  
Versión: 2.7f

Historia de Usuario  
ID: 6 – Eliminación de Jefe de Enseñanza

Como **Administrador de Cursos de la División**, necesito **eliminar a los Jefes de Enseñanza** dejando vacío dicho puesto, pero sin perder los datos del Académico que lo ocupaba.

**Notas:**

- Se deberá dejar sin asignación de Jefe de Enseñanza(valor nulo) la Sede o Institución en la cual el Jefe de Enseñanza ha sido dado de baja.

Figura 7.6: Historia de Usuario - Eliminación de Jefe de Enseñanza.

En la figura 7.7 se muestra la Historia de Usuario - Actualización de Sede.



SIDEP v.2.0



Fecha: 23 de julio del 2012  
Versión: 1.0f

Historia de Usuario  
ID: 11 – Actualización de Sede

Como **Administrador de Cursos de la División**, necesito **actualizar los datos de las Sedes** con las que cuento; además debo poder **actualizar los datos asociados a una Sede y un Curso** como son número de inscritos y observaciones.

**Notas:**

- Se debe cuidar que las claves y nombres de actualización sean únicos.

Figura 7.7: Historia de Usuario - Actualización de Sede.

En la figura 7.8 se muestra la Historia de Usuario - Consulta de Profesor.



SIDEP v.2.0



Fecha: 23 de julio del 2012  
Versión: 1.0f

Historia de Usuario  
ID: 24 – Consulta de Profesores

Como **Administrador de Cursos de la División**, necesito **consultar la información de los Profesores** que tengo registrados en un ciclo escolar específico, debo poder consultarlos de la siguiente manera:

- Todos
- Total de Titulares
- Total de Adjuntos
- Total de No Propuestas
- Total por Sede
- Total por Sede y Curso
- Total por Curso
- Total por Curso y Sede

Estas consultas las podrán realizar los Jefes de Enseñanza con los Profesores de su respectiva Sede. Además necesito **imprimir o guardar en .pdf** cada una de estas consultas.

**Notas:**

- El formato del .pdf es libre.

Figura 7.8: Historia de Usuario - Consulta de Profesores.

### **7.3. I.S.2.3 Verificar la *Especificación de Requerimientos* y proponer una *Solicitud de Cambio*.**

Para asegurar que las funcionalidades identificadas y detalladas anteriormente, satisfacen plenamente las necesidades del Cliente, se realizaron revisiones para que éste las validara. Además se generó un formato para que el cliente pudiera solicitar cambios a funcionalidades. A continuación se presenta con mayor detalle dicho elemento.

- Solicitud de Cambio.

Se utilizará cuando se requiera modificar alguna funcionalidad que ya ha sido definida o bien que ya se ha construido pero debe ser modificada. La plantilla para estas solicitudes se muestra en la figura 7.9, en caso de que el cambio requiera la modificación de otra tarea o producto ésta se plasmará en otra solicitud de cambio y se hará referencia en la solicitud de cambio que generó dicha modificación.

 UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO	<b>SIDEP v.2.0</b>	 Facultad de Medicina
Fecha: 7 de junio del 2012 Versión: 1.0f	Plantilla Solicitud de Cambio	
Producto	ProductoX	
Fecha de Solicitud	dd/mm/aaaa	
Prioridad	Baja/Media/Alta	
Descripción del Cambio Solicitado	Descripción detallada del cambio solicitado sobre el ProductoX	
Productos Dependientes Afectados	Productos afectados en la modificación del Producto X:  ProductoY se afecta en ... ProductoZ se afecta en ...  El cambio en el ProductoX implica un cambio en el ProductoW que se especifica en la SolicituddeCambioX 1.0_f.pdf	
Tiempo estimado	Medido en horas: X horas	

Figura 7.9: Plantilla para Solicitar un Cambio durante el desarrollo del Proyecto.

#### **7.4. I.S.2.4 Validar y obtener la aprobación de la *Especificación de Requerimientos.***

En la junta del día 11 de junio del 2012 se aprobó la Especificación de Requerimientos por parte del Cliente y Administrador de Proyecto, en la cual estuvo presente el Equipo de Desarrollo. Al ser un desarrollo ágil, no hubo necesidad de generar un producto de trabajo como evidencia tangible del hecho.

#### **7.5. I.S.2.5 Documentar la versión preliminar del *Manual de Usuario* o actualizar la versión existente.**

El Equipo de Desarrollo propone como versión preliminar del Manual de Usuario una tabla de contenido, esto porque si bien se tienen claros los requerimientos, aún no se tiene un diseño de lo que será el Sistema por lo que hasta este momento no se puede proponer algo con mayor detalle. La tabla de contenido propuesta para el Manual de Usuario se puede ver en la figura 7.10.



SIDEP v.2.0

Fecha: 8 de junio del 2012  
Versión: 1.0f

Manual de Usuario (Preliminar)

Índice

1. Introducción
2. Requerimientos del Sistema
3. Instalación de la Aplicación
4. Uso de la Aplicación
5. Glosario

Figura 7.10: Índice preliminar del Manual de Usuario que se entregará con el Sistema Final.

## **7.6. I.S.2.6 Verificar y obtener la aprobación del *Manual de Usuario*.**

De acuerdo al Manual de Usuario preliminar presentado en el punto anterior, se sostuvo una junta el día 18 de junio del año 2012 en la cual participaron el Cliente, el Administrador del Proyecto y el Equipo de Desarrollo. En esta reunión se concluyó que el contenido del manual de usuario debe de ser claro, dividiendo el contenido en secciones:

- Primera Sección: Usuarios de Consulta.
- Segunda Sección: Jefes de Enseñanza.
- Tercera Sección: Administrador de Cursos de la División de Estudios de Posgrado.

De manera que se pueda realizar un Manual de Usuario de manera incremental. También se solicitó por parte del Cliente que el Manual sea lo más detallado y claro posible considerando que muchos de los Usuarios tienen conocimientos básicos sobre el uso de equipos de cómputo.

## **7.7. I.S.2.7 Preliminar de Manual de Usuario verificado**

El preliminar de Manual de Usuario aprobado se presenta en la imagen 7.11



SIDEP v.2.0



Fecha: 20 de junio del 2012  
Versión: 1.0f

Manual de Usuario (Preliminar-Aprobado)

### Índice

1. Introducción
2. Requerimientos del Sistema
3. Instalación de recursos para uso de la Aplicación
4. Uso de la Aplicación Nivel Consulta
  - a. Ingreso al sistema
  - b. Selección de ciclo y nivel
  - c. Consulta de datos y selección de filtros
  - d. Impresión de Reportes
5. Uso de la Aplicación Nivel Jefe de Enseñanza
  - a. Movimientos de Profesores
    - i. Alta
    - ii. Actualización
    - iii. Baja
    - iv. Repetición de Profesores del ciclo inmediato anterior
6. Uso de la Aplicación Nivel Admin. De Cursos de la División de Estudios de Posgrado
  - a. Movimientos de Jefes de Enseñanza
    - i. Alta
    - ii. Actualización
    - iii. Baja
  - b. Movimientos de Sedes
    - i. Alta
    - ii. Actualización
    - iii. Baja
  - c. Movimientos de Cursos
    - i. Alta
    - ii. Actualización
    - iii. Baja
7. Glosario de términos

Figura 7.11: Índice preliminar del Manual de Usuario Aprobado que se entregará con el Sistema Final.

---

## Capítulo 8

# Arquitectura y Diseño Detallado del Software.

Durante esta etapa se transforman los requerimientos de software en la Arquitectura del Sistema y en el detallado del software. De acuerdo a la norma ISO/IEC 29110 esta actividad provee de:

- Una revisión por parte de equipo de Trabajo al *Plan del Proyecto* para determinar la asignación de tareas.
- El diseño de la arquitectura del software, componentes de software y las interfaces asociadas.
- El diseño detallado de los componentes de software y sus interfaces.
- Una revisión a la *Especificación de Requerimientos* por parte del Equipo de Trabajo.
- Un diseño de software verificado y con defectos corregidos.
- *Casos y Procedimientos de Prueba* verificados para pruebas de integración.
- Un rastreo de los requerimientos en el diseño de software, casos de prueba y procedimientos de prueba.
- Productos y documentos de diseño bajo control de versiones [7].

La tabla de tareas se muestra en las figuras B.5 y B.6 contenidas en el Apéndice Implementación del Software, las tareas serán detalladas en las siguientes secciones.

### **8.1. I.S.3.1. Asignar tareas a los miembros del Equipo de Trabajo de acuerdo a su rol para el *Plan del Proyecto Actual*.**

Como ya se comentó en el capítulo anterior, el *Equipo de Desarrollo* está conformado por una sola persona así que todas las tareas recaeran sobre ésta.

### **8.2. I.S.3.2. Comprender la Especificación de Requerimientos.**

El Equipo de Desarrollo detalló un documento explicando los requerimientos de acuerdo a su percepción de los mismos, éste fue presentado ante el Administrador de Proyecto y el Cliente en la junta del día 25 de junio del 2012, en base al documento mostrado a continuación fueron detalladas las funcionalidades del sistema para que el Cliente realizara las Historias de Usuario.



SIDEP v.2.0

Fecha: 21 de junio del 2012  
Versión: 1.0f

Especificación de Requerimientos – Equipo de Desarrollo

El Cliente necesita un Sistema que le permita administrar los Profesores que imparten clase en las diversas Sedes que tienen el aval de la División de Estudios de Posgrado de la Facultad de Medicina, dicha administración la llevan a cabo dos tipos de usuario identificados:

- Administrador de Cursos de la División de Estudios de Posgrado de la Facultad de Medicina.
- Jefes de Enseñanza de las diversas Sedes registradas.

Al Administrador de Cursos de la División de Estudios de Posgrado además de permitirle la modificación de Profesores asignados a Cursos, se le debe permitir añadirles o modificar tanto sus nombramientos como su Currícula, la cual sólo será una por Académico registrado.

Además se necesita permitir la consulta e impresión de los Profesores que se encuentra asignados a un curso en una institución o sede dadas, de los Jefes de Enseñanza Institucionales o Nacionales, de los Cursos y Sedes activas en un ciclo escolar específico además de poder consultar los usuarios que tienen acceso por parte del Administrador de Cursos de la División de Estudios de Posgrado.

Se deben agregar movimientos de Administrador como son: Alta, Baja y Actualización de Sedes, Cursos y Usuarios, estos movimientos sólo estarán disponibles para el Administrador de Cursos de la División de Estudios de Posgrado así como para el/los Administradores del Sistema.

Figura 8.1: Especificación de Requerimientos - Equipo de Desarrollo.

### **8.3. I.S.3.3 Documentar o actualizar el *Diseño de Software*.**

Se comenzó definiendo los Paquetes que se utilizarán para la Arquitectura del Sistema, este diagrama permitirá identificar las agrupaciones lógicas mostrando las dependencias entre las mismas, dichos paquetes facilitan la asignación de tareas entre los miembros de un Equipo de Trabajo, el diagrama se puede ver en la figura 8.2.



SIDEP v.2.0

Fecha: 27 de julio del 2012  
Versión: 1.0f

Arquitectura con Diagramas de Paquetes

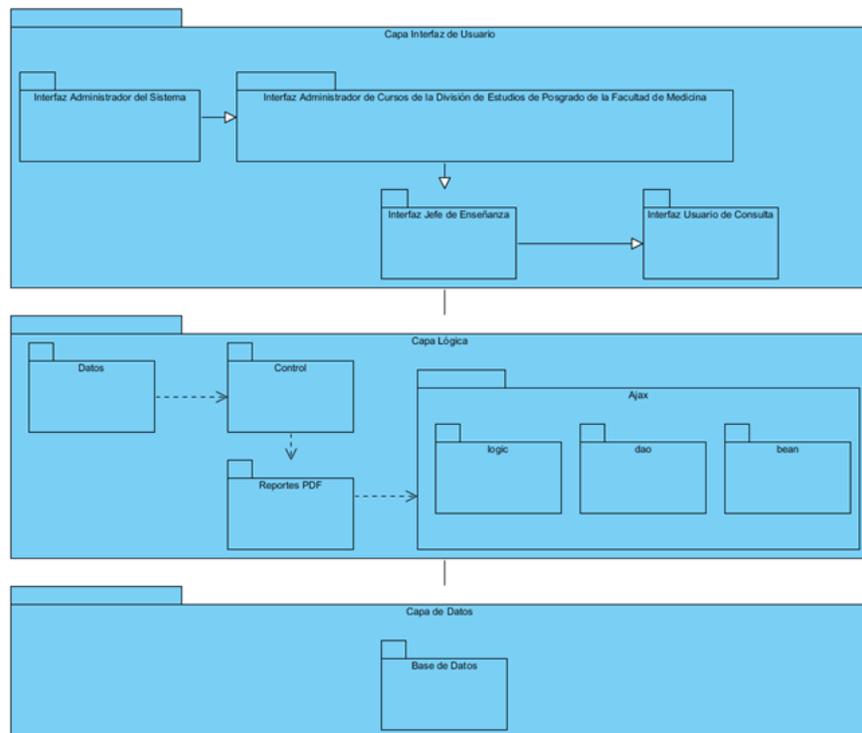
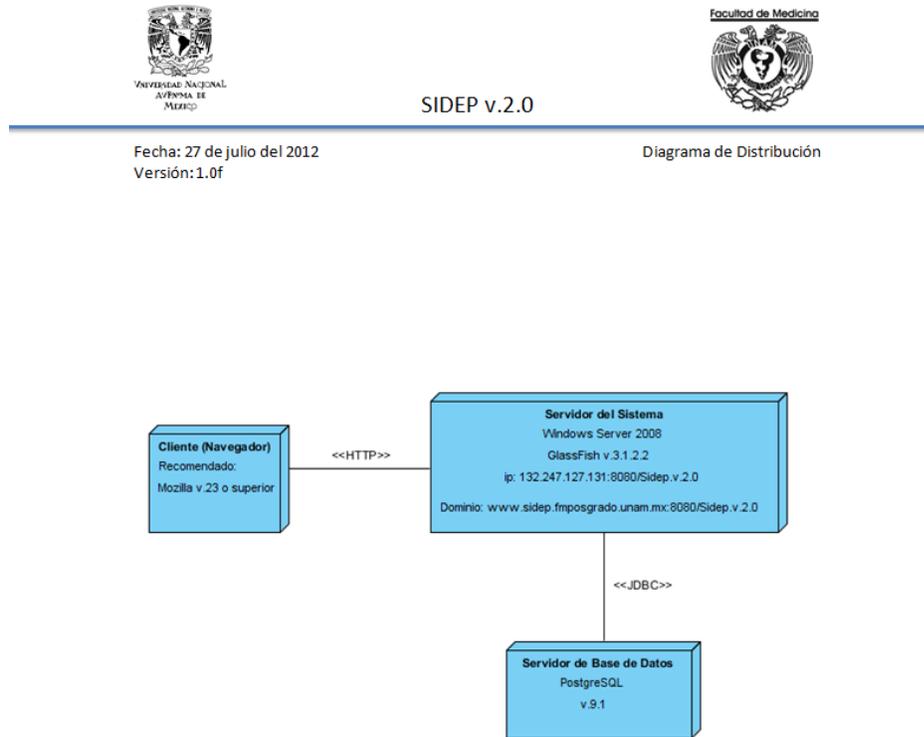


Figura 8.2: Arquitectura con Diagramas de Paquetes.

También se definió el Diagrama de Distribución el cual nos mostrará las entidades que existirán en tiempo de ejecución, especificando los componentes que contiene cada una de las entidades, podemos ver el diagrama en la figura 8.3.



1

Figura 8.3: Diagrama de Distribución.

Además el diseño del software estará dividido en dos secciones, la primera será el

diseño de la Base de Datos y la segunda el detalle de las Historias de Usuario presentadas por el Cliente y documentadas en el capítulo anterior.

- Base de Datos.

Se propone como esquema de Base de Datos un esquema relacional ya que “*las reglas de negocio*” que involucra Sidep nos permiten identificar las siguientes entidades con sus respectivos atributos, además se proponen algunas entidades a manera de catálogos los cuales almacenarán información estática:

- ⊗ Académico.

- ◇ Apellido Paterno.
- ◇ Apellido Materno (en caso de existir).
- ◇ Nombre/s.
- ◇ CURP (único).
- ◇ RFC (único).
- ◇ Teléfono.
- ◇ Correo electrónico.
- ◇ Género.

Además esta entidad generará un catálogo (cgrado) ya que cada Académico tiene un grado de estudios el cual debe almacenarse. Para respetar la normalización de Bases de Datos se extraerá el Género de la tabla Académico para convertirlo en un catálogo (cgenero), así tendremos que Académico incluirá dos llaves foráneas (nidgrado,nidgenero).

- ⊗ cgrado.

- ◇ Abreviatura de grado (Dr., Lic., Enf. , etc.).

- ⊗ cgenero.

- ◇ Abreviatura de género (F, M).

Un Académico puede convertirse en Profesor o bien en Jefe de Enseñanza, así que se modelaron ambas entidades:

- ⊗ Profesor.

- ◇ Clave de Académico Propuesto como Profesor (llave foránea hacia Académico).

- ◇ Categoría de Profesor (Titular, Adjunto). El Profesor deberá estar asignado a un Curso en una Sede en un Ciclo Escolar específico, esto nos llevará a tener el identificador de dicha relación como llave foránea en esta entidad, además para seguir normalizando la base de datos, proponemos extraer la categoría del Profesor como una nueva entidad en forma de catálogo (ccategoriaprofesor).
- ⊗ ccategoriaprofesor
  - ◇ Categoría (Titular, Adjunto)
- ⊗ Jefe de Enseñanza.
  - ◇ Cargo (detalle de puesto).
  - ◇ Categoría de Jefe de Enseñanza (Institucional, Nacional).

Los Jefes de Enseñanza tienen un detalle especial de acuerdo a la Sede o Institución a la que pertenecen, este atributo se llenará de manera libre pero por defecto tendrá el valor Jefe de Enseñanza, al ser la relación Jefe de Enseñanza - Sede o Jefe de Enseñanza - Institución única, proponemos que el identificador del Jefe de Enseñanza sea llave foránea en la Sede o en la Institución según sea el caso. Además para seguir con la normalización de la Base de Datos creamos un nuevo catálogo con las categorías de los Jefes de Enseñanza (ccategoriajefeenseñanza).

- ⊗ ccategoriajefeenseñanza
  - ◇ Categoría (Institucional, Nacional).
- ⊗ Curso
  - ◇ Clave (numérica, la ingresa el usuario).
  - ◇ Nombre.
  - ◇ Nivel (Especialidades, CPAEM)

Los Cursos pueden ser impartidos en dos niveles, reconociendo que son tuplas diferentes ya que son contenidos temáticos distintos, así generamos un nuevo catálogo (cnivel) que además de permitirnos diferenciar el nivel de los Cursos nos permitirá mantener la normalización de la Base de Datos.

- ⊗ cnivel
  - ◇ Nombre de nivel (Especialidades, CPAEM).
- ⊗ Sede
  - ◇ Clave (numérica, la ingrea el usuario).
  - ◇ Nombre.
  - ◇ Ubicación (dirección completa).

Las Sedes pertenecen a una Institución lo que indica que es posible generar un nuevo catálogo (cinstitucion), además al Cliente le interesa saber exactamente en qué estado se encuentra ubicada una Sede así que dicha información se extraerá para crear un nuevo catálogo (cestado).

- ⊗ cinstitucion
  - ◇ Nombre de Institucin.
- ⊗ cestado
  - ◇ Nombre de Estado.

Para relacionar las Sedes, Cursos y Profesores que los imparten de acuerdo a un ciclo escolar identificamos una nueva entidad a manera de catálogo (cciclo) y se propone una que permita relacionar esta información (cse-deccurso).

- ⊗ cciclo
  - ◇ Ciclo Escolar (ej: 2000-2001).
- ⊗ csedeccurso
  - ◇ Observaciones (campo libre para llenar, valor por defecto: Sin Observaciones).
  - ◇ Número de inscritos. Para lograr la relación tendremos como llaves foráneas los identificadores de Ciclo, Curso y Sede además de que la llave primaria estará asociada de manera foránea en la entidad Profesor para saber qué Profesor imparte el curso.
- ⊗ Curricula Profesor
  - ◇ Se creará un atributo por campo del formulario entregado por el Cliente.

Para lograr que un Académico tenga una sola curricula se definirá como llave foránea el identificador del Académico en esta entidad, además de tener una restricción de tipo única sobre el atributo que relaciona las entidades Curricula Profesor - Académico.

- ⊗ Nombramiento
  - ◇ Tipo de Nombramiento.
  - ◇ Nombramiento Actual.
  - ◇ Otro

Los tipos de nombramiento generarán dos nuevos catálogos, lo que permitirá mantener una Base de Datos normalizada, así los tres atributos propuestos se convierten en llaves foráneas hacia dichas entidades además se propone una llave foránea que permitirá asociar el nombramiento con un Académico añadiendo una restricción de tipo única, ya que un Académico sólo puede tener un registro de nombramiento asociado.

- ⊗ ctiponombramiento
  - ◇ Tipo de Nombramiento (3A, 5A).
- ⊗ cdetallenombramiento
  - ◇ Nombre de Nombramiento (Nombramiento en Pregrado, Nombramiento en Posgrado)
- ⊗ Sesion
  - ◇ Nombre (detalle de usuario).
  - ◇ Usuario (login).
  - ◇ Contrasea.
  - ◇ Permisos (Lectura, Lectura - Escritura, Lectura - Escritura - Administración).

La entidad Sesión permitirá crear usuarios para la División de Estudios de Posgrado y para los Jefes de Enseñanza, la manera en que se relacionará esta entidad con la entidad Jefe de Enseñanza será llevando el identificador de la tupla Sesión como llave foránea en el registro del Jefe de Enseñanza.

## ⊗ Movimiento

- ◇ Detalle de movimiento.
- ◇ Tipo de movimiento (Alta, Actualización, Baja).

Ya que se desea llevar un registro de todo movimiento que se realice en la Base de Datos, se propone la entidad Movimiento, la cual tendrá una llave foránea hacia el usuario que realizó el movimiento, detallando el día y el registro modificado en el atributo Detalle de movimiento, el tipo de movimiento se propone como una llave foránea hacia un catálogo cmovimiento.

## ⊗ cmovimiento

- ◇ Tipo de movimiento (Alta, Actualización, Baja).

Adicional a la entidad movimiento se tendrá una entidad bajaprofesor la cual permitirá al Cliente añadir comentarios a cada Profesor que es dado de baja.

## ⊗ bajaprofesor

- ◇ Motivo
- ◇ Detalles de Baja

El atributo motivo será llenado de manera libre por el usuario, el atributo Detalle de Baja deberá ser llenado por el sistema recopilando nombre del Profesor, Sede y Curso; además se tendrán dos llaves foráneas una hacia el identificador del ciclo y otra hacia el identificador del Profesor que ha sido dado de baja.

El esquema de Base de Datos se presenta en las figuras 8.4 y 8.5.





Sidep.v.2.0



Fecha: 30 de Julio del 2012  
Versión: 5.8f

Diagrama de Base de Datos

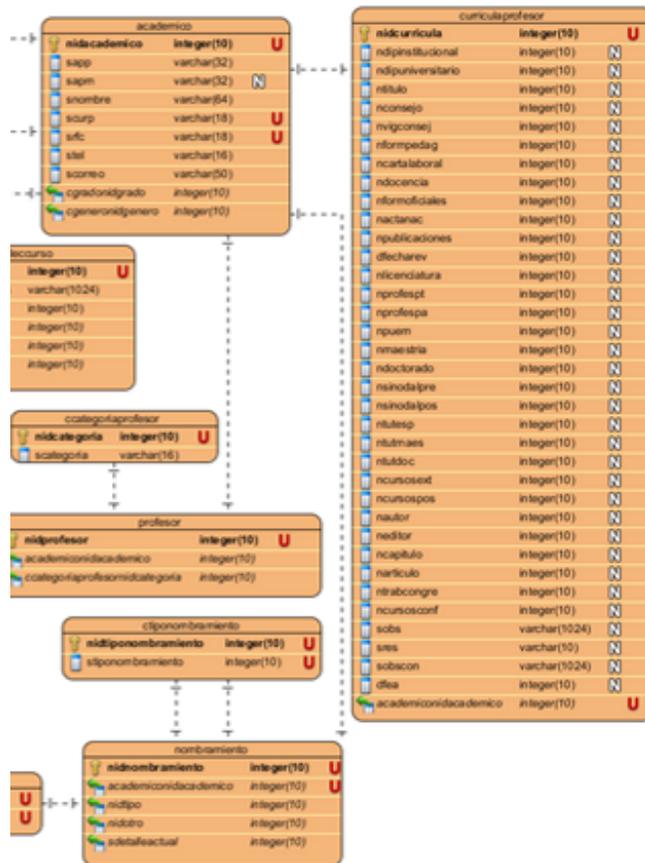


Figura 8.5: Esquema de Base de Datos.

- Historias de Usuario.

Este desarrollo está guiado por una metodología híbrida, para detallar los requerimientos se utilizaron Historias de Usuario y Casos de Uso. Durante la fase de Diseño se agregaron detalles a cada historia, lo que permitió crear un esbozo de lo que sería la implementación de las funcionalidades y posteriormente el sistema final. De esta manera el detallado de Historias de Usuario contendrá:

- ⊗ Prototipos de Interfaz validados por el Cliente.
- ⊗ Detalles de Tecnología requerida.
- ⊗ Secuencia de solución para atender la funcionalidad que el Cliente ha detallado, misma que podrá contener la explicación de:
  - ◇ Las Restricciones sobre atributos en la Base de Datos.
  - ◇ Detalle de los métodos que se deberán crear para el manejo de datos.
  - ◇ Detalle de los Procedimientos Almacenados que se utilizarán para recuperación, inserción, eliminación o actualización de datos.

Se presentan a continuación los *Detalles de Diseño* de las Historias de Usuario:

- ⊗ Alta de Profesor en las figuras 8.6, 8.7, 8.8, 8.9 y 8.10.
- ⊗ Actualización de Jefe de Enseñanza en la figura 8.11.
- ⊗ Eliminación de Profesor en la figura 8.12.
- ⊗ Consulta de Cursos en la figura 8.13.



## SIDEP v.2.0

Fecha: 23 de julio del 2012  
Versión: 2.7f

Historia de Usuario  
ID: 1 – Alta de Profesor

Como **Administrador de Cursos de la División**, necesito **guardar los datos de profesores** (nombre, RFC, CURP, correo electrónico, teléfono, género) para poder **asignarlos a uno o varios cursos**. La asignación de profesores debe cumplir las siguientes condiciones:

- Un Profesor puede ser titular sólo en un curso por ciclo escolar.
- Un Profesor puede ser adjunto en todos los cursos que se desee.
- Un Profesor sólo puede impartir un curso por sede.
- Un Profesor llamado No Propuesto, me permitirá identificar los cursos que no tienen profesor asignado, éste deberá ser registrado de manera automática como titular y adjunto al dar de alta un curso en una sede.

Además necesito que se le permita a los **Jefes de Enseñanza** de cada una de las sedes realizar las mismas acciones pero sólo con los cursos de su sede.

## Notas:

- No todos los Profesores cuentan con dos apellidos, esto porque existen Profesores de procedencia extranjera.
- Sería bueno mantener una relación de grados (Lic. Dr. Tec., etc.) para los Profesores.
- La figura del Profesor No Propuesto puede ser asignada tantas veces como sea necesario durante un ciclo.
- A los Jefes de Enseñanza se les debe permitir repetir las asignaciones de Profesores a Cursos del ciclo inmediato anterior.
- El usuario solicita tener las distintas opciones de ingreso de Profesor disponibles en un menú (Alta de Académico, Alta de Profesor como nuevo registro, Alta de Profesor como registro existente)

## Detalle de diseño:

- Para llenar los combos dependientes se utilizara la tecnología AJAX (JavaScript y XML).
- Dadas las múltiples restricciones que el cliente ha impuesto para asignación de Profesores a Cursos, hemos decidido traer los posibles datos a utilizar de la siguiente manera:

Figura 8.6: Historia de Usuario Detallada - Alta de Profesor.



SIDEP v.2.0

Fecha: 23 de julio del 2012  
 Versión: 2.7f

Historia de Usuario  
 ID: 1 – Alta de Profesor

➤ Inserción de Profesor en Sede y Curso con Académico Existente

**Alta de Profesor en Sede y Curso**  
(Actualizar sistema)

Tipo de Profesor:

Sede:

Curso:

Académico Propuesto:

- Primero se deberá seleccionar el tipo de Profesor a asignar (Titular o Adjunto)
- En segundo paso se deberá seleccionar de una lista la Sede en la que será asignado el Profesor, en caso de querer registrar un Profesor Titular sólo se mostrarán aquellas Sedes que tengan Cursos con la figura No Propuesto para el Profesor Titular.
- Para el tercer paso se mostrará una lista de cursos, sólo los cursos que estan asociados a la sede y tengan como Profesor Titular la imagen de No Propuesto en el ciclo que previamente fue seleccionado serán mostrados.
- Como último paso, se mostrará una lista de Académicos posibles a asignar como Profesores considerando que no aparecerán disponibles aquellos Profesores que ya imparten un curso en la Sede seleccionada; en caso de querer dar de alta un Profesor Titular sólo aparecerán los Académicos que no tienen asignada una titularidad en el ciclo seleccionado.

➤ Inserción de Académico como Nuevo Registro

**Alta de Profesor**  
(Nuevo Registro)

\*Todos los datos son obligatorios

Grado:  Apellido Paterno:  Apellido Materno:

Nombre/s:

CURP:  RFC:  Género:

Teléfono:  e-mail:

Figura 8.7: Historia de Usuario Detallada - Alta de Profesor.



SIDEP v.2.0

Fecha: 23 de julio del 2012  
 Versión: 2.7f

Historia de Usuario  
 ID: 1 – Alta de Profesor

- Sólo se registrará el Académico, posteriormente el Adadémico podrá ser asignado como Profesor.

➤ Inserción de Profesor en Sede y Curso como nuevo Registro

- Primero se deberá seleccionar el tipo de Profesor a asignar (Titular o Adjunto)
  - En segundo paso se deberá seleccionar de una lista la Sede en la que será asignado el Profesor, en caso de querer registrar un Profesor Titular sólo se mostrarán aquellas Sedes que tengan Cursos con la figura No Propuesto para el Profesor Titular.
  - Para el tercer paso se mostrará una lista de cursos, sólo los cursos que estan asociados a la sede y tengan como Profesor Titular la imagen de No Propuesto en el ciclo que previamente fue seleccionado serán mostrados.
  - Por último deberán ingresarse los datos del nuevo Académico, deberá llevarse a cabo una validación de RFC,CURP y correo electrónico, estos deberán ser únicos siempre en la base de datos.
- Para abarcar las 3 opciones de inserción que nos pide el usuario hemos propuesto realizar la inserción mediante procedimientos almacenados, de esta manera tendremos las funciones precargadas en la base de datos lo que se verá reflejado en la optimización del sistema, serán 3 funciones para inserción de Profesores definidas de la siguiente manera:

Figura 8.8: Historia de Usuario Detallada - Alta de Profesor.



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

SIDEP v.2.0



Facultad de Medicina

---

Fecha: 23 de julio del 2012  
Versión: 2.7f

Historia de Usuario  
ID: 1 – Alta de Profesor

- `sp_insertaProfesorAcademicoExistente(tipo,sede,curso,academico,ciclo,nivel)`  
Esta función recibe los parámetros:
  - `tipo - entero` - que nos indica el tipo de Profesor que se desea registrar (Titular o Adjunto).
  - `sede - entero` - que nos indica el id de la Sede en la que se desea registrar el nuevo Profesor.
  - `curso - entero` - que nos indica el id del Curso en que se desea registrar el nuevo Profesor.
  - `academico - entero` - que nos indica el id del Académico propuesto para ser Profesor en la Sede y Curso antes especificados.
  - `ciclo - cadena` - que nos indica el ciclo escolar en el que el Profesor será registrado en la Sede y Curso seleccionados.
  - `nivel - cadena` - que nos indica el nivel (Especialidades - CPAEM) en el cual se esta registrando el nuevo Profesor.
  
- `sp_insertaProfesorNuevoAcademico(grado,app,apm,nombre,curp,rfc,genero,tel,email)`  
Esta función recibe los parámetros:
  - `grado - entero` - que nos indica el tipo de grado que tiene el Académico que se registrará.
  - `app - cadena` - que nos indica el apellido paterno del Académico a registrar.
  - `apm - cadena` - que nos indica el apellido materno (en caso de existir) del Académico a registrar.
  - `nombre - cadena` - que nos indica el nombre/s del Académico a registrar.
  - `curp - cadena` - que nos indica el CURP del Académico a registrar.
  - `rfc - cadena` - que nos indica el RFC del Académico a registrar, deberá llevar homoclave.
  - `genero - entero` - que nos indicará el id del genero del Académico a registrar.
  - `tel - cadena` - que nos indicará el teléfono del Académico a registrar, este puede no ser único ya que se tienen Académicos que comparten línea telefónica.

Figura 8.9: Historia de Usuario Detallada - Alta de Profesor.



SIDEP v.2.0

Fecha: 23 de julio del 2012  
 Versión: 2.7f

Historia de Usuario  
 ID: 1 – Alta de Profesor

- mail - cadena - que nos indicará el correo electrónico del Académico a registrar, este debe ser único.
- sp\_insertaProfesorAcademicoNuevoSedeCurso(tipo,sede,curso,grado,app,apm, nombre, curp, rfc, genero, tel, mail, ciclo, nivel)
- Esta función recibe los parámetros:
- tipo - entero - que nos indica el tipo de Profesor que se desea registrar (Titular o Adjunto).
  - sede - entero - que nos indica el id de la Sede en la que se desea registrar el nuevo Profesor.
  - curso - entero - que nos indica el id del Curso en que se desea registrar el nuevo Profesor.
  - grado - entero - que nos indica el tipo de grado que tiene el Académico que se registrará.
  - app - cadena - que nos indica el apellido paterno del Académico a registrar.
  - apm - cadena - que nos indica el apellido materno (en caso de existir) del Académico a registrar.
  - nombre - cadena - que nos indica el nombre/s del Académico a registrar.
  - curp - cadena - que nos indica el CURP del Académico a registrar.
  - rfc - cadena - que nos indica el RFC del Académico a registrar, deberá llevar homoclave.
  - genero - entero - que nos indicará el id del genero del Académico a registrar.
  - tel - cadena - que nos indicará el teléfono del Académico a registrar, este puede no ser único ya que se tienen Académicos que comparten línea telefónica.
  - mail - cadena - que nos indicará el correo electrónico del Académico a registrar, este debe ser único.
  - ciclo - cadena - que nos indica el ciclo escolar en el que el Profesor será registrado en la Sede y Curso seleccionados.
  - nivel - cadena - que nos indica el nivel (Especialidades - CPAEM) en el cual se esta registrando el nuevo Profesor.

Figura 8.10: Historia de Usuario Detallada - Alta de Profesor.



SIDEP v.2.0



Fecha: 25 de julio del 2012  
Versión: 2.7f

Historia de Usuario  
ID: 5 – Actualización de Jefe de Enseñanza

Como **Administrador de Cursos de la División**, necesito poder **actualizar el Jefe de Enseñanza de una Sede** o de una **Institución**.

#### Notas:

- Se debe verificar que la actualización del Jefe propuesto no exceda el número de asignaciones como Jefe de Enseñanza permitidas (máximo una Institucional y una Nacional).

#### Detalle de diseño:

- Se debe mostrar un menú para poder seleccionar el tipo de actualización, las posibles combinaciones son:
  - Actualización de Académico asignado como Jefe de Enseñanza en Sede (Institucional).
  - Actualización de Académico asignado como Jefe de Enseñanza en Institución (Nacional).
- La acción de actualizar los datos personales de Jefe de Enseñanza se realizará desde el menú actualiza Académico.
- Se deben mostrar listas de selección de acuerdo al tipo de actualización:
  - Actualización de Jefe de Enseñanza en Sede: Se debe mostrar un primer menú con las Instituciones, posteriormente una lista con las Sedes pertenecientes a la Institución seleccionada en el menú anterior tras la selección de la Sede se debe mostrar el nombre del Académico que se encuentra actualmente como Jefe de Enseñanza y una lista con los posibles Académicos para sustituirlo.
  - Actualización de Jefe de Enseñanza en Institución: Se debe mostrar un primer menú con las Instituciones, en el momento en que una Institución es seleccionada se deben mostrar los datos del Académico que se encuentra actualmente como Jefe de Enseñanza y una lista con los posibles Académicos para sustituirlo.

Figura 8.11: Historia de Usuario Detallada Actualización de Jefe de Enseñanza



SIDEP v.2.0



Fecha: 25 de julio del 2012  
Versión: 1.0f

Historia de Usuario  
ID: 3 – Eliminación de Profesor

Como **Administrador de Cursos de la División**, necesito **eliminar Profesores de un Curso** específico, pero debo mantener el registro del Académico para que éste pueda ser utilizado posteriormente en otro curso, ya sea en el ciclo actual o bien en otro ciclo.

#### Notas:

- Cuando un Profesor es eliminado, automáticamente se asigna a ese lugar la imagen de Profesor No Propuesto.

#### Detalle de diseño:

- Dado que las curricula y nombramientos de los Profesores en caso de existir deben eliminarse al ser eliminado el Profesor, dichas llaves en la Base de Datos deberán tener una restricción de borrado en cascada.
- De acuerdo a la información capturada con el usuario, en caso de que un Profesor no sea asignado a un curso durante un ciclo escolar o bien sea eliminado, su curricula y nombramiento deben desaparecer, se deberá de crear un trigger que cada vez que se añada un ciclo escolar revise si el Profesor debe o no mantener su curricula y nombramiento en caso de existir o si estas deben ser eliminadas.
- Cada borrado de Profesor implica una asignación de No Propuesto de manera automática al puesto vacante.
- El menú para seleccionar al Profesor a eliminar deberá mostrar un menu de selección de sedes para posteriormente permitir seleccionar entre los cursos asignados a dicha sede alguno de estos y mostrar los profesores que se encuentran impartiendo clase en el curso y sede seleccionados, ya que de ahí se seleccionará al Profesor que será eliminado.
- El usuario pide una opción para borrar ingresando sólo el nombre o el rfc del Profesor.
- Para el borrado de Profesor se utilizará el procedimiento almacenado: `sp_eliminaProfesor(sede,curso,acad,ciclo,nivel)`, esta función recibe los parámetros:
  - sede - entero - que indica la sede de la cual se borrará el profesor
  - curso - entero - que indica el curso de la cual se borrará el profesor
  - acad - entero - que indica el identificador del Académico del cual buscaremos su identificador de profesor de acuerdo al resto de los datos que recibe el procedimiento.
  - ciclo - cadena - que indica el ciclo escolar del cual se desea eliminar al Profesor.
  - nivel - cadena - que indica el nivel del curso del cual se eliminará el Profesor.

Figura 8.12: Historia de Usuario Detallada Elimina Curso



SIDEP v.2.0



Fecha: 23 de julio del 2012  
Versión: 1.0f

Historia de Usuario  
ID: 26 – Consulta de Cursos

Como **Administrador de Cursos de la División**, necesito **consultar la información de los Cursos** que tengo registrados y que se imparten en un ciclo escolar específico. Además debo poder consultarlos de la siguiente manera:

- Todos
- Cursos por Institución
- Cursos por Institución y Sede

Además necesito **imprimir o guardar en .pdf** cada una de estas consultas.

#### Notas:

- El formato del .pdf es libre.

#### Detalle de diseño:

- La propuesta de vista de datos es:
  - En cualquiera de las consultas aparecerá el total de cursos, estos se recuperarán con el procedimiento almacenado: `sp_obtenerCursos(ciclo,nivel)`, esta función recibe los parámetros:
    - ciclo – cadena – que indica el ciclo escolar que se está consultando
    - nivel - cadena – que indica el nivel (Especialidades o CPAEM) que se consulta.
    - Los datos devueltos y que serán mostrados son:
      - Clave del Curso
      - Nombre del Curso
      - Total de Inscritos por Curso
  - Las consultas de Cursos por Institución e Institución y Sede se agruparan en una sola, la cual tendrá dos filtros mostrados en forma de una lista de selección, la primera será de instituciones, se deberá llenar con el procedimiento: `sp_obtenerInstituciones(ciclo)`, esta función recibe los parámetros:
    - ciclo - cadena - que indica el ciclo escolar que se esta consultando
    - La función sólo recuperará aquellas instituciones que tengan cursos activos en el ciclo escolar especificado.

Figura 8.13: Historia de Usuario Detallada Consulta Cursos

#### **8.4. I.S.3.4 Verificar y obtener la aprobación del *Diseño de Software*.**

La documentación presentada en la sección anterior fue presentada por el Equipo de Trabajo en la junta del día 6 de Agosto del año 2012 al Cliente y Administrador del Proyecto los cuales aprobaron el diseño propuesto así como los requerimientos detallados del sistema.

#### **8.5. I.S.3.5 Establecer los *Casos y Procedimientos de Prueba* para pruebas de integración basados en la *Especificación de Requerimientos y el Diseño de Software*.**

De acuerdo al detallado de funcionalidades, los procedimientos de prueba se harán de dos maneras:

- Para los Casos de Uso.

Se utilizarán Casos de Prueba los cuales en base a una *entrada conocida* y una *salida esperada* definirán si la funcionalidad se lleva a cabo de manera correcta o no. Los *datos de entrada* censarán una precondición y los *datos de salida* una postcondición. El ejemplo se puede ver en la figura 8.14.



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

SIDEP v.2.0

Facultad de Medicina



---

Fecha: 23 de julio del 2012  
Versión: 1.0f

Caso de Prueba  
ID: 1 –Ingresar al Sistema

**Precondición:** El usuario ingresó a la dirección [www.sidep.fmposgrado.unam.mx:8080/Sidep.v.2.0/](http://www.sidep.fmposgrado.unam.mx:8080/Sidep.v.2.0/)

**Postcondición:** El usuario será redirigido a menú.jsp con sus permisos asociados y su estado de Jefe de Enseñanza o no.

Entrada conocida	Salida esperada	Se cumple o no (Sí - No)
<b>Usuario y contraseña</b>	En caso de ser válidos los datos de entrada de cumple la postcondición, en caso contrario se debe regresar a index.jsp.	

Figura 8.14: Ejemplo:Caso de Prueba - Caso de Uso Ingresar al Sistema.

- Para las Historias de Usuario.

Se tomará la Historia de Usuario en donde el cliente indicará mediante una selección (Aprobado, Rechazado) si el módulo cumple con el requerimiento solicitado en la Historia o no. El ejemplo se puede ver en la figura 8.15.



SIDEP v.2.0



Fecha: 23 de julio del 2012  
Versión: 1.0f

Prueba - Historia de Usuario  
ID: 24 - Consulta de Profesores

Como **Administrador de Cursos de la División**, necesito **consultar la información de los Profesores** que tengo registrados en un ciclo escolar específico, debo poder consultarlos de la siguiente manera:

- Todos
- Total de Titulares
- Total de Adjuntos
- Total de No Propuestas
- Total por Sede
- Total por Sede y Curso
- Total por Curso
- Total por Curso y Sede

Estas consultas las podrán realizar los Jefes de Enseñanza con los Profesores de su respectiva Sede. Además necesito **imprimir o guardar en .pdf** cada una de estas consultas.

**Notas:**

- El formato del .pdf es libre.

**Detalle de Prueba:**

Fecha de validación:        /        /        Vo.Bo. Equipo de Trabajo

-----Especificación de prueba usuario-----

Aprobado

Rechazado

Fecha de revisión:        /        /        Vo.Bo. Dr. Agles Avelar Cruz

**Notas:**

Figura 8.15: Ejemplo: Prueba Historia de Usuario - Consulta Profesores

## **8.6. I.S.3.6 Verificar y obtener la aprobación de los *Casos y Procedimientos de Prueba*.**

En la junta sostenida el día 30 de julio del año 2012 en la que participaron el Cliente, el Administrador del Proyecto y el Equipo de Trabajo también se validaron los *Casos y Procedimientos de Prueba* presentados en la sección anterior.

## **8.7. I.S. 3.7 Actualizar el *Registro de Rastreo* / I.S.3.8 Incorporar el *Diseño de Software***

Tras la junta sostenida el día 30 de julio del año 2012, se pudo añadir la documentación generada y aprobada al *Repositorio del Proyecto* con lo cual se dió por concluida la fase de Arquitectura y Diseño Detallado del Software.

---

# Capítulo 9

## Construcción del Software

De acuerdo a la Norma ISO/IEC 29110 en este capítulo se creará el código y los datos del programa a partir del Diseño de Software. Esta actividad proporciona:

- Una revisión del *Plan del Proyecto* por parte del Equipo de Trabajo para determinar las tareas asignadas.
- Una revisión del *Diseño de Software* para determinar la secuencia de Construcción de Software.
- Los *Componentes de Software* para determinar la secuencia de Construcción de Software.
- Monitoreo entre los *Componentes de Software* y el *Diseño de Software*.

### **9.1. I.S.4.1 Asignar tareas a los miembros del Equipo de Trabajo de acuerdo al *Plan del Proyecto Actual*.**

Dado que para este desarrollo el Equipo de Trabajo consta de una sola persona la asignación de tareas es total al único integrante.

## 9.2. I.S.4.2 Entender el *Diseño de Software*.

En la junta llevada a cabo el día 6 de Agosto del año 2012 en la cual participaron el Administrador del Proyecto y el Equipo de Trabajo, éste último detalló de acuerdo a su visión la forma en que se diseñaría el sistema, siendo aprobado por el Administrador del Proyecto. El entendimiento del Diseño de Software se garantizó al desarrollar las Historias de Usuario junto con el Cliente. Cualquier duda era atendida durante las reuniones semanales directamente con el Administrador del Proyecto o, de ser necesario, se acudía con el Cliente.

## 9.3. I.S.4.3 Construir los *Componentes de Software* basados en la parte detallada del *Diseño de Software*.

Para la construcción del sistema se utilizaron las tecnologías que se detallan a continuación:

- PostgreSQL. Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y su código disponible de manera libre. Utiliza un modelo cliente-servidor. Utiliza *multiprocesos* lo que garantiza que el fallo en un proceso no afectará al resto y el sistema continuará funcionando.
- Java Server Pages (JSP). Páginas web basadas en Java que permiten el desarrollo web con contenido dinámico, tienen el aspecto de una página tradicional HTML pero contienen código Java incrustado. Cuando una página es solicitada por un usuario el código HTML pasa de manera directa a él, mientras que las porciones de código Java son ejecutadas por el servidor cuando la solicitud es recibida.
- AJAX. Acrónimo de *Asynchronous JavaScript And XML* - JavaScript asíncrono y XML, es una técnica para el desarrollo de aplicaciones interactivas. Las aplicaciones se ejecutan del lado del Cliente, es decir, en el navegador mientras que se mantiene una comunicación de manera asíncrona con el servidor de aplicación. AJAX permite realizar cambios sobre páginas web sin necesidad de

recargarlas mejorando la experiencia del usuario en interactividad, velocidad y usabilidad de la aplicación [4].

- iReport. Diseñador gráfico de código abierto. Accede a bases de datos a través de JDBC (Java Database Connectivity). Genera informes en formato .pdf, .rtf, .xml, .xls, .csv, .html, .docx, .odt o texto simple.

Para la implementación del software se eligió una arquitectura de tres capas ya que nos otorga como principal ventaja que, en caso de requerir algún cambio o bien dar mantenimiento al sistema, sólo se atacará al nivel requerido, facilitando el mantenimiento del sistema.

Para mantener la integridad de los datos así como la seguridad de la Base de Datos todas las conexiones que el sistema realizará a la misma serán implementadas con *Procedimientos Almacenados* así la capa de datos estará conformada por éstos y la clase *conexion.java*. La capa lógica será implementada en Java mientras que la capa de presentación (interfaz) será implementada con Ajax, esto permitirá realizar consultas dinámicas las cuales se actualizarán en tiempo real, cabe mencionar que los métodos y clases requeridas para el funcionamiento de AJAX serán implementados basados en una arquitectura de 3 capas.

#### **9.4. I.S.4.4 Diseñar o actualizar los casos de pruebas unitarias y aplicarlos para verificar que los *Componentes de Software* implementan la parte detallada del *Diseño de Software*.**

El Equipo de Trabajo conformado por un único integrante decidió realizar las pruebas unitarias en cada terminación de módulo, éstas se realizaban de acuerdo a la función implementada y si el Equipo de Trabajo decidía que el módulo trabajaba de manera correcta éste se daba por terminado.

## 9.5. I.S.4.5 Corregir los defectos encontrados hasta lograr una prueba exitosa.

Ya que el Equipo de Trabajo estuvo integrado por un único miembro, las correcciones se llevaron a cabo al concluir con las *Pruebas Unitarias* de cada módulo que se construía. No se permitía avanzar en la construcción en caso de que el módulo que se estaba desarrollando no funcionara de manera correcta.

## 9.6. I.S.4.6 Actualizar el *Registro de Rastreo* incorporando *Componentes de Software* construidos o modificados.

El desarrollo estuvo basado en Historias de Usuario, las cuales fueron utilizadas como *Registro de Rastreo* ya que la construcción de los módulos del sistema estuvo basada completamente en las mismas. Además el Equipo de Desarrollo llevó a cabo las pruebas unitarias basándose en las Historias de Usuario, la actualización del *Registro de Rastreo* se llevó a cabo indicando en cada una de las Historias de Usuario el módulo en el cual está reflejada dicha historia. A manera de ejemplo, en la figura 9.1, bajo la sección Registro de Rastreo, se indican las dos instancias de la secuencia en las cuales se encuentra implementada la Historia de Usuario.



SIDEP v.2.0



Fecha: 25 de julio del 2012  
Versión: 2.7f

Historia de Usuario  
ID: 5 – Actualización de Jefe de Enseñanza

Como **Administrador de Cursos de la División**, necesito poder **actualizar el Jefe de Enseñanza** de una **Sede** o de una **Institución**.

#### Notas:

- Se debe verificar que la actualización del Jefe propuesto no exceda el número de asignaciones como Jefe de Enseñanza permitidas (máximo una Institucional y una Nacional).

#### Detalle de diseño:

- Se mostrará un menú para seleccionar el tipo de actualización, las posibles combinaciones son:
  - Actualización de Académico asignado como Jefe de Enseñanza en Sede (Institucional).
  - Actualización de Académico asignado como Jefe de Enseñanza en Institución (Nacional).
- La acción de actualizar los datos personales de Jefe de Enseñanza se realizará desde el menú Actualiza Académico.
- Se mostrarán las listas de selección de acuerdo al tipo de actualización:
  - Actualización de Jefe de Enseñanza en Sede: Se deberá mostrar un primer menú con las Instituciones, posteriormente una lista con las Sedes pertenecientes a la Institución seleccionada. Tras la selección de la Sede, se deberá mostrar el nombre del Académico que se encuentra actualmente como Jefe de Enseñanza y una lista con los posibles Académicos para sustituirlo.
  - Actualización de Jefe de Enseñanza en Institución: Se deberá mostrar un primer menú con las Instituciones, en el momento en que una Institución es seleccionada se deberán mostrar los datos del Académico que se encuentra actualmente como Jefe de Enseñanza. Finalmente se mostrará una lista con los posibles Académicos para sustituirlo.

#### Registro de Rastreo:

menu\_movimiento\_jefes.jsp -> actualizacion\_jefe.jsp

1

Figura 9.1: Registro de Rastreo incorporado a una Historia de Usuario.

## **9.7. I.S.4.7 Incorporar *Componentes de Software* y *Registro de Rastreo*.**

Las versiones finales de los productos obtenidos en este capítulo se pueden encontrar en el Apéndice B: Implementación del Software.

---

## Capítulo 10

# Integración y Pruebas del Software

La actividad de Integración y Pruebas del Software asegura que los componentes de software integrados cumplan los requerimientos del software. Esta actividad provee:

- Una revisión del *Plan del Proyecto* para determinar la asignación de tareas.
- Comprensión de los *Casos de Prueba y Procedimientos* y del ambiente de integración.
- Los *Componentes de Software* integrados, defectos corregidos y resultados documentados.
- Monitoreo de requerimientos y el diseño para el producto de software integrado.
- Documentación y verificación de los manuales de usuario y de operación.
- El *Software* verificado integrado a la línea base [7].

Las actividades a desarrollar durante este capítulo han sido tomadas de la Norma ISO/IEC 29110, la tabla de tareas se puede ver en el Apéndice B: Implementación del Software en las figuras B.8 y B.9

## 10.1. I.S.5.1 Asignar tareas a los miembros del Equipo de Trabajo.

Dado que para este desarrollo el Equipo de Trabajo consta de una sola persona la asignación de tareas es total al único integrante.

## 10.2. I.S.5.2 Entender los *Casos y Procedimientos de Prueba*.

Los módulos creados en la etapa de Construcción serán probados de manera híbrida, de la siguiente manera de acuerdo a como fueron analizados y diseñados:

- Casos de Uso. Se llevarán a cabo pruebas de caja negra y caja blanca.
- Historias de Usuario. De acuerdo a las Historias de Usuario entregadas por el Cliente se construyó el sistema, para verificar que el sistema cumplía con los requisitos especificados en las mismas se llevarán a cabo pruebas con el Cliente. Los Casos de Prueba consistieron en entregar cada una de las Historias de Usuario generadas al Cliente, solicitándole que especificara en la parte posterior de la Historia la forma en que ésta se probaría. Al tener todas las pruebas especificadas, se procedió a ejecutarlas sobre cada Historia, primero de manera interna y posteriormente junto con el Usuario a modo de obtener su validación.

En la junta del día 29 de Octubre del año 2012 se llevó a cabo una reunión en la División de Estudios de Posgrado de la Facultad de Medicina, participaron el Cliente, el Administrador del Proyecto y el Equipo de Trabajo. En ella el Equipo de Trabajo explicó de manera breve, la forma en que debían llevarse a cabo las pruebas, así como el llenado de formatos para el reporte de defectos encontrados en cada Historia de Usuario.

### 10.3. I.S.5.3 Integrar el *Software* usando los *Casos y Procedimientos de Prueba* para las pruebas de integración.

La integración se llevó a cabo dentro del mismo desarrollo, al ser un Equipo de Trabajo reducido se decidió que la implementación del Software se haría de manera incremental. Siguiendo el principio ágil de Integración continua, ésta se llevó a cabo construyendo e integrando los módulos de la siguiente manera:

- index. Fue el primer módulo que se construyó.
- menu\_ciclos y salir. Se construyeron de manera paralela, el módulo menu\_ciclos definirá sobre qué datos se llevarán a cabo las consultas o bien los movimientos, el módulo salir toma todas las variables de sesión creadas durante la ejecución del sistema y las finaliza para terminar la sesión del usuario.
- Consultas. Se construyeron de la siguiente manera:
  - ⊗ Profesores. Las distintas opciones de consulta de Profesores se construyeron de la siguiente manera:
    - ◇ Todos, Titular, Adjunto, No Propuesto (en su versión total).
    - ◇ Todos, Titular, Adjunto, No Propuesto (Con filtros Sede - Especialidad).
    - ◇ Todos, Titular, Adjunto, No Propuesto (Con filtros Especialidad - Sede).
  - ⊗ Jefes de Enseñanza. El orden de construcción de las consultas para Jefes de Enseñanza fue: Todos, Institucionales, Nacionales.
  - ⊗ Cursos. El orden de construcción de las consultas para Cursos fue: Todos, Por Institución - Sede.
  - ⊗ Sedes. El orden de construcción de las consultas para Sedes fue: Todas, Por Curso - Institución, Por Institución - Curso.
  - ⊗ Nombramientos. El orden de construcción de las consultas para Nombramientos fue: Profesores con, Profesores sin.

- ⊗ Usuarios. El orden de construcción de las consultas de Usuarios fue: Todos, Jefes de Enseñanza, División.
  
- Movimientos. Se integraron en el siguiente orden:
  - ⊗ Profesores.
  - ⊗ Jefes de Enseñanza.
  - ⊗ Cursos.
  - ⊗ Sedes.
  - ⊗ Nombramientos.
  - ⊗ Usuarios

Para todos los casos, los módulos integrados fueron Alta y Actualización, en algunos casos también se integró el módulo Baja.

#### **10.4. I.S.5.4 Realizar pruebas de *Software* usando *Casos y Procedimientos de Prueba*.**

Los *Casos de Prueba* se basaron en las Historias de Usuario, se pidió al Cliente una redacción de la forma en que probaría la Historia así como una selección de aprobación o rechazo del módulo probado, se pueden ver ejemplos en las figuras 10.1 y 10.2. Para los Casos de Uso se realizaron Casos de Prueba especificando los valores de entrada y la salida que debía arrojar el sistema, en caso de realizarlo con éxito el módulo se aprobaba, en caso contrario se revisaba para corregir errores y se presentaba nuevamente al Cliente.



SIDEP v.2.0



Fecha: 29 de julio del 2012  
Versión: 1.0f

Historia de Usuario  
ID: 1 – Alta de Profesor

**Detalle de Prueba:**

Fecha de validación: 30 / 10 /12      Vo.Bo. Equipo de Trabajo ✓

Probaré que un Profesor se pueda dar de alta de dos maneras:

- Seleccionaré una categoría para el Profesor, seleccionaré una Sede, seleccionaré un Curso y finalmente seleccionaré a un Profesor registrado.
- Ingresaré los datos personales de un Académico posteriormente seleccionaré una categoría para el Profesor, una Sede y un Curso.

Aprobado

Rechazado

Fecha de revisión: 5 /11 /12      Vo.Bo. Dr. Agles Cruz Avelar

**Notas:**

Me gustaría que si elijo asignar un Profesor Titular sólo se muestren las Sedes en las cuales falta asignar Profesores de este tipo, así como en caso de querer asignar Profesores Adjuntos se muestren sólo los Profesores que no son Titulares en algún Curso durante el ciclo previamente seleccionado.

Figura 10.1: Detalle de prueba de Historia de Usuario - Alta de Profesor



SIDEP v.2.0



Fecha: 25 de julio del 2012  
Versión: 1.0f

Historia de Usuario  
ID: 3 – Eliminación de Profesor

**Detalle de Prueba:**

Fecha de validación: 30 / 10 /12 Vo.Bo. Equipo de Trabajo ✓

Probaré que un Profesor se pueda eliminar de la siguiente manera:

- Seleccionaré una Sede y un Curso, posteriormente de los Profesores mostrados eliminaré alguno de ellos.
- Seleccionaré un Profesor de una lista que contenga la información de todos ellos.
- Tras eliminar el Profesor por alguno de los métodos anteriores, verificaré que la plaza de Profesor se encuentre asignada a un No Propuesto.

Aprobado

Rechazado

Fecha de revisión: 6 /11 /12 Vo.Bo. Dr. Agles Cruz Avelar

**Notas:**

Figura 10.2: Detalle de prueba de Historia de Usuario - Eliminación de Profesor

## 10.5. I.S.5.5 Corregir los defectos encontrados.

Tras las pruebas llevas a cabo por el Cliente basadas en las Historias de Usuario, se corrigieron los errores encontrados en ellas. Posteriormente se obtuvo la validación del Cliente para estas correcciones.

## 10.6. I.S.5.6 Actualizar el *Registro de Rastreo* en caso de ser necesario.

Dado que el desarrollo se llevó a cabo con Historias de Usuario el *Registro de Rastreo* se dió a través de ellas, éstas permitieron obtener una traza de los requerimientos del Cliente los cuales fueron especificados en cada Historia de Usuario. Durante las actividades de Arquitectura y Diseño Detallado del Software se incluyeron, en algunos casos, prototipos de interfaz y los procedimientos almacenados que permitieron implementar la funcionalidad integrándola a la Base de Datos. En caso de que el sistema liberado requiriese de algún mantenimiento o cambio posterior, las Historias de Usuario, los comentarios del código y el *Manual de Mantenimiento* serán utilizados para atender dicha necesidad.

## 10.7. I.S.5.7 Documentar el *Manual de Operación* o actualizar el manual actual, en caso de ser apropiado.

El Manual de Operación sólo estará disponible para los administradores del sistema y estará compuesto, además de lo contenido en el Manual de Usuario, por las secciones mostradas en la figura 10.3 las cuales permitirán generar la información necesaria y asociada al nuevo ciclo escolar o bien eliminar todos los datos asociados a cierto ciclo escolar.



SIDEP v.2.0



Fecha: 7 de noviembre del 2012  
Versión: 1.0f

Manual de Operación

- 5. Administración de Ciclos
  - a. Alta
  - b. Eliminación

Figura 10.3: Índice del contenido especial para el Manual de Operación.

## **10.8. I.S.5.8 Verificar y obtener la aprobación del *Manual de Operación*, en caso de ser necesario. Verificar la consistencia del *Manual de Operación* con el *Software*.**

En la junta del día 5 de noviembre del año 2012 Administrador de Proyecto aprobó el contenido del índice del punto anterior, contemplando así las funcionalidades extras solicitadas para los Administradores del Sistema.

## **10.9. I.S.5.9 Documentar el *Manual de Usuario* o actualizar el actual.**

Tras la Construcción del Sistema y realizar las pruebas respectivas de manera satisfactoria, el Equipo de Trabajo actualizó el Manual de Usuario propuesto en el Capítulo 6. Análisis de Requerimientos de Software ya que el Cliente solicitó una estructura especial para éste. La nueva propuesta esta dividida de acuerdo al tipo de usuario al que va dirigido, quedando de la siguiente manera:

- De Consulta. En el se encontrarán los detalles para consulta de datos, se explicará la manera en que se deberán utilizar los filtros para la generación de datos específicos, además se añadirá la explicación de cómo se podrán generar reportes para su posterior almacenamiento o bien su impresión.
- Jefe de Enseñanza. En el se explicará la manera en que se podrán administrar los Profesores de los Cursos que se imparten en la Sede de la cual se es Jefe de Enseñanza. Además se añadirán las funciones del Usuario de Consulta.
- Administrador de Cursos de la División. En el se detallará la manera en que se podrán administrar los Cursos que se imparten en las Sedes registradas, se explicará el procedimiento para la administración de Sedes, Cursos, Académicos, Nombramientos, Currículas Profesores, Jefes de Enseñanza y Usuarios. También se añadirán las funciones de Usuario de Consulta.

El índice de los tres Manuales se puede ver en las figuras B.11, B.12 y B.13 del Apéndice B: Implementación del Software.

### **10.10. I.S.5.10 Verificar y obtener la aprobación del *Manual de Usuario*.**

En la junta del día 12 de noviembre del año 2012, el Equipo de Trabajo mostró los Índices de los diferentes Manuales de Usuario, los cuales fueron aprobados por el Cliente y el Administrador de Proyecto.

### **10.11. I.S.5.11 Incorporar los *Casos y Procedimientos de Prueba, Reporte de Pruebas, Manual de Operación y Manual de Usuario* como parte de la línea base.**

Las versiones finales de los Casos de Prueba de los Casos de Uso, las Historias de Usuario y los Manuales se encuentran en el Apéndice B: Implementación del Software.

---

# Capítulo 11

## Entrega de Productos

De acuerdo a la norma ISO/IEC 29110 esta actividad suministra el producto de software integrado al Cliente, además nos proporciona:

- Una revisión del *Plan del Proyecto* por parte del Equipo de Trabajo para determinar la asignación de tareas.
- Un manual de mantenimiento verificado.
- La entrega del producto de software y la documentación aplicable de acuerdo con las *Instrucciones de Entrega*.

La lista de tareas ha sido tomada de la Normal ISO/IEC 29110, la cual se puede encontrar en la figura B.10 en el Apéndice B: Implementación de Software.

### **11.1. I.S.6.1 Asignar tareas a los miembros del equipo de trabajo.**

El Equipo de Trabajo se constituyó por un solo integrante, dada esta situación las tareas fueron asignadas al mismo.

## 11.2. I.S.6.2 Comprender la *Configuración de Software*.

El sistema Sidep.v.2.0 es un sistema web el cual requiere para su correcto funcionamiento:

- Servidor GlassFish v.3.1.2.2.
- Manejador de Bases de Datos: PostgreSQL.v.9.1.
- Navegador Web con JavaScript integrado (recomendado: Mozilla v.23).

La información detallada de la instalación y configuración se encuentra en el Apéndice B: Implementación del Software. en la sección: Manual de Administración del Sistema.

## 11.3. I.S.6.3 Documentar el *Manual de Mantenimiento*.

El Manual de Mantenimiento estará conformado por las Historias de Usuario, comentarios en el código del sistema y scripts que generan la base de datos así como las restricciones y procedimientos almacenados en ella. Además se incluirán las instrucciones de ejecución de los mismos y algunos de los diagramas generados durante la documentación del proyecto. Así además de la estructura del proyecto generado por NetBeans y su contenido se añadirán:

- Diagrama de Paquetes.
- Diagrama de Distribución.
- Diagrama de Base de Datos.
- crea\_esquema.sql
- crea\_restricciones.sql
- crea\_sp\_consulta.sql
- crea\_sp\_insercion.sql

- crea\_sp\_actualizacion.sql
- crea\_sp\_eliminacion.sql
- crea\_sp\_general.sql
- datos.sql

El archivo datos.sql contiene los datos hasta el día de la migración de Sidep.v.1.0 a Sidep.v.2.0.

#### **11.4. I.S.6.4 Verificar y obtener la aprobación del *Manual de Mantenimiento*.**

En la junta del día 26 de noviembre del año 2012, el Equipo de Trabajo mostró al Administrador del Proyecto el *Manual de Mantenimiento* de acuerdo a la estructura detallada en la sección anterior, el mismo fue aprobado por el Administrador del Proyecto.

#### **11.5. I.S.6.5 Incorporar el *Manual de Mantenimiento* a la línea base de la *Configuración de Software*.**

El Manual de Mantenimiento se incorporó al repositorio tal como muestra el informe de Estado de la Configuración en el Apéndice B: Implementación del Software.

#### **11.6. I.S.6.6 Llevar a cabo la entrega de acuerdo a las *Instrucciones de Entrega*.**

El día 7 de diciembre del año 2012 se llevó a cabo una reunión en la que participaron el Cliente, el Administrador de Proyecto y el Equipo de Trabajo. En la misma

se llevó a cabo la entrega del Sistema obteniéndose el documento presentado a continuación en las figuras 11.1 y 11.2 misma que avala el término de manera correcta del proyecto.



## SIDEP v.2.0



Fecha: 7 de diciembre del 2012  
Versión: 1.0f

Entrega Final del Software

División de Estudios de Posgrado de la Facultad de Medicina.

De una parte, **Dr. Agles Cruz Avelar**, Administrador de Cursos de Especialidad Médica de la División de Estudios de Posgrado de la Facultad de Medicina de la Universidad Nacional Autónoma de México, en adelante el "**Cliente**".

De otra parte, **Fis. David Velázquez Portilla**, Jefe del Departamento de Cómputo y Sistemas de la División de Estudios de Posgrado de la Facultad de Medicina de la Universidad Nacional Autónoma de México, en adelante el "**Administrador de Proyecto**".

Finalmente, **Erick Orlando Matla Cruz**, Prestador de Servicio Social con número de cuenta 30402531-8 y proveniente de la Facultad de Ciencias de la Universidad Nacional Autónoma de México, en adelante el "**Equipo de Trabajo**".

En este acto el Equipo de Trabajo entrega el sistema con todos sus documentos:

- Código fuente del software y sus pruebas, arquitectura necesaria para el correcto funcionamiento mostrada en el *Diagrama de Distribución*. Entorno de control de versiones y resto de documentación.
- Manual de Mantenimiento del Software, que incluye diseño del programa, descripción de la codificación del software y guía de pruebas que garantizan el correcto funcionamiento del mismo.
- Manual de Usuario, con la información necesaria para que los usuarios utilicen correctamente la aplicación, desglosado por los diferentes perfiles de usuario.
- Guía de Instalación, con la información necesaria para la implementación de la aplicación.

El **Cliente** recibe el software Sidep.v.2.0 con toda su documentación.

El Equipo de Trabajo, el Administrador de Proyecto y el cliente dan por realizada la entrega del software pactado.



SIDEP v.2.0

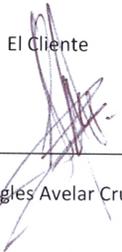
Fecha: 7 de diciembre del 2012  
Versión: 1.0f

Entrega Final del Software

Este documento se firma por duplicado y a un solo efecto.

De conformidad las tres partes:

El Cliente



---

Dr. Agles Avelar Cruz

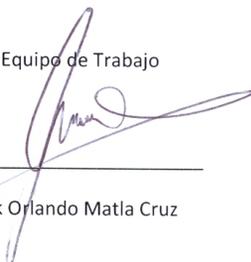
El Administrador de Proyecto



---

Fis. David Velázquez Portilla

El Equipo de Trabajo



---

Erick Orlando Matla Cruz

Figura 11.2: Documento Entrega Final del Software.

---

# Capítulo 12

## Ejecución del Plan de Proyecto

Las tareas que establece la Norma ISO/IEC 29110 para esta actividad tienen como finalidad implementar el plan documentado del proyecto. Esta actividad facilita:

- El Reporte de Avance del proyecto actualizado.
- El análisis y evaluación de los cambios solicitados al plan; y su impacto en costos, calendario de trabajo y requerimientos técnicos.
- La inclusión de los cambios aprobados al plan.
- Las revisiones y los acuerdos con el Equipo de Trabajo y el Cliente.
- El respaldo del *Repositorio del Proyecto*, el cual es recuperado en caso de ser necesario [7].

La Lista de Tareas se puede ver en el Apéndice A: Administración del Proyecto en la figura A.4.

### **12.1. A.P.2.1 Monitorear la ejecución del Plan del Proyecto y registrar la información actual en el Reporte de Avance.**

El monitoreo lo llevó a cabo el Administrador del Proyecto en cada una de las juntas semanales que se tenían, el monitoreo consistía en responder a 3 preguntas específicas tomadas de SCRUM y modificadas para el proyecto:

- ¿Qué has hecho desde la semana pasada?
- ¿Qué es lo que harás hasta la reunión de la próxima semana?
- ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo? [11]

Además del monitoreo semanal se implementó un monitoreo diario con las 3 preguntas tomadas de SCRUM:

- ¿Qué has hecho desde ayer?
- ¿Qué es lo que harás hasta la reunión de mañana?
- ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo? [11]

En algunas ocasiones tanto en las reuniones semanales como en las diarias el Administrador del Proyecto revisaba el avance del sistema pidiendo al Equipo de Trabajo que ejecutara el sistema para comprobar los avances alcanzados.

## **12.2. A.P.2.2 Analizar y evaluar el impacto en costo, tiempo e impacto técnico de la *Solicitud de Cambio*.**

El Equipo de Trabajo acordó con el Cliente y el Administrador del Proyecto que la *Solicitud de Cambio* se encontraría en todo momento disponible para poder solicitarlos en caso de ser necesario. Además el Equipo de Trabajo acordó que en caso de existir el ingreso de una *Solicitud de Cambio* ésta se analizaría con el Administrador del Proyecto para hacerle saber al Cliente el impacto que tendría en el desarrollo del proyecto.

Durante el desarrollo del Proyecto se planteó una única solicitud de cambio la cual se puede ver en la figura 12.1. En la misma se puede ver el detalle del cambio solicitado así como el impacto que ésta causó al desarrollo del proyecto.

### **12.3. A.P.2.3 Conducir reuniones de revisión con el Equipo de Trabajo.**

Estas reuniones permitieron identificar problemas, revisar el estado de los riesgos, registrar acuerdos y darles seguimiento hasta su conclusión [7]. Las reuniones como se comentó anteriormente fueron llevadas a cabo por el Administrador del Proyecto y el Equipo de Trabajo, en algunas ocasiones también participó el cliente y fueron reuniones diarias o semanales siguiendo la metodología SCRUM para agilizar el desarrollo del proyecto y dado el tamaño del Equipo de Trabajo.

### **12.4. A.P.2.4 Realizar reuniones con el Cliente.**

Las reuniones con el Cliente las llevó a cabo el Administrador del Proyecto, se aprovechó que Equipo de Desarrollo, Administrador del Proyecto y Cliente se encontraban en el mismo lugar de trabajo, por lo cual en caso de tener que resolver dudas o bien generar acuerdos el Equipo de Trabajo acudía con el Administrador del Proyecto a plantear las mismas. Posteriormente el Administrador del Proyecto acordaba una reunión breve en la cual se resolvían las dudas y tomaban acuerdos con el Equipo de Trabajo.

Además en las reuniones semanales en las cuales participó el Cliente se buscó aprovechar al máximo las mismas para aclarar dudas de manera directa entre el Equipo de Trabajo y el Cliente.



SIDEP v.2.0

Fecha: 15 de agosto del 2012  
Versión: 1.0f

Solicitud de Cambio CPAEM

Producto	Módulo CPAEM
Fecha de Solicitud	15/08/2012
Prioridad	Alta
Descripción del Cambio Solicitado	<p>Historia de Usuario:</p> <p>Si selecciono CPAEM sólo deberé poder cargar pdf's de los temarios de los Cursos que se imparten en una Sede específica, los temarios no deben ser eliminados y estos se generan cada ciclo escolar. Los Cursos tienen duración variable, no todos son de un ciclo escolar y deberá estar disponible el temario mientras el Curso se encuentre activo en la Sede.</p>
Productos Dependientes Afectados	Se afecta <b>menu.jsp</b> ya que este módulo no deberá mostrar las opciones de consulta y movimiento en caso de que el nivel seleccionado en <b>menuciclos.jsp</b> sea CPAEM.
Tiempo estimado	Medido en horas: 25 horas

Figura 12.1: Solicitud de Cambio CPAEM

## **12.5. A.P.2.5 Realizar el *Respaldo del Repositorio del Proyecto* de acuerdo a la Estrategia de Control de Versiones.**

El *Respaldo del Repositorio del Proyecto* se llevaba a cabo después de cada una de las entregas parciales con el Administrador del Proyecto y el Cliente, esto es, cada que se tenía una junta semanal en la cual participaban Equipo de Trabajo, Administrador del Proyecto y Cliente y se revisaban avances del sistema. Si la revisión era satisfactoria, se creaba la línea base y se versionaba el Repositorio del Proyecto, se le añadía un README.txt el cual contenía una breve descripción de las funcionalidades cumplidas y el estado de avance del sistema. Posteriormente se realizaba el Respaldo del Repositorio el cual se almacenaba en tres diferentes medios de almacenamiento extraíbles, dos del Equipo de Trabajo y uno del Administrador del Proyecto.

## **12.6. A.P.2.6 Realizar la recuperación del *Repositorio del Proyecto* utilizando el *Respaldo del Repositorio del Proyecto*.**

Afortunadamente durante el desarrollo de este proyecto no se requirió la restauración mediante la utilización del *Respaldo del Repositorio del Proyecto*, sin embargo para verificar que el respaldo realmente funcionaba, se hacían pruebas tras la creación de éste. Las pruebas que el Equipo de Trabajo llevó a cabo consistían en restaurar el *Respaldo del Repositorio del Proyecto* en su equipo portátil personal, la restauración se hacía en el siguiente orden:

- Esquema de Bases de Datos.
- Restricciones sobre tablas y procedimientos almacenados de la Base de Datos.
- Proyecto en NetBeans.
- Verificación de ejecución del proyecto.

- Verificación de existencia de la documentación generada hasta el momento del respaldo.

---

# Capítulo 13

## Evaluación y Control del Proyecto

De acuerdo a la Norma ISO/IEC 29110, esta actividad permite evaluar el desempeño del plan contra los compromisos previamente documentados. Esta actividad permite:

- Evaluar el desempeño actual del plan y el progreso contra los objetivos.
- Identificar y evaluar el impacto en costo, calendarización, desviaciones de desempeño y problemas.
- Revisar los riesgos del proyecto e identificar nuevos riesgos.
- Documentar solicitudes de cambio, adoptar acciones correctivas determinadas y monitorear los cambios hasta su cierre [7].

La Lista de Tareas extraída de la Norma ISO/IEEC 29110 se puede ver en la figura A.5 del Apéndice A: Administración del Proyecto.

### **13.1. A.P.3.1 Evaluar el progreso del proyecto con respecto al Plan del Proyecto.**

El progreso del proyecto se verificó de manera diaria y de manera semanal en las juntas que se tenían entre el Administrador del Proyecto y el Equipo de Trabajo, incluso en algunas de las juntas en que participó el Cliente se verificó el progreso en el desarrollo. Dicha verificación se llevó a cabo tomando en cuenta el Diagrama

de Gantt y la Lista de Tareas con tiempo y fechas especificadas para el desarrollo, presentadas en el Capítulo 4. Planeación del Proyecto, que se pueden ver en las figuras 5.18, 5.19, 5.20 y 5.21 del mismo.

De acuerdo a las revisiones llevadas a cabo se puede decir que la evaluación del progreso del proyecto es completamente satisfactoria ya que se cumplió en tiempo y forma con los productos que se especificaron de acuerdo a las fechas acordadas.

### **13.2. A.P.3.2 Establecer acciones para corregir desviaciones o problemas e identificar riesgos que amenacen el cumplimiento del plan.**

Durante el proyecto fueron detectados pocos problemas o desviaciones que afectaran el correcto desarrollo del proyecto, se listan a continuación tanto los problemas como las desviaciones que ocurrieron durante el desarrollo, además, se anexa la manera en que fueron enfrentados.

- Ausencia del Cliente.

Se aclaraban las dudas con el Administrador del Desarrollo o bien con la Secretaría del Cliente, misma que tiene contacto profundo con la lógica del negocio. En caso de que fuera necesaria la presencia del Cliente se continuaba con el desarrollo de algún otro módulo y se esperaba a que el Cliente se presentara en la División.

Nivel de ocurrencia: Bajo.

- Ausencia del Administrador del Proyecto o del Equipo de Trabajo.

Se posponía la reunión y revisión diaria para el día siguiente.

Nivel de ocurrencia: Bajo.

- Problemas de corriente eléctrica en la División.

Se recurría al trabajo en casa por parte del Equipo de Trabajo.

Nivel de ocurrencia: Bajo.

- Sobre carga de trabajo para el Equipo de Desarrollo.

El Equipo de Desarrollo re organizaba sus tiempos para cumplir con todas sus

tareas asignadas.

Nivel de ocurrencia: Medio.

### **13.3. A.P.3.3 Identificar cambios a requerimientos y/o al *Plan del Proyecto*.**

El Equipo de Trabajo, el Administrador de Proyecto y el Cliente acordaron que para solicitar un cambio se debía seguir un protocolo, dicho protocolo involucró la *Solicitud de Cambio*, presentada en el Capítulo 6. Análisis de Requerimientos de Software y que se puede ver en la figura 7.9, el proceso de identificación de cambios fué el siguiente:

- El Cliente llenaba la *Solicitud de Cambio* y la entregaba al Administrador del Proyecto.
- El Administrador del Proyecto la entregaba al Equipo de Desarrollo.
- El Equipo de Desarrollo la analizaba y estimaba tiempo.
- El Equipo de Desarrollo se reunía con el Administrador del Proyecto para hacerle saber el tiempo y costo sobre el *Plan del Proyecto*.
- En caso de ser necesario se reunían Cliente, Administrador de Proyecto y Equipo de Trabajo, en caso contrario el Administrador de Proyecto informaba al Cliente el costo y tiempo del cambio.

Durante el desarrollo del proyecto sólo se solicitó un cambio, el cual fué sobre el módulo de Administración del nivel CPAEM, dicho cambio se especificó en la *Solicitud de Cambio* que se puede ver en la figura 12.1 del Capítulo 11. Ejecución del Plan de Proyecto.

---

# Capítulo 14

## Cierre del Proyecto

De acuerdo a la Norma ISO/IEC 29110 en esta actividad se proporciona documentación y productos del proyecto de acuerdo con los requerimientos del contrato. Esta actividad nos permite:

- La entrega de productos como fueron especificados en las *Instrucciones de Entrega*.
- Contar con un soporte de aceptación del producto por parte del cliente de acuerdo a las *Instrucciones de Entrega*.
- Finalizar el proyecto y firmar el *Documento de Aceptación*.

La Lista de Tareas extraída de la Norma ISO/IEEC 29110 se puede ver en la figura A.6 del Apéndice A: Administración del Proyecto.

### **14.1. A.P.4.1 Formalizar la conclusión del proyecto de acuerdo a las *Instrucciones de entrega* establecidas en el *Plan del Proyecto*.**

El día 7 de Diciembre del año 2012 se llevó a cabo la reunión final entre el Cliente, el Administrador del Proyecto y el Equipo de Trabajo en la cual se dió por liberado el sistema Sidep.v.2.0 y concluido el desarrollo del mismo al entregarse el sistema y documentación de manera completa.

## 14.2. A.P.4.2 Actualizar el *Repositorio del Proyecto*.

Además de la entrega del sistema en la última reunión llevada a cabo el día 7 de Diciembre del año 2012, el Equipo de Trabajo entregó el *Repositorio del Proyecto Actualizado*, éste incluye:

- Proyecto de NetBeans con todo el código fuente, comentarios y .war generado.
- Repositorio de Documentación del Proyecto estructurado de acuerdo a la *Especificación del Repositorio del Proyecto* presentada en el Capítulo 4. Planeación del Proyecto y que puede verse en la figura 5.25 del mismo.
- Documentos versionados de acuerdo al *Control de Versiones* presentado en el Capítulo 4. Planeación del Proyecto y que puede verse en la figura 5.23 del mismo.

---

# Capítulo 15

## Resultados

Como resultado del proceso híbrido que se llevó a cabo durante el desarrollo de este proyecto se obtuvo el sistema Sidep.v.2.0 que se puede ver en la siguiente dirección:

*<http://www.sidep.fmposgrado.unam.mx/sidep.v.2.0>*

Mismo que tras el análisis que se llevó a cabo de su predecesor *Sidep v.1.0*, corrige los errores ahí encontrados de la siguiente manera:

- Se creó un esquema relacional de Base de Datos mismo que sustituye a un esquema que intentaba ser relacional pero no contenía llaves primarias ni foráneas, entre otras restricciones de integridad.
- Se creó un esquema de Base de Datos normalizado en Tercera Forma Normal ya que el esquema anterior redundaba en información, además de tener tablas y atributos innecesarios que incluso no se utilizaban.
- Se creó una aplicación para el manejo de la información de la base de datos empleando como tecnología base de desarrollo JAVA y AJAX, misma que sustituye al antiguo sistema desarrollado en PHP y en el cual se encontraron problemas como ligas rotas o formularios perdidos.

En la figura 15.1 se puede ver la pantalla de inicio de sesión del producto generado, mismo que es compatible con los navegadores Mozilla Firefox v.24, Opera v.17.0, Google Chrome v.30.0 e Internet Explorer 10.

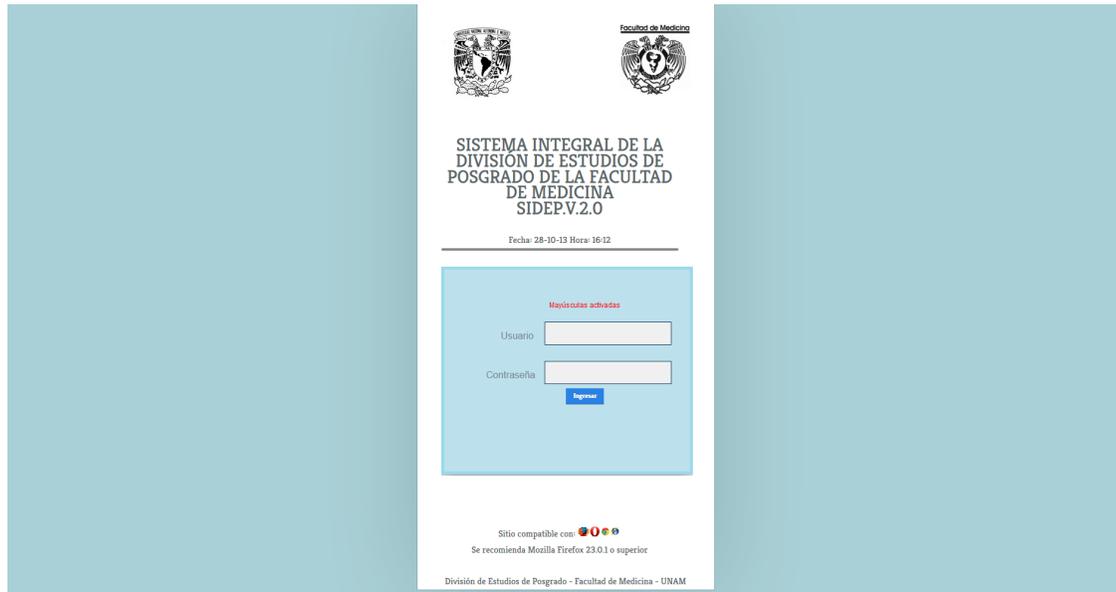


Figura 15.1: Pantalla principal de Sidep.v.2.0

Sidep.v.2.0 contiene un reporteador que genera documentos en formato PDF implementado con iReport el cual permite la impresión de consultas de datos, implementadas con AJAX, o bien el almacenamiento de los reportes generados, un ejemplo de un reporte generado se puede ver en la figura 15.2.

El módulo de movimientos mantiene la integridad de los datos al permitir realizar altas, bajas y actualizaciones de datos de la entidad seleccionada, ya que al recuperar los datos de la base de datos los *procedimientos almacenados* creados en ésta no permiten movimientos considerados como inválidos para la lógica del negocio.

En cuanto al Desarrollo del Proyecto se puede decir que emplear una metodología híbrida permitió obtener un mejor producto al involucrar de manera más directa al Cliente en el desarrollo del mismo logrando cubrir sus necesidades satisfactoriamente. Además se atacó la problemática de tener un Equipo de Trabajo reducido, permitiendo resolver dudas y mostrar avances de manera diaria al aprovechar que tanto Cliente, Administrador del Proyecto y Equipo de Trabajo se encontraban en el mismo lugar.

		UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE MEDICINA DIVISIÓN DE ESTUDIOS DE POSGRADO			
Sedes 2012-2013					
CLAVE	TOTAL DE CURSOS	TOTAL DE INSCRITOS	SEDE	UBICACIÓN	
504	1	0	ASOCIACIÓN PARA EVITAR LA CEGUERA EN MÉXICO	504	7 ASOCIACIÓN PARA EVITAR LA CEGUERA EN MÉXICO CALLE VICENTE GARCÍA TORRES NO. 46 COL. BARRIO SAN LUCAS COYOACÁN  MÉXICO, D.F. 04030 1084-1404, CONM. 1084-1400 EXT. 1085 Y 1100  http://www.apec.org.mx/
529	1	0	CENTRO DE REHABILITACION INFANTIL TELETON, SISTEMA DE CENTROS DE REHABILITACION TELETON, FUNDACION TELETON MEXICO, A.C.	529	7 CENTRO DE REHABILITACION INFANTIL TELETON, SISTEMA DE CENTROS DE REHABILITACION TELETON, FUNDACION TELETON MEXICO, A.C. AV. GUSTAVO BAZ NO. 219 COL. SAN PEDRO BARRIENTOS TLANEPANTLA EDO. DE MÉXICO 5317-2530 5321-2220
324	2	0	CENTRO DE SALUD DR. JOSÉ CASTRO VILLAGRANA	324	9 CENTRO DE SALUD DR. JOSÉ CASTRO VILLAGRANA CALLE CARRASCO ESQ. CON AV. COAPA COL. TORIELLO GUERRA TLALPAN MÉXICO, D.F. 14050 5606-6516, 5606-3516 5528-1173  http://www.salud.df.gob.mx/
310	2	0	CENTRO DERMATOLÓGICO DR. LADISLAO DE LA PASCUA	310	9 CENTRO DERMATOLÓGICO DR. LADISLAO DE LA PASCUA AV. DR. VÉRTIZ NO. 464, ESQ. AV. CENTRAL

Figura 15.2: Ejemplo de Reporte generado en Sidep.v.2.0

---

# Capítulo 16

## Conclusiones

Tras el desarrollo de este proyecto se puede reafirmar la importancia que hoy tiene la Ingeniería de Software para el desarrollo de Software y Sistemas Computacionales, permitiendo la abstracción de la lógica de un negocio no sólo para un Programador o Equipo de Trabajo (*Métodos Tradicionales*) sino que es posible involucrar de manera directa al Cliente (*Métodos Ágiles*) con lo cual se puede obtener un producto más cercano a las necesidades de éste, y en consecuencia maximizando los beneficios para él.

Además el trabajo permite observar que la aplicación de una metodología híbrida permite aprovechar de mejor manera los beneficios de las metodologías involucradas, culminando el proyecto con un producto de mejor calidad ya que se involucra al Cliente en el desarrollo y permite un mejor entendimiento con el Equipo de Trabajo, reflejando en éste un aprendizaje y satisfacción para los involucrados.

Como Equipo de Trabajo puedo decir que el desarrollo cumplió con el objetivo fijado al principio del proyecto que era la corrección de los errores que afectaban a Sidep.v.1.0 y que impactaban en el desarrollo de las actividades administrativas y académicas de la División de Estudios de Posgrado de la Facultad de Medicina.

De manera personal me permitió aprender nuevas tecnologías (AJAX, iReport), además de involucrarme en un ámbito que era completamente ajeno a mí como lo es la medicina, si bien no soy ni seré médico tras el desarrollo de este proyecto, si conocí cosas del sector salud de las cuales no tenía idea que existieran. Remarcando así el amplio campo de aplicación que actualmente tiene mi carrera.

---

# Apéndice A

## Administración del Proyecto

Aquí se encuentran las Tablas de Tareas extraídas de la Norma ISO/IEC 29110[7] de los capítulos:

- Capítulo 4. Planeación del Proyecto.
- Capítulo 11. Ejecución del Plan de Proyecto.
- Capítulo 12. Evaluación y Control del Proyecto.
- Capítulo 13. Cierre del Proyecto.

Además de documentación en versiones finales generada en los mismos. La escala de colores con la cual han sido coloreadas las tablas de acuerdo al tipo de desarrollo llevado a cabo se presenta en la figura A.1, que indica de izquierda a derecha que el desarrollo fue ágil, tradicional o híbrido respectivamente.



Figura A.1: Escala de colores para las Tablas de Tareas.

Table 6 — PM.1 task list

Role	Task List	Input Products	Output Products
PM TL	PM.1.1 Review the <i>Statement of Work</i>	<i>Statement of Work</i>	<i>Statement of Work [reviewed]</i>
PM CUS	PM.1.2 Define with the Customer the <i>Delivery Instructions</i> of each one of the deliverables specified in the <i>Statement of Work</i> .	<i>Statement of Work [reviewed]</i>	<i>Project Plan - Delivery Instructions</i>
PM TL	PM.1.3 Identify the specific tasks to be performed in order to produce the deliverables and their software components identified in the <i>Statement of Work</i> . Include tasks in the SI process along with verification, validation and reviews with Customer and Work Team tasks to assure the quality of work products. Identify the tasks to perform the <i>Delivery Instructions</i> . Document the <i>Tasks</i> .	<i>Statement of Work [reviewed]</i>	<i>Project Plan -Tasks</i>
PM TL	PM.1.4 Establish the <i>Estimated Duration</i> to perform each task.	<i>Tasks</i>	<i>Project Plan -Estimated Duration</i>
PM TL	PM.1.5 Identify and document the resources: human, material, equipment and tools, standards, including the required training of the Work Team to perform the project. Include in the schedule the dates when resources and training will be needed.	<i>Statement of Work</i>	<i>Project Plan -Resources</i>
PM TL	PM.1.6 Establish the <i>Composition of Work Team</i> assigning roles and responsibilities according to the <i>Resources</i> .	<i>Resources</i>	<i>Project Plan -Composition of Work Team</i>
PM TL	PM.1.7 Assign estimated start and completion dates to each one of the tasks in order to create the <i>Schedule of the Project Tasks</i> taking into account the assigned resources, sequence and dependency of the tasks.	<i>Tasks</i> <i>Estimated Duration</i> <i>Composition of Work Team</i>	<i>Project Plan -Schedule of the Project Tasks</i>
PM	PM.1.8 Calculate and document the project <i>Estimated Effort and Cost</i> .	<i>Schedule of the Project Tasks</i>	<i>Project Plan -Estimated Effort</i>

Figura A.2: Tabla 6 de la norma ISO/IEC 29110 [7] perteneciente al Capítulo 4. Planeación del Proyecto.

Role	Task List	Input Products	Output Products
		<i>Resources</i>	<i>and Cost</i>
PM TL	PM.1.9 Identify and document the risks which may affect the project.	<i>All elements previously defined</i>	<i>Project Plan</i> <i>-Identification of Project Risks</i>
PM TL	PM.1.10 Document the <i>Version Control Strategy</i> in the <i>Project Plan</i> .		<i>Project Plan</i> <i>-Version Control Strategy</i>
PM	PM.1.11 Generate the <i>Project Plan</i> integrating the elements previously identified and documented.	<i>Tasks</i> <i>Estimated Duration</i> <i>Resources</i> <i>Composition of Work Team</i> <i>Schedule of the Project Task</i> <i>Estimated Effort and Cost</i> <i>Identification of Project Risks</i> <i>Version Control Strategy</i> <i>Delivery Instructions</i>	<i>Project Plan</i>
PM TL	PM.1.12 Include product description, scope, objectives and deliverables in the <i>Project Plan</i> .	<i>Statement of Work</i> <i>(Product Description, Scope, Objectives and Deliverables)</i>	<i>Project Plan</i>
PM TL	PM.1.13 Verify and obtain approval of the <i>Project Plan</i> .  Verify that all <i>Project Plan</i> elements are viable and consistent. The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by PM.	<i>Project Plan</i>	<i>Verification Results</i> <i>Project Plan [verified]</i>
PM CUS	PM.1.14 Review and accept the <i>Project Plan</i> .  Customer reviews and accepts the <i>Project Plan</i> , making sure that the <i>Project Plan</i> elements match with the <i>Statement of Work</i> .	<i>Project Plan [verified]</i>	<i>Meeting Record</i> <i>Project Plan [accepted]</i>
PM TL	PM.1.15 Establish the project repository using the <i>Version Control Strategy</i> .	<i>Version Control Strategy</i>	<i>Project Repository</i>

Figura A.3: Tabla 6 de la norma ISO/IEC 29110 [7] perteneciente al Capítulo 4. Planeación del Proyecto.

Table 7 — PM.2 task list

Role	Task List	Input Products	Output Products
PM TL WT	PM.2.1 Monitor the <i>Project Plan</i> execution and record actual data in <i>Progress Status Record</i> .	<i>Project Plan</i>	<i>Progress Status Record</i>
PM TL	PM 2.2 Analyse and evaluate the Change Request for cost, schedule and technical impact.  The <i>Change Request</i> can be initiated externally by the Customer or internally by the Work Team. Update the <i>Project Plan</i> , if the accepted change does not affect agreements with Customer.  <i>Change Request</i> , which affects those agreements, needs to be negotiated by both parties (see PM.2.4).	<i>Change Request[initiated]</i>  <i>Project Plan</i>	<i>Change Request[evaluated]</i>  <i>Project Plan[updated]</i>
PM TL WT	PM.2.3 Conduct revision meetings with the Work Team, identify problems, review risk status, record agreements and track them to closure	<i>Project Plan</i>  <i>Progress Status Record</i>  <i>Correction Register</i>  <i>Meeting Record</i>	<i>Meeting Record [updated]</i>
PM CUS TL WT	PM.2.4 Conduct revision meetings with the Customer, record agreements and track them to closure.  <i>Change Request</i> initiated by Customer or initiated by Work Team, which affects the Customer, needs to be negotiated to reach acceptance of both parties.  If necessary, update the <i>Project Plan</i> according to new agreement with Customer.	<i>Project Plan</i>  <i>Progress Status Record</i>  <i>Change Request[evaluated]</i>  <i>Meeting Record</i>	<i>Meeting Record [updated]</i>  <i>Change Request[accepted]</i>  <i>Project Plan[updated]</i>
PM	PM.2.5 Perform backup according to the <i>Version Control Strategy</i> .	<i>Version Control Strategy</i>	<i>Project Repository Backup</i>
PM	PM.2.6 Perform <i>Project Repository</i> recovery using the <i>Project Repository Backup</i> , if necessary.	<i>Project Repository Backup</i>	<i>Project Repository [recovered]</i>

Figura A.4: Tabla 7 de la norma ISO/IEC 29110 [7] perteneciente al Capítulo 11. Ejecución del Plan de Proyecto.

Table 8 — PM.3 task list

Role	Task List	Input Products	Output Products
PM TL WT	PM.3.1 Evaluate project progress with respect to the <i>Project Plan</i> , comparing: <ul style="list-style-type: none"> <li>- actual tasks against planned tasks</li> <li>- actual results against established project objectives</li> <li>- actual resource allocation against planned resources</li> <li>- actual cost against budget estimates</li> <li>- actual time against planned schedule</li> <li>- actual risk against previously identified</li> </ul>	<i>Project Plan</i> <i>Progress Status Record</i>	<i>Progress Status Record [evaluated]</i>
PM TL WT	PM.3.2 Establish actions to correct deviations or problems and identified risks concerning the accomplishment of the plan, as needed, document them in <i>Correction Register</i> and track them to closure.	<i>Progress Status Record [evaluated]</i>	<i>Correction Register</i>
PM TL WT	PM.3.3 Identify changes to requirements and/or <i>Project Plan</i> to address major deviations, potential risks or problems concerning the accomplishment of the plan, document them in <i>Change Request</i> and track them to closure.	<i>Progress Status Record [evaluated]</i>	<i>Change Request [initiated]</i>

Figura A.5: Tabla 8 de la norma ISO/IEC 29110 [7] perteneciente al Capítulo 12. Evaluación y Control del Proyecto.

Table 9 — PM.4 task list

Role	Task List	Input	Output
PM CUS	PM.4.1. Formalize the completion of the project according to the <i>Delivery Instructions</i> established in the <i>Project Plan</i> , providing acceptance support and getting the <i>Acceptance Record</i> signed.	<i>Project Plan</i> <i>Software Configuration [delivered]</i>	<i>Acceptance Record</i> <i>Software Configuration [accepted]</i>
PM	PM.4.2 Update <i>Project Repository</i> .	<i>Software Configuration [accepted]</i> <i>Project Repository</i>	<i>Project Repository [updated]</i>

Figura A.6: Tabla 9 de la norma ISO/IEC 29110 [7] perteneciente al Capítulo 13. Cierre del Proyecto.



SIDEP v.2.0

Fecha: 6 de junio del 2012  
Versión: 1.0f

Plan del Proyecto

Este documento especifica la manera en que se trabajará el desarrollo del sistema Sidep.v.2.0, se incluye la documentación en el repositorio la cual indica las siguientes reglas del desarrollo:

- Un estándar de documentación y plantillas para documentación los cuales se encuentran definidos en los documentos:
  - plantillageneral\_1.0f.pdf
  - plantillaproductbacklog\_1.0f.pdf
  - plantillasprintplanningmeeting\_1.0f.pdf
  - plantilladailyscrumbacklog\_1.0f.pdf
  - plantillalistatareas\_1.0f.pdf
- El *Enunciado de Trabajo, Material y Herramientas Requeridos* para el desarrollo, *Estándares* para el desarrollo y el *Equipo de Trabajo* definidos en:
  - enunciadodetrabajo\_1.0f.pdf
  - materialyherramientas\_1.0f.pdf
  - estandarcodificacion\_1.0ff.pdf
  - equipo\_1.0f.pdf
- Las tareas a desarrollar durante el proyecto se encuentran definidas en *listatareas\_1.0f.pdf*, así como el calendario para cumplir dichas tareas se encuentra especificado en *gantt\_1.0f.pdf*.

De acuerdo a estos documentos se llevará a cabo el desarrollo del proyecto.

Figura A.7: Plan del Proyecto.



## SIDEP v.2.0

Fecha: 8 de junio del 2012  
Versión: 1.0f

Especificación de Requerimientos

El sistema facilitará la administración de las sedes, cursos, profesores y jefes de enseñanza que la División de Estudios de Posgrado de la Facultad de Medicina tiene a su cargo y que se actualizan cada ciclo escolar (julio-junio). La administración consiste en:

- Permitir el registro de nuevas sedes, cursos y profesores evitando repetir información de cualquiera de ellos. También deberá permitir el registro de nuevos Jefes de Enseñanza sin embargo estos se asignan a cada una de las sedes por lo que en caso de existir alguno se deberá preguntar si se desea reemplazar o bien sustituir, para el segundo, se debe mantener el registro del profesor asignado y sólo quitarlo del puesto de Jefe de Enseñanza de la Sede previamente seleccionada.
- Permitir la baja de cursos eliminando toda relación con sedes y profesores, sin embargo se debe mantener la información personal y académica del profesor.
- Permitir la baja de sedes eliminando toda relación con cursos y profesores, se debe mantener la información personal y académica del profesor.
- Permitir asignar profesores a un curso que se imparte en una sede específica para un ciclo escolar determinado, además, en el caso de que sea un nuevo ciclo escolar se deberá permitir la repetición de los profesores asignados el semestre inmediato anterior al curso en la sede seleccionada.
- Permitir la consulta de profesores totales, por sede, por especialidad y por sede y especialidad.
- Permitir la consulta de cursos por sede y de sedes por cursos.
- Permitir el registro de información curricular y nombramiento de un profesor seleccionado así como permitir la actualización de la misma.
- Permitir el acceso a Jefes de Enseñanza para modificación de los profesores asignados en los cursos de **LA SEDE** en la que se encuentra a cargo el Jefe de Enseñanza.
- Permitir el acceso a consulta de información a usuarios de la División del Posgrado de Medicina.
- Permitir el acceso **TOTAL** al administrador de los cursos de la División del Posgrado de Medicina.

Figura A.8: Especificación de Requerimientos del Sistema.

---

# Apéndice B

## Implementación del Software

Aquí se encuentran las Tablas de Tareas extraídas de la Norma ISO/IEC 29110[7] de los capítulos:

- Capítulo 5. Inicio de la Implementación de Software.
- Capítulo 6. Análisis de Requerimientos.
- Capítulo 7. Arquitectura y Diseño Detallado del Software.
- Capítulo 8. Construcción del Software.
- Capítulo 9. Integración y Pruebas del Software.
- Capítulo 10. Entrega de Productos.

Además de documentación en versiones finales generada en los mismos. La escala de colores con la cual han sido coloreadas las tablas de acuerdo al tipo de desarrollo llevado a cabo se presenta en la figura B.1, que indica de izquierda a derecha que el desarrollo fue ágil, tradicional o híbrido respectivamente.



Figura B.1: Escala de colores para las Tablas de Tareas.

Table 15 — SI.1 task list

Role	Task List	Input Products	Output Products
PM TL WT	SI.1.1 Revision of the current <i>Project Plan</i> with the Work Team members in order to achieve a common understanding and get their engagement with the project.	<i>Project Plan</i>	<i>Project Plan[reviewed]</i>
TL WT	SI.1.2 Set or update the implementation environment.	<i>Project Plan [reviewed]</i>	

Figura B.2: Tabla 15 de la norma ISO/IEC 29110 [7] perteneciente al Capítulo 5. Inicio de la Implementación de Software.

Table 16 — SI.2 task list

Role	Task List	Input Products	Output Products
TL WT	SI.2.1 Assign tasks to the Work Team members in accordance with their role, based on the current <i>Project Plan</i> .	<i>Project Plan</i> [reviewed]	
AN CUS	SI.2.2 Document or update the <i>Requirements Specification</i> .  Identify and consult information sources (customer, users, previous systems, documents, etc.) in order to get new requirements.  Analyze the identified requirements to determine the scope and feasibility.  Generate or update the <i>Requirements Specification</i> .	<i>Project Plan (Product Description)</i>	<i>Requirements Specification</i>
AN	SI.2.3 Verify and obtain approval of the <i>Requirements Specification</i> .  Verify the correctness and testability of the <i>Requirements Specification</i> and its consistency with the <i>Product Description</i> . Additionally, review that requirements are complete, unambiguous and not contradictory. The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by AN. If significant changes were needed, initiate a <i>Change Request</i> .	<i>Requirements Specification</i> <i>Project Plan (Product Description)</i>	<i>Verification Results</i> <i>Requirements Specification [verified]</i> <i>Change Request [initiated]</i>
CUS AN	SI.2.4 Validate and obtain approval of the <i>Requirements Specification</i> .  Validate that <i>Requirements Specification</i> satisfies needs and agreed upon expectations, including the user interface usability. The results found are documented in a <i>Validation Results</i> and corrections are made until the document is approved by the CUS.	<i>Requirements Specification [verified]</i>	<i>Validation Results</i> <i>Requirements Specification [validated]</i>
AN	SI.2.5 Document the preliminary version of the <i>Software User Documentation</i> or update the present manual.  (optional)	<i>Requirements Specification [validated]</i>	<i>Software User Documentation [preliminary]</i>

Figura B.3: Tabla 16 de la norma ISO/IEC 29110 [7] perteneciente al Capítulo 6. Análisis de Requerimientos.

Role	Task List	Input Products	Output Products
AN	<p>SI.2.6 Verify and obtain approval of the <i>Software User Documentation</i></p> <p>Verify consistency of the <i>Software User Documentation</i> with the <i>Requirement Specification</i>. The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by AN. If significant changes were needed, initiate a <i>Change Request</i>.</p> <p>(optional)</p>	<p><i>Software User Documentation</i> [preliminary]</p> <p><i>Requirement Specification</i></p>	<p><i>Verification Results</i></p> <p><i>Software User Documentation</i> [preliminary, verified]</p> <p><i>Change Request</i> [initiated]</p>
TL	<p>SI.2.7 Incorporate the <i>Requirements Specification</i>, and *<i>Software User Documentation</i> to the <i>Software Configuration</i> in the baseline.</p> <p>*(optional)</p>	<p><i>Requirements Specification</i> [validated]</p> <p>*<i>Software User Documentation</i> [preliminary, verified]</p>	<p><i>Software Configuration</i></p> <p><i>Requirements Specification</i> [validated, baselined],</p> <p>*<i>Software User Documentation</i>[preliminary, verified, baselined]</p>

Figura B.4: Tabla 16 de la norma ISO/IEC 29110 [7] perteneciente al Captulo 6. Análisis de Requerimientos.

Table 17 — SI.3 task list

Role	Task List	Input Products	Output Products
TL AN DES	SI.3.1 Assign tasks to the Work Team members related to their role according to the current <i>Project Plan</i> .	<i>Project Plan</i>	
AN DES	SI.3.2 Understand <i>Requirements Specifications</i> .	<i>Requirements Specification [validated, baselined]</i>	
AN DES	SI.3.3 Document or update the <i>Software Design</i> :  Analyze the <i>Requirements Specification</i> to generate the architectural design, its arrangement in subsystems and software components defining the internal and external interfaces. Describe in detail, the appearance and the behaviour of the interface, based on the <i>Requirements Specification</i> in a way that resources for its implementation can be foreseen.  Provide the detail of software components and their interfaces to allow the construction in an evident way.  Generate or update the <i>Traceability Record</i> .	<i>Requirements Specification [validated, baselined]</i>	<i>Software Design</i> <i>Traceability Record</i>
AN DES	SI.3.4 Verify and obtain approval of the <i>Software Design</i>  Verify correctness of <i>Software Design</i> documentation, its feasibility and consistency with their <i>Requirement Specification</i> . Verify that the <i>Traceability Record</i> contains the adequate relationships between requirements and the <i>Software Design</i> elements. The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by AN. If significant changes were needed, initiate a <i>Change Request</i> .	<i>Software Design</i> <i>Traceability Record</i> <i>Requirement Specification [validated, baselined]</i>	<i>Verification Results</i> <i>Software Design [verified]</i> <i>Traceability Record [verified]</i> <i>Change Request [initiated]</i> .
DES	SI.3.5 Establish or update <i>Test Cases and Test Procedures</i> for integration testing based on <i>Requirements Specification</i> and <i>Software Design</i> .  Customer provides testing data, if needed.	<i>Requirements Specification [validated, baselined]</i>  <i>Software Design [verified, baselined]</i>	<i>Test Cases and Test Procedures</i>

Figura B.5: Tabla 17 de la norma ISO/IEC 29110 [7] perteneciente al Capítulo 7. Arquitectura y Diseño Detallado del Software.

Role	Task List	Input Products	Output Products
DES AN	SI.3.6 Verify and obtain approval of the <i>Test Cases and Test Procedures</i> .  Verify consistency among <i>Requirements Specification, Software Design and Test Cases and Test Procedures</i> . The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by AN.	<i>Test Cases and Test Procedures</i> <i>Requirements Specification [validated, baselined]</i> <i>Software Design [verified, baselined]</i>	<i>Verification Results</i> <i>Test Cases and Test Procedures [verified]</i>
DES	SI.3.7 Update the <i>Traceability Record</i> incorporating the <i>Test Cases and Test Procedures</i> .	<i>Test Cases and Test Procedures [verified]</i> <i>Traceability Record [updated]</i>	<i>Traceability Record [updated]</i>
TL	SI.3.8 Incorporate the <i>Software Design, and Traceability Record</i> to the <i>Software Configuration</i> as part of the baseline.  Incorporate the <i>Test Cases, and Test Procedures</i> to the <i>Project Repository</i> .	<i>Software Design [verified]</i> <i>Test Cases and Test Procedures [verified]</i> <i>Traceability Record [verified]</i>	<i>Software Configuration</i> <i>Software Design [verified, baselined]</i> <i>Test Cases and Test Procedures [verified]</i> <i>Traceability Record [verified, baselined]</i>

Figura B.6: Tabla 17 de la norma ISO/IEC 29110 [7] perteneciente al Capítulo 7. Arquitectura y Diseño Detallado del Software.

Table 18 — SI.4 task list

Role	Task List	Input Products	Output Products
TL PR	SI.4.1 Assign tasks to the Work Team members related to their role, according to the current <i>Project Plan</i> .	<i>Project Plan</i>	
PR	SI.4.2 Understand <i>Software Design</i> .	<i>Software Design</i> [verified, baselined]	
PR	SI.4.3 Construct or update <i>Software Components</i> based on the detailed part of the <i>Software Design</i> .	<i>Software Design</i> [verified, baselined], <i>Traceability Record</i> [verified, baselined]	<i>Software Components</i>
PR	SI.4.4 Design or update unit test cases and apply them to verify that the <i>Software Components</i> implements the detailed part of the <i>Software Design</i> .	<i>Software Components</i>	<i>Software Components</i> [unit tested]
PR	SI.4.5 Correct the defects found until successful unit test (reaching exit criteria) is achieved.	<i>Software Components</i> [unit tested]	<i>Software Components</i> [corrected]
PR	SI.4.6 Update the <i>Traceability Record</i> incorporating <i>Software Components</i> constructed or modified.	<i>Software Components</i> [corrected] <i>Traceability Record</i> [verified, baselined].	<i>Traceability Record</i> [updated]
TL	SI.4.7 Incorporate <i>Software Components</i> and <i>Traceability Record</i> to the <i>Software Configuration</i> as part of the baseline.	<i>Software Components</i> [corrected] <i>Traceability Record</i> [updated]	<i>Software Configuration</i>  <i>Software Components</i> [corrected, baselined]  <i>Traceability Record</i> [updated baselined]

Figura B.7: Tabla 18 de la norma ISO/IEC 29110 [7] perteneciente al Capítulo 8. Construcción del Software.

Table 19 — SI.5 task list

Role	Task List	Input Products	Output Products
TL PR	SI.5.1 Assign tasks to the work team members related to their role, according to the current <i>Project Plan</i> .	<i>Project Plan</i>	
PR	SI.5.2 Understand <i>Test Cases and Test Procedures</i> .  Set or update the testing environment.	<i>Test Cases and Test Procedures</i> [verified, baselined]	
PR	SI.5.3 Integrates the <i>Software</i> using <i>Software Components</i> and updates <i>Test Cases and Test Procedures</i> for integration testing, as needed.	<i>Software Components</i> [corrected, baselined]  <i>Test Cases and Test</i>	<i>Software</i>  <i>Test Cases and Test Procedures</i>
		<i>Procedures</i> [verified]  <i>Traceability Record</i> [updated, baselined]	
PR CUS	SI.5.4 Perform <i>Software</i> tests using <i>Test Cases and Test Procedures</i> for integration and document results in <i>Test Report</i> .	<i>Software</i>  <i>Test Cases and Test Procedures</i>	<i>Software</i> [tested]  <i>Test Report</i>
PR	SI.5.5 Correct the defects found and perform regression test until exit criteria is achieved.	<i>Software</i> [tested], <i>Test Report</i> .  <i>Test Cases and Test Procedures</i>  <i>Traceability Record</i> [updated, baselined]	<i>Software</i> [corrected]  <i>Test Report</i> [defects eliminated]
PR	SI.5.6 Updates the <i>Traceability Record</i> if appropriate.	<i>Software</i> [corrected]  <i>Traceability Record</i> [updated, baselined].	<i>Traceability Record</i> [updated]
PR	SI.5.7 Document the <i>Product Operation Guide</i> or update the current guide, if appropriate.	<i>Software</i> [tested]	<i>Product Operation Guide</i>
PR DES	SI.5.8 Verify and obtain approval of the <i>Product Operation Guide</i> , if appropriate (see SI.5.7)	<i>Product Operation Guide</i>  <i>Software</i> [tested]	<i>Verification Results</i>  <i>Product Operation Guide</i> [verified]

Figura B.8: Tabla 19 de la norma ISO/IEC 29110 [7] perteneciente al Capítulo 9. Integración y Pruebas del Software.

Role	Task List	Input Products	Output Products
	Verify consistency of the <i>Product Operation Guide</i> with the <i>Software</i> . The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by DES.		
AN	SI.5.9 Document the <i>Software User Documentation</i> or update the current one, if appropriate.	<i>Software</i> [tested] <i>Software User Documentation</i> [preliminary] (optional)	<i>Software User Documentation</i>
AN CUS	SI.5.10 Verify and obtain approval of the <i>Software User Documentation</i> , if appropriate (see SI.5.9)  Verify consistency of the <i>Software User Documentation</i> with the <i>Software</i> . The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by CUS.	<i>Software User Documentation</i> <i>Software</i> [tested]	<i>Verification Results</i> <i>Software User Documentation</i> [verified]
TL	SI.5.11 Incorporate the <i>Test Cases and Test Procedures</i> , <i>Software</i> , <i>Traceability Record</i> , <i>Test Report</i> , <i>Product Operation Guide</i> and <i>Software User Documentation</i> to the <i>Software Configuration</i> as part of the baseline.	<i>Test Cases and Test Procedures</i> <i>Software</i> [tested] <i>Test Report</i> <i>Traceability</i>	<i>Software Configuration</i> <i>Test Cases and Test Procedures</i> [baselined] <i>Software</i> [tested, baselined] <i>Traceability Record</i> [updated, baselined]
		<i>Record</i> [updated] <i>Product Operation Guide</i> [verified] <i>Software User Documentation</i> [verified]	<i>Test Report</i> [baselined] <i>Product Operation Guide</i> [verified, baselined] <i>Software User Documentation</i> [verified, baselined]

Figura B.9: Tabla 19 de la norma ISO/IEC 29110 [7] perteneciente al Capítulo 9. Integración y Pruebas del Software.

Table 20 — SI.6 task list

Role	Task List	Input Products	Output Products
TL WT	SI.6.1 Assign tasks to the work team members related to their role, according to the current <i>Project Plan</i> .	<i>Project Plan</i>	
DES	SI.6.2 Understand <i>Software Configuration</i> .	<i>Software Configuration</i>	
DES	SI.6.3 Document the <i>Maintenance Documentation</i> or update the current one.	<i>Software Configuration</i>	<i>Maintenance Documentation</i>
DES TL	SI.6.4 Verify and obtain approval of the <i>Maintenance Documentation</i> .  Verify consistency of <i>Maintenance Documentation</i> with <i>Software Configuration</i> . The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by TL.	<i>Maintenance Documentation</i>  <i>Software Configuration</i>	<i>Verification Results</i> <i>Maintenance Documentation [verified]</i>
TL	SI.6.5 Incorporate the <i>Maintenance Documentation</i> as baseline for the <i>Software Configuration</i> .	<i>Software Configuration</i> <i>Maintenance Documentation [verified]</i>	<i>Software Configuration</i> <i>Maintenance Documentation [verified, baselined]</i>
TL	SI.6.6 Perform delivery according to <i>Delivery Instructions</i> .	<i>Delivery Instructions</i> <i>Software Configuration</i>	<i>Software Configuration [delivered]</i>

Figura B.10: Tabla 20 de la norma ISO/IEC 29110 [7] perteneciente al Captulo 10. Entrega de Productos.



SIDEP v.2.0



Fecha: 7 de noviembre del 2012  
Versión:1.0f

Manual de Usuario

1. Introducción
2. Selección de Ciclo Escolar y Nivel
3. Consultas e Impresión de Reportes
  - a. Profesores
    - i. Todos
      1. Sin filtro
      2. Con filtro Sede – Especialidad
      3. Con filtro Especialidad - Sede
    - ii. Titular
      1. Sin filtro
      2. Con filtro Sede – Especialidad
      3. Con filtro Especialidad - Sede
    - iii. Adjunto
      1. Sin filtro
      2. Con filtro Sede – Especialidad
      3. Con filtro Especialidad - Sede
    - iv. No Propuesto
      1. Sin filtro
      2. Con filtro Sede – Especialidad
      3. Con filtro Especialidad - Sede
  - b. Jefes de Enseñanza
    - i. Todos
    - ii. Institucionales
    - iii. Nacionales
  - c. Cursos
    - i. Todos
    - ii. Con filtro Institución - Sede
  - d. Sedes
    - i. Todas
    - ii. Por Curso – Institución
    - iii. Por Institución - Curso
  - e. Nombramientos
    - i. Profesores con
    - ii. Profesores sin

Figura B.11: Índice del contenido del Manual de Usuario de Consulta.



SIDEP v.2.0



Fecha: 7 de noviembre del 2012  
Versión: 1.0f

Manual de Jefe de Enseñanza

4. Movimientos

- a. Profesores
  - i. Alta
  - ii. Actualización
  - iii. Baja

Figura B.12: Índice del contenido del Manual de Usuario Jefe de Enseñanza.



SIDEP v.2.0

Fecha: 7 de noviembre del 2012  
Versión: 1.0f

Manual de Usuario Administrador de Cursos

4.Movimientos

- a. Profesores
  - i. Alta
  - ii. Actualización
  - iii. Baja
- b. Jefes de Enseñanza
  - i. Alta
  - ii. Actualización
  - iii. Baja
- c. Cursos
  - i. Alta
  - ii. Actualización
  - iii. Baja
- d. Sedes
  - i. Alta
  - ii. Actualización
  - iii. Baja
- e. Nombramientos
  - i. Alta
  - ii. Actualización
- f. Currículas
  - i. Alta
  - ii. Actualización
- g. Usuarios
  - i. Alta
  - ii. Actualización

Figura B.13: Índice del contenido del Manual de Usuario Administrador de Cursos de la División de Estudios de Posgrado.

---

# Apéndice C

## Informe del Estado de Configuración

Tomando como base la Tabla 23 de la Norma ISO/IEC 29110, que se muestra en las figuras C.2, C.3, C.4, C.5, C.6, C.7, C.8, la tabla indica una lista de entradas, salidas y productos internos de los procesos, sus descripciones, estados posibles y la fuente del producto, dicha tabla se utilizó como base para generar el Informe del Estado de Configuración que se detallará en este Apéndice. El Informe del Estado de Configuración permite recopilar la documentación generada durante este trabajo y que justifica el seguimiento de la Norma ISO/IEC 29110[7].

La escala de colores con la cual han sido coloreadas las tablas de acuerdo al tipo de desarrollo llevado a cabo se presenta en la figura C.1, que indica de izquierda a derecha que el desarrollo fue ágil, tradicional o híbrido respectivamente.



Figura C.1: Escala de colores para las Tablas de Tareas.

**Table 23 — Product Descriptions**

	Name	Description	Source
1.	<i>Acceptance Record</i>	Documents the customer acceptance of the deliverables of the project. It may have the following characteristics: <ul style="list-style-type: none"> <li>- Record of the receipt of the delivery</li> <li>- Identifies the date received</li> <li>- Identifies the delivered elements</li> <li>- Records the verification of any Customer acceptance criteria defined</li> <li>- Identifies any open issues (if applicable)</li> <li>- Signed by receiving Customer</li> </ul>	Project Management
2.	<i>Change Request</i>	Identifies a software, or documentation problem or desired improvement, and requests modifications. It may have the following characteristics: <ul style="list-style-type: none"> <li>- Identifies purpose of change</li> <li>- Identifies request status (new, accepted, rejected)</li> <li>- Identifies requester contact information</li> <li>- Impacted system(s)</li> <li>- Impact to operations of existing system(s) defined</li> <li>- Impact to associated documentation defined</li> <li>- Criticality of the request, date needed by</li> </ul> The applicable statuses are: initiated, evaluated, and accepted.	Software Implementation Customer Project Management
3.	<i>Correction Register</i>	Identifies activities established to correct a deviation or problem concerning the accomplishment of a plan. It may have the following characteristics: <ul style="list-style-type: none"> <li>- Identifies the initial problem</li> <li>- Defines a solution</li> <li>- Identifies corrective actions taken</li> <li>- Identifies the ownership for completion of defined actions</li> <li>- Identifies the open date and target closure date</li> <li>- Contains a status indicator</li> <li>- Indicates follow up actions</li> </ul>	Project Management
4.	<i>Maintenance Documentation</i>	Describes the <i>Software Configuration</i> and the environment used for development and testing (compilers, design tools, construction and tests). It may have the following characteristics: <ul style="list-style-type: none"> <li>- Includes or refers to all Software Configuration elements developed during implementation</li> <li>- Identifies environment used for development and testing (compilers, design tools, construction and tests tools)</li> </ul> It is written in terms that maintenance personnel can understand The applicable statuses are: verified and baselined.	Software Implementation
5.	<i>Meeting Record</i>	Records the agreements established with Customer and/or Work Team. It may have the following characteristics: <ul style="list-style-type: none"> <li>- Purpose of meeting</li> </ul>	Project Management

Figura C.2: Tabla 23 Descripción de productos de la Normal ISO/IEC 29110.

	Name	Description	Source
		<ul style="list-style-type: none"> <li>- Attendees</li> <li>- Date, place held</li> <li>- Reference to previous minutes</li> <li>- What was accomplished</li> <li>- Identifies issues raised</li> <li>- Any open issues</li> <li>- Agreements</li> <li>- Next meeting, if any.</li> </ul> <p>The applicable status is: updated.</p>	
6.	<i>Product Operation Guide</i>	<p>Contains the necessary information to install and manage the software. It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>- Criteria for operational use</li> <li>- A description of how to operate the product including:                             <ul style="list-style-type: none"> <li>- operational environment required</li> <li>- supporting tools and material (e.g. user manuals) required</li> <li>- possible safety warnings</li> <li>- start-up preparations and sequence</li> <li>- frequently asked questions (FAQ)</li> <li>- sources of further information and help to operate the product</li> </ul> </li> <li>- Certification and safety approvals</li> <li>- Warranty and replacement instructions</li> <li>- It should be written in terms that the personnel responsible for the operation can understand.</li> </ul> <p>The applicable statuses are: verified and baselined.</p>	Software Implementation
7.	<i>Progress Status Record</i>	<p>Records the status of the project against the <i>Project Plan</i>. It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>- Status of actual tasks against planned tasks</li> <li>- Status of actual results against established objectives / goals</li> <li>- Status of actual resource allocation against planned resources</li> <li>- Status of actual cost against budget estimates</li> <li>- Status of actual time against planned schedule</li> <li>- Status of actual risk against previously identified</li> <li>- Record of any deviations from planned tasks and reason why.</li> </ul> <p>The applicable status is: evaluated.</p>	Project Management
8.	<i>Project Plan</i>	<p>Presents how the project processes and activities will be executed to assure the project's successful completion, and the quality of the deliverable products. It includes the following elements which may have the characteristics as follows:</p> <ul style="list-style-type: none"> <li>- <i>Product Description</i> <ul style="list-style-type: none"> <li>o Purpose</li> <li>o General Customer requirements</li> </ul> </li> <li>- <i>Scope</i> description of what is included and what is not</li> <li>- <i>Objectives</i> of the project</li> </ul>	Project Management

Figura C.3: Tabla 23 Descripción de productos de la Normal ISO/IEC 29110.

	Name	Description	Source
		<ul style="list-style-type: none"> <li>- <i>Deliverables</i> - list of products to be delivered to Customer</li> <li>- <i>Tasks, including</i> verification, validation and reviews with Customer and Work Team, to assure the quality of work products. Tasks may be represented as a Work Breakdown Structure (WBS).</li> <li>- <i>Estimated Duration</i> of tasks</li> <li>- <i>Resources</i> (humans, materials, standards, equipment and tools) including the required training, and the schedule when the resources are needed.</li> <li>- <i>Composition of Work Team</i></li> <li>- <i>Schedule of the Project Tasks</i>, the expected start and completion date for each task, and the relationship and dependencies of the tasks.</li> <li>- <i>Estimated Effort and Cost</i></li> <li>- <i>Identification of Project Risks</i></li> <li>- <i>Version Control Strategy</i> <ul style="list-style-type: none"> <li>- Product repository tools or mechanism identified</li> <li>- Location and access mechanisms for the repository specified</li> <li>- Version identification and control defined</li> <li>- Backup and recovery mechanisms defined</li> <li>- Storage, handling and delivery (including archival and retrieval) mechanisms specified</li> </ul> </li> <li>- <i>Delivery Instructions</i> <ul style="list-style-type: none"> <li>- Elements required for product release identified (i.e., hardware, software, documentation etc.)</li> <li>- Delivery requirements</li> <li>- Sequential ordering of tasks to be performed</li> <li>- Applicable releases identified</li> <li>- Identifies all delivered software components with version information</li> <li>- Identifies any necessary backup and recovery procedures</li> </ul> </li> </ul> <p>The applicable statuses are: verified, accepted, updated and reviewed.</p>	
9.	<i>Project Repository</i>	<p>Electronic container to store project work products and deliveries. It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>- Stores project work products</li> <li>- Stores released deliverables products</li> <li>- Storage and retrieval capabilities</li> <li>- Ability to browse content</li> <li>- Listing of contents with description of attributes</li> <li>- Sharing and transfer of work products between affected groups</li> <li>- Effective controls over access</li> <li>- Maintain work products descriptions</li> <li>- Recovery of archive versions of work products</li> <li>- Ability to report work products status</li> <li>- Changes to work products are tracked to <i>Change Requests</i></li> </ul> <p>The applicable statuses are: recovered and updated.</p>	Project Management

Figura C.4: Tabla 23 Descripción de productos de la Normal ISO/IEC 29110.

	Name	Description	Source
10.	<i>Project Repository Backup</i>	Repository used to backup the <i>Project Repository</i> and, if necessary, to recover the information.	Project Management
11.	<i>Requirements Specification</i>	<p>Identifies the software requirements. It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>- Introduction –general description of software and its use within the scope of the customer business;</li> <li>- Requirements description:               <ul style="list-style-type: none"> <li>- Functionality – established needs to be satisfied by the software when it is used in specific conditions. Functionality must be adequate, accurate and safe.</li> <li>- User interface – definition of those user interface characteristics that allow to understand and learn the software easily so the user be able to perform his/her tasks efficiently including the interface exemplar description;</li> <li>- External interfaces – definition of interfaces with other software or hardware;</li> <li>- Reliability – specification of the software execution level concerning the maturity, fault tolerance and recovery;</li> <li>- Efficiency – specification of the software execution level concerning the time and use of the resources;</li> <li>- Maintenance – description of the elements facilitating the understanding and execution of the future software modifications;</li> <li>- Portability – description of the software characteristics that allow its transfer from one place to other;</li> <li>- Design and construction limitations/constraints – needs imposed by the customer;</li> <li>- Interoperability – capability for two or more systems or software components be able to change information each other and use it.</li> <li>- Reusability – feature of any product/sub-product, or a part of it, so that it can be used by several users as an end product, in the own software development, or in the execution of other software products.</li> <li>- Legal and regulative – needs imposed by laws, regulations, etc.</li> </ul> </li> </ul> <p>Each requirement is identified, unique and it is verifiable or can be assessed.</p>	Software Implementation

Figura C.5: Tabla 23 Descripción de productos de la Normal ISO/IEC 29110.

	Name	Description	Source
		The applicable statuses are: verified, validated and baselined.	
12.	Software	Software item (software source and executable code) for a Customer, constituted by a collection of integrated <i>Software Components</i> .  The applicable statuses are: tested and baselined.	Software Implementation
13.	Software Component	A set of related code units.  The applicable statuses are: unit tested, corrected and baselined.	Software Implementation
14.	Software Configuration	A uniquely identified and consistent set of software products including:  <ul style="list-style-type: none"> <li>- Requirements Specification</li> <li>- Software Design</li> <li>- Traceability Record</li> <li>- Software Components</li> <li>- Software</li> <li>- Test Cases and Test Procedures</li> <li>- Test Report</li> <li>- Product Operation Guide</li> <li>- Software User Documentation</li> <li>- Maintenance Documentation</li> </ul> The applicable statuses are: delivered and accepted.	Software Implementation
15.	Software Design	Textual and graphical information on the software structure. This structure may include the following parts:  Architectural High Level Software Design – Describes the overall <i>Software</i> structure: <ul style="list-style-type: none"> <li>- Identifies the required <i>Software Components</i></li> <li>- Identifies the relationship between <i>Software Components</i></li> <li>- Consideration is given to any required:                             <ul style="list-style-type: none"> <li>- software performance characteristics</li> <li>- hardware, software and human interfaces</li> <li>- security characteristics</li> <li>- database design requirements</li> <li>- error handling and recovery attributes</li> </ul> </li> </ul> Detailed Low Level Software Design – includes details of the <i>Software Components</i> to facilitate its construction and test within the programming environment; <ul style="list-style-type: none"> <li>- Provides detailed design (could be represented as a prototype, flow chart, entity relationship diagram, pseudo code, etc.)</li> <li>- Provides format of input / output data</li> <li>- Provides specification of data storage needs</li> <li>- Establishes required data naming conventions</li> <li>- Defines the format of required data structures</li> <li>- Defines the data fields and purpose of each required data element</li> <li>- Provides the specifications of the program structure</li> </ul> The applicable statuses are: verified and baselined.	Software Implementation

Figura C.6: Tabla 23 Descripción de productos de la Normal ISO/IEC 29110.

	Name	Description	Source
16.	<i>Software User Documentation</i>	<p>Describes the way of using the software based on the user interface. It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>- User procedures for performing specified tasks using the Software</li> <li>- Installation and de-installation procedures</li> <li>- Brief description of the intended use of the Software (the concept of operations)</li> <li>- The supplied and required resources</li> <li>- Needed operational environment</li> <li>- Availability of problem reporting and assistance</li> <li>- Procedures to access and exit the Software</li> <li>- Lists and explains software commands and system-provided messages to the user</li> <li>- As appropriate for the identified risk, it includes warnings, cautions, and notes, with corrections</li> <li>- It includes troubleshooting and error correction procedures.</li> </ul> <p>It is written in terms understandable by users.</p> <p>The applicable statuses are: preliminary, verified and baselined.</p>	Software Implementation
17.	<i>Statement of Work</i>	<p>Description of work to be done related to software development. It may Include:</p> <ul style="list-style-type: none"> <li>- <i>Product Description</i> <ul style="list-style-type: none"> <li>o Purpose</li> <li>o General Customer requirements</li> </ul> </li> <li>- <i>Scope</i> description of what is included and what is not</li> <li>- <i>Objectives</i> of the project</li> <li>- <i>Deliverables</i> list of products to be delivered to Customer</li> </ul> <p>The applicable status is: reviewed.</p>	Customer
18.	<i>Test Cases and Test Procedures</i>	<p>Elements needed to test code. Test Case may include:</p> <ul style="list-style-type: none"> <li>- Identifies the test case</li> <li>- Test items</li> <li>- Input specifications</li> <li>- Output specifications</li> <li>- Environmental needs</li> <li>- Special procedural requirements</li> <li>- Interface dependencies</li> </ul> <p>Test Procedures may include:</p> <ul style="list-style-type: none"> <li>- Identifies: test name, test description and test completion date</li> <li>- Identifies potential implementation issues</li> <li>- Identifies the person who completed the test procedure</li> <li>- Identifies prerequisites</li> <li>- Identifies procedure steps including the step number, the required action by the tester and the expected results</li> </ul> <p>The applicable statuses are: verified and baselined.</p>	Software Implementation
19.	<i>Test Report</i>	<p>Documents the tests execution. It may include:</p> <ul style="list-style-type: none"> <li>- A summary of each defect</li> <li>- Identifies the related test case</li> </ul>	Software Implementation

Figura C.7: Tabla 23 Descripción de productos de la Normal ISO/IEC 29110.

	Name	Description	Source
		<ul style="list-style-type: none"> <li>- Identifies the tester who found each defect</li> <li>- Identifies the severity for each defect</li> <li>- Identifies the affected function(s) for each defect</li> <li>- Identifies the date when each defect originated</li> <li>- Identifies the date when each defect was resolved</li> <li>- Identifies the person who resolved each defect</li> </ul> <p>The applicable status is: baselined.</p>	
20.	<i>Traceability Record</i>	<p>Documents the relationship among the requirements included in the <i>Requirements Specification, Software Design elements, Software Components, Test Cases and Test Procedures. It may include:</i></p> <ul style="list-style-type: none"> <li>- Identifies requirements of <i>Requirements Specification</i> to be traced</li> <li>- Provides forward and backward mapping of requirements to <i>Software Design elements, Software Components, Test Cases and Test Procedures.</i></li> </ul> <p>The applicable statuses are: verified, baselined and updated.</p>	Software Implementation
21.	<i>Verification Results</i>	<p>Documents the verification execution. It may include the record of:</p> <ul style="list-style-type: none"> <li>- Participants</li> <li>- Date</li> <li>- Place</li> <li>- Duration</li> <li>- Verification check-list</li> <li>- Passed items of verification</li> <li>- Failed items of verification</li> <li>- Pending items of verification</li> <li>- Defects identified during verification</li> </ul>	Project Management  Software Implementation
22.	<i>Validation Results</i>	<p>Documents the validation execution, It may include the record of:</p> <ul style="list-style-type: none"> <li>- Participants</li> <li>- Date</li> <li>- Place</li> <li>- Duration</li> <li>- Validation check-list</li> <li>- Passed items of validation</li> <li>- Failed items of validation</li> <li>- Pending items of validation</li> <li>- Defects identified during validation</li> </ul>	Software Implementation

Figura C.8: Tabla 23 Descripción de productos de la Normal ISO/IEC 29110.

## C.1. Documento de Aceptación.

Se generó durante la última reunión entre Cliente, Administrador del Proyecto y Equipo de Trabajo, se puede ver en las figuras 11.1 y 11.2 y en el cual se cumplen las características que la Norma ISO/IEC 29110 establece puede tener el documento:

- Registro de la recepción de la entrega.
- Identificación de la fecha de recepción.
- Identificación de los elementos entregados.
- Registro de la verificación de los criterios de aceptación definidos por parte del Cliente.
- Identificación de cualquier asunto pendiente (en caso de ser aplicable).
- Firma de recibido por parte del Cliente [7].

## C.2. Solicitud de Cambio.

Se estableció la forma de llevar a cabo una solicitud de cambio entre el Cliente, Administrador del Proyecto y Equipo de Trabajo, como ejemplo, la única *Solicitud de Cambio* solicitada se presenta en la figura 7.9 del Capítulo 6. Análisis de Requerimientos de Software, en ella se puede apreciar que se cumple con la información que la Norma ISO/IEC 29110 sugiere a continuación:

- Propósito del cambio.
- Estado de la solicitud.
- Información de contacto del solicitante.
- Sistema(s) impactado(s).
- Impacto en la operación de sistemas existentes.
- Impacto en la documentación asociada.
- Criticidad de la solicitud y fecha en que se requiere.

Al final de este Apéndice podrá observarse en las figuras C.9 y C.10 a manera de ejemplo una Historia de Usuario en la cual se identificarán ésta y algunas de las tareas especificadas por la Norma ISO/IEC 29110.

### C.3. Registro de Defectos.

De acuerdo a la Norma ISO/IEC 29110 en este producto se asientan los defectos encontrados, desviaciones o problemas relativos al cumplimiento de un plan para posteriormente determinar las actividades necesarias para corregirlos. Este registro puede contener:

- El problema inicial.
- Una solución.
- Las acciones correctivas a tomar.
- Responsable de la conclusión de las acciones definidas.
- Fecha de apertura y fecha de cierre esperada.
- Un indicador de estado.
- Acciones de seguimiento.

Durante el desarrollo del proyecto el *Registro de Defectos* se llevó a cabo de manera ágil, los mismos fueron registrados en las Historias de Usuario, en las cuales se incluyó un espacio para que el Cliente registrara los defectos en caso de encontrarlos durante la prueba que ejecutó. Al final del Apéndice se podrá observar en las figuras C.9 y C.10 un ejemplo que justifica el seguimiento de la Norma ISO/IEC 29110.

### C.4. Manual de Mantenimiento.

La Norma ISO/IEC 29110 especifica que las características que éste puede tener son las siguientes:

- Incluye o se refiere a todos los elementos de la *Configuración del Software* desarrollados durante la implementación.
- Identifica el ambiente utilizado para el desarrollo y realización de pruebas.

Estos puntos se cumplen con las figuras 5.15 del Capítulo 4. Planeación del Proyecto y 6.1 del Capítulo 5. Inicio de la Implementación de Software.

## **C.5. Minuta.**

La Norma ISO/IEC 29110 sugiere que el contenido de una Minuta puede contener la siguiente información:

- Propósito de la reunión.
- Asistentes.
- Fecha y lugar.
- Referencia a minutas previas.
- Qué se logró
- Identifica cuestiones planteadas.
- Cualquier asunto abierto.
- Acuerdos.
- Próxima reunión (en caso necesario).

Dadas las reuniones tomadas de SCRUM que nos guiaron para el desarrollo de este proyecto y considerando que el lugar de trabajo del Cliente, el Administrador del Proyecto y el Equipo de Trabajo era el mismo, se decidió no llevar registro físico de todas y cada una de las reuniones llevadas a cabo durante el desarrollo. Sin embargo en los casos en que era necesaria la formalización de un acuerdo, éste se plasmaría en un documento que era avalado por las partes involucradas.

## **C.6. Manual de Operación.**

Se entregó estructurado de acuerdo al índice presentado en la sección 9.5 Capítulo 9. Integración y Pruebas del Software y que también contiene lo detallado en la sección 9.9 del mismo Capítulo. Se compone de:

- Secciones del Manual de Usuario (Usuario de Consulta y Usuario Jefe de Enseñanza).
- Secciones del Manual de Administrador de Cursos de la División de Estudios de Posgrado de la Facultad de Medicina.

- Sección de Administración de Ciclos Escolares (Alta,Baja).

De acuerdo a esto se puede ver que se cumple con la información que la Norma ISO/IEC especifica y la cual es:

- Criterios para la operación.
- Una descripción de cómo operar el producto, incluyendo:
  - ⊗ ambiente operativo requerido.
  - ⊗ herramientas de apoyo y material requerido (por ejemplo, manuales de usuario).
  - ⊗ posibles alertas de seguridad.
  - ⊗ preparativos para la puesta en marcha y secuencia de pasos.
  - ⊗ preguntas frecuentes (FAQ).
- Certificación y aprobaciones de seguridad.
- Garantía e instrucciones de reemplazo.
- Debe estar escrito en términos que el personal responsable de la operación pueda entender.

## **C.7. Reporte de Avance.**

La Norma ISO/IEC 29110 establece que un Reporte de Avance puede contener la siguiente información:

- Estado real de las tareas contra las tareas planeadas.
- Estado de los resultados reales contra los objetivos/metas establecidos.
- Estado de los recursos asignados reales contra los recursos planeados.
- Estado de los costos reales contra los presupuestos estimados.
- Estado calendario real contra el calendario planeado.
- Estado de los riesgos actuales con respecto a los identificados previamente.
- Registro de cualquier.

En el desarrollo de este proyecto, los Reportes de Avance se llevaron a cabo en las juntas diarias o semanales que el Equipo de Trabajo mantenía con el Administrador del Proyecto y en las cuales en algunas ocasiones participó el Cliente. Las juntas permitieron cumplir con los puntos antes mencionados, ya que al responder las tres preguntas que SCRUM establece, el avance del proyecto se hacía visible para todos los participantes.

## **C.8. Plan del Proyecto.**

La Norma ISO/IEC 29110 especifica que un Plan de Proyecto puede contener la siguiente información:

- Descripción de producto:
  - ⊗ Propósito.
  - ⊗ Requerimientos generales del cliente.
- Descripción de alcance respecto a lo incluido y lo no incluido.
- Objetivos del proyecto.
- Entregables lista de productos a entregar al cliente.
- Tareas, incluyendo verificaciones, validaciones y revisiones con el cliente y equipo de trabajo que permitan asegurar la calidad de los productos de trabajo. Las tareas pueden ser representadas como una Estructura de Desglose de Trabajo (EDT).
- Duración estimada de las tareas
- Recursos (humanos, materiales, estándares, equipos y herramientas), incluyendo la capacitación necesaria. Incluye la identificación y programación de los recursos.
- Composición del equipo de trabajo.
- Calendario de las tareas del proyecto, indicando la fecha de inicio y fecha de finalización previstas para cada tarea, y las relaciones y dependencias entre ellas.

- Esfuerzo y el costo estimado.
- Identificación de los riesgos del proyecto.
- Estrategia para el control de versiones.
- Herramientas de repositorio o mecanismos identificados.
- Localización y mecanismos de acceso para el repositorio especificado.
- Identificación y control de versiones definidos.
- Respaldo y mecanismos de recuperación definidos.
- Mecanismos de almacenamiento, manipulación y entrega especificados (incluyendo archivo y recuperación).
- Instrucciones de entrega.
- Elementos requeridos para la liberación del producto (por ejemplo, hardware, software, documentación, etc.).
- Requerimientos de entrega.
- Tareas a realizar en orden secuencial.
- Liberaciones aplicables identificadas.
- Información de la versión de todos los componentes de software entregados.
- Procedimientos de respaldo y recuperación necesarios [7].

En el desarrollo del proyecto se decidió cumplir con algunos de los puntos especificados en la Norma ISO/IEC 29110 con los siguientes documentos:

- Enunciado de Trabajo que se puede ver en la figura 5.5 del Capítulo 4. Planeación del Proyecto.
- Plan del Proyecto que se puede ver en la figura A.7 del Apéndice A.
- Material y Herramientas que se puede ver en la figura 5.15 del Capítulo 4. Planeación del Proyecto.
- Estándar de Codificación que se puede ver en la figura 5.16 del Capítulo 4. Planeación del Proyecto.

- Definición del Equipo de Trabajo que se puede ver en la figura 5.17 del Capítulo 4. Planeación del Proyecto.
- Lista de Tareas que se puede ver en las figuras 5.18, 5.19 y 5.20.
- Diagrama de Gantt que se puede ver en la figura 5.21.
- Control de versiones especificado en la sección 4.2 del Capítulo 4. Planeación del Proyecto.

Es claro que para varios de los puntos marcados por la norma no existe evidencia física que los demuestre, sin embargo gracias a SCRUM se pudieron atender durante las reuniones diarias o debido a los canales de comunicación cortos y claros, que existieron a lo largo de todo el desarrollo, entre el Equipo de Trabajo y el Cliente.

## C.9. Repositorio del Proyecto.

El repositorio completo generado durante el proyecto se estructuró de acuerdo a lo especificado en el Capítulo 4. Planeación del Software en la figura 5.25, el mismo ya con los documentos y los fuentes generados finalizó de la siguiente manera:

En el directorio *Versiones* se encuentran versiones del documento generadas antes de la final, aquí solo se especifican versiones finales y la existencia de este directorio sin su contenido.

- Sidep v.2.0
  - Diagramas
    - ◇ Diagrama de Paquetes.
    - ◇ Diagrama de Base de Datos.
    - ◇ Diagrama de Distribución.
    - ◇ Diagrama de Gantt.
    - ◇ Diagrama General de Casos de Uso.
  - Documentación - Software
    - ◇ AP
      - \* 1.Planeación del Proyecto.

- plantillageneral\_1.0.doc
  - plantillageneral\_1.0f.pdf
  - enunciadodetrabajo\_1.0.doc
  - enunciadodetrabajo\_1.0f.pdf
  - plantillaproductbacklog\_1.0.doc
  - plantillaproductbacklog\_1.0f.pdf
  - plantillasprintplanningmeeting\_1.0.doc
  - plantillasprintplanningmeeting\_1.0f.pdf
  - plantillalistatareas\_1.0.doc
  - plantillalistatareas\_1.0f.pdf
  - listatareas\_1.0.doc
  - listatareas\_1.0f.pdf
  - listatareastiempo\_1.0.doc
  - listatareastiempo\_1.0f.pdf
  - materialyherramientas\_1.0.doc
  - materialyherramientas\_1.0f.pdf
  - estandarcodificacion\_1.0.doc
  - estandarcodificacion\_1.0f.pdf
  - equipo\_1.0.doc
  - equipo\_1.0f.pdf
  - listatareastiempoyfechas\_1.0.doc
  - listatareastiempoyfechas\_1.0f.pdf
  - riesgos\_1.0.doc
  - riesgos\_1.0f.pdf
  - versiones\_1.0.doc
  - versiones\_1.0f.pdf
  - alcance\_1.0.doc
  - alcance\_1.0f.doc
  - repositorio\_1.0.doc
  - repositorio\_1.0f.pdf
- \* 2.Ejecución del Plan de Proyecto.
- plantillasolicituddecambio\_1.0.doc

- plantillasolicituddecambio\_1.0f.pdf
- \* 3.Evaluación y Control del Proyecto.
  - Sin documentación
- \* 4.Cierre del Proyecto.
  - entregadesoftware\_1.0.doc
  - entregadesoftware\_1.0f.pdf
- ◇ IS
  - \* 1.Inicio de la Implementación de Software.
    - ambiente\_1.0.doc
    - ambiente\_1.0f.pdf
    - tareassignadas\_1.0.doc
    - tareassignadas\_1.0f.pdf
  - \* 2.Análisis de Requerimientos de Software.
    - Casos de Uso
      - ★ diagramageneraldecasosdeuso\_5.4.doc
      - ★ diagramageneraldecasosdeuso\_5.4f.pdf
      - ★ 1ingresoalsistema\_5.6.doc
      - ★ 1ingresoalsistema\_5.6f.doc
      - ★ 9salirdelsistema\_1.0.doc
      - ★ 9salirdelsistema\_1.0f.pdf
    - Historias de Usuario
      - ★ Ciclo
        - 22altadeciclo\_1.0.doc
        - 22altadeciclo\_1.0f.pdf
        - 23eliminaciondeciclo\_1.0.doc
        - 23eliminaciondeciclo\_1.0f.pdf
        - Versiones
      - ★ Currícula
        - 16altadecurricula\_2.7.doc
        - 16altadecurricula\_2.7f.pdf
        - 17actualizaciondecurricula\_1.0.doc

- 17actualizaciondecurricula\_1.0f.pdf
- 18eliminaciondecurricula\_1.0.doc
- 18eliminaciondecurricula\_1.0f.pdf
- 29consultadecurricula\_1.0.doc
- 29consultadecurricula\_1.0f.pdf
- Versiones
- ★ Curso
  - 7altadecurso\_2.7.doc
  - 7altadecurso\_2.7f.pdf
  - 8actualizaciondecurso\_2.7.doc
  - 8actualizaciondecurso\_2.7f.pdf
  - 9eliminaciondecurso\_2.7.doc
  - 9eliminaciondecurso\_2.7f.pdf
  - 26consultadecursos\_1.0.doc
  - 26consultadecursos\_1.0f.pdf
  - Versiones
- ★ Jefe de Enseñanza
  - 4altadejefe\_1.0.doc
  - 4altadejefe\_1.0f.pdf
  - 5actualizaciondejefe\_2.7.doc
  - 5actualizaciondejefe\_2.7f.pdf
  - 6eliminaciondejefe\_2.7.doc
  - 6eliminaciondejefe\_2.7f.pdf
  - 25consultadejefes\_1.0.doc
  - 25consultadejefes\_1.0f.pdf
  - Versiones
- ★ Nombramiento
  - 13altadenombramiento\_2.7.doc
  - 13altadenombramiento\_2.7f.pdf
  - 14actualizaciondenombramiento\_2.7.doc
  - 14actualizaciondenombramiento\_2.7f.pdf
  - 15eliminaciondenombramiento\_1.0.doc

- 15eliminaciondenombramiento\_1.0f.pdf
  - 28consultadenombramientos\_1.0.doc
  - 28consultadenombramientos\_1.0f.pdf
  - Versiones
- ★ Profesor
- 1altadeprofesor\_2.7.doc
  - 1altadeprofesor\_2.7f.pdf
  - 2actualizaciondeprofesor\_2.7.doc
  - 2actualizaciondeprofesor\_2.7f.pdf
  - 3eliminaciondeprofesor\_2.7.doc
  - 3eliminaciondeprofesor\_2.7f.pdf
  - 24consultadeprofesores\_1.0.doc
  - 24consultadeprofesores\_1.0f.pdf
  - Versiones
- ★ Sede
- 10altadesede\_1.0.doc
  - 10altadesede\_1.0f.pdf
  - 11actualizaciondesede\_1.0.doc
  - 11actualizaciondesede\_1.0f.pdf
  - 12eliminaciondesede\_1.0.doc
  - 12eliminaciondesede\_1.0f.pdf
  - 27consultadesedes\_1.0.doc
  - 27consultadesedes\_1.0f.pdf
  - Versiones
- ★ Usuario
- 19altadeusuario\_2.7.doc
  - 19altadeusuario\_2.7f.pdf
  - 20actualizaciondeusuario\_1.0.doc
  - 20actualizaciondeusuario\_1.0f.pdf
  - 21eliminaciondeusuario\_1.0.doc
  - 21eliminaciondeusuario\_1.0f.pdf
  - 30consultadeusuarios\_1.0.doc

- 30consultadeusuarios\_1.0f.pdf
- Versiones
- plantillasolicituddecambio\_1.0.doc
- plantillasolicituddecambio\_1.0f.pdf
- manualdeusuariopreliminar\_1.0.doc
- manualdeusuariopreliminar\_1.0f.pdf
- \* 3.Arquitectura y Diseño Detallado del Software.
  - especificacionderequerimientos\_1.0.doc
  - especificacionderequerimientos\_1.0f.pdf
  - diagramadepaquetes\_1.0.doc
  - diagramadepaquetes\_1.0f.pdf
  - diagramadedistribucion\_1.0.doc
  - diagramadedistribucion\_1.0f.pdf
  - basededatos\_5.8.doc
  - basededatos\_5.8f.pdf
  - 1ingresaralsistemaprueba\_1.0.doc
  - 1ingresaralsistemaprueba\_1.0f.pdf
  - Versiones
  - Historias de Usuario Diseño  
Contiene las mismas Historias detalladas en el punto anterior sólo que éstas incluyen detalles de diseño añadidas por el Equipo de Trabajo.
- \* 4.Construcción del Software.
  - solicituddecambiocpaem\_1.0.doc
  - solicituddecambiocpaem\_1.0f.pdf
  - Versiones
  - Historias de Usuario Construcción  
Contiene las mismas Historias detalladas en el punto anterior sólo que éstas incluyen detalles de rastreo de construcción de las mismas añadidas por el Equipo de Trabajo.
- \* 5.Integración y Pruebas del Software.
  - indicemanualadmincursosfinal\_1.0.doc

- indicemanualadmincursosfinal\_1.0f.pdf
- indicemanualadminsisis\_1.0.doc
- indicemanualadminsisis\_1.0f.pdf
- indicemanualdeusuarioconsulta\_1.0.doc
- indicemanualdeusuarioconsulta\_1.0f.pdf
- indicemanualjefefinal\_1.0.doc
- indicemanualjefefinal\_1.0f.pdf
- Versiones
- Historias de Usuario Construcción

Contiene las mismas Historias detalladas en el punto anterior sólo que éstas incluyen detalles de las Pruebas llevadas a cabo por el Cliente y las cuales fueron revisadas por el Equipo de Trabajo.

\* 6. Entrega de Productos.

- entregafinal\_1.0.doc
- entregafinal\_1.0f.pdf

⊗ Interfaz

- ◇ index.html
- ◇ menu\_ciclos.html
- ◇ menu.html
- ◇ plantilla\_consultas.html
- ◇ plantilla\_movimientos.html
- ◇ plantilla\_reportes.jasper

⊗ NetBeans

- ◇ index.jsp
- ◇ index\_proceso.jsp
- ◇ menu.jsp
- ◇ salir.jsp
- ◇ style.css
- ◇ AJAX
  - \* ajax\_consuta.js

- \* ajax\_masDetalle.js
- ◇ ReportesPDF
  - \* cursos
    - curso\_institucion\_sede.jasper
    - curso\_institucion\_sede.jrxml
    - curso\_institucion.jasper
    - curso\_institucion.jrxml
    - cursos\_sin\_filtro.jasper
    - cursos\_sin\_filtro.jrxml
  - \* jefes
    - jefe\_institucion.jasper
    - jefe\_institucion.jrxml
    - jefe\_filtro\_institucion.jasper
    - jefe\_filtro\_institucion.jrxml
    - jefe\_filtro\_institucion\_sede.jasper
    - jefe\_filtro\_institucion\_sede.jrxml
    - jefes\_sin\_filtro.jasper
    - jefes\_sin\_filtro.jrxml
  - \* nombramientos
    - profesores\_con.jasper
    - profesores\_con.jrxml
    - profesores\_sin.jasper
    - profesores\_sin.jrxml
    - nombramiento\_de\_profesor.jasper
    - nombramiento\_de\_profesor.jrxml
  - \* profesores
    - profesores\_filtro\_especialidad.jasper
    - profesores\_filtro\_especialidad.jrxml
    - profesores\_filtro\_especialidad\_sede.jasper
    - profesores\_filtro\_especialidad\_sede.jrxml
    - profesores\_filtro\_sede.jasper
    - profesores\_filtro\_sede.jrxml

- profesores\_filtro\_sede\_especialidad.jasper
- profesores\_filtro\_sede\_especialidad.jrxml
- profesores\_sin\_filtro.jasper
- profesores\_sin\_filtro.jrxml
- \* sedes
  - sedes\_curso\_institucion.jasper
  - sedes\_curso\_institucion.jrxml
  - sedes\_filtro\_curso.jasper
  - sedes\_filtro\_curso.jrxml
  - sedes\_filtro\_institucion.jasper
  - sedes\_filtro\_institucion.jrxml
  - sedes\_filtro\_institucion\_curso.jasper
  - sedes\_filtro\_institucion\_curso.jrxml
  - sedes\_sin\_filtro.jasper
  - sedes\_sin\_filtro.jrxml
- \* generaPDF.jsp
- ◇ WEB-INF
- ◇ ciclos
  - \* proceso\_menuciclos.jsp
  - \* menuciclos.jsp
  - \* menuciclos\_proceso.jsp
- ◇ consultas
  - \* cursos
    - menu\_consulta\_Curso.jsp
    - ajax\_detalleCurso.jsp
    - ajax\_masDetalleCurso.jsp
  - \* jefes
    - menu\_consulta\_Jefe.jsp
    - ajax\_detalleJefe.jsp
    - ajax\_masDetalleJefe.jsp
  - \* curriculas

- menu\_consulta\_Curricula.jsp
- ajax\_detalleCurricula.jsp
- \* nombramientos
  - menu\_consulta\_Nombramiento.jsp
  - ajax\_detalleNombramiento.jsp
- \* profesores
  - menu\_consulta\_Profesor.jsp
  - ajax\_detalleProfesor.jsp
  - ajax\_masDetalleProfesor.jsp
- \* sedes
  - menu\_consulta\_Sede.jsp
  - ajax\_detalleSede.jsp
  - ajax\_masDetalleSede.jsp
- \* proceso\_consultas.jsp
- ◇ movimientos
  - \* cursos
    - menu\_movimiento\_Curso.jsp
    - proceso\_menu\_movimiento\_Curso.jsp
    - menu\_movimiento\_Curso\_proceso.jsp
  - \* jefes
    - menu\_movimiento\_Jefe.jsp
    - proceso\_menu\_movimiento\_Jefe.jsp
    - menu\_movimiento\_Jefe\_proceso.jsp
  - \* curriculas
    - menu\_movimiento\_Curricula.jsp
    - proceso\_menu\_movimiento\_Curricula.jsp
    - menu\_movimiento\_Curricula\_proceso.jsp
  - \* nombramientos
    - menu\_movimiento\_Nombramiento.jsp
    - proceso\_menu\_movimiento\_Nombramiento.jsp
    - menu\_movimiento\_Nombramiento\_proceso.jsp

- \* profesores
  - o menu\_movimiento\_Profesor.jsp
  - o proceso\_menu\_movimiento\_Profesor.jsp
  - o menu\_movimiento\_Profesor\_proceso.jsp
- \* sedes
  - o menu\_movimiento\_Sede.jsp
  - o proceso\_menu\_movimiento\_Sede.jsp
  - o menu\_movimiento\_Sede\_proceso.jsp
- \* proceso\_movimientos.jsp
- ◇ scripts
  - \* Scripts.js
- ◇ Source Packages
  - \* ajax.bean
    - o AcademicoBean.java
    - o CursoBean.java
    - o EstadoBean.java
    - o SedeBean.java
    - o detalleBean.java
    - o masDetalleBean.java
    - o detalleAjaxBean.java
  - \* ajax.dao
    - o detalleAjaxDao.java
    - o masDetalle.java
    - o filtro.java
  - \* ajax.logic
    - o ajaxBo.java
  - \* control
    - o Ciclo.java
    - o Control.java
    - o Curso.java
    - o Jefe.java

- Nivel.java
- Profesor.java
- Sede.java
- Nombramiento.java
- Curricula.java
- \* datos
  - Conexion.java
- ◇ Libraries
- ◇ Configuration Files
  - \* MANIFEST.MF

## C.10. Respaldo del Repositorio del Proyecto.

La Norma ISO/IEC 29110 nos indica lo siguiente sobre el Respaldo del Repositorio del Proyecto: Repositorio usado para respaldar el Repositorio del Proyecto, y en caso necesario recuperar la información [7].

Durante el desarrollo del proyecto el respaldo se llevaba a cabo cada que un módulo era terminado, el Equipo de Trabajo realizaba copias del repositorio del proyecto en el equipo de trabajo y dos unidades de almacenamiento extraíble. En una de ellas, la perteneciente al Administrador del Proyecto, se añadía un archivo nombrado README.txt que contenía la información del avance del proyecto y la fecha en que se realizaba el respaldo.

## C.11. Especificación de Requerimientos.

La Norma ISO/IEC 29110 especifica que la Especificación de Requerimientos puede contener la siguiente información:

- Introducción descripción general del software y dentro del ámbito de negocio del cliente;
- Descripción de requerimientos:

- ⊗ Funcionalidad - necesidades que debe satisfacer el software cuando se usa en condiciones específicas. La funcionalidad debe ser adecuada, precisa y segura.
- ⊗ Interfaz de usuario - definición de las características de la interfaz de usuario que permitan comprender y aprender el uso de software fácilmente para que el usuario sea capaz de realizar sus tareas eficientemente. Incluye la descripción del modelo de interfaz.
- ⊗ Interfaces externas - definición de las interfaces con otro software o hardware.
- ⊗ Fiabilidad - especificación del nivel de ejecución de software referente a su madurez, tolerancia a fallas y recuperación.
- ⊗ Eficiencia - especificación del nivel de ejecución del software en relación con el tiempo y el uso de los recursos.
- ⊗ Mantenimiento - descripción de los elementos para facilitar la comprensión y ejecución de futuras modificaciones del software.
- ⊗ Portabilidad - descripción de las características de software que permiten transferirlo de un lugar a otro.
- ⊗ Diseño y construcción de las limitaciones/restricciones - necesidades impuestas por el cliente.
- ⊗ Interoperabilidad - Capacidad para usar o intercambiar información entre dos o más sistemas o componentes de software.
- ⊗ Reutilización - característica de cualquier producto o subproducto o de alguna de sus partes, para ser utilizado por varios usuarios como un producto final, en el desarrollo o ejecución de otros productos de software.
- ⊗ Legales y regulativos - necesidades impuestas por las leyes, regulaciones, etc [7].

A excepción de dos módulos que fueron realizados mediante Casos de Uso (ingresar al sistema y salir del sistema), el resto se realizó mediante metodología ágil. Al final de este Apéndice se añaden las imágenes C.9 y C.10 que permiten ver la manera en que se cumplió con ésta y otras tareas del desarrollo.

## **C.12. Software.**

La Norma ISO/IEC 29110 nos indica lo siguiente sobre el Software: Elemento de software (código fuente y código ejecutable) para un cliente, integrado por un conjunto de Componentes de Software [7].

De acuerdo a lo indicado por la Norma, el Software se entregó en el Repositorio del Proyecto (código fuente) y el ejecutable estará alojado en la dirección:

[www.sidep.fmposgrado.unam.mx:8080/sidep.v.2.0/](http://www.sidep.fmposgrado.unam.mx:8080/sidep.v.2.0/)

## **C.13. Componente de Software.**

La norma ISO/IEC 29110 describe el Componente de Software como: Conjunto de unidades de código relacionadas [7]. Para el desarrollo llevado a cabo por el Equipo de Trabajo y dado que los detalles del software fueron especificados mediante Historias de Usuario, las pruebas unitarias fueron ejecutadas de acuerdo a lo que el Equipo de Trabajo entendió en cada una de las Historias de Usuario generadas, en caso de que la prueba fuera exitosa, se liberaba el módulo. Para relacionar las unidades de código, se añadieron a las Historias de Usuario registros de rastreo por parte del Equipo de Trabajo, al final del Apéndice se muestran las figuras C.9 y C.10 ejemplificando ésta y otras secciones del Apéndice.

## **C.14. Configuración de Software.**

De acuerdo a la Norma ISO/IEC 29110, esta sección deberá incluir:

- Especificación de Requerimientos.
- Diseño de Software.
- Registro de Rastreo.
- Componentes de Software.
- Software.
- Casos y Procedimientos de Prueba.
- Reporte de Pruebas / Resultados de la Pruebas.

- Manual de Operación.
- Manual de Usuario.
- Manual de Mantenimiento

El Desarrollo cumplió con lo especificado por la Norma de distintas maneras, la Especificación de Requerimientos se llevo a cabo mediante Historias de Usuario, además el Equipo de Trabajo aprovechó las constantes reuniones con el Administrador del Proyecto y Cliente para aclarar dudas de la lógica del negocio, estas Historias de Usuario no sólo permitieron lograr el primero de los puntos que indica la Norma sino que también sirvieron para el Diseño del Software, Registro de Rastreo, identificación de los Componentes del Software, ejecución de Casos y Procedimientos de Prueba, Reportear y mostrar Resultados de las Pruebas, estos puntos llevados a cabo por el Equipo de Trabajo mediante la agregación de los comentarios respectivos en las Historias de Usuario. Al final del Apéndice se muestran las figuras C.9 y C.10 ejemplificando ésta y otras secciones del Apéndice.

Los Manuales se pueden ver en las figuras B.11, B.12, B.13 del Apéndice B y en la figura 10.3 del Capítulo 9. Integración y Pruebas del Software que indica la sección extra que completa el Manual de Operación.

## C.15. Diseño de Software.

La Norma ISO/IEC 29110 indica que el contenido del Diseño de Software ser puede dividir en dos y propone:

- Diseño Arquitectónico del Software (alto nivel) - Describe la estructura global del Software:
  - ⊗ Identifica los Componentes de Software requeridos.
  - ⊗ Identifica las relaciones entre los Componentes de Software.
  - ⊗ Consideraciones requeridas:
    - ⊗ características de desempeño de software.
    - ⊗ hardware, software e interfaces humanas.
    - ⊗ características de seguridad.

- ⊗ requerimientos de diseño de base de datos.
- ⊗ manejo de errores y atributos de recuperación.
- Diseño Detallado del Software (bajo nivel) - incluye detalles de los Componentes de Software para facilitar su construcción y prueba dentro del ambiente de desarrollo:
  - ⊗ Proporciona detalle del diseño (puede ser representado como un prototipo, diagrama de flujo, diagrama entidad-relación, pseudo código, etc.)
  - ⊗ Proporciona el formato de entrada / salida de los datos.
  - ⊗ Proporciona la especificaciones del almacenamiento de los datos.
  - ⊗ Establece convenciones de nombrado de los datos.
  - ⊗ Define el formato de las estructuras de datos.
  - ⊗ Define los campos de datos y el propósito de cada elemento de los datos.
  - ⊗ Proporciona las especificaciones de la estructura del programa

De acuerdo a lo que la Norma sugiere y dado que se llevó un diseño basado en Historias de Usuario que el Cliente generó, al final del Apéndice se muestran las figuras C.9 y C.10 a manera de ejemplo que justifican el seguimiento de la Norma, además se pueden observar en la figura 5.23 del Capítulo 4. el estándar de codificación utilizado para este proyecto. Además en la sección 7.3 del Capítulo 7. Arquitectura y Diseño Detallado del Software se tiene una explicación detallada de las características del software, con esto se puede asegurar el seguimiento de la Norma.

## C.16. Manual de Usuario.

La Norma nos proporciona la siguiente lista de la información que puede contener el Manual de Usuario:

- - Procedimientos del usuario para realizar tareas específicas utilizando el Software.
- Procedimientos de instalación y desinstalación.
- Breve descripción del uso previsto del Software (el concepto de operaciones).

- Los recursos provistos y requeridos.
- Entorno operacional requerido.
- Facilidad para reportar problemas y asistencia.
- Procedimientos para acceder y salir del Software.
- Relación y explicación de los comandos del Software y de los mensajes del sistema hacia el usuario.
- Advertencias, precauciones y notas para identificar riesgos y su forma de corrección.
- Incluye procedimientos para solución de problemas y corrección de errores.

El Equipo de Trabajo realizó el Manual de Usuario de acuerdo a las necesidades especificadas por el Cliente, los índices de los Manuales de Usuario se pueden observar en la figuras B.11, B.12 y B.13.

## C.17. Enunciado de Trabajo.

De acuerdo a la Norma ISO/IEC 29110 éste puede incluir:

- Descripción del producto:
  - ⊗ Propósito
  - ⊗ Requerimientos generales del cliente.
- Alcance, que describa que sí incluye y qué no.
- Objetivos del proyecto.
- Entregables, lista de productos a entregar al cliente.

Para justificar el seguimiento de la Norma se puede ver el Capítulo 3. Planteamiento del Problema en su totalidad, el cual representa todos los puntos además de la figura 5.5 del Capítulo 4. Planeación del Proyecto.

## C.18. Casos y Procedimientos de Prueba.

La Norma ISO/IEC 29110 nos indica que los Casos de prueba pueden incluir:

- Identificación del caso de prueba.
- Elementos a probar.
- Especificaciones de entrada.
- Especificaciones de salida.
- Ambiente requerido.
- Requerimientos de procedimientos especiales.
- Dependencias de interfaz.

y que los Procedimientos de Prueba pueden incluir:

- Identificación de: nombre de la prueba, descripción de la prueba y la fecha de terminación de prueba.
- Identificación de posibles problemas de aplicación.
- Nombre de la persona que realiza el procedimiento de prueba.
- Identificación de los requerimientos previos.
- Identificación de los pasos del procedimiento, incluyendo el número de paso, la acción requerida por el probador y los resultados esperados.

En este desarrollo, los Casos y Procedimientos de Prueba fueron llevados a cabo de manera ágil mediante la prueba de cada una de las Historia de Usuario generadas por parte del Cliente, las mismas guiaron al Equipo de Trabajo para el diseño y construcción de cada uno de los módulos del sistema, al final del Apéndice se puede observar una imagen que detalla la forma en que se llevaron a cabo los Casos y Procedimientos de Prueba.

## C.19. Reporte de Pruebas.

El Cliente probó cada uno de los módulos verificando que se cumpliera lo descrito en las Historias de Uso especificando los datos de entrada y la salida esperada, en caso

de encontrar alguna falla o querer dejar un comentario éste era añadido a la Historia de Usuario en la sección respectiva, permitiendo al Equipo de Trabajo corregir el error para realizar una segunda prueba y poder liberar el módulo en caso de que la corrección del módulo fuera exitosa. Al final del Apéndice se pueden observar las imágenes C.9 y C.10 que muestran la manera en que ésto se llevó a cabo, permitiendo así cumplir con las sugerencias de la Norma ISO/IEC 29110 las cuales nos indican que el Reporte de Pruebas puede contener:

- Un resumen de cada defecto.
- Identificación del caso de prueba en cuestión.
- Nombre del probador que encontró cada defecto.
- Severidad de cada defecto.
- Identificación de la(s) función(es) afectada(s) por cada defecto.
- Fecha en que cada defecto fue originado.
- Fecha en que cada defecto fue resuelto.
- Nombre de la persona quin resolvi cada defecto.

## **C.20. Rastreo.**

La Norma ISO/IEC 29110 nos indica que el Rastreo puede incluir:

- Especificación de los requerimientos por rastrear.
- Proporciona el mapeo (hacia adelante y hacia atrás) de los requerimientos a los elementos del Diseño de Software, los Componentes de Software, los Casos de Prueba y los Procedimientos de Prueba.

El Equipo de Trabajo llevó a cabo el Rastreo dentro de las Historias de Usuario, como se podrá ver al final del Apéndice en las figuras C.9 y C.10, el Rastreo se añadió a las mismas al final de cada una de las Historias de Usuario.

## C.21. Listas de Verificación.

La Norma ISO/IEC 29110 dice que las Listas de Verificación pueden incluir:

- Participantes.
- Fecha.
- Lugar.
- Duración.
- Lista de verificación.
- Elementos aprobados.
- Elementos no aprobados.
- Elementos pendientes de verificar.
- Defectos identificados durante la verificación.

Al ser un desarrollo guiado por Historias de Usuario, la verificación se llevó a cabo en las mismas y la realizó el Cliente, al final del Capítulo se verá ejemplificado esto en las figuras C.9 y C.10.

## C.22. Listas de Validación

Al igual que las Listas de Verificación la Norma ISO/IEC 29110 nos indica que las Listas de Validación pueden incluir:

- Participantes.
- Fecha.
- Lugar.
- Duración.
- Lista de verificación.
- Elementos aprobados.
- Elementos no aprobados.

- Elementos pendientes de validar.
- Defectos identificados durante la validación.

De la misma manera que sucedió con las Listas de Verificación, las Listas de Validación se llevaron a cabo de manera ágil. Así en el ejemplo siguiente mostrado en las figuras C.9 y C.10 se puede ver ésta y las secciones mencionadas a ejemplificar con la Historia de Usuario.

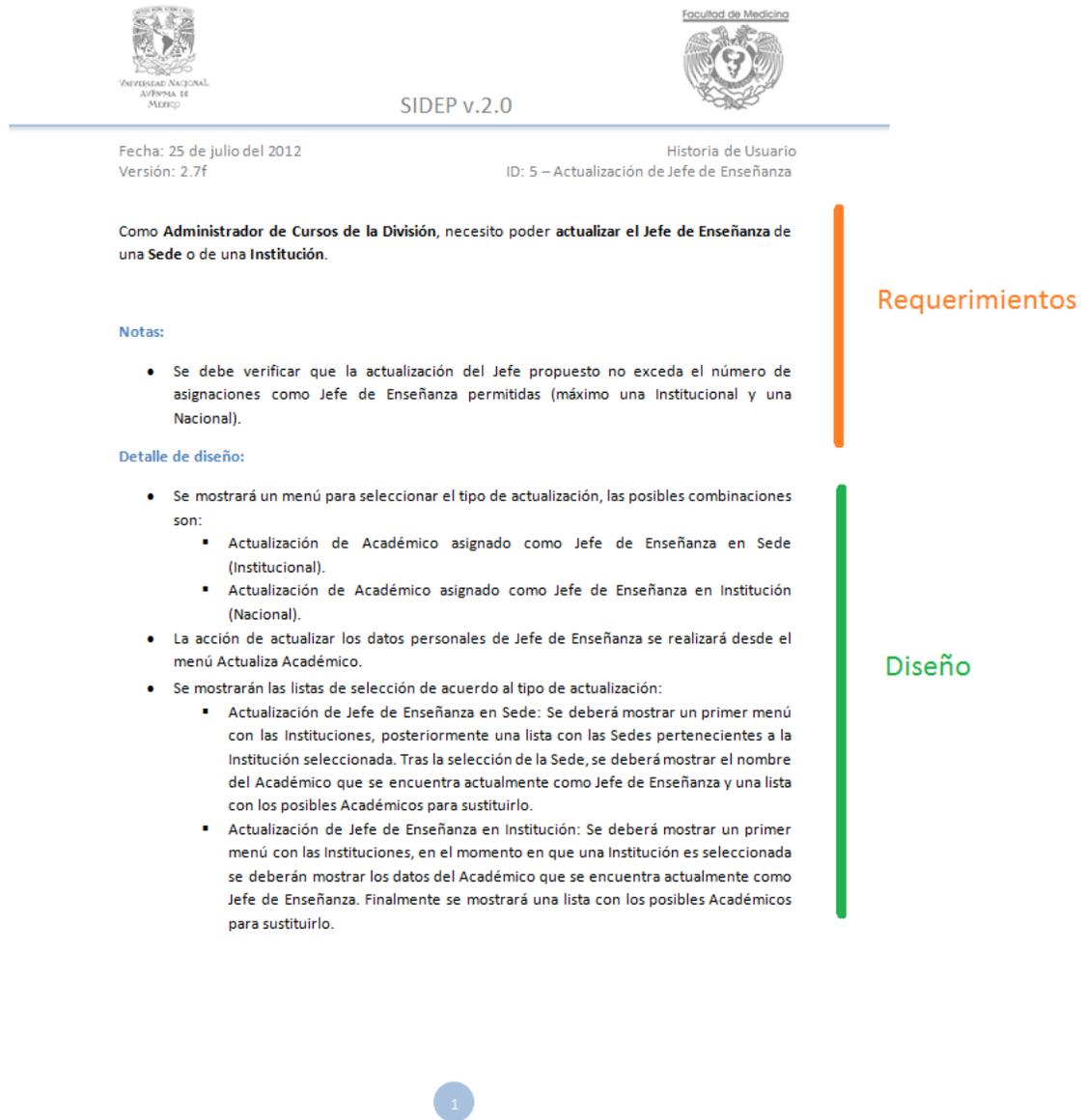


Figura C.9: Historia de Usuario que ejemplifica las diversas tareas ejecutadas durante el Desarrollo del Proyecto.



UNIVERSIDAD NACIONAL  
AVINMA DE  
MEXICO

SIDEP v.2.0



Facultad de Medicina

---

Fecha: 25 de julio del 2012  
Versión: 2.7f

Historia de Usuario  
ID: 5 – Actualización de Jefe de Enseñanza

**Pruebas:**

Fecha de validación:	30 / 10 / 12	Vo.Bo. Equipo de Trabajo	✓
Fecha de revisión:	5 / 11 / 12	Vo.Bo.Dr. Agles Cruz Avelar	✓

Aprobado

Rechazado

Notas de Rechazo (en caso de ser necesario):

**Detalle de Prueba:**

Probaré el módulo actualizando un Jefe de Enseñanza de la siguiente manera:

- Actualizaré un Jefe de Enseñanza de una Sede cualquiera ingresando un Académico nuevo.
- Actualizaré un Jefe de Enseñanza de una Sede cualquiera seleccionado un Académico registrado.
- Intentaré actualizar un Jefe de Enseñanza de una Sede cualquiera intentando asignar un Académico ya asignado como Jefe de Enseñanza (no se debe poder).

**Registro de Rastreo:**

menu\_movimiento\_jefes.jsp -> actualizacion\_jefe.jsp

Validación

Verificación

Registro de Defectos

Pruebas

Registro de Rastreo

Figura C.10: Historia de Usuario que ejemplifica las diversas tareas ejecutadas durante el Desarrollo del Proyecto.

---

# Bibliografía

- [1] David J. Anderson. *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, 2010.
- [2] Mike Cohn. *Agile Estimating and Planning*. Prentice Hall, 2006.
- [3] División de Estudios de Posgrado de la Facultad de Medicina de la Universidad Nacional Autónoma de México. División de estudios de posgrado. facultad de medicina, unam. <http://www.fmposgrado.unam.mx>, 2013. Visitado: 2013-07-01.
- [4] Jesse James Garrett. Ajax: A new approach to web applications. <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/>, 2005. Visitado: 2013-10-10.
- [5] Watts S. Humphrey. *PSP: A Self-Improvement Process for Software Engineers*. Pearson, 2005.
- [6] IBM. Ibm rational unified process (rup). <http://www-01.ibm.com/software/rational/rup/>, 2013. Visitado: 2013-07-01.
- [7] ISO/IEC. Software engineering lifecycle profiles for very small entities (vses) part 5-1-1: Management and engineering guide:generic profile group: Entry profile, 2012. ISO/IEC TR 29110-5-1-1 Technical Report.
- [8] Rumbaugh J. Jacobson I., Booch G. *El Proceso Unificado de Desarrollo de Software*. Pearson, 2000.
- [9] Carnegie Mellon. *SEI Report on Undergraduate Software Engineering Education*. Software Engineering Institute, 1990.

- 
- [10] Peter Hughes Paul Fisher, James Mc Daniel. System development life cycle models and methodologies, 2010. Canadian Society for International Health Certificate Course in Health Information Systems Module 3: System Analysis and Database Development Part 3: Life Cycle Models and Methodologies.
  - [11] Ken Schwaber. *Agile Project Management with Scrum*. Microsoft Press, 2004.
  - [12] IEEE Computer Society. *IEEE Standard Glossary of Software Engineering Terminology*. IEEE STD 610.12-1990, 1990.
  - [13] Jeff Ullman y Jennifer Widom. *First Course in Database Systems 3er.Ed.* Prentice Hall, 2008.
  - [14] Wang Y. *Software Engineering Foundations: A Software Science Perspective*. Auerbach Publications, 2008.