



Universidad de las Ciencias  
Informáticas

**Facultad 1**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS**

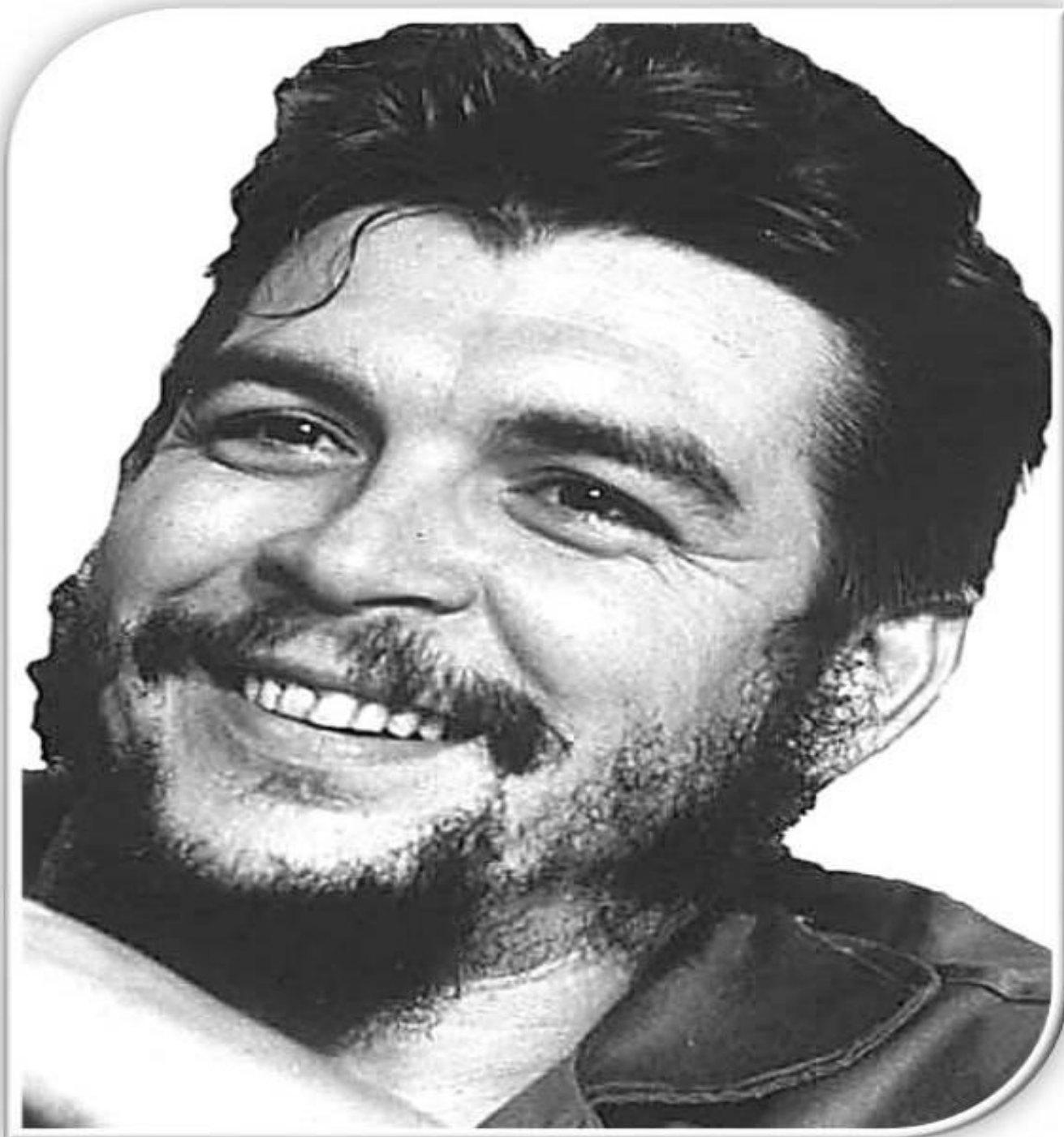
**Título:** Versionado de la Información Personal en el Sistema de Gestión  
Universitaria



**Autores:** Osneidis de la Caridad Mendoza Tamayo  
Manuel Ernesto Velázquez Sánchez

**Tutores:** Ing. Norges Sánchez Tumbarell  
Ing. Arlennys Susana Velázquez Hidalgo  
Ing. Yoan Carlos Machado Espinosa

**La Habana, Junio de 2013**  
**“Año 55 de la Revolución”**



*“Recuerden que el eslabón más alto que puede alcanzar la especie humana es ser revolucionario”.*

*Ernesto “Che” Guevara*



## Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos al Centro de Informatización Universitaria de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Osneidis de la Caridad Mendoza Tamayo

**(Autora)**

\_\_\_\_\_  
Manuel Ernesto Velázquez Sánchez

**(Autor)**

\_\_\_\_\_  
Ing. Norges Sánchez Tumbarell

**(Tutor)**

\_\_\_\_\_  
Ing. Arlennys Susana Velázquez Hidalgo

**(Tutora)**

\_\_\_\_\_  
Ing. Yoan Carlos Machado Espinosa

**(Tutor)**



## Dedicatoria

*Dedico mi Trabajo de Diploma a mis padres Marina y Osmany por ser el motor que me impulsa y me da fuerza para seguir adelante.*

*A mi abuela Amanda por ser la luz de mi vida y de mis ojos.*

*A mis hermanos José Ángel, Keibis y Osmany para que le sirva de guía en su camino por la vida.*

*A mi hermana de corazón y de alma Oslaydis por ser la persona que me enseñó lo bueno y malo de la vida.*

*A mis amigos de siempre que confiaron en mí, en especial a Amalia, Yanisleydis, Yasel y Manuel.*

***Osneidis de la C. Mendoza Tamayo***

*Dedico el trabajo de diploma a lo más sagrado de mi vida, mis padres.*

*A mi madrastra, mi segunda madre.*

*A mi padrino Jorge (El Moro) por inspirarme mucha fe y confianza en la realización de mi tesis.*

*A esa persona especial que desde que la conocí no ha dejado de ayudarme.*

*A mis amigos de siempre que estuvieron presentes en cada momento feliz y triste que pase en la universidad.*

***Manuel E. Velázquez Sánchez***



## Agradecimientos

*A las personas que más quiero, mis padres, porque dieron todo por ayudarme a cumplir este sueño, por estar siempre ahí en los momentos difíciles y por formar a la persona fuerte y emprendedora que soy hoy.*

*Gracias mamita por tus regaños y consejos que a pesar de ser muchos, me enseñaron a ver la vida desde otro punto de vista. Gracias papi por estar presente siempre en cada momento en que te he necesitado. Gracias a los dos por apoyarme en toda las metas que me trace en la vida.*

*A lo más lindo y grande de mi vida, ¡Abuela te adoro!, un enorme beso y un gran abrazo para ti desde el fondo de mi corazón, porque desde el momento en que llegue al mundo me llenaste de un cariño y amor infinito. Gracias a ti aprendí a ser incondicional con la gente y a creer en la bondad de la humanidad.*

*A mi hermana, ¡Oly te quiero mucho!, por a acompañarme en mi camino por la vida, siempre aconsejándome y brindándome tu apoyo.*

*A mis hermanos, José Ángel que a pesar de ser un cabeza dura lo quiero con la vida, a Keibis y Osmanito por sus muestras de cariño. ¡Los adoro!*

*A mi familia en general por aportar un granito de arena en mi formación, por apoyarme y brindarme su amor. A mis tíos Inés María, Juan José, Rafael, a mis primos Osmani, Juan Carlos, y Charliet que aunque lejos siempre estuvo presente. A mis primas Arasays y Arais porque han sido como las hermanas menores que no tuve.*

*A mis amigos de corazón y alma. A Amalia por su amistad incondicional y por mostrarme que más que mi amiga es como mi hermana. A Yasel por apoyarme y brindarme su mano siempre sin esperar nada a cambio, gracias por ser mi amigo. A Yanisleydis por mostrarme que puedo contar contigo y por enseñarme a confiar más en mí.*

*En especial a un amigo del alma, Manuel, que aunque no está aquí siempre lo llevo presente en mi corazón, porque me mostro los más lindos sentimientos y la más hermosa bondad, gracias donde quiere que estés.*

*Agradecer a una persona muy especial, a mi novio, ¡Yoan te quiero mucho!, gracias por apoyarme y estar a mi lado en todo momento.*



*A mis amigos UCI, que más que una familia me brindaron un hogar, gracias a los que están y los que conocí que no están en estos momentos. A la mimi mayor y a la flaca, ¡Siempre las voy a recordar!, gracias a las dos por su linda amistad.*

*A mi grupo del alma el piquete del 1107 porque con ellos disfrute cosas muy bonitas, en especial a Royli, Javier, Israel, Dannier y Nilberto. Al grupo 6 porque me acogieron desde el primer día con mucho cariño, gracias a todos.*

*A mi compañero de tesis Manuel por su paciencia, y confianza en mí.*

*A mis tutores Norges, Yoan Carlos y Arlennys por preocuparse y apoyarme durante el trabajo de diploma, y por contribuir con mi formación profesional.*

*A todas las personas que no mencione, pero con las que compartí en este pedacito de mundo que es la UCI, y de las que me voy a llevar un grato recuerdo, a ti muchas gracias por ofrecerme tu amistad.*

***Osneidis de la C. Mendoza Tamayo***



## Agradecimientos

*Al impulsor de la Revolución Cubana, Fidel Castro, por haber creado esta universidad con el propósito de crear una tropa de futuros revolucionarios y comprometidos con la historia de la Revolución.*

*A mis padres, por educarme y hacer de mí la persona que soy, por sacrificarse siempre por mí, porque de ellos aprendí a luchar por mi futuro, les agradezco por todo lo que soy y pueda llegar a ser. A lo más grande de mi vida, mi mamita querida, por estar ahí siempre cada vez que me hace falta, por inspirarme confianza y tranquilidad cuando la necesito. A mi papito por ayudarme a ser un joven ejemplar, por ser un faro para mí en lo profesional. A los dos los amo con todas las fuerzas de mi corazón y más.*

*A todos mis familiares habaneros, en especial a mi tío Angelito, mis primas Nena y Elaine, a Vivian, Imilsys, Luis Lázaro, Trelly, a todos ellos por haber facilitado mi estancia durante los 5 años de la carrera, por ayudarme en todo y por saber que siempre podré contar con ellos para lo que necesite.*

*A mi padrino el Moro, porque ha sido como un padre para mí, por su preocupación y voluntad en ayudarme en lo imposible, por ser el instrumento de una persona especial que desde que la conocí no me ha fallado jamás.*

*A mis tutores Norges, Yoan Carlos y Arlennys por su incansable paciencia conmigo, por su preocupación constante durante el trabajo de diploma, por contribuir con mi formación profesional y personal desde que me vinculé al proyecto productivo.*

*Al quinteto holguinero de Daniel, Gisela, Vladimir, Carlos y Diana, que nos conocimos por primera vez en la guagua al entrar a la UCI en primer año y que por 5 años nos hemos mantenido siempre juntos en travesuras, en el estudio, en las alegrías de la universidad y en especial a Gise y Vladi por convertirse en mis hermanos desde que los conocí y hasta siempre.*

*A mi tío Cruzata y mi prima Liumi, por estar siempre pendientes de mí en el transcurso de estos 5 años de la universidad.*

*A todos los profesores que tuve durante la universidad, por aportar cada uno un granito de arena en mi formación como profesional de esta Revolución.*



*Al secretariado de la FEU viejo y nuevo, por haberme apoyado en todas las tareas vinculadas con la organización mientras fui dirigente y propiciaron que pudiera trabajar en mi tesis, en especial a Nestor, Claudia, Katy y Wendy.*

*A todos mis compañeros del apartamento, del aula y del proyecto, que de una forma u otra hemos convivido en lo profesional y lo social, especialmente a Yander e Israel, por haberme ayudado en varios códigos de la tesis, a JD, Lenier y Osneidis que junto conmigo formamos un grupo que será inolvidable en el trabajo con las bases de datos del Sistema de Gestión Universitaria.*

*Y a todas las personas que no mencioné, pero que siempre están presente en mi corazón, por los momentos agradables que disfrutamos en la vida universitaria, agradecer a la persona amiga que conocí en el docente o siendo dirigente de la FEU, a esa persona que alegró mi corazón en muchos momentos con su amor, a ti muchas gracias por haberte conocido siendo yo un estudiante de la Universidad de las Ciencias Informáticas.*

***Manuel E. Velázquez Sánchez***





## Resumen

El Centro de Informatización Universitaria de la Universidad de las Ciencias Informáticas tiene como encargo el desarrollo del Sistema de Gestión Universitaria (SGU) en el que están integradas varias áreas de procesos que se materializan en la universidad. En estas áreas se evidencia la descentralización de la información personal que no siempre satisfacen a la dirección de la casa de altos estudios, lo cual incide en afectaciones de los índices de veracidad y actualidad de la información y, por ende, la calidad de la misma. Por esta razón el objetivo de la investigación consiste en desarrollar una propuesta para el Versionado de la Información Personal en el SGU, una solución informática que identifique, controle, notifique y registre los cambios que se originan en la información personal aplicando técnicas de sincronización entre bases de datos a las áreas inscritas en dicho sistema. Con la propuesta planteada, al implementarse un sincronizador entre los sistemas inscritos en el SGU se logra que la información de una persona sea la misma en cada área; conociendo además la versión realizada para cada una de ellas. Para la elaboración de la propuesta de solución se utilizaron herramientas, tecnologías y lenguajes libres como: Visual Paradigm, NetBeans, PostgreSQL, entre otras. Para guiar la solución se utilizó un proceso de desarrollo de *software* con enfoque ágil basado en el nivel 2 de CMMI<sup>1</sup> que permite adaptarse a los cambios y a las nuevas necesidades de los clientes.

**Palabras claves:** *base de datos, información personal, sincronización, sistema de gestión universitaria, versión, versionado de información personal.*

---

<sup>1</sup> CMMI (*Capability Maturity Model Integration*, de su significado en español Integración de Modelos de Madurez de Capacidades).



## Contenido

<b>INTRODUCCIÓN</b> .....	1
<b>Fundamentación teórica</b> .....	5
1.1 Introducción .....	5
1.2 Marco conceptual .....	5
1.2.1 <i>Antecedentes de las bases de datos</i> .....	6
1.2.2 <i>Técnicas para el particionado de tablas</i> .....	7
1.2.3 <i>Análisis de soluciones existentes</i> .....	8
1.2.4 <i>Resultado del análisis de las soluciones existentes</i> .....	11
1.3 Necesidad del versionado de la información personal para el SGU .....	11
1.4 Herramientas, tecnologías y lenguajes para el desarrollo .....	12
1.4.1 <i>Herramientas de desarrollo</i> .....	12
1.4.2 <i>Lenguajes utilizados</i> .....	15
1.4.3 <i>Marco de trabajo</i> .....	16
1.5 Proceso de desarrollo de software .....	17
1.5.1 <i>Proceso de desarrollo con enfoque ágil basado en el nivel 2 de CMMI</i> .....	17
1.6 Conclusiones parciales .....	18
<b>Descripción y análisis de la solución</b> .....	19
2.1 Introducción .....	19
2.2 Modelo de dominio .....	19
2.3 Mapa conceptual .....	20
2.4 Propuesta de solución .....	21
2.5 Sincronización en las áreas del SGU .....	22
2.6 Relación de la propuesta de solución con el SGU .....	22
2.7 Estrategia para el levantamiento de requerimientos .....	23
2.8 Especificación de requerimientos de software .....	24
2.8.1 <i>Requerimientos funcionales</i> .....	25
2.8.2 <i>Requerimientos no funcionales</i> .....	28
2.9 Descripción de la arquitectura y el diseño .....	28
2.9.1 <i>Arquitectura Cliente-Servidor</i> .....	28



2.9.2 Patrón arquitectónico Modelo-Vista-Controlador.....	29
2.9.3 Patrón arquitectónico Dos Capas.....	31
2.9.4 Patrones de diseño.....	32
2.10 Estándares de nomenclatura para la versión de la información.....	34
2.11 Modelo de datos.....	35
2.12 Diagrama de despliegue.....	36
2.12.1 Descripción del diagrama de despliegue.....	37
2.13 Conclusiones parciales.....	38
<b>Implementación y validación de la solución.....</b>	<b>39</b>
3.1 Introducción.....	39
3.2 Estándares para el diseño de prototipos.....	39
3.2.1 Distribución de la información en la propuesta de información.....	39
3.2.2 Tipos de mensajes.....	41
3.3 Técnicas de programación.....	43
3.3.1 Programación Orientada a Objetos (POO).....	43
3.4 Estándares de codificación.....	43
3.4.1 Identación, llaves de apertura y cierre y tamaño de las líneas.....	44
3.4.2 Conversión de nomenclatura.....	44
3.4.3 Estructuras de control.....	45
3.4.4 Documentación.....	46
3.4.5 Buenas prácticas.....	46
3.5 Validación de la solución propuesta.....	46
3.6 Validación de requerimientos.....	47
3.6.1 Resultado de la revisión de los requerimientos.....	48
3.7 Estrategia de prueba.....	48
3.7.1 Pruebas unitarias.....	49
3.7.2 Pruebas de integración.....	52
3.7.3 Pruebas de sistema.....	53
3.7.4 Pruebas de aceptación.....	57
3.8 Conclusiones parciales.....	57



<b>CONCLUSIONES GENERALES</b> .....	58
<b>RECOMENDACIONES</b> .....	59
<b>BIBLIOGRAFÍA REFERENCIADA</b> .....	60
<b>BIBLIOGRAFÍA CONSULTADA</b> .....	63
<b>GLOSARIO DE TÉRMINOS</b> .....	64
<b>ANEXOS</b> .....	65
<b>Anexo No.1</b> Descripción de las clases del modelo de dominio.....	65
<b>Anexo No.2</b> Tabla Resumen de la información personal sabana .....	66
<b>Anexo No.3</b> Modelos de entrevistas .....	67
<b>Anexo No.4</b> Especificación de requerimientos funcionales .....	68
<b>Anexo No.5</b> Plan de iteraciones.....	90
<b>Anexo No.6</b> Requerimientos no funcionales. ....	91
<b>Anexo No.7</b> Modelos de datos.....	93
<b>Anexo No.8</b> Criterios para validar requerimientos del cliente .....	93
<b>Anexo No.9</b> Descripción de la técnica del camino básico.....	93
<b>Anexo No.10</b> Casos de pruebas de caja blanca. ....	96
<b>Anexo No.11</b> Pruebas de integración. ....	100
<b>Anexo No.12</b> Casos de pruebas de caja negra. ....	103
<b>Anexo No.13</b> Resultado de las pruebas funcionales.....	123



## Figuras

Figura 1. Etapas de diseño de un programa. ....	7
Figura 2. Sincronización de los nodos en la base de datos Cassandra. ....	10
Figura 3. Modelo de dominio de la solución. ....	20
Figura 4. Mapa conceptual. ....	21
Figura 5. Mapa taxonómico. ....	23
Figura 6. Arquitectura Cliente – Servidor. ....	29
Figura 7. Patrón arquitectónico Modelo-Vista-Controlador. ....	30
Figura 8. Patrón arquitectónico 2-Capas. ....	31
Figura 9. Modelo físico del versionado de información personal. ....	36
Figura 10. Diagrama del modelo de despliegue. ....	37
Figura 11. Vista de presentación. ....	39
Figura 12. Vista de escritorio. ....	40
Figura 13. Vista de gestión de procesos. ....	41
Figura 14. Mensaje de información. ....	42
Figura 15. Mensaje de error. ....	42
Figura 16. Mensaje de confirmación. ....	42
Figura 17. Identación, llaves de apertura y cierre. ....	44
Figura 18. Variables. ....	44
Figura 19. Clases. ....	44
Figura 20. Funciones. ....	45
Figura 21. Estructuras de control. ....	45
Figura 22. Condiciones en varias líneas. ....	46
Figura 23. Documentación. ....	46
Figura 24. Buenas prácticas. ....	46
Figura 25. Tabla Resumen de la información personal sabana. ....	66
Figura 26. Modelo#1 de entrevista para obtención de requerimientos. ....	67
Figura 27. Modelo#2 de entrevista para identificar información sensible. ....	67
Figura 28. Modelo físico del sincronizador. ....	93
Figura 29. Resultados de las pruebas funcionales. ....	123



## Tablas

Tabla 1. Requerimientos funcionales del sistema. ....	25
Tabla 2. Especificación de requerimientos. Crear información. ....	26
Tabla 3. Clasificación requerimientos no funcionales. ....	28
Tabla 5. Prueba de integración entre las funcionalidades: Información y Versionar información. ....	53
Tabla 6. Parte 1. Caso de prueba caja negra. RF_V11_Crear información. ....	55
Tabla 7. Parte 2. Caso de prueba caja negra. RF_V11_Crear información. ....	55
Tabla 8. No conformidades de las pruebas funcionales. ....	57
Tabla 9. Descripción de las clases del modelo de dominio. ....	65
Tabla 10. Especificación de requerimiento. Identificar cambio en la información sensible. ....	68
Tabla 11. Especificación de requerimiento. Identificar quién cambió la información sensible. ...	68
Tabla 12. Especificación de requerimiento. Identificar cuál es la información sensible. ....	68
Tabla 13. Especificación de requerimiento. Notificar el cambio. ....	68
Tabla 14. Especificación de requerimiento. Identificar prioridad de sincronización. ....	68
Tabla 15. Especificación de requerimiento. Sincronizar información al Núcleo. ....	68
Tabla 16. Especificación de requerimiento. Sincronizar información a los sistemas. ....	69
Tabla 17. Especificación de requerimiento. Identificar indicadores de sincronización. ....	69
Tabla 18. Especificación de requerimiento. Registrar indicadores de sincronización. ....	69
Tabla 19. Especificación de requerimiento. Listar información. ....	69
Tabla 20. Especificación de requerimiento. Ver detalles de información. ....	70
Tabla 21. Especificación de requerimiento. Crear información. ....	71
Tabla 22. Especificación de requerimiento. Modificar información. ....	72
Tabla 23. Especificación de requerimiento. Modificar filtros a información. ....	74
Tabla 24. Especificación de requerimiento. Listar tablas asociadas a la información. ....	75
Tabla 25. Especificación de requerimiento. Ver detalles de tablas asociadas a la información. ...	76
Tabla 26. Especificación de requerimiento. Crear tablas asociadas a la información. ....	77
Tabla 27. Especificación de requerimiento. Modificar tablas asociadas a la información. ....	79
Tabla 28. Especificación de requerimiento. Listar atributos de la información. ....	81
Tabla 29. Especificación de requerimiento. Ver detalles de atributos de la información. ....	82
Tabla 30. Especificación de requerimiento. Crear atributos de la información. ....	83
Tabla 31. Especificación de requerimiento. Modificar atributos de la información. ....	84
Tabla 32. Especificación de requerimiento. Listar versión de una información. ....	86
Tabla 33. Especificación de requerimiento. Ver detalles de versión de una información. ....	87
Tabla 34. Especificación de requerimiento. Listar informaciones versionadas. ....	88
Tabla 35. Especificación de requerimiento. Versionar información. ....	89
Tabla 36. Especificación de requerimiento. Asociar la versión y el atributo de una información. ....	90
Tabla 37. Plan de iteraciones. ....	90
Tabla 38. Requerimientos no funcionales. ....	91



Tabla 39. Criterios para validar requerimientos del cliente. ....	93
Tabla 40. CENIA_ SGU_C_DCP -OAB: RFR19, RFR20, RFR21, RFR22.....	96
Tabla 41. CENIA_ SGU_C_DCP -RTA: RFR15, RFR16, RFR17, RFR18. ....	98
Tabla 42. Prueba de integración entre las funcionalidades: Atributos y Versionar información. ....	100
Tabla 43. Prueba de integración entre las funcionalidades: Información y Tablas asociadas. ...	101
Tabla 44. Prueba de integración entre las funcionalidades: Información y Atributos. ....	101
Tabla 45. Prueba de integración entre las funcionalidades: Información y Versión. ....	101
Tabla 46. Prueba de integración al Módulo: Núcleo de Gestión Universitaria. ....	102
Tabla 47. Prueba de integración al Módulo: Pregrado.....	102
Tabla 48. CENIA_ SGU_C_DCP_RF_V11: Listar_información. ....	103
Tabla 49. CENIA_ SGU_C_DCP_RF_V13: Modificar_información. ....	106
Tabla 50. CENIA_ SGU_C_DCP_RF_V14: Ver_Detalles_información. ....	109
Tabla 51. CENIA_ SGU_C_DCP_RF_V15: Modificar_filtro_información. ....	110
Tabla 52. CENIA_ SGU_C_DCP_RF_V25: Listar_informaciones versionadas.....	115
Tabla 53. CENIA_ SGU_C_DCP_RF_V26: Crear_versionado_información.....	120



## **INTRODUCCIÓN**

El avance y desarrollo de la información y la comunicación constituye un hecho de actualidad; los procesos de transmisión y recepción de contenidos, información y mensajes siguen siendo una necesidad de todos los tiempos. Las Tecnologías de la Información y las Comunicaciones (TIC) se han convertido en un eslabón fundamental para la gestión de la información y los recursos.

La información es uno de los elementos esenciales de la cotidianidad, y la necesidad de almacenarla y recuperarla es una constante casi invariable. De igual forma, los mecanismos para transferirla desde su origen hasta la aplicación, o viceversa, han sido diversos con el decursar de los años. Unos de los más empleados han sido las bases de datos, capaces de coleccionar, recopilar y estructurar los datos. Las bases de datos han ido cambiando y modernizando la forma de gestionar la información en el mismo nivel tecnológico de otros *softwares*, pero al encontrarnos en un mundo regido por el conocimiento, donde cada día se generan grandes volúmenes de informaciones, se hace necesario establecer mecanismos que permitan la administración de estos datos de forma accesible y segura. De esta forma surgen los Sistemas Gestores de Bases de Datos (SGBD).

Desde el surgimiento de la informática, en Cuba se han desarrollado numerosos sistemas de gestión con el objetivo de informatizar la sociedad y, a su vez, resolver los problemas existentes en cuanto a organización y almacenamiento de grandes volúmenes de información.

La Universidad de las Ciencias Informáticas (UCI), surgida por la necesidad de integrar a nuestro país al avance tecnológico mundial, cuenta con diferentes centros de desarrollo de *softwares*, dentro de los que se encuentra el Centro de Informatización Universitaria (CENIA), que desarrolla el Sistema de Gestión Universitaria (SGU), una aplicación encargada de administrar todos los procesos y servicios que se brindan en la UCI.

En las bases de datos del SGU se hace necesario el control del versionado de la información personal a partir de las diferentes entradas, con el objetivo de evitar una incorrecta manipulación de la información personal que pueda afectar a los subprocesos que dependan de los datos almacenados.

En las áreas que intervienen en la gestión universitaria de la UCI, se llevan a cabo un conjunto de acciones que implican la verificación de datos asociados a estudiantes, trabajadores docentes y trabajadores vinculados a la Gestión Universitaria (GU). Cada una de las áreas, por la naturaleza de sus trabajadores, se preocupa de que en su radio de acción la información esté según su gestión personal para corroborar la veracidad de estos datos personales, de manera que no se retroalimentan de otras





áreas ni propician agilidad en procesos acreditativos y de control sobre datos legales de la persona. Esto define una descentralización de la información personal y en consecuencia, en la UCI, ocurre el hecho de que una persona esté con información diferente en varios de los sistemas que informatizan los procesos de la GU. Además, en los procesos donde se gestiona información personal, no se tienen almacenadas las instancias para un determinado momento; por ejemplo: cuando se solicita el cierre de matrícula de un período no es posible determinar la matrícula inicial en el sistema; primero, porque este carece de un mecanismo que la delimite, y segundo, porque durante el período ocurren cambios no controlados en la información personal. Esto trae como consecuencia que cuando se solicita una información estadística por la dirección de la institución, se envíen datos que no satisfacen la solicitud realizada, por lo que los índices de veracidad y actualidad de la información son afectados y, por tanto, la calidad de la información personal. Debido a lo expuesto anteriormente queda formulado el siguiente **problema de investigación** ¿Cómo mantener actualizada la información personal en el Sistema de Gestión Universitaria a partir de los cambios originados por los proveedores internos de información, teniendo en cuenta las reglas de negocios definidas para cada área de proceso?, constituyendo como **objeto de estudio** el control del versionado de la información personal dentro de las bases de datos del Sistema de Gestión Universitaria de la Universidad de las Ciencias Informáticas, y centrado en el **campo de acción** del versionado de la información personal en las bases de datos del Núcleo, Pregrado, Cooperación, Postgrado y Residencia.

Para darle solución a lo antes planteado, se propone como **objetivo general**, desarrollar una solución informática que identifique, controle, notifique y registre los cambios que se originan en la información personal, aplicando técnicas de sincronización entre bases de datos a las áreas inscritas en el Sistema de Gestión Universitaria de la Universidad de las Ciencias Informáticas. Para darle cumplimiento al objetivo planteado, se trazan los siguientes **objetivos específicos** dentro de la investigación:

- ✓ Caracterizar el estado del arte sobre sistemas homólogos que identifiquen, controlen y sincronicen los cambios de información personal entre diferentes flujos de negocios.
- ✓ Describir las tecnologías a utilizar para el desarrollo del versionado de la información personal.
- ✓ Elaborar un diseño de la estrategia para el versionado de la información personal.
- ✓ Realizar una propuesta para versionar la información personal en un instante de tiempo determinado.
- ✓ Implementar y validar la estrategia para versionar la información personal.

Para guiar la investigación se propone la siguiente **idea a defender**: Si se garantiza el versionado de la información personal en el Sistema de Gestión Universitaria de la Universidad de las Ciencias Informáticas, se mantiene actualizada la información personal.



Con el propósito de guiar el trabajo hacia el alcance de los objetivos trazados, se definen las siguientes **tareas de investigación:**

- ✓ Caracterización del proceso de control de la información personal en sistemas con múltiples entradas a nivel nacional e internacional.
- ✓ Análisis de sistemas que automaticen control de la información personal.
- ✓ Caracterización de la metodología de desarrollo a utilizar.
- ✓ Caracterización de las herramientas que se utilizarán en el marco de trabajo.
- ✓ Análisis de técnicas para el particionado de tablas en bases de datos.
- ✓ Análisis de técnicas de ingeniería de requerimientos.
- ✓ Diseño del modelado de la base de datos y de la arquitectura.
- ✓ Definición de los patrones para la construcción de la solución propuesta.
- ✓ Definición del estándar de codificación y explicación de las técnicas de programación a utilizar.
- ✓ Implementación de las funcionalidades para la comunicación entre las bases de datos del Sistema de Gestión Universitaria.
- ✓ Definición de las pruebas a realizar a la solución.
- ✓ Realización de las pruebas de integración y pruebas funcionales.

Con el correcto cumplimiento de las tareas se espera obtener los siguientes resultados:

#### **Posibles resultados**

- ✓ Grupo sistémico de funcionalidades a nivel de dato que integran la información personal en las áreas inscritas en el Sistema de Gestión Universitaria de la Universidad de las Ciencias Informáticas.
- ✓ Sincronización de bajo acoplamiento de la información personal contenida en el Núcleo del Sistema de Gestión Universitaria respecto al resto de los áreas inscritas.
- ✓ Disminución del tiempo y esfuerzo en la suscripción de personas en las áreas del Sistema de Gestión Universitaria.
- ✓ Mejora en la calidad de la información personal consultada por la dirección de la universidad.

Para dar cumplimiento a las tareas de la investigación, se tuvo en cuenta un grupo de métodos de investigación científica:



### **Métodos teóricos**

**Analítico – sintético:** método utilizado en el análisis de la bibliografía consultada y la realización de síntesis sobre la información relevante acerca de sistemas de control de cambio en la información personal.

**Análisis Histórico – Lógico:** método utilizado para describir los antecedentes de las bases de datos, así como el estudio y análisis de las principales herramientas para el manejo de datos reproduciendo lo esencial.

**Modelación:** método empleado para describir en diagramas, mediante los componentes que los integran, la dinámica de los procesos e interacción entre objetos a partir de sus relaciones.

### **Métodos empíricos.**

**Entrevistas:** método empleado con el objetivo de obtener información de especialistas con conocimientos, sobre el trabajo con bases de datos y la manipulación de información personal.

### **El trabajo de diploma está compuesto de 3 capítulos:**

**Capítulo 1. Fundamentación teórica:** se recogen los principales conceptos y definiciones que se deben conocer de las bases de datos. Se analizan las soluciones existentes tanto en el ámbito internacional como en el nacional, y se describen las herramientas a utilizar para el diseño e implementación de la solución.

**Capítulo 2. Descripción y análisis de la solución:** se realiza la especificación de los requerimientos a tener en cuenta durante el desarrollo de la propuesta de solución, la descripción de la arquitectura y el diseño de la estrategia a elaborar.

**Capítulo 3. Implementación y validación de la solución:** se presenta la implementación de la estrategia elaborada, y la validación funcional de la misma.



# Fundamentación teórica

## 1.1 Introducción

En este capítulo se presentan definiciones, conceptos, características y aspectos significativos relacionados con las bases de datos. Además, se expone un conjunto de soluciones existentes a nivel internacional y nacional de sistemas que controlan información personal con múltiples entradas, que automatizan los procesos y supervisan la comunicación entre las bases de datos; así como los cambios que les realizan a dicha información personal, permitiendo detallar las principales tendencias y tecnologías sobre las que se apoya la propuesta final.

## 1.2 Marco conceptual

Para lograr una mejor comprensión de la investigación, a continuación se abordan un conjunto de conceptos que son necesarios para el entendimiento de la investigación y que están estrechamente relacionados con el dominio del problema.

Un **dato** es el antecedente necesario para llegar al conocimiento de una cosa o para deducir las consecuencias de un hecho, es la información concreta sobre hechos, elementos, que permite estudiarlos, analizarlos o conocerlos. (*Definicion.de 2013*)

Para la informática, los datos son expresiones generales que describen características de las entidades sobre las que operan los algoritmos. Estas expresiones deben presentarse de una cierta manera para que puedan ser tratadas por una computadora. En este caso, los datos por sí solos tampoco constituyen información, sino que esta surge del adecuado procesamiento de los mismos. Para entender un proceso de versionado de información personal, primeramente se debe partir de conocer los conceptos relacionados a versión, información y personal.

Una **versión** es el modo que tiene cada sujeto de hacer algo o de referir un mismo suceso. La versión de un *software* o solución informática es un número o nombre que se asigna a un programa o conjunto de información para mencionar su nivel de desarrollo y su actualización. Lo habitual es que el versionado esté dado por números, separados por un punto. (*Definicion.de 2013*)

La **información** es un conjunto de datos con significado que estructura el pensamiento de los seres vivos, especialmente, del ser humano. Para la informática, la información es el conjunto de datos organizados y procesados que constituyen mensajes, instrucciones, operaciones, funciones y cualquier tipo de actividad que tenga lugar en relación con un ordenador. (*Definicionabc 2013*)



El **personal** es un adjetivo que hace referencia a lo perteneciente o relativo a la persona. El término suele utilizarse para nombrar a lo propio o particular de ella. En el sector empresarial, el personal es el conjunto de las personas que trabajan en un mismo organismo, empresa o entidad. (Definición de 2013)

Debido a los planteamientos anteriores, los autores de la investigación definen como **información personal** a la información que identifica o caracteriza a una persona dentro del ámbito de la sociedad y al **versionado de información personal** como el proceso donde se obtiene una versión de la información personal actualizada, evitando inconsistencias futuras.

La **sincronización** consiste en la necesidad de garantizar el acceso múltiple y simultáneo a la base de datos, de modo que los datos puedan ser compartidos por diferentes usuarios a la vez. Con la realización de la sincronización como proceso de propagación de los cambios en la información se puede mantener la consistencia de los datos almacenados. (Matos 2004)

Una **base de datos** es un sistema formado por un conjunto de datos almacenados en un soporte legible, organizados internamente por registros y campos que permiten el acceso directo, y por programas que manipulen ese conjunto de datos. Permiten recuperar cualquier variedad de información, ya sea referencias, documentos textuales, imágenes o datos estadísticos. (Anzaldo 2005). Se le puede denominar como el lugar donde se almacenan los datos en reposo y donde se pueden acceder desde diferentes aplicaciones, sistemas o programas. (Rozic 2004). También el término de base de datos se puede manejar como un conjunto de información o de datos relacionados que se encuentran agrupados o estructurados. (Camallea 2004). Por tanto una base de datos engloba a la información concerniente, disponibles para los usuarios, con la finalidad de eliminar la redundancia o al menos minimizarla. Se puede entender como el lugar donde se almacenan una serie de datos organizados, relacionados entre sí y donde estos serán accesibles desde cualquier aplicación.

### 1.2.1 Antecedentes de las bases de datos

El desarrollo de un proyecto transita por un grupo de etapas en su ciclo de vida (**Ver Figura 1**). En su primera etapa se encuentran los datos, que es el eslabón fundamental a la hora de realizar cualquier tarea. La segunda etapa es el proceso, que incluye un conjunto de operaciones o tareas que se realizan en un orden específico y definido. La última etapa es referente a la información, que es el resultado obtenido al ser procesados los datos; el objetivo es adquirir información segura y precisa en el menor tiempo posible. Los datos pueden estar en movimiento o en reposo, estos últimos hay que almacenarlos en archivos, ficheros o bases de datos. (Rozic 2004)



Figura 1. Etapas de diseño de un programa.

Inicialmente la información se almacenaba en archivos, estos estaban compuestos por registros y cada registro por un conjunto de campos. Los archivos se organizaban de tres formas diferentes:

- Archivos secuenciales (se realiza uno tras otro).
- Archivos indexados (se anexa el archivo al índice).
- Archivos relativos o directos (se determina su posición física).

Los archivos tienen algunos inconvenientes como son:

- Redundancia de datos.
- Problema de consistencia e integridad de los datos.
- Dificultad para modificar la estructura lógica de los datos.

Dadas estas dificultades surge la necesidad de una nueva forma de almacenamiento de datos, que además de ser robusta, permite su manipulación y modificación según las necesidades del usuario, es por eso que aparece el término “base de datos”.(Rozic 2004)

### 1.2.2 Técnicas para el particionado de tablas

Para realizar el versionado de la información personal es indispensable que los datos personales estén clasificados en información sensible e información variable. Se entiende como información sensible aquella que no debe ser modificada por personas no autorizadas, entiéndase carné de identidad, nombre y apellidos, etc. Sin embargo, la información variable puede ser modificada en cualquier proceso universitario que lo requiera, por ejemplo, el número de serie del carné de identidad o el nombre de usuario de la persona. Para que la información quede clasificada de la manera propuesta es necesario utilizar un tipo de particionado de tablas en PostgreSQL. El particionado de tablas es una técnica que se usa para reducir la cantidad de lecturas físicas a la base de datos cuando ejecutamos consultas, existen dos principales modalidades de particionado: horizontal y vertical. Tanto las técnicas de particionado horizontal como vertical de una tabla, intentan disminuir los accesos a disco, ordenando físicamente los datos por un factor en común.(Rozic 2004)

- **Particionado horizontal:** Esta modalidad consiste en tener varias tablas con las mismas columnas en cada una de ellas y distribuir la cantidad de registros en estas tablas (generalmente se particiona separando la data por años, meses, etc.). Un ejemplo de este particionado es cuando se tienen tres tablas: registro2001, registro2002 y registro2003, y en cada tabla se guardan los registros de los



años correspondientes. Esto garantiza una mejora en el rendimiento considerable cuando se realicen consultas sobre las tablas, ya que la data estará distribuida en tres partes y dependiendo del año, se sabe en cual tabla buscar.

- **Particionado vertical:** Esta modalidad busca disminuir los accesos a disco a través de la proyección (selección de columnas). El particionado vertical, bien puede ser considerado desde el diseño (dos o más tablas, unidas por una llave con cardinalidad 1:1). El modo de simular el diseño es creando ambas tablas y crear una vista global de ambas, que incluyan todas las columnas.

Analizados los diferentes tipos de particionado de tablas se decide escoger el particionado vertical, siendo el más factible para el desarrollo de la solución. Este tipo de particionado garantiza clasificar la información en sensible y variable, y así poder definir las políticas de actualización de la información personal como parte del versionado de la información personal dentro de las bases de datos del Sistema de Gestión Universitaria (SGU).

### 1.2.3 Análisis de soluciones existentes

La revolución científica y tecnológica en que se encuentra el mundo, está implícita en todos los campos de la sociedad a nivel internacional, provocando que las soluciones se diversifiquen cada vez más aparejadas al desarrollo alcanzado. Existen diferentes soluciones para el intercambio y el control de información personal, así como para la sincronización de bases de datos. A continuación se describen algunas que serán útiles en el desarrollo de la investigación.

#### 1.2.3.1 Altamira Employees

*Altamira Employees* es un *software* de personal altamente personalizable desarrollado para la administración de recursos humanos, dando la flexibilidad de diseñar la plataforma HRM<sup>2</sup>, según los requerimientos de cada organización. Toda la información importante sobre un empleado del personal (datos personales, posiciones, beneficios, retribuciones, etc.) puede ser rastreada históricamente a través de una base de datos online que permite rastrear el historial de toda la información del empleado y así se pueden conocer las diferentes versiones realizadas a dicho empleado. (Altamira 2011)

Este sistema es una potente solución al problema de la gestión de la información del personal, pero presenta algunas violaciones de las reglas del negocio dentro del SGU, porque los empleados de *Altamira Employees* pueden acceder a la plataforma a través de sus cuentas particulares, visualizar y actualizar sus datos. Para el desarrollo de la investigación estos servicios son perjudiciales porque el personal no

---

<sup>2</sup> HRM (*Human Resource Management*, o sus siglas en español Gestión de Recursos Humanos).



debe ser capaz de actualizar sus datos personales, ya que la información versionada debe ser única en el sistema y solamente actualizada por el personal correspondiente.

### 1.2.3.2 DSR Gestión del Personal

Es un *software* desarrollado por *Dataprocess Soluciones Registradas (DSR)* que proporciona medios fáciles y flexibles para organizar y controlar los datos del empleado. En su base de datos la información almacenada en cada una de las fichas de empleado contiene, tanto los datos personales del empleado como varios códigos de estados administrativos. Entre estos códigos existe uno para contratos de empleo, que puede utilizar para asignar niveles o contratos de empleo estandarizados. La información confidencial de un empleado se controla a través de una tabla asociada a la ficha del empleado. Esto le permite dar un acceso relativamente amplio a la información básica del archivo del empleado y poner unas restricciones de seguridad más rigurosas en los datos confidenciales. (DSR 2009)

Este sistema soluciona el problema de la gestión de la información personal, dividiendo la misma en confidencial y básica, pero tiene la contradicción que es un *software* propietario, y para adquirirlo es necesario pagar elevados costos. Además es una solución general para el sector empresarial que no puede ser adaptada a las características del SGU con la facilidad y eficiencia de un *software* desarrollado específicamente para el mismo.

### 1.2.3.3 La base de datos Cassandra

Es una base de datos no relacional distribuida y basada en un modelo de almacenamiento de clave-valor desarrollada por la *Apache Software Foundation*. La arquitectura de Cassandra es descentralizada porque cada uno de los servidores posee exactamente la misma información y cumple las mismas funciones que los demás, es diferente al modelo administrador/esclavos en el cual si el administrador deja de funcionar toda la base de datos esta vulnerable a cualquier fallo, además, Cassandra implementa un protocolo P2P<sup>3</sup> y presenta una lista de todas las máquinas, tanto las que están activas, como las que no. (Herrera 2012) Cassandra permite añadir nodos a la base de datos de forma horizontal, es decir, lo añadido y se acopla, sin más, se añade un nodo al clúster y ya se encarga el sistema de replicar los datos y mantener la consistencia. Si pasa lo contrario, que se quita o se pierde un nodo de un clúster también se recompone automáticamente para sostener la caída de ese servidor.

---

<sup>3</sup> El término P2P es el acrónimo de la frase en inglés "Peer to Peer" que en español es "par a par" o punto a punto, esto quiere decir que es un tipo de red que no tiene un clientes ni servidores fijos, en cambio se compone de una serie de nodos que se comportan simultáneamente como clientes y como servidores respecto de los demás nodos de la red





Para sincronizar los nodos, Cassandra utiliza la técnica *Consistent Hashing*, que gestiona el uso de nodos para grabar u obtener datos y seleccionar el nodo que almacenará la información en primer lugar, para luego replicar los datos al resto de servidores. La idea consiste en que se sitúan todos los nodos en el perímetro de una circunferencia en la que la posición la define un valor entre 0 y 1, y a partir del valor devuelto por una función que calcula valores en este rango, y que puede ser aleatoria o no, se decide que nodo va a atender la lectura o escritura solicitada. Una vez escrito el registro, el nodo se sincroniza con el siguiente, y este a su vez con el siguiente, así hasta que todos han sido actualizados (**Ver Figura 2**). En esta línea también se soporta autoreparado de datos inconsistentes tras una consulta. Esto es, devuelvo lo que me piden y aviso al resto para que me sincronicen, si el dato estaba mal se corrige, con lo que los datos más consultados siempre son consistentes.(Celano 2009)

El análisis de la técnica antes mencionada es una fuente de estudio importante para el desarrollo de la investigación, ya que para realizar el versionado de la información personal es necesario que la sincronización entre las bases de datos del SGU sea la mejor posible.

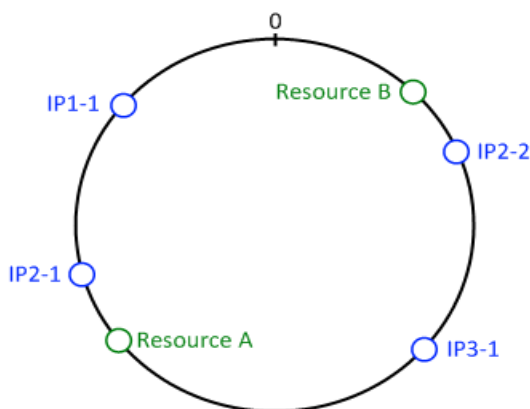


Figura 2. Sincronización de los nodos en la base de datos Cassandra.

#### 1.2.3.4 Sistema Integral de Gestión Empresarial Cedrux (ERP Cubano)

Cedrux conocido también como ERP<sup>4</sup> cubano, actualmente está siendo desarrollado por un equipo multidisciplinario, compuesto por especialistas de diferentes entidades del país y la UCI. En su proyección inicial se muestra como un paquete de soluciones integrales de gestión, basado en los principios de independencia tecnológica y en la inclusión de las funcionalidades generales presentes en los procesos

<sup>4</sup> ERP (*Enterprise Resource Planning*), en español sistemas de planificación de recursos empresariales son sistemas de gestión de información que automatizan muchas de las prácticas de negocio asociadas con los aspectos operativos o productivos de una empresa.



característicos de la economía cubana. En su implementación inicial se desarrolló para las áreas de contabilidad, costos y procesos, capital humano, planificación financiera y logística, todo construido dentro de una sola plataforma de negocios.

El área de capital humano es la responsable de gestionar todos los procesos que interactúan con una persona en el sistema. En esta área la información personal está dividida en varias tablas, en una almacenan la información básica definida para su sistema y en el resto de las demás los atributos que son complementarios para una persona. A diferencia del SGU, el sistema Cedrux tiene almacenada la información personal solamente en una base de datos, propiciando así una centralización de la información personal y reduciendo los problemas que se emiten en la investigación. La arquitectura de datos implementada en Cedrux es similar a la del SGU, pero se diferencian en que Cedrux no gestiona información personal en cada una de sus áreas, haciendo más fácil la gestión y el control de la información personal. Otra diferencia que posee con respecto al SGU es que al no tener diferentes áreas con información personal, no presenta los problemas de tener una persona con información diferente en una de ellas. No obstante, Cedrux sirve de apoyo a la investigación como fuente de estudio en el control de la información de las personas, por las características similares vistas en su base de datos de personal.

#### **1.2.4 Resultado del análisis de las soluciones existentes**

Después de analizar algunas de las soluciones existentes, se puede apreciar que las mismas tienen amplias facilidades en el control de la información de los datos personales. Como resultado de cada uno de los sistemas analizados, se determinó que las funcionalidades que tienen incorporadas no cumplen con las expectativas para el SGU, debido a que algunas no están enfocadas en la centralización de los datos sensibles de una persona (entiéndase a datos que no deban cambiar), los cuales deben ser versionados de forma controlada. Sin embargo, constituyen una importante fuente de estudio, pues presentan características que se acoplan a las necesidades del versionado de la información personal como son los procesos de gestión eficiente de la información de una persona y la sincronización de bases de datos. Además, no se pueden utilizar ya que las soluciones por sí sola no logran juntar ambos procesos al mismo tiempo y no son fáciles de integrar a la arquitectura definida para el SGU. Basándose en las limitaciones, con vista a mejorar el proceso de versionado de información personal, se necesita una solución propia que responda específicamente a las necesidades de la información personal dentro del SGU.

#### **1.3 Necesidad del versionado de la información personal para el SGU**

El SGU está necesitado de un control eficiente de los cambios que originan los proveedores de información personal, para que la misma no esté duplicada en algunas de las áreas que pertenecen a dicho sistema. La sincronización de una base de datos con las demás, es la actividad fundamental que se



debe garantizar durante el versionado de la información personal. Bajo esta situación se hace necesaria la vinculación de una serie de métodos, estrategias y procedimientos que den paso a la elaboración de mecanismos que garanticen la sincronización de la información personal en las bases de datos pertenecientes al SGU.

Ante esta inminente necesidad surge la idea de versionar la información personal en el SGU, con el objetivo de controlar los cambios que se realizan en algunas de sus áreas a través del sincronizador de bases de datos, registrando posteriormente la versión que se crea de la información personal y contribuyendo así a no afectar el resto de los procesos universitarios.

## 1.4 Herramientas, tecnologías y lenguajes para el desarrollo

Con el objetivo de lograr el producto antes mencionado, que no sólo solucione los problemas existentes, sino que además logre unificar calidad y eficiencia, se realiza una caracterización de las herramientas, lenguajes y tecnologías a utilizar en la investigación, las cuales fueron definidas por las políticas de desarrollo del Centro de Informatización Universitaria (CENIA).

### 1.4.1 Herramientas de desarrollo

#### 1.4.1.1 Herramienta Case

##### ➤ Visual Paradigm 8.0

Visual Paradigm es un tipo de herramienta CASE (*Computer Aided Software Engineering*) para realizar el modelado de una base de datos, que sirve de soporte para el modelado de UML. Proporciona un diagramación visual y el diseño de sus proyectos, esto ayuda a crear de manera rápida aplicaciones de calidad y a un menor costo. Facilita el código y compatibilidad hasta con diez lenguajes, permitiendo realizar tanto ingeniería inversa como directa. Proporciona la generación de bases de datos y la transformación de diagramas de Entidad-Relación en tablas de ellas. (Rozic 2004)

Es una herramienta multiplataforma y útil para la generación de código fuente en PHP<sup>5</sup>. Resiste múltiples usuarios trabajando sobre el mismo trabajo y ofrece la posibilidad de que los mismos perciban los cambios hechos en tiempo real. Su licencia es gratuita y comercial.

#### 1.4.1.2 Herramienta de prototipado

##### ➤ Evolus Pencil 1.3.5

Es una aplicación de animación y dibujo multiplataforma que corre tanto en *MacOS*, en *Windows* y en *Linux*. Permite crear de forma fácil y rápida *wireframes* y prototipos de navegación. Estos se usan en el

---

<sup>5</sup> PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje de programación del lado del servidor, de código abierto muy útil para diseñar de forma rápida y eficaz.



proceso de diseño de una página web y se pueden realizar desde el *Firefox*, solo se debe dibujar o arrastrar los elementos (botones, cajas de búsqueda o texto) dentro del lienzo o página. (group 2010)

Características principales:

- Construcción de prototipos.
- La exportación a HTML, PNG, documento Openoffice.org, documento de Word y PDF.
- Instalación de plantillas definidas por el usuario.
- Las operaciones de dibujo estándar: alineación, escalar y rotar.
- La adición de los objetos externos.

### 1.4.1.3 Administrador de base de datos

#### ➤ PgAdmin III 1.10.2

PgAdmin es una herramienta de código abierto para diseñar, mantener y administrar las bases de datos de PostgreSQL. Está diseñado para responder a las necesidades de los usuarios, como la escritura de consultas SQL para crear bases de datos complejas. PgAdmin III está disponible en más de una docena de idiomas y es una herramienta libre distribuida bajo la licencia de PostgreSQL. La aplicación contiene un editor SQL con resaltado de sintaxis, cuenta con un editor de código de la parte del servidor, se puede lanzar scripts programados.

Presenta una excelente ayuda, con pantalla de información, tablas, índices, secuencias, vistas, programas de arranque, funciones y lenguajes. Permite configurar usuarios, grupos, privilegios a través de preguntas y respuestas. (Pgadmin 2010)

### 1.4.1.4 Entorno de Desarrollo Integrado (IDE)

#### ➤ NetBeans 7.2

NetBeans es un Entorno de Desarrollo Integrado de código abierto y gratuito para los desarrolladores de *software*. Ofrece todas las herramientas necesarias para crear aplicaciones de escritorio profesionales, empresariales, web y móviles con el lenguaje Java, JavaFX, C/C ++ y lenguajes dinámicos como PHP, JavaScript, Groovy y Ruby. (Pedraza 2009)

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de *software* llamados módulos. NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de cien socios en todo el mundo. *Sun Microsystems* fundó el proyecto de código abierto NetBeans en junio del 2000 y continúa siendo el patrocinador principal de los proyectos. Es considerado el IDE estándar en el desarrollo



con Java, por lo que su constante renovación y mejora es una prioridad en la comunidad mundial que colabora con este lenguaje.

#### 1.4.1.5 Sistema Gestor de Bases de Datos (SGDB)

##### ➤ PostgreSQL 8.4.1

PostgreSQL es un SGBD relacional de *software* libre. Contiene características de la orientación a objetos, como puede ser tipos de datos, herencia, funciones, restricciones, disparadores, reglas e integridad transaccional. Está disponible bajo una Licencia BSD (*Berkeley Software Distribution*), puede tener uso tanto comercial como no comercial, lleva más de 15 años desarrollándose y su arquitectura goza de una excelente reputación por su fiabilidad e integridad de los datos. Presenta valiosas características como permite la replicación asincrónica, transacciones anidadas, copias de seguridad en caliente, un sofisticado planificador/optimizador de consultas, un registro nativo de escrituras anticipadas para ser tolerantes a fallos de *hardware* y presenta un control concurrente multiversión (MVCC). Actualmente es el más utilizado por la comunidad de *software* libre después de MySQL, aunque presenta una estructura superior a este y propone más funcionalidades, es escalable tanto en la cantidad bruta de datos que puede manejar como en el número de usuarios concurrentes que puede atender. (Rafaelma 2004)

#### 1.4.1.6 Servidor Web

##### ➤ Apache 2.2

Es un servidor HTTP<sup>6</sup> (*Hypertext Transfer Protocol*) de código abierto. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. Es uno de los servidores web más utilizado en el mundo. Su costo es gratuito, presenta gran fiabilidad y extensibilidad convirtiéndose así en una herramienta potente y muy fiable. Es un *software* que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo y gracias a ser modular se han desarrollado diversas extensiones entre la que se destaca PHP. (Apache 2009)

---

<sup>6</sup> Es el Protocolo de transferencia de hipertexto (HTTP) que permite la transferencia de archivos entre un navegador (el cliente) y un servidor web, localizado mediante una cadena de caracteres denominada dirección URL.



## 1.4.2 Lenguajes utilizados

### 1.4.2.1 Lenguaje de modelado

#### ➤ UML 2.0

Lenguaje Unificado de Modelado o *Unified Modeling Language* (UML, por sus siglas en inglés), es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema. Brinda un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocios, funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de *software* reutilizables. Es la sucesión de una serie de métodos de análisis y diseño orientados a objetos. Está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de *software* previo al proceso intensivo de escribir código. (EIDV 2012)

### 1.4.2.2 Lenguajes de programación

#### ➤ PHP 5.3

PHP (acrónimo de "PHP: *Hypertext Preprocessor*") es un lenguaje de programación del lado del servidor, de código abierto muy útil para diseñar de forma rápida y eficaz. Es un poderoso lenguaje de secuencia de comandos diseñado específicamente para permitir a los programadores crear aplicaciones web con distintas prestaciones de forma rápida. Una de sus características más potentes es el soporte para varios gestores de bases de datos entre las que se pueden mencionar InterBase, Microsoft SQL, MySQL, Oracle, Informix, PostgreSQL. Es *software* libre, por lo que es menos costoso utilizarlo. Lenguaje multiplataforma que funciona tanto para *Unix* (con *Apache*) como para *Windows* (con *Microsoft Internet Information Server*). Es soportado por cualquier navegador, pero para que sus páginas funcionen tiene que ser soportado por el servidor. Permite aplicar técnicas de programación orientada a objetos, no requiere definición de tipos de variables y posee manejo de excepciones desde PHP5. (Hinostraza 2005)

Esta versión ha sido re-escrita por completo, logrando un soporte integrado con SOAP<sup>7</sup> (*Simple Object Access Protocol*) y mejoras en el soporte para la programación orientada a objetos.

---

<sup>7</sup> Es un protocolo de comunicación donde dos objetos en diferentes procesos se comunican a través de un intercambio de datos usando archivos de tipo XML. Utiliza cualquier protocolo que permita transportar mensajes de texto, siendo HTTP el más utilizado.



### ➤ JavaScript 1.2

Es un lenguaje de programación interpretado, se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. A pesar de su nombre, no guarda ninguna relación directa con el lenguaje de programación Java, tienen semánticas y propósitos diferentes.

Los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM (*Document Object Model*).

JavaScript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, con funciones y estructuras de datos complejas. Además, pone a disposición del programador todos los elementos que forman la página web, para que este pueda acceder a ellos y modificarlos dinámicamente.(Pérez 2009)

### ➤ XML

El Lenguaje de Marcado Extensible (*Extensible Markup Language, XML*) es un metalenguaje que permite definir lenguajes de marcado adecuados a usos específicos. Aunque a primera vista un documento XML puede parecer similar a HTML hay una diferencia fundamental: un documento XML contiene datos que se autodefinen, o sea, no posee etiquetas prefijadas con anterioridad, ya que es el propio diseñador el que las crea, dependiendo del contenido del documento. En XML se separa el contenido de la presentación de forma total.(Melián 2009)

Entre sus ventajas se encuentra su aceptación casi universal, su legibilidad y su carácter auto-contenido, si bien el tamaño de los documentos XML es mayor que el de sus equivalentes binarios y su procesamiento requiere más recursos, por lo que no resulta adecuado en aplicaciones en las que la eficiencia sea un objetivo prioritario.

### 1.4.3 Marco de trabajo

#### **Gestión Universitaria Universo Digital Gestión Documental (GUUD)**

GUUD es un marco de trabajo propuesto por el equipo de arquitectura del departamento de Gestión Universitaria del CENIA. El mismo integra a su vez los marcos de trabajo CodeIgniter en su versión 1.7.3 y JQuery 1.3.2 en una sola infraestructura, razón por la cual posee las mismas características que estos. En



esta integración se incluyen mejoras y modificaciones hechas específicamente al CodeIgniter, además de creación de *plugins* y componentes de interfaz de usuario en JQuery.

#### ➤ **Del lado del servidor: CodeIgniter 1.7.3**

CodeIgniter es un marco de trabajo para el desarrollo de aplicaciones Web usando PHP. Este permite el desarrollo de proyectos mucho más rápidos que si se escribiera código desde cero. Provee una rica colección de librerías para las tareas necesarias más comunes. Permite concentrarse en el desarrollo del proyecto en cuestión, minimizando la cantidad de código necesaria para realizar las tareas. Usa el patrón de diseño arquitectónico Modelo-Vista-Controlador como paradigma de arquitectura de desarrollo, el cual separa en tres capas distintas: la representación de datos, el interfaz de usuario y el controlador de eventos respectivamente. (Alvarez 2009)

#### ➤ **Del lado del cliente: jQuery 1.3.2**

JQuery es un nuevo tipo de biblioteca o marco de trabajo de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, permitiendo manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX al sistema. JQuery, al igual que otras librerías, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código. Es decir, con las funciones propias de esta librería se logran grandes resultados en menos tiempo y espacio. La gran ventaja es que permite cambiar el contenido de la página Web sin necesidad de recargarla, utilizando DOM y AJAX de manera extremadamente sencilla gracias a su sintaxis. (García Vidal 2012)

### **1.5 Proceso de desarrollo de software**

La UCI actualmente se encuentra inmersa en un proceso de mejora de los procesos que se desarrollan como parte de la construcción del *software* basado en el modelo de CMMI (*Capability Maturity Model Integration*, de su significado en español Integración de Modelos de Madurez de Capacidades). El proceso de mejora está encaminado a que la universidad alcance una certificación internacional del nivel 2 del modelo CMMI. Este hecho la convertiría en la primera empresa cubana certificada con este modelo, el cual es de referencia para el crecimiento de capacidades y madurez, que se enfoca tanto en procesos de Administración como de Ingeniería de Sistemas y *Software*.

#### **1.5.1 Proceso de desarrollo con enfoque ágil basado en el nivel 2 de CMMI**

El CENIA con el objetivo de mejorar y fortalecer su cultura de calidad en el desarrollo de *software*, lleva a cabo este proceso utilizando el conjunto de prácticas propuestas por el modelo CMMI para el nivel 2 fundamentalmente en el área de proceso Administración de Requisitos (REQM) y algunas prácticas de las metodologías ágiles XP (Programación Extrema) y Scrum. (Castro Llano 2012)





## **CMMI**

CMMI es un modelo de referencia para el crecimiento y evaluación de capacidades y madurez de una organización, se enfoca tanto en procesos de administración como en los de ingeniería de sistemas y *software*. Parte de las prácticas actuales y logra una mejora continua a través de una cultura de procesos.

## **XP (Programación Extrema)**

Programación Extrema es una metodología ágil de desarrollo de *software* que posee cuatro tareas fundamentales: planificación, diseño, desarrollo y pruebas. Esta metodología está basada en la simplicidad durante el desarrollo, la comunicación entre las partes implicadas (clientes y desarrolladores) y la retroalimentación para poder reutilizar el código desarrollado. En su concepción establece entregas frecuentes con posibilidad de refactorización continua, permitiendo mejorar el diseño cada vez que se añade una funcionalidad.

## **Scrum**

Scrum es una metodología ágil enfocada a la gestión de proyectos. Sus principales características se pueden resumir en dos: el desarrollo de *sprint* o iteraciones y reuniones a lo largo del desarrollo. Las iteraciones en Scrum tienen una duración máxima de 30 días y el resultado de cada uno de ellas define un incremento del producto a desarrollar. La evolución del proyecto por la metodología se define a través de reuniones diarias donde el trabajo del día anterior es revisado por el equipo, previendo además la labor a realizar el día siguiente.

CMMI permite especificar qué deben hacer los procesos para obtener y entregar productos y servicios de calidad mientras que el enfoque ágil dado por las prácticas de las metodologías antes mencionadas permite describir como llevarlo a cabo.

## **1.6 Conclusiones parciales**

En este capítulo fueron abordados diferentes conceptos y términos que permiten una mejor comprensión del dominio de la investigación. Se llevó a cabo un estudio sobre las técnicas de particionado de tablas, seleccionándose el particionado vertical, por ser la que más se ajusta a las necesidades del versionado de la información personal.

Se planteó que las soluciones existentes que controlan información personal cumplen parcialmente con las características que requiere el proceso de versionado de la información personal del SGU, pero no son suficientes, por tanto se ve la necesidad de desarrollar una solución informática que solucione los problemas existentes. Se caracterizaron diversas herramientas útiles para el desarrollo de la solución y se describió el proceso de desarrollo de *software* que guía la investigación.



# 2

## CAPÍTULO 2

## Descripción y análisis de la solución

### 2.1 Introducción

En este capítulo se abordan temas sobre los principales procesos que se llevan a cabo para establecer la estrategia de solución para el versionado de la información personal en el Sistema de Gestión Universitaria (SGU). Para que la estrategia sea satisfactoria y cumpla con las expectativas, primeramente es vital el análisis del modelo de dominio para una mejor comprensión de la solución propuesta. Luego se definen los procesos mediante un mapa conceptual y se explican los requerimientos funcionales y no funcionales. Además se realiza un estudio de la arquitectura y el diseño a utilizar, y se muestra el diagrama lógico de datos o diagrama de clases persistentes con la descripción de las tablas de la base de datos.

### 2.2 Modelo de dominio

El modelo de dominio es una representación visual de clases conceptuales o de objetos reales en un marco de interés, que consiste en un conjunto de diagramas de clases, sin definición de operaciones, donde se pueden representar los atributos de las clases, las relaciones entre las entidades que interactúan en el dominio del problema así como su cardinalidad.(Synergix 2010)

Para la mejor comprensión de la propuesta de solución planteada, se propone realizar un modelo de clases del dominio (**Ver Figura 3**), donde estarán representadas las clases conceptuales identificadas. Con este modelo se pretende contribuir a la comprensión de la solución, esquematizando las entidades presentes que permitirá dejar claro el entorno y el alcance de la propuesta de solución para el versionado de la información personal en el SGU.

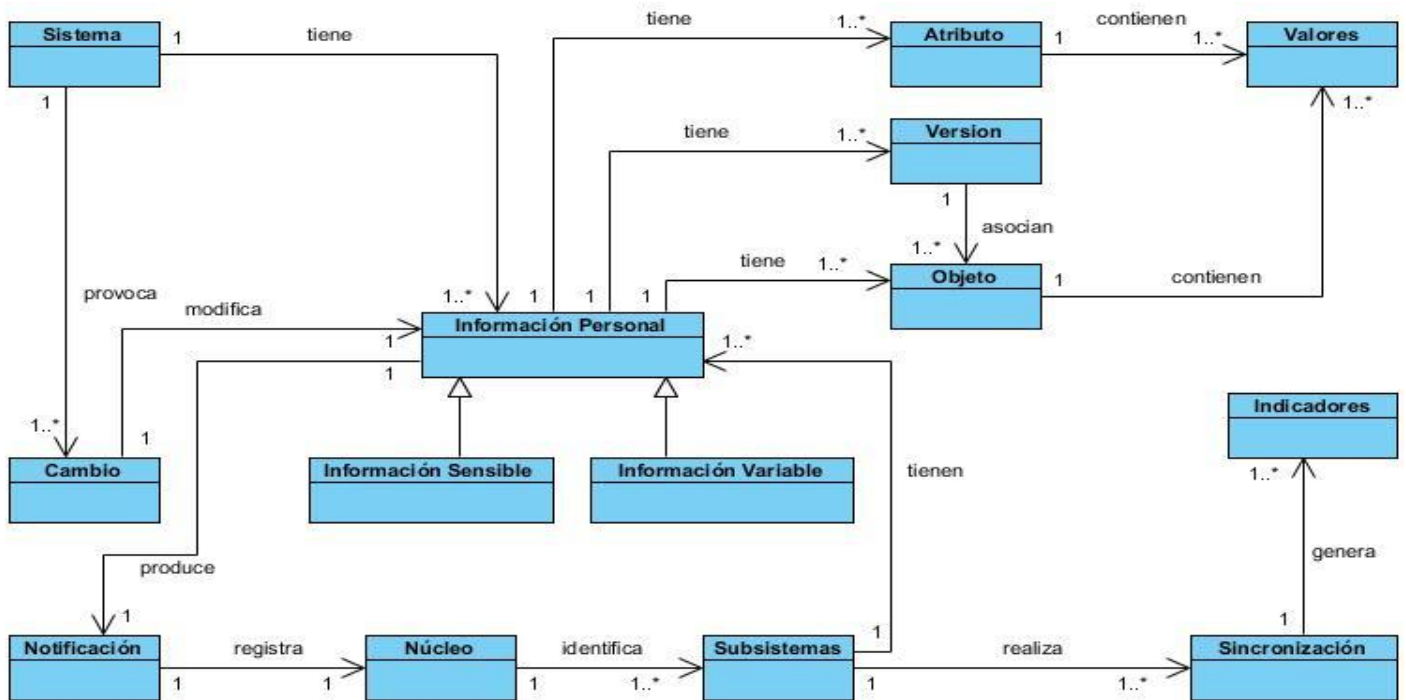


Figura 3. Modelo de dominio de la solución.

### Descripción de las clases del modelo de dominio

Para comprender las clases que interactúan en el modelo de dominio se realizó una descripción de las mismas con el objetivo de caracterizarlas de acuerdo a su funcionamiento dentro de la solución propuesta. El **Anexo 1. Descripción de las clases del dominio**, muestra esta caracterización.

### 2.3 Mapa conceptual

Los mapas conceptuales son un instrumento de representación del conocimiento, que dan una idea clara de conceptos complejos que facilitan la enseñanza-aprendizaje de una persona. En él, se descompone el espacio del problema en unidades comprensibles (conceptos). Se puede ver como un modelo que relaciona cuáles son los términos importantes y cómo se relacionan. Para explicar mejor el proceso de versionado de la información personal se llevó a cabo el siguiente mapa conceptual:

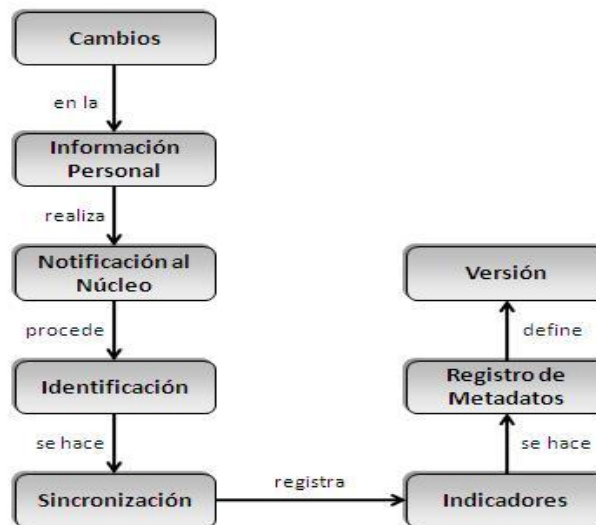


Figura 4. Mapa conceptual.

## 2.4 Propuesta de solución

Por los problemas existentes en la gestión universitaria con el manejo de la información personal, se decide crear una solución con técnicas de sincronización entre bases de datos para asegurar la veracidad de la información. El resultado a obtener será un conjunto de funcionalidades a nivel de aplicación combinadas con otras a nivel de dato, constituyendo una eficiente solución para versionar la información personal y así poder controlar los cambios que se realizan. La solución va a nutrirse de la importancia que tenga la información personal en los procesos universitarios que integran el SGU. Con ella, se pretende lograr una mejora en la manipulación de la información que pueda afectar a los subprocesos que dependan de los datos almacenados, con tiempos de respuesta admisibles y con una centralización de la información personal. **Ver Anexo 2. Tabla Resumen de la información personal sabana.**

En la propuesta de solución, cuando se realiza un cambio en la información personal se procede a controlar el cambio en todas las áreas inscritas del SGU. Para ello se ejecutan notificaciones con el objetivo de registrar en el Núcleo los cambios realizados. Luego se realiza la identificación de la sincronización, donde se verifica cuáles bases de datos se afectan y cuáles de ellas se pueden actualizar. Para el proceso de identificación se realizó un particionado de tablas en las áreas del SGU, donde se identifica por cada uno de ellas la información sensible y variable según el negocio de cada área, permitiendo así que a la hora de sincronizar los cambios se conozcan qué bases de datos se pueden sincronizar. Al realizarse la identificación, se procede a ejecutar la sincronización en las bases de datos, de donde se genera un registro de indicadores de sincronización, en los que se archivarán la fecha en



que se realiza el cambio, la persona a la que se actualizó, así como otros parámetros que pueden ser útiles para mantener el control de la información personal en las bases de datos pertenecientes al SGU. Posteriormente, si la información personal es sensible, entonces se realiza el versionado, para ello se almacena en un registro de metadatos los datos requeridos para versionar una información personal y luego se identifica la numeración de la versión a utilizar. Si la información personal es variable y se requiere versionar, entonces se realiza el proceso anterior, ya que esta última parte de la solución puede ser aplicable a cualquier tipo de información que se quiera versionar, pues asegura que se puede almacenar en cualquier tiempo determinado.

## **2.5 Sincronización en las áreas del SGU**

El proceso de sincronización entre las bases de datos del SGU es una de las principales técnicas que garantiza el cumplimiento del objetivo planteado para la construcción de la solución. Como se menciona anteriormente, la sincronización en la solución parte del particionado de tablas realizado a la información personal, de donde se identifica la información sensible y variable. Primeramente se debe aclarar que todos los cambios en la información personal se sincronizan, sin tener que conocerse el tipo de información, la única diferencia que se realiza en el proceso es que cuando se termina de sincronizar una información sensible, siempre se procede a versionar la información personal, en cambio, si la información es variable solo se procederá a versionar si se desea almacenar una instancia de ella en ese tiempo transcurrido. Para ejecutar la sincronización se identificaron un conjunto de negocios donde se almacena la información personal clasificada en grandes grupos como: información nominal, información de superación, atributos físicos, entre otros. Cada uno de estos negocios recibe una consulta con los cambios que aún no se han sincronizado y este se encarga de dividir la información según su tipo y ejecutarlo en cada área. A medida que se ejecuta la sincronización de cada resultado, se va almacenando en un registro los indicadores de sincronización. En este registro (que no es más que una tabla en la base de datos) se guarda la persona que se sincronizó, el proveedor del área, y el identificador de la información sincronizada. Este registro permite tener almacenado detalles de la sincronización para así mantener un control de la información personal que se sincroniza.

## **2.6 Relación de la propuesta de solución con el SGU**

El SGU está compuesto por doce áreas que identifican la gestión universitaria dentro de la Universidad de las Ciencias Informáticas (UCI). La solución propuesta para el versionado de la información personal en principio integrará a nivel de dato a cinco áreas (Núcleo, Pregrado, Postgrado, Cooperación y Residencia), y con el resto a medida de que se vayan desarrollando las mismas. En cada una de las bases de datos estarán alojadas todas las estructuras a nivel de dato que comprenden al versionado de la información



personal, con el objetivo de que cada área pueda versionar su propia información sin necesidad de acudir a otra. Las funcionalidades a nivel de dato son extensibles a cualquier área, permitiendo que a medida que se vayan desarrollando los restantes se incorporen con extrema facilidad. Las funcionalidades a nivel de aplicación con que cuenta la solución se alojarán en el módulo de Configuración, el cual es el encargado de gestionar la información necesaria para realizar las configuraciones del SGU.

La solución estará implementada de forma tal, que el administrador de bases de datos tendrá que escoger el área donde desee versionar una información personal, asegurando así, que se comunique con cada una de ellas y pueda realizar las operaciones desde un mismo lugar. El siguiente mapa taxonómico permite conocer cómo se agrupan las funcionalidades de la solución según las opciones o procesos que gestionan dentro del SGU.

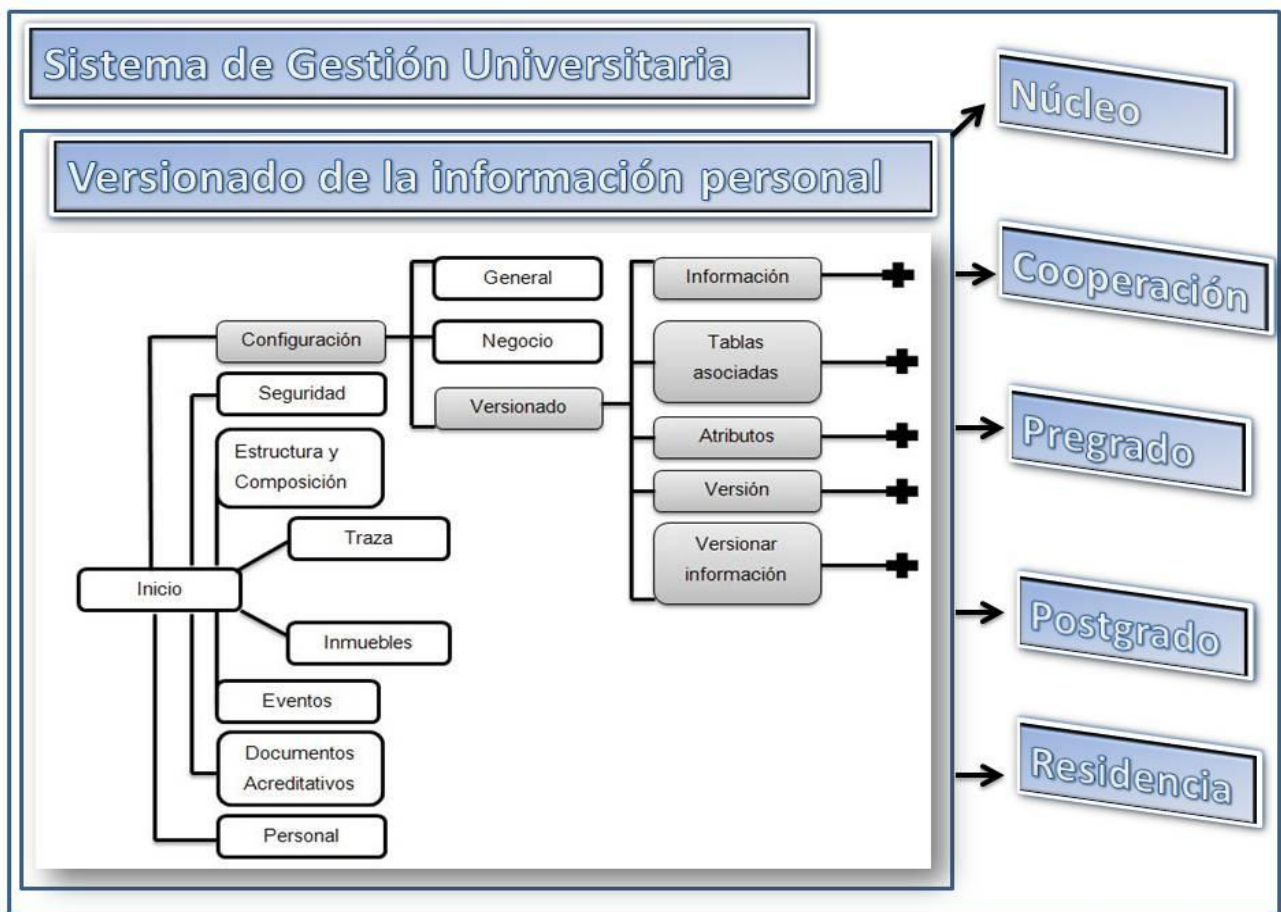


Figura 5. Mapa taxonómico.

## 2.7 Estrategia para el levantamiento de requerimientos

La obtención de requerimientos es una de las cuatro actividades que define la Ingeniería de Requisitos, que ayudan a reconocer la importancia que tiene para el desarrollo de un proyecto de *software* la



realización de una especificación y administración adecuada de los requerimientos de los usuarios. La obtención de requerimientos es el proceso mediante el cual los interesados en un sistema de *software* descubren, revelan y entienden sus requerimientos. (Pressman 2007). Hay muchas técnicas para identificar los requerimientos, dentro de las que se encuentran: sesiones de tormentas de ideas, entrevistas, cuestionarios, observación de campo, revisión de la documentación técnica, análisis de mercado, ingeniería inversa, simulaciones y prototipos. De las técnicas antes mencionadas, para la captura de requerimientos de la solución propuesta se utilizaron las siguientes:

**Entrevistas:** se realizaron reuniones analista-interesado, donde se realizaron una serie de preguntas, para lograr una mayor comprensión de la problemática existente, y a partir de ahí poder identificar los requerimientos de la solución. **Ver Anexo 3. Modelos de Entrevista.**

Con las entrevistas realizadas se lograron obtener diversas opiniones y recomendaciones de los especialistas, quienes dieron importantes aportes e ideas que ayudaron posteriormente a la realización de la solución final. Los entrevistados mencionaron de manera general los beneficios que se podrán obtener al aplicar la propuesta desarrollada, dentro de los que se encuentran:

- Mayor control de la información: se podrá mantener un mayor control sobre los cambios que ocurran sobre la información sensible de la persona en el SGU.
- Centralización de los datos: permitirá mantener centralizada toda la información personal.
- Actualización de la información: se podrá mantener actualizada la información, permitiendo una mejora en los índices de veracidad.

**Sesiones de tormentas de ideas (*Brainstorming*):** se realizaron reuniones en grupo, con la participación del cliente y el resto de los interesados, con el objetivo de generar ideas respecto a las funcionalidades e integración de la solución al SGU.

## 2.8 Especificación de requerimientos de software

La especificación de requerimientos es la base que permite verificar si se alcanzaron o no los objetivos establecidos en el proyecto, ya que estos son un reflejo detallado de las necesidades de los clientes o usuarios del sistema. (Scribd inc 2012)

La especificación de requerimientos es una descripción completa del comportamiento del sistema que se va a desarrollar. En esta etapa se debe generar una información clara y precisa de los aspectos más relevantes del producto, ya que esta actividad es el hilo conductor de todo el desarrollo del *software*. A continuación se describen los requerimientos funcionales y no funcionales que fueron definidos para el versionado de la información personal.



### 2.8.1 Requerimientos funcionales

Los requerimientos funcionales de un sistema describen lo que el sistema debe hacer. Estos requerimientos dependen del tipo de *software* que se desarrolle, de los posibles usuarios del mismo y del enfoque general de la organización tomado al redactar los requerimientos.(Sommerville 2005). A continuación en la **Tabla 1** se muestran los requerimientos funcionales definidos para la propuesta de solución con la complejidad calculada a través de los criterios definidos en el documento para realizar la *Evaluación de Requisitos de Software*.

**Tabla 1. Requerimientos funcionales del sistema.**

Nombre	Requerimiento funcional	Complejidad
RF1	Identificar cambio en la información sensible.	Alta
RF2	Identificar quién cambió la información sensible.	Alta
RF3	Identificar cuál es la información sensible.	Alta
RF4	Notificar el cambio.	Alta
RF5	Identificar prioridad de sincronización.	Alta
RF6	Sincronizar información al Núcleo.	Alta
RF7	Sincronizar información a los sistemas.	Alta
RF8	Identificar indicadores de sincronización.	Media
RF9	Registrar indicadores de sincronización.	Media
RF10	Listar información.	Media
RF11	Ver detalles de información.	Media
RF12	Crear información.	Alta
RF13	Modificar información.	Media
RF14	Modificar filtros a información.	Media
RF15	Listar tablas asociadas a la información.	Alta
RF16	Ver detalles de tablas asociadas a la información.	Media
RF17	Crear tablas asociadas a la información.	Media
RF18	Modificar tablas asociadas a la información.	Media
RF19	Listar atributos de la información.	Alta





<b>RF20</b>	Ver detalles de atributos de la información.	Media
<b>RF21</b>	Crear atributos de la información.	Media
<b>RF22</b>	Modificar atributos de la información.	Media
<b>RF23</b>	Listar versión de una información.	Alta
<b>RF24</b>	Ver detalles de versión de una información.	Media
<b>RF25</b>	Listar informaciones versionadas.	Media
<b>RF26</b>	Versionar información.	Alta
<b>RF27</b>	Asociar la versión y el atributo de una información.	Alta

### 2.8.1.1 Descripción de los requerimientos funcionales

En la **Tabla 2** se muestra un ejemplo de la especificación del requerimiento *Crear información* que corresponde a una de las funcionalidades más importantes de la solución, la cual tiene una complejidad *Alta*, por lo que es arquitectónicamente significativa. El resto de las descripciones de los requerimientos funcionales se pueden encontrar en el **Anexo 4. Especificaciones de requerimientos**.

**Tabla 2. Especificación de requerimientos. Crear información.**

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
<b>RF12</b>	Crear información	<p>Permite registrar la información a versionar.</p> <p>El administrador selecciona el módulo Configuración del SGU y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Información.</p> <p>Se escoge el sistema sobre el cual se va a crear la nueva información, y se muestra un listado con las informaciones de esa área.</p> <p>En el área de iconos flotantes se selecciona la opción Crear.</p> <p>Para crear la información se introducen los datos: Nombre, Descripción y Estado (Habilitado/Deshabilitado).</p> <p>Cuando se crea una nueva información se muestra el mensaje "El elemento ha sido</p>	Alta	Alta



		creado satisfactoriamente". En el área de iconos flotantes se muestran las opciones de Listar y Ayuda.	
<b>Prototipo</b>			
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>
	Nombre	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Admite letras, números y guión bajo.</li> <li>✓ Admite entre 3 y 250 caracteres.</li> </ul>
	Descripción	Varchar	<ul style="list-style-type: none"> <li>✓ Admite entre 0 y 1000 caracteres válidos.</li> <li>✓ Admite letras, números, espacios y guión bajo</li> <li>✓ Solo admite 30 caracteres por palabra.</li> </ul>
	<b>Observaciones</b>		

### Plan de iteración

El Plan de iteración tiene como entrada las especificaciones de requerimientos descritas anteriormente. En el mismo se establecen las iteraciones necesarias para desarrollar el producto y se definen qué requerimientos se deben implementar en cada una de estas. El objetivo de cada iteración es obtener una versión funcional del sistema que, aunque no cuente con todos los requerimientos definidos, constituye un resultado de valor para el cliente. **Ver Anexo 5. Plan de iteraciones.**



### 2.8.2 Requerimientos no funcionales

Los requerimientos no funcionales son aquellos que se refieren a las propiedades o cualidades que el producto debe tener. Son características que hacen al producto atractivo, usable, rápido o confiable. Los requerimientos no funcionales de la solución fueron clasificados, en la **Tabla 3** se puede encontrar el resultado de esa clasificación. De manera general las restricciones de la solución quedaron plasmadas en el **Anexo 6. Requerimientos no funcionales**.

Tabla 3. Clasificación requerimientos no funcionales.

Prioridad	Cantidad
Usabilidad	3
Seguridad	2
Eficiencia	2
Soporte	2
Hardware	2
Rendimiento	2
Restricciones del diseño	6
Interface de Hardware	2
Portabilidad	1
Estándar Aplicable	2

## 2.9 Descripción de la arquitectura y el diseño

La Arquitectura de *Software*, también denominada Arquitectura Lógica, es un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del sistema. En la solución que se elabora se proponen dos arquitecturas, una para las funcionalidades a nivel de aplicación basada el estilo arquitectónico Cliente-Servidor, haciendo uso del patrón de arquitectura Modelo-Vista-Controlador ya que el marco de trabajo empleado utiliza este patrón por su flexibilidad y múltiples ventajas, y otra para las funcionalidades a nivel de dato, que comprenderán la implementación del sincronizador de bases de datos haciendo uso del patrón de arquitectura 2 Capas.

### 2.9.1 Arquitectura Cliente-Servidor

La arquitectura Cliente-Servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, que le da respuesta. Normalmente, el servidor



es una máquina bastante potente que puede actuar como servidor de aplicación, de depósito de datos, o sencillamente brindar determinados servicios. (Billy Reynoso 2004)

Por otro lado, los clientes suelen ser estaciones de trabajo que realizan varias solicitudes o peticiones al servidor. Ambas partes deben estar conectadas entre sí mediante una red. Una representación gráfica de este tipo de arquitectura sería la siguiente:

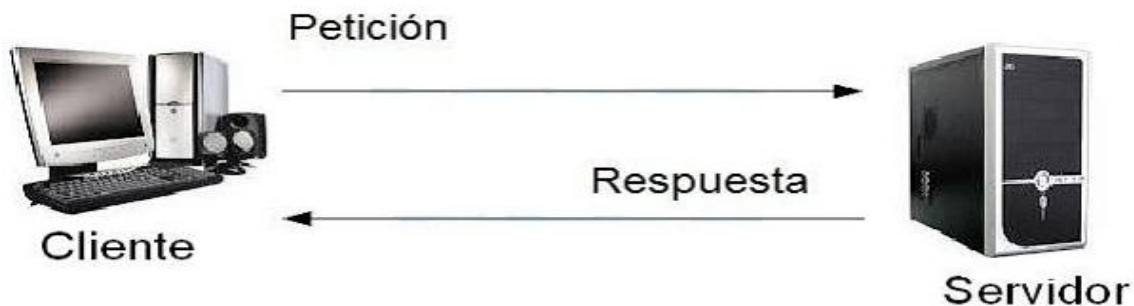


Figura 6. Arquitectura Cliente – Servidor.

Este tipo de estilo arquitectónico es el más utilizado en la actualidad, debido a su avance y evolución en estos últimos años. Se puede decir que esta arquitectura necesita tres tipos de *software* para su correcto funcionamiento:

- *Software* de gestión de datos: se encarga de la manipulación y gestión de los datos almacenados y requeridos por las diferentes aplicaciones. Normalmente, se aloja en el servidor.
- *Software* de desarrollo: se aloja en los clientes y solo en aquellos que se dedique al desarrollo de aplicaciones.
- *Software* de interacción con los usuarios: también reside en los clientes y es la aplicación gráfica de usuario para la manipulación de datos, siempre a nivel usuario (consultas principalmente).

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, los servidores de Bases de Datos, entre otros.

### 2.9.2 Patrón arquitectónico Modelo-Vista-Controlador

Es un patrón de arquitectura de *software* típicamente utilizado para la creación de una aplicación Web, empleado en la mayoría de los marcos de trabajo utilizados en el desarrollo *web*, en PHP ("*Hypertext Preprocessor*") u otros lenguajes. Se caracteriza por separar en varios grupos las complejidades de las



distintas partes que componen una aplicación, como la vista y la lógica, así como el acceso a la base de datos.

El Modelo-Vista-Controlador (MVC) es un patrón de arquitectura de *software* que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos, permitiendo realizar una programación multicapa. (Guglielmetti 2004)

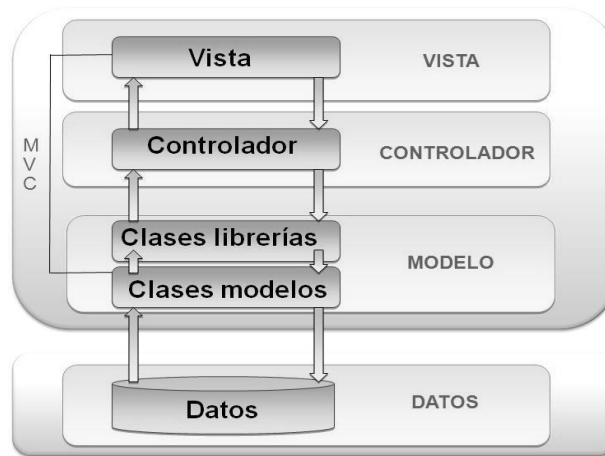


Figura 7. Patrón arquitectónico Modelo-Vista-Controlador.

Para la representación del patrón MVC de la **Figura 7** se muestra a continuación una descripción de cada uno de los componentes:

**Modelo:** representa la lógica del negocio y la información del sistema. La lógica de datos asegura la integridad y permite derivar nuevos datos. Las clases modelo y las librerías se encuentran en esta capa. Las librerías son las encargadas de organizar la lógica del negocio, mientras que las modelos son las que acceden a los datos. Las clases librerías envían los datos a las controladoras, después de obtenerlos de las modelos.

**Vista:** presenta el modelo en un formato adecuado para interactuar, usualmente es la interfaz de usuario. Son clases que a través de archivos JavaScript, estilos CSS y la librería JQuery conforman el código HTML necesario para mostrar la información requerida por el usuario, que se obtienen de las clases controladoras.

**Controlador:** responde a las acciones del usuario e invoca cambios en el modelo o genera la vista apropiada dependiendo de las peticiones del usuario. Sirve como un intermediario entre el modelo, la vista y cualquier otro recurso necesario para procesar una petición. Utiliza los métodos que contienen las clases librerías para realizar peticiones de datos a las clases modelo, que son las que tienen acceso a los datos y



estas al obtenerlos se los envía a través de las librerías a la controladora, la cual una vez que los obtiene se los entrega a las vistas.

**Este patrón brinda una serie de ventajas como:**

- Diseño modular y poco acoplado, favoreciendo la reutilización.
- Mayor cohesión ya que cada elemento del patrón está altamente especializado en su tarea.
- Las vistas proveen mayor flexibilidad y agilidad pues se pueden crear múltiples vistas de un modelo, las vistas pueden anidarse, sincronizar y otras más opciones.
- Más claridad en el diseño.

### 2.9.3 Patrón arquitectónico Dos Capas

La arquitectura a nivel de dato está representada por dos capas lógicas, que contienen funcionalidades encargadas del procesamiento de reglas de negocio (capa de negocios), y la relacionada con el modelado y el acceso a datos (capa de abstracción de datos). En la **Figura 8** se muestra la representación de este patrón y a continuación la descripción de las capas.



Figura 8. Patrón arquitectónico 2-Capas.

#### Capa de negocios:

En esta capa se recogen un conjunto de servicios que contiene las funcionalidades del negocio. Para la interacción de esta capa con la base de datos se usan las clases de tipo conector, que manejan las entidades del negocio que son necesarias persistir en una fuente de datos.



### Capa de abstracción de datos:

La capa de abstracción de datos es el componente que da soporte a las funcionalidades de la capa de negocios, y que se encuentra relacionada con una fuente de datos. Esta capa permite un alto nivel de abstracción con relación a la base de datos y modela todas las operaciones que debe realizar el conector con la fuente de datos.

### Beneficios del patrón Dos Capas:

- Flexibilidad: como cada capa puede ser manejada y escalada de forma independiente, la flexibilidad se incrementa.(Pelaez 2009)
- Carga de base de datos reducida: el servidor de base de datos se beneficia de menos conexiones.
- Mejoras en las posibilidades de mantenimiento: debido a que cada capa es independiente de la otra, los cambios o actualizaciones pueden ser realizados sin afectar la aplicación como un todo.

### 2.9.4 Patrones de diseño

Los patrones de diseño son los principios generales de soluciones que aplican ciertos estilos que ayudan a la creación de *software*. Una descripción de un problema y la solución a la que le da el nombre y que se puede aplicar en nuevos contextos.(Pressman 2007)

### Patrones para asignar responsabilidades (GRASP):

Los patrones GRASP (*General Responsibility Assignment Software Patterns*) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el *software* orientado a objetos. (Lago 2007). A continuación se mencionan los patrones GRASP utilizados en el desarrollo de la solución:

- **Experto en información:** consiste en asignar la responsabilidad al experto en información que tiene o puede tener los datos involucrados (atributos), es decir, la clase que tiene la información necesaria lleva a cabo la responsabilidad. Este patrón se evidencia en las clases librerías, que son las que cuentan con la información necesaria para cumplir las responsabilidades sobre los elementos de negocio.
- **Creador:** permite decidir cuáles serán las clases creadoras de otras clases. Responde que clase debe ser responsable de crear una nueva instancia. Este patrón se manifiesta en la implementación del sincronizador a través de la extensión Objetos de Datos de PHP (PDO por sus siglas en inglés), que se encarga de crear los objetos de conexión a las bases de datos.



- **Alta cohesión:** Se utiliza para mantener manejable la complejidad, asignando responsabilidades de manera que la información que almacena una clase sea coherente y esté relacionada con la clase.
- **Bajo acoplamiento:** consiste en diseñar las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases. Estos dos últimos patrones se relacionan porque la propia implementación de CodeIgniter contiene estos patrones nivelados, pues permite el uso de los componentes de forma individual, evidenciando el bajo acoplamiento y así como la dependencia entre ellos o alta cohesión.

### Patrones GOF:

Los patrones de diseño GOF (*Gang Of Four*) clasifican en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento. (Lago 2007). A continuación se mencionan los patrones GOF utilizados en el desarrollo de la solución:

**Creacionales:** tratan con las formas de crear instancias de objetos. El objetivo de estos patrones es abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos se crean o se inicializan.

- **Singleton** (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Este patrón se evidencia en todas las clases controladoras, porque estas clases son instancias únicas. Además se evidencia en la clase ioc (*Inversion of Control*) para la interacción entre los módulos.

**Comportamiento:** Los patrones de comportamiento ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos.

- **Command** (*Orden*): Encapsula una operación en un objeto, permitiendo ejecutar dicha operación sin necesidad de conocer el contenido de la misma. Este patrón se utiliza en la implementación del sincronizador cuando se realizan llamadas a las funciones del objeto PDO y se utilizan sin conocer de qué forma están implementadas.
- **Mediator** (*Mediador*): Define un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto. Este patrón se evidencia en la implementación en la creación de librerías que funcionan como mediadoras entre las clases controladoras y los modelos o acceso a datos.





- **Observer** (*Observador*): Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. Este patrón se pone de manifiesto en la implementación en la clase *loader* que es el objeto load de las clases controladoras, encargada de cargar los elementos del marco de trabajo dígase, librerías, modelos y se encarga de actualizar la controladora instanciada.

### Patrones de bases de datos:

Los patrones de diseño de una base de datos permiten al usuario crear bases de datos más fortalecidas ya que constituyen una guía que especifica cómo debe ser la misma. (Blaha 2010). A continuación se describen algunos de los patrones utilizados para el desarrollo de la solución.

- **Llaves subrogadas**

Este patrón es muy utilizado pues se decide generar una llave primaria única para cada entidad en vez de usar un atributo identificador en el contexto dado. Normalmente, se usa enteros en columnas *identity* o GUID (*Global Unique Identifier*), que está demostradas que no se repiten o que pueden hacerlo con una probabilidad extremadamente baja. Esto permite que las tablas sean más fáciles de consultar por el identificador, dado que se conoce el mismo tipo de dato de todos en cada tabla. Mediante este patrón, fueron generados los identificadores de todas las tablas pertenecientes al modelo de datos que se muestra más adelante.

- **Modelo entidad-atributo-valor**

El patrón entidad-atributo-valor es la representación de un modelo flexible donde se pueden representar objetos con sus atributos. Es un acercamiento al modelo orientado a objeto representado en el modelo relacional, donde la entidad *Class* representa las clases, la entidad *Attribute* representa los atributos de las clases, por su parte la entidad *Object* representa las instancias de las clases, mientras que la entidad *Value* representa los valores de cada atributo para cada objeto dado. En la solución se utiliza este patrón para el almacenamiento de la información que surge nueva a cada momento.

## 2.10 Estándares de nomenclatura para la versión de la información

La numeración de una versión de un *software* distingue la actualidad del mismo ya sea en su fase de desarrollo, como en el despliegue. Para definir la numeración de la versión de la información personal se realizó un minucioso estudio sobre las tendencias actuales de las versiones. El estudio arrojó que el método más habitual de asignar una versión a un documento es mediante un número o un grupo de números, aunque no existe un modo fijo de numerar una versión, se deja al criterio de cada desarrollador. En el estudio realizado se caracterizaron algunos tipos de numeraciones de versiones con el objetivo de



utilizar una de ellas para la nomenclatura de la versión de la información personal. A continuación se caracterizan las numeraciones de versiones estudiadas:

### **Versiones x.y.z**

Una de las prácticas más conocidas son las versiones x.y.z. En este caso se utilizan dos o tres cifras decimales para indicar la importancia de los cambios realizados. El cambio de la primera cifra indica cambios más importantes que el de la segunda. El significado de la numeración de la versión x.y.z es el siguiente: la primera cifra (x) indica la versión mayor del documento, la segunda cifra (y) indica la versión menor del documento y la tercera cifra (z) indica la segunda versión menor. (Borrell 2006)

### **Versiones enumeradas**

Las versiones enumeradas son un método de numeración compatible que consiste en numerar las versiones mediante un número decimal. La primera versión es la 1, la segunda es la 2 y así sucesivamente. Esta numeración suele utilizarse en el desarrollo cuando el control de versiones se lleva de forma automática mediante algún programa de control de versiones.

Después del estudio realizado, se arrojó como resultado que ninguna de las anteriores numeraciones es considerada efectiva para enumerar las versiones del versionado de la información personal, porque carecen de información acerca de los detalles de cada versión. Por tanto se plantea que la versión a utilizar en el versionado de la información personal es una especialización de las versiones x.y.z llamada versión a.b.c denotando la primera cifra (a) al proveedor de la información (área que provee la información), la segunda cifra (b) el número de versión de la información y la tercera cifra (c) el número de cambios realizados a la versión de la información.

### **Un ejemplo de esta numeración de versión sería:**

Si se tiene una versión de la información Cierre de Matrícula donde el proveedor 2 se refiere a Pregrado, la versión de la información es 1 y la cantidad de cambios es 3 entonces la versión 2.1.3 indica que se han hecho 3 cambios para la primera versión de la información Cierre de Matrícula del proveedor Pregrado.

## **2.11 Modelo de datos**

Las bases de datos necesitan una definición de su estructura que le permitan almacenar datos, reconocer el contenido y recuperar la información. Los modelos de datos permiten describir los elementos que intervienen en una realidad o en un problema dado. Típicamente un modelo de datos es un lenguaje que describe:



- Las estructuras de datos: El tipo de los datos que hay en la base y la forma en que se relacionan.
- Las restricciones de integridad: Un conjunto de condiciones que deben cumplir los datos para reflejar correctamente la realidad deseada.
- Operaciones de manipulación de los datos: Operaciones de agregado, borrado, modificación y recuperación de los datos almacenados en la base de datos.

Con el modelo de datos se puede llegar a un modelo físico sin convertirse en un proceso complejo, dando una visión al programador de lo que necesita y debe hacer el sistema. Para ver el modelo físico del sincronizador dirigirse al **Anexo 7. Modelo físico del sincronizador**.

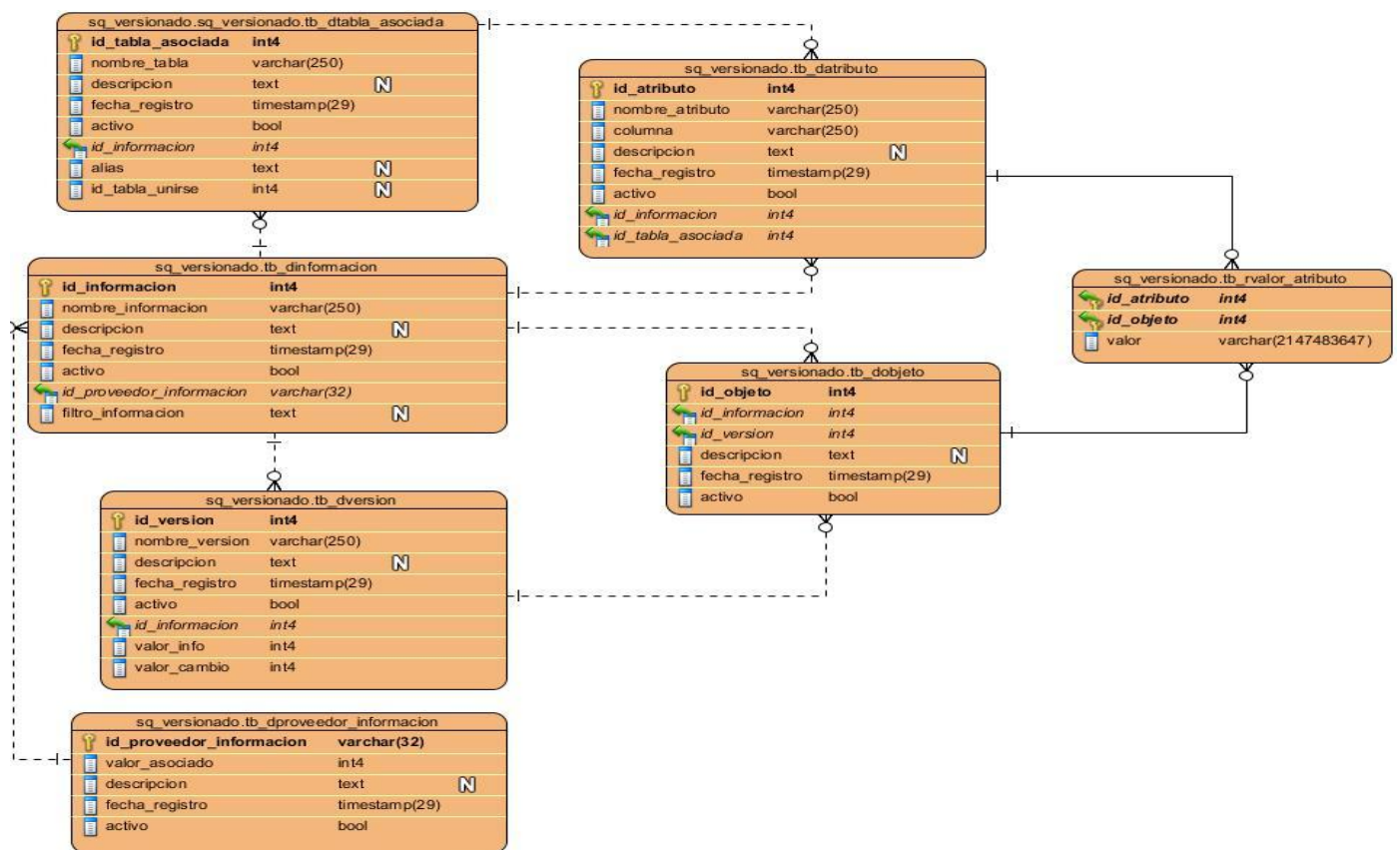


Figura 9. Modelo físico del versionado de información personal.

## 2.12 Diagrama de despliegue

Los nodos de un diagrama de despliegue representan elementos físicos con capacidad de proceso o facilitadores de algún servicio. Las máquinas físicas y los procesadores se representan como nodos y la construcción interna puede ser representada por nodos o artefactos embebidos. Los estereotipos permiten precisar la naturaleza del equipo: dispositivos, procesadores y memoria. El diagrama de despliegue



describe la arquitectura física del sistema durante la ejecución, en términos de procesadores, dispositivos y componentes de *software*. En la **Figura 10** se presenta el modelo de despliegue de la solución:

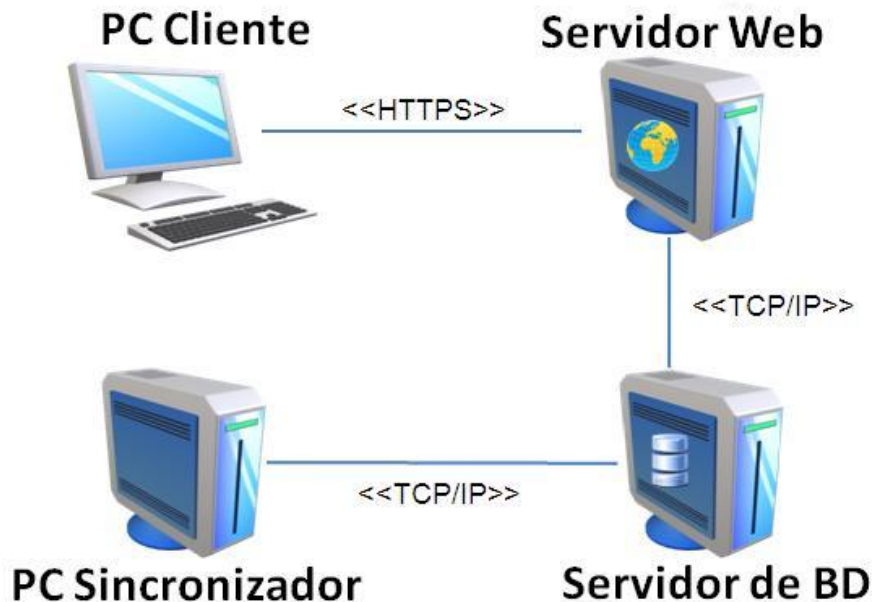


Figura 10. Diagrama del modelo de despliegue.

### 2.12.1 Descripción del diagrama de despliegue

El diagrama de despliegue representado en la figura anterior muestra la siguiente distribución:

**PC Cliente:** ordenador cliente capaz de conectarse al servidor de aplicaciones mediante el protocolo seguro de comunicaciones HTTPS. Ordenador que cuente con un navegador actualizado y que siga los estándares *web* (*Mozilla Firefox v2* o superior). Se recomiendan estaciones de trabajo con sistema operativo *GNU/Linux*.

**Servidor Web:** ordenador en el que se encuentra el servidor *web Apache*, este será el lugar en el que se gestione todo el contenido de la aplicación. El mismo establecerá comunicación con los ordenadores clientes mediante protocolo seguro HTTPS y con el servidor de base de datos por medio de la familia de protocolos TCP/IP.

**PC Sincronizador:** ordenador que será utilizado para el sincronizador de las bases de datos, este estará ejecutándose automáticamente y requiere de un intérprete de PHP para su funcionamiento. El mismo establecerá comunicación con el servidor de base de datos por medio de la familia de protocolos TCP/IP.



**Servidor de BD:** ordenador en el que se encuentra el gestor de bases de datos PostgreSQL capaz de mantener persistente la información generada a utilizar. Además dará respuesta a las peticiones hechas por la aplicación.

### **2.13 Conclusiones parciales**

En el capítulo se describieron las actividades que integran el proceso de versionado de la información personal, las que unidas al análisis de las necesidades del cliente, permitieron identificar y especificar los requerimientos de la propuesta de solución. Se describieron los elementos fundamentales de la arquitectura y el diseño. Se obtuvo el modelo físico de la base de datos de la solución, observándose pocos cambios a partir del modelo de entidades conceptuales mostrado y se realizó un análisis de cómo la propuesta de solución se debe integrar con el Sistema de Gestión Universitaria.



# Implementación y validación de la solución

## 3 CAPÍTULO 3

### 3.1 Introducción

En este capítulo se especifica la implementación y prueba de la solución versionado de la información personal en el Sistema de Gestión Universitaria (SGU). Además se describen las técnicas de validación de requerimientos y las pruebas de *software* aplicadas así como sus resultados, lo que permite verificar el cumplimiento de los requerimientos definidos y garantizar la calidad de la solución desarrollada.

### 3.2 Estándares para el diseño de prototipos

Para realizar el diseño de los prototipos se tuvo como guía las pautas del diseño visual establecidas en el Manual de Directrices del SGU, que permite la uniformidad en todas las páginas *web* que lo componen.

#### 3.2.1 Distribución de la información en la propuesta de información

La vista de presentación es la primera vista que se le muestra a cualquier usuario, mediante la que podrá autenticarse y acceder al sistema.

Figura 11. Vista de presentación.

- 1- Cabezal o área de identificación.
- 2- Área de entrada de datos.

La vista de escritorio se muestra luego de la autenticación del usuario, en la que podrá seleccionar el subproceso horizontal, módulo o servicio al que desee acceder y tenga los permisos requeridos.



Figura 12. Vista de escritorio.

- 1- Área de subprocessos horizontales
- 2- Área del nombre de la aplicación.
- 3- Área de nombre de usuario.

- 4- Área de líneas de procesos.
- 5- Área de servicios horizontales.
- 6- Área de pie de página.

La vista de gestión de procesos permite el acceso a los módulos del subsistema seleccionado por el usuario y las funcionalidades que poseen.

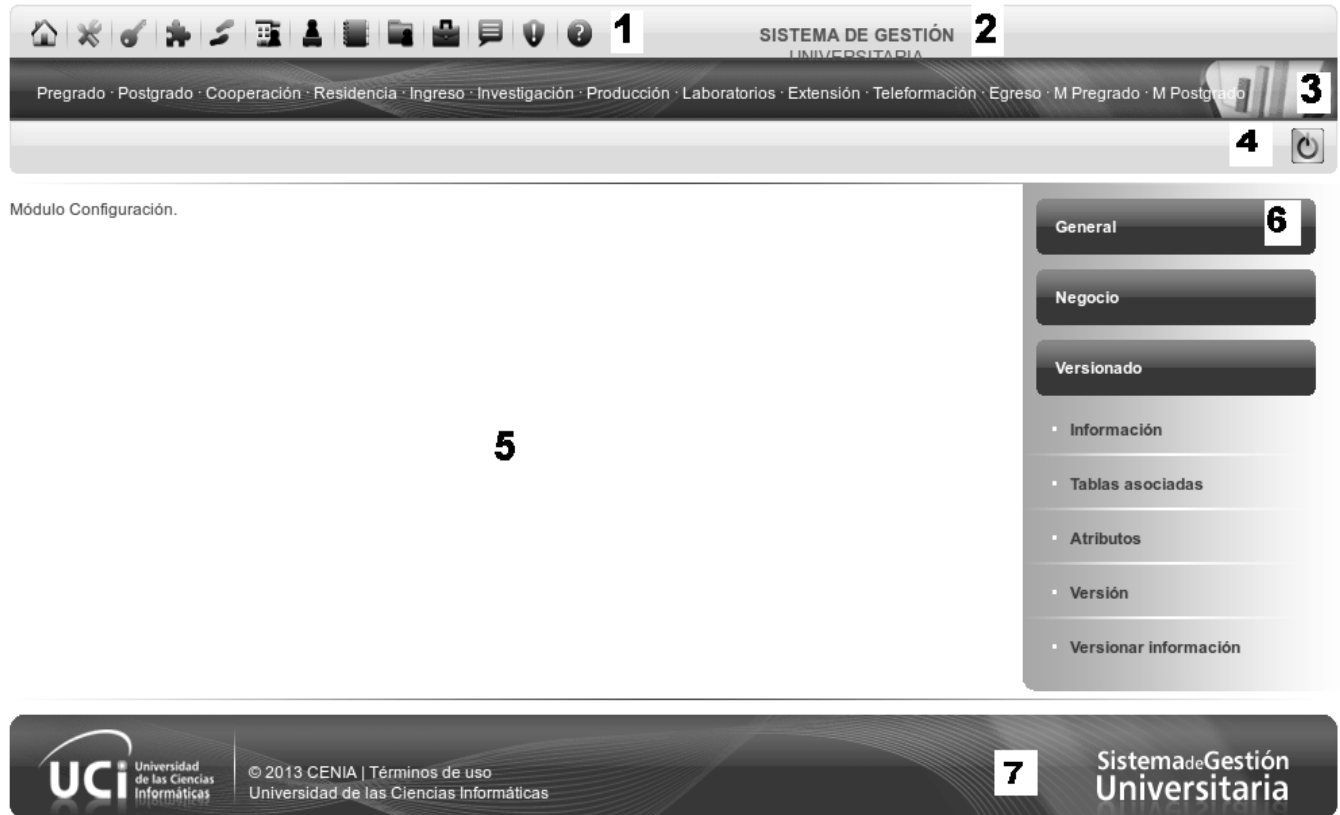


Figura 13. Vista de gestión de procesos.

- |                                      |                           |
|--------------------------------------|---------------------------|
| 1- Área de subprocessos horizontales | 5- Área de contexto.      |
| 2- Área del nombre de la aplicación. | 6- Área de módulos.       |
| 3- Área de líneas de procesos.       | 7- Área de pie de página. |
| 4- Área de nombre de usuario.        |                           |

En el área del menú de módulos podrá seleccionar el módulo de Versionado que contiene los siguientes procesos:

- Información.
- Tablas asociadas.
- Atributos.
- Versión.
- Versionar información.

### 3.2.2 Tipos de mensajes

Se definieron 4 tipos de mensajes:

- Mensajes de información.
- Mensajes de error.





- Mensajes de confirmación.

### Mensajes de información para el cliente

Se utilizan para brindar una información al usuario, cuando se crea un elemento, se modifica, o se asocia:

- “El elemento ha sido creado satisfactoriamente.”
- “El elemento ha sido modificado satisfactoriamente.”



Figura 14. Mensaje de información.

### Mensajes de error

Se utilizan para mostrar al usuario cuando ha realizado una acción incorrecta:

- “El elemento ya existe.”



Figura 15. Mensaje de error.

### Mensajes de confirmación

Son los que se utilizan cuando es necesario asegurarse que el usuario desea realizar una acción deseada, por ejemplo cuando se cancela una acción sobre un elemento, es necesario asegurarse que eso es lo que desea.

- ¿Está seguro de realizar la acción?

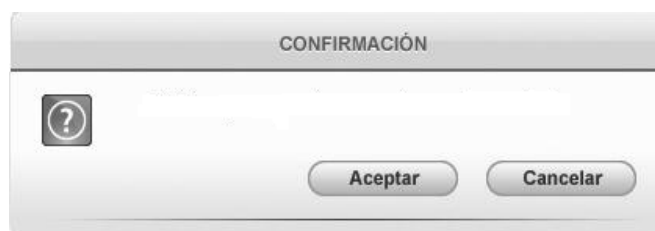


Figura 16. Mensaje de confirmación.



### 3.3 Técnicas de programación

Las técnicas de programación constituyen una parte fundamental en el proceso de desarrollo e ingeniería del *software* dentro del ámbito informático. Cada técnica tiene sus propias características y distintos métodos para darle solución a determinados problemas, es de gran importancia aprender a implementarlas a la hora de realizar cualquier proyecto de desarrollo de *software*.

#### 3.3.1 Programación Orientada a Objetos (POO)

La POO es un paradigma o técnica de programación que utiliza la abstracción para crear modelos basados en el mundo real, los *objetos*. Un objeto es un ente real o abstracto con ciertas características y funcionalidades que lo definen. Las propiedades y funcionalidades de un objeto se especifican en su *clase*, la cual vendría a ser el molde de cada instancia particular de este. Con esta técnica un programa puede ser descompuesto en una serie de objetos que interactúan entre sí para lograr un propósito común (Bonatata 2003). La POO está basada en cuatro aspectos: definición de tipos de datos abstractos, herencia, encapsulamiento y polimorfismo.

En la POO encapsular significa que se reúne y controla todo el grupo resultante en un conjunto y no de forma individual. La abstracción es un término externo al objeto, que controla la forma en que es visto por los demás. La herencia se define como una jerarquía de clases derivadas y la relación entre estas, donde se comparte la estructura y el comportamiento de una o más clases consideradas como clases padres. El polimorfismo constituye la definición de múltiples clases con funcionalidades diferentes, pero con métodos o propiedades denominados de forma idéntica. (Bonatata 2003)

### 3.4 Estándares de codificación

Los estándares de codificación son un estilo de programación homogénea en un proyecto. Estos son de vital importancia durante la etapa de construcción del *software* ya que permite que todo el personal del proyecto pueda entender de forma fácil el código, es decir, los estándares de codificación son un conjunto de reglas a seguir por los desarrolladores con el objetivo de establecer un orden y un formato común en el código fuente del sistema en desarrollo.

Para el desarrollo de la solución informática se utilizaron los estándares de codificación establecidos por el Centro de Informatización Universitaria (CENIA), con el propósito de estandarizar las nomenclaturas en la implementación del sistema y obtener un producto estable y eficiente.



### 3.4.1 Identación, llaves de apertura y cierre y tamaño de las líneas

Se debe usar la indentación sin tabulaciones, con un equivalente a 4 espacios, para mantener integridad en las revisiones svn. El uso de las llaves “{}” será en una nueva línea. La longitud de las líneas de código es aproximadamente de 75-80 caracteres, para mantener la legibilidad del código.

```
public function obtenerInformacionesActivasIOC($param, $modulo)
{
    $informacion = $this->_ci->ioc->$modulo->obtenerInformacionesActivas($param);
    return $informacion;
}
```

Figura 17. Identación, llaves de apertura y cierre.

### 3.4.2 Conversión de nomenclatura

**Variables:** se rigen por la nomenclatura *camelCase*. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula.

```
$datos = $this->input->all_post();
$bd = $datos['bd'];
```

Figura 18. Variables.

**Clases:** siempre comienzan con mayúscula, en caso de nombre compuesto las palabras se separan con el carácter subrayado “\_” y el resto en minúscula.

```
class Atributo extends MY_Controller
{
    //put your code here
    public function __construct()
    {
        parent::__construct();
        $this->load->library("atributo_lib");
    }
}
```

Figura 19. Clases.

**Funciones:** se rigen por la nomenclatura *camelCase*. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por espacio luego de la coma que los separa.



```
public function obtenerTablasUnirseActivasIOC($param, $modulo)
{
    $tabla_unirse = $this->_ci->ioc->$modulo->obtenerTablasUnirseActivas($param);
    return $tabla_unirse;
}
```

Figura 20. Funciones.

**Ficheros:** siempre se escriben en minúscula y en caso de nombres compuestos se usa el caracter subrayado “\_”.

**Vistas:** intuitivo y relacionado con el formulario y/o vista que representa.

**Modelos:** con el mismo nombre de la clase que representa que contiene en el nombre el sufijo `_mdl`.

**Librerías:** con el mismo nombre de la clase que representa que contiene en el nombre el sufijo `_lib`.

**Controladoras:** con el mismo nombre de la clase que representa.

### 3.4.3 Estructuras de control

Se incluye un espacio entre las estructuras de control (*if*, *for*, *foreach*, *while*, *switch*) y los paréntesis. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales. Esto aumenta la legibilidad y disminuye la probabilidad de errores lógicos.

```
if($this->version_lib->chequearNombreRepetido($all_post) == TRUE)
{
    throw new Exception_Error('SYS005');
}
if($this->version_lib->registrarVersion($all_post) != FALSE)
{
    $this->message('SYS001');
}
else
{
    throw new Exception_Error('SYS006');
}
```

Figura 21. Estructuras de control.



Si las condiciones son muy largas que sobrepasan el tamaño de la línea, estas se dividen en varias líneas.

```
if(!empty(json_decode($post_vars['cadena']->text_buscar)
|| !empty(json_decode($post_vars['cadena']->text_informaciones))

{
    $datos_buscador = json_decode($post_vars['cadena']);
    echo( grid_json($this->atributo_lib, 'obtenerCantidadAtributosIOCBuscar',
                    'obtenerAtributosIOCBuscar', array($datos_buscador)));
}
```

Figura 22. Condiciones en varias líneas.

#### 3.4.4 Documentación

Todos los archivos deben de tener la documentación asociada al mismo. Para esto debe de cumplir con el siguiente bloque al principio de cada clase.

```
/**
 * Librería versionado para pregrado
 *
 * @author mevs
 */
```

Figura 23. Documentación.

#### 3.4.5 Buenas prácticas

Los valores booleanos y nulos siempre se escriben con mayúscula, para facilitar la legibilidad del código, se usa una línea en blanco antes de las estructuras de control y definición de las funciones.

```
$post_vars = $this->input->all_post(TRUE);
$this->load->helper('grid');
```

Figura 24. Buenas prácticas.

### 3.5 Validación de la solución propuesta

Cuando se construye un *software* la calidad no se puede probar, sino que se va construyendo a lo largo del desarrollo del mismo. Cuanto más errores se encuentren en las revisiones, más se podrá asegurar la calidad de la propuesta de solución.

En la investigación al utilizar un proceso de desarrollo con enfoque ágil basado en el nivel 2 de CMMI se deben aplicar validaciones a los requerimientos y realizar pruebas durante el ciclo de vida del producto. En los siguientes epígrafes se podrá encontrar las validaciones que se le hicieron a los requerimientos y la estrategia de pruebas a utilizar.



### 3.6 Validación de requerimientos

La validación de requerimientos es el proceso por el cual se determina si los requerimientos relevados son consistentes con las necesidades del cliente. Además examina las especificaciones para asegurar que los requerimientos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajusta a los estándares establecidos. (Pressman 2007)

#### ➤ Criterios de Validación

En el proceso de desarrollo se define un documento para la validación de requerimientos que permiten identificar los criterios para validar los requerimientos del cliente. Con la generación del documento “*Criterios para validar requerimientos del cliente*” se pudo conocer si los requerimientos funcionales y no funcionales quedan aprobados después de evaluar todos los criterios definidos. Los criterios de validación aparecen recogidos en el **Anexo 8. Criterios para validar requerimientos del cliente.**

#### ➤ Técnicas de Validación

Las siguientes técnicas empleadas para la validación de los requerimientos fueron definidas por *Sommerville* que permiten que el proceso de validación de la solución tenga una mejor calidad:

- **Revisiones de requerimientos:** se realizaron reuniones con el cliente para la lectura, revisión y corrección de la definición de requerimientos, de este modo se pudo obtener los errores presentes en la especificación de los requerimientos de la solución y validar la correcta interpretación de la información.
- **Construcción de prototipos:** este enfoque de validación se realizó con el objetivo de facilitar a los usuarios interesados en el producto una imagen de lo que sería el futuro sistema, a partir de los requerimientos recogidos en la especificación, entregando una versión del producto final. Su principal propósito fue obtener y validar los requerimientos esenciales, manteniendo abiertas, las opciones de implementación. A través de ellos se pudo comprobar que se estaba construyendo el sistema correcto, que la aplicación producía las salidas correctas y que cumplía con las necesidades y requerimientos del usuario.
- **Generación de casos de prueba:** la realización de casos de prueba posibilitó la verificación del cumplimiento de los requerimientos funcionales, se pudo comprobar que los mismos tenían la eficiencia requerida.



### 3.6.1 Resultado de la revisión de los requerimientos

Con la aplicación de los criterios para validar requerimientos del cliente se obtuvo el 100 % de los requerimientos aprobados. En total se validaron 50 requerimientos del cliente.

Durante la revisión de requerimientos se realizaron verificaciones en el documento “Especificaciones de requerimientos”, este proceso comprendió las siguientes verificaciones:

- **Verificaciones de validez:** los requerimientos deben cumplir con las necesidades del cliente. Luego de realizar algunos análisis y razonamientos surgieron funciones adicionales y cambios en las que ya estaban identificadas.
- **Verificaciones de consistencia:** los requerimientos no deben contradecirse en las especificaciones escritas, no debe haber restricciones o descripciones que estén opuestas a las reglas definidas.
- **Verificaciones de completitud:** los requerimientos deben incluir todas las funcionalidades propuestas por el cliente, satisfacer de manera general todas las necesidades acordadas.
- **Verificabilidad:** para evitar posibles discusiones entre los miembros del equipo del proyecto y el cliente, se revisó que los requerimientos estuvieran descritos de manera que puedan ser verificables, o sea, que se puedan diseñar casos de pruebas orientadas a estos y que demuestren que el sistema a entregar responde a las necesidades del cliente

Al concluir el proceso de revisión se detectaron algunas inconsistencias que fueron erradicadas de inmediato. Entre las más comunes se pueden citar:

- Se interpretaron de forma incorrecta algunas de las funcionalidades y características solicitadas por el cliente.
- Descripciones poco detalladas de algunos de los requerimientos.
- Errores ortográficos.
- Ausencia de opciones en el área de iconos flotantes.
- Componentes innecesarios en los prototipos de interfaz.

### 3.7 Estrategia de prueba

La estrategia de prueba del *software* describe el enfoque y los objetivos generales de las actividades de prueba. Esta debe incluir pruebas de bajo nivel que verifiquen que todos los pequeños segmentos de código fuente se han implementado correctamente, así como pruebas de alto nivel que validen las principales funciones del sistema frente a los requerimientos del cliente.(Pressman 2007). La estrategia



que se ha de seguir a la hora de evaluar dinámicamente un sistema debe permitir comenzar por los componentes más simples y más pequeños e ir avanzando progresivamente hasta probar todo el *software* en su conjunto. Para los efectos de la investigación se define una estrategia basada en las técnicas y métodos propuestos por *Roger S. Pressman* en la cual se propone realizar:

1. Pruebas unitarias.
2. Pruebas de integración.
3. Pruebas del sistema (funcionales).
4. Pruebas de aceptación.

A continuación se procede a la descripción y realización de cada una de las pruebas antes mencionadas.

### 3.7.1 Pruebas unitarias

Las pruebas de unidad se centran en la verificación de los elementos más pequeños del *software* que se puedan probar (Pressman 2005). Se prueba la interfaz del módulo para asegurar que la información fluye de forma adecuada hacia y desde la unidad de programa que está siendo probada. Además, se examinan las estructuras de datos locales para asegurar que los datos que se mantienen temporalmente conservan su integridad mediante los pasos de ejecución del algoritmo. En el ámbito de la presente investigación, para la realización de esta prueba se empleó el método de caja blanca y para generar los casos de pruebas la técnica empleada es la prueba del camino básico.

#### ➤ Método de caja blanca

El método de caja blanca o estructural, se basa en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa. El objetivo de la técnica es diseñar casos de prueba para que se ejecuten, al menos una vez, todas las sentencias del programa, y todas las condiciones tanto en su vertiente verdadera como falsa. Como es de suponer, es poco práctico realizar una prueba exhaustiva de todos los caminos de un programa.

#### ➤ Técnica del camino básico

Existen ciertas técnicas para el diseño de este tipo de pruebas que permiten decidir qué sentencias o caminos se deben examinar con los casos de prueba. La técnica del camino básico se basa en obtener una medida de la complejidad del diseño procedimental de un programa (o de la lógica del programa). Esta medida es la complejidad ciclomática de *McCabe* y representa un límite inferior para el número de casos de pruebas que se deben realizar para asegurar que se ejecuta cada camino del programa. En el **Anexo 9. Descripción de la técnica del camino básico** se muestra el procedimiento que se debe realizar con esta técnica.





➤ **Pruebas unitarias en CodeIgniter**

Para la realización de las pruebas de caja blanca y con el objetivo de verificar el resultado real de la prueba para cada uno de los caminos, se empleó un mecanismo que posee CodeIgniter (parte estructural del marco de trabajo, Gestión Universitaria Universo Digital Gestión Documental (GUUD) en este caso) para la automatización de pruebas unitarias y que se describe a continuación:

CodeIgniter posee una librería o clase especializada en la ejecución de pruebas estructurales. Aunque es bastante sencilla, pues cuenta con una sola función de evaluación y dos funciones de resultados, permite determinar con certeza si un código específico produce el tipo de dato y resultado esperado. Para correr una prueba utilizando dicha librería es necesario suministrar el código a probar y un resultado esperado de la siguiente forma:

```
$this->unit->run (código, resultado esperado, 'nombre de prueba');
```

Donde “código” es el segmento de código que se desea probar, “resultado esperado” es lo que se espera que devuelva la evaluación del código (puede ser un tipo de dato o un valor literal) y “nombre de prueba” es un nombre opcional que se le puede dar a la prueba. Los tipos de datos posibles son: “is\_string”, “is\_bool”, “is\_true”, “is\_false”, “is\_int”, “is\_numeric”, “is\_float”, “is\_double”, “is\_array”, “is\_null”.

**Casos de pruebas de caja blanca**

Tabla 4. CENIA\_SGU\_C\_DCP -OI: RFR10, RFR11, RFR12, RFR13, RFR14.

<b>Prueba estructural de caja blanca</b>	<b>Código caso de prueba:</b> CENIA_SGU_C_DCP -OI: RFR10, RFR11, RFR12, RFR13, RFR14.
<b>Probador:</b> Manuel E. Velázquez Sánchez	
<b>Código al que se aplica:</b>	
<pre style="font-family: monospace; font-size: 0.9em;">public function obtenerInformacion() {     \$post vars = \$this-&gt;input-&gt;all post(true);     \$this-&gt;load-&gt;helper('grid');      if(isset(\$post vars['cadena']) &amp;&amp; !empty (\$post vars['cadena']))     {         \$bd = json decode(\$post vars['cadena']);         echo grid json(\$this-&gt;informacion lib, 'obtenerCantidadInformacionIOC',             'obtenerInformacionIOCOrdenar', array(\$bd));     }     else     {         \$dat = array();         echo grid json(\$this-&gt;informacion lib, 'obtenerCantidadInformacionIOC',             'obtenerInformacionIOCOrdenar', array(\$dat));     } }</pre>	



<p><b>Complejidad ciclomática:</b>  <math>V(G) = (A - N) + 2 = (4 - 4) + 2 = 2</math></p> <p><b>Caminos independientes:</b></p> <ol style="list-style-type: none"> <li>1) 1 - 2 - 4</li> <li>2) 1 - 3 - 4</li> </ol>	<p><b>Representación del grafo:</b></p> <pre> graph TD     1((1)) --&gt; 2((2))     1((1)) --&gt; 3((3))     2((2)) --&gt; 4((4))     3((3)) --&gt; 4((4))     </pre>
<b>Caso de prueba para el camino básico 1</b>	
<b>Descripción:</b> los datos de entrada es la información a obtener.	
<b>Condición de ejecución:</b> se envían los datos de entrada.	
<b>Procedimiento prueba automatizada</b>	
<b>Datos de entrada:</b>	arreglo vacío
<b>Tipo de dato esperado:</b>	is_array
<b>Función de evaluación:</b>	
<pre> \$resultadoEsperado="Array"; \$nombrePrueba="Prueba: Obtener Información"; echo \$this-&gt;ejecutarPrueba(\$this-&gt;ObtenerInformacion(), \$resultadoEsperado , \$nombrePrueba);     </pre>	
<b>Evaluación del caso de prueba:</b>	Satisfactoria
<b>Caso de prueba para el camino básico 2</b>	
<b>Descripción:</b> los datos de entrada es la información a obtener.	
<b>Condición de ejecución:</b> los datos de entrada no se envían.	
<b>Procedimiento prueba automatizada</b>	
<b>Datos de entrada:</b>	arreglo vacío
<b>Tipo de dato esperado:</b>	is_array
<b>Función de evaluación:</b>	
<pre> \$resultadoEsperado="Array"; \$nombrePrueba="Prueba: Obtener Información"; echo \$this-&gt;ejecutarPrueba(\$this-&gt;ObtenerInformacion(), \$resultadoEsperado , \$nombrePrueba);     </pre>	
<b>Evaluación del caso de prueba:</b>	Satisfactoria
<b>Resultado final de la prueba:</b> satisfactoria en su totalidad.	



En el **Anexo 10. Casos de pruebas de caja blanca** se describen detalladamente algunos de los casos de pruebas de caja blanca realizados a la propuesta de solución y cuyos resultados fueron satisfactorios en su totalidad.

### 3.7.2 Pruebas de integración

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es coger los módulos probados mediante la prueba de unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño. (Pressman 2005). Para la realización de esta prueba se empleó el método de integración incremental.

#### ➤ Método de integración incremental ascendente

La prueba de integración incremental es la antítesis del enfoque no incremental. El programa se construye y se prueba en pequeños segmentos en los que los errores son más fáciles de aislar y corregir. El enfoque ascendente permite comenzar la integración desde los componentes hojas, y combinar el siguiente que se debe probar con el conjunto que ya ha sido probado. Por esta razón se escogió el enfoque incremental ascendente para la realización de las pruebas de integración de la propuesta de solución. Al finalizar las pruebas de integración no fueron detectados errores asociados a la interacción entre los elementos (agrupaciones funcionales y funcionalidades), ni en la integración con los módulos de los que se obtiene información.

#### ➤ Técnica de casos de prueba de caja negra

En la investigación se utilizan los casos de prueba de caja negra para probar los elementos mencionados anteriormente. Para cada elemento se creó un caso de prueba en donde se recogen todos los datos necesarios. Dentro de estos datos se encuentran:

- **Número de caso de prueba:** corresponde al nombre que se le da al caso de prueba está identificado por las siglas Int\_n, donde Int corresponde a la identificación de las pruebas de integración y n es un número consecutivo que se le asigna.
- **Agrupación a integrar o módulo a integrar:** corresponde al nombre del elemento a probar. Depende de lo que se quiera probar.
- **Condiciones de ejecución:** hace referencia a los pasos a seguir antes de realizar la prueba.
- **Descripción de la prueba:** hace referencia como lo dice su nombre a la descripción de la prueba.



- **Entradas/Pasos de ejecución:** hace referencia a los pasos que se deben realizar para ejecutar la prueba.
- **Resultado esperado:** como lo indica su nombre es el resultado que se espera cuando se ejecutan los pasos.
- **Evaluación:** hace referencia al resultado de la prueba luego de ser ejecutada, donde el resultado puede ser satisfactorio o insatisfactorio.

A continuación se muestra en la **Tabla 5** un ejemplo de caso de prueba de integración. En este se integran la funcionalidad de “Información” a la funcionalidad “Versionar información” de la agrupación funcional “Versionado”. Los demás casos de prueba generados aparecen relacionados en el **Anexo 11. Casos de pruebas de integración.**

Tabla 5. Prueba de integración entre las funcionalidades: Información y Versionar información.

Número de caso de prueba: Int_1	
<b>Funcionalidad a integrar:</b>	Información
<b>Condiciones de ejecución:</b>	Se debe crear la información: Matrícula Pregrado
<b>Descripción de la prueba:</b>	Comprobar que cuando se vaya a crear un versionado de información del área Pregrado, se liste la información Matrícula Pregrado.
<b>Entrada/Pasos de ejecución:</b>	El usuario se dirige a la agrupación funcional Versionado y accede a la funcionalidad Versionar información, selecciona el área Pregrado, escoge la opción de Crear en el área de iconos flotantes. Llena los datos para Crear versionado de información, seleccionando la información Matrícula Pregrado.
<b>Resultado esperado:</b>	Se lista la información: Matrícula Pregrado.
<b>Evaluación:</b>	Satisfactoria

Los casos de prueba se fueron aplicando de forma ascendente, donde se probaron las integraciones entre las funcionalidades y luego las integraciones con los módulos a los que integra la solución. Luego de haber aplicado los casos de prueba se pudo identificar que en el 100 % de los casos el resultado fue satisfactorio.

### 3.7.3 Pruebas de sistema

Este tipo de prueba está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema para verificar que se han integrado adecuadamente todos los elementos del mismo (*hardware* u otro *software*) y que realizan las funciones adecuadas. (Pressman 2005)



## Pruebas funcionales

Las pruebas funcionales tienen por objetivo probar que los sistemas desarrollados cumplen con las funciones específicas para los que han sido creados. La función de un sistema es lo que hace dicho sistema, y normalmente es descrita en una especificación de requerimientos, una especificación funcional o en casos de uso. En el contexto de este trabajo de diploma se utilizó el método de caja negra y la técnica de particiones equivalentes para evaluar funcionalmente la solución.

### ➤ Método de caja negra

Las pruebas de caja negra son donde los probadores o analistas de pruebas no enfocan su atención en cómo se generan las respuestas del sistema, sino en el funcionamiento de la interfaz del sistema. Estas pruebas, también conocidas como *pruebas de comportamiento*, se llevan a cabo sobre la interfaz del *software* y permiten obtener un conjunto de condiciones de entrada que ejercitan completamente todos los requerimientos funcionales del programa.(Pressman 2005). El término “caja negra” se debe a que el comportamiento del componente que se prueba solo puede ser determinado estudiando sus entradas y las salidas obtenidas a partir de ellas.

### ➤ Técnica de particiones equivalentes

Para confeccionar los casos de prueba de caja negra existen distintas técnicas. Entre ellas se encuentra la técnica de particiones equivalentes que se basa en identificar todas las particiones para un sistema o componente. Las particiones de equivalencia son un conjunto de datos donde cada uno de los miembros de los conjuntos debería ser procesado equivalentemente. Los casos de pruebas se diseñan para que las entradas o salidas pertenezcan a estas particiones.

### ➤ Casos de pruebas de caja negra

Para la realización de las pruebas de caja negra se generaron los artefactos “*Diseño de casos de pruebas basado en requerimientos*”. Por cada requerimiento funcional del sistema se generó un documento en donde se recogen todos los datos necesarios para probar la interfaz. A continuación se describe uno de los principales casos de prueba asociado a un requerimiento de prioridad alta. Para ver los diseños de casos de pruebas en su totalidad dirigirse al **Anexo 12. Casos de prueba de caja negra.**



### Condiciones de ejecución

- El usuario debe estar autenticado en el sistema con el rol de Administrador.

**Tabla 6. Parte 1. Caso de prueba caja negra. RF\_V11\_Crear información.**

Escenario	Descripción	Nombre	Descripción
EC 1.1 Insertar datos correctamente.	Este escenario permite insertar correctamente una nueva información a versionar utilizando datos correctos.	V	V
		Matrícula Pregrado	Pequeña descripción de la información Matrícula Pregrado.
EC 1.2 Insertar elementos repetidos.	Este escenario permite insertar una información a versionar utilizando datos existentes en el sistema.	I	V
		Matrícula Pregrado	Pequeña descripción de la información Matrícula Pregrado.
EC 1.3 Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para insertar una información a versionar correctamente.	I	NA
		(Vacío)	-
EC 1.4 Insertar datos incorrectos.	Mediante este escenario se introducen datos incorrectos para insertar una información a versionar.	I	NA
		El usuario introduce menos de 3 caracteres.	-
		I	I
		"fgc%' a"	"fgc%' a"
		I	NA
		El usuario introduce más de 250 caracteres.	-
		V	I
Matrícula Pregrado	El usuario introduce palabras de más de 30 caracteres.		
EC 1.5 Cancelar operación	Mediante este escenario se cancela la operación de crear una información a versionar.	NA	NA
		-	-

**Tabla 7. Parte 2. Caso de prueba caja negra. RF\_V11\_Crear información.**

Estado	Respuesta del sistema	Flujo central
V Habilitado	El sistema muestra el mensaje: El elemento ha sido creado satisfactoriamente.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". A continuación selecciona el área del que va a crear



		la información y en el área de iconos flotantes selecciona la opción de Crear. El usuario inserta los datos correctamente y presiona el botón Aceptar.
V Habilitado	El sistema muestra el mensaje: El elemento ya existe.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". A continuación selecciona el área del que va a crear la información y en el área de iconos flotantes selecciona la opción de Crear. El usuario inserta los datos repetidos y presiona el botón Aceptar.
V Habilitado	El sistema muestra en rojo el mensaje: "Campo requerido" sobre el campo que debe ser llenado de forma obligatoria.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". A continuación selecciona el área del que va a crear la información y en el área de iconos flotantes selecciona la opción de Crear. El usuario inserta los datos incompletos y presiona el botón Aceptar.
V Deshabilitado	El sistema muestra el mensaje de error: "Entre al menos 3 caracteres." sobre el campo requerido.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". A continuación selecciona el área del que va a crear la información y en el área de iconos flotantes selecciona la opción de Crear. El usuario inserta los datos incorrectamente y presiona el botón Aceptar.
V Deshabilitado	El sistema muestra en rojo el mensaje: "No se admite apóstrofes ni porcentos." sobre el campo requerido.	
V Deshabilitado	El sistema no permite seguir escribiendo.	
V Deshabilitado	El sistema muestra el mensaje: "Ha excedido el número de letras permitidas para una palabra"	
V Deshabilitado	El sistema muestra el mensaje de confirmación ¿Está seguro de realizar la acción?	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". A continuación selecciona el área del que va a crear la información y en el área de iconos flotantes selecciona la opción de Crear. El usuario inserta o no los datos correctamente y presiona el botón Cancelar.

La realización de estas pruebas permitió detectar algunas no conformidades la cuales fueron erradicadas inmediatamente. A continuación se muestra la relación de no conformidades por iteración:



Tabla 8. No conformidades de las pruebas funcionales.

Iteración	Cantidad de no conformidades	Asociadas a
1ra	26	Errores de interfaz, de validación y no visualización de los mensajes de confirmación.
2da	12	Errores de interfaz y errores ortográficos.
3ra	-	No se encontraron no conformidades.

Para ver gráficamente los resultados de las pruebas funcionales dirigirse al **Anexo 13. Resultados de las pruebas funcionales.**

### 3.7.4 Pruebas de aceptación

Las pruebas de aceptación son básicamente pruebas funcionales sobre el sistema completo, ya que tienen como objetivo obtener la aceptación final del cliente antes de la entrega del producto para su paso a producción. Su ejecución es facultativa del cliente, y en el caso de que no se realicen explícitamente, se dan por incluidas dentro de las pruebas del sistema. Las pruebas de aceptación son, a menudo, responsabilidad del usuario o del cliente, aunque cualquier persona involucrada en el negocio puede realizarlas.

La ejecución de las pruebas de aceptación requiere un entorno de pruebas que represente el entorno de producción. (Pressman 2005). Para el desarrollo de estas pruebas se tomó como referencia el artefacto de especificación de requerimientos de *software* y se comprobó que el sistema cubre satisfactoriamente los requerimientos acordados. Por tanto, el desarrollo de esta prueba fue satisfactorio, quedando como constancia el acta de aceptación del cliente (*Adjunto al documento*).

#### ➤ Prueba alfa

La prueba alfa se lleva a cabo, por un cliente, en el lugar de desarrollo. Se usa el *software* de forma natural con el desarrollador como observador del usuario y registrando los errores y los problemas de uso. Las pruebas alfa se llevan a cabo en un entorno controlado.

### 3.8 Conclusiones parciales

En este capítulo se detallaron los estándares de diseño y codificación empleados en la implementación de la solución. Se describieron las técnicas para validar los requerimientos y las pruebas utilizadas para asegurar la calidad del *software*. Los resultados fueron satisfactorios, asegurando que la solución implementada no contiene errores y tiene la aceptación requerida.





## **CONCLUSIONES GENERALES**

- El estudio de los diferentes sistemas homólogos en el mundo y en Cuba, permitió obtener los conocimientos necesarios para una mayor comprensión del objeto de estudio y sentar las bases de la investigación. Además quedó demostrado que no cumplen con las necesidades actuales del Sistema de Gestión Universitaria, por lo que quedan descartados para ser utilizados.
- El proceso de desarrollo de *software* utilizado guió satisfactoriamente la elaboración de la solución, garantizando la obtención de los artefactos definidos.
- Los patrones de diseño y estándares de codificación usados arrojaron los resultados esperados, logrando una mayor organización y homogeneidad en la implementación de la propuesta de solución.
- Los resultados obtenidos por las pruebas realizadas, mostraron el cumplimiento de los requerimientos, la estabilidad y la calidad necesaria de la solución para un correcto funcionamiento.



## **RECOMENDACIONES**

- Agregar funcionalidades a la solución que permitan a un proveedor con información versionada decidir si actualiza la misma por otra.
- Extender la sincronización a otros grupos de información.



## **BIBLIOGRAFÍA REFERENCIADA**

- **Altamira. (2011).** "Base de Datos de empleados de Altamira Employees." [En línea 2011] [Citado 27 de Diciembre de 2012], Disponible en: <http://www.altamirahrm.com/es/software/sistema-de-recursos-humanos/features/base-de-datos-empleados>.
- **Alvarez, Miguel Angel (2009).** "Introducción a jQuery." [En línea 2009] [Citado 10 de Enero de 2013], Disponible en: <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>.
- **Anzaldo, Juan (2005).** Breve Historia de las Bases de datos. [Citado 20 de Diciembre de 2012].
- **Apache. (2009).** "Apache Joomla en Colombia." [En línea 2009] [Citado 15 de Enero de 2013], Disponible en: <http://www.joomlaencolombia.net/zona-newbie/diccionario/118-apache>
- **Billy Reynoso, Carlos. (2004).** Introducción a la Arquitectura de Software. [Citado 18 de Marzo de 2013].
- **Blaha, Michael. (2010).** "Patterns of Data Modeling. Emerging Directions in Database Systems and Applications. Edición ilustrada. [s.l.]: Taylor & Francis Group,." [En línea 2010] [Citado 3 Enero 2012]. Disponible en: [http://eva.uci.edu/mod/resource/view.php?id=11576&subdir=/Diseno\\_de\\_BD/Patrones\\_de\\_diseno\\_](http://eva.uci.edu/mod/resource/view.php?id=11576&subdir=/Diseno_de_BD/Patrones_de_diseno_) en:
- **Bonatata, Maximiliano. (2003).** "Programación y Algoritmos.". Edición ilustrada. Buenos Aires: MP Ediciones, 2003 ISBN: 9789875261563. [En línea 2003] [Citado 5 Marzo 2013]. Disponible en: <http://sunshine.prod.uci.edu/search/programaci%C3%B3n%20orientada%20a%20objetos>.
- **Borrell, G. (2006).** " WikiLearning. El control de Versiones." [En línea 2006] [Citado 5 de Enero de 2013], Disponible en: [http://www.wikilearning.com/tutorial/control\\_de\\_versiones\\_con\\_subversion-el\\_control\\_de\\_versiones/19266-1](http://www.wikilearning.com/tutorial/control_de_versiones_con_subversion-el_control_de_versiones/19266-1).
- **Camallea, Noel L.Nuñez. (2004).** Gestión de bases de datos [Citado 6 de Enero de 2013].
- **Castro Llano, Eilen. (2012).** PROPUESTA PARA LA INTEGRACIÓN DE PRÁCTICAS DE LAS METODOLOGÍAS XP Y SCRUM CON EL PROCESO DE ADMINISTRACIÓN DE REQUISITOS DEL NIVEL 2 DE CMMI. [Citado el: 8 de Febrero de 2013].
- **Celano, José. (2009).** "Blog iQ Ingenieros. Cassandra DB: ¿Qué tienen Facebook, Twitter, y Digg en común?." [En línea 2009] [Citado 8 de enero de 2013], Disponible en: <http://blog.iqingenieros.com/2009/11/cassandra-db-que-tienen-facebook.html>.
- **Definicion.de. (2013).** "General , definición de personal"., [En línea 2013] [Citado 28 de Mayo de 2013], Disponible en: <http://definicion.de/personal/>.
- **Definicion.de. (2013).** "General tecnología, definición de datos.", [En línea 2013] [Citado 11 de Enero de 2013], Disponible en: <http://definicion.de/datos/>.
- **Definicion.de. (2013).** "General tecnología, definición de versión." , [En línea 2013] [Citado 11 de Enero 2013], Disponible en: <http://definicion.de/version/>
- **Definicionabc. (2013).** "Tecnología, información.", [En línea 2013] [Citado 11 de Enero de 2013], Disponible en: <http://www.definicionabc.com/tecnologia/informacion.php>.
- **DSR, HP. (2009).** "Suite GP RRHH." [En línea 2009] [Citado 11 de Febrero de 2013], Disponible en: [http://www.gestiondelpersonal.com/\\_pdf/RRHH-Avanzados](http://www.gestiondelpersonal.com/_pdf/RRHH-Avanzados).



- **EIDV. (2012).** "EIDV Soluciones Tecnológicas." [En línea 2012] [Citado 13 de Febrero de 2013.], Disponible en: <http://www.eidv.com.ar/articulos/tecnologia-de-lainformacion/lenguaje-unificado-de-modelado/>.
- **García Vidal, Yanio. (2012).** Documento de Arquitectura de Gestión Universitaria. [Citado 15 de Enero de 2013].
- **group. (2010).** "Pencil Project working." [En línea 2010.] [Citado 12 de Febrero de 2013.], Disponible en: <http://pencil.evolus.vn/en-US/Home.aspx>.
- **Guglielmetti, Marcos. (2004).** "MASTERMAGAZINE." [En línea 2004] [Citado 14 de Marzo de 2012.] Disponible en: <http://www.mastermagazine.info/termino/4294.php>.
- **Herrera, Mauricio. (2012).** " Apache Cassandra." [En línea 2012] [Citado 28 de enero de 2013], Disponible en: <http://maoherrera.blogspot.com/2012/11/breve-resena-historica-creado-para.html>.
- **Hinostroza, Raul Rodas. (2005).** "Usuarios y Grupos en Linux " [En línea 2005] [Citado 6 de Enero de 2013], Disponible en: <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>
- **Lago, Ramiro. (2007).** "Patrones de diseño software." [En línea 2007.] [Citado 24 de Marzo de 2012.] , Disponible en: [http://www.proactiva-calidad.com/java/patrones/index.html#algunos\\_patrones](http://www.proactiva-calidad.com/java/patrones/index.html#algunos_patrones).
- **Matos, Rosa Maria Garcia. (2004).** Sistemas de Bases de Datos [Citado 12 de Enero de 2013].
- **Melián, Marlene Montalvo (2009).** "XML el nuevo lenguaje universal." [En línea 2009] [Citado 10 de Enero de 2013], Disponible en: <http://www.bibliociencias.cu/gsd/collect/eventos/archives/HASH0104/f016d031.dir/doc.pdf>.
- **Pedraza, Julian. (2009).** " Netbeans." [En línea 2009] . [Citado 18 de Febrero de 2013], Disponible en: [http://julianpedraza.blogspot.com/2009\\_08\\_01\\_archive.html](http://julianpedraza.blogspot.com/2009_08_01_archive.html).
- **Pelaez, Juan. Carlos. (2009).** "Arquitectura basada en capas. Extractado de La Guía de Arquitectura Versión 2.0a del grupo de Patterns and Practices de Microsoft.", [En línea 2012] [Citado 20 de Mayo de 2013], Disponible en: <http://geeks.ms/blogs/jkpelaez/archive/2009/05/29/arquitectura-basada-en-capas.aspx>.
- **Pérez, Javier Eguíluz (2009).** "¿Qué es JavaScript?"[En línea 2009] [Citado 10 de Enero de 2013] Disponible en: <http://www.librosweb.es/javascript>.
- **Pgadmin. (2010).** "Pgadmin PostgreSQL Tools." [En línea 2010] [Citado 28 de Noviembre de 2011], Disponible en: <http://www.pgadmin.org/>.
- **Pressman, Roger. S. (2005).** Ingeniería del Software un enfoque práctico. Pruebas. [Citado 15 de Abril de 2013]. Disponible en: [http://eva.uci.cu/mod/resource/view.php?id=8500&subdir=/Ediciones\\_del\\_Pressman/Pressman\\_5ta\\_edicion](http://eva.uci.cu/mod/resource/view.php?id=8500&subdir=/Ediciones_del_Pressman/Pressman_5ta_edicion).
- **Pressman, Roger. S. (2007).** Ingeniería del Software un enfoque práctico. Estrategia de pruebas. Quinta Edición. s.l. : Mc Graw Hill, 2007, 305 p, ISBN: 97-0105-473-3. [Citado 5 de Marzo de 2013]. Disponible en: [http://eva.uci.cu/mod/resource/view.php?id=8500&subdir=/Ediciones\\_del\\_Pressman/Pressman\\_6ta\\_edicion](http://eva.uci.cu/mod/resource/view.php?id=8500&subdir=/Ediciones_del_Pressman/Pressman_6ta_edicion)
- **Rafaelma. (2004).** " PostreSQL\_es."[En línea 2010] [Citado 1 de Febrero de 2013], Disponible en: <http://www.postgresql.org>.



- 
- **Rozic, Sergio Ezequiel. (2004).** Bases de datos y sus aplicaciones con SQL. [Citado 10 de Enero de 2013].
  - **Scribd inc. (2012).** "Scribd." [En línea 2012][Citado 2 de Marzo de 2013.] Disponible en: <http://www.scribd.com/about>.
  - **Sommerville, Ian. (2005).** Ingeniería del Software. Séptima Edición. [Citado 15 de Febrero de 2013].
  - **Synergix. (2010).** "Tecnología y synergix. Modelo de dominio." [En línea 2010] [Citado 12 de Febrero de 2013.], Disponible en: <http://synergix.wordpress.com/?s=modelo+dominio>.



## **BIBLIOGRAFÍA CONSULTADA**

- **ÁLVAREZ, Sara.** Arquitectura cliente-servidor. [En línea]. [Fecha de consulta: 25 de febrero de 2012]. Disponible en: <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>
- **GIBERT GINESTÁ, M. y PÉREZ MORA, O.** *Bases de Datos en PostgreSQL*. Disponible en: [http://dataprix.com/files/UOC\\_OpenSource\\_bases\\_de\\_datos\\_en\\_PostgreSQL.pdf](http://dataprix.com/files/UOC_OpenSource_bases_de_datos_en_PostgreSQL.pdf).
- **PEÑÁLVER Romero, G. M.** *MA-GMPR-UR2 Metodología Ágil para proyectos de software libre*. Tutor: Ing. Meneses Abad, A. Facultad 10. Universidad de las Ciencias Informáticas (UCI), 2008.
- **POSTGRESQL GLOBAL DEVELOPMENT GROUP.** *PostgreSQL: The world's most advanced open source database* [En línea]. [Fecha de consulta: 23 de marzo de 2012]. Disponible en: <http://www.postgresql.org/>
- **PRESSMAN, Roger S.** *Ingeniería de Software: Un enfoque práctico*. 5ta Edición. New York: McGraw-Hill Companies, 2010 [fecha de consulta: 5 Octubre 2012]. Disponible en: [http://eva.uci.cu/mod/resource/view.php?id=8500&subdir=/Ediciones\\_del\\_Pressman/Pressman\\_5ta\\_edicion](http://eva.uci.cu/mod/resource/view.php?id=8500&subdir=/Ediciones_del_Pressman/Pressman_5ta_edicion).
- **JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH.** *El Proceso Unificado de Desarrollo de Software* [en línea]. Madrid: Pearson Education S.A, 2000 [fecha de consulta: 5 Octubre 2011]. Disponible en: [http://eva.uci.cu/mod/resource/view.php?id=8500&subdir=/El\\_Proceso\\_Unificado\\_de\\_Desarrollo](http://eva.uci.cu/mod/resource/view.php?id=8500&subdir=/El_Proceso_Unificado_de_Desarrollo). ISBN: 8478290362
- **GARCÍA VIDAL, Yanio.** Documento de Arquitectura de Gestión Universitaria. Patrones Gof. La Habana: s.n., Marzo de 2012.
- **GARCÍA VIDAL, Yanio.** Documento de Arquitectura de Gestión Universitaria. Patrones GRASP. La Habana : s.n., Marzo de 2012.
- **SCRIBD, C.** Diagrama de despliegue. Diagrama de despliegue. [En línea] Marzo de 2009. [Citado el: 1 de Marzo de 2012.] <http://www.scribd.com/doc/53551175/11/>.
- **SOMMERVILLE, Ian.** *Técnicas de validación de requerimientos, Ingeniería del Software*. Septima Edición. Madrid : Pearson Educación.S.A., 2005, 144 p, ISBN 84-7829-074-5.
- **PRESSMAN, Roger S.** *Ingeniería del Software un enfoque práctico. Estrategias de pruebas*. Quinta Edición. s.l. : Mc Graw Hill, 2007, 305 p, ISBN: 97-0105-473-3.
- **Matos, Rosa Maria Garcia. (2004).** *Sistemas de Bases de Datos* [Citado el: 12 de Enero de 2013].
- **Castro Llano, Eilen. (2012).** *PROPUESTA PARA LA INTEGRACIÓN DE PRÁCTICAS DE LAS METODOLOGÍAS XP Y SCRUM CON EL PROCESO DE ADMINISTRACIÓN DE REQUISITOS DEL NIVEL 2 DE CMMI*. [Citado el: 8 de Febrero de 2013].
- **Lago, Ramiro. (2007).** "Patrones de diseño software." [En línea 2007.] [Citado el: 24 de Marzo de 2012.] Disponible en: [http://www.proactiva-calidad.com/java/patrones/index.html#algunos\\_patrones](http://www.proactiva-calidad.com/java/patrones/index.html#algunos_patrones).
- **Rozic, Sergio Ezequiel. (2004).** *Bases de datos y sus aplicaciones con SQL*. [Citado el: 10 de Enero de 2013].



## GLOSARIO DE TÉRMINOS

**A**

**Aplicación o Solución Informática:** Programas con los cuales el usuario final interactúa a través de una interfaz y realizan tareas útiles para este.

**C**

**Control:** Comprobar, inspeccionar, intervenir, dirigir o dominar una organización o sistema.

**Clúster:** Un clúster es simplemente una colección de componentes que se unen y trabajan como un solo componente para proveer alta disponibilidad. Cuando hablamos de *clusters* de bases de datos, nos referimos a una arquitectura en la que tenemos varios equipos con parte de los datos del usuario trabajando al unísono como un solo sistema. La arquitectura de un clúster de base de datos viene definida por la manera en que se almacenan los datos en cada nodo.

**D**

**DOM:** *Document Object Model* es una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos.

**E**

**Estereotipos:** Mecanismos de extensión del lenguaje UML, pequeñas etiquetas que aplicadas a los elementos o relaciones de un diagrama indican significado adicional con miras a incrementar el campo de aplicación del lenguaje.

**F**

**FTP:** *File Transfer Protocol* (Protocolo Transferencias Archivos) es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor.

**H**

**HTTPS:** *Hypertext Transfer Protocol Secure* (Protocolo Seguro de Transferencia de Hipertexto) es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hipertexto, es decir, es la versión segura de HTTP.

**HTML:** *Hypertext Markup Language* (Lenguaje de Marcas de Hipertexto) es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

**I**

**Informatizar:** Proceso de aplicar sistemas o equipos informáticos al tratamiento de la información.

**IDE:** *Integrated Development Environment* (Entorno de desarrollo integrado) es un programa compuesto por un conjunto de herramientas para un programador.

**M**

**Metadatos:** Los metadatos son datos altamente estructurados que describen información, describen el contenido, la calidad, la condición y otras características de los datos. Es "Información sobre información" o "datos sobre los datos".



## ANEXOS

### Anexo No.1 Descripción de las clases del modelo de dominio

Tabla 9. Descripción de las clases del modelo de dominio.

Clases del modelo	Descripción de la clase
<b>Sistema</b>	Se refiere al Sistema de Gestión Universitaria, la cual genera toda la información que se maneja de los procesos universitarios en la UCI.
<b>Atributo</b>	Contiene todas las columnas de cada una de las tablas asociadas a la información que se va a versionar.
<b>Versión</b>	Contiene la versión definida para cada información versionada.
<b>Objeto</b>	Contiene todos los objetos resultantes de una información y la versión asociada a cada uno de ellos.
<b>Valores</b>	Contiene los valores de un atributo asociados a sus objetos.
<b>Información personal</b>	Contiene toda la información respecto a una persona.
<b>Información sensible</b>	Hereda los atributos que se consideran sensibles de una persona por su importancia dentro de los procesos universitarios.
<b>Información variable</b>	Hereda los atributos que se consideran variables de una persona, y que pueden cambiar sin afectar el negocio de la investigación.
<b>Cambio</b>	Contiene los cambios que provoca el sistema y que de ser información sensible, entonces hay que controlarlos para garantizar el versionado.
<b>Notificación</b>	Encargada de notificar al Núcleo de todos los cambios ocurridos en la información sensible de una persona.
<b>Núcleo</b>	Clase rectora de todos los sistemas inscritos en el Sistema de Gestión Universitaria a nivel de bases de datos; controla todos los cambios que le son notificados de una base de datos y verifica a cuáles bases de datos puede actualizar.
<b>Subsistemas</b>	Encargada de identificar la prioridad entre las áreas con información personal para decidir cuál de ellas debe sincronizarse primero.
<b>Sincronización</b>	Encargada de sincronizar las bases de datos para que todas estén actualizadas, garantizando el versionado de la información personal.
<b>Indicadores</b>	Registra todos los indicadores de sincronización y así mantener un registro de los cambios efectuados para posteriormente ejecutar el versionado si la información lo requiere.





## Anexo No.2 Tabla Resumen de la información personal sabana

No	Sabana Campos	BD Pregrado		BD Cooperacion		BD Residencia		BD Postgrado	
		Sencible	Variable	Sencible	Variable	Sencible	Variable	Sencible	Variable
1	Id persona		x		x		x		x
2	Primer nombre	x		x		x		x	
3	Segundo nombre	x		x		x		x	
4	Primer apellido	x		x		x		x	
5	Segundo apellido	x		x		x		x	
6	Carnet de identidad	x		x		x		x	
7	Serie del carnet		x		x		x		x
8	Id expediente		x		x		x		x
9	Nombre usuario		x		x		x		x
10	Id foto		x		x		x		x
11	N sexo	x		x		x		x	
13	N color piel	x		x			x		x
15	N color pelo		x		x		x		x
17	N color ojos		x	x			x		x
19	N discapacidad		x		x		x		x
21	N estado civil	x			x		x	x	
23	N ciudadanía		x	x			x	x	
25	N nacionalidad		x	x			x	x	
27	N municipio	x		x			x		x
29	N provincia	x		x			x		x
31	N estructura		x		x		x		x
33	N responsabilidad		x		x		x		x
35	N rol universitario		x		x		x		x
37	N categoria docente		x		x		x		x
39	N categoria investigativa		x		x		x		x
41	N grado academico		x		x		x		x
43	Peso		x		x		x		x
44	Estatura		x	x			x		x
45	Telefono		x		x		x		x
46	Telefono celular		x		x		x		x
47	Fax		x		x		x		x
48	Correo electronico		x		x		x		x
49	Baja		x		x		x		x
50	Tiene foto		x		x		x		x
	<b>Total: 50</b>	<b>15</b>	<b>35</b>	<b>20</b>	<b>30</b>	<b>7</b>	<b>41</b>	<b>13</b>	<b>35</b>

Figura 25. Tabla Resumen de la información personal sabana.



## Anexo No.3 Modelos de entrevistas

### MODELO DE ENTREVISTA

Nombre:

Apellidos:

Cargo que Ocupa:

Departamento al que pertenece:

#### PREGUNTAS:

1. ¿En qué consiste el proceso de versionado de la información?
2. Características fundamentales del versionado.
3. ¿A qué sistemas va enfocado el versionado?
4. Objetivo principal del versionado de información.
5. ¿Existe algún mecanismo para llevar a cabo el proceso en la actualidad?
6. ¿Qué beneficios brinda el proceso de versionado?
7. Tiempo que demora el proceso de versionado.
8. ¿Existen herramientas que ayuden a la ejecución del proceso?
9. ¿Qué limitaciones presenta el proceso actualmente?
10. ¿Qué desearía aportar al proceso para hacer el trabajo más eficiente?

**Figura 26. Modelo#1 de entrevista para obtención de requerimientos.**

### MODELO DE ENTREVISTA

Nombre:

Apellidos:

Cargo que Ocupa:

Departamento al que pertenece:

#### PREGUNTAS:

1. ¿En qué consiste la información sensible de una persona?
2. ¿A qué sistemas va enfocado la definición de información sensible?
3. Objetivo principal de identificar la información sensible.
4. ¿Existe algún mecanismo para llevar a cabo la identificación?
5. ¿Qué beneficios brindará esta identificación?
6. ¿Qué limitaciones presentaría esta identificación?

**Figura 27. Modelo#2 de entrevista para identificar información sensible.**



## Anexo No.4 Especificación de requerimientos funcionales

Tabla 10. Especificación de requerimiento. Identificar cambio en la información sensible.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF1]	Identificar cambio en la información sensible.	Permite identificar cualquier cambio que ocurra sobre la información sensible de la persona en cualquier área.	Alta	Alta
	<b>Observaciones</b>			

Tabla 11. Especificación de requerimiento. Identificar quién cambió la información sensible.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF2]	Identificar quién cambió la información sensible.	Permite identificar en que área cambió la información sensible.	Alta	Alta
	<b>Observaciones</b>			

Tabla 12. Especificación de requerimiento. Identificar cuál es la información sensible.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF3]	Identificar cuál es la información sensible.	Permite identificar cuál es la información sensible que cambió para poder establecer un orden de prioridad.	Alta	Alta
	<b>Observaciones</b>			

Tabla 13. Especificación de requerimiento. Notificar el cambio.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF4]	Notificar el cambio.	Se encarga de notificar al núcleo sobre el cambio ocurrido en las áreas de procesos.	Alta	Alta
	<b>Observaciones</b>			

Tabla 14. Especificación de requerimiento. Identificar prioridad de sincronización.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF5]	Identificar prioridad de sincronización	Permite identificar qué información debe ser sincronizada primero.	Alta	Alta
	<b>Observaciones</b>			

Tabla 15. Especificación de requerimiento. Sincronizar información al Núcleo.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF6]	Sincronizar	Se encarga de sincronizar la	Alta	Alta



	información al Núcleo	información al núcleo.		
	<b>Observaciones</b>			

**Tabla 16. Especificación de requerimiento. Sincronizar información a los sistemas.**

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF7]	Sincronizar información a los sistemas	Se encarga de sincronizar la información del Núcleo al resto de las áreas.	Alta	Alta
	<b>Observaciones</b>			

**Tabla 17. Especificación de requerimiento. Identificar indicadores de sincronización.**

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF8]	Identificar indicadores de sincronización	Se encarga de identificar los indicadores de información, para poder llevar un seguimiento de los cambios que se realizan en el sistema.	Media	Alta
	<b>Observaciones</b>			

**Tabla 18. Especificación de requerimiento. Registrar indicadores de sincronización.**

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF9]	Registrar indicadores de sincronización	Se encarga de registrar los indicadores para poder conocer lo que ocurre en el sistema con la información personal.	Media	Alta
	<b>Observaciones</b>			

**Tabla 19. Especificación de requerimiento. Listar información.**

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF10]	Listar información	<p>Muestra un listado con la información a versionar.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Información.</p> <p>Se selecciona el sistema y se muestra un listado con las informaciones a versionar y en el área de iconos internos las opciones: Ver detalles, Modificar y Añadir filtro.</p> <p>En el área de iconos flotantes se</p>	Media	Media




























	muestran las opciones: Crear, Actualizar y Ayuda.		
<b>Prototipo</b>			
<b>Información a versionar</b>   			
<b>Sistemas: *</b> <input type="text" value="Pregrado"/> 			
<b>Nombre</b>  <input type="text"/> <span style="float: right;">Cantidad por página <input type="text" value="5"/> </span>			
<input checked="" type="radio"/> Matrícula de pregrado   			
<input checked="" type="radio"/> Pregrado 2   			
<input checked="" type="radio"/> Pregrado INFO   			
<input checked="" type="radio"/> Pregrado 4   			
<input checked="" type="radio"/> Pregrado 3   			
  Página <input type="text" value="1"/> de 2   Resultados encontrados 8			
<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
Sistemas	Varchar	<input checked="" type="checkbox"/> Campo requerido. <input checked="" type="checkbox"/> Campo de selección.	
Cantidad por página	Varchar	<input checked="" type="checkbox"/> Campo de selección.	
<b>Observaciones</b>			

Tabla 20. Especificación de requerimiento. Ver detalles de información.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF11]	Ver detalles de información	<p>Permite ver los detalles de la información a versionar.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Información.</p> <p>Se selecciona el sistema y se muestra un listado con las informaciones a versionar y el área de iconos internos.</p> <p>En el área de iconos internos se selecciona la opción: Ver detalles.</p> <p>Se muestran los datos en forma de ventana emergente: Nombre, Proveedor, Filtro, Estado y</p>	Media	Media



	Descripción.		
<b>Prototipo</b>			
<b>Observaciones</b>			

Tabla 21. Especificación de requerimiento. Crear información.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF12]	Crear información.	<p>Permite registrar la información a versionar.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Información.</p> <p>Se selecciona el sistema sobre el cual se va a crear la nueva información, y se muestra un listado con las informaciones de esa área.</p> <p>En el área de iconos flotantes se selecciona la opción Crear.</p> <p>Para crear la información se introducen los datos: Nombre, Descripción y Estado (Habilitado/Deshabilitado),</p> <p>Cuando se crea una información nueva se muestra el mensaje "El elemento ha sido creado satisfactoriamente".</p> <p>En el área de iconos flotantes se muestran las opciones: Listar y Ayuda.</p>	Alta	Media
<b>Prototipo</b>				



### Crear información

**Nombre: \*** 
**Descripción:**

**Habilitado**

Campos	Tipos de Datos	Reglas o Restricciones
Nombre	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Admite letras, números y guión bajo.</li> <li>✓ Admite entre 3 y 250 caracteres.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>
Descripción:	Varchar	<ul style="list-style-type: none"> <li>✓ Admite entre 0 y 1000 caracteres válidos.</li> <li>✓ Admite letras, números, espacios y guión bajo.</li> <li>✓ Solo admite 30 caracteres por palabra.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>
<b>Observaciones</b>		

**Tabla 22. Especificación de requerimiento. Modificar información.**

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
<b>[RF13]</b>	Modificar información	Permite modificar la información a versionar. El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Información. Se selecciona el sistema y se muestra un listado con las informaciones a versionar y el área de iconos internos.	Media	Media



		<p>En el área de iconos internos se selecciona la opción: Modificar información.</p> <p>Para modificar la información se cambian los datos: Nombre, Descripción y Estado (Habilitado/Deshabilitado),</p> <p>Cuando se modifica una información se muestra el mensaje “El elemento ha sido modificado satisfactoriamente”.</p> <p>En el área de iconos flotantes se muestran las opciones: Listar y Ayuda.</p>		
<b>Prototipo</b>				
<div style="border: 1px solid gray; padding: 10px;"> <div style="text-align: right;">☰ ?</div> <p><b>Modificar información</b></p> <p><b>Nombre: *</b> <input type="text"/></p> <p><b>Descripción:</b> <input style="width: 150px; height: 80px;" type="text"/></p> <p><input checked="" type="checkbox"/> <b>Habilitado</b></p> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </div> </div>				
<b>Campos</b>		<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
Nombre	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Admite letras, números y guión bajo.</li> <li>✓ Admite entre 3 y 250 caracteres.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>		
Descripción:	Varchar	<ul style="list-style-type: none"> <li>✓ Admite entre 0 y 1000 caracteres válidos.</li> <li>✓ Admite letras, números, espacios y guión bajo.</li> <li>✓ Solo admite 30 caracteres por palabra.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>		
<b>Observaciones</b>				





Tabla 23. Especificación de requerimiento. Modificar filtros a información.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF14.]	Modificar filtros a información	<p>Permite modificar los filtros para una información determinada.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Información.</p> <p>Se selecciona el sistema y se muestra un listado con las informaciones a versionar y el área de iconos internos. En el área de iconos internos se selecciona la opción: Modificar filtro de información.</p> <p>Para Modificar un filtro se introducen los datos: Tabla asociada, Atributo, Signo, Valor y Conector de filtro.</p> <p>Cuando se modifica un filtro se muestra el mensaje "El elemento ha sido modificado satisfactoriamente".</p> <p>En el área de iconos flotantes se muestran las opciones: Listar y Ayuda.</p>	Media	Media
<b>Prototipo</b>				
<div style="border: 1px solid gray; padding: 10px;"> <p style="text-align: right;"> </p> <p><b>Modificar filtros a información</b></p> <div style="display: flex; justify-content: space-between;"> <div style="width: 30%;"> <p><b>Tabla asociada:</b></p> <input type="text" value="-Seleccione-"/> </div> <div style="width: 30%;"> <p><b>Atributo:</b></p> <input type="text" value="-Seleccione-"/> </div> <div style="width: 30%;"> <p><b>Signo:</b></p> <input type="text" value="-Seleccione-"/> </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 30%;"> <p><b>Valor:</b></p> <input type="text"/> </div> <div style="width: 30%;"> <p><b>Conector de filtro:</b></p> <input type="text" value="-Seleccione-"/> </div> </div> <div style="display: flex; justify-content: center; margin-top: 10px;"> <input type="button" value="Adicionar"/> <input type="button" value="Borrar"/> </div> <p style="margin-top: 10px;"><b>Filtrado de la Información:</b></p> <div style="display: flex; justify-content: flex-end; margin-top: 10px;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </div> </div>				
<b>Campos</b>		<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	



Tabla asociada	Varchar	✓ Campo de selección.
Atributo	Varchar	✓ Campo de selección.
Signo	Varchar	✓ Campo de selección.
Valor	Varchar	<ul style="list-style-type: none"> <li>✓ Admite letras, números y guión bajo.</li> <li>✓ Admite entre 3 y 250 caracteres.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>
Conector de filtro	Varchar	✓ Campo de selección.
<b>Observaciones</b>		

**Tabla 24. Especificación de requerimiento. Listar tablas asociadas a la información.**

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF15.]	Listar tablas asociadas a la información	<p>Muestra un listado con las tablas asociadas a la información a versionar.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Tablas asociadas.</p> <p>Se escoge el sistema y se muestra un listado con el nombre de las tablas asociadas de ese sistema y en el área de iconos internos las opciones: Ver detalles y Modificar.</p> <p>Se puede realizar la búsqueda más específica a través del nombre de la tabla asociada o mediante un filtro.</p> <p>En el área de iconos flotantes se muestran las opciones: Crear y Actualizar.</p>	Alta	Media
<b>Prototipo</b>				



Campos	Tipos de Datos	Reglas o Restricciones
Criterio de búsqueda	Varchar	<ul style="list-style-type: none"> <li>✓ Admite letras, números y guión bajo.</li> <li>✓ Admite entre 0 y 250 caracteres.</li> <li>✓ Solo admite 30 caracteres por palabra.</li> </ul>
Sistema	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Campo de selección.</li> </ul>
Filtrar búsqueda	Varchar	<ul style="list-style-type: none"> <li>✓ Campo de selección.</li> </ul>
Información	Varchar	<ul style="list-style-type: none"> <li>✓ Campo de selección.</li> </ul>
Cantidad por página	Varchar	<ul style="list-style-type: none"> <li>✓ Campo de selección.</li> </ul>
<b>Observaciones</b>		

Tabla 25. Especificación de requerimiento. Ver detalles de tablas asociadas a la información.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF16.]	Ver detalles de tablas asociadas a la información	<p>Permite ver los detalles de la tabla asociada a una información.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Tablas asociadas.</p>	Media	Media



	<p>Se selecciona el sistema y se muestra un listado con las tablas asociadas a una información y el área de iconos internos.</p> <p>En el área de iconos internos se selecciona la opción: Ver detalles.</p> <p>Se muestran los datos en forma de ventana emergente: Nombre, Información, Tabla a unirse, Condición, Alias, Estado y Descripción.</p>		
<b>Prototipo</b>			
<b>Observaciones</b>			

Tabla 26. Especificación de requerimiento. Crear tablas asociadas a la información.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF17.]	Crear tablas asociadas a la información	<p>Registra los datos de las tablas asociadas a una información.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Tablas asociadas.</p> <p>Se selecciona el sistema sobre el cual se va a crear la nueva tabla asociada, se muestra un listado con las tablas asociadas a esa área.</p> <p>En el área de iconos flotantes se selecciona la opción Crear.</p> <p>Para crear la tabla asociada se</p>	Media	Media



introducen los datos: Nombre, Información, Descripción, Tabla a unirse, Condición, Alias y Estado (Habilitado/Deshabilitado).  
 Cuando se crea una tabla asociada nueva se muestra el mensaje “El elemento ha sido creado satisfactoriamente”.  
 En el área de iconos flotantes se muestran las opciones: Listar y Ayuda.

**Prototipo**

**Crear tabla asociada**

Campos	Tipos de Datos	Reglas o Restricciones
Nombre	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Admite letras, números y guión bajo.</li> <li>✓ Admite entre 3 y 250 caracteres.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>
Información	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Campo de selección.</li> </ul>
Descripción	Varchar	<ul style="list-style-type: none"> <li>✓ Admite entre 0 y 1000 caracteres válidos.</li> <li>✓ Admite letras, números, espacios y guión bajo.</li> <li>✓ Solo admite 30 caracteres por palabra.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>
Tabla a unirse	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Campo de selección.</li> </ul>



	Condición	Varchar	<ul style="list-style-type: none"> <li>✓ Admite entre 3 y 250 caracteres.</li> <li>✓ Admite números, letras, espacio, guión bajo, puntos y paréntesis.</li> </ul>
	Alias	Varchar	<ul style="list-style-type: none"> <li>✓ Admite entre 3 y 250 caracteres válidos.</li> <li>✓ Admite números, letras, espacio y guión bajo.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>
	<b>Observaciones</b>		

**Tabla 27. Especificación de requerimiento. Modificar tablas asociadas a la información.**

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF18.]	Modificar tablas asociadas a la información	<p>Permite modificar los datos de las tablas asociadas a una información.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Tablas asociadas.</p> <p>Se selecciona el sistema y se muestra un listado con las tablas asociadas a la información a versionar y el área de iconos internos.</p> <p>En el área de iconos internos se selecciona la opción: Modificar tabla asociada.</p> <p>Para modificar la tabla asociada se cambian los datos: Nombre, Información, Descripción, Tabla a unirse, Condición, Alias y Estado (Habilitado/Deshabilitado).</p> <p>Cuando se modifica una tabla asociada se muestra el mensaje "El elemento ha sido modificado satisfactoriamente".</p> <p>En el área de iconos flotantes se muestran las opciones: Listar y Ayuda.</p>	Media	Media
	<b>Prototipo</b>			



### Modificar tabla asociada



<b>Nombre: *</b> <input type="text" value="sq_gestion_personal.tb_destudia"/>	<b>Información: *</b> <input type="text" value="-Seleccione-"/>	<b>Descripción:</b> <div style="border: 1px solid gray; height: 100px;"></div>
<b>Tabla a unirse:</b> <input type="text" value="-Seleccione-"/>	<b>Condición:</b> <input type="text" value="tb_dpersona_data_plus.id_pers"/>	
<input checked="" type="checkbox"/> <b>Habilitado</b>	<b>Alias:</b> <input type="text"/>	

Campos	Tipos de Datos	Reglas o Restricciones
Nombre	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Admite letras, números y guión bajo.</li> <li>✓ Admite entre 3 y 250 caracteres.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>
Información	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Campo de selección.</li> </ul>
Descripción	Varchar	<ul style="list-style-type: none"> <li>✓ Admite entre 0 y 1000 caracteres válidos.</li> <li>✓ Admite letras, números, espacios y guión bajo.</li> <li>✓ Solo admite 30 caracteres por palabra.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>
Tabla a unirse	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Campo de selección</li> </ul>
Condición	Varchar	<ul style="list-style-type: none"> <li>✓ Admite entre 3 y 250 caracteres.</li> <li>✓ Admite números, letras, guión bajo, puntos y paréntesis.</li> </ul>
Alias	Varchar	<ul style="list-style-type: none"> <li>✓ Admite entre 3 y 250 caracteres válidos.</li> <li>✓ Admite números, letras, espacio y guión bajo.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>
<b>Observaciones</b>		



Tabla 28. Especificación de requerimiento. Listar atributos de la información.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente												
[RF19.]	Listar atributos de la información	<p>Muestra un listado con los atributos de la información.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Atributos.</p> <p>Se selecciona el sistema y se muestra un listado con los atributos de la información y en el área de iconos internos las opciones: Ver detalles y Modificar.</p> <p>En el área de iconos flotantes se muestran las opciones: Crear, Actualizar y Ayuda.</p>	Alta	Media												
<b>Prototipo</b>																
<div style="border: 1px solid gray; padding: 10px;"> <p><b>Atributo de la información</b> <span style="float: right;">📄 🔄 ?</span></p> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;"> <p style="text-align: right;"><b>Filtrar búsqueda:</b></p> <input style="width: 100%;" type="text" value="nombre de atributo"/> <input style="float: right; margin-right: 10px;" type="button" value="Buscar"/> <div style="float: right;"> <input style="width: 80%;" type="text" value="-Selecione-"/> <input style="width: 20%;" type="button" value="v"/> </div> </div> <div style="display: flex; justify-content: space-between; margin-bottom: 10px;"> <div style="width: 45%;"> <p><b>Sistemas:*</b></p> <input style="width: 90%;" type="text" value="Pregrado"/> <input style="width: 5%;" type="button" value="v"/> </div> <div style="width: 45%;"> <p><b>Información:</b></p> <input style="width: 90%;" type="text" value="-Selecione-"/> <input style="width: 5%;" type="button" value="v"/> </div> </div> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;"> <p style="text-align: right;">Cantidad por página <input type="text" value="5"/> <input type="button" value="v"/></p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 5px;">Nombre <span style="font-size: small;">⌵</span></th> <th style="text-align: right; padding: 5px;"></th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;"><input checked="" type="radio"/> Primer Apellido</td> <td style="text-align: right; padding: 5px;">🔍 🗑️</td> </tr> <tr> <td style="padding: 5px;"><input checked="" type="radio"/> Primer Nombre</td> <td style="text-align: right; padding: 5px;">🔍 🗑️</td> </tr> <tr> <td style="padding: 5px;"><input checked="" type="radio"/> Segundo Apellido</td> <td style="text-align: right; padding: 5px;">🔍 🗑️</td> </tr> <tr> <td style="padding: 5px;"><input checked="" type="radio"/> Segundo Nombre</td> <td style="text-align: right; padding: 5px;">🔍 🗑️</td> </tr> <tr> <td style="padding: 5px;"><input checked="" type="radio"/> Situación Escolar</td> <td style="text-align: right; padding: 5px;">🔍 🗑️</td> </tr> </tbody> </table> </div> <div style="text-align: right; font-size: small; margin-top: 10px;"> <input type="button" value="«"/> <input type="button" value="◀"/> <span style="margin: 0 5px;">Página</span> <input type="text" value="1"/> <span style="margin: 0 5px;">de 2</span> <input type="button" value="▶"/> <input type="button" value="»"/> <span style="margin-left: 20px;">Resultados encontrados 5</span> </div> </div>					Nombre <span style="font-size: small;">⌵</span>		<input checked="" type="radio"/> Primer Apellido	🔍 🗑️	<input checked="" type="radio"/> Primer Nombre	🔍 🗑️	<input checked="" type="radio"/> Segundo Apellido	🔍 🗑️	<input checked="" type="radio"/> Segundo Nombre	🔍 🗑️	<input checked="" type="radio"/> Situación Escolar	🔍 🗑️
Nombre <span style="font-size: small;">⌵</span>																
<input checked="" type="radio"/> Primer Apellido	🔍 🗑️															
<input checked="" type="radio"/> Primer Nombre	🔍 🗑️															
<input checked="" type="radio"/> Segundo Apellido	🔍 🗑️															
<input checked="" type="radio"/> Segundo Nombre	🔍 🗑️															
<input checked="" type="radio"/> Situación Escolar	🔍 🗑️															
Campos	Tipos de Datos		Reglas o Restricciones													
Criterio de búsqueda	Varchar		<ul style="list-style-type: none"> <li>✓ Admite letras, números y guión bajo.</li> <li>✓ Admite entre 0 y 250 caracteres.</li> <li>✓ Solo admite 30 caracteres por palabra.</li> </ul>													





Sistema	Varchar	✓ Campo requerido. ✓ Campo de selección.
Filtrar búsqueda	Varchar	✓ Campo de selección.
Información	Varchar	✓ Campo de selección.
Cantidad por página	Varchar	✓ Campo de selección.
<b>Observaciones</b>		

Tabla 29. Especificación de requerimiento. Ver detalles de atributos de la información.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF20.]	Ver detalles de atributos de la información	<p>Permite ver los detalles de los atributos de la información.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Atributos.</p> <p>Se selecciona el sistema y se muestra un listado con los atributos asociados a una información y el área de iconos internos.</p> <p>En el área de iconos internos se selecciona la opción: Ver detalles.</p> <p>Se muestran los datos en forma de ventana emergente: Nombre, Columna, Información, Tabla asociada, Estado y Descripción.</p>	Media	Media
<b>Prototipo</b>				
<b>Observaciones</b>				



Tabla 30. Especificación de requerimiento. Crear atributos de la información.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF21.]	Crear atributos de la información	<p>Registra los datos de los atributos de la información.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Atributos.</p> <p>Se selecciona el sistema sobre el cual se va a crear el nuevo atributo, y se muestra un listado con los atributos asociados a esa área.</p> <p>En el área de iconos flotantes se selecciona la opción Crear.</p> <p>Para crear el atributo se introducen los datos: Nombre, Información, Descripción, Columna, Tabla asociada y Estado (Habilitado/Deshabilitado).</p> <p>Cuando se crea un atributo nuevo se muestra el mensaje "El elemento ha sido creado satisfactoriamente".</p> <p>En el área de iconos flotantes se muestran las opciones: Listar y Ayuda.</p>	Media	Media
<b>Prototipo</b>				
<div style="border: 1px solid black; padding: 10px;"> <p style="text-align: right;">☰ ?</p> <p><b>Crear atributo</b></p> <div style="display: flex; justify-content: space-between;"> <div style="width: 30%;"> <p><b>Nombre:*</b></p> <input type="text"/></div> <div style="width: 30%;"> <p><b>Información:*</b></p> <div style="border: 1px solid gray; padding: 2px;">-Seleccione-</div> </div> <div style="width: 30%;"> <p><b>Descripción:</b></p> <div style="border: 1px solid gray; height: 80px;"></div> </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 30%;"> <p><b>Columna:*</b></p> <input type="text"/></div> <div style="width: 30%;"> <p><b>Tabla asociada:*</b></p> <div style="border: 1px solid gray; padding: 2px;">-Seleccione-</div> </div> </div> <p><input checked="" type="checkbox"/> <b>Habilitado</b></p> <div style="display: flex; justify-content: flex-end; margin-top: 10px;"> <div style="border: 1px solid gray; padding: 5px 15px; margin-right: 10px;">Aceptar</div> <div style="border: 1px solid gray; padding: 5px 15px;">Cancelar</div> </div> </div>				
<b>Campos</b>		<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
Nombre		Varchar	✓ Campo requerido.	



			<ul style="list-style-type: none"> <li>✓ Admite letras, números y guión bajo.</li> <li>✓ Admite entre 3 y 250 caracteres.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>
	Información	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Campo de selección.</li> </ul>
	Descripción	Varchar	<ul style="list-style-type: none"> <li>✓ Admite entre 0 y 1000 caracteres válidos.</li> <li>✓ Admite letras, números, espacios y guión bajo.</li> <li>✓ Solo admite 30 caracteres por palabra.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>
	Columna	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Admite letras, números y guión bajo.</li> <li>✓ Admite entre 3 y 250 caracteres.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>
	Tabla asociada	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Campo de selección.</li> </ul>
	<b>Observaciones</b>		

**Tabla 31. Especificación de requerimiento. Modificar atributos de la información.**

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF22]	Modificar atributos de la información	<p>Permite modificar los datos de los atributos de la información.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Atributos.</p> <p>Se selecciona el sistema y se muestra un listado con los atributos y el área de iconos internos.</p> <p>En el área de iconos internos se selecciona la opción: Modificar atributo.</p> <p>Para Modificar un atributo se introducen los datos: Nombre, Información, Descripción, Columna, Tabla asociada y Estado (Habilitado/Deshabilitado).</p>	Media	Media



Cuando se modifica un atributo se muestra el mensaje “El elemento ha sido modificado satisfactoriamente”.  
En el área de iconos flotantes se muestran las opciones: Listar y Ayuda.

**Prototipo**

**Modificar atributo**

☰ ?

<b>Nombre:*</b>	<b>Información:*</b>	<b>Descripción:</b>
<input type="text" value="Primer Apellido"/>	<input type="text" value="-Seleccione-"/>	<div style="border: 1px solid gray; height: 100px;"></div>
<b>Columna:*</b>	<b>Tabla asociada:*</b>	
<input type="text" value="primer_apellido"/>	<input type="text" value="-Seleccione-"/>	
<input checked="" type="checkbox"/> <b>Habilitado</b>		

Campos	Tipos de Datos	Reglas o Restricciones
Nombre	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Admite letras, números y guión bajo.</li> <li>✓ Admite entre 3 y 250 caracteres.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>
Información	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Campo de selección.</li> </ul>
Descripción	Varchar	<ul style="list-style-type: none"> <li>✓ Admite entre 0 y 1000 caracteres válidos.</li> <li>✓ Admite letras, números, espacios y guión bajo.</li> <li>✓ Solo admite 30 caracteres por palabra.</li> <li>✓ No admite apóstrofes ni por ciento.</li> </ul>
Columna	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Admite letras, números y guión bajo.</li> <li>✓ Admite entre 3 y 250 caracteres.</li> <li>✓ No admite apóstrofes ni</li> </ul>



			por ciento.
Tabla asociada		Varchar	✓ Campo requerido. ✓ Campo de selección.
<b>Observaciones</b>			

**Tabla 32. Especificación de requerimiento. Listar versión de una información.**

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente																		
[RF23.]	Listar versión de una información	<p>Muestra un listado con las versiones de una información.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Versión.</p> <p>Se selecciona el sistema y se muestra un listado con las versiones de la información y en el área de iconos internos la opción: Ver detalles.</p> <p>En el área de iconos flotantes se muestran las opciones: Actualizar y Ayuda.</p>	Alta	Media																		
<b>Prototipo</b>																						
<div style="border: 1px solid gray; padding: 10px;"> <p><b>Versión de información</b> <span style="float: right;">🔄 ?</span></p> <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 45%;"> <input type="text" value="nombre de versión"/> <input type="button" value="Buscar"/> </div> <div style="width: 45%;"> <p><b>Filtrar búsqueda:</b></p> <input type="text" value="-Seleccione-"/> </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 45%;"> <p><b>Sistemas:*</b></p> <input type="text" value="Pregrado"/> </div> <div style="width: 45%;"> <p><b>Información:</b></p> <input type="text" value="-Seleccione-"/> </div> </div> <div style="margin-top: 20px;"> <p style="text-align: right;">Cantidad por página 5</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Nombre</th> <th style="width: 60%;">Información</th> <th style="width: 20%;"></th> </tr> </thead> <tbody> <tr> <td><input checked="" type="radio"/> 2.5.1</td> <td>Matrícula Pregrado</td> <td></td> </tr> <tr> <td><input checked="" type="radio"/> 2.5.3</td> <td>Matrícula Pregrado</td> <td></td> </tr> <tr> <td><input checked="" type="radio"/> 1.1.1</td> <td>Prueba Pregrado</td> <td></td> </tr> <tr> <td><input checked="" type="radio"/> 1.2.3</td> <td>Prueba Pregrado</td> <td></td> </tr> <tr> <td><input checked="" type="radio"/> 3.5.2</td> <td>Matrícula Pregrado</td> <td></td> </tr> </tbody> </table> <p style="text-align: center; margin-top: 10px;"> <span>«</span> <span>«</span> <span>1</span> <span>»</span> <span>»</span> <span>»</span> Resultados encontrados 5                 </p> </div> </div>					Nombre	Información		<input checked="" type="radio"/> 2.5.1	Matrícula Pregrado		<input checked="" type="radio"/> 2.5.3	Matrícula Pregrado		<input checked="" type="radio"/> 1.1.1	Prueba Pregrado		<input checked="" type="radio"/> 1.2.3	Prueba Pregrado		<input checked="" type="radio"/> 3.5.2	Matrícula Pregrado	
Nombre	Información																					
<input checked="" type="radio"/> 2.5.1	Matrícula Pregrado																					
<input checked="" type="radio"/> 2.5.3	Matrícula Pregrado																					
<input checked="" type="radio"/> 1.1.1	Prueba Pregrado																					
<input checked="" type="radio"/> 1.2.3	Prueba Pregrado																					
<input checked="" type="radio"/> 3.5.2	Matrícula Pregrado																					
<b>Campos</b>		<b>Tipos de Datos</b>		<b>Reglas o Restricciones</b>																		



	Criterio de búsqueda	Varchar	<ul style="list-style-type: none"> <li>✓ Admite letras, números y guión bajo.</li> <li>✓ Admite entre 0 y 250 caracteres.</li> <li>✓ Solo admite 30 caracteres por palabra.</li> <li>✓ Admite entre 3 y 250 caracteres.</li> </ul>
	Sistema	Varchar	<ul style="list-style-type: none"> <li>✓ Campo requerido.</li> <li>✓ Campo de selección.</li> </ul>
	Filtrar búsqueda	Varchar	<ul style="list-style-type: none"> <li>✓ Campo de selección.</li> </ul>
	Información	Varchar	<ul style="list-style-type: none"> <li>✓ Campo de selección.</li> </ul>
	Cantidad por página	Varchar	<ul style="list-style-type: none"> <li>✓ Campo de selección.</li> </ul>
	<b>Observaciones</b>		

**Tabla 33. Especificación de requerimiento. Ver detalles de versión de una información.**

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF24.]	Ver detalles de versión de una información	<p>Permite conocer las características fundamentales de la versión de una información.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Versión.</p> <p>Se selecciona el sistema y se muestra un listado con las versiones asociadas a una información y el área de iconos internos.</p> <p>En el área de iconos internos se selecciona la opción: Ver detalles.</p> <p>Se muestran los datos en forma de ventana emergente: Nombre, Información, Valor de información, Valor del cambio, Estado y Descripción.</p>	Media	Media
<b>Prototipo</b>				



<div style="border: 1px solid gray; padding: 5px; background-color: #f0f0f0; display: inline-block;">             Detalles de versión de información <span style="float: right;">✕</span> </div> <p> <b>Nombre:</b> 2.5.1  <b>Información:</b> Matrícula Pregrado  <b>Valor de información:</b> 5  <b>Valor del cambio:</b> 1  <b>Estado:</b> Habilitado  <b>Descripción:</b> </p> <div style="text-align: center; margin-top: 10px;"> <input type="button" value="Cerrar"/> </div>	
<b>Observaciones</b>	

Tabla 34. Especificación de requerimiento. Listar informaciones versionadas.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF25.]	Listar informaciones versionadas	<p>Muestra un listado con las informaciones versionadas.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Versionar información.</p> <p>Se selecciona el sistema y se muestra un listado con los atributos, objetos y valores asociados a una información.</p> <p>En el área de iconos flotantes se muestran las opciones: Crear, Actualizar y Ayuda.</p>	Media	Media
<b>Prototipo</b>				



Campos	Tipos de Datos	Reglas o Restricciones
Sistema	Varchar	✓ Campo requerido. ✓ Campo de selección.
Filtrar búsqueda	Varchar	✓ Campo de selección.
Información	Varchar	✓ Campo de selección.
Versión	Varchar	✓ Campo de selección.
Cantidad por página	Varchar	✓ Campo de selección.
<b>Observaciones</b>		

Tabla 35. Especificación de requerimiento. Versionar información.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF26.]	Versionar información	<p>Se encarga de definir la versión para la información.</p> <p>El administrador selecciona el módulo Configuración del Sistema de Gestión Universitaria y luego en el menú lateral, en la agrupación funcional Versionado selecciona la opción Versionar información.</p> <p>Se selecciona el sistema sobre el cual se va a versionar la información, y se muestra un listado con los atributos, objetos y valores asociados a la misma.</p>	Alta	Alta








		<p>En el área de iconos flotantes se selecciona la opción Crear.</p> <p>Para crear la versión se selecciona la información.</p> <p>Cuando se crea una versión de información se muestra el mensaje "El elemento ha sido creado satisfactoriamente".</p> <p>En el área de iconos flotantes se muestran las opciones: Listar y Ayuda.</p>		
<b>Prototipo</b>				
<p><b>Crear versionado de información</b>  </p> <p><b>Información a versionar:*</b></p> <p><input type="text" value="-Seleccione-"/> </p> <p><input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/></p>				
<b>Campos</b>		<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
Información a versionar		Varchar	✓ Campo requerido. ✓ Campo de selección.	
<b>Observaciones</b>				

Tabla 36. Especificación de requerimiento. Asociar la versión y el atributo de una información.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF27.]	Asociar la versión y el atributo de una información	Se encarga de asociar la versión y el atributo a una información.	Alta	Media
	<b>Observaciones</b>			

## Anexo No.5 Plan de iteraciones.

Tabla 37. Plan de iteraciones.

Iteración	Descripción	Requerimientos a implementar	Duración total
<b>Iteración 1</b>	En esta iteración se van a implementar los requerimientos para el control de los cambios y el versionado de la información personal que tienen una prioridad alta para el cliente.	RF1,RF2,RF3,RF4,RF1, RF15,RF19,RF23,RF26,RF27	01/01/2013-01/03/2013 (2 meses)



<b>Iteración 2</b>	En esta iteración se van a implementar los requerimientos para el control de los cambios y el versionado de la información personal, que tiene una prioridad media para el cliente.	RF10,RF11,RF12,RF13,RF14,RF16,RF17,RF18,RF20,RF21,RF22,RF24,RF25	05/03/2013-07/04/2013 (1 mes)
<b>Iteración 3</b>	En esta iteración se van a implementar los requerimientos para la sincronización de los cambios de la información personal, que tienen una prioridad media y alta para el cliente.	RF5,RF6,RF7,RF8,RF9	22/04/2013-22/05/2013 (1 mes)

## Anexo No.6 Requerimientos no funcionales.

Tabla 38. Requerimientos no funcionales.

<b>Requerimientos no funcionales</b>	
<b>Usabilidad</b>	
<b>1</b>	La solución podrá ser utilizada por cualquier usuario con las siguientes características: <ul style="list-style-type: none"> <li>• Conocimientos básicos relativos al uso de una computadora.</li> <li>• Conocimientos sólidos con administración de base de datos.</li> </ul>
<b>2</b>	El sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo por parte del usuario, los cuales deben poder acceder de manera rápida y efectiva a la información solicitada.
<b>3</b>	El sistema debe adaptarse al lenguaje y términos utilizados por los clientes en la rama abordada con vistas a una mayor comprensión por su parte sobre la herramienta de trabajo.
<b>Seguridad</b>	
<b>4</b>	La seguridad de la base de datos está a nivel de roles, con el fin de mantener la integridad de los datos en función del acceso de cada uno de ellos, trayendo consigo además la protección de la información.
<b>5</b>	El sistema debe brindar protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
<b>Eficiencia</b>	
<b>6</b>	La solución debe ofrecer tiempos de respuestas admisibles para la ejecución de las funciones que garantizan el versionado de la información.
<b>Soporte</b>	
<b>7</b>	La solución tendrá facilidad de mantenimiento, de actualización, que se realizara periódicamente de acuerdo a las necesidades del cliente.
<b>8</b>	El sistema brinda como apoyo una Ayuda contextual en la cual se refleja detalladamente la explicación de cada una de las pantallas con sus respectivas funcionalidades.



<b>Hardware</b>	
<b>9</b>	Los requerimientos mínimos para que el servidor de base de datos funcione correctamente son: Tarjeta de Red: Mas 100 MB/s, Procesador: Intel Core2 Dúo a 2.00 GHZ o superior, RAM: 1 GB o más, Disco duro: 160
<b>10</b>	Para el desarrollo: PC Intel Pentium 4 o superior, CPU 3GHZ o superior, 512 MB RAM o superior, 160 GB HDD o superior.
<b>Rendimiento</b>	
<b>11</b>	Se debe garantizar la consistencia y disponibilidad de la información en todo momento, por lo que se requiere además un tiempo de recuperación mínimo.
<b>12</b>	La solución debe hacer un uso racional de los recursos de <i>hardware</i> .
<b>Restricciones del diseño</b>	
<b>13</b>	El lenguaje de programación deberá ser PHP 5.3 o superior.
<b>14</b>	Como Entorno de Desarrollo Integrado se empleará NetBeans 6.7.1 o una versión superior.
<b>15</b>	Como servidor Web se explotará Apache 2.2.2.
<b>16</b>	El sistema gestor de bases de datos deberá ser PostgreSQL 8.4.1.
<b>17</b>	El diseño de la base de datos se realizará con Visual Paradigm 8.0 o una versión superior.
<b>18</b>	El sistema operativo a utilizar en el entorno de desarrollo deberá ser: GNU Linux.
<b>Interfaces de hardware</b>	
<b>19</b>	La comunicación entre el servidor de aplicaciones y la base de datos se lleva a través del protocolo de conexión segura TCP/IP.
<b>20</b>	La comunicación entre el cliente y el servidor de aplicaciones se lleva a través del protocolo HTTPS.
<b>Portabilidad</b>	
<b>21</b>	El sistema será multiplataforma.
<b>Estándares Aplicables</b>	
<b>22</b>	El diseño de la solución estará guiado por lo establecido en el documento CENIA-SGU-0120_0ArqSWBase-v1.0.
<b>23</b>	La solución estará guiada por un proceso de desarrollo con enfoque ágil basado en el nivel 2 de CMMI.



## Anexo No.7 Modelos de datos

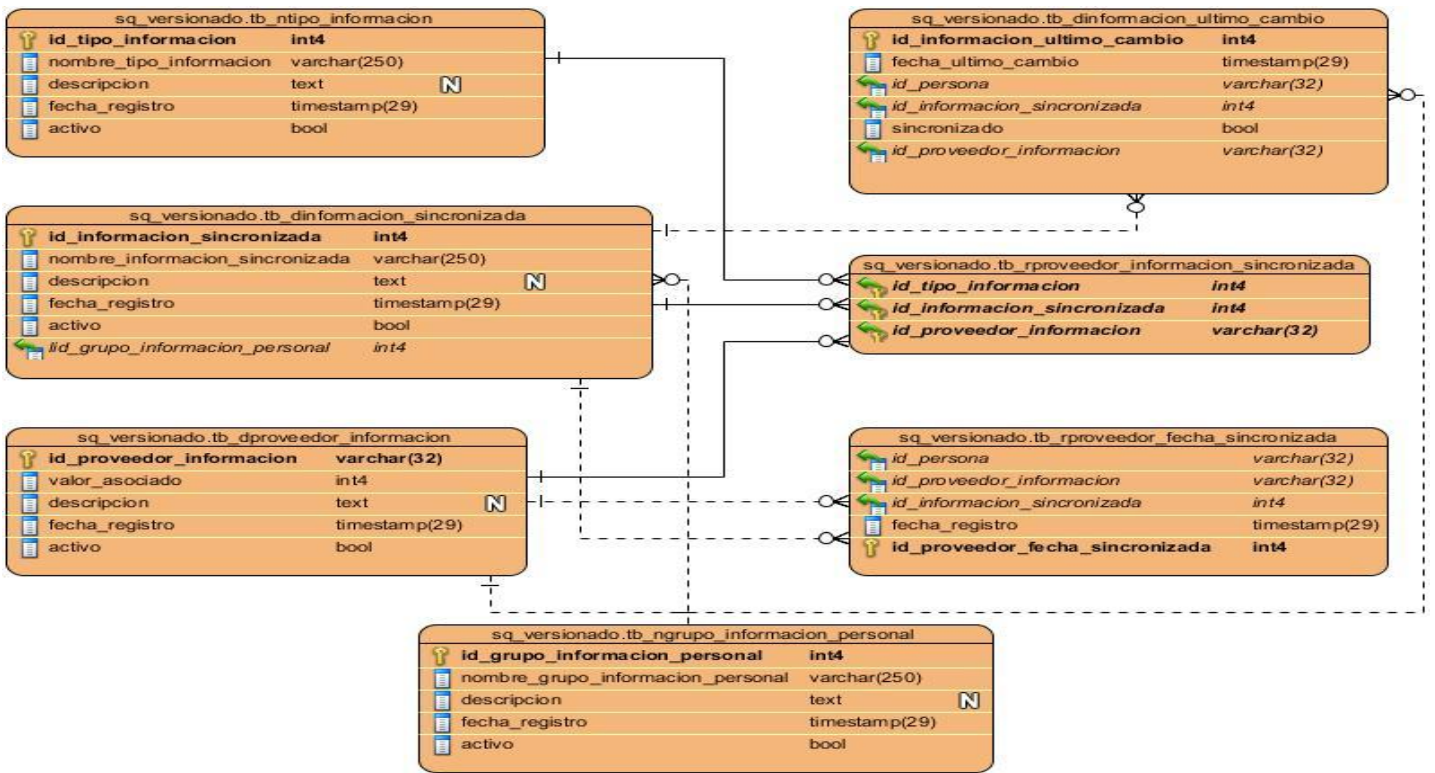


Figura 28. Modelo físico del sincronizador.

## Anexo No.8 Criterios para validar requerimientos del cliente

Tabla 39. Criterios para validar requerimientos del cliente.

Criterios para validar requerimientos del cliente
• ¿El proveedor del REQ es un proveedor válido?
• ¿El REQ está identificado como único?
• ¿El REQ es modificable?
• ¿El REQ no es ambiguo?
• ¿El REQ está completo?
• ¿El REQ es congruente con otros REQ relacionados?
• ¿El REQ puede ser implementado?
• ¿El REQ puede ser probado?
• ¿El resultado de la evaluación de impacto es positivo?
• ¿El REQ está correcto?
• ¿El REQ es traceable?

## Anexo No.9 Descripción de la técnica del camino básico

Esta técnica se basa en obtener una medida de la complejidad del diseño procedimental de un programa (o de la lógica del programa). Esta medida es la complejidad ciclomática de *McCabe*, y representa un



límite superior para el número de casos de prueba que se deben realizar para asegurar que se ejecuta cada camino del programa. Los pasos a realizar para aplicar esta técnica son:

Paso 1. Representar el programa en un grafo de flujo.

Paso 2. Calcular la complejidad ciclomática.

Paso 3. Determinar el conjunto básico de caminos independientes.

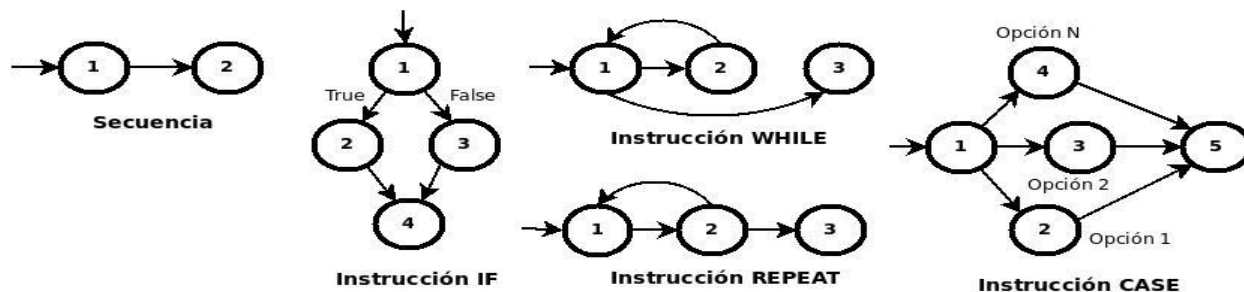
Paso 4. Derivar los casos de prueba que fuerzan la ejecución de cada camino.

A continuación se detallan cada uno de estos pasos.

### Paso 1. Representar el programa en un grafo de flujo

El grafo de flujo se utiliza para representar flujo de control lógico de un programa. Para ello se utilizan los tres elementos siguientes:

- **Nodos:** representan cero, una o varias sentencias en secuencia. Cada nodo comprende como máximo una sentencia de decisión (bifurcación).
- **Aristas:** líneas que unen dos nodos.
- **Regiones:** áreas delimitadas por aristas y nodos. Cuando se contabilizan las regiones de un programa debe incluirse el área externa como una región más.
- **Nodos predicado:** cuando en una condición aparecen uno o más operadores lógicos (AND, OR, XOR...) se crea un nodo distinto por cada una de las condiciones simples. Cada nodo generado de esta forma se denomina nodo predicado. Cada construcción lógica de un programa tiene una representación. La figura siguiente muestra dichas representaciones.



### Representación en grafo de flujo de las estructuras lógicas de un programa.

#### Paso 2. Calcular la complejidad ciclomática

La complejidad ciclomática es una métrica del *software* que proporciona una medida cuantitativa de la complejidad lógica de un programa. En el contexto del método de prueba del camino básico, el valor de la complejidad ciclomática define el número de caminos independientes de dicho programa, y por lo tanto, el número de casos de prueba a realizar. Primero se mostrará cómo calcular ciclomática, a partir de un grafo de flujo, para obtener el número de caminos a identificar. Posteriormente se demostrará cómo se



identifican esos caminos. Existen varias formas de calcular la complejidad ciclomática de un programa a partir de un grafo de flujo:

1. El número de regiones del grafo coincide con la complejidad ciclomática,  $V(G)$ .

2. La complejidad ciclomática,  $V(G)$ , de un grafo de flujo  $G$  se define como:

$$V(G) = \text{Aristas} - \text{Nodos} + 2$$

3. La complejidad ciclomática,  $V(G)$ , de un grafo de flujo  $G$  se define como:

$$V(G) = \text{Nodos Predicado} + 1$$

La complejidad ciclomática determina el número de casos de prueba que deben ejecutarse para garantizar que todas las sentencias de un programa se han ejecutado al menos una vez, y que cada condición se habrá ejecutado en sus vertientes verdadera y falsa.

### **Paso 3. Determinar el conjunto básico de caminos independientes**

Un camino independiente es cualquier camino del programa que introduce, por lo menos, un nuevo conjunto de sentencias de proceso o una condición, respecto a los caminos existentes. En términos del diagrama de flujo, un camino independiente está constituido por lo menos por una arista que no haya sido recorrida antes de la definición del camino. En la identificación de los distintos caminos de un programa para probar se debe tener en cuenta que cada nuevo camino debe tener el mínimo número de sentencias nuevas o condiciones nuevas respecto a los que ya existen. De esta manera, se intenta que el proceso de depuración sea más sencillo. El conjunto de caminos independientes de un grafo no es único. No obstante, a continuación se muestran algunas heurísticas para identificar dichos caminos:

a) Elegir un camino principal que represente una función válida que no sea un tratamiento de error. Debe intentar elegirse el camino que atraviese el máximo número de decisiones en el grafo.

b) Identificar el segundo camino mediante la localización de la primera decisión en el camino de la línea básica alternando su resultado mientras se mantiene el máximo número de decisiones originales del camino inicial.

c) Identificar un tercer camino, colocando la primera decisión en su valor original a la vez que se altera la segunda decisión del camino básico, mientras se intenta mantener el resto de decisiones originales.

d) Continuar el proceso hasta haber conseguido tratar todas las decisiones, intentando mantener como en su origen el resto de ellas.

### **Paso 4. Derivar los casos de prueba que fuerzan la ejecución de cada camino**

El último paso es construir los casos de prueba que fuerzan la ejecución de cada camino.



## Anexo No.10 Casos de pruebas de caja blanca.

Antes de proceder con la descripción de cada uno de los casos de pruebas es necesario tener en cuenta las siguientes consideraciones:

- El caso de prueba es identificado con un código con el formato:

**CENIA\_SGU\_C\_DCP** + *identificador de la acción que se realiza (OI: obtener información, OAB: obtener atributos buscar, RTA: registrar tabla asociada)* + *identificadores de los requerimientos a los que hace referencia (separados por comas).*

Tabla 40. CENIA\_SGU\_C\_DCP -OAB: RFR19, RFR20, RFR21, RFR22.

<b>Prueba estructural de caja blanca</b>	<b>Código caso de prueba: CENIA_SGU_C_DCP -OAB: RFR19, RFR20, RFR21, RFR22.</b>
<b>Probador: Manuel E. Velázquez Sánchez</b>	
<b>Código al que se aplica:</b> <pre> public function obtenerAtributosBuscar(\$datos busqueda) {     \$texto filtro=array();     if (!empty(\$datos busqueda-&gt;text buscar))     {         \$texto filtro[] = "lower(tb datributo.nombre atributo)         LIKE '%'  sq versionado.fn cadena busqueda('" . \$datos busqueda-&gt;text buscar . ")  '%"";     }     if (!empty(\$datos busqueda-&gt;text informaciones))     {         \$texto filtro[] = "tb datributo.id informacion = '" . \$datos busqueda-&gt;text informaciones. "'";     }      \$filtro = implode(' AND ', \$texto filtro);      \$atributos = \$this-&gt; ci-&gt;tb datributo mdl-&gt;obtenerAtributosPorNombre(null,null,'', \$filtro);      return \$atributos; } </pre>	
<b>Complejidad ciclomática:</b> $V(G) = (A - N) + 2 = (6 - 5) + 2 = 3$  <b>Caminos independientes:</b> <ol style="list-style-type: none"> <li>1) 1 - 2 - 3 - 4 - 5</li> <li>2) 1 - 3 - 4 - 5</li> <li>3) 1 - 3 - 5</li> </ol>	<b>Representación del grafo:</b> <pre> graph TD     1((1)) --&gt; 2((2))     1((1)) --&gt; 3((3))     2((2)) --&gt; 3((3))     3((3)) --&gt; 4((4))     3((3)) --&gt; 5((5))     4((4)) --&gt; 5((5)) </pre>



<b>Caso de prueba para el camino básico 1</b>	
<b>Descripción:</b> los datos de entrada es el atributo a obtener.	
<b>Condición de ejecución:</b> se envían los datos de entrada.	
<b>Procedimiento prueba automatizada</b>	
<b>Datos de entrada:</b>	arreglo vacío
<b>Tipo de dato esperado:</b>	is_array
<b>Función de evaluación:</b> \$resultadoEsperado="Array"; \$nombrePrueba="Prueba: Obtener Atributos por Búsqueda"; echo \$this->ejecutarPrueba(\$this->atributo_lib->ObtenerAtributosBuscar(\$datos_búsqueda), \$resultadoEsperado, ,\$nombrePrueba);	
<b>Evaluación del caso de prueba:</b>	Satisfactoria
<b>Caso de prueba para el camino básico 2</b>	
<b>Descripción:</b> los datos de entrada es el atributo a obtener.	
<b>Condición de ejecución:</b> se envían los datos de entrada.	
<b>Procedimiento prueba automatizada</b>	
<b>Datos de entrada:</b>	arreglo vacío
<b>Tipo de dato esperado:</b>	is_array
<b>Función de evaluación:</b> \$resultadoEsperado="Array"; \$nombrePrueba="Prueba: Obtener Atributos por Búsqueda"; <b>echo</b> \$this->ejecutarPrueba(\$this->atributo_lib->ObtenerAtributosBuscar(\$datos_búsqueda), \$resultadoEsperado, ,\$nombrePrueba);	
<b>Evaluación del caso de prueba:</b>	Satisfactoria
<b>Caso de prueba para el camino básico 3</b>	
<b>Descripción:</b> los datos de entrada es el atributo a obtener.	
<b>Condición de ejecución:</b> los datos de entrada no se envían.	
<b>Procedimiento prueba automatizada</b>	
<b>Datos de entrada:</b>	arreglo vacío
<b>Tipo de dato esperado:</b>	is_array
<b>Función de evaluación:</b> \$resultadoEsperado="Array"; \$nombrePrueba="Prueba: Obtener Atributos por Búsqueda";	





<code>echo \$this-&gt;ejecutarPrueba(\$this-&gt;atributo_lib-&gt;ObtenerAtributosBuscar(\$datos_búsqueda), \$resultadoEsperado, \$nombrePrueba);</code>	
<b>Evaluación del caso de prueba:</b>	Satisfactoria
<b>Resultado final de la prueba:</b> satisfactoria en su totalidad.	

Tabla 41. CENIA\_SGU\_C\_DCP -RTA: RFR15, RFR16, RFR17, RFR18.

<b>Prueba estructural de caja blanca</b>	<b>Código caso de prueba:</b> CENIA_SGU_C_DCP -RTA: RFR15, RFR16, RFR17, RFR18.
<b>Probador:</b> Manuel E. Velázquez Sánchez	
<b>Código al que se aplica:</b>	
<pre> public function registrarTablaAsociada() {     if(\$this-&gt;input-&gt;is post back(array('nombre tabla asociada')))     {         \$all post = \$this-&gt;input-&gt;all post(true);          if(\$this-&gt;tabla asociada lib-&gt;chequearNombreRepetidoIOC(\$all post) == true)         {             throw new Exception Error('SYS005');         }         if(\$this-&gt;tabla asociada lib-&gt;registrarTablaAsociadaIOC(\$all post) != false)         {             \$this-&gt;message('SYS001');         }         else         {             throw new Exception Error('SYS006');         }     }     else     {         throw new Exception Error('SYS007');     } } </pre>	



<p><b>Complejidad ciclomática:</b>  <math>V(G) = (A - N) + 2 = (10 - 8) + 2 = 4</math></p> <p><b>Caminos independientes:</b></p> <ol style="list-style-type: none"> <li>1) 1 - 2 - 4 - 5 - 6 - 8</li> <li>2) 1 - 2 - 4 - 5 - 7 - 8</li> <li>3) 1 - 2 - 5 - 7 - 8</li> <li>4) 1 - 3 - 8</li> </ol>	<p><b>Representación del grafo:</b></p>
---	---

**Caso de prueba para el camino básico 1**

**Descripción:** los datos de entrada son los datos de la tabla asociada a registrar.

**Condición de ejecución:** se envían los datos de entrada.

**Procedimiento prueba automatizada**

<b>Datos de entrada:</b>	arreglo vacío
--------------------------	---------------

<b>Tipo de dato esperado:</b>	is_true
-------------------------------	---------

**Función de evaluación:**

```

$resultadoEsperado="true";
$nombrePrueba="Prueba: Registrar Tabla Asociada";
echo $this->ejecutarPrueba($this->RegistrarTablaAsociada(), $resultadoEsperado, $nombrePrueba);
    
```

<b>Evaluación del caso de prueba:</b>	Satisfactoria
---------------------------------------	---------------

**Caso de prueba para el camino básico 2**

**Descripción:** los datos de entrada son los datos de la tabla asociada a registrar.

**Condición de ejecución:** se envían los datos de entrada.

**Procedimiento prueba automatizada**

<b>Datos de entrada:</b>	arreglo vacío
--------------------------	---------------

<b>Tipo de dato esperado:</b>	is_true
-------------------------------	---------

**Función de evaluación:**

```

$resultadoEsperado="true";
$nombrePrueba="Prueba: Registrar Tabla Asociada";
    
```



<code>echo \$this-&gt;ejecutarPrueba(\$this-&gt;RegistrarTablaAsociada(), \$resultadoEsperado , \$nombrePrueba);</code>	
<b>Evaluación del caso de prueba:</b>	Satisfactoria
<b>Caso de prueba para el camino básico 3</b>	
<b>Descripción:</b> los datos de entrada son los datos de la tabla asociada a registrar.	
<b>Condición de ejecución:</b> los datos de entrada no se envían.	
<b>Procedimiento prueba automatizada</b>	
<b>Datos de entrada:</b>	arreglo vacío
<b>Tipo de dato esperado:</b>	is_true
<b>Función de evaluación:</b> \$resultadoEsperado="true"; \$nombrePrueba="Prueba: Registrar Tabla Asociada"; <code>echo \$this-&gt;ejecutarPrueba(\$this-&gt;RegistrarTablaAsociada(), \$resultadoEsperado , \$nombrePrueba);</code>	
<b>Evaluación del caso de prueba:</b>	Satisfactoria
<b>Caso de prueba para el camino básico 4</b>	
<b>Descripción:</b> los datos de entrada son los datos de la tabla asociada a registrar.	
<b>Condición de ejecución:</b> los datos de entrada no se envían.	
<b>Procedimiento prueba automatizada</b>	
<b>Datos de entrada:</b>	arreglo vacío
<b>Tipo de dato esperado:</b>	is_false
<b>Función de evaluación:</b> \$resultadoEsperado="true"; \$nombrePrueba="Prueba: Registrar Tabla Asociada"; <code>echo \$this-&gt;ejecutarPrueba(\$this-&gt;RegistrarTablaAsociada(), \$resultadoEsperado , \$nombrePrueba);</code>	
<b>Evaluación del caso de prueba:</b>	Satisfactoria
<b>Resultado final de la prueba:</b> satisfactoria en su totalidad.	

## Anexo No.11 Pruebas de integración.

Tabla 42. Prueba de integración entre las funcionalidades: Atributos y Versionar información.

<b>Número de caso de prueba:</b> Int_2	
<b>Funcionalidad a integrar:</b>	Atributo
<b>Condiciones de ejecución:</b>	Se debe crear el atributo Primer Apellido para la información Matrícula Pregrado
<b>Descripción de la prueba:</b>	Comprobar que cuando se a versionar la información Matrícula Pregrado se liste el atributo: Primer Apellido.
<b>Entrada/Pasos de ejecución</b>	El usuario se dirige a la agrupación funcional Versionado y accede a la



	funcionalidad Versionar información, selecciona el área Pregrado, busca la información Matrícula Pregrado y se lista el atributo Primer Apellido.
<b>Resultado esperado:</b>	Para la información: Matrícula Pregrado, se liste el atributo: Primer Apellido
<b>Evaluación:</b>	Satisfactoria

**Tabla 43. Prueba de integración entre las funcionalidades: Información y Tablas asociadas.**

<b>Número de caso de prueba: Int_3</b>	
<b>Funcionalidad a integrar:</b>	Información
<b>Condiciones de ejecución:</b>	Se debe crear la información: Cierre Matrícula
<b>Descripción de la prueba:</b>	Comprobar que cuando se vaya a crear una tabla asociada en el área Núcleo de Gestión Universitaria, se liste la información Cierre Matrícula.
<b>Entrada/Pasos de ejecución:</b>	El Administrador se dirige a la agrupación funcional Versionado y accede a la funcionalidad Tablas asociadas, selecciona el área Núcleo de Gestión Universitaria, escoge la opción de Crear en el área de iconos flotantes. Llena los datos para Crear tabla asociada, seleccionando la información: Matrícula Pregrado.
<b>Resultado esperado:</b>	Se lista la información: Cierre Matrícula.
<b>Evaluación:</b>	Satisfactoria

**Tabla 44. Prueba de integración entre las funcionalidades: Información y Atributos.**

<b>Número de caso de prueba: Int_4</b>	
<b>Funcionalidad a integrar:</b>	Información
<b>Condiciones de ejecución:</b>	Se debe crear la información: Especializaciones de persona.
<b>Descripción de la prueba:</b>	Comprobar que cuando se vaya a crear un atributo en el área Núcleo de Gestión Universitaria, se liste la información Especializaciones de persona.
<b>Entrada/Pasos de ejecución:</b>	El Administrador se dirige a la agrupación funcional Versionado y accede a la funcionalidad Atributos, selecciona el área Núcleo de Gestión Universitaria, escoge la opción de Crear en el área de iconos flotantes. Llena los datos para Crear atributo, seleccionando la información Especializaciones de persona.
<b>Resultado esperado:</b>	Se lista la información: Especializaciones de persona.
<b>Evaluación:</b>	Satisfactoria

**Tabla 45. Prueba de integración entre las funcionalidades: Información y Versión.**

<b>Número de caso de prueba: Int_5</b>	
<b>Funcionalidad a integrar:</b>	Información
<b>Condiciones de ejecución:</b>	Se debe crear la información: Matrícula Pregrado.



<b>Descripción de la prueba:</b>	Comprobar que cuando se vaya a listar las versiones de información del área Pregrado se muestre las versiones de la información Matrícula Pregrado.
<b>Entrada/Pasos de ejecución:</b>	El Administrador se dirige a la agrupación funcional Versionado y accede a la funcionalidad Versión, selecciona el área Pregrado y se lista las versiones de la información: Matrícula Pregrado.
<b>Resultado esperado:</b>	Se listan las versiones de información: Matrícula Pregrado.
<b>Evaluación:</b>	Satisfactoria

**Tabla 46. Prueba de integración al Módulo: Núcleo de Gestión Universitaria.**

<b>Número de caso de prueba: Int_6</b>	
<b>Modulo a integrar:</b>	Núcleo de Gestión Universitaria
<b>Condiciones de ejecución:</b>	Se debe versionar la información: Cierre Matrícula.
<b>Descripción de la prueba:</b>	Comprobar que cuando se vaya a listar las informaciones versionadas del área Núcleo de Gestión Universitaria se muestra los datos de la información Cierre Matrícula.
<b>Entrada/Pasos de ejecución:</b>	El Administrador se dirige a la agrupación funcional Versionado y accede a la funcionalidad Versionar información, selecciona el área Núcleo de Gestión Universitaria, y se filtran los datos de las versiones por la información: Cierre Matrícula.
<b>Resultado esperado:</b>	Se listan los datos de la información: Cierre Matrícula.
<b>Evaluación:</b>	Satisfactoria

**Tabla 47. Prueba de integración al Módulo: Pregrado.**

<b>Número de caso de prueba: Int_6</b>	
<b>Modulo a integrar:</b>	Pregrado
<b>Condiciones de ejecución:</b>	Se debe versionar la información: Matrícula Pregrado.
<b>Descripción de la prueba:</b>	Comprobar que cuando se vaya a listar las informaciones versionadas del área Pregrado, se muestren los datos de la información Matrícula Pregrado.
<b>Entrada/Pasos de ejecución:</b>	El Administrador se dirige a la agrupación funcional Versionado y accede a la funcionalidad Versionar información, selecciona el área Pregrado, y se filtran los datos de las versiones por la información: Matrícula Pregrado.
<b>Resultado esperado:</b>	Se listan los datos de la información: Matrícula Pregrado.
<b>Evaluación:</b>	Satisfactoria



## Anexo No.12 Casos de pruebas de caja negra.

Tabla 48. CENIA\_SGU\_C\_DCP\_RF\_V11: Listar\_información.

Escenario	Descripción	Sistemas	Nombre	Encabezado del grid	Cantidad por página	Respuesta del sistema	Flujo central
EC 1.1 Mostrar datos correctamente.	Mediante este escenario se mostrará al usuario el listado de las informaciones a versionar del sistema seleccionado.	V	V	NA	NA	El sistema muestra el listado de los elementos actualizados hasta la fecha que coincida con el sistema seleccionado.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". Seguidamente se selecciona el sistema y se muestra el listado de informaciones a versionar con una barra de iconos internos con las acciones a desarrollar sobre cada elemento: Ver detalles, Modificar y Añadir filtro. Además se muestra una barra de iconos flotantes con las acciones a realizar sobre el contexto: Crear, Actualizar y Ayuda.
		Pregrado	Grupos de informaciones a versionar que coincidan con el sistema seleccionado.	-	Criterio para seleccionar la cantidad de elementos a mostrar por página.		
EC 1.2 Seleccionar correctamente cantidad de elementos por página.	Mediante este escenario el usuario escoge la cantidad de elementos a mostrar por página (los valores a	V	NA	NA	V	El sistema muestra la cantidad de elementos según la opción escogida(Se puede escoger	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". Seguidamente se
		Pregrado	-	-	El usuario escoge la opción de 5		
		V	NA	NA	V		



	escoger son 5, 10, 15 y 20).	Pregrado	-	-	El usuario escoge la opción de 10	5,10,15,20)	selecciona el sistema y se muestra el listado de informaciones a versionar con una barra de iconos internos con las acciones a desarrollar sobre cada elemento: Ver detalles, Modificar y Añadir filtro. Además se muestra una barra de iconos flotantes con las acciones a realizar sobre el contexto: Crear, Actualizar y Ayuda.
		V	NA	NA	V		
		Pregrado	-	-	El usuario escoge la opción de 15		
		V	NA	NA	V		
		Pregrado	-	-	El usuario escoge la opción de 20		
EC 1.3 Ordenar las informaciones a versionar correctamente.	Mediante este escenario se da clic en el encabezado del listado ordenándolo así alfabéticamente. (Icono que se muestra como una flecha).	V	NA	V	NA	El sistema ordena los elementos de forma ascendente. Ordena primero los habilitados y luego los deshabilitados.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". Seguidamente se selecciona el sistema y se muestra el listado de informaciones a versionar con una barra de iconos internos con las acciones a desarrollar sobre cada elemento: Ver detalles, Modificar y Añadir filtro. Además se muestra una barra de iconos flotantes con las acciones a realizar sobre el contexto: Crear, Actualizar y Ayuda.
		Pregrado	-	Se da clic encima del elemento y se activa la flecha de arriba	-		
		V	NA	V	NA	El sistema ordena los elementos de forma descendente. Ordena primero los habilitados y luego los deshabilitados.	
		Pregrado	-	Se da clic encima del elemento y se activa la flecha de abajo.	-		



EC 1.4 No existen elementos creados.	Mediante este escenario en caso de que no exista creado ningún elemento se muestra un listado vacío.	V	V	NA	NA	El sistema muestra el listado vacío.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". Seguidamente se selecciona el sistema y se muestra el listado de informaciones a versionar con una barra de iconos internos con las acciones a desarrollar sobre cada elemento: Ver detalles, Modificar y Añadir filtro. Además se muestra una barra de iconos flotantes con las acciones a realizar sobre el contexto: Crear, Actualizar y Ayuda.
		Pregrado	(Vacío)	-	-		
EC 1.5 Actualizar.	Se actualiza el listado con las informaciones versionadas existentes hasta la fecha.	V	V	NA	NA	El sistema muestra un listado con las informaciones versionadas existentes hasta la fecha.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". Seguidamente se selecciona el sistema y se muestra el listado de informaciones a versionar con una barra de iconos internos con las acciones a desarrollar sobre cada elemento: Ver detalles, Modificar y Añadir filtro. Además se muestra una barra de iconos flotantes con las acciones a realizar sobre el contexto: Crear, Actualizar y Ayuda.
		Pregrado	Grupos de informaciones a versionar que coincidan con el sistema seleccionado.	-	-		





Tabla 49. CENIA\_SGU\_C\_DCP\_RF\_V13: Modificar\_información.

Escenario	Descripción	Nombre	Descripción	Estado	Respuesta del sistema	Flujo central
EC 1.1 Modificar datos correctamente	Este escenario permite modificar una información a versionar utilizando datos correctos.	V	V	V	El sistema muestra el mensaje: El elemento ha sido modificado satisfactoriamente.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". A continuación selecciona el sistema del que va a modificar la información y en el área de iconos internos selecciona la opción de "Modificar". El usuario modifica los datos correctamente y presiona el botón Aceptar.
		Matrícula Pregrado	Pequeña descripción de la información Matrícula Pregrado	Habilitado		
		V	NA	V		
		Cierre Matrícula	-	Deshabilitado		
EC 1.2 Modificar	Mediante este	I	NA	V	El sistema muestra en rojo	El usuario una vez



insertando datos incompletos.	escenario no se introducen todos los datos para modificar una información a versionar correctamente.	(Vacío)	-	Deshabilitado	el mensaje: "Campo requerido" sobre el campo que debe ser llenado de forma obligatoria.	autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". A continuación selecciona el sistema del que va a modificar la información y en el área de iconos internos selecciona la opción de "Modificar". El usuario no introduce todos los campos requeridos y presiona el botón Aceptar.
EC 1.3 Modificar insertando datos incorrectos.	Mediante este escenario se introducen datos incorrectos para modificar una información a versionar.	I	NA	V	El sistema muestra el mensaje de error: "Entre al menos 3 caracteres." sobre el campo requerido.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". A continuación selecciona el sistema del que va a modificar la información y en el área de iconos internos selecciona la
		El usuario introduce menos de 3 caracteres.	-	Habilitado		
		I	I	V	El sistema muestra en rojo el mensaje: "No se admite apóstrofes ni porcientos." sobre el campo requerido.	
		"fgc%' a"	"fgc%' a"	Habilitado		
I	I	V	El sistema no permite			



		El usuario introduce más de 250 caracteres.	El usuario introduce más de 1000 caracteres.	Deshabilitado	seguir escribiendo.	opción de "Modificar". El usuario modifica los datos incorrectamente y presiona el botón Aceptar.
		V	I	V	El sistema muestra el mensaje: "Ha excedido el número de letras permitidas para una palabra"	
		Estudiantes	El usuario introduce palabras de más de 30 caracteres.	Deshabilitado		
EC 1.4 Cancelar operación	Mediante este escenario se cancela la operación de modificar una información a versionar.	NA	NA	NA	El sistema muestra el mensaje de confirmación ¿Está seguro de realizar la acción?	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". A continuación selecciona el sistema del que va a modificar la información y en el área de iconos internos selecciona la
		-	-	-		



						opción de "Modificar". El usuario modifica o no los datos correctamente y presiona el botón Cancelar.
--	--	--	--	--	--	---

Tabla 50. CENIA\_SGU\_C\_DCP\_RF\_V14: Ver\_Detalles\_información.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Ver detalles	Mediante este escenario se muestra toda la información del elemento seleccionado.	El sistema muestra una ventana emergente con los siguientes datos: Nombre, Proveedor, Filtro, Estado (habilitado/deshabilitado) y Descripción.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". A continuación selecciona el sistema del que va a crear la información. Seguidamente selecciona sobre el elemento en el área de iconos internos la opción de "Ver detalles".



EC 1.2 Cerrar la ventana de ver detalles por el botón Cerrar	Mediante este escenario se puede cerrar la ventana de Ver detalles.	El sistema cierra la ventana y muestra el listado con los elementos existentes.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". A continuación selecciona el sistema del que va a crear la información Seguidamente selecciona sobre el elemento en el área de iconos internos la opción de "Ver detalles" y presiona la opción Cerrar.
--	---	---	--

Tabla 51. CENIA\_SGU\_C\_DCP\_RF\_V15: Modificar\_filtro\_información.

Escenario	Descripción	Tabla asociada	Atributo	Signo	Valor	Conector de filtro.	Respuesta del sistema	Flujo central
EC 1.1 Modificar Insertando datos correctamente	Este escenario permite modificar correctamente un filtro a la información a versionar utilizando datos correctos.	V	V	V	V	V	El sistema muestra el mensaje: El elemento ha sido Modificado satisfactoriamente .	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". A continuación
		sq_gestion_personal	nombre_compu esto	diferente	2563	Y		



								selecciona el sistema del que va a modificar el filtro a la información y en el área de iconos interno selecciona la opción de Añadir filtro. El usuario modifica los datos correctamente y presiona el botón Aceptar.
EC 1.2 Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para modificar un filtro de información a versionar correctamente.	I	V	V	V	V	El sistema muestra en rojo el mensaje: "Campo requerido" sobre el campo que debe ser llenado de forma obligatoria.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las
		(Vacío)	nombre_compu esto	igual	2476	Y		
		I	V	V	I	V		
		(Vacío)	nombre_compu esto	igual	(Vacío)	Y		
		V	I	I	V	V		



		sq_gestion_personal	(Vacío)	(Vacío)	2478	O		opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". A continuación selecciona el sistema del que va a modificar el filtro a la información y en el área de iconos interno selecciona la opción de Añadir filtro. El usuario modifica insertando los datos incompletos y presiona el botón Aceptar.
EC 1.3 Insertar datos incorrectos.	Mediante este escenario se introducen datos incorrectos para modificar un filtro de información a versionar.	V	V	V	I	V	El sistema muestra el mensaje de error: "Entre al menos 3 caracteres." sobre el campo requerido.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario
		sq_gestion_personal	nombre_compu esto	contiene	El usuario introduce menos de 3 caracteres.	Y		
		V	V	V	I	V	El sistema	



		sq_gestion_personal	nombre_compu esto	contiene	"fgc%' a"	Y	muestra en rojo el mensaje: "No se admite apóstrofes ni porcientos." sobre el campo requerido.	selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". A continuación selecciona el sistema del que va a modificar el filtro a la información y en el área de iconos interno selecciona la opción de Añadir filtro. El usuario modifica insertando los datos incorrectos y presiona el botón Aceptar.
		V	V	V	I	V	El sistema no permite seguir escribiendo.	
		sq_gestion_personal	nombre_compu esto	contiene	El usuario introdu ce más de 250 caracte res.	Y		
		V	V	V	I	V	El sistema muestra el mensaje: "Ha excedido el número de letras permitidas para una palabra"	
EC 1.4 Cancelar	Mediante este	NA	NA	NA	NA	NA		El usuario una





operación	escenario se cancela la operación de modificar un filtro de una información a versionar.	-	-	-	-	-	El sistema muestra el mensaje de confirmación ¿Está seguro de realizar la acción?	vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Información". A continuación selecciona el sistema del que va a modificar el filtro a la información y en el área de iconos interno selecciona la opción de Añadir filtro. El usuario modifica insertando los datos correctamente y presiona el botón Cancelar.
-----------	--	---	---	---	---	---	---	---



Tabla 52. CENIA\_SGU\_C\_DCP\_RF\_V25: Listar\_informaciones versionadas.

Escenario	Descripción	Sistemas	Nombre	Encabezado del grid	Filtro de búsqueda	Información	Cantidad por página	Respuesta del sistema	Flujo central
EC 1.1 Mostrar datos correctamente.	Mediante este escenario se puede mostrar la lista de informaciones versionadas seleccionando el sistema.	V	V	NA	V	V	NA	El sistema muestra el listado de los elementos actualizados hasta la fecha que coincida con el sistema seleccionado.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Versionar información". Seguidamente se selecciona el sistema y se muestra un listado de atributos, objetos y valores asociados a las informaciones versionadas. Además se muestra una barra de iconos flotantes con las acciones a realizar sobre el contexto: Crear, Actualizar y Ayuda.
		Pregrado	Grupos de informaciones versionadas que coincidan con el sistema y la información seleccionada.	-	Información	Seleccionar una información existente.	Criterio para seleccionar la cantidad de elementos a mostrar por página.		



EC 1.2 Seleccionar correctamente cantidad de elementos por página.	Mediante este escenario el usuario escoge la cantidad de elementos a mostrar por página (los valores a escoger son 5, 10, 15 y 20).	V	V	NA	NA	NA	V	El sistema muestra la cantidad de elementos según la opción escogida (Se puede escoger 5, 10, 15, 20)	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Versionar información". Seguidamente se selecciona el sistema y se muestra un listado de atributos, objetos y valores asociados a las informaciones versionadas. Además se muestra una barra de iconos flotantes con las acciones a realizar sobre el contexto: Crear,
		Pregrado	Grupos de informaciones versionadas que coincidan con el sistema y la información seleccionada.	-	-	-	El usuario escoge la opción de 5		
		V	NA	NA	V	V	V		
		Pregrado	Grupos de informaciones versionadas que coincidan con el sistema y la información seleccionada.	-	Información	Seleccionar una información existente	El usuario escoge la opción de 10		
		V	NA	NA	V	V	V		
		Pregrado	Grupos de informaciones versionadas que coincidan con el sistema y la información seleccionada.	-	Información	Seleccionar una información existente	El usuario escoge la opción de 15		
V	NA	NA	NA	NA	V				



		Pregrado	Grupos de informaciones versionadas que coincidan con el sistema y la información seleccionada.	-	-	-	El usuario escoge la opción de 20		Actualizar y Ayuda.
EC 1.3 Ordenar las informaciones a versionar correctamente.	Mediante este escenario se da clic en el encabezado del listado ordenándolo así alfabéticamente. (Icono que se muestra como una flecha).	V	NA	V	NA	NA	NA	El sistema ordena los elementos de forma ascendente. Ordena primero los habilitados y luego los deshabilitados.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Versionar información". Seguidamente se selecciona el sistema y se muestra un listado de atributos, objetos y valores asociados a las informaciones versionadas. Además se muestra una barra de iconos flotantes con las acciones a realizar sobre el
		Pregrado	-	Se da clic encima del elemento y se activa la flecha de arriba	-	-	-		
		V	NA	V	V	V	NA	El sistema ordena los elementos de forma descendente. Ordena primero los habilitados y luego los deshabilitados.	
		Pregrado	-	Se da clic encima del elemento y se activa la flecha de abajo.	Información	Seleccionar una información existente	-		



									contexto: Crear, Actualizar y Ayuda.
EC 1.4 No existen elementos creados.	Mediante este escenario en caso de que no exista creado ningún elemento se muestra un listado vacío.	V	V	NA	V	I	NA	El sistema muestra el listado vacío.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Versionar información". Seguidamente se selecciona el sistema y se muestra un listado de atributos, objetos y valores asociados a las informaciones versionadas. Además se muestra una barra de iconos flotantes con las acciones a realizar sobre el contexto: Crear,
		Pregrado	(Vacio)	-	Información	Seleccionar una información que no ha sido versionada.	-		



									Actualizar y Ayuda.
		V	V	NA	NA	NA	NA	El sistema muestra un listado con las informaciones versionadas existentes hasta la fecha.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Versionar información". Seguidamente se selecciona el sistema y se muestra un listado de atributos, objetos y valores asociados a las informaciones versionadas.
EC 1.5 Actualizar.	Se actualiza el listado con las informaciones versionadas existentes hasta la fecha.	Pregrado	Grupos de informaciones versionadas que coincidan con el sistema seleccionado.	-	-	-	-		



									Además se muestra una barra de iconos flotantes con las acciones a realizar sobre el contexto: Crear, Actualizar y Ayuda.
--	--	--	--	--	--	--	--	--	---

Tabla 53. CENIA\_SGU\_C\_DCP\_RF\_V26: Crear\_versionado\_información.

Escenario	Descripción	Información a versionar:	Respuesta del sistema	Flujo central
EC 1.1 Insertar datos correctamente	Este escenario permite seleccionar correctamente la información a versionar	V	El sistema muestra el mensaje: El elemento ha sido creado satisfactoriamente.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Versionar información". A continuación selecciona el sistema del que va a versionar y en el área de iconos flotantes selecciona la opción de Crear. El usuario selecciona la información correctamente y presiona el botón Aceptar.



		Cierre Matrícula		
EC 1.3 Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para versionar una información correctamente.	I (Vacío)	El sistema muestra en rojo el mensaje: "Campo requerido" sobre el campo que debe ser llenado de forma obligatoria.	El usuario una vez autenticado en el sistema selecciona el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Versionar información". A continuación selecciona el sistema del que va a versionar y en el área de iconos flotantes selecciona la opción de Crear. El usuario selecciona la información correctamente y presiona el botón Aceptar.
EC 1.5 Cancelar operación	Mediante este	V		El usuario una vez autenticado en el sistema selecciona





	escenario se cancela la operación de versionar una información.	Cierre Matrícula	El sistema muestra el mensaje de confirmación ¿Está seguro de realizar la acción?	el Módulo "Configuración". El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Versionado" la funcionalidad "Versionar información". A continuación selecciona el sistema del que va a versionar y en el área de iconos flotantes selecciona la opción de Crear. El usuario selecciona la información correctamente y presiona el botón Aceptar.
--	---	------------------	---	--



### Anexo No.13 Resultado de las pruebas funcionales.

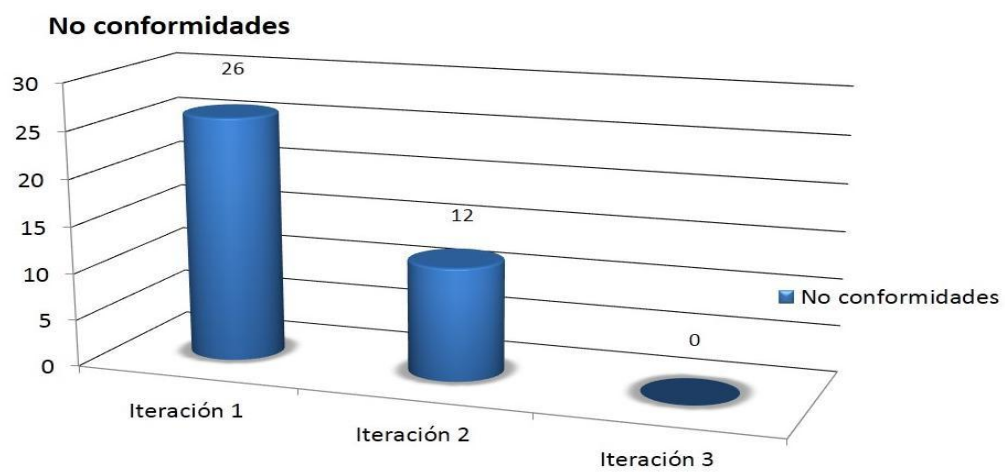


Figura 29. Resultados de las pruebas funcionales.