



**PROTOTIPO SOFTWARE PARA EL SOPORTE DE REGISTRO ACADÉMICO
EN INSTITUCIONES EDUCATIVAS DE NIVEL BÁSICO Y MEDIA VOCACIONAL**

LAURA PULIDO DÍAZ

LUZ DARY MÉNDEZ URIBE

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA
2011**



**PROTOTIPO SOFTWARE PARA EL SOPORTE DE REGISTRO ACADÉMICO
EN INSTITUCIONES EDUCATIVAS DE NIVEL BÁSICO Y MEDIA VOCACIONAL**

LAURA PULIDO DÍAZ

LUZ DARY MÉNDEZ URIBE

**Proyecto de grado presentado como requisito para optar al título de
Ingeniería de Sistemas**

**Director
EMIRO MUÑOZ JERÉZ
Profesor Cátedra
Escuela de Ingeniería de Sistemas e Informática**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA
2011**



DEDICATORIA

Detrás de cada triunfo hay una serie de elementos y seres que trabajan en función del mismo. Precisamente a aquellos seres quiero dedicar esto:

En primer lugar a DIOS, Dios quién dirige y dispone los recursos y talento de toda obra.

A mis padres por ser los patrocinadores y principales responsables de que éste aquí, por confiar en mí y ayudar a que mi propia confianza creciera.

A mi esposo por su paciencia, apoyo y compañía durante éste tiempo

A mis hijos por inspirarme día a día pasión por la vida

A mis hermanos por acompañarme a lo largo de la vida.

Laura



El camino fue impredecible, lleno de aciertos e inconvenientes, todo en función de un objetivo que ahora cumplido presenta motivo de felicidad, espero haber sido la esponjita que se nutrió día a día de conocimiento y fortaleza suficientes para continuar con mi misión en ésta vida y no defraudar a los seres maravillosos a quienes dedico esto:

En primer lugar a DICS, por disponer de mi vida en la mejor forma posible y darme ésta oportunidad.

A mis padres, quienes bajo el patrocinio afectivo, económico, moral y de buenos ejemplos, me motivan a seguir adelante

A mis 10 Super-amigos de la U, por aparecer en mi vida cuando más los he necesitado.

A Chester Andrés mi mascota y amigo.

Luz Dary



AGRADECIMIENTOS

A nuestro director de proyecto, por acogernos en el momento más idóneo sin importar las adversidades; por servir de guía y apoyo a lo largo de éste tiempo, gracias a sus consejos, su paciencia y enseñanzas, esto fue posible.

A la División de Servicios de Información de la Universidad Industrial de Santander, en general por hacernos partícipes en el cambio tecnológico, lo cual nos dejó conocimientos y experiencia suficiente para facilitar el desarrollo del presente proyecto.

A los profesores, por compartir su conocimiento y experiencias tratando de que seamos mejores personas y profesionales.

A Cecilia por su paciencia, dedicación y colaboración durante todos estos años

A nuestros compañeros en general, por su apoyo y amistad a largo de la carrera universitaria.



Tabla de Contenido

Introducción	19
Presentación del Documento	20
Resumen Ejecutivo	21
I. Descripción y Justificación	24
1. Definición del Problema	24
2. Planeación del Proyecto.....	25
3. Justificación.....	26
4. Impacto	27
5. Alcance y Límites	27
6. Entidades Interesadas.....	28
II. Marco Conceptual y Tecnológico.....	29
1 JAVA Enterprise Edition 5 (JEE5)	29
2 Entorno de Desarrollo de la Aplicación	38
3 Herramientas Utilizadas	40
III. Metodología de Desarrollo.....	45
1 Presentación de la Metodología.....	45
2 Descripción de la Metodología	46
3 C.I.D.U. Ciclo Incremental Documentado y Usable.....	51
- A Keep It Short and Simple	51
- B Fases C.I.D.U.....	51
Fase 1: Análisis Previo.....	51
Fase 2: Especulación	55
Fase 3: Diseño y Desarrollo	58
Fase 4: Pruebas	61
Fase 5: Documentación	62
Artefactos C.I.D.U.	63
Algo más sobre C.I.D.U.....	64
IV. Proceso de Desarrollo.....	66



1	Aspectos Generales a Considerar	66
2	Aspectos Generales sobre la sistematización de procesos	68
3	CIDU Matrícula.....	71
4	CIDU Pagos	72
5	CIDU Talento Humano	75
6	CIDU Correo	77
7	CIDU Académico.....	80
V. Prototipo Desarrollado.....		83
Conclusiones y Recomendaciones.....		89
Bibliografía.....		92
ANEXOS.....		95



Listado de Figuras

Ilustración 1. Conexión de Roles CIDU.....	47
Ilustración 2. Fases que componen cada ciclo	65
Ilustración 3. Dendograma (Clustering)	108
Ilustración 4. MDS (Escalamiento multidimensional)	108
Ilustración 5. Estructura Simple de Árbol	111
Ilustración 6. La misma estructura diagramada de forma diferente	111
Ilustración 7. Flecha indica movimiento corriente abajo hacia el fin de la tarea...	111
Ilustración 8. Barra cruzada indica que el movimiento corriente arriba no está permitido	111
Ilustración 9. Flechas múltiples clarifican la dirección.....	111
Ilustración 10. Un punto "continúa hacia" referencia al lector hacia otro diagrama	114
Ilustración 11. Un punto "continúa desde", retomando desde donde salimos de 6a	114
Ilustración 12. Un ejemplo de uso de un área para representar una ventana pop-up	114
Ilustración 13. Un ejemplo de uso de un área iterativa para representar una estructura repetida en un catálogo de productos.....	114
Ilustración 14. Una referencia de flujo sirve tanto como punto "continúa hasta", como punto "continúa desde"	115
Ilustración 15. El área de flujo referida en 9a.....	115



Listado de Tablas

Tabla 01. Niveles y componentes de una aplicación Java EE	29
Tabla 02. Actividades del Propietario del Producto	47
Tabla 03. Actividades del Líder del Proyecto	48
Tabla 04. Actividades del Usuario Representativo	48
Tabla 05. Actividades del Coordinador Experto	48
Tabla 06. Actividades del Equipo de Desarrollo.....	48
Tabla 07. Información básica requerida por caso de uso	55
Tabla 08. Lista de verbos a utilizar para definir casos de uso	55
Tabla 09 Datos básicos por requerimiento	56
Tabla 10. Datos básicos por tipo de usuario.....	56
Tabla 11. Descripción general de artefactos por fase.....	63
Tabla 11. Niveles y componentes de una aplicación Java EE	29
Tabla 12. Roles de Equipo SOFGESCOL.....	66
Tabla 13. Artefactos Análisis Previo Ciclos: “Matrícula” y “Control de Pagos”	70
Tabla 14. Artefactos fase Especulación. Ciclo: “Matrícula”	71
Tabla 15. Artefactos fase Diseño y Desarrollo. Ciclo: “Matrícula”.....	71
Tabla 16. Artefactos fase Pruebas. Ciclo: “Matrícula”	72
Tabla 17. Artefactos fase Documentación. Ciclo: “Matrícula”	72
Tabla 18. Artefactos fase Especulación. Ciclo: “Pagos”	73
Tabla 19. Artefactos fase Diseño y Desarrollo. Ciclo: “Pagos”	73
Tabla 20. Artefactos fase Pruebas. Ciclo: “Pagos”	73
Tabla 21. Artefactos fase Documentación. Ciclo: “Pagos”	74
Tabla 22. Artefactos fase Especulación. Ciclo: “Talento Humano”	75
Tabla 23. Artefactos fase Diseño y Desarrollo. Ciclo: “Talento Humano”	76
Tabla 24. Artefactos fase Pruebas. Ciclo: “Talento Humano”	76
Tabla 25. Artefactos fase Documentación. Ciclo: “Talento Humano”	76
Tabla 26. Artefactos fase Especulación. Ciclo: “Correo”	77
Tabla 27. Artefactos fase Diseño y Desarrollo. Ciclo: “Talento Correo”	78
Tabla 28. Artefactos fase Pruebas. Ciclo: “Correo”	78
Tabla 29. Artefactos fase Documentación. Ciclo: “Correo”	78
Tabla 30. Artefactos fase Especulación. Ciclo: “Académico”	80
Tabla 31. Artefactos fase Diseño y Desarrollo. Ciclo: “Académico”	80
Tabla 32. Artefactos fase Pruebas. Ciclo: “Académico”	81
Tabla 33. Artefactos fase Documentación. Ciclo: “Académico”	81



Listado de Anexos

Anexo A – Metodologías en las que se basa CIDU	95
Anexo B – Principios de Usabilidad	104
Anexo C – Card Sorting	106
Anexo D – Vocabulario Visual.....	109
Anexo E – Formato Acta de Reunión (CIDU)	116
Anexo F – Formato Informe de Reunión (CIDU).....	117
Anexo G – Formato Principios y Técnicas (CIDU)	118
Anexo H – Formato Arquitectura de Información (CIDU).....	119
Anexo I – Formato Especificación de Requisitos (CIDU).....	123
Anexo J – Formato Alternativas de Diseño Gráfico (CIDU)	128
Anexo K – Formato Story Board (CIDU)	131
Anexo M – Principios y Técnicas (SOFGESCOL).....	134
Anexo N – Caracterizaciones (SOFGESCOL)	137
Anexo O – Especificación de Requisitos (SOFGESCOL).....	148



Glosario

ERS: Especificación de Requisitos Software; hace referencia al documento trazable identificado como Anexo I.

PROTOTIPO (SOFTWARE): Programa de evaluación que precede a una herramienta software a implementar en una organización.

TRAZABILIDAD: Atributo de la información que describe su capacidad para ser ubicada o localizada eficientemente.

UML: Lenguaje de modelado unificado; conjunto de diagramas utilizados para el apoyo del desarrollo de sistemas de información. Su objetivo es simplificar las ideas para el desarrollo y hacer entendible las características del sistema tanto para el cliente como para los desarrolladores.

XML: Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

EFICACIA: CERO ERROR durante el desarrollo del proceso

EFFECTIVIDAD: CERO quejas y reclamos por parte de los clientes internos y el cliente final

EFICIENCIA: CERO DESPERDICIO, CERO INTERRUPCIONES, CERO DEMORAS

MEJORA CONTÍNUA: Tomar acciones para mejorar continuamente el desempeño de los procesos y productos/servicio.

EFFECTIVIDAD: La efectividad es la combinación de eficacia (consecución de metas esperadas) y eficiencia (medios y recursos para alcanzar dichas metas).

EFICACIA: La eficacia es el criterio de evaluación con respecto a la consecución de algún objetivo a través del resultado. Si el resultado que se obtuvo es el esperado o concuerda con el objetivo, entonces se es eficaz.



MAPA DE PROCESOS: Es el diagrama que muestra la estructura de la organización con un enfoque basado en procesos, donde se ven cada uno de los procesos y su relación con los demás.

NORMA ISO 9001:2000: Norma publicada por la International Organization for Standardization (ISO) que procura estandarizar la calidad en las organizaciones a través de pautas que deben cumplir los Sistemas de Gestión de Calidad.

PROCESO: De acuerdo a la Norma ISO 9001:2000, un proceso en una organización es una actividad que utiliza recursos y que se gestiona con el fin de permitir que los elementos de entrada se transformen en resultados (ver enfoque basado en procesos).

TRAZABILIDAD: Atributo de la información que describe su capacidad para ser ubicada o localizada eficientemente.

UML: Lenguaje de modelado unificado; conjunto de diagramas utilizados para el apoyo del desarrollo de sistemas de información. Su objetivo es simplificar las ideas para el desarrollo y hacer entendible las características del sistema tanto para el cliente como para los desarrolladores.

XML: Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones

IDE : Entorno de desarrollo integrado.



Resumen

Titulo

PROTOTIPO SOFTWARE PARA EL SOPORTE DE REGISTRO ACADÉMICO EN INSTITUCIONES EDUCATIVAS DE NIVEL BÁSICO Y MEDIA VOCACIONAL¹

Autores

LAURA PULIDO DIAZ

LUZ DARY MENDEZ URIBE²

Palabras Clave

JAVA, JEE5, REPORTE, IREPORT, MATRICULA, REGISTRO ACADÉMICO, COLEGIO

Contenido

Tener una solución informática no solo implica tener una herramienta software que realice ciertas actividades que el usuario haría por si solo en más tiempo y con mayor dificultad, también debe ser fácil de usar e incluir la mayor parte de funcionalidades requeridas, a las cuales se pueda acceder en forma casi intuitiva.

Sin embargo, el desarrollo de soluciones informáticas no siempre contempla aspectos de usabilidad claves para la aceptación y uso de las TIC's. El software genérico puede considerarse una TIC no adaptable a muchas organizaciones y no asequible a las que carecen de recursos como las instituciones públicas de educación básica y media vocacional, en quienes se centra la realización del presente proyecto en base al soporte de actividades propias del proceso "Registro Académico".

Dicho soporte se brindará en base al prototipo "SOFGESCOL", que busca satisfacer en gran medida las expectativas de los usuarios asociados al proceso y permite que se adapten al uso de la herramienta con gran facilidad. SOFGESCOL es un sistema web desarrollado en base al estándar JEE5 y una nueva metodología propuesta llamada "Ciclo Incremental Documentado y Usable (CIDU)" la cual fue diseñada en consideración del potencial de algunas metodologías conocidas tanto de desarrollo software como de diseño, que aporta una serie de documentos y formatos, para facilitar el seguimiento del proceso de desarrollo y control de calidad de la herramienta.

¹ Trabajo de Grado

² Facultad de Ingenierías Físico-mecánicas, Escuela de Sistemas e Informática, Director: Emiro Muñoz Jerez



Abstract

Title

SOFTWARE PROTOTYPE FOR THE SUPPORT OF ACADEMIC RECORDS AND ENROLLMENT IN SCHOOLS HAVING LEVELS BASIC AND VOCATIONAL³

Author

LAURA PULIDO DIAZ
LUZ DARY MENDEZ URIBE⁴

Key Words

JAVA, JEE5, REPORT, IREPORT, ENROLLMENT, ACADEMIC RECORD, SCHOOL

Content

Having a software solution involves not only have a software tool so final users can perform certain activities that would by itself over time and with greater difficulty, should be easy to use and include most of the functionality required, to which can be access in an almost intuitive and interactive environment.

However, the development of software solutions had not always in mind key aspects of usability to get easier and correct the common use of TICs. The software can be considered a generic TIC is not adaptable to many organizations and not accessible to those without resources like public institutions of elementary and secondary vocational education, who focuses on the implementation of this project based on support activities of the process "Academic Record".

Such support will be provided based on the prototype "SOFGESCOL" which largely seeks to meet the expectations of users associated with the process and allows to adapt the use of the tool with ease. SOFGESCOL web is a system developed based on JEE5 standard and a proposed new methodology called "Incremental Cycle Documented and Usable (CIDU)" which was designed in consideration of the potential of some well-known methodologies of software development and design, which provides a number of documents and formats to facilitate monitoring of the development process and quality control tool.

³ Final College Project

⁴ Faculty of Mechanical-Physics Engineering, School of Systems and Information, Director:Emiro Muñoz Jerez



Introducción

Pese al fomento de desarrollo y uso de las TIC, por parte del gobierno, existen organizaciones que no cuentan con soluciones informáticas reales, debido a grandes costos por asumir o al adquirir herramientas software genéricas las cuales no incluyen muchos de sus procesos, o simplemente son difíciles de usar. Por tanto, cabe recordar que parte del éxito en el uso de las TIC's está constituido por la aceptación del usuario y para ello es necesario considerar la influencia positiva de ciertas reglas y métodos de diseño que las metodologías de desarrollo no suelen contemplar al asumir la presencia de diseñadores durante un proyecto.

Sin embargo, es posible que el integrar un profesional orientado al diseño no represente una solución ideal, lo cual nos permite reconocer que el proceso de desarrollo como tal es quién debe asumir ésta tarea. Por ello, el presente proyecto propone una metodología de desarrollo que incorpora los aspectos positivos de metodologías de desarrollo existentes, métodos de diseño (asociados a ergonomía y usabilidad) y mejoras del proceso a fin de obtener herramientas software de gran calidad.

Dicha metodología, habrá de evolucionar durante el proyecto dirigiendo a su vez el desarrollo de una herramienta informática que permita dar soporte adecuado a las actividades del proceso a sistematizar inicialmente seleccionado: el proceso "Registro Académico" en instituciones educativas del sector oficial. Dicho proceso, debe gestionar un gran volumen de datos generando información en forma apropiada para ser utilizada por procesos recíprocos. Esto conlleva a preservar la integridad de dicha información y establecer los controles necesarios para ello, así como realizar seguimiento a cada etapa del proceso a fin de presentar resultados óptimos, entre otros aspectos. Se espera que el prototipo final cumpla tanto con requisitos funcionales asociados al proceso como a condiciones de uso ideales acordes con lo establecido por las normas de usabilidad.



Presentación del Documento

CAPÍTULO I. DESCRIPCIÓN Y JUSTIFICACIÓN. Define aspectos generales relacionados con el proyecto a fin de realizar su respectiva presentación y definir el contexto para su realización.

CAPÍTULO II. MARCO CONCEPTUAL Y TECNOLÓGICO. Presenta el Marco Conceptual Tecnológico describiendo uno a uno los elementos que la componen y define características generales de las herramientas utilizadas durante el desarrollo de SOFGESCOL.

CAPÍTULO III. METODOLOGÍA DE DESARROLLO. Describe las bases y fundamentos teóricos, que han sido aplicados durante el desarrollo del proyecto. Diseño de la metodología de desarrollo utilizada durante el proyecto

CAPÍTULO IV. PROCESO DE DESARROLLO. Descripción general de las actividades desarrolladas durante cada etapa del proceso de desarrollo y presentación de los resultados.

CAPÍTULO V. PROTOTIPO DESARROLLADO. Presentación del prototipo, aplicación de pruebas finales y de usabilidad.

CAPÍTULO VI. CONCLUSIONES. Contiene conclusiones generadas a partir del desarrollo del proyecto y recomendaciones para el desarrollo de futuras aplicaciones.



Resumen Ejecutivo

Buscando facilitar la comprensión del documento se provee esta sección que deberá tomarse como una explicación o revisión superficial al tema completo incluido en el mismo, sin embargo lo incluido en esta sección no se especifica en el resto del documento.

El documento comprende dos puntos, el desarrollo de la aplicación jee5 SOFGESCOL (Software de Gestión para Colegios) con aplicativos para gestión de instituciones educativas y la descripción de la metodología C.I.D.U. (Ciclo Incremental Documentado y Usable) de desarrollo usada para este proceso.

SOFGESCOL:

SOFGESCOL fué desarrollado en Netbeans 6.7.1, Lenguaje Java, Persistencia Toplink, Interfaz con componentes Ice Faces, Base de Datos en Postgres, reportes con IReport 4.

Contiene los módulos:

- Matricula
- Correo
- Pagos
- Talento Humano
- Académico (Moodle)

Cada módulo solo con las funcionalidades básicas conocidas.

Esta diseñado con requerimientos tomados de colegios públicos de nivel básico y media vocacional y generalizado para colegios privados.

La arquitectura es cliente servidor enfocado a redes internas más que a internet, ya que Moodle es una herramienta bastante pesada en concurrencia, por lo que los recursos necesarios para lograr un acceso de velocidad aceptable por internet y en concurrencia son de alto costo para instituciones educativas.

C.I.D.U.:

El desarrollo del software a través de esta, permitió que la definición de la metodología fuera más aplicable dado el paso de experimentación durante el ciclo teórico del mismo.



Primero, cabe aclarar que en su nombre destacamos “Usable” tanto por la inclusión de estándares para la medición de usabilidad como por la facilidad de uso de la metodología misma.

Está determinado por cinco fases:

- Análisis Previo
- Especulación
- Diseño y Desarrollo
- Pruebas
- Documentación

Cada una de estas fases contiene actividades relacionadas con evaluación de usabilidad, determinación de artefactos a usar y desarrollo de la aplicación.

Varias actividades tienen como resultado final un artefacto similar a la lista de requerimientos que busca aclarar los requerimientos y forma de desarrollo.

Estas fases son similares a las de las demás metodologías, para entender la diferencia proponemos la siguiente comparativa en resumen:

Características	Metodologías basadas en Procesos - Tradicional	Metodologías Ágiles (Actividades)	SCRUM – Actividades y Procesos	CIDU - Actividades
Seguimiento Documentado del Uso de Recurso	X		X	-
Documentación base para ser Usada en el desarrollo Exhaustiva	X	X	X	X
documentación de Código Fuente	X	-	X	
Documentación de Código Fuente Limitada				X
Auditoria de Actividades	X			
Confianza en el personal		X	X	X
Definición inicial del plan (trayecto)	X	X		
Definición Ciclica del Plan			X	X



Definición Flexible Modular, Modificable				X
Especificación de Actividades en el Plan		X	X	
Entrevistas para el Plan				X
Entrega Única Final	X			
Entregas consecutivas por ciclos completos		X	X	X

Conclusión del Proyecto:

Para herramientas TIC en empresas donde los usuarios varían bastante en nivel de escolaridad, alfabetización tecnológica, edad, posibilidad de acceso a medios y recursos como computadores e internet, etc..., se debe dar prioridad a la usabilidad durante el proceso de desarrollo, ya que la funcionalidad no se ve reflejada en los procesos empresariales si no existe esta.

Aunque actualmente los contratos se hacen sobre funcionalidades, esto crea un ambiente de hostilidad y desconfianza en el momento en que el aplicativo funciona pero no era lo que el usuario quería o no se siente cómodo en la interacción con el mismo. Se logra mayor satisfacción en los stakeholders cuando la usabilidad esta primero incluso si las funcionalidades aún no están completas.

El desarrollo iterativo en base a usabilidad puede ser entonces el método que lleva a los fundamentos del manifiesto ágil: *“A los individuos y su interacción, por encima de los procesos y las herramientas. El software que funciona, por encima de la documentación exhaustiva. La colaboración con el cliente, por encima de la negociación contractual. La respuesta al cambio, por encima del seguimiento de un plan.”*⁵

⁵ Referirse a: http://es.wikipedia.org/wiki/Manifiesto_%C3%A1gil



I. Descripción y Justificación

1. Definición del Problema

La implementación de herramientas informáticas para el desarrollo de diferentes procesos al interior de las organizaciones, debe tener objetivos más allá de brindar soporte y facilitar su desarrollo, puesto que dichas herramientas deben a su vez atender las necesidades generadas por exigencias y altos estándares de calidad establecidos hoy en día. Sin embargo, muchas de ellas se limitan a realizar tareas básicas, privando al usuario de funcionalidades que aportarían o facilitarían aún más su trabajo.

Este problema suele presentarse en sistemas genéricos, los cuales representan una buena alternativa para casi cualquier tipo de organización, en caso de lograr adaptarse. Pero, son muchas las áreas que quedan fuera del mercado de estos paquetes empresariales y muestra de ello, son las instituciones de educación básica y media vocacional que pertenecen al sector público y las cuales no podrían disponer fácilmente de una herramienta que atienda necesidades particulares. Es probable que debido a sus recursos limitados dichas instituciones en un intento desesperado, logren adaptarse a herramientas genéricas realizando modificaciones pertinentes a muchos de sus procesos tradicionales, pero no todos podrán ser sometidos a dichos cambios como el caso particular que nos ocupa, el proceso de "Registro Académico", proceso administrativo de gran importancia y no considerado en dichas soluciones informáticas.

Por ello se presenta el siguiente prototipo, que busca no solo desarrollar a plenitud cada actividad que conforma el proceso de Registro Académico, sino también busca brindar un aporte significativo a este proceso que permita mejorar su desarrollo actual. Esto, mediante una plena identificación de recursos que pueda proporcionar la herramienta brindando el control necesario sobre el desarrollo e información del proceso, atendiendo a las necesidades de calidad, buenas prácticas y prestando especial atención a características adicionales, cuya consideración promueva el cumplimiento de cada objetivo establecido para dicho proceso.



2. Planeación del Proyecto

- A Misión

La misión del presente proyecto y aquellos que partan del mismo, consiste en identificar necesidades de tipo tecnológico al interior de instituciones educativas, con el fin de desarrollar soluciones tecnológicas que permitan solventar las necesidades identificadas. A su vez pretende presentar la metodología de desarrollo creada como alternativa de solución a problemas presentados durante el proceso de desarrollo de productos software.

- B Visión

Incentivar el uso de soluciones tecnológicas para el fortalecimiento y mejora de los procesos que hacen parte de las instituciones, así como fomentar la creación de nuevos servicios que complementen la herramienta desarrollada. De igual manera promover el uso de la metodología CIDU como alternativa útil al proceso de desarrollo de nuevas herramientas Software.

- C Objetivos

Objetivo General

Desarrollar un prototipo software en entorno Web para sistematizar el proceso de registro académico en instituciones educativas de nivel básico y media vocacional, utilizando el estándar Java Enterprise Edition (JEE5), aplicando normas básicas de ergonomía y usabilidad para software.

Objetivo Específico

Diseñar y aplicar una metodología de diseño y desarrollo centrada en el usuario, haciendo énfasis en el desarrollo rápido de aplicaciones.

Proveer a la institución educativa un sistema para el mantenimiento de registros de matrícula y pre-matricula de estudiantes, facilitando información estadística a la institución en base a los datos recolectados.

Proveer a la institución educativa sistema ágil para la generación de recibos de matrícula, teniendo en cuenta los valores y descuentos determinados por concepto de costos educativos, inscripción y pensión.

Proveer a la institución educativa funcionalidades pertinentes para un seguimiento básico de recurso humano, tomando en cuenta el registro de hojas de vida y proyectos a desarrollar dentro de la institución durante el año lectivo.



Proveer a la institución educativa un sistema de registro de notas, permitiendo elegir la forma de calificación y nomenclatura a usar para este fin.

Proveer a la institución educativa un sistema de generación de informes académicos por periodos académicos y anuales, y otros imprimibles relacionados con los procesos académicos.

Proveer a la institución educativa un sistema de comunicación interna con administración de perfiles para facilitar el seguimiento del conducto regular.

Proveer al sistema desarrollado la arquitectura necesaria para facilitar el acceso, a través de red interna, a perfiles administrativos y docentes y, a través de internet, a padres de familia y estudiantes.

3. Justificación

Los procesos de tipo administrativo que se manejan al interior de cualquier organización, suelen jugar un papel muy importante en la toma de decisiones de la misma, ya que por lo general se encuentran estrechamente ligados a los demás procesos. Lo anterior, sumado a las implicaciones que puede traer un manejo inadecuado, se traduce en una seria intención por parte de las organizaciones para invertir tiempo y recursos a fin de mejorar y fortalecer dichos procesos tanto como sea necesario.

Afortunadamente, desde hace años se viene promoviendo la creación y utilización de nuevas tecnologías, lo cual ha permitido que organizaciones públicas utilicen tecnologías de la Información y Comunicación como herramientas, soportes y canales para el tratamiento y acceso a la información que generan y requieren en gran parte de sus procesos. Pero la intención de continuar y fortalecer la innovación tecnológica, como ingrediente fundamental en el desarrollo socio-económico del país, puede tornarse un fracaso si consideramos la inequidad en cuanto a difusión y utilización de recursos tecnológicos se refiere.

Como reflejo de esta situación se presenta un caso particular, existen instituciones educativas del sector oficial que cuentan con limitaciones tecnológicas en sus procesos administrativos, lo cual dificulta el desarrollo de los mismos. Tanto así, que una vez identificados los errores en cuanto a respaldo y veracidad de la información manejada, surge la necesidad de repetir los procesos para realizar las correcciones pertinentes. Uno de ellos es el de "Registro Académico", encargado de gestionar la mayor parte de la información en éste tipo de instituciones, por ello resulta conveniente que dicho proceso esté bien diseñado, con información acerca de lo que ocurre y controles necesarios para preservar la



información, ya que un mal manejo de la misma puede provocar pérdidas en tiempo y dinero.

Aunque el uso de herramientas tecnológicas para la gestión de dicho proceso resulta ser una solución conveniente, lo cierto es que en la actualidad se tienen muy pocas opciones de éste tipo. Básicamente productos software estándar diseñados pensando en entidades comerciales y por consiguiente no se ajustan a necesidades particulares de éstas organizaciones.

Por ello se propone en el siguiente proyecto el desarrollo de un prototipo software, que permita a las instituciones educativas, mediante el uso de TIC's, la transformación del proceso "Registro Académico" a un proceso administrativo sistemático que garantice el accionar seguro de la información a instituciones educativas de nivel básico y media vocacional. Ofreciendo un software para la correcta gestión de procesos y servicios TIC, a través de la aplicación de marcos de referencia, buenas prácticas y metodologías adecuadas que soporten las tareas de gestión y control necesarias para el proceso a sistematizar.

4. Impacto

Este proyecto será de gran importancia para los autores, debido al uso de nuevas tecnologías y metodologías, los conocimientos adquiridos y la experiencia durante su desarrollo, lo cual influirá positivamente en el desempeño como futuros profesionales.

De otro lado, la tecnología Java 5 en combinación con los demás elementos tecnológicos considerados para el desarrollo de éste proyectos, permitirá ofrecer a estas instituciones, una herramienta que podrá mejorar considerablemente el desarrollo del proceso en cuestión, evitando la presencia de problemas frecuentes, presentando una reducción de tiempo y costos.

Se prevé además, un impacto positivo al interior de la institución, la cual con base en los resultados obtenidos por el uso de la nueva aplicación, optará por promover el uso de éste tipo de herramientas que favorecen el desarrollo de los procesos administrativos.

5. Alcance y Límites

Como resultado del presente proyecto se tendrá un prototipo software para la gestión de Registro académico, basado en las normas y estándares, administrativos y Gerenciales dentro de las organizaciones de los colegios, en consideración de la información requerida.



Lo anterior en cumplimiento de los objetivos establecidos y la especificación de Requisitos en base a cada uno de los subprocesos relacionados en la sección 3.3 del presente documento, incluyendo aquellos requisitos que se desprendan de los formatos, guías y procedimientos establecidos en la documentación del Proceso de Registro Académico, que se marcan actualmente como desarrollados de forma manual y, las ayudas necesarias para asegurar la usabilidad del sistema.

Nota: El sistema resultado de este proyecto, no incluirá características necesarias para cumplir con la sistematización de otros procesos.

- A Características del prototipo

- Prototipo orientado a la Web
- Concede permisos para realizar actividades, (según el perfil actual de usuario), los cuales son configurables.
- Los formularios asociados al subproceso de matrícula están basados en la información recaudada en el Sistema Nacional de Información de Matriculas SIMAT y de acuerdo a los formatos existentes para dicho procedimiento.
- Generación de informes requeridos.
- Acceso a información de acuerdo a los perfiles de usuario.
- Permite la comunicación entre usuarios a fin de manejar el conducto regular.

- B Que no hace el prototipo

- No es una herramienta para realizar pago de matrículas en línea os específicos
- Si bien permite la generación de informes, no está automatizado el envío de estos a personas u organismos interesados en su contenido.

6. Entidades Interesadas

Universidad Industrial de Santander

La Universidad Industrial de Santander, como entidad dedicada al fortalecimiento de la región, entregando profesionales productivos y emprendedores con alto nivel educativo, encuentra importante el fomento de las herramientas tecnológicas en niveles de educación básica y media vocacional en busca de fortalecer los pilares de sus futuros profesionales.



II. Marco Conceptual y Tecnológico.

1 *JAVA Enterprise Edition 5 (JEE5)*

La plataforma Java™, Enterprise Edition (Java EE), evoluciona paulatinamente y en su versión JEE5 presta especial atención en ofrecer a los desarrolladores un potente conjunto de APIs que permiten reducir el tiempo de desarrollo, reducir la complejidad y mejorar el rendimiento de las aplicaciones y esto gracias a:

Un modelo de programación simplificado.

Descriptores de despliegue XML opcionales

Modelo de Aplicación Java EE

La aplicación Java EE modelo define una arquitectura para la ejecución de los servicios multi-hilo en aplicaciones que ofrecen escalabilidad, accesibilidad y capacidad de gestión. Este modelo presenta dos partes necesarias para que los servicios multi-hilo funcionen: la lógica de negocio y la presentación.

Aplicaciones distribuidas de varios niveles

Esta plataforma utiliza un modelo de aplicaciones distribuidas multicapa para aplicaciones empresariales, donde cada aplicación se divide en componentes de acuerdo a su función y luego son instalados en equipos diferentes según el nivel al que correspondan según la escala en el entorno Java EE.

Tabla 01. Niveles y componentes de una aplicación Java EE

Niveles o capas	Tipo de Componentes	Componentes	Equipos
<i>Presentación</i>	Componentes de cliente		Máquina Cliente
<i>Aplicación</i>	Componentes Web	Servlets JavaServer Faces	Servidor Java EE
<i>Servicios y Dominio</i>	Componentes empresariales Componentes de tecnología	Java Servlet, JavaServer Faces y JavaServer Pages™ (JSPTM)	Servidor de aplicaciones multiproceso
<i>Persistencia</i>	Sistema de información	Empresa JavaBeans™ (EJB™)	Servidor EIS.



	empresarial (EIS) Componentes de negocio	componentes (beans de empresa)	
--	---	--------------------------------	--

Fuente: Autor

Nota: Las aplicaciones no siempre consideran todos los niveles.

Componentes Web

Componentes Web Java EE son los servlets o páginas creadas con tecnología JSP (páginas JSP) y / o la tecnología JavaServer Faces. Servlets son clases de lenguaje de programación Java dinámicamente que tramitar las solicitudes y construir respuestas. JavaServer Faces tecnología se basa en la tecnología de servlets y JSP y proporciona un marco de componentes de interfaz de usuario para aplicaciones web.

Componentes de Negocio

Se encargan de la lógica del negocio a fin de resolver o satisfacer las necesidades de la empresa, para ello se reciben datos de aplicaciones cliente, se procesan y de ser necesario se almacenan. De igual manera puede recuperar datos y devolverlos al cliente.

Sistema de Información a nivel empresarial (EIS)

La empresa puede tener un sistema de información a nivel EIS incluyendo paquetes software alternativos y gracias al cual se facilitará la conectividad entre aplicaciones y bases de datos.

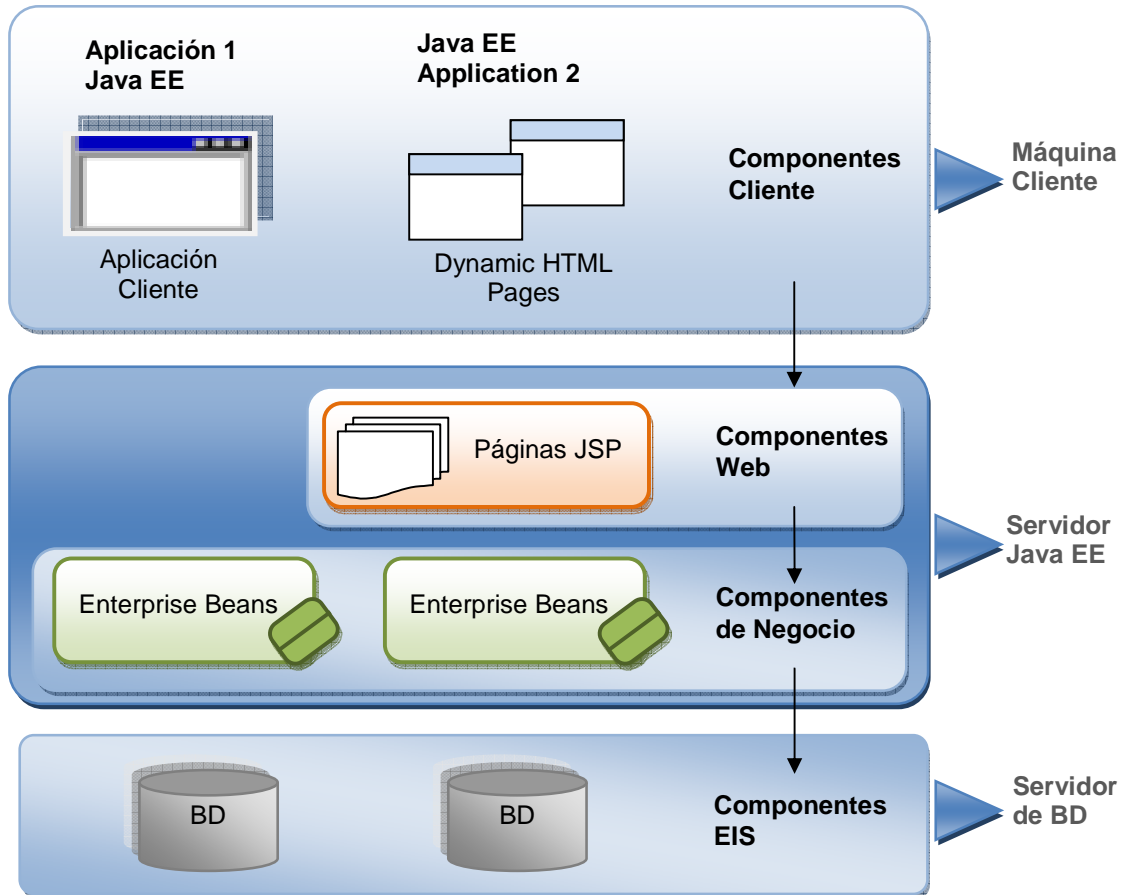
Dichos componentes están escritos en lenguaje de programación Java y se han compilado en la misma forma que cualquier programa en el idioma. La diferencia radica en que los componentes estándar: son ensamblados en una aplicación Java EE, se verifica que estén bien formados y en cumplimiento de la especificación Java EE, y se despliegan a la producción, donde se ejecutan y gestionado por el servidor de Java EE.

Seguridad

A nivel declarativo la plataforma proporciona reglas de control de acceso que son definidas por el desarrollador y son interpretadas durante el despliegue. A su vez proporciona mecanismos estándar de acceso sin necesidad de incluirlos en la aplicación, pues ella funcionará en una gran variedad de entornos de seguridad.



ILUSTRACIÓN 3. NIVELES Y COMPONENTES DE UNA APLICACIÓN JAVA EE



Fuente: Autor

EJB 3.0 Enterprise JavaBeans

La arquitectura EJB es una arquitectura para el desarrollo y despliegue de aplicaciones de negocios basadas en componentes. Las aplicaciones que utilizan la arquitectura Enterprise JavaBeans se caracterizan por ser escalables, transaccionales, multi-usuario y seguras.

Un Enterprise JavaBean agrupa funcionalidades para una aplicación, sin embargo, a diferencia de un JavaBean un EJB es un deployable component, término que implica que existe un ambiente de ejecución, éste ambiente es precisamente un EJB Container, que es parte de un java application server.

El objetivo de Enterprise JavaBeans 3.0 es simplificar el desarrollo de aplicaciones Java y estandarizar el API de persistencia para la plataforma Java. El propósito de EJB 3.0 es mejorar la arquitectura EJB, al reducir su complejidad desde el punto de vista de los desarrolladores.



Características de EJB 3.0

- EJB 3.0 elimina la necesidad de las interfaces EJB y descriptores de despliegue innecesarios y el requisito de que las clases bean implementen la interfaz `javax.ejb.EnterpriseBean`.
- Simplifica los EJBs para asemejarse a los POJOs o JavaBeans.
- Simplifica la configuración a través de valores por defecto y del uso de anotaciones de metadata en lugar de descriptores de despliegue.
- Se soporta la inyección de dependencias para utilizar EJBs, recursos y variables de entorno.
- Los beans de entidad no requieren ninguna interfaz de componente.
- Los beans de entidad utilizarán ahora una interfaz POJO.
- Elimina la necesidad de implementar métodos callback de ciclos de vida innecesarios.
- Búsqueda e invocación de métodos simplificada.
- Requiere menos clases e interfaces
- Virtualmente elimina los descriptores de despliegue a través de largas anotaciones
- Direcciones y especificaciones más típicos a través de la anotación por defecto
- Proporciona más limpio, más fácil, y estandarizado el mapeo objeto-relacional
- Elimina la necesidad de búsqueda de código
- Agrega soporte para la herencia, polimorfismo y consultas polimórficas.
- Agrega soporte para nombrar consultas estática y dinámicamente
- Proporciona un lenguaje Java Persistence de consulta – adicional a EJB QL
- Facilita la prueba entidades fuera del contenedor EJB



- Puede ser utilizado fuera del contenedor

Tipos de EJB

EJBs de Entidad (*Entity EJBs*)

Están encargados de modelar los datos de negocio y por consiguiente su principal objetivo es encapsular los objetos del lado del servidor que almacena los datos. Los EJBs de entidad presentan la característica fundamental de la persistencia:

Persistencia gestionada por el contenedor (CMP): el contenedor se encarga de almacenar y recuperar los datos del objeto de entidad mediante el mapeo de una tabla de la **base de datos**.

Persistencia gestionada por el bean (BMP): el propio objeto entidad se encarga, mediante una **base de datos** u otro mecanismo, de almacenar y recuperar los datos a los que se refiere, por lo cual, la responsabilidad de implementar los mecanismos de persistencia es del programador.

EJBs de Sesión (*Session EJBs*)

Están encargados de gestionar el flujo de la información en el servidor. Generalmente sirven a los clientes como una fachada de los servicios proporcionados por otros componentes disponibles en el servidor. Puede haber dos tipos:

con estado (*stateful*). Los beans de sesión con estado son objetos distribuidos que poseen un estado. El estado no es persistente, pero el acceso al bean se limita a un solo cliente. En otras palabras, dentro de la sesión del usuario estos beans se van a encargar de almacenar datos en variables de instancia y esos datos van a tener un significado concreto durante toda la conversación mantenida entre el cliente y el bean.

Sin estado (*stateless*). Los beans de sesión sin estado son objetos distribuidos que carecen de estado asociado permitiendo por tanto que se los acceda concurrentemente. Además, no dispone de variables de instancia en las cuales se guarden datos que puedan ser compartidos entre los distintos métodos del bean; por lo general, contarán con ciertos métodos en función de un trabajo particular e independiente y los resultados obtenidos a partir de las operaciones realizadas al interior del bean no dependerán de ningún estado relacionado con la conversación que el usuario actual del sistema mantiene con éste tipo de bean.



Anotaciones más utilizadas

@ *stateful*: al principio de la clase

@ *stateless*: al principio de la clase

@*Local*: Si el bean va a ser utilizado por una clase que se ejecute dentro de la misma JVM
en la que se ejecuta el bean de sesión entonces la interface podría ser Local

@*Remote*: si el bean va a ser llamado por una clase que se ejecuta en una JVM distinta entonces la interface tendría que ser Remote.

EJBs dirigidos por mensajes (*Message-driven EJBs*)

Son similares a los bean de sesión, pero permiten el procesamiento asíncrono de mensajes usando el *Java Messaging System (JMS)*. Este servicio funciona a través de colas de mensajes, que es donde los clientes envían sus peticiones, y estas colas son controladas por los Message Driven Beans, los cuales procesan los mensajes que hay en ellas y ejecutan ciertos servicios dependiendo del mensaje procesado. Este tipo de beans se aproximan más a la forma conceptual de los beans de sesión sin estado en el sentido que son beans que no deben almacenar estado alguno.

Anotaciones más utilizadas

Para declarar un MDB (Message Driven Bean) sólo se debe establecer la anotación @MessageDriven a una clase que implemente la interface MessageListener, e indicar el nombre de la cola del contenedor (la cuál es accesible vía JNDI) que va a controlar el MDB.

JPA (API de persistencia de Java)

El Java Persistence API (JPA) ofrece una alternativa diferente para presentar los datos asociados a una base de datos relacional, mediante la asignación a objetos de Java ("entidades persistentes"). Estos objetos son almacenados de tal forma que se puede acceder a ellos en un momento posterior, y la entidad conservará un estado persistente, incluso después de terminar la aplicación que la utiliza, pero estos objetos no tienen nada en común con los Entity Beans 2.x, pues las entidades JPA son clases POJOs, no extienden de ninguna clase y no implementan ninguna Interface. Además de simplificar el modelo de persistencia de la entidad, el API Java Persistence estandariza el mapeo objeto-relacional y ayuda a simplificar el modelo de persistencia de la entidad.



Para que una clase sea considerada persistente, es necesario seguir unas reglas JPA:

La clase tiene que ser identificada como una entidad usando la anotación @Entity

Una propiedad de la clase debe tener un identificador anotado con @Id

Debe existir un constructor sin argumentos

A fin de manipular el objeto persistente creado y realizar las operaciones básicas que corresponden al mantenimiento de una tabla en una base de datos, tales como crear, consultar, actualizar y borrar (métodos CRUD), serán realizadas a través de la interface javax.persistence.EntityManager de JPA.

El EntityManager puede ser visto como una clase DAO que nos provee de un set de métodos clásicos (persist, remove) y buscadores (find).

La interface EntityManager no tiene un método update. Los updates se hacen a través de las propiedades "setters".

Luego borraré el objeto usando EntityManager.remove(), notar que éste código usa transacciones explícitas. Es por eso que los métodos persist, update, y remove son llamados entre transacción.begin() y transacción.commit().

Java Persistence API consta de tres áreas:

El Java Persistence API

El lenguaje de query

El mapeo de los metadatos objeto/relacional

Anotaciones más utilizadas

Relaciones múltiples de la entidad

Uno a uno: Cada entidad se relaciona con una sola instancia de otra entidad. En éste caso se utilizan las anotaciones @ OneToOne (javax.persistence.OneToOne) y la relación debe establecerse en la misma forma para las dos entidades referenciadas.

Uno a muchos: Una entidad, puede estar relacionada con varias instancias de otras entidades. Éste tipo de relación, se utiliza anotaciones @OneToMany (javax.persistence.OneToMany) en los campos o propiedades persistentes. En su



lugar las otras entidades presentarán una relación del tipo Muchos a uno con respecto a ésta entidad.

Muchos a uno: Múltiples instancias de una entidad pueden estar relacionadas con una sola instancia de otra entidad. Esta multiplicidad es lo contrario a la relación uno a muchos. Para éste caso se utilizan anotaciones `@ManyToOne` (`javax.persistence.ManyToOne`) en los campos o propiedades persistentes. En contraparte, la entidad referenciada debe establecer una relación uno a muchos con respecto a ésta entidad.

Muchos a muchos: En este caso varias instancias de una entidad pueden relacionarse con múltiples instancias de otras entidades. Este tipo de relación utiliza anotaciones `javax.persistence.ManyToMany` en los campos o propiedades persistentes y la relación debe establecerse en la misma forma para las dos entidades referenciadas.

Relaciones y borrado en cascada

Existen entidades que utilizan relaciones con dependencias de relaciones de otra entidad.. Esto se llama borrado en cascada. Las relaciones de borrado en cascada se especifican utilizando `cascade=REMOVE`, elemento que viene en la especificación de las relaciones `@OneToOne` y `@OneToMany`.

Entity Manager

Las entidades son gestionadas por el Entity Manager. Éste se representa por instancias `@EntityManager` (`javax.persistence.EntityManager`). A cada instancia de `EntityManager` se le asocia con un contexto de persistencia. Un contexto de persistencia define el ámbito particular, bajo el cual se crean o se eliminan las instancias de la entidad.

El contexto de persistencia: Es un conjunto de instancias que existen en un almacén de datos. El interfaz `EntityManager` define los métodos que se utilizan para interactuar con el contexto de persistencia.

Interfaz Entity Manager

La API `entityManager` crea y elimina instancias de entidades persistentes, busca entidades a partir de su clave primaria y permite ejecutar queries.

Contenedor de Manager Entity

Con un contenedor de entidades, una instancia de un `EntityManager` de un contexto persistente se propaga automáticamente por el contenedor de aplicación a todos los componentes que utilizan la instancia del `EntityManager` en una sola transacción Java (JTA).



JSF (Java Server Faces)

JavaServer Faces (JSF) es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL.

JSF incluye:

Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.

Un conjunto por defecto de componentes para la interfaz de usuario.

Dos bibliotecas de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.

Un modelo de eventos en el lado del servidor.

Administración de estados.

Beans administrados.

ICE Faces

ICEfaces es un framework Ajax de código abierto que permite a los desarrolladores de aplicaciones Java EE crear y desplegar aplicaciones Web en servidor rich Internet application (RIA), utilizando el lenguaje Java. ICEfaces aprovecha todas las herramientas y entornos de ejecución basados en estándares Java EE.

Modelo de aplicaciones de Desarrollo

La arquitectura ICEfaces ofrece un modelo de aplicación centrado en el servidor, donde se desarrolla toda la lógica de aplicaciones en Java puro, y se ejecuta en un entorno estándar de aplicaciones Java de tiempo de ejecución del servidor. Esto le permite al desarrollador librarse de las complejidades del desarrollo Ajax de bajo nivel en JavaScript, ya que utiliza el Ajax ICEfaces habilitando una suite que contiene todos los componentes estándar de JSF.

. A su vez implica que la infraestructura existente de Java EE, entornos de desarrollo, y las mejores prácticas siguen siendo pertinentes y útiles.



Hay tres elementos básicos de la arquitectura ICEfaces

1. ICEfaces Framework
El marco es una extensión de la norma Framework JSF, con la diferencia clave en ICEfaces relativas a la fase de representación. El Marco ICEfaces proporciona una completa gestión en tiempo de ejecución de Ajax Push con el servidor iniciado por las API de representación, y se integra a la perfección ese mecanismo con el ciclo de vida de JSF.
2. Ajax Puento
El puente dispone de un servidor Ajax-resident y los elementos residentes en el cliente que coordinan la comunicación basada en Ajax entre el navegador cliente y el servidor de aplicaciones residentes. El puente es el responsable de la entrega de los cambios incrementales de presentación de la fase de representar en el cliente de explorador y volver a montar los cambios en el DOM del navegador para afectar cambios en la presentación. El puente es también responsable de la detección de la interacción del usuario con la presentación y realización de eventos de usuario a la solicitud de procesamiento a través de la norma del ciclo de vida de JSF. El puente de Ajax se establece automáticamente en la carga la primera página de la solicitud y coordina actualizaciones de presentación y transmisión de eventos de usuario para toda la vida útil de la aplicación.
3. *ICEfaces Suite de componentes*
La suite de componentes proporciona todos los bloques de construcción para la aplicación de interfaz de usuario. Incluye tanto los componentes estándar de JSF, y una amplia gama de componentes de avanzada que permiten a los desarrolladores ensamblar interfaces sofisticadas aplicaciones de manera eficiente.
<http://en.wikipedia.org/wiki/ICEfaces>
http://es.wikipedia.org/wiki/JavaServer_Faces

2 **Entorno de Desarrollo de la Aplicación**

Plataforma Netbeans

Es a grandes rasgos una base modular y extensible que puede ser utilizada como estructura de integración para la creación de grandes aplicaciones de escritorio. La plataforma provee servicios convencionales asociados al desarrollo de aplicaciones, para facilitar las tareas del programador. El fortalecimiento y evolución de la misma, son resultado del trabajo de empresas independientes asociadas (especializadas en desarrollo software) que proporcionan extensiones que pueden ser integradas fácilmente en la plataforma, a beneficio de muchos.



Características Generales

Estructura de la interfaz de usuario: Las ventanas, menús, barra de herramientas, y otros componentes están disponibles por la plataforma. Como resultado, ayuda al usuario a centrarse en acciones específicas, las cuales condensan el código, haciéndolo más eficiente y menos propenso a errores. La interfaz de usuario completa ofrecida por la Plataforma de Netbeans está basada al 100% en AWT/Swing y puede ser extendida con tus propios componentes.

Editor de datos: El poderoso editor de Netbeans, dentro del IDE de Netbeans, puede ser usado por las aplicaciones del usuario. Las herramientas y funcionalidad del editor pueden ser extendidas y adaptadas a los propósitos de la aplicación en forma rápida y sencilla.

Visualización de la utilización: La Plataforma de Netbeans cuenta con una estructura disponible, que permite al usuario integrar sus propias opciones de diálogo en forma sencilla, permitiéndole guardar y restablecer las características en una presentación amigable.

Framework inteligente: Ofrece herramientas simples para crear asistentes extensibles y amigables al usuario, guiándolo a través de pasos complejos en la aplicación.

Sistemas de datos: En términos de la plataforma de Netbeans, los datos pueden ser locales o disponibles mediante FTP, CVS, una base de datos o un archivo XML. Por medio de la abstracción, el acceso a los datos por un módulo es transparente a los demás módulos. El acceso al dato actual no es por lo tanto de preocupación, desde que se trabaja con las API's de la plataforma de Netbeans.

Internacionalización: Provee clases y métodos que permiten la internacionalización de la Ayuda de Java y otros recursos. Se pueden almacenar fácilmente constantes textuales en las propiedades de archivos. También recupera los textos e iconos aplicables al país en común y la sus características del lenguaje.

Ayuda del sistema: Ofrece un sistema central para la integración y visualización de tópicos de ayuda al usuario final. Los módulos individuales pueden contribuir con sus propios tópicos al sistema de ayuda de la aplicación y también provee un contexto sensitivo de ayuda.

http://www.sun.com/emrkt/innercircle/newsletter/latam/0207latam_feature.html



3 Herramientas Utilizadas

Enterprise Architec 7.1 (Vs. Trial)

Enterprise Architect (EA) es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad.

Características

Velocidad, estabilidad y buen rendimiento

El Lenguaje Unificado de Modelado provee beneficios significativos para ayudar a construir modelos de sistemas de software rigurosos y donde es posible mantener la trazabilidad de manera consistente. Enterprise Architect soporta este proceso en un ambiente fácil de usar, rápido y flexible.

Trazabilidad de extremo a extremo

Enterprise Architect provee trazabilidad completa desde el análisis de requerimientos hasta los artefactos de análisis y diseño, a través de la implementación y el despliegue.

Combinados con la ubicación de recursos y tareas incorporados, los equipos de Administradores de Proyectos y Calidad están equipados con la información que ellos necesitan para ayudarles a entregar proyectos en tiempo.

Construido sobre las bases de UML 2.1

Las bases de Enterprise Architect están construidas sobre la especificación de UML 2.0 - pero no se detiene ahí! Usa Perfiles UML para extender el dominio de modelado, mientras que la Validación del Modelo asegura integridad.

<http://www.sparxsystems.com.ar/products/ea.html>

http://www.sparxsystems.com.ar/products/ea_features.html

Potgresql

Es una herramienta que permite gestionar bases de datos relacionales orientadas a objetos. Su versión 8.4, (la cual fue seleccionada) permite el análisis de datos mucho más sencillo a través de funcionalidades avanzadas de ANSI SQL:2003, como las funciones window, expresiones comunes de tabla y joins recursivos. Las mejoras en los procedimientos almacenados, como los valores por omisión para



los argumentos y los argumentos de largo variable hacen más simple y compacta la programación en la base de datos.

Ésta versión fue seleccionada pues aumenta el rendimiento de las aplicaciones, puesto que se han *añadido muchas optimizaciones, como semi-joins y anti-joins, que otorgan una notable mejora en el tiempo de ejecución en algunas de nuestras consultas más exigentes.* Debido a estas nuevas características PostgreSQL 8.4 podrá atender a muchos más usuarios que antes.

Características

Buen Rendimiento: PostgreSQL entrega tiempos de respuesta consistentes sin importar la carga del servidor. Varias características nuevas mejoran significativamente la velocidad de operaciones específicas como: Tiempos de recuperación mejorados, Búfer circular de «tuplestore», Comparaciones, Ordenamiento Top-N, Asignación de XID «perezosa», entre otras.

Bases de datos grandes: Los usuarios podrán construir almacenes de datos aún más grandes con PostgreSQL, en base a las siguientes mejoras: Recorridos sincronizados, Protección de Cache L2 y Reducción de tamaño de encabezados Varlena («Var-Varlena»)

Administración: PostgreSQL presenta ventajas representativas en comparación a otras bases de datos comerciales respecto a la fácil administración y algunas adicionales incluidas en ésta versión que la hacen más fácil de administrar y proporcionan mayor información al DBA son: Salida de log CSV, Soporte de SSPI y GSSAPI, Configuración GUC por función, Múltiples esclavos Autovacuum, pg_standby, ORDER BY Nulls First/Last.

Facilita el desarrollo de aplicaciones: Varias mejoras ayudan a que PostgreSQL esté al nivel de las mejores bases de datos comerciales, permitiendo el desarrollo de aplicaciones de bases de datos complejas y de múltiples capas, incluyendo: Búsqueda exhaustiva en texto, Invalidación de planes, Cursores actualizables
Nuevos tipos de datos: Soporte XML, UUID, Arreglos de tipos compuestos, ENUM
Procedimientos almacenados: Dos características nuevas expanden la utilidad de PL/pgSQL, el lenguaje de procedimientos almacenados más popular de PostgreSQL: Soporte a «funciones tablas» y Cursores desplazables (scrollable)

Documentación: La documentación en HTML y las páginas *man* son instaladas junto con PostgreSQL; además, puede navegar, buscar y comentar en nuestra extensa documentación en línea interactiva.

Licencia: PostgreSQL usa la licencia BSD, la cual sólo requiere que el código fuente licenciado mantenga la información de derechos de autor y licenciamiento.



Características particulares incluidas en ésta versión

Restauración de base de datos en procesos paralelos, que acelera la recuperación de un respaldo hasta 8 veces.

Privilegios por Columna, que permiten un control más granular de datos confidenciales.

Configuración de ordenamiento configurable por base de datos, lo cual hace a PostgreSQL más útil en entornos con múltiples idiomas.

Actualizaciones “en el lugar” desde 8.3 a 8.4 con muy bajo downtime, gracias al uso de pg_migrator beta.

Nuevas Herramientas para Monitoreo de Consultas que le otorgan a los administradores mayor información sobre la actividad del sistema.

PgAdminIII

Es un software para el mantenimiento e integración de bases de datos, el cual fue seleccionado a su vez, por facilitar la edición de scripts de la base de datos (Script de base de datos es el código elemental de la base de datos sobre el cual se crean las tablas, columnas y campos especiales). Ésta última ventaja permite que cuando la base de datos sea corrupta (Tengamos que modificar campos específicos, crear tablas o eliminar tablas, cambiar constrain, etc.) se pueda eliminar la base de datos y generar nuevamente la base de datos con los datos nuevos, o llevar la base de datos de un computador a otro sin ningún inconveniente. Esta herramienta facilita también generación y restauración de la base de datos.

Nota: Los inconvenientes de manejar esta herramienta es que al aumentar el número de tablas es difícil encontrar las relaciones de la base de datos es por esto que se utilizo el software EMS.

EMS

Ésta herramienta facilita el diseño, modelado y creación de bases de datos, de igual manera permite la integración de cada nuevo paquete (bases de datos de cada aplicación en caso de requerirlo). Fue seleccionada para uso durante el proyecto gracias a una de sus principales funcionalidades, permitir el modelado de



las bases de datos y generar en base a ello el código requerido en postgresQL para su correspondiente creación.

A su vez, facilita el manejo de las bases de datos en base a los gráficos generados que representan cada una de las tablas y ayuda a generar la documentación en base a un conjunto de archivos HTML que incluyen información básica requerida.

Características

- Soporte completo de SQL Server 2005
- Soporte de SQL Server 2008
- Soporte de datos Unicode
- La rápida gestión de bases de datos y navegación
- Fácil gestión de todos los objetos de SQL Server
- Herramientas avanzadas de manipulación de datos
- Una gestión eficaz de la seguridad
- Excelentes herramientas visuales y de texto para elaboración de consultas
- Impresionantes los datos de exportación e importación de
- diseñador visual de base de datos para manejar la estructura de base de datos en unos pocos clics
- Fácil de usar asistentes de realizar tareas de servidor SQL administrativa
- Potentes herramientas para hacer su trabajo con SQL Server es tan fácil como puede ser
- Informe de diseño con claros en el asistente de la construcción el uso
- Nueva interfaz gráfica de usuario del estado de la técnica Netbeans IDE (Entorno de Desarrollo Integrado) 6.7.1

Cam Studio

Esta herramienta permite grabar fácilmente aquello que sucede en una zona específica del escritorio la cual puede ser seleccionada y delimitada, incluso puedes grabar audio desde un micrófono. Al terminar la grabación se genera un archivo en formato AVI o SWF, el cual puede integrarse a una aplicación. Aunque existen opciones alternativas, ésta herramienta fue seleccionada puesto que los videos generados son de tamaño reducido y con soporte para Streaming de vídeo sobre flash. A su vez, durante el proceso de grabado se requieren menos recursos de equipo de cómputo.

Nota: Esta herramienta fue utilizada en el proceso de creación de los respectivos videos de pruebas y manuales de usuario.



En éste proyecto se aplica el concepto de Persistencia gestionada por el contenedor. Por consiguiente, un Entity Bean será una clase (POJO: Plain Old Java Objects) que representa una tabla de una base de datos, y cada instancia de ésta clase representa un registro de la tabla. Para esto, se llevo a cabo el correspondiente “mapeo” entre las propiedades de una clase y los campos que pertenecen a la tabla. A su vez, es necesario especificar las relaciones que tienen las clases entre sí (uno a uno, uno a muchos, muchos a uno y muchos a muchos), e identificar una clave primaria que identifica a ese registro de forma única dentro de la tabla una clave primaria que identifica a ese registro de forma única dentro de la tabla. Las anteriores configuraciones, deben realizarse a través de anotaciones y el API que se encarga de gestionar todos los aspectos relativos a la persistencia es JPA (Java Persistent API).



III. Metodología de Desarrollo.

1 *Presentación de la Metodología*

Definición del Problema

En base a una investigación general sobre las últimas tendencias en el desarrollo de productos software, se identificaron algunas pautas generales a seguir reflejadas en:

Mejora en tiempos de desarrollo, mejora en la satisfacción del cliente y una seria intención por estandarizar el proceso de desarrollo en el punto máximo de calidad.

Por consiguiente, se hace necesaria una búsqueda de elementos que ayuden a garantizar la presencia de estos tres ingredientes en el desarrollo del proyecto. Soluciones generales a cada caso se presentaron por separado, por ello puede decirse que en la actualidad no se cuenta con una metodología que favorezca en conjunto las características propuestas para el desarrollo del presente proyecto lo cual deja como alternativa el diseño de una nueva metodología que combine en forma conveniente una propuesta ágil de desarrollo, que involucre métodos de usabilidad a fin de promover productos afines con las expectativas y necesidades del cliente y que además permita generar la documentación fuente necesaria para futuros proyectos.

¿Porqué diseñar una nueva metodología?

Si se realiza un bagaje histórico considerando las metodologías de desarrollo que han aparecido en las últimas décadas, se denota un mejoramiento progresivo de las mismas en comparación con métodos previos. La idea es permitir una evolución de nuevos modelos contemplando aspectos en los cuales hayan fallado sus antecesores y esto permite incluso combinar antiguas prácticas para lograr mejores resultados de acuerdo a las falencias que se quieren cubrir.

Por ello se puede decir que la informática como disciplina puede evolucionar en forma independiente dentro del ámbito de otras ingenierías y de hecho, esto simula una necesidad que gracias al tiempo y a la comunidad informática se nutre mediante cambios importantes que se han dado al proceso de desarrollo de productos software.

Sin embargo el objetivo que prima no ha sido logrado del todo puesto que se quiere pasar del conjunto de buenas prácticas a un conjunto de patrones, de



estándares que se puedan aplicar dada la situación, lo cual se ha presentado en otras ingenierías. Visto el proceso de desarrollo como tal, con características afines a cualquier otro proceso y necesidades latentes en espera de una solución que no puede ser aportada por metodologías existentes, se tiene una premisa de calidad que representa nuestra causa “Mejora Continua de procesos”, lo cual motiva el desarrollo de una nueva metodología.

2 Descripción de la Metodología

Presentación General de CIDU

Para iniciar el desarrollo de ésta nueva metodología, fue necesario recopilar la información necesaria respecto a diferentes técnicas y métodos que servirán de base a la nueva metodología, dicha información compone la síntesis que corresponde al Anexo A del presente documento. Éste marco teórico, permite un posterior refinamiento de estas técnicas y mediante un diseño apropiado que incorpora las mejores y más avanzadas prácticas en el proceso de desarrollo que será caracterizado nace “CIDU” sigla que traduce: CICLO INCREMENTAL DOCUMENTADO Y USABLE, nombre que corresponde a las principales características que comporte cada uno de los componentes o módulos que se integran en una aplicación a desarrollar.

Por ser una metodología ágil para el desarrollo de aplicaciones software, se trabaja en función de los artefactos necesarios para llevar a cabo un apropiado proceso de desarrollo sin hacer hincapié en extensa documentación. La idea es disminuir recursos a utilizar y para ello se presentan documentos anexos de los formatos propuestos para ser diligenciados por el equipo de desarrollo, a fin de estandarizar los entregables generados, esto para facilitar la adaptación a CIDU. A su vez se presentan documentos de ayuda para el desarrollo de las diferentes actividades, estos documentos recopilan y sintetizan la información requerida.

- A Aspectos Generales

Número de Participantes en el equipo

La metodología propuesta ha sido diseñada considerando que el número de participantes en un equipo de desarrollo varía en forma considerable, por lo cual haremos referencia a un mínimo de participantes del equipo a fin de completar los perfiles definidos, por consiguiente hablaríamos de 5 personas. Si se trata de menos integrantes en el equipo, cada persona se encargara de tareas según los roles que le sean asignados. En caso de contar con un mayor números personas



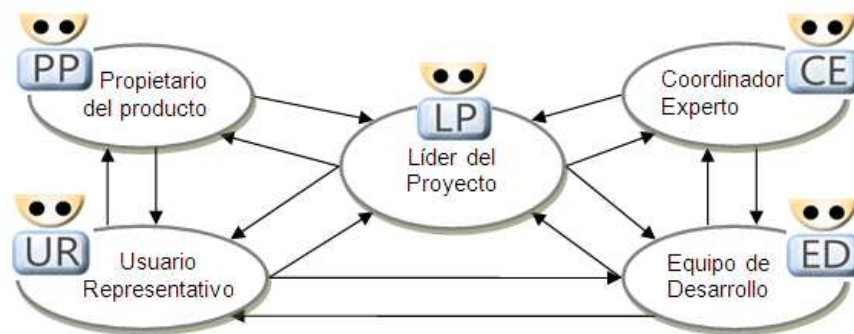
se hará una distribución en subgrupos de trabajo de acuerdo al tamaño que ha sido previamente indicado y los cuales trabajarán en forma paralela.

- B Identificar Roles

La idea en realidad es encontrar en cada miembro del equipo de trabajo, cualidades que puedan asociarlo claramente dentro de un rol específico, a su vez se definirán las tareas a realizar en cada caso, pues no siempre se contará con el número de personas requerido para asumir todos los roles.

Un rol define las cualidades que debe poseer la persona que lo representa y el asignar dicho rol a una persona en particular implica la asignación de actividades específicas. Un mismo rol puede estar asociado a varias personas y de igual manera una persona puede tener varios roles.

ILUSTRACIÓN 1. CONEXIÓN DE ROLES CIDU



Fuente: Autor

Tabla 02. Actividades del Propietario del Producto

Rol	Propietario del Producto (PP)
Descripción	Representa el vínculo directo con la organización cliente, es quien fomenta la aceptación de la futura herramienta al interior de la organización y propone el proyecto como solución a necesidades particulares que han sido identificadas. Además es responsable en gran parte del proyecto.
Actividades	Distribuir apropiadamente el presupuesto asignado para el proyecto, prestando especial atención a los intereses de su organización. Determinar las prioridades del producto software de acuerdo a las necesidades de la organización. Facilitar espacios de comunicación con los usuarios de la futura aplicación. Promover el mantenimiento y realización de mejoras del producto, una vez terminado el proyecto actual.

Fuente: Autor



Tabla 03. Actividades del Líder del Proyecto

<i>Rol</i>	Líder del Proyecto (LP)
<i>Descripción</i>	Es el principal canal de comunicación entre la organización y el equipo de trabajo, su misión es velar por los intereses del cliente y del equipo de desarrollo.
<i>Actividades</i>	Asignar recursos y presentar alternativas apropiadas al cliente Delegar responsabilidades, previa identificación y aceptación de roles. Planificar cada detalle durante el desarrollo del proyecto Incentivar la participación del cliente durante el proceso de desarrollo Definir planes de contingencia que ayuden a preservar el proceso de desarrollo y el producto, de riesgos inminentes. Regular el proceso de desarrollo mediante el monitoreo y la identificación de inconvenientes, que impidan la ejecución prevista de las actividades.

Fuente: Autor

Tabla 04. Actividades del Usuario Representativo

<i>Rol</i>	Usuario Representativo(UR)
<i>Descripción</i>	Es un usuario seleccionado para comunicar con claridad todo lo relacionado con las funcionalidades del futuro sistema, esta persona debe conocer a fondo las tareas básicas de usuarios afines.
<i>Actividades</i>	Relacionar con detalle cada tarea cotidiana que será asociada con el producto software, en relación a su perfil. Presentar alternativas del proceso actual cuando le sean solicitadas Presentar sugerencias frente a los resultados mostrados Participar en el desarrollo de las pruebas realizadas al producto

Fuente: Autor

Tabla 05. Actividades del Coordinador Experto

<i>Rol</i>	Coordinador Experto (CE)
<i>Descripción</i>	De acuerdo a sus capacidades una persona del equipo puede ser coordinador experto en un área específica, por ejemplo diseño de arquitectura, pruebas, entre otras.
<i>Actividades</i>	Ayudar a resolver las dificultades que estas áreas traen implícitas de manera que se puedan solucionar con facilidad y rapidez. Velar por el mejoramiento del proceso mismo

Fuente: Autor

Tabla 06. Actividades del Equipo de Desarrollo

<i>Rol</i>	Equipo (E)
<i>Descripción</i>	Este rol obedece en realidad a cada uno de los participantes del equipo de desarrollo, quienes realizan funciones de Analista, Arquitecto, Desarrollador, Tester.
<i>Actividades</i>	Seleccionar, planificar y/o aplicar los métodos de indagación necesarios para el levantamiento de requisitos.



Identificar los requisitos en base a los resultados
Presentar los requerimientos en forma clara y concisa, mediante la creación de la especificación de requerimientos.
Definir la arquitectura a utilizar y mejorar en forma paulatina el acoplamiento de la aplicación.
Definir lineamientos generales del diseño y la implementación.
Construir cualquier prototipo necesario para probar aspectos riesgosos desde el punto de vista técnico en el proyecto
Codificar los componentes asignados para su desarrollo en cada iteración.
Crear y aplicar tests individuales sobre el código desarrollado
Documentar y realizar los cambios pertinentes según pruebas aplicadas
Planificar, diseñar y aplicar las correspondientes pruebas funcionales (automatizadas o manuales) en base a la especificación de requisitos correspondiente de cada iteración.
Analizar los resultados generados en cada prueba y de acuerdo a ellos solicitar los cambios y medidas pertinentes para dar solución a los inconvenientes identificados.

Fuente: Autor

- C Complejidad de la aplicación a desarrollar

Existen características definidas que influyen al determinar el grado de complejidad o dificultad al desarrollar la nueva herramienta, lo cual es clave a la hora de establecer tiempos de desarrollo y distribución de recursos. En ocasiones puede hablarse de mayor documentación a fin de dejar memorias que faciliten trabajos futuros de éste tipo y de mayores controles de calidad para el producto desarrollado. El grado de complejidad se define en consideración de los siguientes factores:

- D Factores del equipo de desarrollo

Hay que considerar factores asociados al recurso humano que pueden influir seriamente en el proceso de desarrollo, pues no siempre se trabaja bajo condiciones ideales. Entre estos factores se tienen:

- Experiencia en aplicaciones
- Factores comunicacionales
- Experiencia en el lenguaje y las herramientas
- Continuidad del personal
- Experiencia en la plataforma
- Capacidad de los programadores
- Capacidad de los analistas



En base a ellos puede darse una mejor aproximación del tiempo utilizado en el desarrollo de cada actividad y por tanto del desarrollo total.

- E Entrevista General

Su principal objetivo es acordar una idea general del producto requerido, relacionando los procesos al interior de la organización cliente que se verán reflejados en la herramienta desarrollada. A éste punto es necesario haber definido los analistas del equipo de desarrollo quienes están encargados de identificar y recrear una visión general e inicial del producto, la cual será objeto de las modificaciones que se consideren necesarias. Adicional a esto, se debe concientizar al cliente sobre la importancia de su participación en el desarrollo del producto, a fin de obtener resultados acordes con las expectativas planteadas, lo cual implica en su momento conceder los espacios y tiempo necesarios para la participación de cada persona requerida en el proceso de desarrollo.

Nota: Cabe destacar que el tiempo de participación de personal cliente, debe ser acordado en consideración del cumplimiento de sus actividades y las prioridades establecidas en el proyecto.



3 C.I.D.U. Ciclo Incremental Documentado y Usable

- A Keep It Short and Simple⁶

Basados en la filosofía “Manténgalo Corto y Simple”, esta metodología propone un proceso de desarrollo iterativo y a la vez incremental, la idea es que el proyecto total sea dividido en varios mini proyectos a fin de facilitar el proceso. Al trabajar en el desarrollo de cada uno de ellos es necesario considerar todas las etapas del ciclo de desarrollo convencional y al terminar se tendrá un producto final que será tratado como subproducto o incremento. El proceso de desarrollo por cada subproyecto será un ciclo CIDU que generará un componente o módulo, en términos del modelo SCRUM se trataría de una especie de sprint.

Esto quiere decir que cada ciclo podrá involucrar una técnica o modelo en su propio proceso de desarrollo, para lo cual se ha seleccionado la metodología ESPIRAL. Bueno, en realidad se utilizará una adaptación de ella puesto que se involucran y omiten ciertas actividades del modelo habitual a fin de producir cambios favorables en el resultado. Se consideran entonces 5 etapas y cada una de ellas involucra una serie de actividades y artefactos como muestra de los resultados en cada una de ellas.

- B Fases C.I.D.U.

Fase 1: Análisis Previo

Descripción

El principal objetivo de ésta etapa es diseñar el esqueleto del módulo o una estructura base que irá tomando forma a medida que transcurren las etapas del ciclo. Entonces, es necesario capturar información de calidad suficiente, y para ello se utilizarán métodos de diseño claves a la hora de recolectar y clasificar datos, a fin de generar la información requerida y la cual hará parte de los artefactos creados en ésta fase.

⁶ Principio KISS, referirse a : http://es.wikipedia.org/wiki/Principio_KISS



Actividades

Entrevistas Generales

Establecer contacto con el cliente a fin de crear una idea general de la futura aplicación, además de conocer las expectativas frente a su desarrollo y motivarlo a participar en el proceso de desarrollo, destacando su importancia para generar óptimos resultados.

Análisis de documentación y Sistemas previos

Se considera una visita importante al contexto de las actividades que desarrollará el sistema. La idea general es deducir los inconvenientes presentados a partir de herramientas previas y el reconocimiento de documentos a manejar, con miras a mejorar el desarrollo actual de las actividades.

Aplicar Métodos de Indagación

La etapa de recolección de datos es común a todas las metodologías de desarrollo y para ello existen diversas técnicas, las cuales deben ser seleccionadas con cuidado para obtener buenos resultados. La mejor forma de hacerlo es identificando previamente los principios de usabilidad a evaluar los cuales se encuentran en el ANEXO B.⁷

Usar Métodos de Categorización de contenidos

La etapa de recolección de datos es común a todas las metodologías de desarrollo y para ello existen diversas técnicas, las cuales deben ser seleccionadas con cuidado para obtener mejores resultados y la mejor forma de hacerlo es identificando previamente los principios de usabilidad a evaluar.⁸

Nota: El ANEXO C presenta el método “Card Sorting” en forma más detallada.

Análisis de Líneas de Comunicación

La comunicación al interior de la aplicación se produce a diferentes niveles:
Usuario-Software
Software-Usuario

⁷ <http://www.sidar.org/recur/desdi/traduc/es/visitable/inquiry.htm#Cont>

⁸ <http://www.sidar.org/recur/desdi/traduc/es/visitable/prototype.htm#Func>



Usuario- Usuario a través del Software
Usuario-Usuario en el proceso real manual

Como resultado de dicho análisis, se genera el diagrama de secuencia que corresponde a la documentación UML. Un diagrama de secuencia muestra las interacciones entre objetos ordenadas en secuencia temporal. Muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad descrita por el escenario.

Arquitectura de la Información (Análisis de Información)

Ésta etapa corresponde a la Modelización del enfoque DOM y la aplicación de una técnica de categorización de contenidos como Card Sorting. En base a DOM se realiza un análisis a la información recolectada acerca de los usuarios a fin de Identificar tanto los posibles roles, como las actividades finales o metas que quieren realizar.

Tomando como referencia los resultados de la técnica de categorización de contenidos y del análisis basado en DOM, se determinan claramente los posibles módulos de la aplicación y las actividades asociadas a ellos. Con lo cual se pueden realizar las siguientes actividades:

Establecimiento de Rutas

Cada uno de los espacios que conforman la aplicación, deben estar interconectados, para ello resulta conveniente tener una barra de navegación que permita mostrar los puntos principales de la aplicación es decir los módulos generales, a fin de permitir que el usuario del sistema tenga fácil acceso a la información y funciones requeridas. De igual manera se debe permitir acceso directo a ellos y a cada uno de los sub-módulos que contiene, el módulo previamente seleccionado, informado al usuario de las opciones que tiene a la mano. También es necesario mostrar al usuario botones básicos que le sirvan de ayuda en caso de sentirse perdido, tales como inicio, mapa del sitio y ayuda.

Como resultado de las pautas anteriores se tiene la Estructura de Navegación, que puede ser organizada jerárquicamente o en forma lineal, cualquiera de estas opciones depende de los criterios del encargado.



Identificación de Procesos Internos

A cada perfil de usuario corresponden tareas y permisos particulares, es así como el ordenador central permite mostrar al usuario solo la información que le compete sin que esto altere el diseño y disposición a los servicios que se requieren del sistema. La descripción de cada uno de estos procesos internos y necesarios para generar tal resultado, harán parte del informe generado como resultado de éste proceso de identificación.

Arquitectura de Información Final

Para esto se puede diseñar un diagrama general que contenga todos los módulos de la aplicación, indicando por cada uno: descripción general, logotipo o colores utilizados, perfil de usuario, descripción de actividades.

Diagrama de Flujo de Arquitectura Final

Es un mapa de diseño de la información que contiene la aplicación, indicando niveles de acceso; para su realización deben estar previamente clarificados los objetivos comunicativos considerando las necesidades del usuario, para organizar el contenido en forma adecuada. La idea es conservar una jerarquía de la información y estructurar el contenido, para que los usuarios puedan tener una idea general de cómo funcionará la aplicación en cuanto lo vean. Este diagrama es una medio claro y sencillo para identificar las categorías temáticas, niveles y vínculos de la aplicación

Nota: Para la diseño de los diagramas de Arquitectura de Información y de Interacción se presenta el Anexo D el cual contiene el vocabulario visual a utilizar para su creación.

Artefactos

Principios y Técnicas	= Formato CIDU (ANEXO)
Documento Arquitectura de Información	= Formato CIDU (ANEXO)



Fase 2: Especulación

Descripción

Esta etapa previa al desarrollo tiene por objetivos básicos definir el alcance y visión del proyecto, así como diseñar un plan de trabajo en base al cual podrá desarrollarse el componente. En base a los artefactos de la etapa previa se especula o se genera una propuesta acerca de lo que se puede producir y en acuerdo con el cliente se definen prioridades a fin de diseñar un plan de trabajo para desarrollar el componente.

Actividades

Documentar Casos de uso Especulativos

En base a nociones básicas e impresiones generadas durante los primeros encuentros con el usuario surgen los primeros diseños de casos de uso, los cuales serán sometidos a cambios en base a lo acordado con el usuario final, definiendo por cada uno los siguientes atributos relacionados en la tabla:

Tabla 07. Información básica requerida por caso de uso

Campo	Descripción
<i>Nombre</i>	
<i>Actores Relacionados</i>	
<i>Pre-condición</i>	
<i>Post-condición</i>	

Fuente: Autor

A fin de estandarizar el proceso, se presenta una lista de los posibles verbos a utilizar para definir el nombre en cada caso.

Tabla 08. Lista de verbos a utilizar para definir casos de uso

Verbo	Descripción
<i>Aprobar</i>	Dar visto bueno. Se utiliza cuando el proceso es complejo, es decir, que implica varias rutas alternas.
<i>Asignar</i>	Relacionar datos
<i>Calcular</i>	Realizar operaciones matemáticas
<i>Cancelar</i>	Recuperar total o parcialmente un proceso
<i>Cargar</i>	Importar datos a una tabla
<i>Cerrar</i>	Terminar un proceso el cual no podrá volver a ser modificado
<i>Consultar</i>	Consultar datos



<i>Crear</i>	Insertar un registro o Iniciar un proceso
<i>Eliminar</i>	Borrar definitivamente información del sistema
<i>Enviar</i>	Enviar notificación a otro usuario para continuar con el proceso
<i>Finalizar</i>	Finalizar vigencia de registros
<i>Generar</i>	Proceso que realiza N transacciones de forma automática. Ej.: Generar boletines
<i>Guardar</i>	Guardar información que no implique la relación de datos
<i>Hacer mantenimiento</i>	Mantenimiento de las tablas soporte (registro, actualización y borrado)
<i>Listar</i>	Mostrar en pantalla un reporte y/o Imprimirlo
<i>Modificar</i>	Actualización de datos
<i>Revisar</i>	Aceptar o rechazar un proceso. Insertar o actualizar un estado.
<i>Rechazar</i>	No aprobar. Se utiliza cuando el proceso es complejo, es decir, que implica varias rutas alternas.
<i>Verificar</i>	Validar datos

Fuente: Autor

Listar Requerimientos

Con base en información previa y la obtenida durante las entrevistas, se lleva a cabo un análisis de la misma a fin de establecer los requerimientos generales del sistema, con ello se quiere satisfacer las necesidades no sólo de las personas a cargo del proceso sino de los usuarios del sistema y quienes requieren los servicios del mismo. Por cada requisito se tendrán los datos relacionados en la presente tabla:

Tabla 09. Datos básicos por requerimiento

<i>Código</i>	
<i>Nombre</i>	
<i>Actor</i>	
<i>Propósito</i>	
<i>Resultado</i>	

Fuente: Autor

A su vez se identificarán los perfiles de usuario del sistema y se definirán las tareas a realizar según la siguiente tabla

Tabla 10. Datos básicos por tipo de usuario

<i>Perfil</i>	
<i>Habilidades</i>	
<i>Actividades</i>	

Fuente: Autor



Aprobación y priorización de Requerimientos

En cuanto el listado de requerimientos sea aprobado de común acuerdo con el cliente, haciendo las correcciones pertinentes se procede a realizar una priorización de los requerimientos en base a las necesidades más importantes del cliente. Ésta priorización será clave en el desarrollo del ciclo en éste momento se hace referencia al documento de Programación de Actividades Específicas (PAE). El formato estándar creado para tal efecto se encuentra en el ANEXO del presente documento.

Documentación de Casos de Uso no especulativos

Una vez definidos los requerimientos del sistema, se procede a realizar una revisión de los casos de uso contemplados hasta el momento. Éste proceso es realizado bajo la supervisión de la director del proyecto, ya que la influencia de los mismos durante su desarrollo puede generar cambios importantes, debido a que estos representan los EJB DE SESION ó LÓGICA DEL NEGOCIO, concepto que se verá en detalle más adelante. Cabe recordar que por lo general un caso de uso está asociado mínimo a un requisito particular, por tanto es conveniente que en base a la corrección de los requisitos establecidos se determinen cuales son los casos de uso definitivos para continuar desarrollando la documentación UML correspondiente en cada caso:

- Diagrama de casos de uso
- Diagrama de Actividades
- Diagrama de Secuencia

Diseñar Programa de Actividades Específicas (PAE)

Se pueden identificar una serie de actividades por usuario las cuales pueden ser programadas en forma individual o en conexión a actividades previamente funcionales. Una vez identificadas pueden ser asignadas y controladas en cuanto a su desarrollo mediante el formato creado para ello, ANEXO .



Artefactos

- Documentación UML
- Especificación de Requisitos
- Programa de Actividades Específicas (PAE)

Fase 3: Diseño y Desarrollo

Descripción

Esta fase tiene por objetivo principal completar el desarrollo de cada una de las funcionalidades de un incremento del producto, que han sido determinadas por el equipo en la fase anterior, en base a lo dispuesto en el PAE. Para ello se deben clarificar los requerimientos pendientes, administrar los cambios de acuerdo a las evaluaciones y observaciones dadas por los futuros usuarios para realizar las mejoras pertinentes. Se tienen una serie de actividades a fin de realizar tanto un diseño lógico como un diseño físico de la aplicación. El primero está relacionado con la forma en que el sistema habrá de cumplir con los requerimientos, sirve de gran ayuda en éste punto tener aprobados los diagramas de arquitectura de información y diseño de interacción respectivamente, ya que en base a ellos se debe diseñar la estructura y distribución de elementos en las diferentes pantallas, se debe identificar claramente la información a recolectar, a procesar y los resultados a mostrar en cada caso (formato, comprobación de registro, informe, entre otros), en realidad se tienen más tareas implícitas cuyo objetivo común es crear especificaciones de software completas para su posterior desarrollo.

Actividades

Análisis Contextual

Para llevar a cabo el análisis contextual se deben identificar los objetos, la plataforma y el escenario óptimo

Objetos

En este caso se entiende por objetos cosas físicas o conceptuales, es necesario incluir la mayor parte aún cuando parezcan irrelevantes. Como ejemplo de los que



debe presentarse como resultado de la identificación de objetos se tiene la siguiente lista:

- Mensaje Inicial
- Login
- Password
- Botón de ingreso
- Botón de Ayuda
- Botón de Cerrar
- Encabezado
- Pie de página
- Etc...

Nota: Cada uno de los escenarios tiene sus propios objetos

Plataforma

En cuanto a plataforma se refiere, debe definirse también el estándar a utilizar, el motor de base de datos, el servidor de aplicaciones y el lenguaje. Se definen los exploradores en los cuales podrá ser cargada la aplicación, por lo general se han de considerar los más usados y de ser posible las últimas versiones de los mismos

Escenario Óptimo

Se tendrán muchos posibles escenarios, cada uno corresponde a cada una de las actividades asignadas a cada uno de los perfiles de usuario. En tal caso cada uno de estos escenarios debe tener una breve descripción de la actividad realizada, el perfil del usuario que la lleva a cabo y un nombre que lo identifique.

Diseño de Interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento.

Diseño de Bases de Datos

A su vez ésta actividad implica seleccionar un motor de base adecuado a las necesidades de la aplicación. Al diseñar la base de datos se deben considerar varios aspectos entre los que se destacan:



- El tamaño de la información a manejar
- El tipo de información
- La velocidad de acceso requerida
- La facilidad de acceso y extracción
- Las condiciones ofrecidas por el manejador de base de datos
- Acceso eficiente a la información con redundancia mínima
- Diseño de esquemas con la forma normal
- Información Adicional
- Especificación de limitantes

Como resultado de ésta actividad se tiene el diagrama de estructura de datos, el cual describe a grandes rasgos la relación entre las entidades de un sistema y el conjunto de información relacionado.

Elaborar Propuesta de Diseño Gráfico

En base a los diagramas de arquitectura de información, de diseño de interacción y los objetos identificados durante el análisis contextual, se pueden definir claramente los elementos de pantalla en cada página a diseñar. Existen ciertas páginas con características comunes que pueden utilizar la misma presentación y distribución de elementos, a su vez existen elementos que son comunes a la mayor parte de páginas de la aplicación, lo cual propone el uso de CSS a fin de facilitar el desarrollo de la aplicación. CSS es un lenguaje de hojas de estilo creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS también representa la mejor forma de separar los contenidos de la presentación o aspecto de la página, permitiendo cierta independencia entre diseñador y programador. En cuanto al diseño de interfaz gráfica se pueden presentar diferentes alternativas: de ubicación de elementos, de diagramación y de color entre otras, la idea de manejar varias propuestas es que el cliente seleccione la más acorde a sus necesidades.

Aprobación de Diseño Lógico y Gráfico

En cuanto sean seleccionadas las alternativas para diseño de interfaz gráfica, se podrá crear las plantillas finales (css, psd, gif, html y js) a utilizar por la aplicación y los Story Board de las plantillas.

Desarrollo de la Aplicación (Diseño Físico)

Previa selección, instalación y configuración de las herramientas software a utilizar, los desarrolladores se disponen a escribir los programas necesarios para el perfecto funcionamiento del sistema lógico diseñado. A su vez es necesario documentar el código a medida que se va desarrollando el producto software, para ello se utilizan comentarios dentro del código fuente cuyo objetivo es realizar una



descripción básica de variables, métodos utilizados, entre otros que sirvan para facilitar futuros mantenimientos.

Artefactos

- Diseño de interacción (DI).
- Mapa del Sitio (MS).
- Diagrama de estructura de datos (DED)
- Diccionario de Datos
- Alternativas de Diseño de Interfaz Gráfica (ADIG).
- Alternativas de Diseño Final (ADF). Plantillas finales (css, psd, gif, html y js).
- Story board de las plantillas Finales (SB).
- Diagrama de Navegación

Fase 4: Pruebas

Descripción

El principal objetivo de ésta fase consiste en asegurar que una vez integrados todos los componentes, la herramienta generada esté disponible en condiciones ideales y para ello se utilizan los componentes uno a uno en forma experimental. CIDU evoca los aspectos positivos de metodologías de desarrollo existentes y su fase de pruebas no es la excepción, así que comparte el punto de vista de XP(eXtreme Programming) quien considera que una herramienta debe ponerse a prueba tanto como sea posible, lo cual presenta efectos secundarios convenientes para el desarrollo de cualquier proyecto, como el incremento en la calidad de sus resultados, reducción del número de errores, aumento en seguridad, entre otros.

Actividades

Pruebas unitarias

Para ello es necesario utilizar datos de prueba seleccionados de forma conveniente a fin de comprobar que el sistema no caiga en errores comunes o previstos, también pueden participar los usuarios finales del sistema realizando tareas cotidianas mediante la herramienta a fin de observar su interacción con ella y tomar nota de los cambios que se necesitan, para realizarlos posteriormente. Por lo general están encargadas de verificar el código y son las usualmente practicadas a la par con el desarrollo de la herramienta.

Desarrollo de pruebas funcionales

Destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida por el cliente final



Aplicar inspecciones y/o test de usabilidad

La aplicación de métodos que sean de tipo inspección o test de usabilidad permite evaluar el producto desde su contexto. (). A este punto la aplicación de pruebas y tests de usabilidad juega un papel importante pues como se manifestó en ocasiones previas dichos resultados mostrarán el éxito de la solución tecnológica.

Balance de Resultados

Por último se ha de verificar que el producto cumpla a cabalidad con las especificaciones acordadas, lo cual se refleja en la combinación de resultados de las pruebas practicadas.

Artefactos

- Inspecciones y/o test de usabilidad
- Material generado en base al desarrollo de esta etapa (videos, audio, etc)

Fase 5: Documentación

Descripción

El proceso de desarrollo debe involucrar una fase de documentación necesaria para facilitar la futura evolución de la herramienta creada, ya que los documentos como resultado de ella deben informar sobre cada aspecto importante del desarrollo del producto incluyendo observaciones, recomendaciones, mejoras o mantenimiento posterior.

Actividades

Desarrollo de Documentación

Al hablar de la documentación necesaria se hace referencia a la creación de los manuales para el desarrollador, para cada usuario del futuro sistema y de ser posible un documento informando al líder del proyecto sobre los inconvenientes presentados, soluciones alternativas encontradas en cada caso y las mejoras propuestas para la herramienta creada. Adicional a esto es necesario documentar apropiadamente el código fuente generado desde inicios del proyecto a fin de beneficiar mantenimientos futuros, el Anexo D del presente documento relaciona unas normas básicas para documentar código fuente mediante javadoc en caso de trabajar con java, a su vez se muestran unos formatos estándar para crear manuales de usuario final y para desarrollador.

Nota: Cabe destacar que los manuales pueden ser documentos o videos.



Capacitación de usuarios

Según la cantidad de personas que correspondan a un rol de usuario en particular se define el número de participantes en la capacitación.

Artefactos

- Manuales para Usuarios del Sistema
- Manual para el desarrollador
- Manual Líder del proyecto
- Diccionario de Datos o documentos generados por javadoc

Artefactos C.I.D.U.

Tabla 11. Descripción general de artefactos por fase

ANÁLISIS PREVIO	
<i>Principios y Técnicas</i>	Documento para definir los parámetros de usabilidad prioritarios del proyecto y las técnicas para medición de estos
<i>Caracterizaciones</i>	Permite especificar los procesos a sistematizar desde el punto de vista del cliente, a modo de diagramación de un proceso de gestión de calidad
<i>Documento Arquitectura de Información</i>	Pese a carecer de resultados tangibles aparentes la suma de actividades puede generar un documento que las combine como el caso del Documento de arquitectura de información, el cual consta de una perfecta combinación entre el Análisis de líneas de comunicación, la modelización del enfoque DOM y la Arquitectura de información.
ESPECULACIÓN	
<i>Documentación UML</i>	Diagrama de casos de uso, Diagrama de Actividades, Diagrama de Secuencia
<i>Especificación de Requisitos</i>	Lista de requerimientos tomados con prioridad definida
<i>Programa de Actividades Específicas (PAE)</i>	Programación para el desarrollo de las actividades correspondientes a los requerimientos obtenidos
DISEÑO Y DESARROLLO	
Diseño de interacción (DI).	Diagramación en bocetos del software y sus elementos funcionales
Mapa del Sitio (MS).	Estructura de Información definida para el software
Diagrama de estructura de datos (DED)	Este es el diagrama entidad relación de la base de datos con especificación de campos, llaves, etc
Diccionario de Datos	Es una estructura básica que almacena datos del sistema incluyendo por cada uno nombre, descripción, alias, contenido y organización, entre otros datos opcionales a fin de identificar procesos donde se emplean los datos, sitios



Alternativas de Diseño de Interfaz Gráfica (ADIG).	Opciones Gráficas en boceto para pruebas con el usuario
Alternativas de Diseño Final (ADF). Plantillas finales (css, psd, gif, html y js).	Alternativas de color, diagramación (ubicación de elementos funcionales).
Story board de las plantillas Finales (SB).	Definición de íconos, fuentes, colores, para cada plantilla, etc
Diagrama de Navegación	Interacción entre actividades y elementos funcionales de la plantilla
PRUEBAS	
Notas	Libro de notas con registro de la respuesta del usuario a cada prueba realizada.
DOCUMENTACIÓN	
Manuales para Usuarios del Sistema	Manuales de instrucciones para el uso del software
Manual para el desarrollador	Explicación de métodos complejos y estructura de desarrollo
Manual Líder del proyecto	Recuento de actividades desarrolladas
Diccionario de Datos o documentos generados por javadoc	Especificación de propiedades y descripciones de los elementos en persistencia

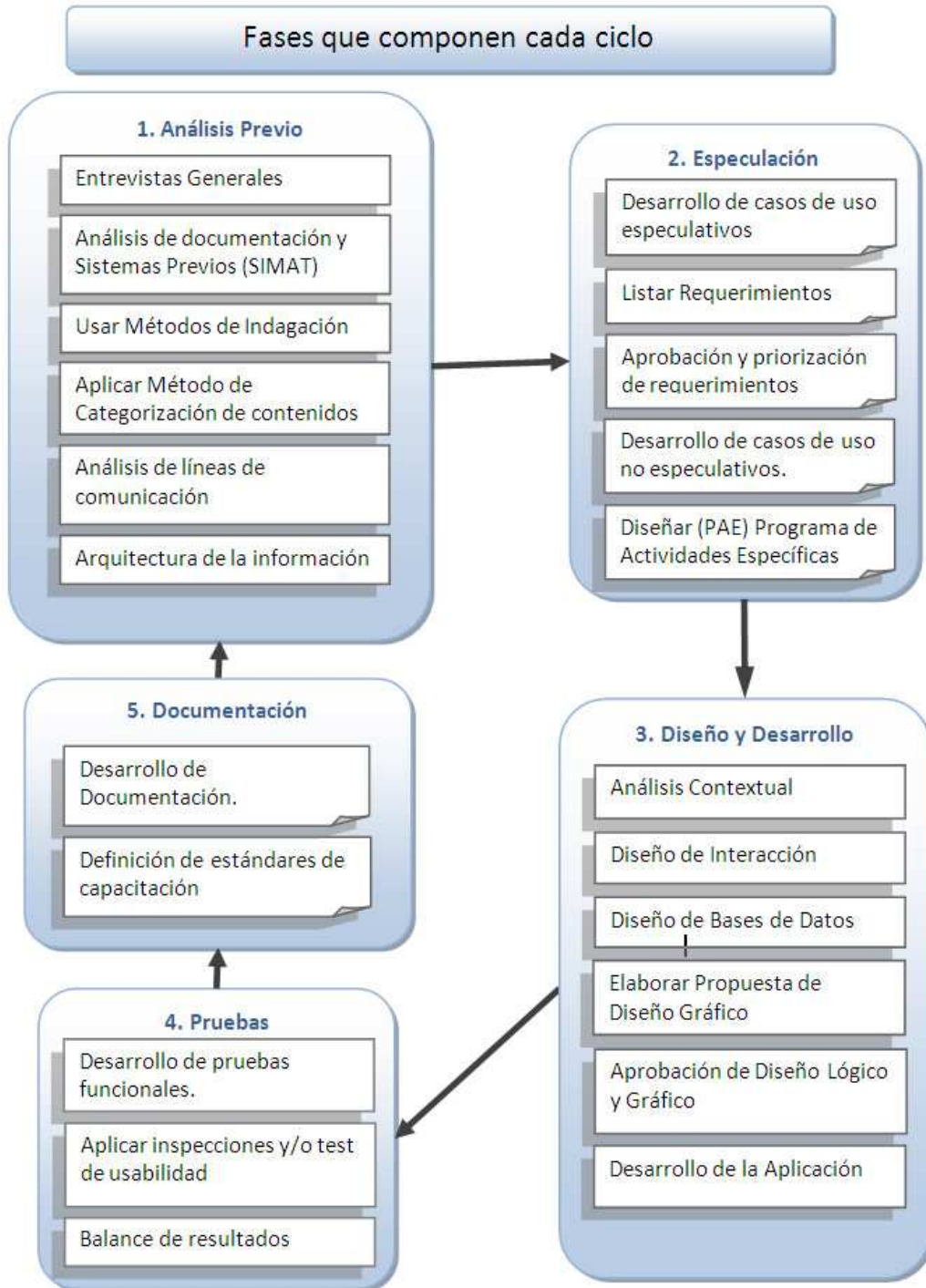
Fuente: Autor

Algo más sobre C.I.D.U.

- Es recomendada para proyectos con equipos de desarrollo conformados de 5 a 25 personas
- Los proyectos a ser desarrollados por CIDU deben ser fácilmente modularizables.
- Ésta metodología puede requerir de mayor inversión económica debido a que el personal que conforme el equipo de desarrollo debe cumplir con un alto perfil profesional.
- La metodología no prescribe técnicas, aunque pueda referirse a algunas conocidas de antemano
- No es necesario el uso de herramientas específicas para desarrollar las actividades propuestas.
- No se plantea un estándar para los entregables, aunque se proponen formatos
- También será independiente del lenguaje o la arquitectura utilizada, así como del tipo de software que se está construyendo.



ILUSTRACIÓN 2. FASES QUE COMPONEN CADA CICLO



Fuente: Autor



IV. Proceso de Desarrollo.

1 Aspectos Generales a Considerar

En etapa temprana a la realización de cualquier proyecto con CIDU se deben considerar 5 aspectos generales que pueden influir en el tiempo de desarrollo que tomará el proyecto en cada etapa, tal y como se menciona en el capítulo anterior:

Número de participantes en el equipo de desarrollo

En éste caso se cuenta para el proyecto con 3 integrantes, el director de la tesis y las estudiantes relacionadas con su desarrollo.

- A Identificar Roles

Tabla 12. Roles de Equipo SOFGESCOL

Participante	Emiro Muñoz J.	Laura Pulido D.	Luz Dary Méndez U.
<i>Líder del Proyecto</i>	X		
<i>Analista</i>		X	X
<i>Arquitecto</i>		X	X
<i>Programador</i>		X	X
<i>Tester</i>		X	X
<i>Coordinador Experto ()</i>		X	
<i>Coordinador Experto ()</i>			X

Fuente: Autor

- B Complejidad de la Aplicación a Desarrollar

Experiencia en aplicaciones: En base al conocimiento y la experiencia adquirida durante la vida universitaria, el personal a cargo del proyecto presenta condiciones adecuadas para su completo desarrollo.

Factores comunicacionales: CIDU requiere de un contacto constante entre los participantes del equipo de desarrollo y del cliente, éste se ve reflejado en las



reuniones realizadas durante el proyecto, y el uso adecuado de herramientas comunicacionales existentes (vía telefónica, correo electrónico, skype, etc.)

Experiencia en el lenguaje y las herramientas: Tanto el lenguaje como las herramientas tecnológicas a utilizar durante el proyecto, han sido seleccionadas cuidadosamente por presentar afinidad con las facultades de los desarrolladores del proyecto, a fin de obtener mejores resultados en el tiempo previsto. Sin embargo se realizó una breve investigación e intercambio de información a partir de falencias existentes respecto a algunas funcionalidades de dichas herramientas.

Continuidad del personal: Podría decirse que no aplica puesto que los participantes no desertan y no cambian en éste tipo de proyecto (Tesis de grado).

Experiencia en la plataforma: Se prevé un manejo adecuado en base a la experiencia con versiones previas. Se cuenta con suficiente información adicional respecto al funcionamiento de dicha plataforma, a fin de superar rápidamente contratiempos generados a partir de ella.

Capacidad de los programadores: Se cuenta con suficiente disposición, dedicación responsabilidad y conocimientos claves en el manejo del lenguaje y las herramientas a usar. Sin embargo, hay que considerar la adquisición de destrezas, competencias y conocimientos adicionales en función de nuevas experiencias así como el manejo y solución de inconvenientes.

Capacidad de los analistas: Se asume que los analistas tomarán posturas adecuadas con respecto a los futuros usuarios a fin de crear la confianza necesaria y obtener la mayor y mejor información, como fuente de análisis posterior.

- C Entrevista General

En consideración del corto tiempo destinado al desarrollo del proyecto se limitó el alcance del mismo a un proceso particular que requiere de gran atención en cualquier organización afín el “Proceso de Registro Académico”, que se compone de tres subprocesos básicos a considerar los cuales presentan problemas particulares en cada caso. La idea general aborda la creación de un sistema estándar que facilite el desarrollo de cada una de las actividades relacionadas con el proceso seleccionado. La postura adoptada por la institución frente al desarrollo de la nueva herramienta, es idónea pues muestran disposición suficiente para servir como fuente de información para el proyecto. Los documentos asociados a ésta actividad son Acta de Reunión AR01 (ANEXO) y el informe de reunión IR01(ANEXO).



En base al último el orden en que serán sistematizados es:

- Subproceso Matrícula y Pagos
- Subproceso Gestión de Talento Humano
- Subproceso Gestión de Información Académica

2 Aspectos Generales sobre la sistematización de procesos

Cada una de las fases que conforman un ciclo CIDU se encuentra relacionada con Documentos y Formatos que han sido diseñados para facilitar el desarrollo de las mismas, como se menciona en el capítulo anterior. Por consiguiente el desarrollo de cada actividad hará énfasis a los anexos que sean necesarios y las observaciones en cada caso.

En éste proyecto se aplica el concepto de Persistencia gestionada por el contenedor. Por consiguiente, un Entity Bean será una clase (POJO: Plain Old Java Objects) que representa una tabla de una base de datos, y cada instancia de ésta clase representa un registro de la tabla. Para esto, se llevo a cabo el correspondiente “mapeo” entre las propiedades de una clase y los campos que pertenecen a la tabla.

Nota: La sistematización del subproceso supone un ciclo CIDU por ello se desarrollarán las fases descritas en el capítulo previo. La identificación de componentes o módulos a desarrollar se realiza en base al documento de Arquitectura de Información.

Sistematización del subproceso "Matrícula y Pagos"

Análisis Previo: Resumen de actividades

Identificar los procedimientos asociados al subproceso:

- “Registro de Matrícula”
- “Control de Pagos”

Recolectar documentos fuente requeridos para la creación del sistema

- Formato de Pre matrícula (DF001)
- Formato de Matrícula (DF002)
- Recibo de Pago para Matrícula(DF003)
- Carné estudiantil (DF004)
- Recibo de Pago por otros conceptos (DF005)



- Decreto de especificación de costos educativos y matrículas del año (DF006)
- Lista de documentos requeridos para matrícula (DF007)

Aplicar Métodos de Indagación

- Entrevistas
- Análisis Contextual

De ser necesario caracterizar procedimientos

- Caracterización del Procedimiento Registro de Matrícula CP01
- Caracterización del Control de Pagos CP02
- El documento de Caracterización de Procedimiento para “Registro de Matrícula” fue aprobado sin mayores contratiempos, en cuanto al de “Control de Pagos” se tiene un par de cambios a realizar, respecto a la actividad de registro de pagos.

Aplicar Métodos de Categorización de contenidos

Se aplicó la técnica de card sorting tipo abierto, a fin de identificar si se puede presentar solo una categoría o más:

- *Categorías:* se entregaron tres papeles de categoría en blanco para que dieran el nombre tendrían.
- *Tarjetas:* Pre – Matricular, Matricular, Generar Recibos, Registrar Pagos, Información Personal, Informes, tres papeles en blanco para tarjetas sugeridas.

En base al análisis cualitativo de resultados se tiene que el personal identificó dos posibles categorías pertenecientes al subproceso cuyos nombres corresponden a “Matrícula” y “Pagos”, las tarjetas fueron asociadas rápidamente de acuerdo a las actividades que corresponden a cada procedimiento y como observación se presenta una tarjeta adicional por cada categoría que corresponde a Informes. Se acuerda con ellos separar los datos del usuario en secciones según corresponda a los formatos para registrar pre-matrícula y matrícula, respectivamente.

En base a esta información se pueden identificar los roles y metas asociados al subproceso en función del enfoque DOM.

Se acordaron los perfiles de usuario para el subproceso, en este caso se tienen:



Administrador (Rectora, Secretaria, Auxiliar de pagos), estudiante, Acudiente. Se aprobó la mayor parte de las tareas asociadas a cada perfil, presentando un par de observaciones que ya fueron corregidas.

En base al método “Investigación Contextual”, se obtuvo información adicional del usuario respecto a expectativas, alternativas de desarrollo a los procedimientos, opiniones y experiencias, lo cual permitirá evaluar el contexto en el que se desarrollan los procedimientos y permitir mejoras de las actividades pertinentes.

En base a Card Sorting y la técnica de Story Board se tiene información suficiente para continuar la creación de los artefactos asociados a la fase actual (Análisis Previo) y pasar a la siguiente.

Observaciones en cuanto a la metodología (Análisis Previo)

En base al desarrollo de ésta primera fase (Análisis Previo) se definieron los artefactos a entregar, y se establece un nuevo orden para el desarrollo de las actividades que corresponden a la fase.

En base al documento de arquitectura de información se definen dos componentes a desarrollar cuyos nombres son: “Matrícula” y “Control de pagos”, pese a estar ligados serán creados en ciclos diferentes que inician a partir de la fase especulación y posteriormente serán integrados.

Nota: Dichos ciclos compartirán la información generada en la fase “Análisis previo”

Tabla 13. Artefactos Análisis Previo Ciclos: “Matrícula” y “Control de Pagos”

Actividades	Artefactos
<i>Usar Métodos de Indagación</i>	<i>Principios y Técnicas (PT01)</i> <i>Caracterizaciones: CP01, CP02</i>
<i>Análisis de documentación y Sistemas Previos (SIMAT)</i>	
<i>Usar métodos de Categorización de contenidos</i>	
<i>Análisis de líneas de comunicación</i>	<i>Diagrama de Arquitectura de Información (DAI01)</i>
<i>Modelización del enfoque DOM</i>	
<i>Arquitectura de la información</i>	

Fuente: Autor



3 **CIDU Matrícula**



Componentes fase: Especulación

Tabla 14. Artefactos fase Especulación. Ciclo: “Matrícula”

Actividades	Artefactos
<i>Documentación de casos de uso especulativos</i>	
<i>Listar Requerimientos</i>	Especificación de requisitos software priorizada ANEXO : ERS [Sección A]
<i>Aprobación y priorización de requerimientos</i>	
<i>Documentación de casos de uso no especulativos</i>	Documentación UML [Sección A]
<i>Diseñar Programa de Actividades Específicas (PAE)</i>	ANEXO : PAE [sección]

Fuente: Autor

Componentes fase: Diseño y Desarrollo

Tabla 15. Artefactos fase Diseño y Desarrollo. Ciclo: “Matrícula”

Actividades	Artefactos
<i>Análisis Contextual</i>	Diseño de interacción (DI). ANEXO [Sección A] Mapa del Sitio (MS). ANEXO [Sección A]
<i>Diseño de interacción</i>	
<i>Diseño de Bases de Datos.</i>	Diagrama de estructura de datos (DED). ANEXO [Sección A] Diccionario de Datos
<i>Propuesta Diseño grafico.</i>	Alternativas de Diseño de Interfaz Gráfica (ADIG). Alternativas de Diseño Final (ADF).Plantillas finales (css, psd, gif, html y js).
<i>Aprobación</i>	Story board de las plantillas Finales (SB). ANEXO
<i>Desarrollo de la Aplicación</i>	SW
<i>Pruebas en tiempo de desarrollo</i>	No aplica

Fuente: Autor

Componentes fase: Pruebas



Tabla 16. Artefactos fase Pruebas. Ciclo: “Matrícula”

Actividades	Artefactos utilizados
<i>Pruebas de requisitos</i>	Especificación de requisitos sección A (utilizado)
<i>Pruebas de unidad para el módulo</i>	No aplica
<i>Pruebas de aceptación</i>	Historial de Uso sección A (utilizado)

Fuente: Autor

Componentes fase: Documentación

Tabla 17. Artefactos fase Documentación. Ciclo: “Matrícula”

Actividades	Artefactos
<i>Creación de Manuales de usuario</i>	<i>Manual General de Usuarios sección E por usuario</i>
<i>Creación manuales equipo de desarrollo</i>	<i>Manual Técnico y Diccionario de Datos sección E.</i>

Fuente: Autor

Observaciones en cuanto a la metodología en base al ciclo

El desarrollo de éste primer ciclo tomó más tiempo del previsto, pues debían generarse las bases del sistema en cuanto a páginas generales del sistema y los primeros mecanismos de control de acceso.

Pese a considerar diversos artefactos dicha documentación se va incrementando a medida que se desarrolla la herramienta. Cada ciclo incorpora una nueva sección en consideración de los artefactos creados. Claro está, las alternativas de diseño de interfaz no se crearán en igual cantidad que el número de ciclos. La idea es que en los primeros ciclos se defina una plantilla en base a la cual se puedan trabajar todos los ciclos.

Para el presente proyecto se modificó la plantilla inicialmente seleccionada debido a una mal distribución del espacio y en consideración de la carga informativa presentada y los inconvenientes generados.

4 **CIDU Pagos**



Componentes fase: Especificación



Tabla 18. Artefactos fase Especificación. Ciclo: “Pagos”

Actividades	Artefactos
<i>Documentación de casos de uso especulativos</i>	
<i>Listar Requerimientos</i>	Especificación de requisitos software priorizada ANEXO : ERS [Sección B]
<i>Aprobación y priorización de requerimientos</i>	
<i>Documentación de casos de uso no especulativos</i>	Documentación UML [Sección B]
<i>Diseñar Programa de Actividades Específicas (PAE)</i>	ANEXO : PAE [sección ■■■]

Fuente: Autor

Componentes fase: Diseño y Desarrollo

Tabla 19. Artefactos fase Diseño y Desarrollo. Ciclo: “Pagos”

Actividades	Artefactos
<i>Análisis Contextual</i>	Diseño de interacción (DI). ANEXO [Sección B] Mapa del Sitio (MS). ANEXO [Sección B]
<i>Diseño de interacción</i>	
<i>Diseño de Bases de Datos.</i>	Diagrama de estructura de datos (DED). ANEXO [Sección B] Diccionario de Datos (Modificado)
<i>Propuesta Diseño grafico.</i>	Alternativas de Diseño de Interfaz Gráfica (ADIG). Alternativas de Diseño Final (ADF).Plantillas finales (css, psd, gif, html y js).
<i>Aprobación</i>	Story board de las plantillas Finales (SB). ANEXO
<i>Desarrollo de la Aplicación</i>	SW
<i>Pruebas en tiempo de desarrollo</i>	No aplica

Fuente: Autor

Componentes fase: Pruebas

Tabla 20. Artefactos fase Pruebas. Ciclo: “Pagos”

Actividades	Artefactos utilizados
<i>Pruebas de requisitos</i>	Especificación de requisitos sección B (utilizado)
<i>Pruebas de unidad para el módulo</i>	No aplica
<i>Pruebas de aceptación</i>	Historial de Uso sección B (utilizado)

Fuente: Autor



Componentes fase: Documentación

Tabla 21. Artefactos fase Documentación. Ciclo: “Pagos”

Actividades	Artefactos
<i>Creación de Manuales de usuario</i>	<i>Manual General de Usuarios sección E por usuario</i>
<i>Creación manuales equipo de desarrollo</i>	<i>Manual Técnico y Diccionario de Datos sección E.</i>

Fuente: Autor

Observaciones en cuanto a la metodología en base al ciclo

La nueva plantilla permite una mejor organización de elementos de pantalla, se elimino uno de estos elementos (el icono de módulo activado) y fue agregado el div de tiempo (Fecha y hora).

El diccionario de datos es un componente que se va modificando a medida que se desarrollan los ciclos, el documento final es considerado como un único artefacto.

Sistematización del subproceso Gestión de Recurso Humano

Análisis Previo: Resumen de actividades

Identificar los procedimientos asociados al subproceso:

“Registro de información”

“Control de Objetivos”

Recolectar documentos fuente requeridos para la creación del sistema

Formato básico de Hoja de Vida (DF001)

Aplicar Métodos de Indagación

Entrevistas

Análisis Contextual

De ser necesario caracterizar procedimientos

Como los procedimientos asociados son intuitivos y se asemejan a mantenimientos generales, se omite el paso de creación de caracterizaciones. Como aclaración especial se pueden asignar objetivos, tareas, y sub-tareas, (se mencionan en el orden de dependencia entre ellos).

Aplicar Métodos de Categorización de contenidos

Se aplico la técnica de card sorting tipo abierto, a fin de identificar si se puede presentar solo una categoría o más:



Categorías: se entregaron tres papeles de categoría en blanco para que dieran el nombre tendrían.

Tarjetas: Buscar Hoja de Vida, Nuevo Docente, Nuevo Empleado, Objetivos, Actividades, Nueva actividad, Nuevo Objetivo, Modificar Objetivo, Modificar Actividad, tres papeles en blanco para tarjetas sugeridas.

En base al análisis cualitativo de resultados se tiene que el personal identifico una categoría, el usuario sugiere opciones de búsqueda en cada caso. Se acuerda separar los datos del usuario en secciones según corresponda a los formatos para información personal e información del colegio respectivamente. De otro lado, a fin de facilitar el desarrollo de tareas cotidianas se propone un módulo adicional “Correo”, que permita la comunicación interna de personal de colegio.

En base a ésta información se pueden identificar los roles y metas asociados al subproceso en función del enfoque DOM.

Se acordaron los perfiles de usuario para el subproceso, en éste caso se tienen: Rector, Secretaria, Docente, Empleado.

En base al método “Investigación Contextual”, se obtuvo información adicional del usuario respecto a expectativas, alternativas de desarrollo a los procedimientos, opiniones y experiencias, lo cual permitirá evaluar el contexto en el que se desarrollan los procedimientos y permitir mejoras de las actividades pertinentes.

En base a Card Sorting y la técnica de Story Board se tiene información suficiente para continuar la creación de los artefactos asociados a la fase actual (Análisis Previo) y pasar a la siguiente.

5 **CIDU Talento Humano**



Componentes fase: Especulación

Tabla 22. Artefactos fase Especulación. Ciclo: “Talento Humano”

Actividades	Artefactos
<i>Documentación de casos de uso especulativos</i>	No aplica
<i>Listar Requerimientos</i>	Especificación de requisitos software priorizada ANEXO : ERS [Sección C]
<i>Aprobación y priorización de requerimientos</i>	
<i>Documentación de casos</i>	Documentación UML [Sección C]



<i>de uso no especulativos</i>	
<i>Diseñar Programa de Actividades Específicas (PAE)</i>	ANEXO : PAE [sección ■■■]

Fuente: Autor

Componentes fase: Diseño y Desarrollo

Tabla 23. Artefactos fase Diseño y Desarrollo. Ciclo: “Talento Humano”

Actividades	Artefactos
<i>Análisis Contextual</i>	Diseño de interacción (DI). ANEXO [Sección C]
<i>Diseño de interacción</i>	Mapa del Sitio (MS). ANEXO [Sección c]
<i>Diseño de Bases de Datos.</i>	Diagrama de estructura de datos (DED). ANEXO [Sección C] Diccionario de Datos (Modificado)
<i>Propuesta Diseño grafico.</i>	Alternativas de Diseño de Interfaz Gráfica (ADIG). Alternativas de Diseño Final (ADF).Plantillas finales (css, psd, gif, html y js).
<i>Aprobación</i>	Story board de las plantillas Finales (SB). ANEXO
<i>Desarrollo de la Aplicación</i>	SW
<i>Pruebas en tiempo de desarrollo</i>	No aplica

Fuente: Autor

Componentes fase: Pruebas

Tabla 24. Artefactos fase Pruebas. Ciclo: “Talento Humano”

Actividades	Artefactos utilizados
<i>Pruebas de requisitos</i>	Especificación de requisitos sección C (utilizado)
<i>Pruebas de unidad para el módulo</i>	No aplica
<i>Pruebas de aceptación</i>	Historial de Uso sección C (utilizado)

Fuente: Autor

Componentes fase: Documentación

Tabla 25. Artefactos fase Documentación. Ciclo: “Talento Humano”

Actividades	Artefactos
-------------	------------



<i>Creación de Manuales de usuario</i>	<i>Manual General de Usuarios sección E por usuario</i>
<i>Creación manuales equipo de desarrollo</i>	<i>Manual Técnico y Diccionario de Datos sección E.</i>

Fuente: Autor

Observaciones en cuanto a la metodología en base al ciclo
 Los artefactos asociados a la “Propuesta de diseño gráfico” y “Aprobación” hacen parte de la última versión y la cual fue utilizada. No es necesario realizar estas actividades en todos los ciclos, a menos que por cada componente se tengan cambios drásticos en la presentación. En ocasiones estas actividades se limitan al diseño de botones y logotipos especiales para el componente.

6 **CIDU Correo**



Pase a carecer de análisis previo asociado a éste componente se acuerda trabajar en funciones de tareas básicas que presta un servicio de correo electrónico convencional incluyendo, Envío, consulta y clasificación de correos así como el manejo de archivos (adjuntar, ver, descargar). Adicional a esto administrar permisos para envío de correos.

Componentes fase: Especulación

Tabla 26. Artefactos fase Especulación. Ciclo: “Correo”

Actividades	Artefactos
<i>Documentación de casos de uso especulativos</i>	
<i>Listar Requerimientos</i>	Especificación de requisitos software priorizada ANEXO : ERS [Sección D]
<i>Aprobación y priorización de requerimientos</i>	
<i>Documentación de casos de uso no especulativos</i>	Documentación UML [Sección D]
<i>Diseñar Programa de Actividades Específicas (PAE)</i>	ANEXO : PAE [sección]

Fuente: Autor



Componentes fase: Diseño y Desarrollo

Tabla 27. Artefactos fase Diseño y Desarrollo. Ciclo: “Talento Correo”

Actividades	Artefactos
<i>Análisis Contextual</i>	Diseño de interacción (DI). ANEXO [Sección D]
<i>Diseño de interacción</i>	Mapa del Sitio (MS). ANEXO [Sección D]
<i>Diseño de Bases de Datos.</i>	Diagrama de estructura de datos (DED). ANEXO [Sección D] Diccionario de Datos (Modificado)
<i>Propuesta Diseño grafico.</i>	Alternativas de Diseño de Interfaz Gráfica (ADIG). Alternativas de Diseño Final (ADF).Plantillas finales (css, psd, gif, html y js).
<i>Aprobación</i>	Story board de las plantillas Finales (SB). ANEXO
<i>Desarrollo de la Aplicación</i>	SW
<i>Pruebas en tiempo de desarrollo</i>	No aplica

Fuente: Autor

Componentes fase: Pruebas

Tabla 28. Artefactos fase Pruebas. Ciclo: “Correo”

Actividades	Artefactos utilizados
<i>Pruebas de requisitos</i>	Especificación de requisitos sección D (utilizado)
<i>Pruebas de unidad para el módulo</i>	No aplica
<i>Pruebas de aceptación</i>	Historial de Uso sección D (utilizado)

Fuente: Autor

Componentes fase: Documentación

Tabla 29. Artefactos fase Documentación. Ciclo: “Correo”

Actividades	Artefactos
<i>Creación de Manuales de usuario</i>	<i>Manual General de Usuarios sección E por usuario</i>
<i>Creación manuales equipo de desarrollo</i>	<i>Manual Técnico y Diccionario de Datos sección E.</i>

Fuente: Autor



Observaciones en cuanto a la metodología en base al ciclo
Si la secuencia de actividades asociadas a un procedimiento es sencilla se puede omitir la creación de caracterizaciones. Sin embargo es importante tomar nota de detalles básicos asociados.

Sistematización del subproceso Gestión de Información Académica

Análisis Previo: Resumen de actividades

Identificar los procedimientos asociados al subproceso:

“Gestión de Carga Académica”

“Control de Notas”

Recolectar documentos fuente requeridos para la creación del sistema

Formato básico de Boletín(DF001)

Aplicar Métodos de Indagación

Entrevistas

Análisis Contextual

De ser necesario caracterizar procedimientos

Como no se cuenta con documentos de caracterización para estos procedimientos, se crearon borradores de dichos documentos en base a la información recolectada acerca de la descripción y pasos a desarrollar en cada caso.

Aplicar Métodos de Categorización de contenidos

Se aplicó la técnica de card sorting tipo abierto, a fin de identificar si se puede presentar solo una categoría o más:

Categorías: se entregaron tres papeles de categoría en blanco para que dieran el nombre tendrían.

Tarjetas: Logro Indicador, Registrar Notas y Fallas, Áreas, Asignaturas, Observaciones, Proyectos, Horario y tres papeles en blanco para tarjetas sugeridas.

En base al análisis cualitativo de resultados se tiene que el personal identificó tres posibles categorías pertenecientes al subproceso cuyos nombres corresponden a “Asignaturas”, “Horarios” y “Notas”, las tarjetas fueron asociadas rápidamente de acuerdo a las actividades que corresponden a cada procedimiento y como observación se presenta tres tarjeta adicionales Nuevo Horario, Consultar Horario y Permisos de Carga. Se presentó titubeo sobre la tarjeta de observaciones pero se acordó que pertenezca a la categoría asignaturas.



Se acordaron los perfiles de usuario para el subproceso, en éste caso se tienen: Administrador (Rectora, Secretaria, Coordinador Académico), estudiante, Acudiente y docente.

En base al método “Investigación Contextual”, se obtuvo información adicional del usuario respecto a expectativas, alternativas de desarrollo a los procedimientos, opiniones y experiencias, lo cual permitirá evaluar el contexto en el que se desarrollan los procedimientos y permitir mejoras de las actividades pertinentes.

En base a Card Sorting y la técnica de Story Board se tiene información suficiente para continuar la creación de los artefactos asociados a la fase actual (Análisis Previo) y pasar a la siguiente.

7 **CIDU Académico**



Componentes fase: Especulación

Tabla 30. Artefactos fase Especulación. Ciclo: “Académico”

Actividades	Artefactos
<i>Documentación de casos de uso especulativos</i>	
<i>Listar Requerimientos Aprobación y priorización de requerimientos</i>	Especificación de requisitos software priorizada ANEXO : ERS [Sección E]
<i>Documentación de casos de uso no especulativos</i>	Documentación UML [Sección E]
<i>Diseñar Programa de Actividades Específicas (PAE)</i>	ANEXO : PAE [sección ■■■]

Fuente: Autor

Componentes fase: Diseño y Desarrollo

Tabla 31. Artefactos fase Diseño y Desarrollo. Ciclo: “Académico”

Actividades	Artefactos
<i>Análisis Contextual</i>	Diseño de interacción (DI). ANEXO [Sección E]
<i>Diseño de interacción</i>	Mapa del Sitio (MS). ANEXO [Sección E]
<i>Diseño de Bases de Datos.</i>	Diagrama de estructura de datos (DED). ANEXO [Sección E]



	Diccionario de Datos (Modificado)
<i>Propuesta Diseño grafico.</i>	Alternativas de Diseño de Interfaz Gráfica (ADIG). Alternativas de Diseño Final (ADF).Plantillas finales (css, psd, gif, html y js).
<i>Aprobación</i>	Story board de las plantillas Finales (SB). ANEXO
<i>Desarrollo de la Aplicación</i>	SW
<i>Pruebas en tiempo de desarrollo</i>	No aplica

Fuente: Autor

Componentes fase: Pruebas

Tabla 32. Artefactos fase Pruebas. Ciclo: "Académico"

Actividades	Artefactos utilizados
<i>Pruebas de requisitos</i>	Especificación de requisitos sección E (utilizado)
<i>Pruebas de unidad para el módulo</i>	No aplica
<i>Pruebas de aceptación</i>	Historial de Uso sección E (utilizado)
<i>Test en voz alta</i>	Video (Generado)

Fuente: Autor

Componentes fase: Documentación

Tabla 33. Artefactos fase Documentación. Ciclo: "Académico"

Actividades	Artefactos
<i>Creación de Manuales de usuario</i>	<i>Manual General de Usuarios sección E por usuario</i>
<i>Creación manuales equipo de desarrollo</i>	<i>Manual Técnico y Diccionario de Datos sección E.</i>

Fuente: Autor

Observaciones en cuanto a la metodología en base al ciclo
El tiempo adicional requerido en la fase de diseño del history card muestra sus frutos en la fase de pruebas al presentar una planeación general de pruebas de funcionalidad. Si recordamos, la función de éste artefacto recae en considerar las posibles rutas a cada paso, el seguir la secuencia mostrada resulta ser un método de prueba de funcionalidades muy acertado. Estas pruebas pueden presentarse a la par con el desarrollo y posteriormente por otro participante del equipo. En el caso de funcionalidades similares como los mantenimientos generales, una secuencia de pasos puede servir para todos.

El tiempo pronosticado para el desarrollo de cada funcionalidad completa según PAE, por lo general no es calculado acertadamente. A medida que se desarrollan



los ciclos pueden darse mejores pronósticos, el cumplimiento del mismo por actividad no es de obligatorio cumplimiento pero deben presentarse esfuerzos para mejorar cada vez.



V. Prototipo Desarrollado.

A grandes rasgos y en base a funcionalidades generales por módulo se presenta SOFGESCOL un sistema que le permitirá:

Módulo Matrícula

- Informar al usuario sobre fechas importantes relacionadas con los trámites de matrícula, así como de los documentos requeridos y otros datos generales.
- Permite matricular estudiantes nuevos y/o antiguos, en consideración de cada uno de sus etapas (pre-matrícula, aprobación de pre-matrícula, etc.)
- Permite administrar el estado de un estudiante en la institución (Activo, Transitorio, Retirado) y consultar el historial de matriculas de años anteriores.
- Permite registrar y actualizar información relacionada con el estudiante, así como la de sus padres y acudiente.
- Permite consultar informes disponibles los cuales corresponden a los sugeridos por el usuario

Módulo de Pagos

- Mantenimiento de la información y valores relacionados con costos de matrícula y conceptos de pago.
- Permitir el registro y control de pagos en base a recibos generados y cancelados
- Consultar estado de cuenta de los estudiantes por año, mostrando detalles sobre recibos asociados a cada pago.
- Generar recibos en base a datos afines, (tales como fecha de pago, concepto, etc.) y seleccionando el o los estudiantes indicados.



Módulo Talento Humano

- Consultar datos personales del estudiante a partir del uso de un directorio, los cuales podrán ser buscados por curso o por estudiante.
- Registrar, consultar, actualizar y/o eliminar información relacionada con los docentes y demás personal que haga parte de la institución.
- Consultar información básica de los docentes relacionados con un curso y/o estudiante, incluyendo horario de atención disponible.
- Permitir el mantenimiento general de proyectos, tareas y objetivos relacionados con personas asociadas a la institución.

Módulo Correo

- Servir de medio informativo básico para mejorar los procesos comunicativos entre los usuarios de la institución y facilitar el seguimiento de conducto regular.
- Permite desarrollar tareas básicas de un servicio de correo electrónico convencional (enviar, recibir, eliminar, destacar correo, etc).

Módulo Académico

- Crear, modificar, consultar y eliminar horarios por curso-materia-docente en base al programa académico del grado.
- Permitir la consulta de horario para estudiantes según el curso al que pertenecen así como en el caso de los docentes visualizar su horario de materias programadas según la sección escogida (primaria, bachillerato, etc.)
- Consultar el historial de notas por alumno durante los años cursados en la institución.
- Consultar el estado académico actual, observaciones, control de asistencia y logros- indicadores por periodo del año lectivo por estudiante
- Consultar el estado académico actual por estudiante, así observaciones e información general relacionada con indicadores y/o logros.



- Permitir a los docentes el registro y actualización de notas, así como las observaciones generadas y las inasistencias de cada estudiante a cargo en el desarrollo de una asignatura específica.
- Permite seleccionar la forma de calificación y la nomenclatura requerida, para la generación de boletines.

Aspectos generales de Seguridad

- Maneja encriptación de contraseñas.
- Restricción de actividades generales por perfil= Se pueden administrar los permisos en cada página para crear, modificar, eliminar o consultar, permitido o no permitido el acceso a las opciones de menú
- Control de Acceso= El menú es dinámico, se pueden generar y asignar permisos de acceso a la aplicación en base al mantenimiento de las entidades "Servicios" y "Perfiles".
- Adicional a esto, mediante el manejo de variables de sesión se concederá acceso único a páginas que al usuario de perfil particular le competen.

Requerimientos Mínimos:

El sistema funcionará correctamente con las siguientes características del servidor:

- Servidor de Bases de datos: PostgreSQL 9
- Servidor: Cualquiera que cumpla con las características para implantar sistemas JEE5.
- Generador de Reportes: IREPORT
- Sistema Operativo del servidor: Los sistemas en los que se probará la aplicación son Windows XP, 7 y Linux Debian.

Arquitectura de Desarrollo:

- Lenguaje: Java Enterprise Edition 5.
- Persistencia: Toplink.
- Componentes de la Interfaz: IceFaces.

Aplicación de Pruebas Finales

Pruebas de Integración



Al concluir todos los ciclos del proyecto, se realiza una comprobación de cada uno de los vínculos asociados por perfil-módulo, para verificar su correcto comportamiento, lo cual resulta una tarea sencilla al contar con el menú dinámico cuyas opciones y permisos se administran desde la BD.

De Sistema

Este tipo de pruebas están basado en el cumplimiento de tres principios básicos
El cumplimiento de todos los requerimientos funcionales, en consideración del producto completo

El correcto funcionamiento y rendimiento en las interfaces hardware, software y de operador.

Adecuación de la documentación de usuarios.

Pues bien, uno a uno los tres principios han sido cumplidos en consideración a las pruebas de requerimientos practicadas previamente y que dan el aval a requerimientos funcionales existentes y que funcionan correctamente. Segundo, la puesta en marcha de la aplicación en la actualidad sobre un servidor lo cual permite verificar el correcto funcionamiento en general. Por último, la documentación generada al final de cada ciclo permite dar plena instrucción tanto a usuarios de la aplicación como a futuros desarrolladores de la aplicación, sobre tareas a desarrollar que le competen, ayuda que podrán utilizar en cuanto la necesiten en físico o desde la aplicación en la sección de ayuda.

De Aceptación

Son aquellas que requieren de la participación de los clientes y definen la aceptación del sistema como su nombre lo indica. Su principal objetivo consiste en verificar que el producto software final está listo para ser implementado y utilizado.

Pasos básicos, para generar pruebas de aceptación

Listar una serie de pasos a seguir y el resultado de los mismos en cada caso.
Nota: El resultado generado en cada paso puede ser un mensaje, un cambio en la BD (registro nuevo, borrado o actualizado), una imagen particular en pantalla, etc.
Identificar la serie de pasos que terminan la secuencia y la serie de pasos que permiten continuar.

Definir los caminos posibles y para cada caso identificar los valores requeridos y sus resultados finales.

Eliminar secuencias redundantes.



Estos pasos son reflejo del Historial de uso (**Anexo G**), el cual fue elaborado previamente y el cual será de gran ayuda en éste punto. En base a él se tuvo un seguimiento por funcionalidad, el paso a paso permitió comprobar el funcionamiento adecuado en la mayor parte de funcionalidades. Algunas de ellas tuvieron que ser modificadas o necesitaron de elementos de pantalla adicionales para recrear lo planteado en el historial de uso, pero gracias al documento se ahorro tiempo.

Historial de Uso (Otra presentación del diagrama de interacción)

Es considerado una recopilación que incluye un resumen por funcionalidad que debe tener el sistema.

Protocolo de "Pensar en Voz Alta".

Esta prueba de usabilidad ha sido seleccionada en función de resultados que permitan que la herramienta mejore, en base a lo descrito por el usuario durante la prueba.

Durante una prueba de usabilidad el usuario debe expresar en voz alta sus impresiones, sentimientos, pensamientos y opiniones sobre lo que está sucediendo, cómo, porqué y como le parecería a él que debiera ser el comportamiento del sistema.

Metodología: Protocolo de pensar en voz alta

Se recogen los comentarios y observaciones del usuario, básicamente en cuanto a la forma en la que este "vive" su experiencia con el interfaz, las situaciones que le son incómodas y comportamientos no-naturales y su modelo mental en la interacción con el site.

Es una técnica puramente subjetiva y cualitativa. Es importante tanto lo expresado por el usuario, como la forma en que lo hace. Probablemente será bueno recoger algunas de sus expresiones en modificaciones futuras del interfaz.

Estos pasos son reflejo del Historial de uso (**Anexo Y**), el cual fue elaborado previamente y será de gran ayuda en éste punto. En base a él se tuvo un seguimiento por funcionalidad, el paso a paso permitió comprobar el funcionamiento adecuado en la mayor parte de funcionalidades. Algunas de ellas tuvieron que ser modificadas o necesitaron de elementos de pantalla adicionales para recrear lo planteado en el historial de uso, pero gracias al documento se ahorro tiempo.



Historial de Uso (Otra presentación del diagrama de interacción)

Es considerado una recopilación que incluye un resumen por funcionalidad que debe tener el sistema.

Protocolo de "Pensar en Voz Alta".

El aporte del usuario se ve reflejado en el uso de ésta prueba, el permitirle manifestar críticas constructivas ayudará a la realización de mejoras para la herramienta. Claro está es necesario definir ciertos límites en cuanto a cambios solicitados que toman más tiempo del debido, pues aunque el cliente siempre tiene la razón las sugerencias pueden tornarse exageradas.

Lista de comprobaciones

Se define un test mediante el cual podrá evaluarse la aplicación en base a diversos aspectos tanto de apariencia y presentación de datos como despliegue de información. En general los resultados fueron acertados debido al uso de métodos orientados a usabilidad desde etapas tempranas al desarrollo del proyecto. El test utilizado para aplicar la prueba corresponde al Anexo Z.



Conclusiones y Recomendaciones.

CONCLUSIONES

- ✓ Es acertado trabajar en función del desarrollo de ciclos completos (CIDU) en analogía con ciclos Gestalt lo cual permite centrar la atención en componentes particulares completos, prestando especial atención a aspectos asociados a usabilidad lo cual optimiza los resultados de cada proyecto.
- ✓ El uso del prototipo promueve el acercamiento a TI, de momento se realiza una presentación del mismo en espera de la respuesta de quienes revisen la propuesta en línea que se tiene como versión de prueba.
- ✓ Al seleccionar herramientas a utilizar es necesario considerar el tiempo de uso a fin de optar por aquellas versiones trial que permiten reducir costos si se da el caso. A su vez se debe tratar de utilizar herramientas preferiblemente conocidas o involucrar en el proyecto a personas que tengan buen manejo de las mismas.
- ✓ La metodología ofrece mejores tiempos en función de la disponibilidad del equipo a cargo, la solución de problemas en equipo y el apoyo del cliente.
- ✓ El prototipo ha sido desarrollado en consideración de la normas establecidas por la Secretaría de Educación, y en base a ello se asume que la herramienta puede ser implantada tanto en instituciones públicas como privadas, pues el funcionamiento básico del proceso sistematizado es semejante.
- ✓ Las herramientas software al igual que las herramientas tangibles deben ser diseñadas en función de las necesidades y expectativas del usuario, de tal manera que el primer con el sistema sean de forma intuitiva y el despliegue de alternativas funcionales ofrecidas éste acorde con la actividad que representa. De ahí la importancia de recurrir a técnicas de usabilidad durante el proceso de desarrollo.
- ✓ El desarrollo de proyectos de grado representan el primer contacto con el mundo profesional real y una oportunidad para poner a prueba y reforzar todo conocimiento adquirido. A su vez, representa un llamado para descubrir talentos ocultos y ampliar los conocimientos adquiridos en base a la experiencia y solución de problemas generados.



- ✓ La relación con el futuro usuario inicialmente suele ser de resistencia a los cambios, por ello es necesario involucrarlo tanto como sea posible y presentarle una visión de la herramienta a desarrollar como un mecanismo para mejorar su desempeño dentro de tareas convencionales. Esto no es tan sencillo como suena, pero en base al uso de CIDU es posible capturar información de calidad requerida y mediante los artefactos generados se puede explicar fácil y claramente la evolución del sistema en base a lo que el usuario conoce y puede proponer.
- ✓ En base a nuestra experiencia durante el proyecto invitamos a la innovación y generación de nuevas metodologías de desarrollo, nuestro campo de desempeño laboral es muy versátil y está siempre a la espera de nuevas soluciones a problemas convencionales que prevalecen.
- ✓ En general los objetivos propuestos para el desarrollo del presente proyecto se han cumplido, ofreciendo dos productos finales que corresponden a una nueva metodología para el desarrollo software llamada CIDU y el desarrollo de un prototipo software con las características básicas esperadas en función de los usuarios y el plus ofrecido por CIDU.
- ✓ Los manuales en línea favorecen la interacción con el sistema y ofrecen información requerida por el usuario en el momento preciso.

RECOMENDACIONES

- ✓ Promover proyectos de grado pro-creación de soluciones informáticas en áreas que carecen de dichos recursos constituye un aporte a la sociedad en función de la ayuda que éstas puedan proporcionar.
- ✓ Para el correcto funcionamiento del sistema es necesario que los usuarios a cargo del manejo de la información, ingrese los datos solicitados en cada caso de manera correcta, a fin de que la información sea fiable y precisa. Pese a validar información el sistema se limita a corroborar tipo de dato en cada caso.
- ✓ Se espera que la herramienta evolucione a partir de la sistematización de otros procesos y la creación de nuevas funcionalidades para el prototipo actual.
- ✓ La metodología diseñada continúa en proceso de prueba y evolución, e irá enriqueciéndose a medida que sea utilizada en base a aclaraciones y



propuestas de futuros usuarios. Así mismo, sus artefactos pueden ser mejorados.

- ✓ Es importante considerar la fase de implantación de las herramientas e incorporarla en una próxima versión de la metodología.
- ✓ El prototipo propone a largo plazo y en cumplimiento de una primer etapa un prototipo final que puede representar una herramienta completa que se pueda ofrecer como alternativa tecnológica para las instituciones educativas.



Bibliografía.

[A] General

- PRESSMAN, *Roger*. Ingeniería del Software – Un enfoque práctico, Quinta Edición.
- BOOCH, *Grady*, JACOBSON, *Ivar*, RUMBAUGH, *James*. El Lenguaje Unificado de Modelado, Segunda Edición.
- DAWSON, *Christian W*, Universidad de Valencia España. El Proyecto de Fin de Carrera de Ingeniería Informática. Primera Edición.
- BOEHM, *Barry*. Software Engineering Economics. New Jersey: Prentice Hall, 1981.
- OBERG, *Roger*; PROBASCO, *Leslee* y ERICSSON, *Maria*. RATIONAL SOFTWARE. Applying requirements management with use cases [online]. Rational Software Corporation, 2003 [consultada febrero de 2010]. Disponible en internet en: <http://www.uml.org.cn/RequirementProject/pdf/apprmuc.pdf>
- SAWYER, *Peter* y KONTOYA, *Gerald*. Software requirements. En: Software Engineering Book Of Knowledge (SWEBOK), capítulo 2. p.18. Disponible en Internet: <http://www.swebok.org>
- THAYER, *Richard* y DORFAM, *Merlin*. Software Requirements Engineering. 2 ed. Los Alamitos, California: IEEE Computer Science Press, 2000. p. 1.
- WIEGERS, *Karl*. Software Requirements. 2 ed. Washington: Microsoft Press, 2003
- DÁVILA, *Nicolás Davyt*. Ingeniería de requerimientos: Una guía para extraer, analizar, especificar y validar los requerimientos de un proyecto. Artículo Técnico (Licenciado en Análisis de Sistemas). Universidad ORT, Uruguay. Facultad de Ingeniería. Área de Sistemas.



[B] Relacionada con la metodología

Sitios Web

- <http://www.sidar.org/recur/desdi/traduc/es/visitable/Herramientas.htm>
[Citado por última vez el 22 de agosto de 2011]
Información a encontrar: Definición y descripción de los principales métodos de usabilidad
- <http://www.sidar.org/recur/desdi/traduc/es/visitable/quees/dcu.htm>
[Citado por última vez el 22 de agosto de 2011]
Información a encontrar: Principios del diseño centrado en el usuario
- <http://www.proyectosagiles.org/beneficios-de-scrum>
[Citado por última vez el 22 de agosto de 2011]
Información a encontrar: Descripción general de los beneficios de SCRUM
- <http://www.sparxsystems.com.ar/products/ea.html>
[Citado por última vez el 22 de agosto de 2011]
Información a encontrar: Propiedades de la herramienta Enterprise Architect
- <http://www.nosolousabilidad.com/articulos/diagramacion.htm>
[Citado por última vez el 22 de agosto de 2011]
Información a encontrar: Usabilidad
- <http://www.proyectosagiles.org/lista-requisitos-priorizada-product-backlog>
[Citado por última vez el 22 de agosto de 2011]
Información a encontrar: Tips para priorizar un listado de requisitos
- <http://www.scribd.com/doc/17800502/Buenas-o-Mejoras-Practicas>
[Citado por última vez el 22 de agosto de 2011]
Información a encontrar: Nociones básicas sobre buenas prácticas
- <http://www.jjg.net/ia/visvocab/spanish.html>
[Citado por última vez el 22 de agosto de 2011]
Información a encontrar: Vocabulario visual para diagramas sencillos



[C] Para el desarrollo de la aplicación

Sitios Web

- <http://download.oracle.com/javaee/5/tutorial/doc/>
[Citado por última vez el 22 de agosto de 2011]
Información a encontrar: Tutorial General del Estándar JEE5
- <http://java.sun.com/developer/technicalArticles/J2EE/jpa/>
[Citado por última vez el 22 de agosto de 2011]
Información a encontrar: Información detallada sobre JPA (Java Persistence API)
- <http://www.oracle.com/technetwork/articles/javaee/jpa-137156.html>
[Citado por última vez el 22 de agosto de 2011]
Información a encontrar: Nociones generales sobre JPA (Java Persistence API)
- [http://en.wikipedia.org/wiki/Persistence_\(computer_science\)](http://en.wikipedia.org/wiki/Persistence_(computer_science))
[Citado por última vez el 22 de agosto de 2011]
Información a encontrar: Pautas generales sobre la persistencia en aplicaciones.

[D] Otros proyectos consultados

- Título: Enfoque metodológico para el Diseño de Interfaces de Usuario
Autor: Javier Mauricio Martínez Gómez
Director: Msc. Esperanza Aguilar de Flórez
Tesis para Maestría en Área Informática (Universidad Industrial de Santander)



Anexo A – Metodologías en las que se basa CIDU

1. SCRUM

1.1. Definición

Scrum es una metodología que permite la auto-gestión del equipo de trabajo, donde sus integrantes deciden como y cuanto tiempo tardará el desarrollo de sus tareas. Cada integrante trabaja en función de un objetivo común y gracias al seguimiento de dichas tareas, se puede ver claramente cuanto avanza el desarrollo del producto final.

Sprints (Iteraciones)

Cada proyecto con SCRUM, se ejecuta en bloques temporales y fijos a los cuales se les llama Sprints. Cada una de estas iteraciones tiene características especiales como:

- ❖ Duración
 - ◆ Cortos (1 a 4 semanas)
 - ◆ Preferiblemente constante, para lograr ritmo
 - ◆ Timeboxed
- ❖ Contenido
 - ◆ Definido en el inicio del sprint, no se cambia
 - ◆ Incluye diseño, codificación, prueba y todo lo que se defina como necesario para tener un ítem “Completo”
 - ◆ El resultado de cada ítem, representa un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite

1.2. Scrum Framework

<i>Roles</i>	Product Owner, ScrumMaster, Team, usuarios
<i>Reuniones</i>	Sprint Planning, Sprint Review, Sprint Retrospective, & Daily Scrum Meeting
<i>Componentes</i>	Product Backlog, Sprint Backlog, and Burndown Chart

1.3. Roles SCRUM

Cada persona que pertenece al equipo SCRUM, tiene establecido un rol y por consiguiente una serie de actividades a realizar y de las cuales se hará responsable, entre los roles a identificar están:

ROL	Descripción	Actividades
<i>Product Owner</i>	Es la persona responsable del proyecto, está encargada de definir	<ul style="list-style-type: none"> ▪ Define la funcionalidad del producto



	y establecer las prioridades del producto que se está desarrollando, haciendo un balance del valor que aportan respecto a su costo, a fin de maximizar la utilidad de lo que se desarrolla y el retorno de la inversión.	<ul style="list-style-type: none"> ▪ Decide la fecha y contenido del release ▪ Es responsable de la rentabilidad del producto (ROI) ▪ Prioriza la funcionalidad de acuerdo a valor de negocio ▪ Modifica las funcionalidades en cada iteración, según se necesite ▪ Acepta o rechaza el resultado del trabajo
<i>Scrum Master</i>	Es la persona encargada de asegurar el seguimiento de la metodología, de guiar las reuniones y ayudar al equipo ante cualquier problema que pueda aparecer. Su responsabilidad es entre otras, reducir el impacto ante presiones externas.	<ul style="list-style-type: none"> ▪ Sirve de interface desde y hacia el equipo ▪ Responsable del cumplimiento de las prácticas y valores de Scrum ▪ Remueve impedimentos ▪ Se encarga que el grupo funcione completamente y sea productivo ▪ Facilita cooperación entre todos los roles ▪ Protege al grupo de interferencias externas
<i>Scrum Team</i>	Cada miembro del equipo de trabajo responsable de implementar la funcionalidad o funcionalidades elegidas por el Product Owner.	<ul style="list-style-type: none"> ▪ Normalmente entre 5-10 personas ▪ Multidisciplinario <ul style="list-style-type: none"> ▪ Programadores, testers, diseñadores gráficos ... ▪ Los miembros deberían ser full-time <ul style="list-style-type: none"> ▪ Algunas excepciones comunes: DBAs, Administradores ▪ Los equipos se auto-organizan <ul style="list-style-type: none"> ▪ Idealmente sin títulos ▪ Los miembros cambian de grupo al finalizar los sprints
<i>Usuarios o Cliente</i>	Son los beneficiarios finales del producto, y son quienes viendo los progresos, pueden aportar ideas, sugerencias o necesidades.	<ul style="list-style-type: none"> ▪

1.4. Reuniones SCRUM

Reunión	Descripción
<u>Planificación de la iteración</u>	El primer día de la iteración se realiza la reunión de planificación de



<p>(Sprint Planning Meeting)</p>	<p>la iteración. Tiene dos partes:</p> <ol style="list-style-type: none"> 1. Selección de requisitos (4 horas máximo). El cliente presenta al <u>equipo</u> la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita. 2. Planificación de la iteración (4 horas máximo). El equipo elabora la <u>lista de tareas de la iteración</u> necesarias para desarrollar los requisitos a que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se autoasignan las tareas.
<p>Sprint Review</p> <p><u>Ejecución de la iteración</u></p>	<p>Cada día el equipo realiza una <u>reunión de sincronización</u> (15 minutos máximo). Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido. En la reunión cada miembro del equipo responde a tres preguntas:</p> <ul style="list-style-type: none"> • ¿Qué he hecho desde la última reunión de sincronización? • ¿Qué voy a hacer a partir de este momento? • ¿Qué impedimentos tengo o voy a tener? <p>Durante la iteración el <u>Facilitador</u> se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad.</p> <ul style="list-style-type: none"> • Elimina los obstáculos que el equipo no puede resolver por sí mismo. <p>Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad.</p>
<p>Sprint Retrospective</p> <p>Inspección y adaptación</p>	<p>El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:</p> <ol style="list-style-type: none"> 4. Demostración (4 horas máximo). El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto. 5. Retrospectiva (4 horas máximo). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. Para ello, el equipo se reúne y discute sobre lo que le gustaría:



	<ul style="list-style-type: none"> ♦ Comenzar a hacer ♦ Dejar de hacer ♦ Continuar haciendo
Daily Scrum Meeting	<p>Durante cada día que del Sprint Backlog se efectuará un Daily Scrum Meeting, se trata de una reunión operativa, informal y ágil, de un máximo de 30 minutos, en la que se le hace 3 preguntas a cada integrante del equipo.</p> <ul style="list-style-type: none"> ❖ Qué tareas ha realizado desde la última reunión (que he hecho). ❖ Sobre qué va a trabajar en el día actual (que voy a hacer hoy). ❖ Identificación de obstáculos o riesgos que impiden o pueden impedir el normal avance (que ayuda necesito). El Scrum Master, debe eliminar aquí cualquier obstáculo que encuentre.

1.5. Componentes SCRUM

⌘ *Product Backlog*

Actúa como plan del proyecto, es una lista priorizada (por el product owner Product Manager, Marketing, Cliente Interno,...) de requerimientos, y que combinan

- ❖ Funcionalidad (user stories): “Cómo usuario quiero poder buscar y reemplazar para evitar tareas repetitivas”
- ❖ Tareas pendientes: “Mejorar el manejo de excepciones”

Nota: Al comienzo de cada Sprint, ésta lista es repriorizada

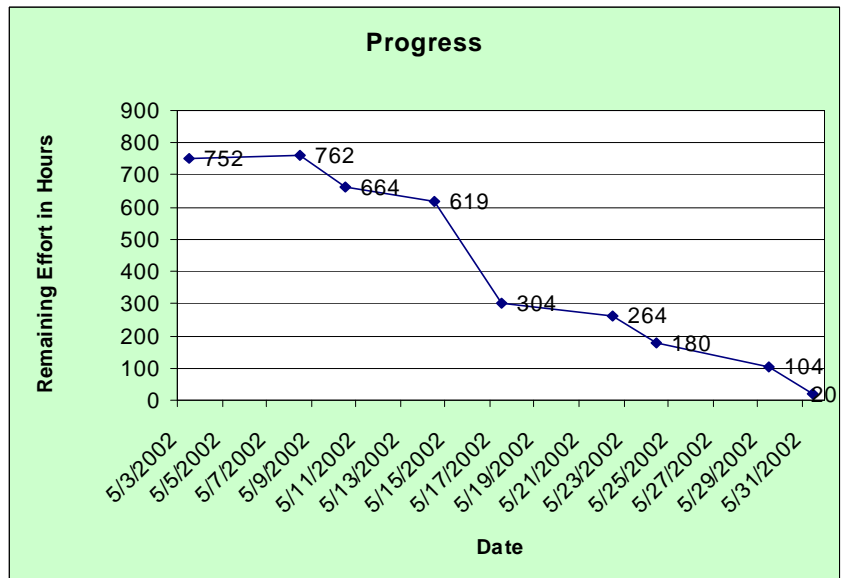
Ejemplo de Product Backlog



	Item #	Description	Est	By
Very High				
	1	Finish database versioning	16	KH
	2	Get rid of unneeded shared Java in database	8	KH
		- Add licensing	-	-
	3	Concurrent user licensing	16	TG
	4	Demo / Eval licensing	16	TG
		Analysis Manager		
	5	File formats we support are out of date	160	TG
	6	Round-trip Analyses	250	MC
High				
		- Enforce unique names	-	-
	7	In main application	24	KH
	8	In import	24	AM
		- Admin Program	-	-
	9	Delete users	4	JM
		- Analysis Manager	-	-
	10	When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab	8	TG
		- Query	-	-
	11	Support for wildcards when searching	16	T&A
	12	Sorting of number attributes to handle negative numbers	16	T&A
	13	Horizontal scrolling	12	T&A
		- Population Genetics	-	-
	14	Frequency Manager	400	T&M
	15	Query Tool	400	T&M
	16	Additional Editors (which ones)	240	T&M
	17	Study Variable Manager	240	T&M
	18	Haplotypes	320	T&M
	19	Add icons for v1.1 or 2.0	-	-
		- Pedigree Manager	-	-
	20	Validate Derived kindred	4	KH
Medium				
		- Explorer	-	-
	21	Launch tab synchronization (only show queries/analyses for logged in users)	8	T&A
	22	Delete settings (?)	4	T&A

⌘ Sprint Burndown Chart

Es un gráfico Vs tiempo, que representa el avance del proyecto

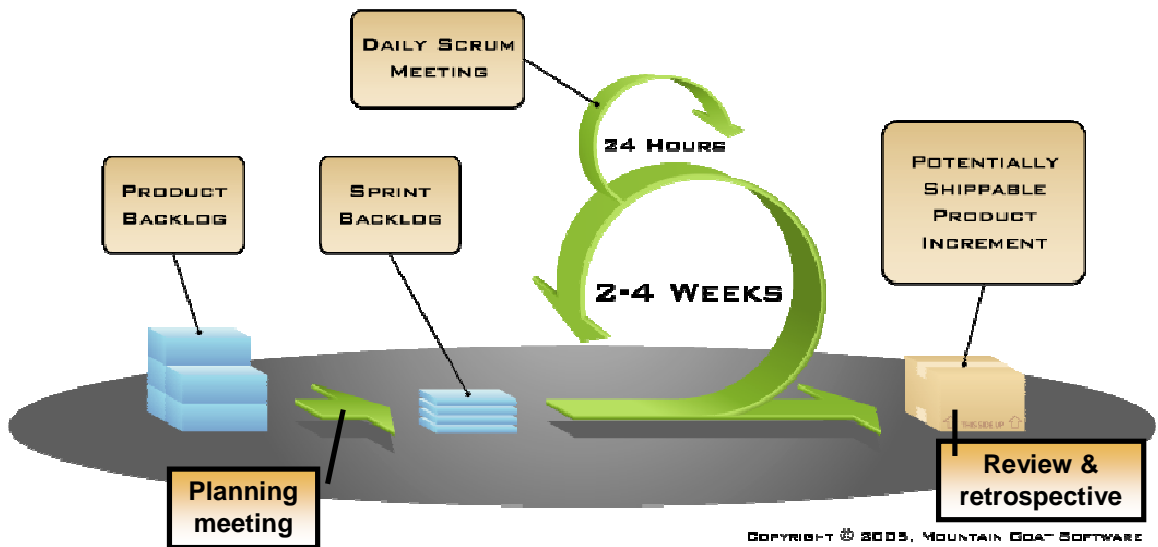


⌘ Sprint Backlog



Corresponde a un grupo particular de tareas del Product Backlog, que se desarrollan en 2 ó 4 semanas y cuya duración será estimada antes de iniciar. En cuanto se inicie, no podrá ser alterado o modificado, solo hasta culminar dichas tareas, podrán realizarse las alteraciones o modificaciones pertinentes formando un nuevo Sprint Backlog.

Dinámica de SCRUM



2. ESPIRAL

2.1. Definición

La metodología de desarrollo planteada para el presente proyecto se denomina *Proceso De Desarrollo en Espiral*, la cual es un derivado del *Proceso de Desarrollo Unificado*.

La gestión ágil de proyectos tiene como objetivos dar garantías a las cuatro demandas principales de la industria en la que se ha generado: Valor, reducción del tiempo de desarrollo, agilidad y fiabilidad.

El diseño de la aplicación se centra en los casos de uso determinados en el proceso de especulación y los casos de uso definidos por la organización.

La gestión ágil, a diferencia de la tradicional, refleja las preferencias declaradas por el manifiesto ágil:

- ♦ La capacidad de respuesta al cambio, sobre el seguimiento de un plan.
- ♦ Los Productos que funcionan frente a especificaciones y documentaciones innecesarias.
- ♦ La colaboración con el cliente frente a la negociación contractual.
- ♦ A las personas y su interacción por encima de los procesos y las



herramientas.

2.2. Fases

El desarrollo ágil parte de la visión, del concepto general del producto o servicio, y sobre ella el equipo va desarrollando pequeños incrementos en la dirección apuntada por la visión, y en el orden de prioridad que necesita el negocio del cliente. Los ciclos breves de desarrollo, se denominan iteraciones y se realizan hasta que se decide no evolucionar más el producto generado. Este esquema de desarrollo está formado por cinco fases:

- 1) *Análisis Previo*: Esta es la fase conceptual en la que se crea la visión del producto o servicio que quiere obtener. Se decide y selecciona al equipo de personas que lo llevarán a cabo. Partir sin una visión determinada genera esfuerzo baldío. Del mismo modo que en términos de empresa, la visión es un factor crítico para el éxito del proyecto. Se necesita tener la visión de lo que se quiere, y conocer el alcance del proyecto.
- 2) *Especulación*: Una vez que se sabe qué es lo que hay que desarrollar, el equipo especula y construye hipótesis sobre la información de la visión, que siempre se es muy general e insuficiente para determinar las implicaciones de un desarrollo (requisitos, diseño, costos). En esta fase se determinan las limitaciones impuestas por el entorno de negocio (costes y agendas principalmente) y se especula la primera aproximación de lo que se puede producir. La gestión ágil investiga y desarrolla tomando como partida la visión del producto. Durante el desarrollo confronta la realidad de lo que va obteniendo. Su valor, posibilidades y la situación de negocio del entorno en cada momento. La fase de especulación se repite en cada iteración del desarrollo, y teniendo como referencia la visión y el alcance del proyecto consiste en:
 - Desarrollo / revisión de los requisitos generales del producto.
 - Desarrollo de una lista con las funcionalidades esperadas.
 - Construcción de un plan de entrega: Fechas en las que se necesitan las versiones, hitos e iteraciones del desarrollo.
 - En función de las características del modelo de gestión y del proyecto puede incluir también una estrategia o planes para la gestión de riesgos.

Si las exigencias de cumplimiento de la organización lo requieren, también se generan información administrativa y financiera.

- 3) *Diseño y Desarrollo*: Se desarrollan las funcionalidades de un incremento del producto, que han sido determinadas por el equipo en la fase anterior
- 4) *Pruebas*: El equipo y los usuarios revisan las funcionalidades construidas



hasta ese momento. Trabajan y operan con el producto real para determinar su alineación y dirección con el objetivo.

- 5) *Documentación:* Al llegar a la fecha de entrega de una versión de producto (fijada en la fase de concepto y revisada en las diferentes fases de especulación), se obtiene el producto esperado. Posiblemente éste seguirá en el mercado, y si se emplea gestión ágil es presumible que se trata de un producto que necesita versiones y mejoras frecuentes para no quedar obsoleto. No quiere decir necesariamente que se ha terminado el proyecto. Lo que se denomina mantenimiento supondrá la continuidad del proyecto en ciclos incrementales hacia la siguiente versión para ir acercándose a la visión del producto, que también es posible que vaya evolucionando con el tiempo conforme cambia el entorno tecnológico.
- 6) *Cierre:* Esta etapa solo se cumple una vez en el desarrollo del proyecto cuando están completas todas las iteraciones necesarias. Constituye la implementación y entrega formal del producto terminado.

3. Métodos de Usabilidad

3.1. Métodos de Indagación

También son conocidos como métodos contextuales, ya que la información básica e inicial surge precisamente del entorno del usuario por ello, estos métodos presentan múltiples formas de aproximación al usuario incluyendo generación de ideas y creatividad al observar al usuario desarrollando sus tareas.

3.2. Métodos de Prototipado y/o categorización de Contenidos

La aplicación de la técnica de prototipado va a ser fundamental en el desarrollo e implementación de los métodos para la inspección y test de un producto, dado que, habitualmente, no será el producto final lo que se someta a los diversos experimentos, sino un prototipo del mismo con unas determinadas características, en virtud de las cuales se enfocarán los métodos en una dirección concreta. En definitiva, se trata de una cuestión indispensable, presente incluso en ciertos métodos contextuales (¡la versión previa de un producto puede constituir una forma de prototipo para la nueva generación!). Por otra parte, los métodos de categorización que se citan, y después su conexión con el amplio y complejo mundo de la Calidad, tan sólo recuerdan la necesidad de una planificación meticulosa que nos ha obligado a hablar muchas veces de etapas de pre-diseño. Más concretamente, la Categorización por Tarjetas, sugerirá casi de inmediato al lector cómo establecer el contenido de los menús de una aplicación informática.

- Prototipado



- Según la funcionalidad reproducida
- Según la fidelidad de la reproducción de la interfaz
- Otras técnicas de prototipado
- Prototipado Rápido
- Prototipado por Vídeo
- Métodos de Categorización

3.3. Métodos de Inspección

Se van a establecer cinco aproximaciones básicas:

- Inspecciones
- Evaluación Heurística
- Paseos Cognitivos
- Listas de Comprobación
- Otras perspectivas

3.4. Métodos de Test

Se puede conseguir una aproximación apropiada a las necesidades del usuario, se pueden contemplar la existencia de múltiples contextos de trabajo, se puede, también, hacer del usuario parte del proceso de diseño en una fase de evaluación o inspección, pero ¿Existe algún método para detectar el origen de los problemas de usabilidad?, ¿Es posible plantear un experimento con un objetivo particular (y no la medida en que un determinado producto sea usable)?.

El test de usabilidad va a desarrollar experimentos para obtener información específica acerca de un diseño. Así, antes de pasar a las técnicas, conviene hacer primero un recorrido por los Conceptos Generales acerca del Test de Usabilidad.

- Protocolos de Expresión del Usuario
- Realización de medidas
- Variantes del Test de Usabilidad clásico



Anexo B – Principios de Usabilidad

Principio	Definición	medición	Como se mide	Método
legibilidad	Interpretación correcta de la información	Cuantitativa, nominal	Se puede medir determinando por porcentajes: De acuerdo a la cantidad de información.	Entrevistas
Eficacia	Que el usuario pueda cumplir con la tarea	cuantitativa	Medir la intención de los usuarios, al interactuar con un portafolio de productos.	Se realizara una aproximación contextual individual por medio de entrevistas al usuario objetivo.
ley de fitts	Pasos efectivos planeados para realizar la tarea	Cuantitativa, nominal	Medir la cantidad de pasos que los usuarios usa n para ejecutar las tareas o acciones dentro de un portafolio de productos, y comparar con nuestro producto.	entrevista directa-observación
Consistencia	Toda la aplicación debe ser consistente en contenido, forma y distribución. (unificación de actividades)	Nominal Tipo de Variable: Cualitativa	Medir los criterios que permiten que el usuario, comprenda y ejecute correctamente el funcionamiento de un portafolio de productos desarrollado en Flash.	Se realizara una Inspección de características.
Autonomía	El usuario debe poseer el control sobre la herramienta	intervalar	Medir la orientación de usuarios, durante el proceso de interacción con un portafolio virtual de productos, teniendo en cuenta las dificultades y habilidades que este presenta.	Se realizara una observación en el análisis de tareas, estudio de tiempos y movimientos al usuario objetivo. Presentación de datos: Tablas



Principio	Definición	medición	Como se mide	Método
Accesibilidad	Crear interfaces que le ayuden al usuario a comprender a donde va, a donde puede ir, y como llegar a ese lugar.	Ordinal	Medir la navegabilidad de los usuarios, durante el proceso de interacción con un portafolio virtual de productos, analizando los comportamientos y apreciaciones de los usuarios.	Se realizara una observación en el análisis de tareas, estudio de tiempos y movimientos al usuario objetivo.
visibilidad	Todas las herramientas deben estar a disposición del usuario.	Intervalar Tipo de Variable: Cualitativa	Medir los factores de compatibilidad entre el modelo mental y conceptual del usuario final	Por observación y entrevista
Anticipación	El producto le ofrezca al usuario lo que necesita	Razón Tipo de Variable: Cualitativa	Medir el grado de anticipación que posee el portafolio	preguntando cual es la tarea siguiente a que desearía realizar el usuario, antes de ejecutarla
Apreciación	Es una medida de las percepciones, opiniones, sentimientos y actitudes generadas en el Usuario por el portafolio.	Cualitativa, razón	Medir el grado de aceptación del portafolio, por parte de los usuarios.	Se realizara un test de usabilidad



Anexo C – Card Sorting

'Card Sorting' u ordenación de tarjetas es una técnica de categorización de contenidos centrada en el usuario, que permite mediante la observación a los usuarios durante la prueba, organizar y clasificar la información en un sitio Web. Sirve de ayuda para la toma de decisiones en la etapa de diseño conceptual (card sorting abierto) y también para evaluar una organización concreta de categorías en etapas de evaluación de usabilidad (card sorting cerrado).

Tipos de 'card sorting'

Abierto: Tiene por principal objetivo descubrir el tipo de clasificación de categorías que sería más conveniente utilizar. El usuario podrá agrupar las categorías en el número de conjuntos que considere necesarios.

Cerrado: Tiene por objeto verificar si una clasificación de información es familiar y comprensible al usuario. Para ello, él podrá agrupar las categorías solo en los grupos que han sido predefinidos y etiquetados.

Pasos a realizar

1. *Seleccionar participantes para la prueba:* En primer lugar identificamos y reunimos al personal que hará parte de la prueba. Al seleccionar estas personas, hay que tener en cuenta que correspondan a perfil y características afines a usuarios promedio de la futura aplicación. El grupo que hará parte de la prueba debe estar compuesto por al menos 5 integrantes;
2. *Establecer el tiempo de duración de la prueba:* El tiempo de duración dependerá de la cantidad de categorías a ordenar. En promedio se debe emplear 30 segundos por categoría.
3. *Inducción a los participantes:* Antes de empezar, hay que aclarar el objetivo de la prueba y que no se trata de una evaluación a los participantes, sino de un aporte. Se indicará además cual será el criterio para la agrupación de categorías: por similitud.
4. *Entregar material para la prueba:* A cada participante se le entregará una serie de tarjetas (previamente desordenadas) con varias categorías
5. *Dar inicio a la prueba:* En cuanto cada persona tenga a la mano el material necesario se dará inicio a la prueba. Si se trata de 'card sorting' de tipo abierto, cabe aclarar que en cuanto terminen la prueba se debe asignar nombres a los grupos creados.

Nota: Existen, así mismo, aplicaciones software con las que poder llevar a cabo la prueba de forma virtual, como [EZSort](#) o [WebSort](#) .



Análisis cualitativo de los resultados

Suele utilizarse cuando el número de participantes en la prueba y el número de categorías es pequeño. Consiste en observar a cada usuario durante la prueba en forma individual y tomar nota de cómo organiza las tarjetas, que problemas presenta al realizar la tarea, que categorías logra asociar rápida e intuitivamente y con cuáles presentó mayor dificultad, qué preguntas hace durante la prueba, etc.

A continuación se realiza un análisis "a ojo" de los grupos creados por los participantes, observando qué categorías guardan más relación con qué categorías según el modelo mental del usuario.

Análisis cuantitativo de los resultados

Si se trata de un gran número de categorías y usuarios, este tipo de análisis es el más conveniente ya que facilita la interpretación de los datos obtenidos gracias a un manejo estadístico de los mismos.

1. Se crea una tabla de co-ocurrencias en una hoja de cálculo. En ésta, con tantas filas y columnas como número de categorías diferentes, se indica el número de veces que cada par de categorías han sido colocadas en un mismo grupo, dando como resultado una matriz simétrica como la siguiente.

	Guía de estudio	Des. Materiales	Foro	Web Personal	Temario	Glosario
Guía de estudio		30	1	1	47	49
Des. Materiales	30		6	9	40	36
Foro	1	6		33	0	1
Web Personal	1	9	33		0	0
Temario	47	40	0	0		55
Glosario	49	36	1	0	55	

Después, sobre esta tabla de co-ocurrencias se aplican algoritmos de reducción dimensional, como son los algoritmos de clustering y de escalamiento multidimensional (MDS), cuya función es simplificar las relaciones entre categorías a un número de dimensiones fácilmente interpretables por inspección visual (2D ó 3D).

Para la aplicación de estos algoritmos hay que disponer de un paquete de software estadístico que lo permita, como SPSS o Statistica.



Como se puede apreciar en las figuras, el resultado de la aplicación del algoritmo de clustering es un dendrograma, y en el MDS una representación geométrica de las categorías, distanciadas según la propia di-similaridad entre éstas.

***** H I E R A R C H I C A L C L U S T E R A N A L Y S I S *****

Dendrogram using Ward Method

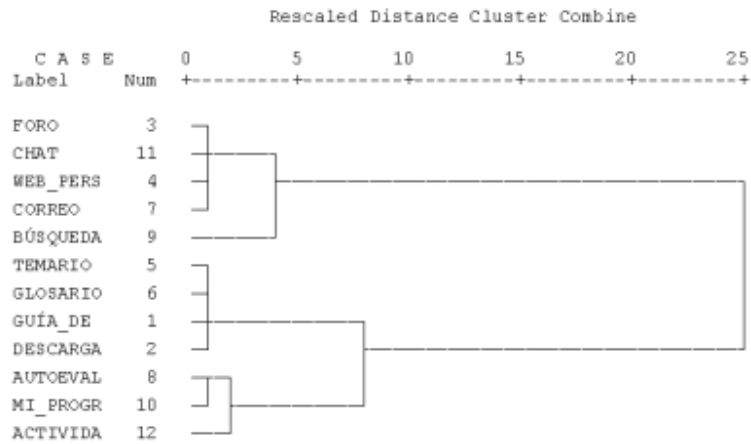


ILUSTRACIÓN 3 DENDOGRAMA (CLUSTERING)

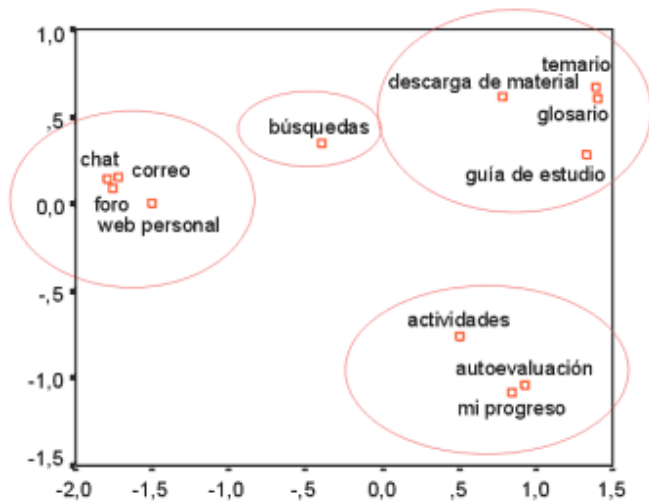


ILUSTRACIÓN 4 MDS (ESCALAMIENTO MULTIDIMENSIONAL)



Anexo D – Vocabulario Visual

Tanto la arquitectura de Información como el diseño de interacción, deben ser mostrados a diferentes personas que se encuentran involucradas en el proyecto y la forma más conveniente para presentarlos es mediante el uso de diagramas lo cual permitirá el desarrollo de requerimientos detallados de navegación e interfaz para cada página. Servirá además como base a la creación de documentos más detallados y específicos a las necesidades de cada perfil de usuario. Algunos requerimientos clave de un vocabulario visual para arquitectura de información y diseño de interacción son:

- **Compatible con pizarra blanca:** El vocabulario visual manejado debe ser tan simple que los diagramas puedan ser dibujados rápidamente a mano, de ser así sin restar claridad al diagrama.
- **Independiente de herramienta:** Dicho vocabulario debe estar diseñado de forma que no requiera de software especializado para construir diagramas.
- **Pequeño y auto-contenido:** Se ha de trabajar con el menor número posible de elementos, conservando relación uno-a-uno entre conceptos y símbolos, de manera que se pueda aprender y aplicar en forma rápida por cualquiera de las personas que tengan acceso a ellos sin importar su nivel de conocimiento.

Trasfondo conceptual

Arquitectura de información y diseño de interacción son dos caras de la misma moneda. (Ver "Los Elementos de la Experiencia de Usuario" para definiciones de los términos como son usados aquí.) Los diagramas de sitios contemporáneos inevitablemente involucran ambas caras. Pero para cada una, los objetivos del diagrama son levemente diferentes.

En ambos casos, el diagrama se enfoca en lo que llamamos la **macro-estructura**, entregando sólo detalle suficiente para permitir a los miembros del equipo ver la "gran foto". La tarea del arquitecto es determinar el nivel apropiado de detalle para lograr este objetivo. El detalle específico a nivel de página o **micro-estructura**, es detallado en otros documentos de los cuales el arquitecto puede no ser directamente responsable de desarrollar.

Diagrama de Arquitectura de Información

Hace énfasis en la estructura conceptual y organización del contenido



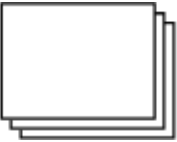

Diagrama de Diseño de Interacción

Hace énfasis en como el usuario se desplaza a través de tareas definidas estableciendo los respectivos pasos a seguir para ello

Nota: Ninguno de ellos debe incluir detalles de interfaz que desvíen la atención o el principal propósito de los mismos



Elementos simples

Figura	Descripción
	<p>Página</p> <p>Unidad básica de presentación, no (necesariamente) una unidad de implementación -- una página puede representar múltiples archivos HTML (como en una interfaz con frames) o unidades múltiples de código (como en includes en el servidor -- SSI -- o una implementación manejada por bases de datos).</p>
	<p>Archivo</p> <p>Parcela de datos sin propiedades de navegación, e incluso pueden ser utilizados fuera de un ambiente de navegación Web o browser (tales como archivos de video, archivos independientes como PDFs ó ejecutables).</p>
	<p>Pila de páginas</p> <p>Sirve para indicar un grupo de páginas funcionalmente idénticas, cuyas propiedades de navegación son inmateriales a la macro-estructura del sitio</p>
	<p>Pila de documentos</p> <p>Grupo de documentos que reciben tratamiento de navegación idéntico y pueden ser clasificadas como una entidad única (tal como una colección de juegos descargables o una librería de manuales de instrucciones en PDF).</p>
	<p>Etiquetas</p> <p>Son usadas en páginas y archivos para identificarlos. Éstas no tienen la necesidad de ser una correlación con designaciones como el elemento <TITLE> HTML o nombres de documentos, pero deben ser únicos para cada página o documento en el diagrama. Identificadores numéricos únicos y designaciones de tipo también entregan una buena forma de llevar el rastro todas las paginas y documentos en un diagrama.</p>



Creando relaciones

Conector

Representa relaciones conceptuales e interpretadas como relaciones de navegación. En el caso de la arquitectura de información, estas relaciones están comúnmente reflejadas a través de la organización jerárquica de páginas en árboles. Sin embargo, esto de ninguna manera es obligatorio ni (en algunos casos) recomendable.

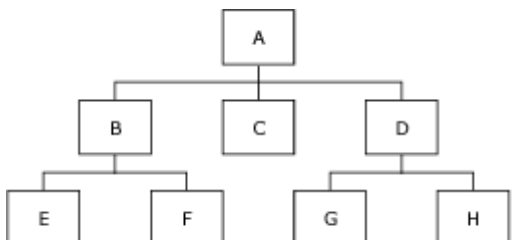


ILUSTRACIÓN 5 ESTRUCTURA SIMPLE DE ÁRBOL

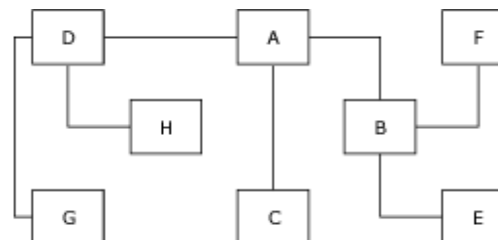


ILUSTRACIÓN 6 LA MISMA ESTRUCTURA DIAGRAMADA DE FORMA DIFERENTE

Flecha

Para crear el diagrama de diseño de Interacción es necesario que las relaciones indiquen a su vez dirección, a fin de mostrar como el usuario va a moverse a través del sistema por área particular. Para ello usamos flecha que nos representen dicha dirección en términos de corriente arriba y corriente abajo, de acuerdo a la posición relativa de movimiento. Las flechas indicarán la dirección en la que probablemente el usuario quiera ir, sin embargo no implica que el usuario no pueda moverse en dirección opuesta.



ILUSTRACIÓN 7 FLECHA INDICA MOVIMIENTO CORRIENTE ABAJO HACIA EL FIN DE LA TAREA

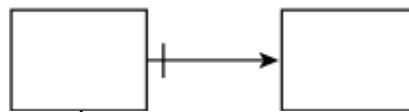


ILUSTRACIÓN 8 BARRA CRUZADA INDICA QUE EL MOVIMIENTO CORRIENTE ARRIBA NO ESTÁ PERMITIDO

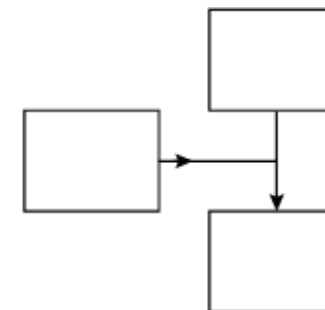

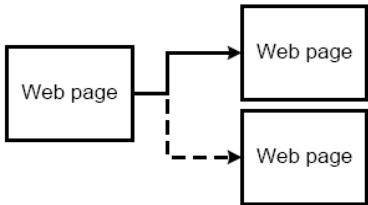



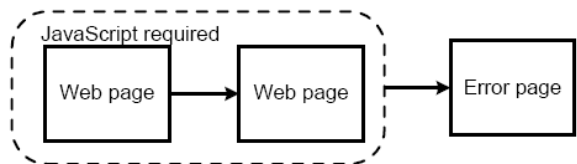

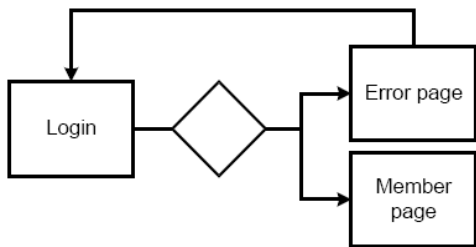



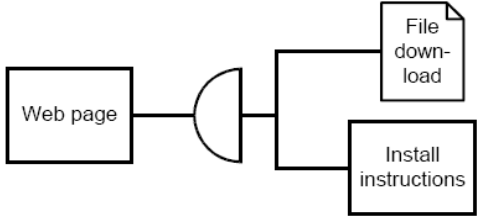

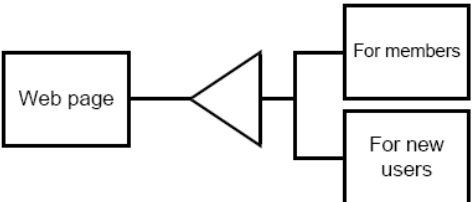

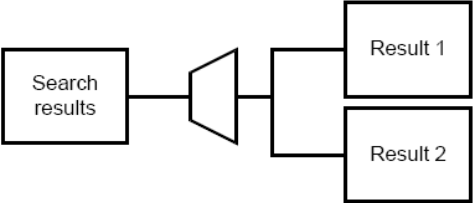

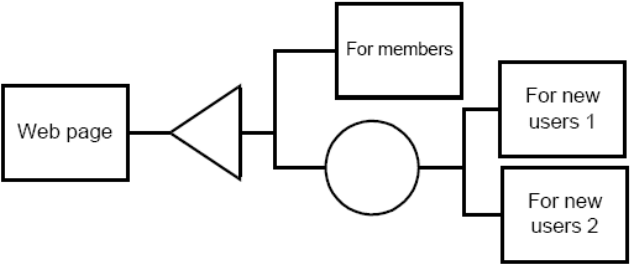
ILUSTRACIÓN 9 FLECHAS MÚLTIPLES CLARIFICAN LA DIRECCIÓN



Elementos de Lógica Condicional

Figura	Descripción	Representación
	<p>Flecha Condicional: Si la condición se cumple, el camino se hace disponible. Si no, no existe camino.</p>	
	<p>Un conector condicional (representado por una línea cortada) es usado cuando un camino puede ser o no ser presentado al usuario dependiendo de si una o más condiciones son cumplidas.</p>	
	<p>Área Condicional: Cuando una o más condiciones aplican a un grupo de páginas, esas páginas son encerradas en un área condicional</p>	
	<p>Punto de decisión: En base a una decisión tomada por el usuario se pueden presentar varios resultados y el sistema selecciona el resultado a presentar (Probablemente el ejemplo más común de esto es manejo de errores en el envío de formularios).</p>	



	<p>Un conjunto concurrente: es usado en casos cuando una acción del usuario genera resultados múltiples simultáneos (tal como abrir una ventana pop-up mientras una página se carga en la ventana principal, o mostrar una página mientras un documento es descargado). Como las flechas, los conjuntos concurrentes tienen dirección. Elementos corrientes arriba se conectan al lado curvo; elementos corrientes abajo se conectan al lado plano.</p>	
	<p>Rama condicional Es usada si el sistema es el encargado una de las opciones mutuamente exclusivas a ser presentadas al usuario. Nota: Los elementos corrientes arriba se conectan a un punto del triángulo; los elementos corrientes abajo se conectan al lado opuesto.</p>	
	<p>Selector condicional Funciona de manera muy similar a la rama condicional, con una diferencia importante: con el selector, los varios caminos corrientes abajo <i>no</i> son mutuamente exclusivos, cualquier número de caminos que satisfagan las condiciones pueden ser presentados al usuario. Ejemplo: Motor de búsqueda.</p>	
	<p>El Racimo puede aparecer corriente abajo desde una rama condicional o un selector condicional. Funciona parecido a la rama condicional, pero presentando más de un camino al usuario.</p>	

Puntos de Continuación



Algunas arquitecturas son tan complejas que no pueden ser capturadas en un diagrama que incluya todo, por esto es necesario separar los diagramas en secciones y para ello utilizamos **Puntos de continuidad ([])** a fin de unir los espacios vacíos entre las páginas.

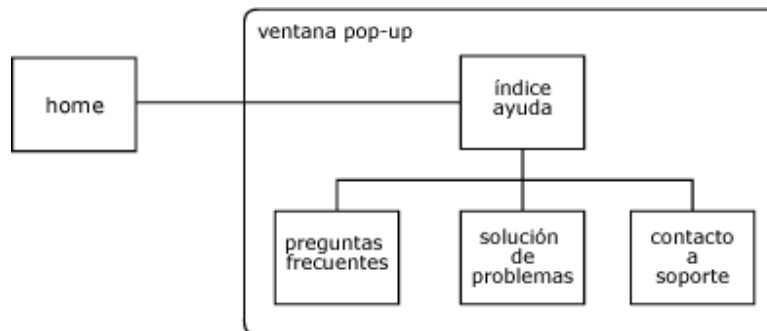
ILUSTRACIÓN 10 UN PUNTO "CONTINÚA HACIA" REFERENCIA AL LECTOR HACIA OTRO DIAGRAMA

ILUSTRACIÓN 11 UN PUNTO "CONTINÚA DESDE", RETOMANDO DESDE DONDE SALIMOS DE 6A

Área

Este elemento (un rectángulo de esquinas redondeadas) es usado para identificar un grupo de páginas que comparten uno o más atributos comunes (tales como aparecer en una ventana pop-up, o tener un tratamiento único de diseño). Se deben usar

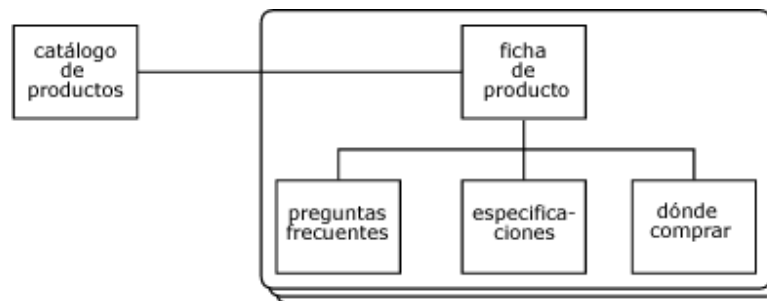
ILUSTRACIÓN 12 UN EJEMPLO DE USO DE UN ÁREA PARA REPRESENTAR UNA VENTANA POP-UP



Área Iterativa

En caso de tener varias estructuras básicas repetidas, se puede utilizar un área iterativa (una pila de rectángulos con esquinas redondeadas) que las represente.

ILUSTRACIÓN 13 UN EJEMPLO DE USO DE UN ÁREA ITERATIVA PARA REPRESENTAR UNA ESTRUCTURA REPETIDA EN UN CATÁLOGO DE PRODUCTOS





Componentes Reutilizables

Áreas de Flujo3

Algunos diseños de interacción requieren que una secuencia de pasos (como por ejemplo, un procedimiento de login) aparezca repetidamente en diferentes contextos a través del diseño. A menudo estas secuencias son solo un componente de una o más tareas que el usuario está tratando de lograr. (Esto es análogo al concepto de sub-rutina en programación de ordenadores). Tal área re-utilizable es llamada un **flujo**, y es representada en el diagrama mediante dos elementos:

Referencia de Flujo

Sirve como marcador para el flujo en cada contexto en el cual se repite.

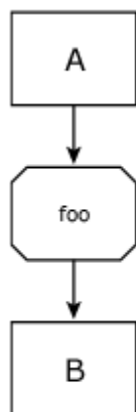


ILUSTRACIÓN 14
UNA REFERENCIA
DE FLUJO SIRVE
TANTO COMO
PUNTO "CONTINÚA
HASTA", COMO
PUNTO "CONTINÚA
DESDE"

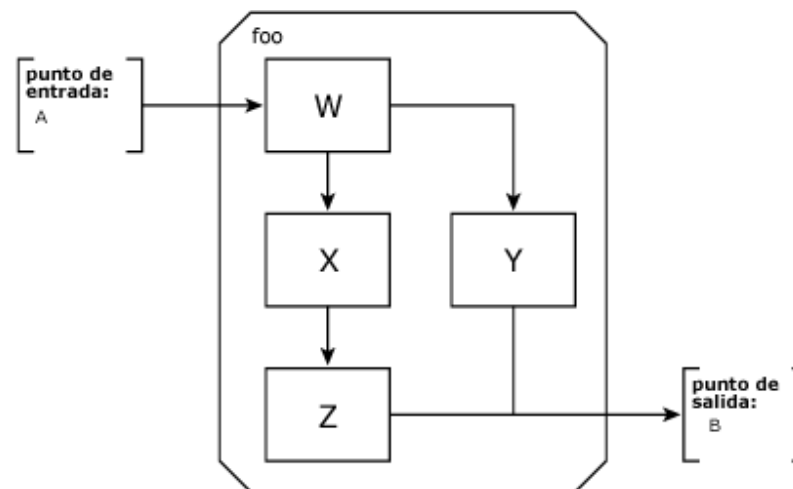
Área de Flujo

Encierra el flujo mismo, en ocasiones requieren el uso de dos tipos de puntos de continuación especiales:

Puntos de entrada:
Éstos son ubicados fuera del área de flujo

Puntos de salida:
dentro del área de flujo, indican que el flujo abarca múltiples diagramas.

ILUSTRACIÓN 15 EL ÁREA DE FLUJO REFERIDA EN 9A





Anexo E – Formato Acta de Reunión (CIDU)

Referencia:

Confidencialidad del Documento:
Público
Tipo de Documento: Interno del Proyecto
Formato General: Acta de Reunión
Código: AR001

NOMBRE DEL PROYECTO | Etapa
SCRUMPIRAL

Logotipo
Proyecto

Fecha: dd/mm/aa

Entidad: Nombre de la empresa cliente

Actividades a desarrollar:

- Descripción Actividad 1
- Descripción Actividad 2
- Descripción Actividad 3

Participantes:

Nombre	Rol	Firma de Asistencia

Documentos entregados

- Nombre documento fuente (DF01)
- Nombre documento fuente (DF02)

Documentos solicitados

- Nombre documento fuente (DF03)

Próxima Reunión	Fecha	día	mes	Año
	Hora	hora	min	Seg

Actividades a desarrollar:

- Descripción Actividad 1
- Descripción Actividad 2
- Descripción Actividad 3





Anexo F – Formato Informe de Reunión (CIDU)

Referencia:

Confidencialidad del Documento:

Público

Tipo de Documento: Interno del Proyecto

Formato General: Informe de Reunión

Código: IR002

NOMBRE DEL PROYECTO | Etapa
SCRUMPIRAL

Logotipo
Proyecto

Fecha: día/mes/año

Entidad: Nombre de la empresa cliente

Autores: Autor 1
Autor2

Nº Acta de Reunión Relacionada

AR001

Objetivos alcanzados

- Descripción objetivo 1
- Descripción Objetivo2

Análisis de Información Recolectada

- Descripción 1
- Descripción 2

<i>Firma Líder del Proyecto</i>



Anexo G – Formato Principios y Técnicas (CIDU)

NOMBRE DEL PROYECTO| Principios y Técnicas

Subproceso: nombre

Autor: nombre

Actualización: dd/mm/aa

Tiempo Máximo:

x Semanas

Fecha de Inicio:

Día/mes/año

Entrega de Resultados:

Día/mes/año

Participantes:

<i>Código</i>	<i>Nombre</i>	<i>Perfil</i>
P01	Participante1	Perfil1
P02	Participante2	Perfil2
P03	Participante3	Perfil3
P04	Participante4	Perfil4

Actividades:

<i>Código</i>	<i>Técnica</i>	<i>Principio</i>	<i>Fecha</i>
A01	Técnica1	Principio1	15/12/2009
A02	Técnica2	Principio2	15/12/2009
A03	Técnica3	Principio3	06/01/2010
A04	Técnica4	Principio4	06/01/2010

Firma Líder del Proyecto



Anexo H – Formato Arquitectura de Información (CIDU)

NOMBRE PROYECTO | Arquitectura de Información

Subproceso: (Nombre subproceso)

Autor: (nombre)

Actualización: (fecha)

Establecimiento de Rutas

Para ello es necesario realizar una Identificación de elementos de Navegación a utilizar:

<i>Elemento de Navegación</i>	<i>Descripción</i>	<i>Elementos que lo conforman</i>	
<i>Navegación Global (Menú Principal)</i>	Permite la integración permanente de todos los módulos de la aplicación, de tal modo que se pueda ir de uno a otro fácilmente.	- Módulo1	- Módulo2
<i>Navegación Histórica</i>	Permite ubicar al usuario del sistema en el contexto general de la página actual.		
<i>Navegación Local 1 (Submenú)</i>	Dentro del módulo seleccionado, debe existir un acceso directo a cada sección, para ello se selecciono la estructura de árbol, donde se muestra cada elemento principal, con posibilidad de ver e ir a las secciones que lo componen (expandiendo y seleccionando una de las opciones).	Módulo1	Opción1 Opción2 Opción3
		Módulo2	Opción4 Opción5 Opción6
<i>Navegación Local 2 (Submenú)</i>		Opción1	Sección1 Sección2 Sección3
		Opción5	Sección4 Sección5
			Sección6



Identificación de perfiles

<i>Perfil</i>	<i>Descripción</i>	<i>Cualidades</i>
Perfil1		
Perfil2		

Identificación de Procesos Internos (Actividades)

Logo módulo1	<i>Módulo</i>	Módulo1
	Descripción	

<i>Sub-módulo</i>	<i>Actividad</i>	<i>Descripción</i>	<i>Perfil</i>
Submodulo1	Actividad1		Perfil1
	Actividad2		Perfil1
Submodulo2	Actividad3		Perfil2
	Actividad4		Perfil1

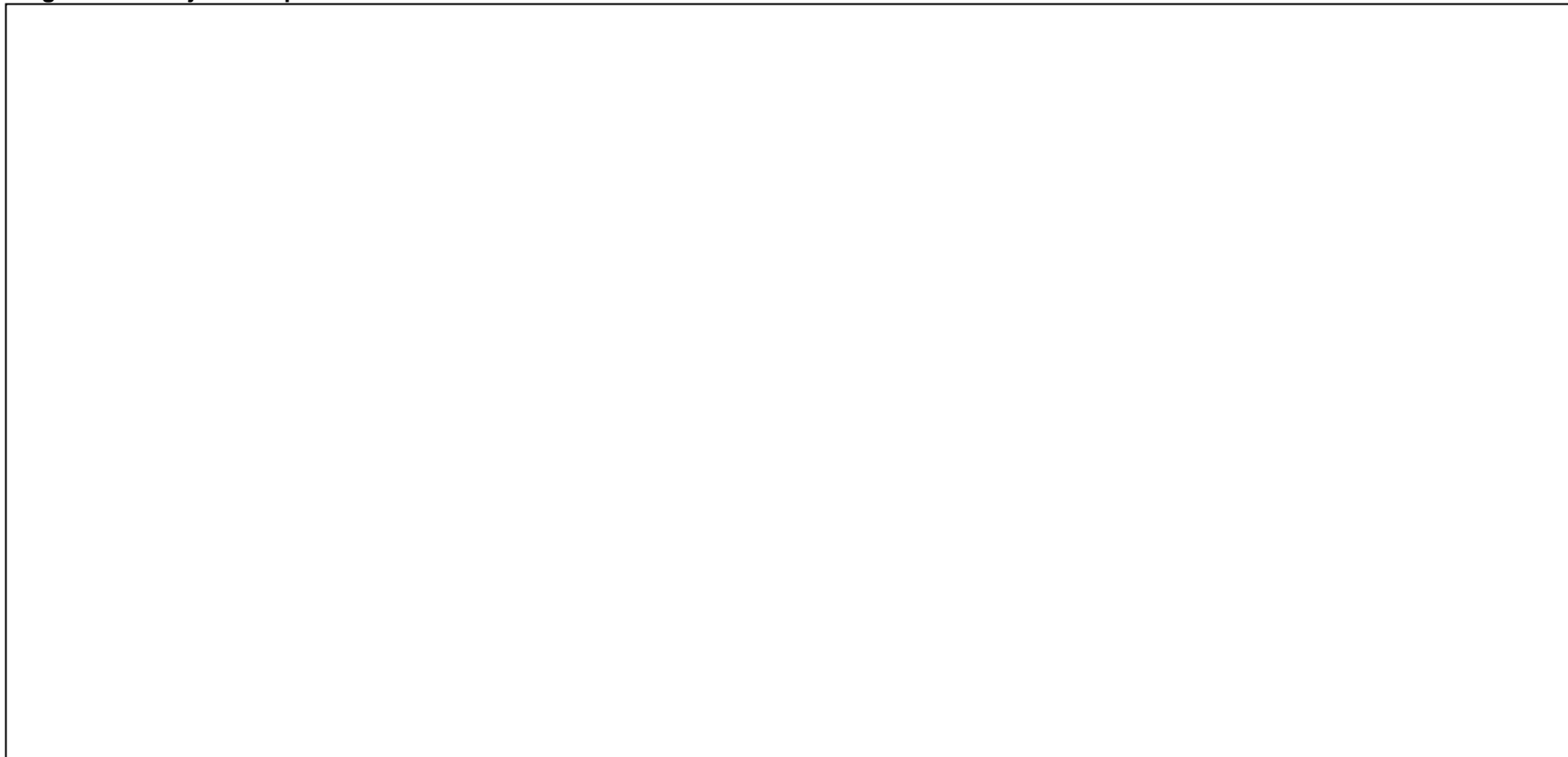
Logo módulo2	<i>Módulo</i>	Módulo2
	Descripción	

<i>Sub-módulo</i>	<i>Actividad</i>	<i>Descripción</i>	<i>Perfil</i>
Submodulo1	Actividad1		Perfil1



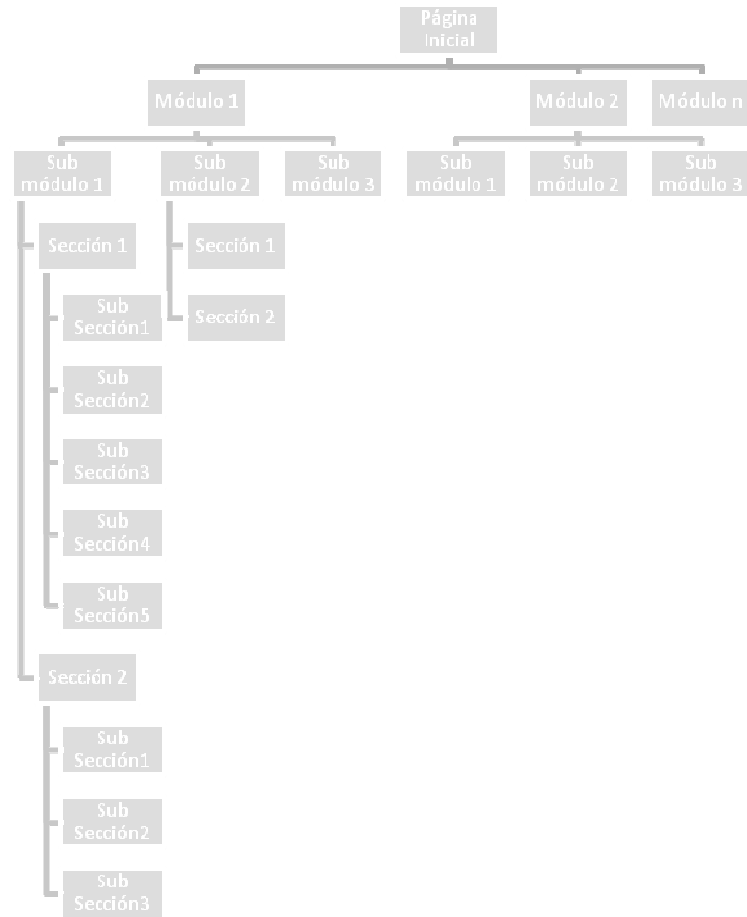
	Actividad2		Perfil2
Submodulo2	Actividad3		Perfil1
	Actividad4		Perfil2

Diagrama de Flujo de Arquitectura Final





Mapa del Sitio



Anexo I – Formato Especificación de Requisitos (CIDU)



**Especificación de
requisitos de software
(ERS)**

Nombre del Proyecto

Revisión N°: 1



Ficha del documento

Fecha	Revisión	Autor	Verificado dep. Calidad
dd/mm/aa	1	Autor 1	Firma
dd/mm/aa	1	Autor 2	Firma

Documento validado por los interesados:

Líder del Proyecto	Propietario del Producto
Nombre	Nombre

Contenido

1 INTRODUCCIÓN

[Inserte aquí el texto]

La introducción de la Especificación de requisitos de software (ERS) debe proporcionar una vista general del presente documento. Debe incluir el objetivo, el alcance, las definiciones y acrónimos, las referencias, y la vista general del ERS.

1.1 Propósito

[Inserte aquí el texto]

- *Propósito del documento*
- *Audiencia a la que va dirigido*

1.2 Ámbito del Sistema

1.3 Definiciones, Acrónimos y Abreviaturas

[Inserte aquí el texto]

Definición de todos los términos, abreviaturas y acrónimos necesarios para interpretar apropiadamente este documento. En ella se pueden indicar referencias a uno o más apéndices, o a otros documentos.

1.4 Referencias



Referencia	Título	Fecha	Autor
Ref. 01	Nombre del libro o documento	aaaa	Autor1
Ref. 02	Nombre del libro o documento	aaaa	Autor2

1.5 Visión General del Documento

Teniendo como base el estándar para la especificación de requisitos de Software IEEE 830 – 1998, fue redactado el presente documento que consta de tres secciones:

La primera sección consta de una introducción, incorporando ciertos elementos que facilitan una visión general de la especificación de recursos del sistema a desarrollar.

En la segunda sección del documento se presenta una descripción general del sistema, a fin de establecer las principales funciones que debe realizar la aplicación, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, todo a grandes rasgos.

Por último, la tercera sección del documento busca definir detalladamente cada uno de los requisitos que debe satisfacer el sistema.

2 Descripción General

2.1 Perspectiva del Producto

[Inserte aquí el texto]

Indicar si es un producto independiente o parte de un sistema mayor. En el caso de tratarse de un producto que forma parte de un sistema mayor, un diagrama que sitúe el producto dentro del sistema e identifique sus conexiones facilita la comprensión.

2.2 Funciones del Producto

[Inserte aquí el texto]

Resumen de las funcionalidades principales que el producto debe realizar, sin entrar en información de detalle.

En ocasiones la información de esta sección puede tomarse de un documento de especificación del sistema de mayor nivel (ej. Requisitos del sistema).

Las funcionalidades deben estar organizadas de manera que el cliente o cualquier interlocutor pueda entenderlo perfectamente. Para ello se pueden utilizar métodos textuales o gráficos.

2.3 Características de los usuarios

Perfil de Usuario	Nombre del perfil
Habilidades	Descripción general de las habilidades que debe tener el usuario
Actividades	Mencionar las actividades relacionadas con el este rol

2.4 Restricciones

[Inserte aquí el texto]



Descripción de aquellas limitaciones a tener en cuenta a la hora de diseñar y desarrollar el sistema, tales como el empleo de determinadas metodologías de desarrollo, lenguajes de programación, normas particulares, restricciones de hardware, de sistema operativo etc.

2.5 Suposiciones y Dependencias

[Inserte aquí el texto]

Descripción de aquellos factores que, si cambian, pueden afectar a los requisitos. Por ejemplo una asunción puede ser que determinado sistema operativo está disponible para el hardware requerido. De hecho, si el sistema operativo no estuviera disponible, la SRS debería modificarse.

2.6 Requisitos Futuros

[Inserte aquí el texto]

Identificación de futuras mejoras al sistema, que podrán analizarse e implementarse en un futuro.

3 Requisitos Específicos

3.1 Interfaces Externas

En fases posteriores, el sistema podrá funcionar con dinero real, estableciendo una interfaz con un banco real.

Interfaces de usuario

Interfaces de hardware

Interfaces de software

Interfaces de comunicación

3.2 Requisitos funcionales

Nombre del Componente: componente1

Requisitos Funcionales del componente1

<i>Código de requisito</i>	ERS_ #
<i>Nombre de requisito</i>	Nombre
<i>Actor</i>	Actor1
<i>Propósito</i>	Descripción de propósito
<i>Resultado</i>	Descripción de resultado

Requisitos no funcionales

Disponibilidad

Portabilidad

3.3 Requisitos de Rendimiento

La aplicación soportará la conexión de varios usuarios de manera simultánea, permitiendo la actualización de información en tiempo real.

3.4 Restricciones de Diseño

No aplican



3.5 Atributos del Sistema

1. El registro de usuarios con perfil de privilegios administrativos se realizará solamente por el administrador del sistema
 2. El acceso al sistema será mediante validación de login y password.
 4. Los estudiantes no podrán modificar los datos relacionados con su información académica
 5. El sistema correrá tanto en plataformas Windows como en Linux.
- Faltan....

3.6. Otros Requisitos

4. Apéndices

5 Índice

Personal involucrado

Nombre	Integrante1
Rol	Rol1
Categoría profesional	
Responsabilidades	Descripción de responsabilidades
Correo Electrónico	Correo
Aprobación	Luz Dary Méndez Uribe
Nombre	
Rol	Planeación, diseño, desarrollo y presentación del proyecto.
Categoría profesional	Estudiante de Ingeniería de Sistemas.
Responsabilidades	Diseño del plan de trabajo de Grado Ejecución del plan de trabajo de Grado
Correo Electrónico	luzdame9@gmail.com
Aprobación	
Nombre	Emiro Muñoz Jerez
Rol	Dirección del trabajo de Grado
Categoría profesional	Ingeniero de Sistemas Profesor Cátedra Escuela de Ingeniería de Sistemas
Responsabilidades	Asesoría para la ejecución del trabajo de Grado Control de la ejecución del proyecto
Correo Electrónico	emiro270273@gmail.com
Aprobación	



Anexo J – Formato Alternativas de Diseño Gráfico (CIDU)

PROYECTO | Alternativas Para Diseño de Interfaz Gráfica

Autor: Nombre del Autor

Actualización: Fecha

1. Alternativas de Color

<p>Pantallazo de plantilla N° PC01</p>	<p>Plantilla N°: PC01</p>
	<p>Color Base 1</p> <input data-bbox="1200 902 1316 981" type="text"/> <p>Color Base 2</p> <input data-bbox="1200 1037 1316 1115" type="text"/>
<p>Pantallazo de plantilla N° PC02</p>	<p>Plantilla N°: PC02</p>
	<p>Color Base 1</p> <input data-bbox="1200 1417 1316 1496" type="text"/> <p>Color Base 2</p> <input data-bbox="1200 1552 1316 1630" type="text"/>



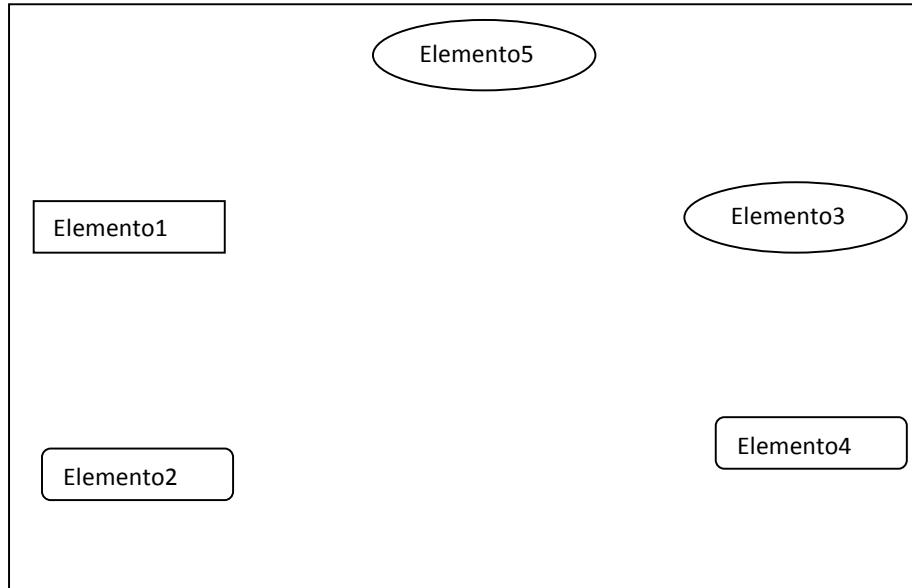
<p>Pantallazo de plantilla N° PC03</p>	Plantilla N° PC03
	<p>Color Base 1</p> <div style="border: 1px solid black; width: 40px; height: 20px; margin: 0 auto;"></div> <p>Color Base 2</p> <div style="border: 1px solid black; width: 40px; height: 20px; margin: 0 auto;"></div>

2. Alternativas de Ubicación

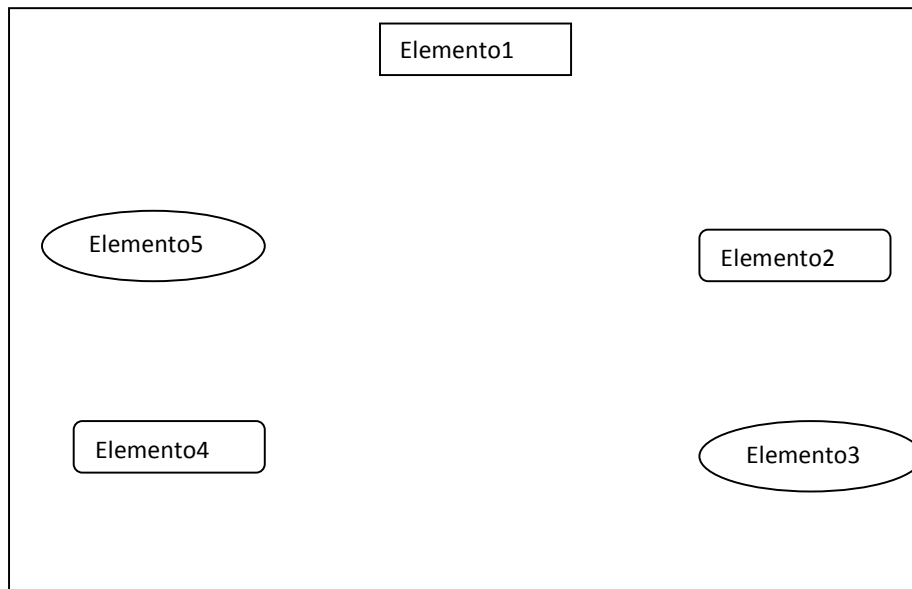
Elemento de pantalla	Descripción
<div style="border: 1px solid black; padding: 2px; display: inline-block;">Elemento1</div>	Descripción general del elemento1
<div style="border: 1px solid black; border-radius: 10px; padding: 2px; display: inline-block;">Elemento2</div>	Descripción general del elemento2
<div style="border: 1px solid black; border-radius: 50%; padding: 2px; display: inline-block;">Elemento3</div>	Descripción general del elemento3
<div style="border: 1px solid black; border-radius: 10px; padding: 2px; display: inline-block;">Elemento4</div>	Descripción general del elemento4
<div style="border: 1px solid black; border-radius: 50%; padding: 2px; display: inline-block;">Elemento5</div>	Descripción general del elemento5



Alternativa de Ubicación N°1



Alternativa de Ubicación N°2





Anexo K – Formato Story Board (CIDU)

<p>IMAGEN GENERAL</p>	<p>Insertar Imagen de la plantilla en general</p>
----------------------------------	---



Plantilla Selectora de Aplicaciones

CAPAS

Insertar Imagen que muestre todas las capas que posee la plantilla

Plantilla Selectora de Aplicaciones

CAPAS GENERALES

Insertar Imagen que muestre las capas generales de la plantilla

Plantilla Selectora de Aplicaciones



Texto
variable
Imágenes

Insertar Imagen que muestre las capas que posee la plantilla, indicando mediante colores, las que corresponden a **Texto variable** e **Imágenes** respectivamente

Plantilla Selectora de Aplicaciones

<i>Nombre</i>	<i>Imagen</i>	<i>Tamaño</i>	<i>Contenido</i>	<i>Texto</i>	<i>Efecto</i>
Imagen1		w px X h px	Imagen	Sin texto	Ninguno
Imagen2		w px X h px	Imagen	Sin texto	
Imagen3		w px X h px	Imagen	Sin texto	
Imagen4		w px X h px	Imagen	Sin texto	
Imagen5		w px X h px	Imagen	Sin texto	
Imagen6		w px X h px	Imagen	Sin texto	



Anexo M – Principios y Técnicas (SOFGESCOL)

SOFGESCOL | Principios y Técnicas

Subproceso: Matricula y Pagos

Autor: Laura Pulido Díaz

Actualización: Diciembre 10 de 2009

Tiempo Máximo:

3 Semanas

Fecha de Inicio:

Diciembre 15 de 2009

Entrega de Resultados:

Enero 6 de 2010

Participantes:

<i>Código</i>	<i>Nombre</i>	<i>Perfil</i>
P01	Luz Dary Méndez Uribe	Analista
P02	Laura Pulido Díaz	Analista
P03	Ana Cecilia Navarro	Pagadora I.E.
P04	Profilia Santos Jaimes	Secretaria I.E.

Actividades:

<i>Código</i>	<i>Técnica</i>	<i>Principio</i>	<i>Fecha</i>
A01	Entrevistas, Caracterización	Eficacia	15/12/2009
A02	Investigación Contextual	Ley de Fitts	15/12/2009
A03	Card Sorting	Autonomía	06/01/2010
A04	Story Board	Accesibilidad	06/01/2010

Firma Líder del Proyecto



SOFGESCOL | Principios y Técnicas

Subproceso: Talento Humano

Autor: Luz Dary Méndez Uribe
Actualización: Diciembre 10 de 2009

Tiempo Máximo:
3 Semanas

Fecha de Inicio:
Diciembre 15 de 2009

Entrega de Resultados:
Enero 6 de 2010

Participantes:

<i>Código</i>	<i>Nombre</i>	<i>Perfil</i>
P01	Luz Dary Méndez Uribe	Analista
P02	Laura Pulido Díaz	Analista
P03	Ana Cecilia Navarro	Pagadora I.E.
P04	Profilia Santos Jaimes	Secretaria I.E.

Actividades:

<i>Código</i>	<i>Técnica</i>	<i>Principio</i>	<i>Fecha</i>
A01	Entrevistas, Caracterización	Eficacia	15/12/2009
A02	Investigación Contextual	Ley de Fitts	15/12/2009
A03	Card Sorting	Autonomía	06/01/2010
A04	Story Board	Accesibilidad	06/01/2010

Firma Líder del Proyecto



SOFGESCOL | Principios y Técnicas

Subproceso: Gestión de Información Académica

Autor: Laura Pulido Díaz
Actualización:

Tiempo Máximo:
3 Semanas

Fecha de Inicio:
Diciembre 15 de 2009

Entrega de Resultados:
Enero 6 de 2010

Participantes:

<i>Código</i>	<i>Nombre</i>	<i>Perfil</i>
P01	Luz Dary Méndez Uribe	Analista
P02	Laura Pulido Díaz	Analista
P05	Ana Dulcelina López	Coordinadora Académica I.E.
P06	Reynaldo Camargo	Coordinador Académico I.E.

Actividades:

<i>Código</i>	<i>Técnica</i>	<i>Principio</i>	<i>Fecha</i>
A01	Entrevistas, Caracterización	Eficacia	15/12/2009
A02	Investigación Contextual	Ley de Fitts	15/12/2009
A03	Card Sorting	Autonomía	06/01/2010
A04	Story Board	Accesibilidad	06/01/2010

Firma Líder del Proyecto

Anexo N – Caracterizaciones (SOFGESCOL)

PROCESO DE REGISTRO ACADÉMICO - SUBPROCESO REGISTRO DE MATRÍCULA		Código: PRA01
PROCEDIMIENTO REGISTRO DE PREMATRÍCULA		Versión: 01
DIAGRAMA DE FLUJO	DESCRIPCIÓN	RESPONSABLE
<pre> graph TD INICIO[INICIO] --> D1{¿El estudiante está preinscrito?} D1 -- Si --> C1[Confirmar información y actualizar datos de ser necesario] D1 -- No --> F1[Formato de pre matrícula] C1 --> F1 F1 --> D2{¿El estudiante está inscrito en otra institución?} D2 -- Si --> C2[Informar en el recibo de pago que debe solicitar su retiro de ella] D2 -- No --> FIN[Fin] C2 --> C3[Recibo de pago por concepto de matrícula] C3 --> FIN </pre>	<ol style="list-style-type: none"> 1. Revisar si se trata de un nuevo estudiante o de uno preinscrito en el sistema. 2. Si el estudiante ya fue inscrito, se confirma la información y de ser necesario se actualizan los datos. 3. Si el estudiante es nuevo se debe diligenciar el formato de pre matrícula 4. Si el estudiantes es nuevo, la secretaria valida que no se encuentre inscrito en otra institución, 5. Si el estudiante nuevo está reportado por otra institución en el SIMAT se informa de esta situación en el recibo de pago para que proceda a realizar el correspondiente cambio. 6. Entonces, se expide el correspondiente recibo de pago por concepto de matrícula, el cual contiene información básica relacionada con el trámite y incluyendo los descuentos pertinentes. 	<p style="text-align: center;">Secretaria</p> <p style="text-align: center;">Secretaria</p> <p style="text-align: center;">Secretaria</p> <p style="text-align: center;">Secretaria</p> <p style="text-align: center;">Secretaria</p> <p style="text-align: center;">Secretaria</p>
		<p style="text-align: center;">Formato de Pre matrícula</p> <p style="text-align: center;">Recibo de pago de matrícula</p>



PROCESO DE REGISTRO ACADÉMICO - SUBPROCESO REGISTRO DE MATRÍCULA		Código: PRA02
PROCEDIMIENTO APROBAR MATRÍCULA		Versión: 01
DIAGRAMA DE FLUJO	DESCRIPCIÓN	RESPONSABLE
<pre> graph TD INICIO([INICIO]) --> DECISION{¿El estudiante está inscrito en otra institución?} DECISION -- Si --> NO_MAT[No podrá matricularse hasta no solucionar ésta situación.] NO_MAT --> DECISION DECISION -- No --> VERIFICAR[Verificar entrega de documentos y confirmar pago] VERIFICAR --> APROBAR[Tomar nota de observaciones y aprobar matrícula.] APROBAR --> FIN{{1}} </pre>	<ol style="list-style-type: none"> 1. Se revisa si el estudiante a matricular recibió el informe de aparición en el SIMAT, debe haber definido esta situación en la institución educativa correspondiente. 2. Si continúa éste problema no podrá matricularse hasta no hacerlo. 3. Se verifica que haya entregado la documentación necesaria para la matrícula, y a su vez se confirma que el pago de recibo se haya efectuado 4. Se escriben las observaciones pertinentes respecto a documentos faltantes y acuerdo de financiación del pago. Luego se procede a aprobar la matrícula del estudiante. 	<p>Secretaria</p> <p>Secretaria</p> <p>Secretaria</p> <p>Secretaria</p>
		DOCUMENTOS REF.



PROCESO DE REGISTRO ACADÉMICO - SUBPROCESO MATRÍCULA		Código: PRA02
PROCEDIMIENTO APROBAR MATRÍCULA		Versión: 01
DIAGRAMA DE FLUJO	DESCRIPCIÓN	RESPONSABLE
<pre> graph TD Start{1} --> A[Aprobación de matrícula por parte del rector] A --> B[Formato de Matrícula] B --> End[FIN] </pre>	<p>6. El rector de la institución debe aprobar la matrícula del estudiante, previa aprobación de la secretaria.</p> <p>7. Se diligencia el formato de matrícula de acuerdo con los datos relacionados en el formato de pre matrícula y de acuerdo al año que va a cursar.</p>	<p>Rector</p> <p>Secretaria</p> <p>Formato de matrícula</p>



PROCESO DE REGISTRO ACADÉMICO - SUBPROCESO REGISTRO DE MATRÍCULA			Código: PRA01
PROCEDIMIENTO REGISTRO DE PREMATRÍCULA			Versión: 01
DIAGRAMA DE FLUJO	DESCRIPCIÓN	RESPONSABLE	DOCUMENTOS REF.
<p>INICIO</p> <p>↓</p> <p>Generación de Formatos de Pre matricula para Alumnos Nuevos y Antiguos</p> <p>↓</p> <p>Entrega del Formato de Prematricula</p> <p>↓</p> <p>Diligenciamiento del Formato y Entrega</p> <p>↓</p> <p>Diligenciamiento de Consignaciones Manualmente</p> <p>↓</p> <p>Entrega de Liquidaciones</p> <p>↓</p> <p>Estudiante paga los costos educativos</p>	<p>7. El Colegio a través de un auxiliar administrativo con capacitación previa en los procesos de matricula genera los formatos de pre matricula para alumnos nuevos y antiguos, teniendo en cuenta todos los datos necesarios para el SIMAT.</p> <p>8. El colegio entrega el formato de prematricula para estudiantes nuevos.</p> <p>9. El estudiante nuevo diligencia el formato de prematricula y lo entrega en la institución.</p> <p>10. ertinentes.</p>	<p>Secretaria</p> <p>Secretaria</p> <p>Secretaria</p> <p>Secretaria</p> <p>Secretaria</p> <p>Secretaria</p>	<p>Formato de Pre matrícula</p> <p>Recibo de pago de matrícula</p>



PROCESO DE REGISTRO ACADÉMICO - SUBPROCESO REGISTRO DE MATRÍCULA		Código: PRA02
PROCEDIMIENTO APROBAR MATRÍCULA		Versión: 01
DIAGRAMA DE FLUJO	DESCRIPCIÓN	RESPONSABLE
<pre> graph TD A[Verificación del Seguro] --> B[Entrega de Documentos] B --> C[Padre de Familia o Acudiente Firma la Matricula] C --> D[Diligenciar Formato de Matricula] D --> E{1} </pre>	<p>5. Se revisa si el estudiante a matricular recibió el informe de aparición en el SIMAT, debe haber definido esta situación en la institución educativa correspondiente.</p> <p>6. Si continúa éste problema no podrá matricularse hasta no hacerlo.</p> <p>7. Se verifica que haya entregado la documentación necesaria para la matrícula, y a su vez se confirma que el pago de recibo se haya efectuado</p> <p>8. Se escriben las observaciones pertinentes respecto a documentos faltantes y acuerdo de financiación del pago. Luego se procede a aprobar la matrícula del estudiante.</p>	<p>Secretaria</p> <p>Secretaria</p> <p>Secretaria</p> <p>Secretaria</p>
		DOCUMENTOS REF.



PROCESO DE REGISTRO ACADÉMICO - SUBPROCESO MATRÍCULA			Código: PRA02
PROCEDIMIENTO APROBAR MATRÍCULA			Versión: 01
DIAGRAMA DE FLUJO	DESCRIPCIÓN	RESPONSABLE	DOCUMENTOS REF.
<pre> graph TD Start([1]) --> Aprobacion[Aprobación de matrícula por parte del rector] Aprobacion --> Formato[Formato de Matrícula] Formato --> Fin([FIN]) </pre>	<p>8. El rector de la institución debe aprobar la matrícula del estudiante, previa aprobación de la secretaria.</p> <p>9. Se diligencia el formato de matrícula de acuerdo con los datos relacionados en el formato de pre matrícula y de acuerdo al año que va a cursar.</p>	<p>Rector</p> <p>Secretaria</p>	<p>Formato de matrícula</p>



PROCESO DE REGISTRO ACADÉMICO - SUBPROCESO CONTROL DE PAGOS			Código: PRA01
PROCEDIMIENTO EXPEDIR RECIBOS DE PAGO			Versión: 01
DIAGRAMA DE FLUJO	DESCRIPCIÓN	RESPONSABLE	DOCUMENTOS REF.
<pre> graph TD INICIO([INICIO]) --> A[Definir el concepto del recibo De pago] A --> B[Registrar los datos del estudiante que realizará el pago] B --> C{¿Aplican descuentos?} C -- Si --> D[Aplicar descuentos] C -- No --> E[Definir costo del recibo de pago] D --> E E --> F[Incluir otros datos] F --> G[Recibo de pago] G --> H([Fin]) </pre>	<ol style="list-style-type: none"> 1. Definir el concepto del recibo de pago (Matrícula, Pensión, etc.) 2. Registrar los datos del estudiante que realizará el pago 3. Definir si aplican descuentos realizados por diferentes conceptos (Nivel de SISBEN, estrato, desplazado.) 4. Definir el costo del recibo de pago 5. Incluir información faltante en el recibo (Fecha límite de pago, Tipo de Pago,) 6. Expedir recibo de pago 	<p>Secretaria</p> <p>Secretaria</p> <p>Secretaria</p> <p>Secretaria</p> <p>Secretaria</p> <p>Secretaria</p>	<p>Recibo de pago</p>



PROCESO DE REGISTRO ACADÉMICO - SUBPROCESO GESTIÓN DE INFORMACIÓN ACADÉMICA.			Código: PRA01
PROCEDIMIENTO GESTIÓN DE CARGA ACADÉMICA			Versión: 01
DIAGRAMA DE FLUJO	DESCRIPCIÓN	RESPONSABLE	DOCUMENTOS REF.
<pre> graph TD INICIO([INICIO]) --> DECISION{¿El área o asignatura requerida ha sido creada?} DECISION -- No --> CREAR[Crear o modificar características de asignatura o área.] DECISION -- Si --> INCLUIR[Incluir asignatura en el plan académico correspondiente.] CREAR --> INCLUIR INCLUIR --> ASIGNAR[Crear y/o modificar proyectos y actividades extracurriculares, asignando un docente responsable en cada caso] ASIGNAR --> FIN{{1}} </pre>	<ol style="list-style-type: none"> 1. De acuerdo a los planes académicos establecidos en la institución para el desarrollo académico e integral del estudiante de cada curso, el coordinador creará, modificará y eliminará las áreas y asignaturas que se consideren necesarias. 2. De crear nuevas áreas se debe asignar en cada caso el director asociado. Si se trata de una nueva asignatura es necesario definir el área a la cual pertenece y relacionarla con el plan académico del grado. 3. El coordinador académico incluirá tanto actividades extracurriculares como proyectos que se quieran desarrollar durante el año, asignando un docente encargado en cada caso. 	<p>Coordinador Académico</p> <p>Coordinador Académico</p> <p>Coordinador Académico</p>	



PROCESO DE REGISTRO ACADÉMICO - SUBPROCESO GESTIÓN DE INFORMACIÓN ACADÉMICA.		Código: PRA02	
PROCEDIMIENTO GESTIÓN DE CARGA ACDÉMICA		Versión: 01	
DIAGRAMA DE FLUJO	DESCRIPCIÓN	RESPONSABLE	DOCUMENTOS REF.
<pre> graph TD Start([1]) --> Step1[Definir los docentes que dictarán cada asignatura y las horas de trabajo semanales por cada uno.] Step1 --> Step2[Definir los grados que estarán disponibles en el nuevo año.] Step2 --> Step3[Definir el número de grupos por grado y asignar a cada uno su respectivo Director de Grupo.] Step3 --> Step4[Definir intensidad horaria por grado] Step4 --> Step5[Definir el número de horas por materia en cada grado.] Step5 --> End([2]) </pre>	<p>4. De acuerdo a sus competencias, cada docente podrá encargarse de dictar una o varias materias. Dicha selección será efectuada de acuerdo a disponibilidad y afinidad con las asignaturas.</p> <p>5. A su vez el coordinador podrá definir los grados que se dictarán, de acuerdo a las jornadas y disponibilidad de las instalaciones de la institución.</p> <p>6. Se definirá también el número de grupos por grado. Para ello podrá crearlos, modificarlos y/o asignarlos a un docente para que sea el respectivo Director</p> <p>7. Definir la intensidad horaria para cada uno de los grados dependiendo de su plan académico</p> <p>8. Definir la intensidad horaria por grupo dependiendo del plan académico de grado</p>	<p>Rector</p> <p>Rector</p> <p>Rector</p> <p>Secretaria</p> <p>Secretaria</p>	

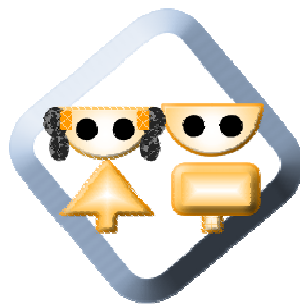


PROCESO DE REGISTRO ACADÉMICO - SUBPROCESO GESTIÓN DE INFORMACIÓN ACADÉMICA.			Código: PRA02
PROCEDIMIENTO GESTIÓN DE CARGA ACDÉMICA			Versión: 01
DIAGRAMA DE FLUJO	DESCRIPCIÓN	RESPONSABLE	DOCUMENTOS REF.
<pre> graph TD Start([2]) --> Step1[Generación de horarios, incluyendo día de la semana, materia, docente, grupo y salón.] Step1 --> Step2[/Informe de Horarios/] Step2 --> Step3[Definir el tiempo laborado por cada docente en el] Step3 --> Step4[/Cronograma académico/] Step4 --> End([FIN]) </pre>	<p>9. En consideración con los pasos anteriores, la disponibilidad de las aulas y la distribución adecuada de tiempo disponible para cada docente y para cada grupo, se definen los respectivos horarios para cada curso y para cada docente.</p> <p>10. Previa revisión por parte de los docentes y el rector, se aprueban los horarios generados y se genera el respectivo informe final de horarios</p> <p>11. Indicar el tiempo laborado por cada docente durante la semana.</p> <p>12. Definir las actividades del cronograma académico en consideración del desarrollo de actividades académicas en general, actividades extracurriculares, celebraciones especiales, entrega de boletines, entre otros.</p>	<p>Secretaria</p> <p>Rector</p> <p>Secretaria</p> <p>Secretaria</p>	<p>Informe General de Horarios para docentes y grupos.</p>



PROCESO DE REGISTRO ACADÉMICO - SUBPROCESO GESTIÓN DE INFORMACIÓN ACADÉMICA.			Código: PRA02
PROCEDIMIENTO CONTROL DE NOTAS			Versión: 01
DIAGRAMA DE FLUJO	DESCRIPCIÓN	RESPONSABLE	DOCUMENTOS REF.
<pre> graph TD INICIO([INICIO]) --> A[Seleccionar el tipo de calificación a utilizar.] A --> B[Registrar información respecto a logros e indicadores, por cada materia para cada grado.] B --> C[/Formato de Boletín de calificaciones por grado y periodo Académico/] C --> D[Registro de notas y fallas, por alumno en su respectivo boletín.] D --> FIN([FIN]) </pre>	<ol style="list-style-type: none"> 1. En primer lugar se debe seleccionar el tipo de calificación que se utilizará en el año, puede ser numérico o por letras. 2. En los dos casos será necesario ingresar información correspondiente a cada logro e indicador por materia, considerando el plan de estudios del grado respectivo. 3. Los datos recolectados respecto a logros e indicadores por materia para los diferentes grados, será clasificada en los periodos académicos fijados. Por grado y según el periodo se creará un formato modelo para boletín de calificaciones. 4. Por cada estudiante, en base al actual periodo académico y el grado, se registraran en el respectivo boletín, los valores asociados a cada logro e indicador de acuerdo a su desempeño, observaciones, el reporte de asistencia a clases y se imprimirá previa revisión. 	<p>Secretaria</p> <p>Docente</p> <p>Secretaria</p> <p>Secretaria</p>	<p>Modelo formato de boletín</p> <p>Boletín de Notas por alumno</p>

Anexo O – Especificación de Requisitos (SOFGESCOL)



**Especificación de
requisitos de software
(ERS)**

**Proyecto SOFGESCOL
Revisión Final**



Ficha del documento

Fecha	Revisión	Autor	Verificado dep. Calidad
2010/01/18	1	Laura Pulido Díaz	Firma
2010/01/18	1	Luz Dary Méndez Uribe	Firma

Contenido

1 INTRODUCCIÓN

El presente documento ha sido creado con el fin de definir las especificaciones y pautas generales que determinarán el desarrollo del prototipo software “SOFGESCOL”, que corresponde al trabajo de grado de las estudiantes Laura Pulido Díaz y Luz Dary Méndez Uribe, para optar por el título de Ingeniero de Sistemas de la Universidad Industrial de Santander. (Bucaramanga - Colombia).

1.6 Propósito

Describir claramente las necesidades a suplir por el nuevo sistema, en concordancia a lo establecido en reuniones previas con los usuarios. Este documento es base en cada una de las etapas de desarrollo herramienta, además de servir como vía de comunicación entre las partes que se encuentran relacionadas con el proyecto. Adicional a esto, sirve de soporte al proceso de verificación de requisitos al cual será sometido, considerando las pruebas para el presente prototipo.

1.7 Ámbito del Sistema

SOFGESCOL es un prototipo software diseñado para servir de soporte a actividades relacionadas con instituciones educativas.

1.8 Definiciones, Acrónimos y Abreviaturas

ESPECIFICACIÓN DE REQUISITOS: Documento estructurado que presenta los resultados de la definición de necesidades, la conceptualización operacional y las actividades de análisis de un diseño de sistema o producto software.

EISI: Escuela de Ingeniería de Sistemas e Informática

ESTÁNDAR: Una norma es un conjunto de reglas estandarizadas que contienen un catálogo de requisitos.

INDICADORES: Parámetros creados para medir y controlar el alcance y cumplimiento de algún objetivo o meta.

JEE5:

PROTOTIPO SOFTWARE: Programa de prueba que precede una herramienta software a implementar en una organización.



REQUISITO: Condición o capacidad necesaria en un sistema o componente de sistema para la solución de un problema o logro de un objetivo.

REQUISITO FUNCIONAL: Operación o función que debe ser efectuada por un sistema o componente del mismo.

REQUISITO NO FUNCIONAL: Condición o restricción establecida para la implementación de los requisitos funcionales.

REUSABILIDAD: Grado en el que un programa se puede utilizar en otras aplicaciones.

SGC: Sistema de Gestión de Calidad.

UIS: Universidad Industrial de Santander

UML: Unified Modeling Language (Lenguaje de Modelado Unificado).

1.9 Referencias

Referencia	Título	Fecha	Autor
Ref. 01	Estándar para la especificación de requisitos software. IEEE- 830	1998	IEEE
Ref. 02	Plan de trabajo de Grado	2009	Pulido Laura, Méndez Luz Dary
Ref. 03	El Lenguaje Unificado de Modelado	1999	BOOCH, Grady; RUMBAUGH, James y JACOBSON, Ivar. Addison Wesley
Ref. 04	Caracterización de Procedimientos de la Gestión de Registro Académico (ANEXO N)	2010	Pulido Laura, Méndez Luz Dary

1.10 Visión General del Documento

El presente documento ha sido creado en función de lo dispuesto por el estándar para la especificación de requisitos de Software IEEE 830 – 1998, y está conformado por tres secciones:

La primera sección consta de una introducción, incorporando ciertos elementos que facilitan una visión general de la especificación de recursos del sistema a desarrollar.

En la segunda sección del documento se presenta una descripción general del sistema, a fin de establecer las principales funciones que debe realizar la aplicación, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, todo a grandes rasgos.

Por último, la tercera sección del documento busca definir detalladamente cada uno de los requisitos que debe satisfacer el sistema.

2 Descripción General

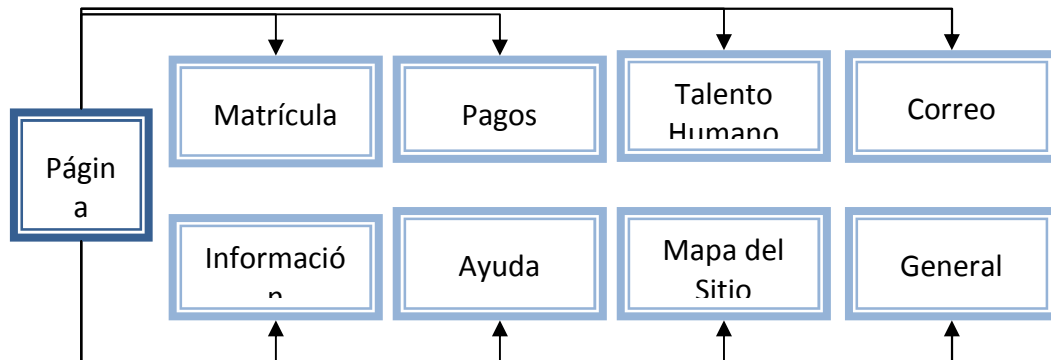
2.1 Perspectiva del Producto

SOFGESCOL es un producto que no está integrado ni hace parte de otro tipo de proyecto, en realidad se trata de un prototipo software independiente. Para su funcionamiento, ésta aplicación será almacenada y se ejecutará desde un servidor WEB; su acceso será mediante uso de los últimos exploradores y los más usados en la actualidad, Mozilla Firefox 3.6 y Google Chrome.



2.2 Funciones del Producto

La Figura 1, representa el diagrama de despliegue de paquetes según UML para el modelado y documentación de sistemas. Muestra una vista estructural que ayuda en el diseño y desarrollo de los mismos.



Obsérvese que el acceso se da a través de un explorador Web, y la aplicación está alojada y se ejecuta desde un servidor Web. En la Tabla aparecen cada uno de los paquetes con las funcionalidades principales, una descripción detallada de los requerimientos y funcionalidades de cada paquete se encuentra en la sección 3 de este documento.

<i>Matrícula</i>	Permite el desarrollo y seguimiento del procedimiento de matrícula en cada etapa. Informa respecto a estados para cada estudiante durante el trámite, sobre pagos a realizar, fechas y requisitos necesarios para hacerlo.
<i>Pagos</i>	Permite la generación de recibos con los conceptos necesarios. Los conceptos y valores que pueden ser inscritos para cada nivel académico, generación de los recibos con los descuentos que se estipulen por grados, niveles, estudiantes, etc.
<i>Recurso Humano</i>	Permite administrar información sobre personal asociado a la institución, así como el vínculo con actividades, objetivos y proyectos desarrollados en la institución.
<i>Correo Electrónico</i>	El componente permite la comunicación entre los usuarios del colegio, para manejar un conducto regular es posible configurar los permisos de los diferentes perfiles para el envío de comunicados.
<i>Información Académica</i>	
<i>General</i>	Permite gestionar los usuarios y los perfiles. Otorgar los permisos adecuados para el acceso a cada uno de los componentes. Se distinguen tres tipos de usuarios: Agentes, Proveedores, Alta Gerencia.



2.3 Características de los usuarios

Tipo Usuario	Root
Habilidades	Navegación Internet, Conocimiento dependencias, enfoque de procesos.
Actividades	Súper administrador del sistema, con permiso para acceder y manipular información a nivel general y quien puede conceder y administrar permisos a usuarios con otros perfiles.
Tipo Usuario	Administrativo
Habilidades	Navegación Internet, Conocimiento dependencias, enfoque de procesos.
Actividades	Ver/Actualizar estados de cuenta Registrar pagos, generar recibos y certificados Mantenimiento general de matrícula en consideración de todos los estados del procedimiento (pre-matrícula, aprobar pre-matrícula, etc) Realizar los mantenimientos generales que le competen Administrar información del personal de la institución Mantenimiento de horarios Generar boletines de notas, previamente aprobados Generar informes de todo tipo Entre otras actividades
Tipo Usuario	Docente
Habilidades	Navegación Internet, Conocimiento dependencias
Actividades	Ver comunicados, calendario de la institución Administrar comunicados para sus estudiantes Actualizar sus datos y horario de atención Crear y/o participar en el desarrollo de objetivos, proyectos y actividades propuestas a las cuales se encuentra asociado Enviar y Recibir correos docentes, estudiantes y de la institución Registrar notas, observaciones y fallas de sus estudiantes
Tipo Usuario	Estudiante
Habilidades	Navegación Internet
Actividades	Ver comunicados, calendario Consultar información relacionada con trámites de matrícula Actualizar Datos Consultar estados de cuenta Consultar datos de docentes y horario de atención Participar en el desarrollo de actividades propuestas y a las cuales se encuentra asociado Enviar y Recibir correos docentes y de la institución Consultar el informe de comportamiento, rendimiento académico y fallas de los estudiantes a cargo
Tipo Usuario	Acudiente
Habilidades	Navegación Internet



Actividades	Ver comunicados, calendario Consultar información relacionada con trámites de matrícula Actualizar Datos Consultar estados de cuenta Consultar datos de docentes y horario de atención Enviar y Recibir correos docentes y de la institución Consultar el informe de comportamiento, rendimiento académico y fallas de los estudiantes a cargo
--------------------	--

2.4 Restricciones

- ♦ Lenguajes y Tecnologías en uso: JEE5, POSTGRESQL, XML, JSP, AJAX, JAVASCRIPT, HTML, IREPORT, FLASH/ACTIONSCRIPT.
- ♦ Los desarrolladores programarán en el IDE NetBeans 6.7.1
- ♦ El desarrollo del prototipo SOFGESCOL deberá cumplir con el plan de trabajo del proyecto de grado.[ANEXO Z], y tomará como parámetros de Registro Académico lo estipulado en el documento Marco de Registro Académico [ANEXO N].
- ♦ Para la creación del prototipo se seguirá la metodología de desarrollo diseñada durante el proyecto.

Requisitos de software y hardware

- ♦ La aplicación se alojará en cualquier servidor que cumpla con las características para implantar sistemas JEE5 (GLASSFISH, JBOSS, etc).
- ♦ *El sistema funcionará correctamente con las siguientes características del servidor:*
 - Servidor de Bases de datos: PostgreSQL 9
 - Servidor: Cualquiera que cumpla con las características para implantar sistemas JEE5.
 - Generador de Reportes: IREPORT
 - Sistema Operativo del servidor: Los sistemas en los que se probará la aplicación son Windows XP, 7 y Linux Debian.

Arquitectura de Desarrollo:

- Lenguaje: Java Enterprise Edition 5.
 - Persistencia: Toplink.
 - Componentes de la Interfaz: IceFaces.
- ♦ Los equipos clientes requieren acceso a internet sin importar su sistema operativo, con exploradores de internet mencionados
 - ♦ Este prototipo NO soporta navegadores WAP ni de dispositivos móviles



2.5 Suposiciones y Dependencias

Se asume una adecuada instalación y configuración del sistema operativo y del software asociado tanto para el servidor como para los equipos cliente, de acuerdo a lo establecido en el anterior numeral. Se asume que cada uno de sus usuarios ha recibido la capacitación pertinente o tiene nociones básicas para interactuar con el sistema y desarrollar a cabalidad cada una de sus funciones.

2.6 Requisitos Futuros

El prototipo software puede servir como base para la herramienta completa de gestión de los procesos de contratación. Se le pueden agregar funcionalidades y mejorar aspectos no funcionales como seguridad, rendimiento y presentación.

3 Requisitos Específicos

3.1 Interfaces Externas

Interfaces de usuario

Los usuarios del prototipo SOFGESCOL han de interactuar con el sistema a través de elementos de pantalla convencionales (tales como, cajas de texto, cajas con lista desplegable dinámica o estática, tablas de datos generadas a partir de una búsqueda, etc.), cada campo de un formulario traerá consigo una etiqueta que indica al usuario el dato solicitado. En casos donde se deben cargar archivos estos no deben superar el tamaño indicado. Los informes serán generados en formato .pdf. En cualquier momento se puede acceder al manual de usuario desde la aplicación la que servirá de guía durante el desarrollo de cualquier actividad.

Interfaces de hardware

El sistema está previsto para trabajar en una configuración de monitor 800x600; esto se debe a que es el estándar más usado. Para el caso de monitores wide screen (no cuadrados), el sistema centrará la interfaz visual, dejando un espacio libre a cada lado de la pantalla.

Interfaces de software

El prototipo no es creado para ser compatible con algún otro sistema de información. Será una plataforma Web, que podrá correr bajo ambiente de sistemas operativos Windows o Linux. Para el desarrollo del producto, se utilizarán los siguientes productos software.

Desarrollo de sistema en JEE5, bases de datos PostgreSQL, y soportes gráficos, generación de informes con IREPORT y otras herramientas.

Interfaces de comunicación

Las comunicaciones que se hagan entre las interfaces del sistema se harán a través de XML. El prototipo no se plantea como un sistema que interactúa con otros sistemas externos, por lo cual las comunicaciones externas no se toman en cuenta.

3.2 Requisitos Funcionales



Generales del Sistema

RFG01: Crear nuevos usuarios

<i>Código</i>	ERS_SOFGESCOL_1
<i>Nombre</i>	Crear nuevos usuarios
<i>Actor</i>	Root, Administrativo, Rector
<i>Propósito</i>	Crear usuarios, asignando a cada uno el perfil relacionado.
<i>Resultado</i>	Muestra al usuario un formulario para agregar nuevos usuarios. Este formulario debe contener los siguientes datos: Nombre de Usuario Contraseña Nombre Cédula Correo Electrónico Cargo en la Organización

RFG02: Editar información de usuarios

<i>Código</i>	ERS_SOFGESCOL_2
<i>Nombre</i>	Editar información de usuarios
<i>Actor</i>	Root, Administrativo, Rector
<i>Propósito</i>	Editar información de los usuarios, e incluso restablecer contraseña de ingreso al sistema por solicitud del usuario
<i>Resultado</i>	Presentar al usuario actual del sistema campos disponibles para editar, en caso de que su perfil conceda tal permiso

RFG03: Editar

<i>Código</i>	ERS_SOFGESCOL_3
<i>Nombre</i>	Editar
<i>Actor</i>	Root, Administrativo, Rector
<i>Propósito</i>	Mostrar al usuario del sistema información y funciones que le competen, dependiendo de los permisos concedidos por su perfil actual.
<i>Resultado</i>	Mediante control de acceso y manejo de sesión, el sistema podrá permitir al usuario acceder a información y funciones que le competen según su perfil actual.

RFG04: Ingreso a la Aplicación

<i>Código</i>	ERS_SOFGESCOL_4
<i>Nombre</i>	Ingreso a la aplicación
<i>Actor</i>	Todos los usuarios del sistema
<i>Propósito</i>	A partir de la validación y registro de ingreso al sistema, el actual usuario tendrá acceso a información y funciones del sistema dependiendo de los permisos relacionados a su perfil actual.
<i>Resultado</i>	Mediante manejo de sesión y control de acceso por parte del sistema, el usuario actual tendrá a su alcance la información y funciones que le competen.

RFC05: Administrar permisos por perfil de usuario

<i>Código</i>	ERS_SOFGESCOL_5
<i>Nombre</i>	Administrar permisos por perfil de usuario
<i>Actor</i>	Root, Administrativo
<i>Propósito</i>	Asignar permisos según perfiles de usuario para realizar actividades básicas para cada página.
<i>Resultado</i>	El usuario podrá definir los permisos necesarios para crear, actualizar, carga masiva, eliminar y acceder a vínculos especiales.



RFM06: Despliegue de informes disponibles

<i>Código</i>	ERS_SOFGESCOL_6
<i>Nombre</i>	Despliegue de informes disponibles
<i>Actor</i>	Administrativo, Rector
<i>Propósito</i>	Listar todos los informes disponibles, haciendo énfasis en los más solicitados por sus usuarios. Cada uno de los elementos de la lista debe contener información básica, que facilite su selección en caso de necesitar su impresión inmediata.
<i>Resultado</i>	Mostrar una lista de todos los informes que puede generar el sistema y presentar el listado preferencial de dichos informes por parte del personal a cargo. Cada elemento del listado muestra una descripción general indicando Nombre, Descripción e incluso una opción de imprimir. Nota =Cada módulo traerá asociada una sección de informes requeridos

RFM07: Visualizar Informe Seleccionado

<i>Código</i>	ERS_MATRICULA_7
<i>Nombre</i>	Visualizar Informe Seleccionado
<i>Actor</i>	Administrativo, Rector
<i>Propósito</i>	El sistema permite seleccionar un informe específico de la lista de informes disponible y a continuación se deben mostrar los datos relacionados a dicho informe.
<i>Resultado</i>	Mostrar la información actual relacionada con el informe seleccionado del listado.

[Sección A] = Requisitos del componente Matrícula

RFM01: Buscar Estudiantes Pre-Matriculados

<i>Código</i>	ERS_MATRICULA_1
<i>Nombre</i>	Buscar estudiantes pre-matriculados
<i>Actor</i>	Administrador
<i>Propósito</i>	De acuerdo a los patrones de búsqueda más frecuentes: Estudiante (número de documento), curso y año, el sistema buscará estudiantes pre-matriculados que cumplan con dicha(s) característica(s).
<i>Resultado</i>	Listar estudiantes pre-matriculados de acuerdo a los criterios de búsqueda establecidos.

RFM02: Listar registros de búsqueda

<i>Código</i>	ERS_MATRICULA_2
<i>Nombre</i>	Listar registros de búsqueda
<i>Actor</i>	Administrador
<i>Propósito</i>	Sin importar el número de registros presentados como resultado de la búsqueda, se mostrará el listado en bloques de 10 por página y además se mostrará el total de registros listados. Cada registro mostrará información básica.
<i>Resultado</i>	El sistema presentará tantas páginas de resultado como sea necesario y cada una estará conformada por un máximo de 10 registros, indicando el número total de registros listados.

RFM03: Visualizar detalles de Pre matricula



Código	ERS_MATRICULA_3
Nombre	Visualizar detalles de Pre matrícula
Actor	Administrador
Propósito	El sistema permite seleccionar un registro de pre matrícula específico de la lista de búsqueda y a continuación visualizar los datos relacionados a dicho formato de pre matrícula.
Resultado	Mostrar la información actual relacionada con el formato de pre matrícula seleccionado del listado de búsqueda

RFM04: Crear Registro de pre matrícula

Código	ERS_MATRICULA_4
Nombre	Crear Registro de pre matrícula
Actor	Administrador
Propósito	Permitir el ingreso de la información necesaria, para llevar a cabo el registro de pre matrícula para cada alumno. Los datos relacionados con éste trámite se encuentran relacionados en el formato de pre matrícula (Anexo B)
Resultado	El registro de pre-matrícula incluye información del estudiante distribuida en 5 secciones: Datos de la pre-matrícula, Datos del Estudiante, Datos de la madre, Datos del padre e Información del acudiente, cada uno de estos campos corresponden al formato de pre matrícula proporcionado por la institución (Anexo B).

RFM05: Informar Verificación SIMAT

Código	ERS_MATRICULA_5
Nombre	Informar Verificación SIMAT
Actor	Administrador
Propósito	Al seleccionar un registro de pre matrícula, el sistema mostrará la información relacionada con éste y habilitará una opción para su aprobación. También puede realizarse ésta acción a partir del listado de pre-matrículas mostrado.
Resultado	En cuanto uno de los registros de pre matrícula sea seleccionado para realizar ésta acción, el sistema mostrará la información correspondiente al registro y se habilitará un botón para realizar su aprobación. De ser necesario se podrán modificar datos para su correspondiente aprobación. Se tiene un campo en cada pre-matrícula listada que permite cambiar éste estado.

RFM06: Modificar Registro de Pre matrícula

Código	ERS_MATRICULA_6
Nombre	Modificar Registro de Pre matrícula
Actor	Administrador
Propósito	La información que corresponde a un registro de pre matrícula, podrá ser modificada por personal autorizado.
Resultado	Los datos que corresponden a un registro de pre matrícula podrán ser modificados después de ser visualizados, en caso de que sea necesario.

RFM07: Aprobar Pre-matrícula

Código	ERS_MATRICULA_7
Nombre	Aprobar pre-matrícula
Actor	Administrador



<i>Propósito</i>	Indicar si la pre-matrícula ha sido o no aprobada, definir el valor a pagar por concepto de matrícula y relacionar a un estudiante con un curso para establecer los cupos disponibles.
<i>Resultado</i>	Cada pre-matrícula debe ser aprobada en consideración de la información académica que contiene el formato. A su vez, es necesario indicar cuál es el descuento considerado para el pago de la matrícula y además se debe seleccionar el curso al que ingresará el estudiante.

RFM8: Crear Registro de Matrícula

<i>Código</i>	ERS_ MATRICULA_ 8
<i>Nombre</i>	Crear Registro de Matrícula
<i>Actor</i>	Administrador
<i>Propósito</i>	Previo validación de la información relacionada al formato de pre matrícula del estudiante seleccionado para el presente procedimiento, el sistema debe permitir el ingreso de la información necesaria, para llevar a cabo el registro de Matrícula. Los datos relacionados con éste trámite se encuentran relacionados en el formato de Matricula (Anexo B)
<i>Resultado</i>	El registro de Matricula distribuye la información del estudiante en 3 secciones: Datos, Seleccionar Curso, y documentos. La primera sección contiene información general incluyendo un campo de valor pagado de acuerdo a la consignación bancaria realizada o al acuerdo con la institución, y campo de observaciones para ingresar información adicional acerca de documentos pendientes por entregar, saldo pendiente, entre otras. La segunda sección permite al usuario del sistema seleccionar el curso asignado para el estudiante y en la última sección se listan los documentos necesarios para el trámite de matrícula informando sobre aquellos que han sido entregados y los documentos faltantes. En cuanto el usuario ingrese la información necesario para dar término a éste trámite el sistema permitirá guardar dicho registro y continuar con un nuevo registro de matrícula.

RFM9: Listar Estudiantes Matriculados

<i>Código</i>	ERS_ MATRICULA_ 9
<i>Nombre</i>	Listar Estudiantes Matriculados
<i>Actor</i>	Administrador
<i>Propósito</i>	Mostrar el listado correspondiente a estudiantes matriculados que Corresponden a los datos de búsqueda (curso, año y/o grado). Nota: cada elemento de la lista tendrá información básica para su identificación.
<i>Resultado</i>	El sistema buscará y mostrará un listado de los estudiantes que han sido matriculados y que corresponden a los patrones de búsqueda ingresados, cada uno de los registros de la lista tendrá Nombre, Documento de Identidad, datos personales generales y un campo adicional que corresponde a la verificación de matrícula.

RFM10: Modificar Registro de Matrícula

<i>Código</i>	ERS_ MATRICULA_ 10
<i>Nombre</i>	Modificar Registro de Matrícula
<i>Actor</i>	Administrador
<i>Propósito</i>	La información que corresponde a un registro de matrícula, podrá ser modificada por personal autorizado.



<i>Resultado</i>	Los datos que corresponden a un registro de matrícula podrán ser modificados después de ser visualizados, en caso de que sea necesario.
------------------	---

RFM11: Aprobar Matrícula

<i>Código</i>	ERS_ MATRICULA_11
<i>Nombre</i>	Aprobar Matrícula
<i>Actor</i>	Rector
<i>Propósito</i>	Registrar aprobación de la matrícula por parte del rector.
<i>Resultado</i>	El rector podrá aprobar la matrícula de cada uno de los estudiantes. Puede hacerlo uno por uno o seleccionar varios registros y hacerlo en forma masiva.

RFM12: Cambiar Curso

<i>Código</i>	ERS_ MATRICULA_12
<i>Nombre</i>	Cambiar Curso
<i>Actor</i>	Administrador
<i>Propósito</i>	Cambiar el curso al cual pertenece el estudiante que ha sido matriculado.
<i>Resultado</i>	Una vez matriculado, personal autorizado podrá cambiar de curso a un estudiante en particular a otro curso de la misma sede.

RFM13: Consultar y/o actualizar Datos

<i>Código</i>	ERS_ MATRICULA_13
<i>Nombre</i>	Consultar y/o actualizar Datos
<i>Actor</i>	Acudiente
<i>Propósito</i>	Permitir la actualización y consulta de datos generales que pueden verse y/o modificarse de los estudiantes que tenga a cargo.
<i>Resultado</i>	El sistema mostrará los estudiantes a cargo del acudiente. Los datos generales del estudiante podrán ser vistos por el acudiente y se permite hacer cambios a los datos que puedan ser modificados según el colegio.

RFM14: Consultar y/o actualizar Datos

<i>Código</i>	ERS_ MATRICULA_14
<i>Nombre</i>	Consultar y/o actualizar Datos
<i>Actor</i>	Estudiante
<i>Propósito</i>	Permitir la actualización y consulta de datos generales del estudiante que pueden verse y/o modificarse.
<i>Resultado</i>	Los datos generales del estudiante podrán ser vistos por él y se permite hacer cambios a los datos que puedan ser modificados según el colegio.

RFM15: Información sobre trámites de matrícula

<i>Código</i>	ERS_ MATRICULA_15
<i>Nombre</i>	Información sobre trámites de matrícula
<i>Actor</i>	Estudiante, Acudiente
<i>Propósito</i>	Mostrar las fechas importantes relacionadas con la matrícula de estudiantes, así como los documentos e información adicional.
<i>Resultado</i>	Los usuarios podrán acceder a información general relacionada al trámite de matrícula.

RFP016: Mantenimientos Generales requeridos



<i>Código</i>	ERS_MATRICULA_16
<i>Nombre</i>	Mantenimientos Generales requeridos
<i>Actor</i>	Administrador
<i>Propósito</i>	Permitir el ingreso de la información necesaria para crear, modificar y eliminar los registros de datos requeridos y que estén asociados a una entidad (BD) tales como: Ciudades, Departamentos, Eps, Ips, Ars, etc.
<i>Resultado</i>	El usuario podrá realizar dichos mantenimientos a la información según lo requiera, en cada uno de los casos podrá actualizar la información en base al uso de formularios diseñados para tales tareas.

[Sección B] = Requisitos del componente Pagos

RFP01: Mantenimiento a Tipos de pago

<i>Código</i>	ERS_PAGOS_1
<i>Nombre</i>	Mantenimiento a Tipos de pago
<i>Actor</i>	Administrativo
<i>Propósito</i>	Ingresar, eliminar y modificar los diferentes tipos de pago que pueden estar asociados a un estudiante.
<i>Resultado</i>	El usuario podrá crear, modificar, eliminar y/o buscar los diferentes tipos de pago, así como los costos que sean necesarios.

RFP02: Registrar Pagos Realizados

<i>Código</i>	ERS_PAGOS_2
<i>Nombre</i>	Registrar Pagos realizados
<i>Actor</i>	Administrativo
<i>Propósito</i>	Registrar pagos celebrados por el estudiante, según diferentes conceptos o tipos de pago.
<i>Resultado</i>	En base a recibos generados y cancelados previamente, se podrán registrar los pagos celebrados y presentar el estado actual de cuenta.

RFP03: Consultar Historial de Pagos

<i>Código</i>	ERS_PAGOS_3
<i>Nombre</i>	Consultar Historial de Pagos
<i>Actor</i>	Administrativo
<i>Propósito</i>	Buscar el historial de pagos de estudiantes por estudiante, año, o estado de cuenta (en mora o a paz y salvo).
<i>Resultado</i>	En base a los pagos pendientes y registrados se mostrarán detalles de recibos generados y cancelados previamente, se podrán registrar los pagos celebrados y presentar el estado actual de cuenta.

RFP04: Ver detalles del Estado de Cuenta

<i>Código</i>	ERS_PAGOS_4
---------------	-------------



<i>Nombre</i>	Ver detalles del Estado de Cuenta
<i>Actor</i>	Acudiente, Estudiante
<i>Propósito</i>	El sistema permite seleccionar un historial de cuenta específico de la lista de historiales disponible y a continuación se deben mostrar los datos relacionados
<i>Resultado</i>	Mostrar la información actual (pagos registrados, pagos pendientes, etc) relacionada con el historial seleccionado del listado.

RFP05: Mantenimiento y búsqueda de becas

<i>Código</i>	ERS_PAGOS_5
<i>Nombre</i>	Mantenimiento y búsqueda de becas
<i>Actor</i>	Administrativo
<i>Propósito</i>	Crear, eliminar, consultar y modificar becas que pueden estar asociadas a un estudiante.
<i>Resultado</i>	El usuario podrá crear, modificar, eliminar y/o buscar becas, por año, por estudiante, y/o por curso. Nota: Beca por estudiante al año.

RFP06: Mantenimiento y búsqueda de Costos

<i>Código</i>	ERS_PAGOS_6
<i>Nombre</i>	Mantenimiento y búsqueda de Costos
<i>Actor</i>	Usuario
<i>Propósito</i>	Ingresar, eliminar, consultar y modificar costos asociados a una matrícula por año, teniendo en cuenta la sección.
<i>Resultado</i>	El usuario podrá crear, modificar, eliminar y/o buscar costos, por año. Nota: Los costos no podrán ser modificados en un año posterior para el que fueron creados.

RFP07: Mantenimiento y búsqueda de descuentos

<i>Código</i>	ERS_PAGOS_7
<i>Nombre</i>	Mantenimiento y búsqueda de descuentos
<i>Actor</i>	Usuario
<i>Propósito</i>	Ingresar, eliminar, consultar y modificar costos asociados a una matrícula por año, teniendo en cuenta la sección.
<i>Resultado</i>	El usuario podrá crear, modificar, eliminar y/o buscar costos, por año. Nota: Los costos no podrán ser modificados en un año posterior para el que fueron creados.

RFP08: Generar recibos

<i>Código</i>	ERS_PAGOS_8
<i>Nombre</i>	Generar recibos
<i>Actor</i>	Usuario
<i>Propósito</i>	Generar recibos para uno o más estudiantes que compartan datos afines



	como el concepto, fecha de pago.
<i>Resultado</i>	El sistema solicitará los datos afines y a continuación se debe seleccionar el o los estudiantes para quienes se generarán los respectivos recibos.

RFP09: Imprimir recibos

<i>Código</i>	ERS_PAGOS_9
<i>Nombre</i>	Imprimir recibos
<i>Actor</i>	Pagador, Administrativo
<i>Propósito</i>	Permitir el ingreso de la información necesaria, para crear y enviar un correo.
<i>Resultado</i>	Mostrar una lista de todos los recibos generados y presentar el listado en formato PDF a fin de imprimirlos sin problemas.

RFP10: Registro y expedición de certificados

<i>Código</i>	ERS_PAGOS_10
<i>Nombre</i>	Registro y expedición de certificados
<i>Actor</i>	Administrativo
<i>Propósito</i>	En base al historial de pagos podrá generarse un certificado que indique en un momento determinado su estado de cuenta (Con saldo pendiente, Al día en los pagos, A paz y Salvo).
<i>Resultado</i>	El sistema genera certificados de pago para los estudiantes, de acuerdo a datos básicos solicitados previamente por el sistema.

RFP11: Mantenimientos Generales requeridos

<i>Código</i>	ERS_PAGOS_11
<i>Nombre</i>	Mantenimientos Generales requeridos
<i>Actor</i>	Pagador
<i>Propósito</i>	Permitir el ingreso de la información necesaria para crear, modificar y eliminar los registros de datos requeridos y que estén asociados a una entidad (BD) tales como: Secciones, días festivos, bancos, sucursales, cuentas.
<i>Resultado</i>	El usuario podrá realizar dichos mantenimientos a la información según lo requiera, en cada uno de los casos podrá actualizar la información en base al uso de formularios diseñados para tales tareas.

[Sección C] = Requisitos del componente Talento Humano

RFT01: Buscar hojas de vida registradas

<i>Código</i>	ERS_TALENTOHUMANO_1
<i>Nombre</i>	Buscar hojas de vida registradas
<i>Actor</i>	Administrativo
<i>Propósito</i>	Buscar hojas de vida de personal registrado de acuerdo al nombre y la clasificación básica de "Personal" o "Docente".



<i>Resultado</i>	Se desplegará un listado de hojas de vida disponibles de acuerdo al tipo seleccionado (“Personal” o “Docente”). El sistema presentará tantas páginas de resultado como sea necesario y cada una estará conformada por un máximo de 10 registros, indicando el número total de registros listados.
------------------	---

RFT02: Registrar nuevo docente

<i>Código</i>	ERS_ TALENTOHUMANO _2
<i>Nombre</i>	Registrar nuevo docente
<i>Actor</i>	Administrativo
<i>Propósito</i>	Permitir el ingreso de la información necesaria, para llevar a cabo el registro de un nuevo integrante del personal docente. Los datos relacionados con éste trámite se encuentran relacionados en el formato de Hoja de Vida (Anexo DF)
<i>Resultado</i>	El sistema presentará un formato solicitando los datos básicos del docente y permitiendo a su vez subir al sistema el archivo que corresponde a su hoja de vida, el cual quedará disponible para ser revisado por personal autorizado.

RFT03: Registrar nuevo personal

<i>Código</i>	ERS_ TALENTOHUMANO _3
<i>Nombre</i>	Registrar nuevo personal
<i>Actor</i>	Administrativo
<i>Propósito</i>	Permitir el ingreso de la información necesaria, para llevar a cabo el registro de un nuevo integrante del personal. Los datos relacionados con éste trámite se encuentran relacionados en el formato de Hoja de Vida (Anexo DF)
<i>Resultado</i>	El sistema presentará un formato solicitando los datos básicos del docente y permitiendo a su vez subir al sistema el archivo que corresponde a su hoja de vida, el cual quedará disponible para ser revisado por personal autorizado.

RFT04: Subir archivo(hoja de vida) relacionado con el registro

<i>Código</i>	ERS_ TALENTOHUMANO _5
<i>Nombre</i>	Subir archivo(hoja de vida) relacionado con el registro
<i>Actor</i>	Administrativo
<i>Propósito</i>	Como complemento al registro básico de datos de personal que labora en la institución, se debe incluir el archivo que corresponde a la hoja de vida de cada persona, la cual debe ser almacenada en el sistema.
<i>Resultado</i>	El sistema permite seleccionar y subir un archivo de tamaño máximo (___MB), en formato definidos (.doc o .PDF), por cada registro de personal o docente.

RFT05: Consultar y modificar hojas de vida registradas

<i>Código</i>	ERS_ TALENTOHUMANO _4
<i>Nombre</i>	Consultar y modificar hojas de vida registradas
<i>Actor</i>	Administrativo



<i>Propósito</i>	Al seleccionar un registro del listado (docentes o personal), se debe mostrar la información relacionada con éste y debe habilitarse para realizar cambios sobre datos modificables a su vez debe permitir guardar la actualización de datos realizada. El usuario actual tendrá acceso al archivo de hoja de vida, podrá verlo y cambiar dicho archivo.
<i>Resultado</i>	El sistema permite seleccionar un registro en particular del listado actualmente disponible y a continuación debe mostrar los datos relacionados a dicha hoja de vida (.). El sistema permitirá ver o descargar el archivo (hoja de vida), así como borrarlo del sistema e ingresar otro.

RFT06: Registrar nuevo proyecto

<i>Código</i>	ERS_ TALENTOHUMANO _6
<i>Nombre</i>	Registrar nueva proyecto
<i>Actor</i>	Administrativo, Docente
<i>Propósito</i>	El usuario podrá registrar nuevos proyectos, en los cuales podrán asignarse como participantes usuarios registrados en el sistema según se necesiten.
<i>Resultado</i>	El sistema solicitará datos básicos requeridos para la creación del nuevo proyecto e informará a los usuarios asociados, acerca de su participación en cuanto ingresen al sistema.

RFT07: Listar proyectos asociados a un usuario

<i>Código</i>	ERS_ TALENTOHUMANO _7
<i>Nombre</i>	Listar proyectos asociados a un usuario
<i>Actor</i>	Usuario
<i>Propósito</i>	Mostrar el listado de proyectos asociados a un usuario.
<i>Resultado</i>	El usuario podrá encontrar en una opción particular del menú de actividades disponibles, el listado de los proyectos a los cuales está ligado y obtendrá información requerida de los mismos.

RFT08: Consultar y modificar proyectos

<i>Código</i>	ERS_ TALENTOHUMANO _8
<i>Nombre</i>	Consultar y modificar proyectos
<i>Actor</i>	Administrador
<i>Propósito</i>	Permitir a personal autorizado modificar la información que corresponde a un registro de proyecto, previamente seleccionado.
<i>Resultado</i>	Los datos que corresponden a un proyecto podrán ser modificados después de ser visualizados, en caso de que sea necesario.

RFT09: Consultar tareas

<i>Código</i>	ERS_ TALENTOHUMANO _9
<i>Nombre</i>	Consultar tareas
<i>Actor</i>	Administrador
<i>Propósito</i>	Listar las tareas que sean objeto de búsqueda según los datos ingresados (Usuario a cargo, nombre de la tarea, estado) indicando por cada una su estado, con el fin de acceder a información relacionada con ella y de ser necesario modificar su estado actual.
<i>Resultado</i>	Mostrar una lista de todas las tareas encontradas, permitiendo a personal autorizado modificar su estado actual y acceder a información que le compete.



RFT10: Consultar tareas

<i>Código</i>	ERS_ TALENTOHUMANO _9
<i>Nombre</i>	Consultar mis tareas
<i>Actor</i>	Personal, Docente
<i>Propósito</i>	Consultar tareas propias del usuario las cuales han sido asignadas previamente, así como permitir su consulta.
<i>Resultado</i>	Mostrar una lista de todas las tareas que estén asociadas al usuario, las cuales podrán ser seleccionadas para obtener mayor información.

[Sección D] = Requisitos del componente Correo

RFC01: Crear y enviar correo

<i>Código</i>	ERS_ CORREO_1
<i>Nombre</i>	Crear y Enviar Correo
<i>Actor</i>	Usuario
<i>Propósito</i>	Permitir el ingreso de la información necesaria, para crear y enviar un correo.
<i>Resultado</i>	El usuario podrá ingresar los datos solicitados y enviar correos a sus contactos. Los datos relacionados incluyen: Destinatario (Para), Asunto, Contenido, archivos adjuntos (no es obligatorio).

RFC02: Listar posibles destinatarios

<i>Código</i>	ERS_ CORREO_2
<i>Nombre</i>	Listar posibles destinatarios
<i>Actor</i>	Usuario
<i>Propósito</i>	Mostrar el listado de contactos o posibles destinatarios asociados al usuario actual del sistema.
<i>Resultado</i>	En cuanto el usuario ingrese el nombre del destinatario y este tenga más de 4 letras el sistema mostrará un listado de posibles destinatarios que coincidan en la palabra escrita (en la casilla de Para), a medida que ingresa más letras del nombre la lista reducirá las posibilidades hasta que el usuario seleccione una de ellas.

RFC03: Clasificar correos

<i>Código</i>	ERS_ CORREO_3
<i>Nombre</i>	Clasificar Correos
<i>Actor</i>	Usuario
<i>Propósito</i>	Clasificar los correos asociados con el usuario de acuerdo a lo siguiente: Si es destinatario del correo (Correos Recibidos), Si es remitente del correo (Correos Enviados), Si éste correo ha sido catalogado previamente como correo destacado (Correos Destacados), si el correo recibido fue previamente eliminado (Correos eliminados). El sistema mostrará en el sub-menú del módulo respectivo, las opciones de clasificación del correo. Nota: Los correos serán eliminados permanentemente del sistema si el remitente y el destinatario comparten la misma decisión.
<i>Resultado</i>	El sistema clasificará los correos registrados en el sistema según



	características básicas (De, Para, Destacado, Eliminado).
--	---

RFC04: Listar correos

<i>Código</i>	ERS_CORREO_4
<i>Nombre</i>	Listar Correos
<i>Actor</i>	Usuario
<i>Propósito</i>	Por cada tipo de correo (Recibidos, Enviados, Clasificados, Eliminados, Destacados.) el usuario podrá ver el listado de los mismos con acceso directo mediante el sub-menú para listar los correos del tipo seleccionado.
<i>Resultado</i>	Al seleccionar cada alternativa el sistema mostrará el listado que corresponde a la categoría seleccionada la cual estará conformada por un máximo de ___ correos, indicando el número total de correos listados. Cada fila tendrá columnas de información básica (Check Para seleccionar, Condición de Revisado, Archivo Adjunto, Remitente, Destacado, Asunto, Fecha.)

RFC05: Ver Correo seleccionado

<i>Código</i>	ERS_CORREO_5
<i>Nombre</i>	Ver Correo seleccionado
<i>Actor</i>	Usuario
<i>Propósito</i>	El usuario podrá ver el contenido y archivos adjuntos relacionados con el correo previamente seleccionado.
<i>Resultado</i>	El sistema permite seleccionar un correo en particular del listado actualmente disponible y a continuación debe mostrar los datos relacionados a dicho correo (Remitente, Asunto, Fecha, Archivos Adjuntos, Contenido).

RFC06: Eliminar correo desde el listado

<i>Código</i>	ERS_CORREO_6
<i>Nombre</i>	Eliminar correo desde el listado
<i>Actor</i>	Usuario
<i>Propósito</i>	Al seleccionar el check de uno o varios de los correos del listado actual (Recibidos, Destacados, Eliminados, Enviados) y dar click en la opción "Eliminar Correos Seleccionados", estos serán borrados.
<i>Resultado</i>	El sistema borrará los correos que sean seleccionados una vez se seleccione la opción "Eliminar Correos Seleccionados". Nota: Si el listado actual es "Eliminados" no se mostrarán nuevamente al usuario. Para las otras 3 categorías dichos correos serán trasladados al tipo "Eliminados" donde se alojarán por ___ días.

RFC07: Eliminar correo durante revisión del mismo

<i>Código</i>	ERS_CORREO_7
<i>Nombre</i>	Eliminar correo durante revisión del mismo
<i>Actor</i>	Usuario
<i>Propósito</i>	Borrar un correo durante la revisión del mismo.
<i>Resultado</i>	Al momento de revisar un correo en particular el sistema mostrará la opción para borrar dicho correo.



RFC08: Crear grupos de usuarios

<i>Código</i>	ERS_CORREO_8
<i>Nombre</i>	Crear grupos de usuarios
<i>Actor</i>	Administrador
<i>Propósito</i>	Crear categoría de usuarios de acuerdo a características que compartan (ejemplo: pertenecen a un curso en particular), para facilitar la selección del destinatario.
<i>Resultado</i>	Se puede crear un grupo de usuarios que hagan parte de un grupo determinado. Cada usuario podrá hacer parte de varios grupos.

RFC09: Administrar permisos para envío de correos

<i>Código</i>	ERS_CORREO_9
<i>Nombre</i>	Administrar permisos para envío de correos
<i>Actor</i>	Administrador
<i>Propósito</i>	Asignar permisos según perfiles de usuario para enviar correo
<i>Resultado</i>	El usuario podrá definir los permisos necesarios para circulación de correos en base a las relaciones entre usuarios.

[Sección E] = Requisitos del componente Académico

RFA01 Mantenimientos Generales

<i>Código</i>	ERS_ACADEMICO_1
<i>Nombre</i>	Mantenimientos Generales
<i>Actor</i>	Administrador
<i>Propósito</i>	Permitir el ingreso de la información necesaria para crear, modificar y eliminar los registros de datos requeridos y que estén asociados a una entidad (BD) tales como: Áreas, Materias, Proyectos, Logro-indicadores.
<i>Resultado</i>	El usuario puede realizar dichos mantenimientos a la información según lo requiera, en cada uno de los casos podrá actualizar la información en base al uso de formularios diseñados para tales tareas. A su vez podrá buscar y/o listar registros en base a campos de búsqueda propuestos.

RFA02 Asignar director de área y listarlos

<i>Código</i>	ERS_ACADEMICO_2
<i>Nombre</i>	Asignar director de área
<i>Actor</i>	Administrador
<i>Propósito</i>	Asignar un director por cada área, al momento de crearla o después de ello. Este usuario tendrá un rol particular a fin de realizar tareas especiales. Nota: el director debe estar previamente registrado.
<i>Resultado</i>	El usuario podrá asignar un director por área a partir de un registro que asocia el docente con éste rol. De éste modo se asume un rol diferente y nuevas tareas particulares

RFA03 Asignar profesor de Materia y listarlos

<i>Código</i>	ERS_ACADEMICO_3
<i>Nombre</i>	Asignar profesor de Materia
<i>Actor</i>	Administrador
<i>Propósito</i>	Asignar un profesor para dictar materia por curso, al momento de crearla o después de ello. Nota: el profesor debe estar previamente registrado.



<i>Resultado</i>	El usuario podrá asignar un profesor por materia de un curso específico
------------------	---

RFA04 Asignar director de grupo y listarlos

<i>Código de requisito</i>	ERS_ ACADEMICO_4
<i>Nombre</i>	Asignar director de grupo
<i>Actor</i>	Administrador
<i>Propósito</i>	Asignar un director de grupo por curso, al momento de crearlo o después de ello. Nota: el profesor debe estar previamente registrado y debe dictar al menos una materia al curso.
<i>Resultado</i>	El usuario podrá asignar un director de grupo a un curso específico

RFA05 Agregar Horarios de Clases

<i>Código</i>	ERS_ ACADEMICO_5
<i>Nombre</i>	Agregar Horarios de Clases
<i>Actor</i>	Administrador
<i>Propósito</i>	El sistema permite seleccionar un grupo de estudiantes de acuerdo a sede, jornada, grado y curso, e ingresar información sobre su correspondiente Horario de Clases.
<i>Resultado</i>	En cuanto sea seleccionado un grupo de estudiantes el sistema habilitará al usuario actual del sistema para agregar información necesaria en el formato de registro de horario el cual incluye información básica como días de la semana, horario, salón y materia.

RFA06 Visualizar Horario de Clases Seleccionado

<i>Código</i>	ERS_ ACADEMICO_6
<i>Nombre</i>	Visualizar Horario de Clases Seleccionado
<i>Actor</i>	Administrador
<i>Propósito</i>	El sistema permite seleccionar un Horario de Clases específico, a partir del ingreso de unos datos de búsqueda (profesor o curso), a continuación se deben mostrar los datos relacionados a dicho horario incluyendo información básica como profesor por materia, asignación de horas por jornada.
<i>Resultado</i>	Mostrar la información actual relacionada con el horario de clases para el grupo indicado según los datos de búsqueda de grupo (sede, jornada, grado y curso), en forma de horario convencional (en función de horas, días, materias, salón) y fácil de comprender para los usuarios.

RFA07 Registro de Observaciones

<i>Código</i>	ERS_ ACADEMICO_7
<i>Nombre</i>	Registro de Observaciones
<i>Actor</i>	Docente
<i>Propósito</i>	Permitir el ingreso de la información necesaria para crear, modificar y eliminar observaciones generales asociadas al comportamiento y rendimiento académico de un estudiante en particular, según el periodo y la materia que el docente dicte.
<i>Resultado</i>	El usuario podrá registrar, actualizar o modificar observaciones, dicha información servirá de complemento al informe de notas que se genera



	en cada periodo.
--	------------------

RFA08 Registro y Control de Notas

<i>Código</i>	ERS_ ACADEMICO_8
<i>Nombre</i>	Registro y Control de Notas
<i>Actor</i>	Docente
<i>Propósito</i>	Permitir el ingreso de la información necesaria para crear, modificar y eliminar notas académicas de un estudiante en particular, según el periodo y la materia que el docente dicte.
<i>Resultado</i>	El usuario podrá registrar, actualizar o modificar observaciones, dicha información en función del desempeño académico del estudiante.

RFC09: Administrar permisos por usuario

<i>Código</i>	ERS_ ACADEMICO_9
<i>Nombre</i>	Administrar permisos por usuario
<i>Actor</i>	Administrador
<i>Propósito</i>	Asignar permisos según perfiles de usuario para realizar actividades relacionadas con el registro y control de notas y subirlas al sistema.
<i>Resultado</i>	El usuario podrá definir los permisos necesarios para registrar, actualizar, carga masiva, eliminar y subir notas al sistema.

RFC10: Generar boletín de notas

<i>Código</i>	ERS_ ACADEMICO_10
<i>Nombre</i>	Administrar permisos por usuario
<i>Actor</i>	Administrador
<i>Propósito</i>	Crear boletín informativo recopilando observaciones, inasistencias y notas por cada estudiante en un periodo determinado.
<i>Resultado</i>	El usuario podrá generar el informe de notas de uno o más estudiantes, según los datos solicitados por el sistema

3.2 Requisitos no funcionales

1 Disponibilidad

El sistema siempre debe estar disponible, por lo menos en un 100% para los administrativos y docentes de la institución.

2 Portabilidad

El software será desarrollado orientado a web, lo cual permitirá una mayor adaptabilidad entre plataforma, mayor facilidad de acceso, entre otras ventajas.

3.3 Requisitos de Rendimiento

La aplicación soportará la conexión de varios usuarios de manera simultánea, permitiendo la actualización de información en tiempo real. Cada usuario tendrá un único rol en caso de realizar, podrán crearse más roles en caso de tener tareas especiales. Recuerde que la asignación de permisos para acceso a tareas específicas se realiza por rol de usuario. Si se comparten roles no



compatible (por ejemplo acudiente – docente), podrá manejarse un usuario por cada rol.

3.4 Restricciones de Diseño

Debe considerarse lo establecido en el test general para lista de comprobaciones (Anexo Y).

3.5 Atributos del Sistema

1. El registro de usuarios con perfil de privilegios administrativos se realizará solamente por el administrador del sistema
 2. El acceso al sistema será mediante validación de login y password.
 - 3.
 4. Los estudiantes no podrán modificar los datos relacionados con su información académica
 5. El sistema correrá tanto en plataformas Windows como en Linux.
- Faltan....

3.6. Otros Requisitos

Idioma

Los títulos, etiquetas, mensajes, y demás contenidos que hacen parte del prototipo deben estar en español. Para efectos particulares generados por la interacción del producto y la base de datos, estos mensajes deben ser entendibles o añadir las etiquetas necesarias en español para que el usuario pueda enterarse de lo que ocurre.



Personal involucrado

Nombre	Laura Pulido Díaz
Rol	Planeación, diseño, desarrollo y presentación del proyecto.
Categoría profesional	Estudiante de Ingeniería de Sistemas.
Responsabilidades	Diseño del plan de trabajo de Grado Ejecución del plan de trabajo de Grado
Correo Electrónico	laura.pulido.diaz@gmail.com
Aprobación	
Nombre	Luz Dary Méndez Uribe
Rol	Planeación, diseño, desarrollo y presentación del proyecto.
Categoría profesional	Estudiante de Ingeniería de Sistemas.
Responsabilidades	Diseño del plan de trabajo de Grado Ejecución del plan de trabajo de Grado
Correo Electrónico	luzdame9@gmail.com
Aprobación	
Nombre	Emiro Muñoz Jerez
Rol	Dirección del trabajo de Grado
Categoría profesional	Ingeniero de Sistemas Profesor Cátedra Escuela de Ingeniería de Sistemas
Responsabilidades	Asesoría para la ejecución del trabajo de Grado Control de la ejecución del proyecto
Correo Electrónico	emiro270273@gmail.com
Aprobación	