

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 8



Sistema De Gestión Extradocente

*Trabajo de Diploma para Optar por el Título de Ingeniero en
Ciencias Informáticas*

Autores

Yadira Cervantes García

Yuri Giralt Barrios

Tutor

Ing. Eduardo Alfonso Sánchez

Ciudad de La Habana, junio 2010

“Año 52 de la Revolución”



“ El hombre debe transformarse al mismo tiempo que la producción progresa, no realizaríamos una tarea adecuada, si fuéramos tan solo productores de artículos, de materias primas y no fuéramos al mismo tiempo productores de hombres. ”

Ernesto Che Guevara

Declaración de auditoría

Declaración de auditoría

Declaramos que somos los únicos autores del trabajo "Sistema de Gestión Extradocente y autorizamos a la Facultad 8 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autores:

Yadira Cervantes García

Nombre y apellidos del autor

Yuri Giralt Barrios

Nombre y apellidos del autor

Tutor:

Ing. Eduardo Alfonso Sánchez

Nombre y apellidos del tutor

Agradecimientos

Agradecimientos

A Fidel y a la Revolución por darnos la maravillosa oportunidad de estudiar en esta universidad y de crecer como hombres libres.

A la UCI por formarnos como profesionales y haber permitido nuestra superación durante estos años.

A todos los profesores que nos han formado en estos 5 años de universidad.

A nuestro tutor por habernos guiado y por haber compartido su investigación y confiado en nosotros.

Yadira y Yuri

La realización de este trabajo no hubiera sido posible sin la colaboración de un grupo de personas, para ellos mis más sinceros agradecimientos:

A mis padres Ramona y Vismark y a mi hermanita Yamilee, los quiero mucho.

A mis abuelos, los quiero mucho.

A toda familia por parte de padre, que aunque estén lejos siempre los recuerdo.

A mis tías y tíos Luisito y Armando los quiero mucho.

A mi prima Mayelín que siempre me ha apoyado y ayudado en todo momento y a mi primita Marien las quiero mucho.

A mis grandes amigos Daylen, Elizabeth y por sobre todo a mi amigo de toda la vida Jorgito, siempre los voy a querer y a extrañar mucho.

A Alejandro por estar siempre ahí cuando lo necesité.

A mis vecinos de La Palma por estar siempre ahí.

A mis compañeros de grupo tanto el antiguo grupo como al nuevo.

A todos mis amigos de la Universidad y a todo el que colaboró con el desarrollo de esta investigación y de mi carrera.

A mi compañero de tesis Yuri por estar siempre ahí.

Yadira

Este trabajo no es solamente el fruto de mi esfuerzo, es la prueba tangible de la colaboración de muchas personas, a las cuales quiero agradecer:

A mis padres, por haber estado siempre conmigo y haberme formado tal cual soy, por haberme mantenido durante todos estos años de estudio y por su constante sacrificio por mi felicidad.

A mi compañera de tesis Yadira Cervantes García por estar pendiente, por su incansable labor, por ser el alma de nuestro equipo.

A Raciel Yera Toledo a José Ernesto Lara Rodríguez a Alejandro Medina Mejías y a Alejandro Morales Torres por su ayuda incondicional, aprendí mucho de ellos.

A todos mis amigos por estar conmigo en los momentos más trascendentales y en los más triviales.

Yuri

Dedicatoria

Dedicatoria

A los tres amores más grandes de mi vida: mi mamá, mi papá y mi hermana, por todo su amor y cariño, por guiarme durante toda mi vida estudiantil, por todo su sacrificio incondicional y darme lo mejor para seguir adelante y hacer de mí lo que hoy soy.

A mis abuelos y en especial a mi abuelo Luis por saber que lo iba a lograr y por haber estado siempre ahí y apoyarme en todo, te extraño mucho.

Yadira

Le dedico este trabajo a mis padres, a mi hermano y a mis abuelos, en especial a Domingo Abigaíl Barrios Machado por ser un ejemplo y una meta.

Yuri

Resumen

En la Universidad de las Ciencias Informáticas se fomentan una serie de actividades extradocentes, pero actualmente en la facultad se hace cada vez más necesario contar con un sistema que permita tanto al estudiantado como al conjunto de trabajadores del centro conocer cuáles son las actividades realizadas, así como las participaciones y resultados obtenidos en cada una de ellas por parte de los estudiantes. Para hacer este proceso un poco más rápido y eficaz se requerirá de información contenida en la facultad así como de información que tenga los estudiantes de la FEU encargados de cada una de estas actividades extradocentes. Dentro de ellas se encuentran: jornadas martianas, mi Web por Cuba, Copa Pascal, festivales de artistas aficionados, encuentros de conocimientos, juegos deportivos entre otras.

Con el presente trabajo titulado “Sistema de Gestión Extradocente”, se pretende tener registrada la trayectoria del estudiante en el plano extradocente para contribuir a la formación de un profesional más integral. Para lograr este objetivo se realizó un estudio minucioso de las herramientas más utilizadas en el mundo para este fin, así como las metodologías de desarrollo del software que guían los procesos de desarrollo, se utilizó como metodología de desarrollo Agile Unified Process, el Lenguaje Unificado de Modelado para la especificación, construcción y documentación del sistema y Visual Paradigm para la creación de los artefactos que se generan durante el ciclo de vida del software. Se utilizó como lenguaje de programación java, el entorno de desarrollo integrado Eclipse, como sistema gestor de base de datos Postgres y como framework Jsf, Spring e Hibernate.

Palabras claves: Sistema, gestión, procesos, casos de uso, modelamiento, análisis.

Índice

Tabla de contenido

Introducción.....	1
Capítulo 1: Fundamentación teórica.....	5
1.1 ¿Qué es la gestión de información? :	5
1.2: Características de los sistemas de información	6
1.3: Sistemas de gestión de información en el mundo.....	6
1.4: Sistema de gestión de información en Cuba.....	7
1.5: Las TIC en la UCI.....	8
1.6: Lenguajes, metodologías y herramientas a utilizar	8
1.6.1: Metodologías de desarrollo de software.....	8
Rational Unified Process (RUP).....	9
Extreme Programming (XP).....	11
Agile Unified Process (AUP).....	13
1.6.2: Análisis de la selección de la Metodología de Desarrollo.....	15
1.7 Lenguaje de modelado	15
1.8: Herramientas de Desarrollo.....	16
Rational Rose.....	17
Visual Paradigm (VP)	18
1.8.1 Análisis de la selección de la herramienta Case.....	18
1.9: Fundamentación del lenguaje de programación a utilizar.....	19
Java	19
PHP.....	20
1.9.1: Análisis de la selección del lenguaje de programación a seleccionar	21
1.10: Plataforma de desarrollo.....	21
1.11: Entorno de Desarrollo.....	24
Eclipse.....	24
NetBeans.....	24
1.11.1: Análisis de la selección del entorno de desarrollo.....	25
1.12: Sistema Gestor de Base de Datos.....	26

MySQL	26
PostgreSQL	26
1.12.1: Análisis de la selección del Sistema Gestor de Base de Datos	27
1.13: Frameworks	27
Presentación	28
Lógica del negocio.....	29
Acceso a datos	29
1.14: Arquitectura Técnica	30
1.15: Conclusiones	31
Capítulo 2: Modelamiento del negocio y requisitos.....	32
2.1: Estado actual de los procesos involucrados en el campo de acción.....	32
2.2: Propuesta de solución	33
2.3: Modelamiento del negocio	34
2.4: Técnicas utilizadas para comprender el negocio	34
2.5: Captura de requisitos	34
2.6: Procesos del negocio	35
2.7: Actores y trabajadores del negocio.....	36
2.8: Modelo de casos de uso.....	37
2.9: Especificaciones de los casos de uso y diagrama de actividades.....	38
2.10: Especificación de los requisitos del software	40
2.11: Requerimientos no funcionales.....	42
2.12: Conclusiones	45
Capítulo 3: Análisis y diseño del sistema.....	46
3.1: Actores del sistema y justificación	46
3.2: Modelo de casos de uso del sistema	47
3.3: Modelo del análisis.....	51
3.4:Clases del análisis.....	52
3.5:Descripción de los casos de uso del sistema	53
3.6:Diagramas de clases del análisis.....	55
3.7: Diagramas de colaboración	56
3.8: Diseño.....	59
3.9: Clases significativas de la arquitectura	59
3.10: Patrones.....	60

3.11: Diagramas de clases del diseño	60
3.12: Conclusiones	62
Capítulo 4: Implementación y pruebas	63
4.1: Modelo de implementación	63
4.2.1: Diagrama de subsistemas de implementación	63
4.2: Diagrama de componentes.....	64
4.3: Diagrama de despliegue.....	64
4.4: Pruebas.....	65
4.5: Métodos de Pruebas.....	65
4.6: Tipos de Pruebas	65
Prueba de funcionalidad	66
4.7: Niveles de pruebas	66
4.8: Conclusiones	67
Conclusiones.....	68
Recomendaciones.....	69
Bibliografía Referenciada	70
Bibliografía	72
Glosario de términos.....	73
Anexos.....	74
Anexo 1: Ejemplo de EyeOS	74
Anexo 2: Vista de la ventana principal del Sistema.	75
Anexo 3: Ejemplo del uso de las TIC en la UCI "Akademos".	75
Anexo 4: Ejemplo del uso de las TIC en la UCI" Entorno Virtual de Aprendizaje (EVA)".....	76
Anexo 5: Caso de uso Registrar participación en juegos inter-años	76
Anexo 6: Caso de uso Registrar participación en juegos inter-facultades.....	77
Anexo 7: Caso de uso Audicionar.....	77
Anexo 8: Caso de uso Registrar participación en Actividades de Conocimiento (AC).	78
Anexo 9: Caso de uso Registrar participación en Mi Web x Cuba (inter-facultad).	79
Anexo 10: Caso de uso Registrar participación en Jornada Científica Estudiantil (inter-facultad).	79
Anexo 11: Caso de uso Registrar participación en Seminario Juvenil Martiano (inter-facultad).80	
Anexo 12: Caso de uso Registrar participación en Copa Pascal (inter-facultad).	80
Anexo 13: Caso de uso Registrar participación en Copa ISW (inter-facultad).....	81
Anexo 14: Caso de uso Registrar participación en Copa Java.	81

<i>Anexo 15: Diagrama de objetos.....</i>	<i>82</i>
<i>Anexo 16: Diagrama de actividades Registrar participación en Actividades de Conocimiento. .</i>	<i>82</i>
<i>Anexo 17: Caso de uso Autenticar usuario.</i>	<i>83</i>
<i>Anexo 18: Caso de uso Gestionar usuario.</i>	<i>86</i>
<i>Anexo 19: Caso de uso Gestionar estudiante.</i>	<i>88</i>
<i>Anexo 20: Caso de uso Gestionar AC.....</i>	<i>90</i>
<i>Anexo 21: Caso de uso Gestionar participación en AC.....</i>	<i>92</i>
<i>Anexo 22: Caso de uso Gestionar resultados en AC.....</i>	<i>94</i>
<i>Anexo 23: Caso de uso Realizar búsqueda de estudiante.</i>	<i>94</i>
<i>Anexo 24: Caso de uso Gestionar reunión.</i>	<i>96</i>
<i>Anexo 25: Diagrama del diseño Gestionar AC.....</i>	<i>96</i>
<i>Anexo 26: Diagrama del diseño Gestionar deporte.</i>	<i>97</i>
<i>Anexo 27: Diagrama del diseño Gestionar reunión.</i>	<i>97</i>
<i>Anexo 28: Caso de prueba Gestionar estudiante.....</i>	<i>99</i>
<i>Anexo 29: Caso de prueba Gestionar actividad.</i>	<i>101</i>
<i>Anexo 30: Caso de prueba Gestionar usuario.....</i>	<i>103</i>
<i>Anexo 31: Diagrama de componentes paquete Brigada (común).....</i>	<i>104</i>
<i>Anexo 32: Diagrama de componentes paquete Brigada (control de actividades).</i>	<i>105</i>
<i>Anexo 33: Diagrama de componentes paquete Brigada (reuniones).</i>	<i>106</i>

Índice de figuras

<i>Figura 1: Los distintos flujos de trabajo en el ciclo de vida de RUP.</i>	<i>11</i>
<i>Figura 2: Fases de XP.....</i>	<i>13</i>
<i>Figura 3: Ciclo de vida de AUP.....</i>	<i>13</i>
<i>Figura 4: Elementos de Java</i>	<i>22</i>
<i>Figura 5: Ciclo de vida bajo la tecnología JSF.</i>	<i>28</i>
<i>Figura 6: Diagrama de casos de uso del negocio.....</i>	<i>38</i>
<i>Figura 7: Diagrama de actividades.</i>	<i>40</i>
<i>Figura 8: Diagrama de casos de uso del sistema.....</i>	<i>47</i>
<i>Figura 9: Paquete general.</i>	<i>48</i>
<i>Figura 10: Paquete actividades culturales.</i>	<i>48</i>
<i>Figura 11: Paquete actividades deportivas.</i>	<i>49</i>
<i>Figura 12: Paquete actividades de conocimiento.</i>	<i>49</i>
<i>Figura 13: Paquete administración.....</i>	<i>50</i>
<i>Figura 14: Paquete Gestionar orden del día.....</i>	<i>50</i>
<i>Figura 15: Paquete Gestionar reunión.</i>	<i>51</i>
<i>Figura 16: Clases del análisis.</i>	<i>52</i>
<i>Figura 17: DCA. CU del sistema Gestionar participación en Actividades Culturales.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Figura 18: DCA. CU del sistema Gestionar resultados en Actividades Culturales.....</i>	<i>55</i>
<i>Figura 19: DCA. CU del sistema Gestionar estudiante.</i>	<i>56</i>
<i>Figura 20: DCA. CU del sistema Gestionar usuario.</i>	<i>¡Error! Marcador no definido.</i>
<i>Figura 21: DCA. CU del sistema Gestionar AC.</i>	<i>¡Error! Marcador no definido.</i>
<i>Figura 22: DCA. CU del sistema Gestionar participación en AC.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Figura 23: DCA. CU del sistema Gestionar resultado en AC.</i>	<i>56</i>
<i>Figura 24: DCA. CU del sistema Gestionar deporte.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Figura 25: DCA. CU del sistema Gestionar participación en juegos.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Figura 26: DCA. CU del sistema Gestionar resultado en juegos.</i>	<i>¡Error! Marcador no definido.</i>
<i>Figura 27: DCA. CU del sistema Autenticar usuario.....</i>	<i>56</i>
<i>Figura 28: Diagrama de colaboración del sistema Gestionar participación en actividades culturales (registrar).</i>	<i>57</i>
<i>Figura 29: Diagrama de colaboración del sistema Gestionar participación en actividades culturales (eliminar).</i>	<i>57</i>
<i>Figura 30: Diagrama de colaboración del sistema Autenticar usuario.....</i>	<i>57</i>

Figura 31: Diagrama de colaboración del sistema Gestionar usuario (registrar).....	58
Figura 32: Diagrama de colaboración del sistema Gestionar usuario (modificar).	58
Figura 33: Diagrama de colaboración del sistema Gestionar usuario (eliminar).	58
Figura 34: Diagrama de colaboración del sistema Gestionar estudiante (registrar).....	58
Figura 35: Diagrama de colaboración del sistema Gestionar estudiante (modificar).	59
Figura 36: Diagrama de colaboración del sistema Gestionar estudiante (eliminar).	59
Figura 37: Diagrama de colaboración del sistema Gestionar AC (registrar).. ¡Error! Marcador no definido.	
Figura 38: Diagrama de colaboración del sistema Gestionar AC (modificar). ¡Error! Marcador no definido.	
Figura 39: Diagrama de colaboración del sistema Gestionar AC (eliminar)... ¡Error! Marcador no definido.	
Figura 40: Diagrama de colaboración del sistema Gestionar participación en AC (registrar). ¡Error! Marcador no definido.	
Figura 41: Diagrama de colaboración del sistema Gestionar participación en AC (eliminar). ¡Error! Marcador no definido.	
Figura 42: Diagrama de colaboración del sistema Gestionar resultados en AC (registrar). ¡Error! Marcador no definido.	
Figura 43: Diagrama de colaboración del sistema Gestionar resultados en AC (modificar). ¡Error! Marcador no definido.	
Figura 44: Diagrama de colaboración del sistema Gestionar Deporte (registrar)..... ¡Error! Marcador no definido.	
Figura 45: Diagrama de colaboración del sistema Gestionar Deporte (modificar). ¡Error! Marcador no definido.	
Figura 46: Diagrama de colaboración del sistema Gestionar Deporte (eliminar). ¡Error! Marcador no definido.	
Figura 47: Diagrama del diseño Autenticar usuario.	61
Figura 48: Diagrama del diseño Gestionar estudiante.....	61
Figura 49: Diagrama del diseño Realizar búsqueda de estudiante.....	62
Figura 50 : Diagrama de Subsistema de implementación.....	63
Figura 51: Diagrama de componentes paquete Administración.	64
Figura 52: Diagrama de Despliegue.....	65

Índice de tablas

<i>Tabla 1: Actores del negocio y descripción.....</i>	<i>36</i>
<i>Tabla 2: Trabajadores y descripción.</i>	<i>37</i>
<i>Tabla 3: Descripción de CU Registrar participación en festivales.....</i>	<i>40</i>
<i>Tabla 4: Actores del sistema.....</i>	<i>46</i>
<i>Tabla 5: Descripción del CU Gestionar participación en actividades culturales.</i>	<i>55</i>

Introducción

En este momento del desarrollo científico - técnico, todo profesional a nivel mundial tiene una gran necesidad de información, vivimos en un mundo, donde a cada minuto, se producen artículos de todo tipo de especialidades, que es imposible seguir, leer, incorporar. Cuba no está ajena al progresivo y constante cambio de la era actual, como consecuencia del desarrollo vertiginoso de las tecnologías de la información y las comunicaciones. En nuestro país se han ido incorporando los últimos avances tecnológicos para manejar toda la información que se genera tanto en el sector productivo como en el científico, en la educación, la salud, el deporte entre otros.

La Universidad de las Ciencias Informáticas es un digno ejemplo del desarrollo informático en nuestro territorio, a diario en nuestro centro se mueve una inmensa cantidad de información para importantes funciones. La Federación Estudiantil Universitaria, como principal organización universitaria frecuentemente se ve entorpecida en su labor por la carencia de un mecanismo que optimice su funcionamiento gestionando la información necesaria.

La idea de este proyecto surge por la creciente necesidad de un registro histórico de cada estudiante a la hora de realizar una valoración general del estudiantado, teniendo en cuenta cada una de las actividades en las que participa, ya sea en un curso o durante toda su carrera. Cada vez es más difícil el trabajo de organización, control y atención de las personas que cada año van formando parte de esta universidad. En estos momentos la labor de control de la participación y los resultados de los estudiantes se realiza de forma manual y muchas veces por apreciación del mismo estudiante sin poder asegurar si es cierta o no la información recibida, esto trae consigo un trabajo engorroso, provocando la pérdida y deterioro de la información, o simplemente no se lleva. La búsqueda y actualización de la información es más difícil y consume más tiempo; no hay forma de acceder a ella cuando se le necesita, por ejemplo a la hora de dar un perfil más completo de un estudiante en la Federación Estudiantil Universitaria para realizar los procesos de integralidad, la entrega de los Premios Mella u otros reconocimientos

Dada esta situación, **el problema científico** de la presente investigación se plantea en los siguientes términos: ¿Cómo contribuir en el proceso de gestión de la información relacionada con la participación y los resultados de los estudiantes en las actividades extradocentes en la Universidad de las Ciencias Informáticas?

Se define como **objetivo general**: Desarrollar un sistema que permita gestionar la información relacionada con la participación y los resultados de los estudiantes en las diferentes actividades extradocentes.

Para darle cumplimiento al objetivo general se plantean los siguientes **objetivos específicos**:

- Realizar y conceptualizar el negocio para el Sistema de Gestión Extradocente.
- Realizar una captura y análisis de los requisitos que debe tener el sistema.
- Diseñar e implementar el sistema de forma tal que responda a los requisitos funcionales y no funcionales.
- Aprobar la propuesta de solución haciendo uso de los métodos de validación de software.

Teniendo como **objeto de estudio**: Los procesos de Gestión de eventos extradocentes.

Enfocando la investigación hacia el **campo de acción**: Actividades extradocentes que realizan los estudiantes de la Universidad de las Ciencias Informáticas.

Mencionados los objetivos específicos se realizarán las siguientes **tareas investigativas**:

- Realizar un estudio del negocio.
- Realizar la captura de requisitos.
- Realizar un estudio de las principales herramientas y las tecnologías existentes para seleccionar las más adecuadas para el desarrollo del sistema.
- Realizar análisis, diseño e implementación de las funcionalidades.
- Analizar y/o implementar un proceso que garantice los principios básicos de la seguridad de la información.
- Efectuar pruebas al sistema después de concluido el mismo.

Teniendo además como **idea a defender**:

Si se realiza el análisis, diseño e implementación de un sistema que gestione la información relacionada con la participación y los resultados de los estudiantes en las diferentes actividades extradocentes se podrá tener un mayor control de cada una de las mismas.

Métodos Científicos.

Teóricos:

- Analítico-sintético: Está dado por el análisis de los documentos generados en el levantamiento y captura de requisitos, extrayendo y analizando los principales elementos relacionados con el objeto de estudio
- Análisis histórico lógico: Nos permite estudiar de forma analítica la trayectoria histórica real de los fenómenos, su evolución y desarrollo además de constatar teóricamente cómo ha evolucionado un determinado fenómeno en un período de tiempo, en toda su trayectoria o en un fragmento temporal de la lógica de su desarrollo, es decir, para el estudio de la evolución y desarrollo de la gestión eventos extradocentes.
- Modelación: Este método nos permite la creación de modelos, (propuestas, alternativas, estrategias). El modelo es una reproducción simplificada de la realidad, cumple una función heurística, que permite descubrir y estudiar nuevas relaciones y cualidades del objeto de estudio. La modelación es justamente el proceso mediante el cual creamos modelos con vistas a investigar la realidad, es decir, para representar de manera funcional y gráfica la aplicación propuesta. Esto se usa cuando a partir de los requisitos capturados se realiza un modelo del sistema a implementar.

Empíricos:

- Entrevista: Se utilizó porque es el más eficiente para obtener información, es decir, conocer las necesidades de los clientes y problemas a resolver, además de conocer cómo se lleva a cabo la gestión de información de las actividades extracurriculares para una mejor comprensión del problema. La entrevista puede ser individual o colectiva, en ambos casos el entrevistador debe realizar una preparación previa, sobre el tema a tratar, y elaborar una guía para su desarrollo. La entrevista consume mucho tiempo y para su realización se debe buscar información sobre las características del entrevistado y su posición en el tema a tratar.

El trabajo se dividió en capítulos que contienen la información referente a la investigación realizada.

- Capítulo I: Fundamentación teórica: Se expone la fundamentación del tema a tratar. Se exponen elementos asociados a los principales conceptos relacionados con el dominio del problema, que son necesarios para entender el negocio para la gestión de actividades extradocentes. Así como las tendencias y tecnologías actuales más usadas que son consideradas para la elaboración del sistema.
- Capítulo II: Modelamiento del negocio y requisitos: Se plantea el modelo del negocio, los requisitos agrupados en funcionales y no funcionales y casos de uso del sistema algunos de los diagramas de desarrollo, así como la expansión de los casos de uso.
- Capítulo III: Análisis y diseño: Se exponen los detalles relacionados con el análisis del sistema propuesto, donde se utilizará para su modelado los diagramas de clase de los casos de uso del sistema.
- Capítulo IV: Implementación y prueba: Se hace todo lo de la implementación del sistema y lo relacionado con las pruebas.

Capítulo 1: Fundamentación teórica

Capítulo 1: Fundamentación teórica

En este capítulo se darán a conocer los aspectos relacionados con la universidad, su estructura organizativa, eventos y acontecimientos que se desarrollan en la misma de forma extradocente y cómo se está llevando a cabo su proceso de informatización. Se exponen elementos asociados a los principales conceptos relacionados con el dominio del problema, que son necesarios para entender el negocio de la gestión de las actividades extradocentes. Así como las tendencias y tecnologías actuales más usadas que son consideradas para la elaboración del sistema.

1.1 ¿Qué es la gestión de información? :

Comprende las actividades relacionadas con la obtención de la información adecuada, a un precio adecuado, en el tiempo y lugar adecuado, para tomar la decisión adecuada. **(Quiroga, 2002)**

También decir que es el conjunto de actividades realizadas para controlar, almacenar y después recuperar la información producida, retenida y recibida por cualquier organización. Son todos los procesos que llevan asociados la recuperación, búsqueda, producción, y almacenamiento de información.

Para poder utilizar la información para tomar decisiones de gestión, hay que gestionar la información (recabar, almacenar y analizar). La información para la gestión y la gestión de la información son dos conceptos diferentes; la información para la gestión es un tipo de información (los datos); la gestión de la información es un tipo de gestión (el sistema), aunque ellas son diferentes, se refuerzan mutuamente y no pueden separarse en las operaciones diarias. Por lo tanto, la gestión de la información implica; determinar la información que se precisa, recoger y analizar la información, registrarla y recuperarla cuando sea necesaria, utilizarla y divulgarla. Determinar la información necesaria para la gestión: Durante la planificación, gestión y supervisión del proyecto. Obtener y analizar la información para gestionarla: La información puede conseguirse de informes de técnicos, libros de registro, formularios de los diferentes ejecutantes, reuniones con la comunidad, entrevistas, observación y mapas comunitarios. Para que la información tenga un uso adecuado tiene que compartirse con los demás interesados o usuarios. Esta información puede ayudarles en sus decisiones de gestión y también puede ayudar al que la recoge a encontrar significados o usos relacionados con la gestión. **(Bartle)**

Capítulo 1: Fundamentación teórica

1.2: Características de los sistemas de información

Los sistemas de información están concebidos para capturar, actualizar, integrar, consultar y analizar información pertinente a la organización a la cual pertenecen.

Los sistemas de información no sólo almacenan datos, sino que estos datos deben procesarse y distribuirse.

Entre las principales bondades de los sistemas de gestión de contenidos está permitir que cualquier persona, sin conocimientos avanzados sobre informática, pueda colocar, modificar o eliminar contenidos del sitio. Esto, sin renunciar a los necesarios controles de calidad que exige un sistema de información que se respete.

Gestionan los usuarios que utilizan la información, que además pueden agregarla.

Poseen una interfaz en correspondencia con la información que contienen.

Integran no sólo datos e información, sino también programas y otros sistemas de información.

Ventajas: La implantación de un sistema de gestión de información, siempre es de gran importancia, porque ayuda a dejar atrás el obsoleto modo manual de recopilar la información, de este modo se pueden asignar privilegios para que solo modifique la información el personal autorizado , esto se puede lograr por medio de la autenticación y la validación correcta. Permite acceder a la información de forma rápida y segura. **(González)**

1.3: Sistemas de gestión de información en el mundo

La actualidad está marcada por la unión tecnológica, donde se experimenta un empaste entre la electrónica, la informática, las telecomunicaciones, entre otros, lo cual tiene su máxima expresión en el crecimiento acelerado que ha tomado la Internet. Es aquí donde aparecen conceptos como: "era de la información", "telemática" o "información". Estas tecnologías de la información y las comunicaciones, han sido conceptualizadas como la integración y convergencia de la computación microelectrónica, las telecomunicaciones y la técnica para el procesamiento de datos, en donde, sus principales componentes son: el factor humano, los contenidos de la información, el equipamiento, la infraestructura material, el software y los mecanismos de intercambio electrónico de información, los elementos de política y regulaciones y los recursos financieros. Por ello se hace imprescindible el estudio y dominio de las tecnologías de la información, en cuanto se tienen en cuenta las influencias que estas

Capítulo 1: Fundamentación teórica

transformaciones le imponen al ser humano como ente social, pues tiende a modificar no sólo sus hábitos y patrones de conducta, sino, también, su forma de pensar. Es de esta forma que se desarrolla su uso en la industria, en el sector empresarial, en la salud, la educación, el ocio y en los propios hogares. Hoy en día hay patrones, normas y actitudes, el gusto popular y la moda, en particular la de los jóvenes y adolescentes, son decididos, en mucho de los casos, por el impacto audiovisual y los cánones que difunden las imágenes y mensajes publicitarios de la Internet.

Por todo lo antes expuesto se genera la siguiente reflexión: la sociedad de la información debe ser para todos, orientándola hacia el desarrollo justo, equitativo y sostenible, lo cual obliga a una conciencia mundial que determine la eliminación de la brecha digital y lograr el acceso realmente universal, inclusivo, no excluyente, a las modernas tecnologías de la info-comunicación. **(Nicot, 2008)**

Ejemplo de sistema gestor de información en el mundo: EyeOS, gestor de información personal online.

EyeOS es un escritorio virtual multiplataforma, libre y gratuito, basado sobre el estilo del escritorio de un sistema operativo. El paquete básico de aplicaciones que vienen por defecto, incluye toda la estructura de un sistema operativo y algunas aplicaciones de tipo suite ofimática como un procesador de textos, un calendario, un gestor de archivos, un mensajero, un navegador, una calculadora y más. El paquete entero está autorizado bajo la licencia libre AGPL. (Ver anexo 1).

1.4: Sistema de gestión de información en Cuba

El **Sistema de Gestión Consular** está dirigido a humanizar la tramitación de los numerosos servicios que ofrecen los consulados cubanos en el extranjero, brindar una atención ágil y eficiente al cliente y conservar grandes volúmenes de información con posibilidad de ser consultada, logrando inmediatez en las respuestas.

De forma general, el Sistema registra los datos generales de las personas que solicitan servicios al consulado y genera los correspondientes recibos de pago. Para dar seguimiento a la tramitación de estos servicios, mantiene una comunicación con otros sistemas instalados en Cuba en las entidades que dan respuestas a los mismos, como por ejemplo el Departamento de Inmigración y Extranjería (DIE) o la Dirección de Asuntos Consulares de los Cubanos Residentes en el Exterior (DACCRES). (Ver anexo 2). **(León)**

Capítulo 1: Fundamentación teórica

1.5: Las TIC en la UCI

La llamada ciudad digital, como centro de altos estudios, se ha apoyado en las TIC para llevar a cabo el proceso de aprendizaje de sus estudiantes. Se utilizan diferentes sistemas que llevan a cabo la gestión de información como es:

Akaderos: Es la encargada de recopilar las evaluaciones de cada estudiante desde su primer año universitario hasta el último, permite saber a cada estudiante el acumulado que tiene en cada año así como el general y todos los cursos optativos, en general una valoración del estudiante. (Ver anexo 3).

Entorno Virtual de Aprendizaje (EVA): Por medio de esta aplicación los profesores pueden acceder a ver todas las actividades hechas por los estudiantes y así darle su calificación y los estudiantes también pueden hacer sus tareas y ver sus evaluaciones por asignatura. (Ver anexo 4).

Es mucho lo que se ha avanzado en cuanto al uso de las TIC dentro de la universidad pero aún quedan muchas cosas por automatizar. Mencionar la necesidad de que la universidad cuente con una aplicación donde pueda obtener la información de los estudiantes que participan en los distintos eventos extradocentes producidos dentro de la misma, es por eso que ninguna de estas aplicaciones anteriores cumple con los requisitos que el sistema que se decidió desarrollar debe cumplir.

1.6: Lenguajes, metodologías y herramientas a utilizar

1.6.1: Metodologías de desarrollo de software

Metodología se refiere a los métodos de investigación que se siguen para alcanzar una gama de objetivos en una ciencia. Son vías que facilitan el descubrimiento de conocimientos seguros y confiables para solucionar los problemas que la vida plantea. En resumen son el conjunto de métodos que se rigen en una investigación científica o en una exposición doctrinal. Una metodología de desarrollo del software define quien está haciendo qué, cuándo, y cómo alcanzar un objetivo específico. Esta proporciona normas para el desarrollo eficiente del software con calidad al captar las mejores prácticas que el estado actual de la tecnología permite. **(Estrada)**

A nivel internacional, las instituciones y empresas dedicadas a la industria de software, emplean en su actividad de desarrollo, modelos, metodologías o procedimientos estándares para desarrollar, instalar y mantener un producto de este tipo. El objetivo

Capítulo 1: Fundamentación teórica

de un proceso de desarrollo de este tipo, es elevar la calidad del software (en todas las fases por las que pasa) a través de una mayor transparencia y control sobre el proceso. Da igual el alcance que se desee, hay que producir lo esperado en el tiempo y con el costo esperado. Es labor del proceso de desarrollo hacer que esas medidas para aumentar la calidad sean reproducibles en cada desarrollo. Estas metodologías establecen un conjunto de actividades que definen cómo se debe hacer el software, quién debe hacer cada actividad, cuándo hacerla y qué se debe hacer.

La implantación de un proceso de desarrollo es una labor más a medio-largo plazo que una labor de resultados inmediatos. Cuesta tiempo que los trabajadores se adapten al proceso, pero una vez superado, la inversión se recupera con creces. Es por ello que no tiene sentido ajustarse a un proceso al pie de la letra, sino que hay que adaptarlo a las necesidades y características de cada empresa, equipo de trabajo o casi a cada proyecto.

En los últimos tiempos la cantidad y variedad de los procesos de desarrollo ha aumentado de forma impresionante, sobre todo teniendo en cuenta el tiempo que estuvo en vigor como ley única el famoso desarrollo en cascada. Se podría decir que en estos últimos años se han desarrollado dos corrientes en lo referente a los procesos de desarrollo, los llamados métodos pesados y los métodos ligeros. La diferencia fundamental entre ambos es que mientras los métodos pesados intentan conseguir el objetivo común por medio de orden y documentación, los métodos ligeros (también denominados métodos ágiles), tratan de mejorar la calidad del software por medio de una comunicación directa e inmediata entre las personas que intervienen en el proceso. **(Guevara, 2008)**

Rational Unified Process (RUP)

RUP es uno de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto, y no tan solo de software. Unifica completamente a un equipo de desarrollo de software y optimiza la productividad de cada uno de los miembros del equipo brindándoles la experiencia de los líderes de la industria y las lecciones aprendidas a través de miles de proyectos. Está fundamentada en un enfoque orientado a modelos de desarrollo basado en componentes, utilizando para ello el Lenguaje de Modelado Unificado (UML, Unified Modeling Language) el que define técnicas de análisis y diseño que ayudan a la confección de una solución sólida de software. **(Guevara, 2008)**

Capítulo 1: Fundamentación teórica

RUP se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental.

- **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, pues describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los Casos de Uso relevantes desde el punto de vista de la arquitectura.
- **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración.

Capítulo 1: Fundamentación teórica

Ciclo de vida de RUP:

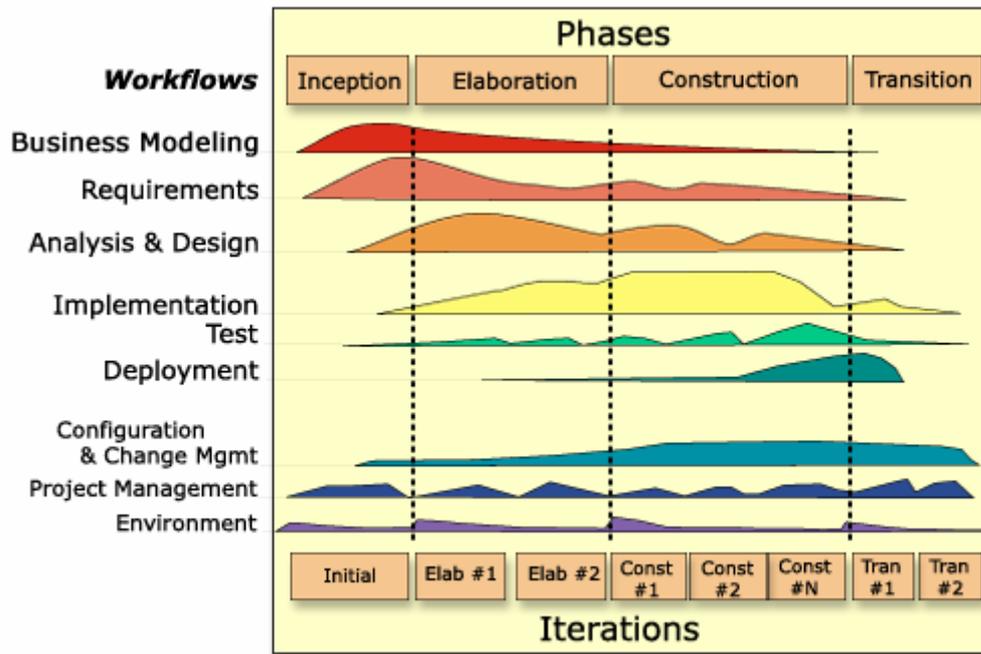


Figura 1: Los distintos flujos de trabajo en el ciclo de vida de RUP.

Ventajas de RUP:

Entre las ventajas que presenta RUP tenemos que reduce riesgos ya que este modelo es iterativo, utiliza componentes, lo cual reduce el tiempo de realización del proyecto, realiza pruebas constantemente con lo que podemos asegurar la calidad del software. Esta metodología genera la documentación que se necesita para validar los artefactos con el cliente, es una metodología muy organizativa y posee varios elementos de planificación que permiten llevar el control del desarrollo del proyecto.

Extreme Programming (XP)

XP es una metodología ágil de desarrollo de software. Está centrada en potenciar las relaciones interpersonales, como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Capítulo 1: Fundamentación teórica

XP cuenta con 4 fases: exploración, planificación, iteración y producción. Es orientada por pruebas y refactorización, se diseña e implementan las pruebas antes de programar la funcionalidad, el programador crea sus propios test de unidad. El objetivo de Xp son grupos pequeños y medianos de construcción de software en donde los requisitos aún son muy ambiguos, cambian rápidamente o son de alto riesgo. Xp busca la satisfacción del cliente tratando de mantener durante todo el tiempo su confianza en el producto. Además, sugiere que el lugar de trabajo sea una sala amplia, si es posible sin divisiones (en el centro los programadores, en la periferia los equipos individuales). Una ventaja del espacio abierto es el incremento en la comunicación y el proporcionar una agenda dinámica en el entorno de cada proyecto. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
5. El programador construye ese valor de negocio.
6. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración. XP mejora un proyecto de software de cuatro formas esenciales: comunicación, simplicidad, retroalimentación y coraje. Los programadores de XP se comunican con sus clientes y entre ellos mismos, para mantener su diseño limpio y simple. Obtienen retroalimentación probando su software desde el primer día. Entregan el sistema a los clientes lo más pronto posible, e implementan los cambios a medida que son sugeridos. Sobre estas bases, los programadores pueden afrontar valerosamente requerimientos y tecnología cambiantes. El ciclo de vida ideal de XP consiste de 4 fases: Exploración, Planificación o Planeamiento, Iteraciones y Producción.
(Chamizo)

Capítulo 1: Fundamentación teórica

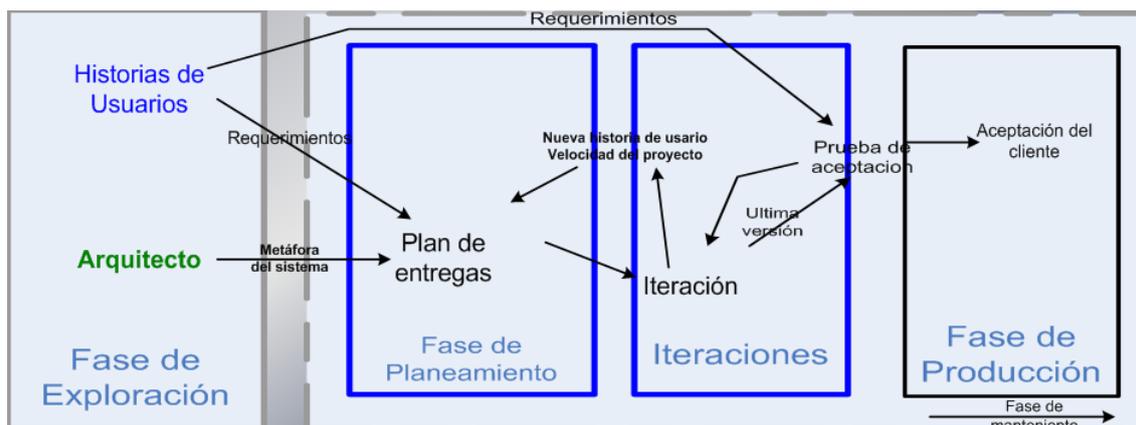


Figura 2: Fases de XP.

Agile Unified Process (AUP)

AUP es una versión simplificada de la Rational Unified Process (RUP). Se describe como fácil de comprender el enfoque al desarrollo de software para aplicaciones empresariales utilizando las técnicas y conceptos de agilidad y aún se mantiene fiel a RUP. El enfoque se aplica técnicas ágiles incluyen Test Driven Development (TDD), Agile Model Driven Development (AMDD), gestión del cambio ágil, y refactoring de bases de datos para mejorar su productividad. La Figura 3 muestra el ciclo de vida de la AUP. En primer lugar, la disciplina de modelo de negocios incluye el RUP de modelado, los requisitos y Análisis y Diseño de las disciplinas.

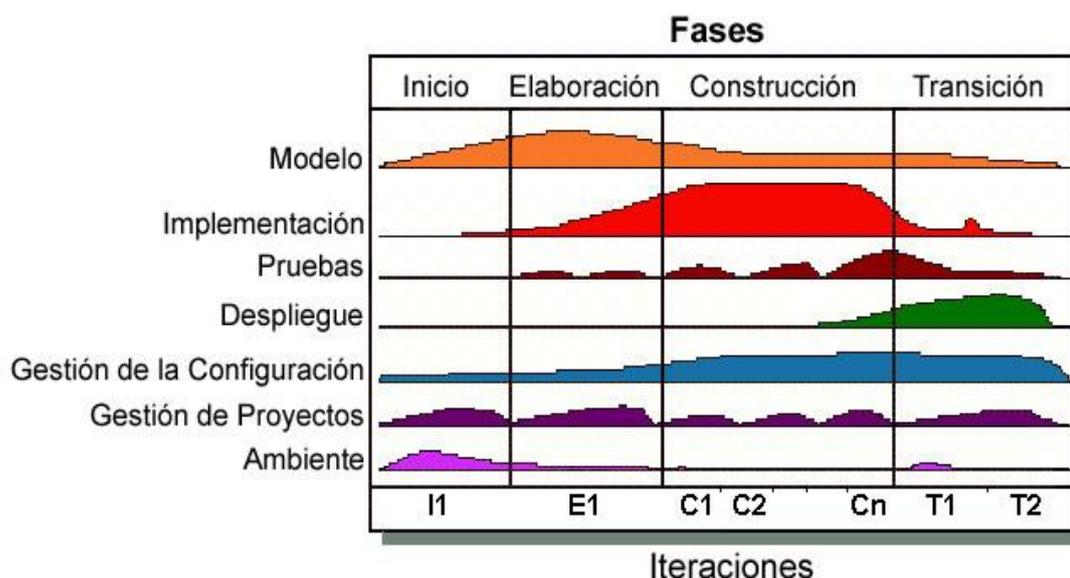


Figura 3: Ciclo de vida de AUP

Capítulo 1: Fundamentación teórica

Fases de AUP:

1. Inicio. El objetivo es identificar el alcance inicial del proyecto, la arquitectura del potencial de su sistema, y para obtener la financiación inicial del proyecto y la aceptación de los interesados.
2. Elaboración. El objetivo es probar la arquitectura del sistema.
3. Construcción. El objetivo es construir software que trabajen sobre una base regular, incremental, que se reúne la más alta prioridad a las necesidades de sus participantes en el proyecto.
4. Transición. El objetivo es validar y desplegar el sistema en su entorno de producción.

Disciplinas de AUP:

- Modelo. El objetivo de esta disciplina es entender el negocio de la organización, el dominio del problema que aborda el proyecto, y definir una solución viable para hacer frente al dominio del problema.
- Implementación: El objetivo de esta disciplina es la de transformar el modelo en código ejecutable y realizar un nivel básico de las pruebas, en las pruebas de unidad en particular.
- Pruebas: El objetivo de esta disciplina consiste en realizar una evaluación objetiva para garantizar la calidad. Esto incluye encontrar defectos, validar que el sistema funciona según lo previsto, y verificar que se cumplan los requisitos.
- Despliegue: El objetivo de esta disciplina es el plan para la entrega del sistema y ejecutar el plan para que el sistema a disposición de los usuarios finales.
- Gestión de la Configuración. El objetivo de esta disciplina es la gestión de acceso a sus artefactos del proyecto. Esto incluye no sólo el seguimiento de las versiones de artefactos en el tiempo, sino también el control y la gestión de los cambios para ellos.
- Gestión de Proyectos. El objetivo de esta disciplina es dirigir las actividades que lleva a cabo en el proyecto. Esto incluye los riesgos de gestión y dirección de personas (la asignación de tareas, el seguimiento del progreso), y coordinar

Capítulo 1: Fundamentación teórica

con las personas y los sistemas fuera del alcance del proyecto para asegurarse de que es entregado a tiempo y dentro del presupuesto.

- Medio Ambiente. El objetivo de esta disciplina es apoyar el resto de los esfuerzos por garantizar que el proceso adecuado, normas de orientación (y directrices), y herramientas (hardware, software) están disponibles para el equipo según sea necesario. **((AUP))**

1.6.2: Análisis de la selección de la Metodología de Desarrollo.

Basado en el análisis de estas metodologías de desarrollo, se decidió usar AUP, porque es la que más se adapta al proyecto a desarrollar así como a las condiciones de trabajo.

Se hace necesario destacar un elemento esencial que distingue al Proceso Ágil Unificado del Proceso Unificado de Desarrollo. De acuerdo con las metodologías tradicionales, el cliente no está obligado a mantener una interacción constante con el equipo de desarrollo, puede seguir el avance del proyecto a través de reuniones planificadas y con un largo período de ocurrencia, siendo necesario por ello una mayor documentación para que exista un control de todo el proceso. AUP establece que debe determinarse una persona que se encargue de velar por el cumplimiento de los requisitos y las prioridades del sistema, esta persona puede ser o bien el cliente en cuestión o un equipo de analistas. En el caso particular, descrito en el documento, este importante rol fue desempeñado por el equipo de analistas, quienes fueron los encargados de hacer cumplir por parte del equipo de desarrollo, las exigencias del cliente de manera eficiente.

Los resultados obtenidos estimulan la puesta en práctica de metodologías ágiles que ofrecen entre sus más connotados beneficios, la capacidad de respuesta a cambios de requisitos a lo largo del desarrollo; entrega continua y en plazos breves de un software funcional; marcada importancia a la simplicidad, eliminando el trabajo innecesario; atención continua a la excelencia técnica y al buen diseño contribuyendo de esta forma a mejoras frecuentes de los procesos y del equipo de desarrollo de manera general. **(Hondares, 2009)**

1.7 Lenguaje de modelado

Un lenguaje de modelado es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar un diseño de software mayormente orientado a objetos.

Capítulo 1: Fundamentación teórica

Algunas organizaciones los usan extensivamente en combinación con una metodología de desarrollo de software, para avanzar de una especificación inicial a un plan de implementación y para comunicar dicho plan a todo un equipo de desarrolladores.

UML es un lenguaje para visualizar, especificar, construir y documentar un sistema de software, ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Los principales beneficios de UML son:

- Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos. **(Serrano)**

Por todos los beneficios que brinda se selecciona UML como lenguaje de modelado.

1.8: Herramientas de Desarrollo

Las herramientas de desarrollo aumentan y facilitan cuantiosamente la velocidad de la producción por lo cual se hace indispensable su uso en todo proceso de desarrollo de software.

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

Sin lugar a dudas las **herramientas CASE** han venido a revolucionar la forma de automatizar los aspectos clave en el desarrollo de los sistemas de información, debido a la gran plataforma de seguridad que ofrecen a los sistemas que las usan y es que

Capítulo 1: Fundamentación teórica

éstas, brindan toda una gama de componentes que incluyen todas o la mayoría de los requisitos necesarios para el desarrollo de los sistemas, han sido creadas con una gran exactitud en torno a las necesidades de los desarrolladores de sistemas para la automatización de procesos incluyendo el análisis, diseño e implantación. **(Case)**

Rational Rose

Rational es una herramienta que abarca todos los aspectos del desarrollo de software, desde su concepción hasta la elaboración del producto. Para ayudar en la implementación del proceso Rational ha desarrollado un conjunto de herramientas integradas que permiten desarrollar las actividades del proceso y obtener una sinergia al integrarse entre las distintas herramientas permitiendo a todo el equipo de desarrollo compartir y utilizar la información necesaria en el momento adecuado. Rational provee herramientas para las siguientes fases del desarrollo de software:

- **Gestión de Requerimientos.**
- **Modelado visual de sistemas basado en UML.**
- **Desarrollo de aplicaciones Web y Java.**
- **Pruebas de software.**
- **Gerenciamiento de la configuración y el cambio. *(Insight)***
- **Facilita el desarrollo de equipos:** Rose proporciona soporte completo equipo, permitiendo a los usuarios (los desarrolladores y analistas) para trabajar con su propia versión del modelo en su propio espacio de trabajo.
- **Se utiliza en todo el proceso de desarrollo de software:** Desde la definición de los requisitos de los usuarios a la aplicación, todos los que trabajan en el proyecto comprende un lenguaje universal. Usted puede utilizar Rose en cualquier etapa del proceso de por vida. Ayuda a descubrir y evitar posteriores errores potencialmente graves.
- **Hace que sea más fácil de gestionar los cambios de modelo:** Cualquier cambio que realice en un modelo de Rational Rose que le haces a los demás mediante una gestión de la configuración y control de versiones (CMVC) del sistema. Esto permite integrar los cambios en el modelo, no importa dónde está en la etapa de desarrollo. Rose normalmente utiliza complemento de

Capítulo 1: Fundamentación teórica

herramientas, tales como Rational ClearCase y Microsoft Visual SourceCafe, para este propósito.

- **Guarda en la creación de documentación adicional del proyecto:** Una ventaja aquí es sólo hay que utilizar los modelos creados en Rose como base para el diseño y desarrollo.
- **Direcciones de software de mala herencia:** Se debe considerar el uso de Rose cuando nos enfrentamos a un software que no se ajusta a las necesidades de los usuarios, ya que Rose permite volver atrás y corregir los defectos en la aplicación de legado. **(P.Charvat)**

Visual Paradigm (VP)

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. **(software, 2007)**

1.8.1 Análisis de la selección de la herramienta Case

Basado en el análisis de estas herramientas, se decidió usar Visual Paradigm como herramienta óptima para el lenguaje uml, porque es la que más se adapta al proyecto a desarrollar así como a las condiciones de trabajo. A continuación alguna de las razones del porqué de esta selección:

Posee entre sus principales características las siguientes:

- **Generación de código e ingeniería inversa:** Brinda la posibilidad de generar código a partir de los diagramas, para plataformas como .Net, Java y PHP, así cómo obtener diagramas a partir del código.
- **Generación de documentación:** Brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.
- **Disponibilidad en múltiples plataformas:** Microsoft Windows (98, 2000, XP, o Vista), Linux, Mac OS X, Solaris o Java.

Capítulo 1: Fundamentación teórica

- **Integración con distintos Ambientes de Desarrollo Integrados (IDE):** Se integra fácilmente con varios IDEs, entre ellos el de Visual Studio y el Eclipse.
- **Interoperabilidad con otras aplicaciones:** Brinda la posibilidad de intercambiar información mediante la importación y exportación de ficheros con aplicaciones como por ejemplo Visio y Rational Rose. Además permite importar y exportar XML.

1.9: Fundamentación del lenguaje de programación a utilizar

Java

El lenguaje de programación Java, fue diseñado por la compañía Sun Microsystems Inc. a principios de los años noventa, con el propósito de crear un lenguaje que pudiera funcionar en redes computacionales heterogéneas y que fuera independiente de la plataforma en la que se vaya a ejecutar. Esto significa que un programa de Java puede ejecutarse en cualquier máquina o plataforma.

La independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Este es el significado de ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo.

El lenguaje Java además tiene un recolector de basura que permite una fácil creación y eliminación de objetos. La recolección de basura de Java es un proceso prácticamente invisible al desarrollador. Es decir, el programador no tiene conciencia de cuándo la recolección de basura tendrá lugar, ya que ésta no tiene necesariamente que guardar relación con las acciones que realiza el código fuente.

En la parte del servidor, Java es muy popular, desde la aparición de la especificación de Servlets y JSP (Páginas Servidoras de Java). Hasta entonces, las aplicaciones web dinámicas de servidor que existían se basaban fundamentalmente en componentes CGI (Interfaz de Entrada Común) y lenguajes interpretados. Ambos tenían diversos inconvenientes, fundamentalmente lentitud, elevada carga computacional o de memoria y propensión a errores por su interpretación dinámica. **(Alvarez)**

PHP

PHP es un lenguaje creado por una gran comunidad de personas. El sistema fue desarrollado originalmente en el año 1994 por Rasmus Lerdorf. El sistema fue denominado Personal Home Page Tools y adquirió relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiese utilizar sus programas en sus propias páginas. Dada la aceptación del primer PHP y de manera adicional, su creador diseñó un sistema para procesar formularios al que le atribuyó el nombre de FI (Form Interpreter) y el conjunto de estas dos herramientas, sería la primera versión compacta del lenguaje: PHP/FI. (desarrolloweb.com)

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. (*Page*)

Ventajas:

- Es un lenguaje multiplataforma.
- Completamente orientado a la web.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. PHP es Open Source, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no estás forzado a pagar actualizaciones anuales para tener una versión que funcione.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Rapidez. PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Está completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.

Capítulo 1: Fundamentación teórica

- Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos. Siempre podrás disponer de ODBC para situaciones que lo requieran. *(Page)*

1.9.1: Análisis de la selección del lenguaje de programación a seleccionar

Después de analizar las características los lenguajes PHP y Java, se decidió hacer uso de Java para el desarrollo de la aplicación, debido a las siguientes características:

- **Simple:** Elimina la complejidad de los lenguajes como C y da paso al contexto de los lenguajes modernos orientados a objetos.
- **Familia:** Como la mayoría de los programadores están acostumbrados a programar en C o en C++, la sintaxis de Java es muy similar al de estos.
- **Robusto:** El sistema de Java maneja la memoria de la computadora por ti. No tienes que preocuparte por apuntadores, memoria que no se esté utilizando, etc. Java realiza todo esto sin necesidad de que uno se lo indique.
- **Seguro:** El sistema de Java tiene ciertas políticas que evitan se puedan codificar virus con este lenguaje. Existen muchas restricciones, especialmente para los applets, que limitan lo que se puede y no puede hacer con los recursos críticos de una computadora.
- **Portable:** Como el código compilado de Java es interpretado, un programa compilado de Java puede ser utilizado por cualquier computadora que tenga implementado el intérprete de Java.
- **Independiente a la arquitectura:** Al compilar un programa en Java, el código resultante es interpretado por diferentes computadoras de igual manera, solamente hay que implementar un intérprete para cada plataforma. De esa manera Java logra ser un lenguaje que no depende de una arquitectura computacional definida.
- **Multithreaded:** Un lenguaje que soporta multithreaded, es un lenguaje que puede ejecutar diferentes líneas de código al mismo tiempo.

1.10: Plataforma de desarrollo

La plataforma Java es el nombre de un entorno o plataforma de computación originaria de Sun Microsystems, capaz de ejecutar aplicaciones desarrolladas usando el Lenguaje de programación Java u otros lenguajes que compilen a bytecode y un

Capítulo 1: Fundamentación teórica

conjunto de herramientas de desarrollo. En este caso, la plataforma no es un hardware específico o un sistema operativo, sino más bien una máquina virtual encargada de la ejecución de aplicaciones, y un conjunto de librerías estándar que ofrecen funcionalidad común.

Hoy en día esta plataforma ha evolucionado en concordancia con el avance tecnológico y se ha convertido en una de las plataformas de programación más usadas por los desarrolladores. Su principal ventaja es que su entorno de desarrollo es independiente de la plataforma sobre la que se trabaje, es decir, sus aplicaciones son funcionales independientemente del sistema operativo sobre el que estén operando.

La plataforma Java consta de dos componentes:

- La Máquina Virtual de Java (JVM)
- La Interfaz de Programación de Aplicaciones de Java (API Java)

A continuación se muestra una representación de los elementos que forman parte de la plataforma Java.

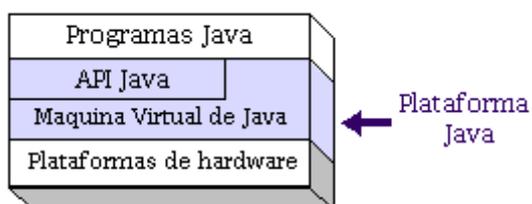


Figura 4: Elementos de Java

El corazón de la Plataforma Java es el concepto común de un procesador “virtual” que ejecuta programas escritos en el lenguaje de programación Java de pilsener. En concreto, ejecuta el código resultante de la compilación del código fuente, conocido como bytecode. Este “procesador” es la máquina virtual de Java o JVM (Java Virtual Machine), que se encarga de traducir (interpretar o compilar al vuelo) el bytecode en instrucciones nativas de la plataforma destino.

Sun define tres plataformas en un intento por cubrir distintos entornos de aplicación. Así, ha distribuido muchas de sus APIs (Application Program Interface) de forma que pertenezcan a cada una de las plataformas:

- Java ME (Java Platform, Micro Edition) o J2ME — orientada a entornos de limitados recursos, como teléfonos móviles, PDAs (Personal Digital Assistant).

Capítulo 1: Fundamentación teórica

- Java SE (Java Platform, Standard Edition) o J2SE — para entornos de gama media y estaciones de trabajo. Aquí se sitúa al usuario medio en un PC de escritorio.
- Java EE (Java Platform, Enterprise Edition) o J2EE — orientada a entornos distribuidos empresariales o de Internet.

Java™ 2 Platform, Enterprise Edition (J2EE™) reduce el coste y la complejidad del desarrollo multi-capa. Las aplicaciones J2EE pueden desplegarse rápidamente y mejorarse fácilmente según responda la empresa a las presiones de la competencia. La plataforma J2EE es un conjunto de herramientas que crean un escenario ideal para el desarrollo y despliegue de aplicaciones escalables en la Web.

- **Portable**, la famosa frase de la Stanford University Network (Sun) 'Escribe una vez, usa en cualquier parte' tiene un gran poder de convocatoria, tu escribes tu aplicación en tu máquina Windows o Linux y cuando la hayas terminado puede utilizarse en cualquier plataforma para la que haya disponible una Máquina Virtual Java (JVM).
- **Escalable**, si tu empresa ve incrementado su número de clientes, nada más sencillo que añadir nuevos componentes J2EE a una aplicación Web para soportar el aumento de clientes, sin tener que reescribir todo el código de nuevo.
- **Altamente Soportada**, prácticamente cualquier gran empresa de software tiene un contenedor de componentes (o servidor de aplicaciones) Web compatibles con J2EE, entre ellas IBM (Websphere), BEA (WebLogic), Apache (Tomcat), la propia Sun con su nuevo servidor de aplicaciones iPlanet, MacroMedia con (JRun), .
- **Segura**, mientras que otros modelos de aplicaciones empresariales requieren medidas de seguridad específicas en cada aplicación, el entorno de seguridad de la plataforma J2EE permite que se definan unas restricciones de seguridad en el momento de despliegue de la aplicación, aislando así las aplicaciones de la complejidad de las implementaciones de seguridad, la plataforma J2EE hace portables una gran complejidad de implementaciones de seguridad. (**Java**)

1.11: Entorno de Desarrollo

Un entorno de desarrollo integrado: IDE, es un programa que incorpora un conjunto de herramientas y que tiene como objetivo ser de soporte a un programador para construir software.

Este puede incorporar un lenguaje o varios de ellos; es considerado un programa de aplicación que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. **(Sánchez)**

Eclipse

La plataforma Eclipse está diseñada para la construcción de entornos de desarrollo (IDEs) que puedan ser utilizados para la construcción de aplicaciones Web, aplicaciones Java de todo tipo, programas C++, y Enterprise JavaBeans (EJBs). La Plataforma Eclipse es una herramienta para todo.

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge.

La característica clave del Eclipse es su extensibilidad a través de los Pluggins que no son más que la mínima unidad de la plataforma que puede ser desarrollado por separado y que le aporta una nueva funcionalidad.

Asimismo, a través de "Pluggins" libremente disponibles es posible añadir control de versiones con Subversion e integración con Hibernate. **(Nicot, 2008)**

NetBeans

NetBeans IDE es una aplicación de código abierto ("open source") diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas, haciendo uso de la tecnología Java.

NetBeans IDE dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles, resaltado de

Capítulo 1: Fundamentación teórica

sintaxis y por si fuera poco sus funcionalidades son ampliables mediante la instalación de packs. *(Ramírez)*

La facilidad de uso de NetBeans también resulta llamativa, ya que no es necesario ir más allá del portal netbeans.org , para buscar las características adicionales y los “Pluggins’ de los paquetes de funciones que abarcan una diversidad de requerimientos desde C/C++ hasta movilidad y la Web. Una vez que se descarga, NetBeans es fácil de utilizar, gracias a su interfaz de usuario y a funciones como la recién anunciada Visual Web Pack (VWP). Con VWP, los componentes JavaServer Faces pueden arrastrarse y soltarse para establecer propiedades y crear códigos para los manipuladores de eventos a nivel del servidor.

Adicionalmente, con el configurador NetBeans Swing GUI, tanto los usuarios finales como los desarrolladores obtienen mayor facilidad de uso. Swing simplifica drásticamente la creación de interfaces gráficas de usuario para grandes aplicaciones de Internet y cliente, además de permitir a los desarrolladores manejar diferentes guías de estilo en diversas plataformas y garantizar la ubicación de aplicaciones en una amplia variedad de lenguajes humanos.

La capacidad de desarrollar de manera eficiente las aplicaciones Java, en una amplia variedad de plataformas, también se debe al respaldo de lenguajes de especificación abiertos para modelado de objetos. Esto permite la edición de dos vías el código fuente se modifica automáticamente junto con los cambios de modelo — y elimina la necesidad de los desarrolladores de tener que referirse constantemente a los comentarios del código fuente. *(IDE)*

1.11.1: Análisis de la selección del entorno de desarrollo

Dentro de las ventajas que tiene NetBeans están las siguientes: NetBeans ayuda a los equipos de desarrollo a utilizar otras mejores prácticas y estándares de la industria para la productividad general del grupo. Un conjunto de estas mejores prácticas incluye soluciones a problemas comunes de configuración que se encuentran en la guía pública de Patrones de Diseño.

Dentro de las ventajas de eclipse está: Eclipse es un software para integrar herramientas de desarrollo que corren sobre un amplio rango de sistemas operativos,

Capítulo 1: Fundamentación teórica

con una arquitectura abierta y basada en Pluggins. Por lo antes expuesto se ha decidido que se va a utilizar el entorno de desarrollo eclipse.

1.12: Sistema Gestor de Base de Datos

Un sistema de gestión de bases de datos (SGBD) es una herramienta que permite insertar, modificar y buscar eficazmente datos específicos en una gran masa de información. El usuario puede precisar estas investigaciones. Los resultados pueden clasificarse según criterios. **(Nicot, 2008)**

MySQL

MySQL para Windows (Versión de desarrollo) es un sistema de administración de una base de datos con soporte para múltiples usuarios.

MySQL usa el lenguaje SQL estandarizado para el almacenamiento, actualización y acceso a información. MySQL es muy rápido y capaz de almacenar grandes cantidades de datos.

MySQL soporta muchos lenguajes de programación distintos como: C, C++, Eiffel, Java, Perl, PHP, Python y TCL. También tiene la opción de protección mediante contraseña, la cual es flexible y segura. **(softonic)**

PostgreSQL

PostgreSQL es un servidor de base de datos objeto relacional libre, liberado bajo la licencia BSD. Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo, dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group). PostgreSQL provee nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- Direcciones IP (IPV4 e IPV6).
- Bloques de direcciones estilo CIDR.
- Direcciones MAC.

Capítulo 1: Fundamentación teórica

- Arrays.

Otras características

- Claves ajenas también denominadas Llaves ajenas o Llaves Foráneas (foreign keys).
- Disparadores (triggers).

PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta (query en inglés). Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido. **(Nicot, 2008)**

1.12.1: Análisis de la selección del Sistema Gestor de Base de Datos

Después de analizar las características los gestores de bases de datos MySQL y PostgreSQL, se decidió hacer uso de este último para el desarrollo de la aplicación, debido a las siguientes características:

- Es el gestor de base de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-visión, soportando casi toda la sintaxis SQL.
- Cuenta con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, perl, tcl y Python). **(Lockhart)**

1.13: Frameworks

Un framework, en el desarrollo de software es una estructura de soporte concretada en la cual un determinado proyecto de software puede afirmarse para su organización y desarrollado. Este en su mayor funcionalidad puede contener soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los componentes de un proyecto. Figura una arquitectura de software que modela las relaciones generales de las entidades del dominio. Suministra una estructura y una metodología de trabajo la cual desarrolla o utiliza las aplicaciones del dominio. Son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. **(Sánchez)**

Presentación

Java Server Faces.

JSF (Java Server Faces) es un marco de trabajo de interfaces de usuario del lado del servidor para aplicaciones Web desarrolladas en Java.

Está compuesto por un API y una implementación para representar componentes UI, manejar su estado, la validación del lado del servidor, conversión de datos y definir la navegación entre las páginas. JSF soporta internacionalización y accesibilidad, proporciona extensibilidad para todas sus características y permite el uso de simples clases java como controladores además de contar con librería de etiquetas para crear componentes dentro de una página JSP.

Este modelo de programación bien definido está regido por el patrón MVC (Modelo Vista Controlador) y facilita de forma significativa la construcción y mantenimientos de aplicaciones Web con UI del lado del servidor.

En la siguiente figura se muestra el ciclo de una petición, donde myUI representa la página creada bajo la tecnología JSF, por otro lado myForm.jsp dibuja los componentes de interface de usuario definido en Java Server Faces.

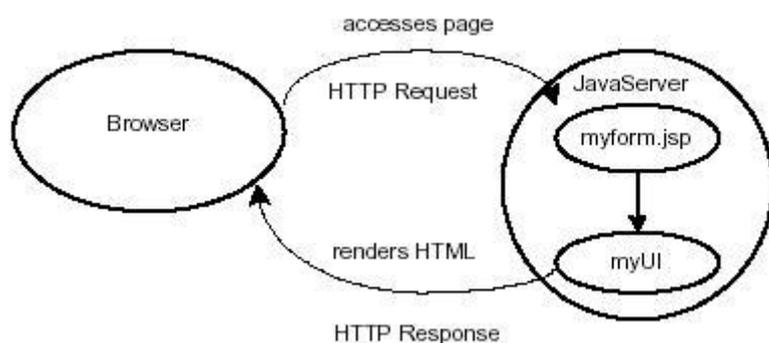


Figura 5: Ciclo de vida bajo la tecnología JSF.

Ajax4jsf.

Ajax4jsf es una librería open source integrable fácilmente con la arquitectura de JSF y con el objetivo de extender su funcionalidades, dotándolas de la tecnología Ajax de forma limpia y sin la necesidad de añadir código javaScrip. Mediante este framework podemos variar el ciclo de vida de una petición JSF, realizar peticiones al servidor de forma automáticas así como control de cualquier evento del usuario.

Su funcionamiento es similar a JSF, mediante sus propias etiquetas se generan eventos que envían peticiones al contenedor Ajax. Estos eventos se pueden producir

Capítulo 1: Fundamentación teórica

por pulsar un botón, un enlace, una región de la pantalla o sencillamente cada cierto tiempo. **(Sánchez)**

Lógica del negocio

Spring Framework es un framework de Java que facilitará la creación de aplicaciones. Diseñado en módulos, con funcionalidades específicas y consistentes con otros módulos, facilita el desarrollo de funcionalidades específicas y hace que la curva de aprendizaje sea favorable para el desarrollador. **(java)**

A pesar de que Spring Framework no obliga a usar un modelo de programación en particular, se ha popularizado en la comunidad de programadores en Java al considerársele una alternativa y una sustitución del modelo de Enterprise JavaBean. Por su diseño el framework ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria.

Mientras que las características fundamentales de este framework pueden emplearse en cualquier aplicación hecha en Java, existen muchas extensiones y mejoras para construir aplicaciones basadas en web por encima de la plataforma empresarial de Java (Java Enterprise Platform). **(Porta)**

Acceso a datos

La persistencia en una aplicación orientada a objetos, es un tema de sumo cuidado ya que se refiere a aquellos objetos que van a persistir en una base de datos o en disco duro a través del tiempo. Una aplicación con un modelo de dominio no interactúa directamente con las filas y las columnas, en las que está estructurada la información en la base de datos, en su lugar lo hará con las entidades definidas en su modelo de dominio, aquí es donde entran a jugar un papel muy importante los Object/Relational Mapping (ORM). **(Silberschatz)**

Hibernate es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

Capítulo 1: Fundamentación teórica

Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la Programación orientada a objetos.

Hibernate ofrece un lenguaje HQL (Hibernate Query Language) para realizar consultas a la base de Datos, lo que permite que no haya que cambiar nada de código para cambiar de Sistema Gestor de Base de Datos, sólo cambiar un parámetro en el fichero de configuración de Hibernate; además de una API para construir las consultas programáticamente (conocida como "criteria"). **(Porta)**

Entre sus bondades más significativas están:

- Admite general una base de datos a partir del modelo de clases, para ello tiene compatibilidad con los motores de base de datos más actuales: Oracle, DB2, MySql, Postgres y más, esto se logra especificándole en un archivo XML el driver a usar en dependencia del motor escogido.
- Mapeo entre las entidades persistentes y las tablas de la base de datos, soportando los diferentes tipos de relaciones y amplia gama de tipos de datos: string, boolean, double, integer, date etc.
- Posee un lenguaje propio de consultas HQL (Object-Relational Mapping) e incorpora un API mediante el cual se pueden efectuar consultas programáticamente llamadas "criteria"; todo esto sin quitar el uso del SQL nativo.

Todas estas características liberan al desarrollador de las preocupaciones del tipo de Motor de Base de Datos, así como el lenguaje nativo a usar en cada caso, aparte de que facilita trabajar limpiamente en un modelo objetual y nos brinda la flexibilidad de usar SQL en caso que la situación lo permita, en la mayoría de los casos para buscar rendimiento. **(Sánchez)**

1.14: Arquitectura Técnica

El estilo arquitectónico en capas tomará parte en la solución proporcionándole ventajas sustanciales como lo son: la centralización de los aspectos de seguridad y

Capítulo 1: Fundamentación teórica

transaccionalidad, no replicación de lógica de negocio en los clientes posibilitando así, que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios reduciendo los costes de mantenimiento y garantizando a su vez una mayor sencillez de los clientes. (*Hondares, 2009*)

1.15: Conclusiones

Debido al estudio previo realizado, en el que se ha llevado a cabo un análisis detallado de las tecnologías y herramientas más usadas en el campo de la informática, se opta por la decisión de desarrollar una aplicación web, sobre el lenguaje Java, con la integración de los frameworks Java Server Faces (JSF), Spring e Hibernate, lo cual daría como ventaja velocidad de desarrollo, además del uso del patrón n capas con tres niveles, facilitando el desarrollo del producto y disminuyendo el acoplamiento. Para lograr tal resultado se propone el uso de las herramientas Eclipse y Visual Paradigm por las facilidades que estas brindan. El proceso como tal será orientado por la metodología AUP, la cual constituye una guía de cómo se debe desarrollar una aplicación de tal escala.

Capítulo 2: Modelamiento del negocio y requisitos

Capítulo 2: Modelamiento del negocio y requisitos

Después de haber analizado el estado del arte y elegido las herramientas y metodología a utilizar para el correcto desarrollo de la aplicación, están todas las condiciones creadas para hacer la propuesta de solución al problema existente. Para implementar la solución propuesta, se siguieron los pasos definidos por la metodología AUP para el proceso de desarrollo. Este capítulo estará centrado en desarrollar el modelo del negocio y requisitos.

2.1: Estado actual de los procesos involucrados en el campo de acción

Actualmente las actividades extradocentes en la facultad 8 se van realizando a lo largo del curso escolar, así como otras actividades que puedan ir surgiendo por creatividad de la FEU y la UJC de la facultad. En estos eventos los máximos organizadores son los integrantes de la dirección de la facultad y los protagonistas son los estudiantes. Cada evento de estos se realiza en una fecha determinada.

Algunos de los eventos son:

- Jornada Científico Estudiantil: En esta actividad como ya lo indica el nombre se desarrollan investigaciones científicas que sean de interés para todos y provechos para la universidad.
- Evento Juvenil Martiano: En este evento es nuestro héroe nacional José Martí el eje principal, con respecto a su vida y obra son los trabajos que se desarrollan
- Mi Web por Cuba: Se presentan sitios web con información referente a nuestro país, acerca de cualquier tema.
- Festivales de Artistas Aficionados: En estos festivales se promueve cultura, todos los estudiantes pueden hacer lo que les guste y resultan premiados.
- Juegos deportivos: Estos juegos se realizan para el divertimento y preparación física de los estudiantes y profesores, pero tampoco cuentan con una aplicación que les permita saber los estudiantes que participan y colaboran.

2.2: Propuesta de solución

Para llevar a cabo nuestra propuesta se propone la implementación de un sistema que permita llevar a cabo la gestión de las actividades extradocentes que tienen los estudiantes. El sistema a implementar debe ser capaz de permitir a los Jefes de brigadas registrar la información sobre la participación de cada estudiante en las actividades que se lleven a cabo en el seno del grupo. Así como permitir que cada estudiante pueda ver la información que el sistema registra sobre su desempeño. También se necesita que el sistema permita registrar los resultados de cada evento en el que el estudiante participa.

Para llevar a cabo nuestra propuesta necesitamos utilizar tecnologías, frameworks, lenguajes de programación, entornos de desarrollo. Todo con el objetivo de lograr un sistema eficiente, que brinde una respuesta rápida, facilidad de uso y soporte. Además hemos de desarrollar un sistema bajo las premisas de la arquitectura cliente – servidor mediante el protocolo HTTPS.

Usaremos como metodología de desarrollo (AUP), esta metodología no concibe el Análisis y Diseño como Flujo de Trabajo donde tiene lugar el refinamiento de los requisitos, la identificación de las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Se hace necesario destacar que no se obvia el análisis y el diseño, evidentemente es necesario a la hora de desarrollar un sistema tener en cuenta cómo será implementado el software a partir de las funcionalidades previstas y las restricciones impuestas.

Para el modelado del sistema se utilizará como herramienta el Visual Paradigm, con una metodología (AUP) basadas en Lenguaje Unificado de Modelación (UML: Unified Model Language) y que permite una mejor construcción, brindando como resultado un software que satisface al máximo las necesidades del cliente. Además se utilizará como entorno de desarrollo integrado para el sistema Eclipse y como plataforma de desarrollo (J2EE), como lenguaje de programación Java y el gestor de base de datos será Postgres. Se va a hacer uso de los siguientes frameworks, JSF en la capa de presentación, Spring para la capa de negocio e Hibernate para la capa de acceso a datos.

Capítulo 2: Modelamiento del negocio y requisitos

2.3: Modelamiento del negocio

El modelamiento del negocio es utilizado para comprender el conjunto de procesos de negocio que tienen lugar dentro de una organización como paso previo para establecer los requisitos del sistema a desarrollar. Tiene como objetivos:

- Entender la estructura y dinámica de la organización.
- Asegurar que los clientes, usuarios finales y desarrolladores tienen un entendimiento común de la organización
- Derivar los requisitos del software necesarios para soportar la organización.

2.4: Técnicas utilizadas para comprender el negocio

Nunca debe perderse de vista por qué se desarrolla el software: para satisfacer necesidades reales, para resolver problemas reales. La única forma de resolver las necesidades reales es comunicarse con aquellos que tienen dichas necesidades. El cliente o usuario es la persona más importante involucrada en el proyecto. Con el objetivo de comprender los procesos de negocio que se debían modelar y de identificar las necesidades reales de los clientes y usuarios, se realizaron varias entrevistas las cuales fueron preparadas con antelación para garantizar la participación del personal necesario. Como resultado de dichas entrevistas se elaboraron diagramas para describir los procesos. **(Nicot, 2008)**

2.5: Captura de requisitos

La captura de requisitos es la actividad mediante la cual el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas que participen en él. Por la complejidad que todo esto puede implicar, la ingeniería de requisitos ha trabajado desde hace años en desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y precisa. Existen un grupo de técnicas que de forma clásica han sido utilizadas para esta actividad en el proceso de desarrollo de este tipo de software, ejemplo de ellas son: **(Nicot, 2008)**

Capítulo 2: Modelamiento del negocio y requisitos

Las entrevistas:

Le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural. A pesar de que las entrevistas son esenciales en el proceso de la captura de requisitos y con su aplicación el equipo de desarrollo puede obtener una amplia visión del trabajo y las necesidades del usuario, es necesario destacar que no es una técnica sencilla de aplicar. Requiere que el entrevistador tenga experiencia en el tema y capacidad para elegir bien a los entrevistados y obtener de ellos toda la información posible en un período de tiempo limitado. Aquí desempeña un papel fundamental la preparación de la entrevista. **(Nicot, 2008)**

Casos de Uso:

Los casos de uso permiten mostrar el contorno (actores) y el alcance (requisitos funcionales expresados como casos de uso) de un sistema. Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los actores del mismo para realizar una determinada función. Los actores son elementos externos (personas, otros sistemas, etc.) que interactúan con el sistema como si de una caja negra se tratase. La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente, sin embargo carecen de la precisión necesaria. Si no se acompañan con una información textual o detallada con otra técnica como pueden ser los diagramas de actividades. **(Nicot, 2008)**

2.6: Procesos del negocio

Son los procesos que deberán ser informatizados por el sistema para un mejoramiento del control de los resultados obtenidos por los estudiantes. (Nicot, 2008)

- Registrar participación en los festivales.
- Registrar participación en los juegos.
- Registrar participación en otras manifestaciones.
- Registrar participación en actividades e conocimientos.
 - Registrar participación en Evento Juvenil Martiano.
 - Registrar participación en Jornada Científica estudiantil.
 - Registrar participación en Mi Web por Cuba.
 - Registrar participación en Copa JAVA.
 - Registrar participación en Copa Pascal.

Capítulo 2: Modelamiento del negocio y requisitos

- Registrar participación en Copa IS.

2.7: Actores y trabajadores del negocio

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio. El término actor significa el rol que algo o alguien juega cuando interactúa con el negocio para beneficiarse de sus resultados. De acuerdo con esta idea un actor del negocio representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio, o sea, ser instancias de un mismo actor. **(Nicot, 2008)**

Por otra parte un trabajador del negocio es una abstracción de una persona (o un grupo de personas), una máquina o un sistema automatizado, que actúa en el negocio realizando una o varias actividades y manipulando entidades del negocio. **(Nicot, 2008)**

Actores del negocio	Justificación
Estudiante	Es el protagonista principal de las actividades que se realizan en la universidad, ya sean culturales, deportivas o de conocimiento.

Tabla 1: Actores del negocio y descripción.

Trabajadores del negocio	Descripción
Estudiante FEU(deporte, cultura, actividades de conocimiento)	Es el encargado de registrar todos los datos relacionado con los estudiantes ya sean datos del estudiante, participación y resultados obtenidos y hacerle llegar al vicedecano de extensión o vicedecano de producción estos datos.
Jurado(profesores EF, profesores, sistema online, coordinadores cultura)	Es el encargado de evaluar la participación del estudiante en alguna

Capítulo 2: Modelamiento del negocio y requisitos

	actividad y darle un resultado.
Jefes de brigada	Son los encargados de controlar las actividades de conocimiento, deportivas y culturales y la información registrada por los estudiantes de la FEU.

Tabla 2: Trabajadores y descripción.

2.8: Modelo de casos de uso

El modelo de casos de uso describe el negocio en términos de casos de uso, que corresponde a lo que generalmente se le llama procesos. En el modelo de casos de uso del negocio se describen los procesos del negocio (casos de uso del negocio) y su interacción con elementos externos (actores), tales como socios y clientes, es decir, describe como se llevan a cabo los procesos del negocio y su objetivo básico es describir cómo se realiza el proceso del negocio en la organización. **(Nicot, 2008)**

El actor es el estudiante.

Los casos de uso son:

- Registrar participación en los festivales.
- Audicionar.
- Registrar participación en los juegos.
- Registrar participación en otras manifestaciones.
- Registrar participación en actividades e conocimientos.
- Registrar participación en Evento Juvenil Martiano.
- Registrar participación en Jornada Científica estudiantil.
- Registrar participación en Mi Web por Cuba.
- Registrar participación en Copa JAVA.
- Registrar participación en Copa Pascal.
- Registrar participación en Copa IS.

Capítulo 2: Modelamiento del negocio y requisitos

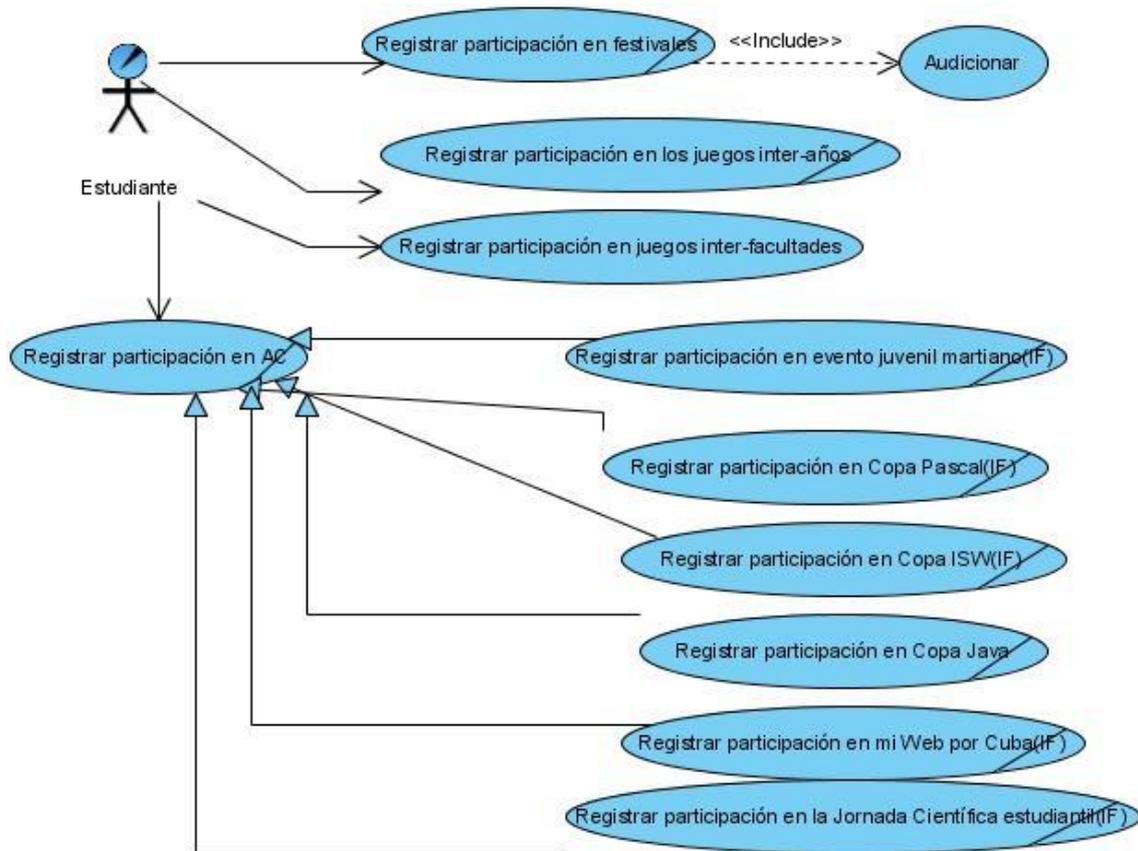


Figura 6: Diagrama de casos de uso del negocio.

2.9: Especificaciones de los casos de uso y diagrama de actividades

A continuación se presenta una descripción detallada del proceso en cada uno de los casos de uso del negocio, con el diagrama de caso de uso correspondiente y el flujo de actividades modelado a través del diagrama de actividades.

Ejemplo el caso de uso Registrar participación en los festivales.

Nota: Ver demás descripciones y caso de uso en los anexos.

Caso de Uso:	Registrar participación en festivales
Actores:	Estudiante
Trabajadores:	Estudiante FEU de cultura, jefe de brigadas y jurado(coordinadores del centro cultural)
Resumen:	

Capítulo 2: Modelamiento del negocio y requisitos

	<p>El caso de uso da comienzo cuando el estudiante da su disposición de participar en el festival y le hace llegar esta decisión al estudiante de la FEU encargado de cultura, quien toma los datos del estudiante así como la manifestación en la que se presentará(música, danza o teatro). Una vez tomados los datos del estudiante este le hace llegar los mismos al jefe de brigada que es el encargado de controlar la participación del estudiante en cualquier actividad. Antes de participar en el festival el estudiante debe presentarse primero para una audición (ir a caso de uso Audicionar). Si pasa la audición durante la participación del estudiante en la actividad este obtiene un resultado dado por el jurado que es registrado por el estudiante de la FEU y por el jefe de brigada finalizando así el caso de uso.</p>
Precondiciones:	
Flujo Normal de Eventos	
Sección ""	
Acción del Actor	Respuesta del Negocio
<p>1- El Estudiante comunica su disposición de participar.</p> <p>2- El estudiante entrega todos los datos (nombre, apellidos, grupo, nombre de la obra, unidad artística) para que quede registrada su participación.</p> <p>3- El estudiante se presenta en la audición.</p> <p>4- El estudiante participa en los festivales.</p>	<p>1.1 El estudiante FEU antes de registrar los datos del estudiante los solicita antes.</p> <p>2.1 El estudiante FEU registra los datos y se los envía al jefe de brigada que también registra la participación del estudiante.</p> <p>3.1 El jurado da una evaluación en la audición.</p> <p>4.1 El jurado da un resultado a la obra.</p> <p>4.2 El estudiante de la FEU registra resultado e informa al jefe de brigada el resultado del estudiante.</p>

Capítulo 2: Modelamiento del negocio y requisitos

	4.3 El jefe de brigada registra la participación del estudiante así como el resultado obtenido terminando así el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	Si el estudiante no pasa la audición no podrá participar en el festival y no quedará registrada su participación.
Poscondiciones	Queda registrada la participación de los estudiantes en los festivales.

Tabla 3: Descripción de CU Registrar participación en festivales

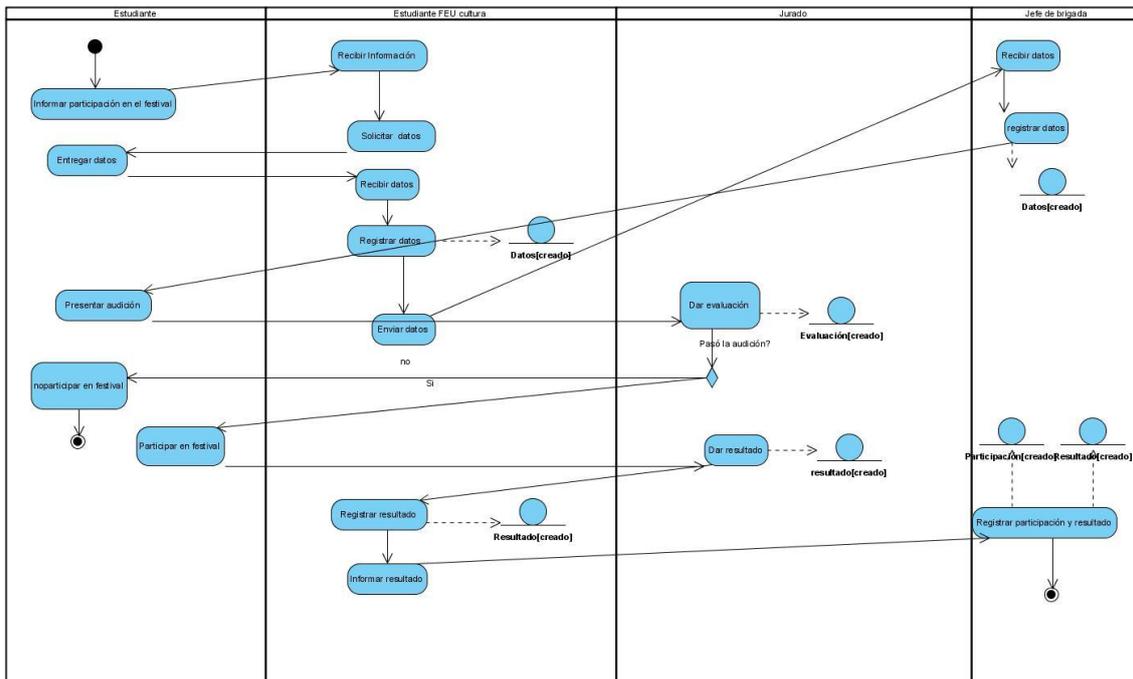


Figura 7: Diagrama de actividades.

2.10: Especificación de los requisitos del software

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Para cumplir con los objetivos trazados el sistema debe cumplir con las siguientes funcionalidades:

Capítulo 2: Modelamiento del negocio y requisitos

- R1 Autenticar usuario
 - R1.1-Introducir nombre de usuario y contraseña.
 - R1.2-Validar datos introducidos.
 - R1.3-Mostrar al usuario las opciones a las que tiene acceso según el rol o permiso asignado.

- R2 Gestionar usuario
 - R2.1- Registrar usuario al sistema.
 - R2.2- Modificar usuario al sistema.
 - R2.3- Eliminar usuario del sistema.

- R3 Gestionar resultado en juegos
 - R3.1- Registrar resultado en juegos
 - R3.2- Modificar resultado en juegos

- R4 Gestionar participación en juegos
 - R4.1- Registrar participación en juegos.
 - R4.2-Eliminar participación en juegos.

- R5 Gestionar deporte
 - R5.1- Registrar deporte
 - R5.2- Eliminar deporte
 - R5.3- Modificar deporte

- R6 Gestionar resultado Actividades de Conocimiento (AC)
 - R6.1- Registrar resultado en AC
 - R6.2-Modificar resultado en AC

- R7 Gestionar participación en AC
 - R7.1- Registrar participación en AC
 - R7.2-Eliminar participación en AC

- R8 Gestionar AC

Capítulo 2: Modelamiento del negocio y requisitos

R8.1- Registrar AC

R8.2- Eliminar AC

R8.3- Modificar AC

- R9 Gestionar resultado en Actividades Culturales.
 - R9.1- Registrar resultado en actividades culturales.
 - R9.2- Modificar resultado en actividades culturales.

- R10 Gestionar participación en Actividades Culturales.
 - R10.1- Registrar participación en actividades culturales.
 - R10.2- Eliminar participación en actividades culturales.

- R11 Realizar búsqueda.
- R12 Generar reporte.
- R13 Gestionar reunión.
 - R13.1 Registrar reunión
- R14 Gestionar acuerdos de reunión.
 - R14.1 Registrar acuerdos de reunión.
 - R14.2- Eliminar acuerdos de reunión.
 - R14.3- Modificar acuerdos de reunión.
- R15 Gestionar opinión.
 - R15.1 Registrar opinión.
 - R15.2- Eliminar opinión.
 - R15.3- Modificar opinión.
- R16 Gestionar estudiante.
 - R16.1 Registrar estudiante.
 - R16.2- Eliminar estudiante.
 - R16.3- Modificar estudiante.

2.11: Requerimientos no funcionales

Los requerimientos no funcionales son características que describen alguna forma o restricción para la realización de algún requerimiento (funcionalidad) o conjunto de ellas e inclusive todos los requerimientos. Se consideran los atributos del sistema, propiedades que debe tener el producto.

Capítulo 2: Modelamiento del negocio y requisitos

A continuación se muestran los requerimientos no funcionales:

Apariencia o interfaz externa

- La interfaz no contiene muchas imágenes para no demorar las respuestas al usuario.
- El diseño de la interfaz es sencillo y claro de usar con reconocimiento visual a través de elementos visibles que identifiquen cada una de sus acciones.
- Es formal, serio y con una navegación sugerente, todo esto teniendo en cuenta el fin con el que se desarrolla la aplicación

Usabilidad

- El sistema podrá ser utilizado por personas que tengan un conocimiento básico en el manejo de las computadoras.
- El sistema deberá estar disponible en todo momento.
- Se debería contar con menús que recompilarán las funcionalidades más importantes del sistema.

Rendimiento

- Las funcionalidades deben de estar divididas en secciones, de modo que no se sobrecarguen los pedidos.
- Se debe soportar el paginado cuando sea mucha la densidad de una sección.
- Las respuestas no deben tardar en ser procesadas más de 20 segundos.
- Las consultas a base de datos no deben contener campos inseríos que sobrecarguen los pedidos al gestor por la red.
- El hardware donde corra la aplicación debe tener suficiente memoria RAM para soportar más de 100 peticiones simultáneas.

Soporte

- Se necesita un servidor de bases de datos que soporte grandes volúmenes de datos (proponemos se use Postgres, por la ventaja de ser libre).
- Se requiere que el producto reciba mantenimiento ante cualquier fallo que ocurra. El sistema es de fácil instalación.

Políticos culturales

- El producto no debe contener palabras en otros idiomas.
- El producto debe respetar los términos empleados normalmente por los especialistas en el tema de las organizaciones que representa.

Capítulo 2: Modelamiento del negocio y requisitos

- Debe contener información acorde a los principios éticos de las organizaciones políticas que representa.
- Toda modificación al funcionamiento establecido en los requerimientos será realizada por la Dirección de la Facultad # 8 (estudiantes de la FEU encargados de cada una de las actividades).

Portabilidad

- El sistema será multiplataforma (Linux o Windows)
- El sistema será multiplataforma (Linux o Windows), bajo los términos de software libre.

Seguridad

- El usuario debe autenticarse antes de entrar al sistema, su autenticación será negociada con el servicio, Lightweight Directory Access Protocol (LDAP) de la universidad y de forma local.
- El sistema se encarga de controlar los diferentes niveles de acceso y funcionalidad de usuarios al sitio.
- Identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema y de asignarles el privilegio que le corresponde.
- Garantiza que la información sea insertada y eliminada únicamente por quien tiene derecho a hacerlo.
- Cada usuario debe contener un rol en el sistema, que no debe de ser suplantado.
- Los campos sujetos a consultas de base de datos deben validar posibles ataques con sentencias SQL, antes de realizar la consulta.

LDAP es un protocolo a nivel de aplicación, que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) al que pueden realizarse consultas. Habitualmente, almacena la información de login (usuario y contraseña) y es utilizado para autenticarse aunque es posible almacenar otra información (datos de contacto del usuario, ubicación de diversos recursos de la red, permisos, certificados). En conclusión, LDAP es un protocolo de acceso unificado a un conjunto de información sobre una red. **(Nicot, 2008)**

Capítulo 2: Modelamiento del negocio y requisitos

Confiabilidad

- Deben establecerse los mecanismos necesarios para el restablecimiento del sistema ante fallos de comunicación u otros, los tiempos mínimos para ello no deben exceder las 6hrs.
- Deben montarse sistemas de respaldo eléctrico en los locales de los servidores para mantener la vitalidad de los servicios.
- Debe hacerse una copia semanal de los datos hacia una zona segura, para garantizar que no se pierdan.

Legales

- El sistema debe ajustarse y regirse por las leyes del software libre, bajo licencia GPL2 o superior. Estando de acuerdo con los principios que persigue la UCI de migrar hacia software libre.
- El sistema debe ajustarse y regirse por las leyes estipuladas para dar cumplimiento a los procesos que se automatizarán.

Software

- Para el funcionamiento del sistema en el servidor es necesario el Sistema Operativo (S.O) Windows 98 o superior, Linux o Unix, en sus versiones de S.O servidores.
- Para el funcionamiento del sistema en las terminales cliente es necesario el S.O Windows 98 o superior, Linux o Unix.

Hardware

- Se necesitan como requerimientos mínimos una PC con procesador Pentium II o superior.

2.12: Conclusiones

En este capítulo se inició el desarrollo de la propuesta del sistema a partir de la realización del estudio de los procesos que componen el negocio así como la interacción entre actor y caso de uso, el modelo de objeto obtenido con los trabajadores y las entidades con las que se relaciona cada uno, y los requisitos que el sistema debe tener y cumplir.

Capítulo 3: Análisis y diseño del sistema

Capítulo 3: Análisis y diseño del sistema

Este capítulo está conformado por los diagramas de casos de uso del sistema; la descripción de algunos de estos, la modelación de los diagramas de clases del análisis correspondiente así como las del diseño; donde se aprecia la interacción del usuario con las diferentes interfaces del sistema y las clases entidades involucradas.

3.1: Actores del sistema y justificación

Actores	Justificación
Usuario	Es una generalización del administrador, de los estudiantes y los profesores, lleva a cabo el proceso de autenticación, reportes y realizar búsquedas.
Administrador	Puede realizar todas las acciones que realiza un usuario. Es el responsable de gestionar usuario.
Estudiante FEU de cultura	Es responsable de gestionar toda la información referente a las actividades culturales.
Estudiante FEU de deporte	Es responsable de gestionar toda la información referente a las actividades de deporte.
Estudiante FEU de las AC	Es responsable de gestionar toda la información referente a las actividades de conocimiento.
Jefes de brigada	Es responsable de llevar a cabo el registro y control de la participación de todos los estudiantes en las diferentes actividades, así como de las reuniones de brigada.
Estudiante FEU de funcionamiento	Es responsable de gestionar todo el orden del día de las reuniones de la juventud.

Tabla 4: Actores del sistema

3.2: Modelo de casos de uso del sistema

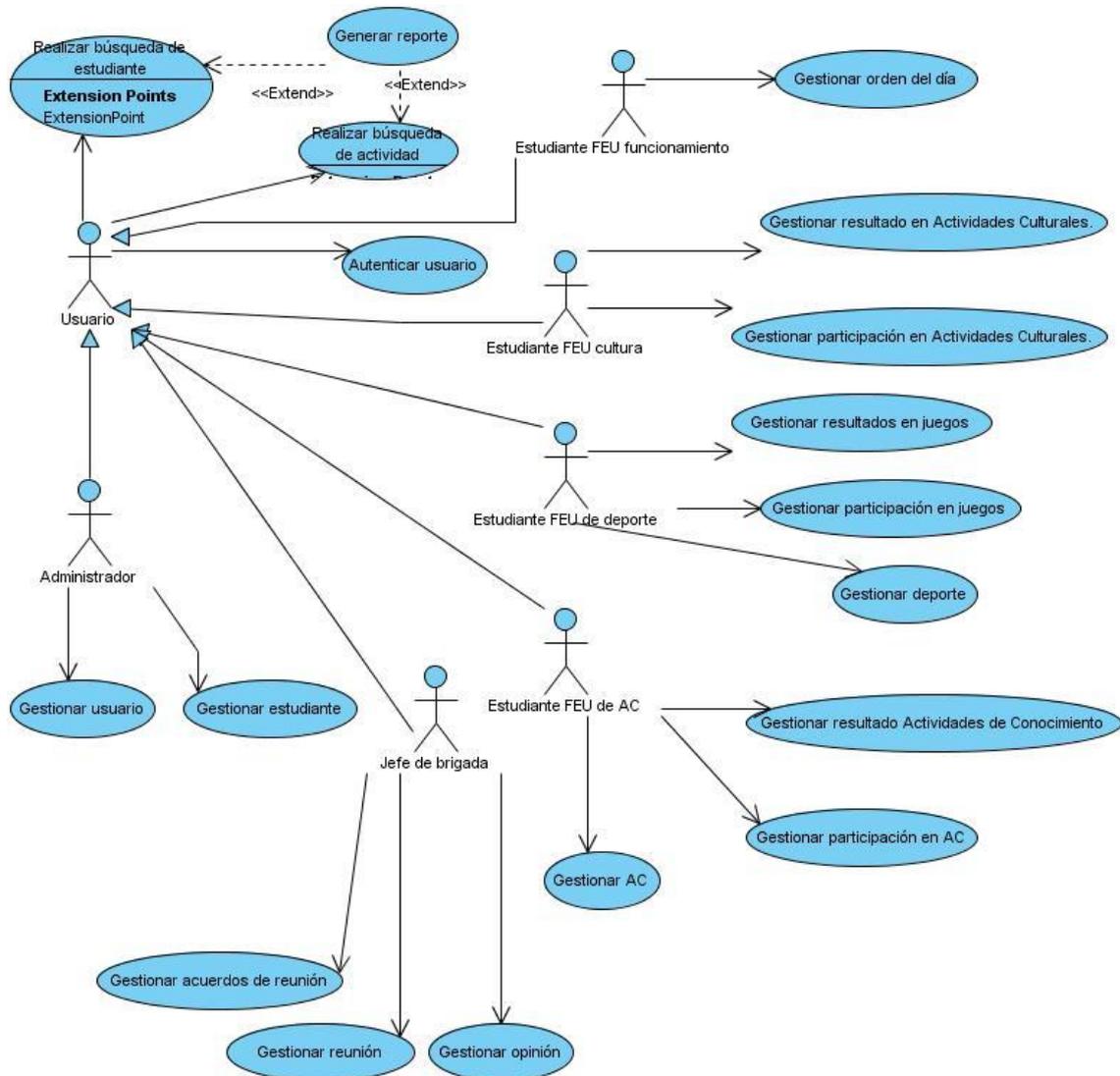


Figura 8: Diagrama de casos de uso del sistema.

Puesto que el sistema tiene una gran complejidad por la cantidad de requisitos funcionales con los cuales debe cumplir, se decidió dividir en paquetes según los casos de usos requeridos para dar soporte a un determinado actor del sistema.

Capítulo 3: Análisis y diseño del sistema

El paquete General contiene los casos de uso que son utilizados por el usuario ya sea una búsqueda, un reporte o la autenticación de un usuario.

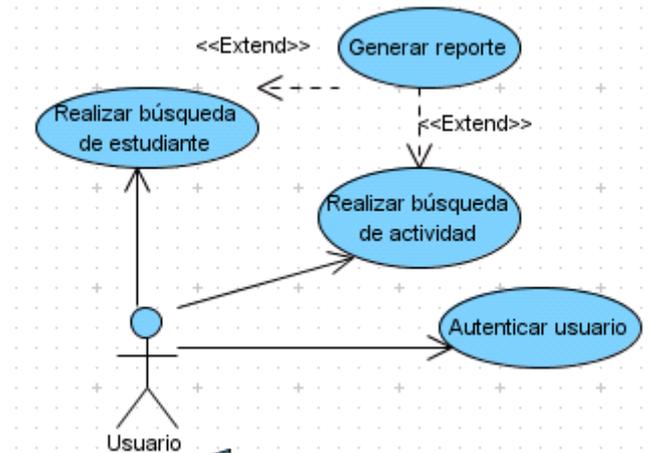


Figura 9: Paquete general.

El paquete Gestionar Actividades culturales es el utilizado por el estudiante FEU de cultura y contiene todos los casos de usos que tienen que ver con actividades culturales.

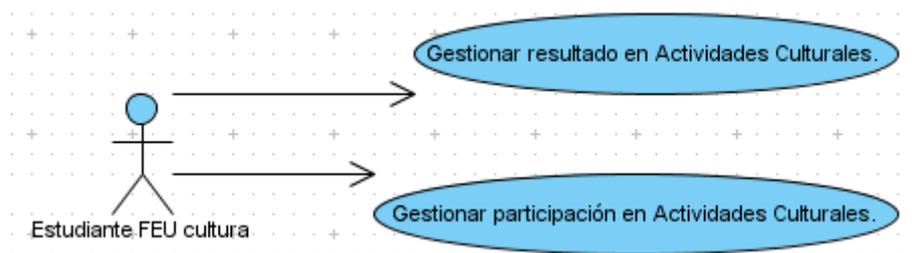


Figura 10: Paquete actividades culturales.

El paquete Gestionar Actividades deportivas es el utilizado por el estudiante FEU de deporte y contiene todos los casos de usos que tienen que ver con actividades deportivas.

Capítulo 3: Análisis y diseño del sistema

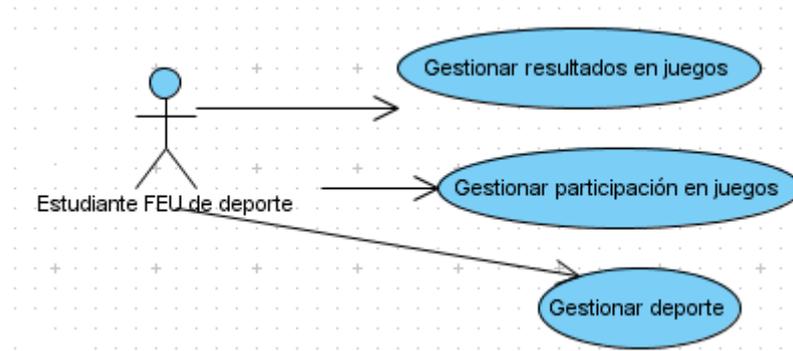


Figura 11: Paquete actividades deportivas.

El paquete Gestionar Actividades de Conocimiento es el utilizado por el estudiante FEU de actividades de conocimiento y contiene todos los casos de usos que tienen que ver con actividades de conocimiento.

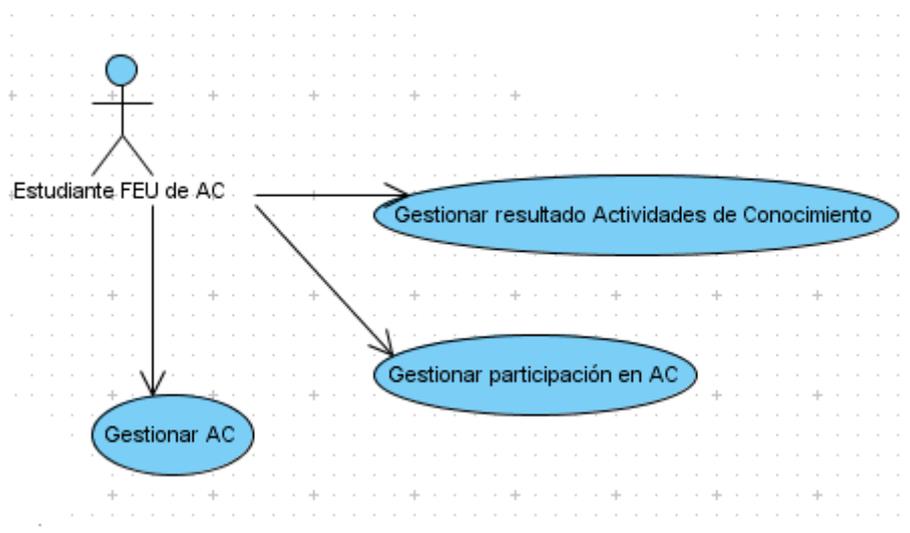


Figura 12: Paquete actividades de conocimiento.

El paquete Administración es el utilizado por el administrador del sistema y contiene todos los casos de uso que tiene que ver con el administrador.

Capítulo 3: Análisis y diseño del sistema

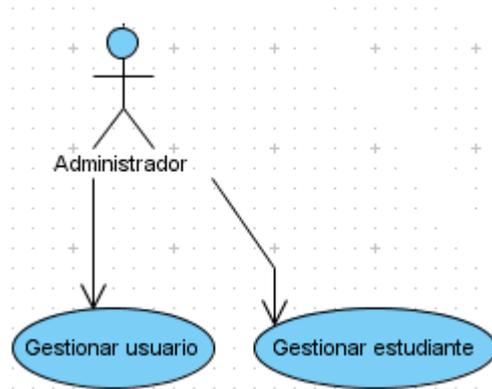


Figura 13: Paquete administración.

El paquete Gestionar orden del día es el utilizado por el estudiante FEU de funcionamiento y contiene el caso de uso que tienen que ver con este tipo de actividad.

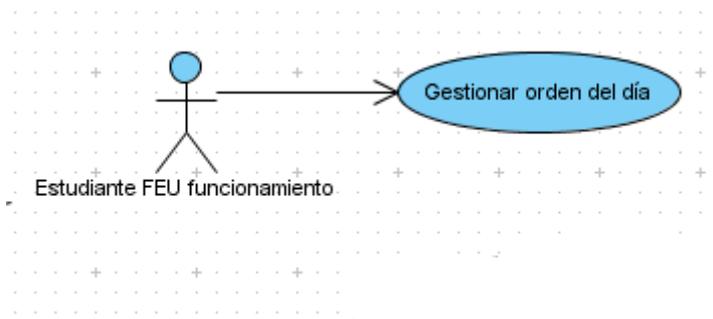


Figura 14: Paquete Gestionar orden del día.

El paquete Gestionar reunión es el utilizado por el jefe de brigada y contiene los casos de uso que tienen que ver con este tipo de actividad.

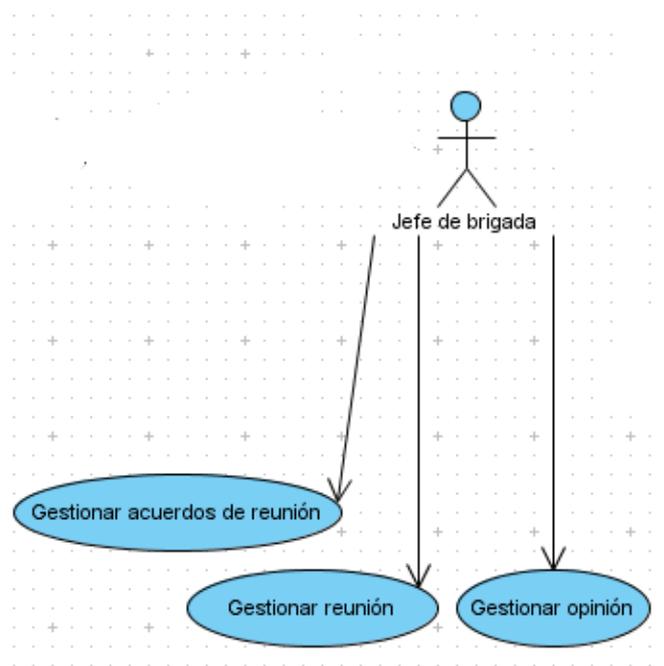


Figura 15: Paquete Gestionar reunión.

3.3: Modelo del análisis

Se analizaron los requisitos descubiertos en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es obtener una comprensión más precisa de los requisitos y una descripción de los mismos que sea más fácil de mantener y que nos ayude a estructurar el sistema entero incluyendo su arquitectura. En el análisis podemos estructurar los requisitos de manera que nos facilite su comprensión, su preparación, su modificación y en general su mantenimiento. Esta estructura (basada en clases de análisis y paquetes) es independiente de la estructura que se dio a los requisitos (basada en casos de uso). Sin embargo existe una trazabilidad directa entre esas distintas estructuras, la cual se define entre en el modelo de casos de uso y realizaciones del caso de uso en el modelo de análisis.

Modelo del análisis:

En la construcción del modelo del análisis se tienen que identificar las clases que describen la realización de casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el diagrama de clases del análisis, que por lo general se descompone para agrupar las clases en paquetes. **(Nicot, 2008)**

3.4: Clases del análisis

Es una abstracción de una o varias clases del diseño del sistema que centra su atención en los requisitos funcionales. El comportamiento de estas se define mediante una descripción textual conocida como responsabilidades, sus atributos son mayoritariamente conceptuales y con frecuencia pasan a ser clases en las etapas de diseño e implementación. Las relaciones son más conceptuales que las usadas en la etapa de diseño e implementación en el caso de la asociación no se le dan importancia a la navegabilidad y pueden usarse las relaciones de generalización. Las clases de análisis se identifican con uno de tres estereotipos básicos que pueden ser de interfaz, control o entidad. **(Nicot, 2008)**

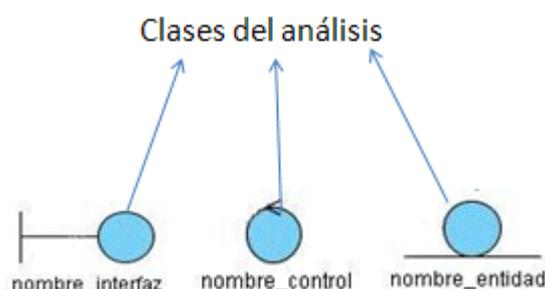


Figura 16: Clases del análisis.

- **Clase interfaz**
 - Modelan la interacción actor-caso de uso.
 - Ventanas, formularios, comunicación con otros sistemas o dispositivos.
Se identifican a partir de los actores:
 - Una clase para cada interacción Actor-Caso de uso.
 - Una clase para cada sistema externo.
 - Una clase para cada dispositivo que se utilice.

- **Clase entidad:**
 - Modelan la información del sistema.
 - Modelan el comportamiento asociado a una información.
Se identifican a partir de:
 - Objetos o entidades del negocio.
 - Glosario de términos.
 - Conceptos de los modelos conceptuales o modelo del dominio.

Capítulo 3: Análisis y diseño del sistema

➤ **Clase control:**

- Coordinan el trabajo de las clases.
- Encapsulan comportamiento de un caso de uso.
- Funciones complejas.

Se identifican a partir de:

- En principio se define una clase control para cada caso de uso.
- No usar clase control si el flujo es simple.
- Crear más de una cuando algún flujo puede re-usarse en otro caso de uso.
- Una clase control por cada actor.

3.5: Descripción de los casos de uso del sistema

A continuación se describirá un caso de uso del sistema (Gestionar participación en actividades culturales), mostrando el curso normal de los eventos de cada uno. (Ver los demás en Anexos).

Caso de Uso:	Gestionar participación en actividades culturales
Actores:	Estudiante FEU de cultura
Resumen:	El caso de uso se inicia cuando el estudiante FEU necesita gestionar la participación de un estudiante en una actividad cultural (registrar participación o eliminar participación) y realiza una búsqueda de la actividad sobre la que desea trabajar filtrándola por la fecha de realización de esta. Si la actividad seleccionada tiene alguna realización en la fecha entrada el sistema muestra un listado con los participantes en dicha actividad (nombre, apellidos). Luego el estudiante de la FEU selecciona una de las opciones disponibles (registrar participación o eliminar participación terminando así el caso de uso).
Precondiciones:	El usuario ha sido identificado como estudiante FEU de cultura.
Referencias	R10.1,R10.2
Prioridad	Crítico
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema

Capítulo 3: Análisis y diseño del sistema

<p>1-El Estudiante FEU accede a la interfaz que le corresponde.</p> <p>3-El estudiante FEU selecciona la actividad a buscar e introduce la fecha por la que desea filtrar.</p> <p>5. El estudiante FEU elige la acción a realizar.</p>	<p>2. El sistema muestra un formulario con las actividades disponibles y un campo de entrada para la fecha.</p> <p>4. El sistema muestra un listado con todos los estudiantes.</p> <p>6.</p> <ul style="list-style-type: none"> ✓ Si elige registrar participación en actividad cultural debe ir a la sección Registrar participación en actividad cultural. ✓ Si elige eliminar participación en actividad cultural debe ir a la sección Eliminar participación en actividad cultural.
Sección “Registrar participación en actividad cultural”	
	1. El sistema muestra una interfaz para realizar una búsqueda de estudiantes mediante el nombre.
2. El estudiante FEU introduce datos de la actividad a buscar.	3. El sistema realiza la búsqueda de dicha actividad.
	4. La actividad y el estudiante son encontrados.
5. El estudiante FEU introduce los datos de la participación.	6. Queda registrada su participación finalizando el caso de uso.
	7. Se muestra un mensaje satisfactorio.
Flujos Alternos	
Acción del actor	Respuesta del sistema
Acción 4	Si el estudiante y la actividad no son encontrados el sistema emite un mensaje de que no existen.
Sección “Eliminar participación en actividad cultural”	
	1. El sistema busca una interfaz de

Capítulo 3: Análisis y diseño del sistema

	eliminar participación en actividad cultural.
2. El estudiante FEU escoge lo que desea eliminar y presiona aceptar.	3. El sistema procede a obtener el estudiante y a la eliminación de la actividad seleccionada, y a la actualización de la base de datos, terminando así el caso de uso
.	4. Se muestra un mensaje satisfactorio.
Flujos alternos	
Acción del actor	Respuesta del sistema
Prototipo de interfaz	
Poscondiciones	Se actualiza en la base de datos la tabla que contiene la participación de las actividades culturales.

Tabla 5: Descripción del CU Gestionar participación en actividades culturales.

3.6: Diagramas de clases del análisis

Un diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas. Ver demás en anexos.

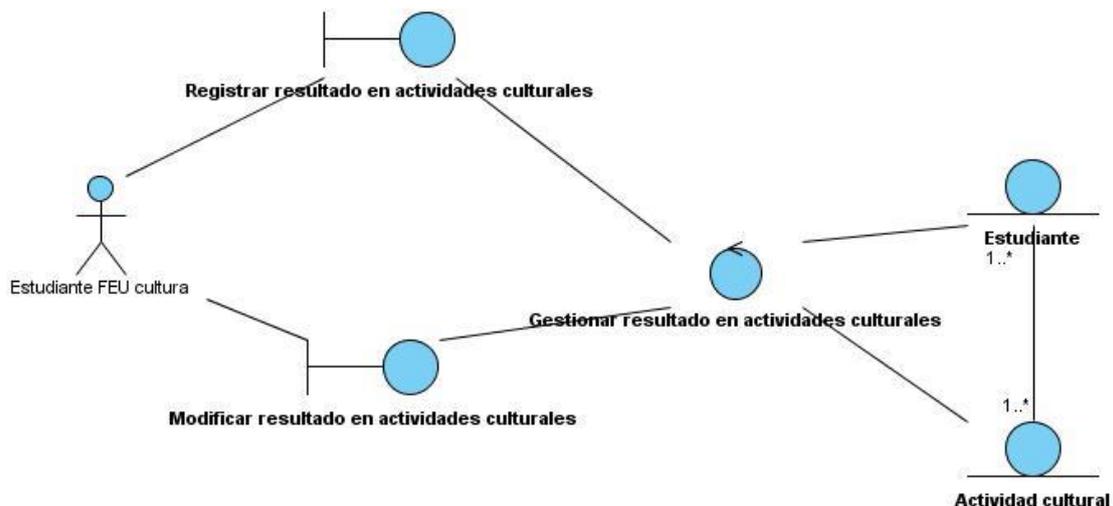


Figura 17: DCA. CU del sistema Gestionar resultados en Actividades Culturales.

Capítulo 3: Análisis y diseño del sistema

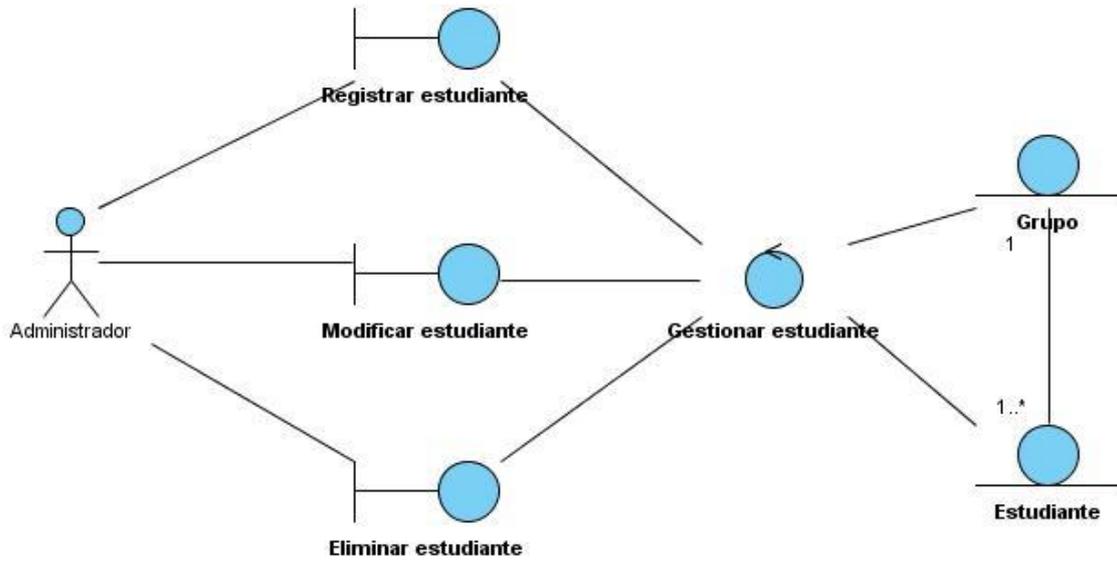


Figura 188: DCA. CU del sistema Gestionar estudiante.

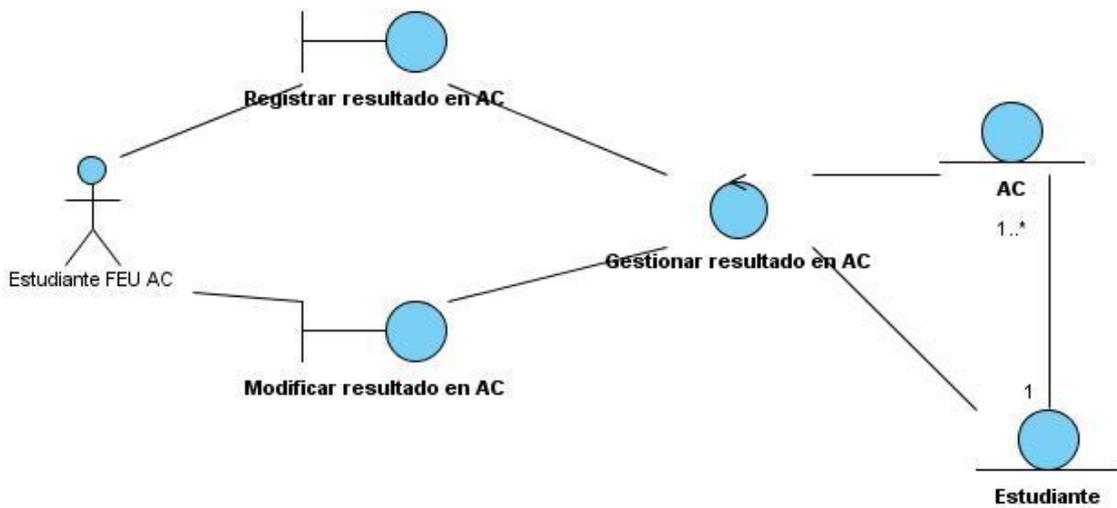


Figura19: DCA. CU del sistema Gestionar resultado en AC.

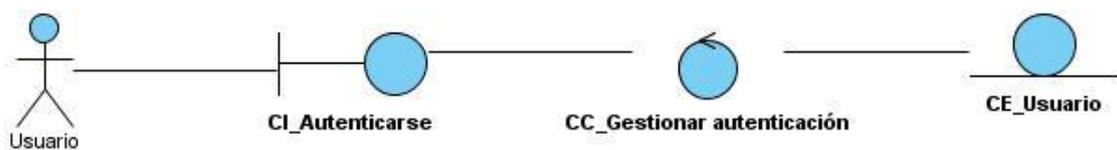


Figura 190: DCA. CU del sistema Autenticar usuario.

3.7: Diagramas de colaboración

Un diagrama de colaboración es una forma alternativa al diagrama de secuencia de mostrar un escenario. Este tipo de diagrama muestra interacciones entre objetos organizados entorno a los objetos y los enlaces entre ellos. Los diagramas de

Capítulo 3: Análisis y diseño del sistema

colaboración explican gráficamente las interacciones entre las instancias del modelo de objetos. (Nicot, 2008).

En la figura 28 y 29 se muestra el diagrama de colaboración para el caso de uso Gestionar participación en Actividades Culturales. Ver demás en anexos.

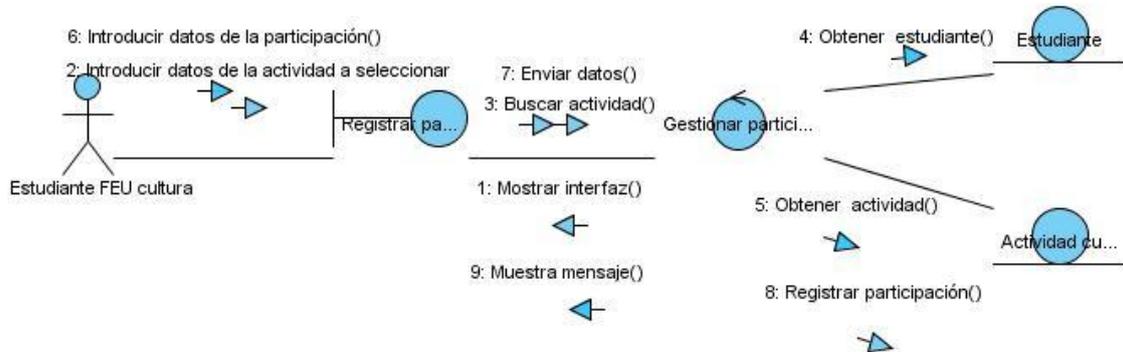


Figura 201: Diagrama de colaboración del sistema Gestionar participación en actividades culturales (registrar).

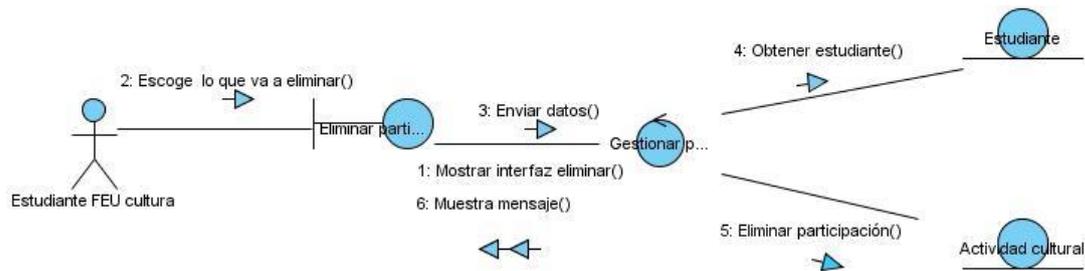


Figura 212: Diagrama de colaboración del sistema Gestionar participación en actividades culturales (eliminar).

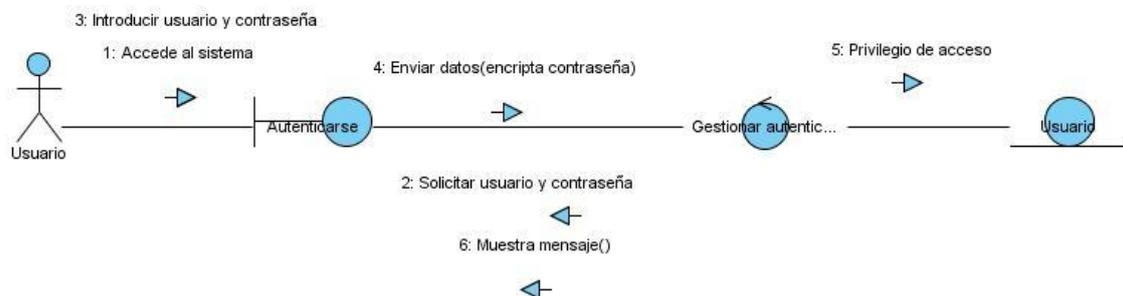


Figura 23: Diagrama de colaboración del sistema Autenticar usuario.

Capítulo 3: Análisis y diseño del sistema



Figura 24: Diagrama de colaboración del sistema Gestionar usuario (registrar).



Figura 25: Diagrama de colaboración del sistema Gestionar usuario (modificar).



Figura 26: Diagrama de colaboración del sistema Gestionar usuario (eliminar).

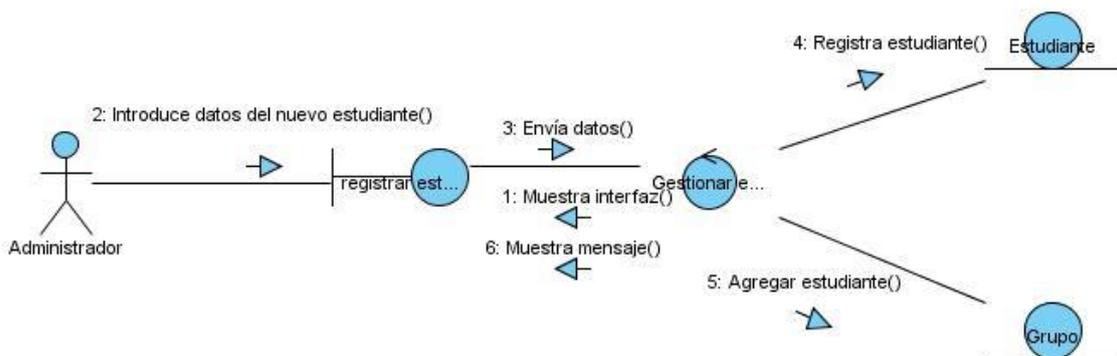


Figura 27: Diagrama de colaboración del sistema Gestionar estudiante (registrar).

Capítulo 3: Análisis y diseño del sistema

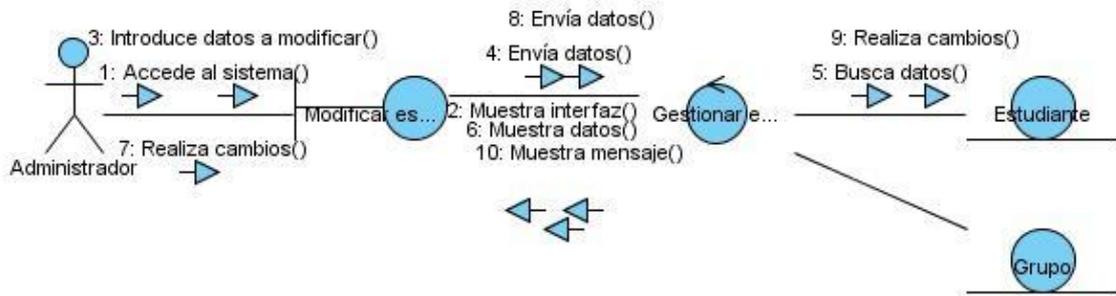


Figura 28: Diagrama de colaboración del sistema Gestionar estudiante (modificar).

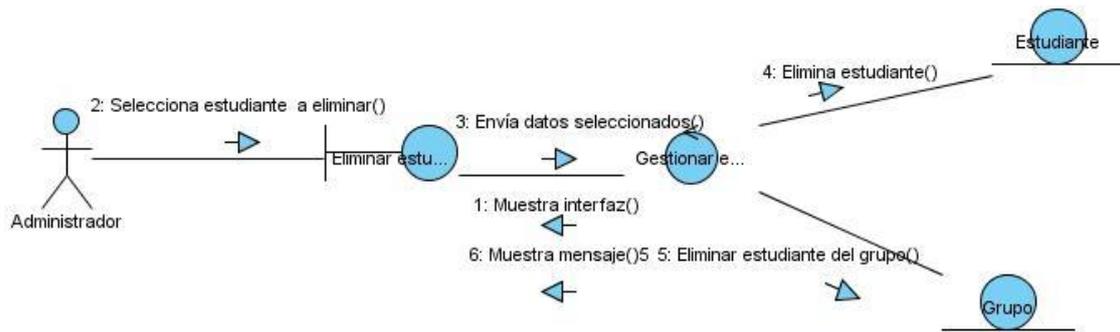


Figura 29: Diagrama de colaboración del sistema Gestionar estudiante (eliminar).

3.8: Diseño

El diseño es una representación significativa de ingeniería de algo que se va a construir. El objetivo es producir un modelo o representación del sistema que será implementado. Como macro actividad posterior al análisis esta se encarga de refinarlo por lo que este debe ser suficiente para el sistema pueda ser erigido sin ambigüedades. En este apartado cobra especial atención los patrones de diseño representando soluciones a problemas comunes en el desarrollo del software. **(Sánchez)**

3.9: Clases significativas de la arquitectura

Entre las clases que afectan directamente el diseño de los respectivos sub módulos y separándolas por los paquetes de incidencia están:

Bean.

La clase EntidadPersistenteBase de la cual heredan todas las clases persistentes.

Dao.

Capítulo 3: Análisis y diseño del sistema

La clase DaoGenericoImpl que encapsula todo el trabajo con el framework de persistencia hibernate y de la cual heredan todas las clases con la responsabilidad de trabajar con la base de datos.

Web

La clase BaseBean del cual heredan todos los beans manejado para el trabajo con jsf, esta clase agrupa funcionalidades afines como la navegación entre páginas, muestra de mensajes, etc.

3.10: Patrones

Para la construcción de dicha solución como se explica en el capítulo anterior se usaron diferentes framework. Dichos framework incorporan muchos patrones como es el caso de Jsf.

En Hibernate, el framework encargado de la persistencia es importante reflejar el uso del patrón proxy, con la finalidad de tener una representación de un objeto, específicamente se utiliza cuando se van a obtener objetos de las base de datos y estos tienen relación con otros objetos, las relaciones se representan por medio de proxy para evitar cargar datos que no serán utilizados por el momento.

Uno de los patrones GOF usados es el Fachada con el fin de proveer una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema, sobre la base de esto están construidas por cada módulo una clase interfaz que unifique todos los servicios que puede brindar dicho subsistema.

El acceso a datos está regido por el patrón DAO (data acces object) que consiste en utilizar un objeto de acceso a datos para abstraer y encapsular todos los accesos a la fuente de datos.

3.11: Diagramas de clases del diseño

A continuación se muestra el diseño de los casos de usos escogidos por los autores del presente documento. Estos son algunos a los cuales se les realizaron el análisis anteriormente.

Capítulo 3: Análisis y diseño del sistema

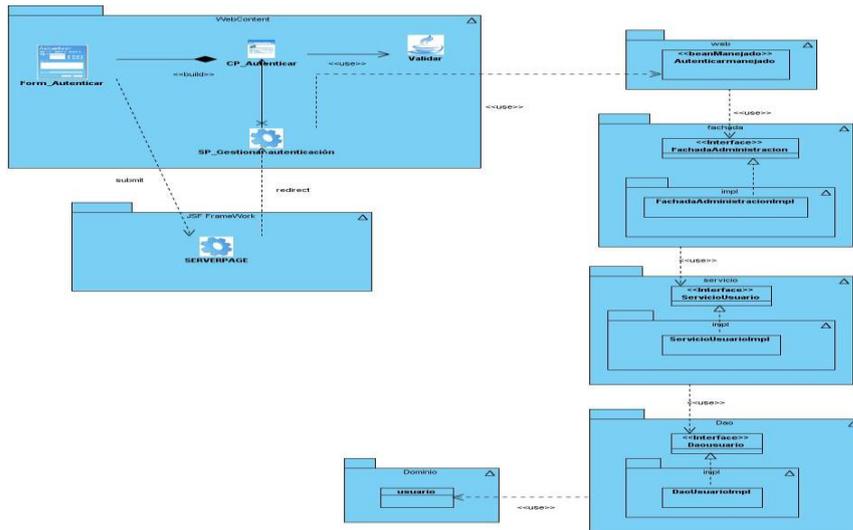


Figura 22: Diagrama del diseño Autenticar usuario.

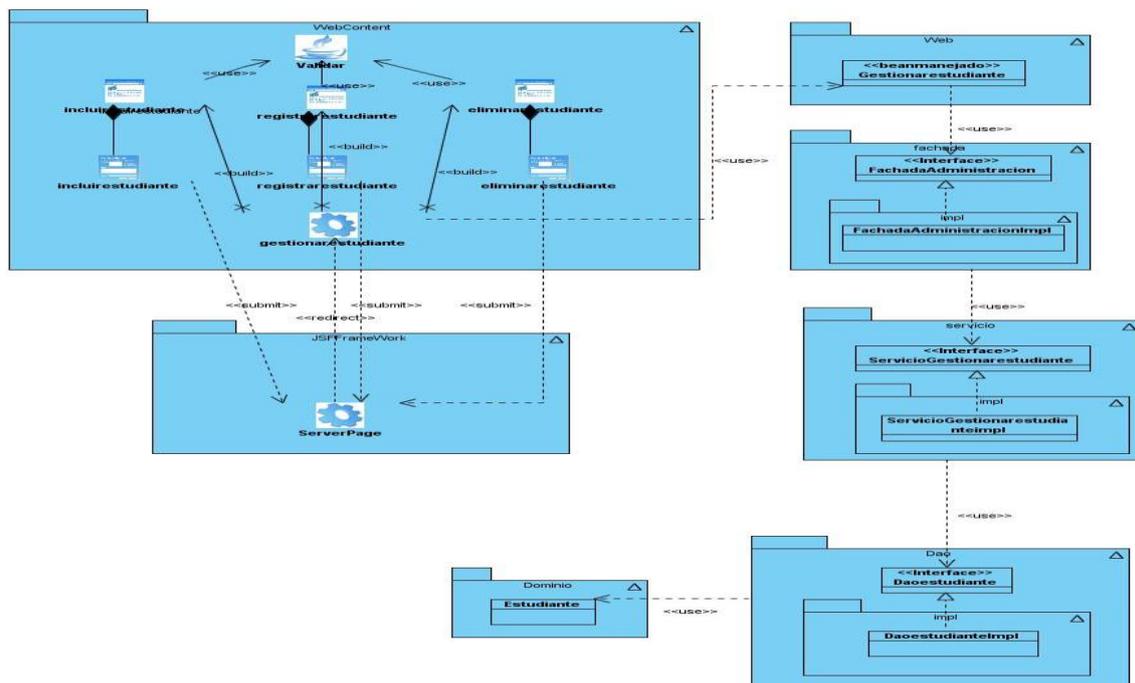


Figura 23: Diagrama del diseño Gestionar estudiante.

Capítulo 3: Análisis y diseño del sistema

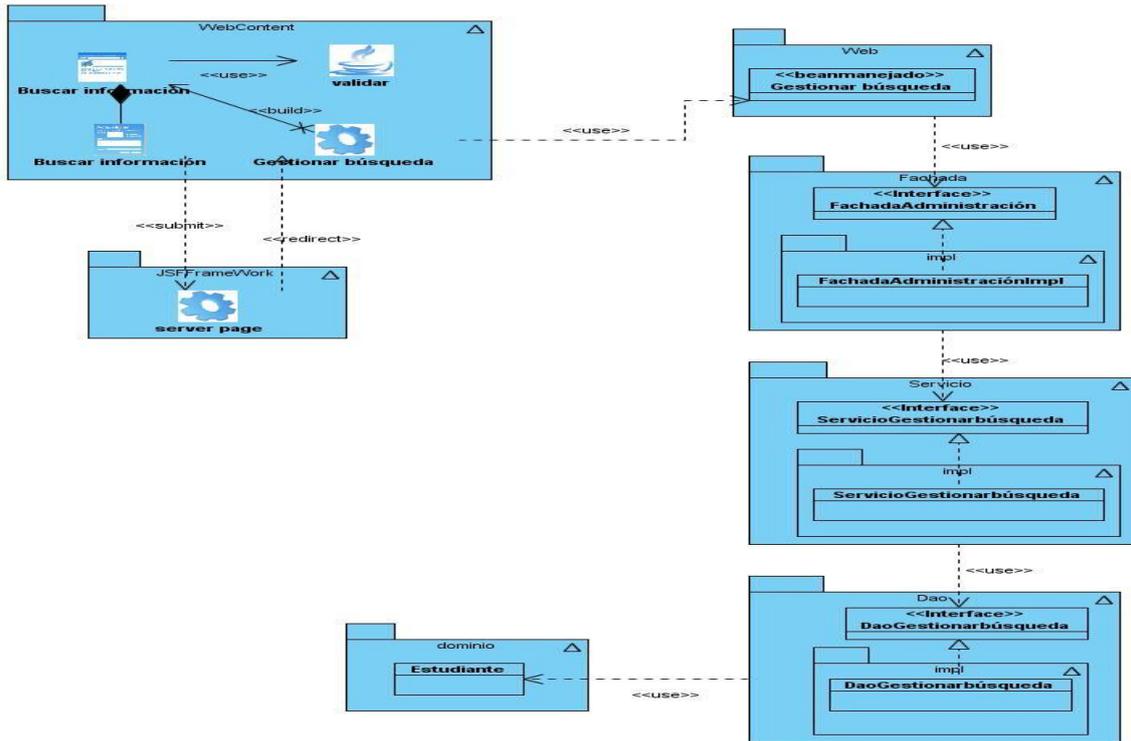


Figura 24: Diagrama del diseño Realizar búsqueda de estudiante.

3.12: Conclusiones

En el presente capítulo se trataron los temas referentes al análisis y diseño de un Sistema de Gestión Extradocente partiendo de los artefactos generados en el flujo de levantamiento de requisitos. Como resultado de las tareas se obtuvieron los diagramas de clases de análisis y diseño así como las realizaciones de los casos de usos; estos contribuyeron a la creación de los subsistemas de implementación y los diagramas de implementación.

Capítulo 4: Implementación y pruebas

En este capítulo se presentarán los artefactos más importantes del flujo de trabajo de implementación, el modelo de despliegue y el modelo de implementación para una mejor descripción de las vistas estáticas del sistema. Se presentan además los tipos de prueba a realizarle al sistema para verificar su correcto funcionamiento con el objetivo de encontrar posibles fallos y corregirlos antes de la liberación del producto.

4.1: Modelo de implementación

Consiste en una visión general de lo que tiene que ser implementado y un apartado para cada iteración con los componentes y subsistemas a implementar durante esa iteración. Se generan una serie de artefactos que constituyen la composición física de la implementación del sistema como son los diagramas de subsistemas y componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares, donde se detalla esencialmente la relación que existe desde las clases y componentes del modelo de diseño a subsistemas y componentes físicos. En este modelo se ajustan los subsistemas formados por los elementos de implementación y se definen las dependencias entre subsistemas.

4.2.1: Diagrama de subsistemas de implementación

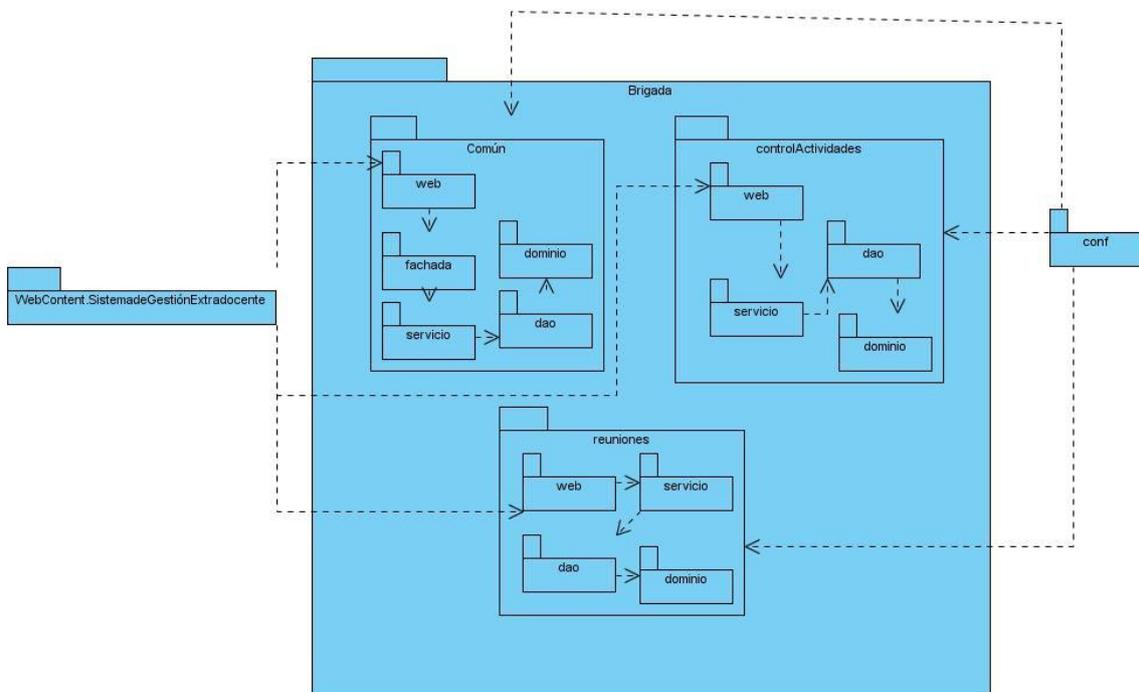


Figura 25 : Diagrama de Subsistema de implementación.

4.2: Diagrama de componentes

Un diagrama de componentes ilustra los fragmentos de software y los controladores que conforman un sistema, permiten editar de forma segura todo el contenido que llevarán los archivos de datos de la aplicación. Tienen un nivel de abstracción más elevado que un diagrama de clases. Usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Estos diagramas se utilizan para describir la vista de implementación estática de un sistema.

A continuación se representa el diagrama de componente:

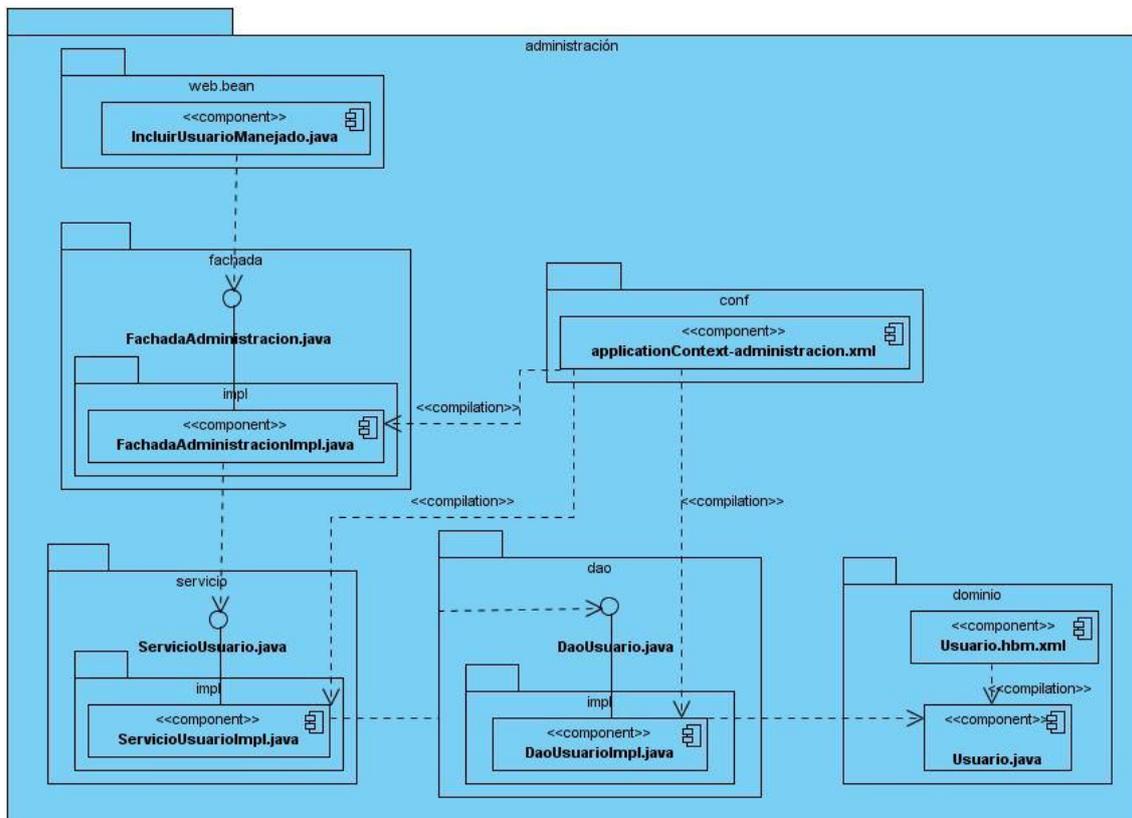


Figura 26: Diagrama de componentes paquete Administración.

4.3: Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes *hardware* y *software* en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes *software* (procesos y objetos que se ejecutan en ellos).

Estarán formados por instancias de los componentes *software* que representan manifestaciones del código en tiempo de ejecución (los componentes que sólo sean

Capítulo 4: Implementación y pruebas

utilizados en tiempo de compilación deben mostrarse en el diagrama de componentes). Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes *software*, objetos, procesos (caso particular de un objeto). **(Acosta.)**

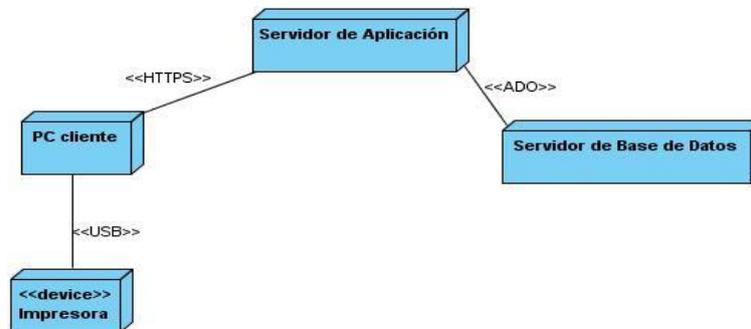


Figura 27: Diagrama de Despliegue.

4.4: Pruebas

Las pruebas del software se integran dentro de las diferentes fases del ciclo de AUP, estas se centran principalmente en la evaluación o la valoración de la calidad del producto.

4.5: Métodos de Pruebas

Entre los métodos de pruebas utilizados en la validación de la propuesta de solución está:

Prueba de Caja Negra.

Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. Una prueba de caja negra examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software.

4.6: Tipos de Pruebas

Durante el flujo de trabajo de pruebas verificamos el resultado de la implementación probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema a ser entregadas a los clientes.

Dentro de los tipos de prueba definidos para la propuesta se encuentran las pruebas de funcionalidad, Seguridad, Disponibilidad y Red, Rendimiento, Compatibilidad,

Capítulo 4: Implementación y pruebas

Resistencia o Stress y Usabilidad. De los mismos, solo las pruebas de funcionalidad son las que se adaptan al rol.

Prueba de funcionalidad

Estas pruebas tienen como objetivo verificar la funcionalidad del sistema al fijar la tensión en la validación de las funciones, métodos, servicios y casos de usos.

Metas

Validar que la aplicación:

- Cumpla con los requisitos funcionales especificados en el diseño de la solución por medio de casos de usos.
- Cumpla con los requisitos no funcionales especificados en el diseño de la solución.

4.7: Niveles de pruebas

Prueba del Sistema: El software ensamblado totalmente con cualquier componente hardware que requiera, se prueba para comprobar que se cumplen los requisitos funcionales. Cualquier pieza de software completo, desarrollado o adquirido, puede verse como un sistema que debe probarse, ya sea para decidir acerca de su aceptación, para analizar defectos globales o para estudiar aspectos específicos de su comportamiento, tales como seguridad o rendimiento. A este tipo de pruebas donde se estudia el producto completo se les llama Pruebas de Sistema.

Pruebas de Aceptación: Estas pruebas las realiza el cliente. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. Estas pruebas no se realizan durante el desarrollo, pues sería impresentable al cliente; sino que se realizan sobre el producto terminado e integrado.

Se hicieron 3 iteraciones, en la primera cuando se le hicieron las respectivas pruebas a la aplicación se obtuvieron 15 no conformidades, una vez recogidas se arreglaron entonces se hizo una segunda iteración donde se obtuvieron otras 5 disminuyendo en una escala considerable el número de no conformidades y quedando completamente erradicadas en una tercera iteración, quedando el cliente satisfecho con lo hecho.

Tomando en cuenta los datos anteriores, se muestra la siguiente gráfica donde se representa la relación de las no conformidades.

Capítulo 4: Implementación y pruebas

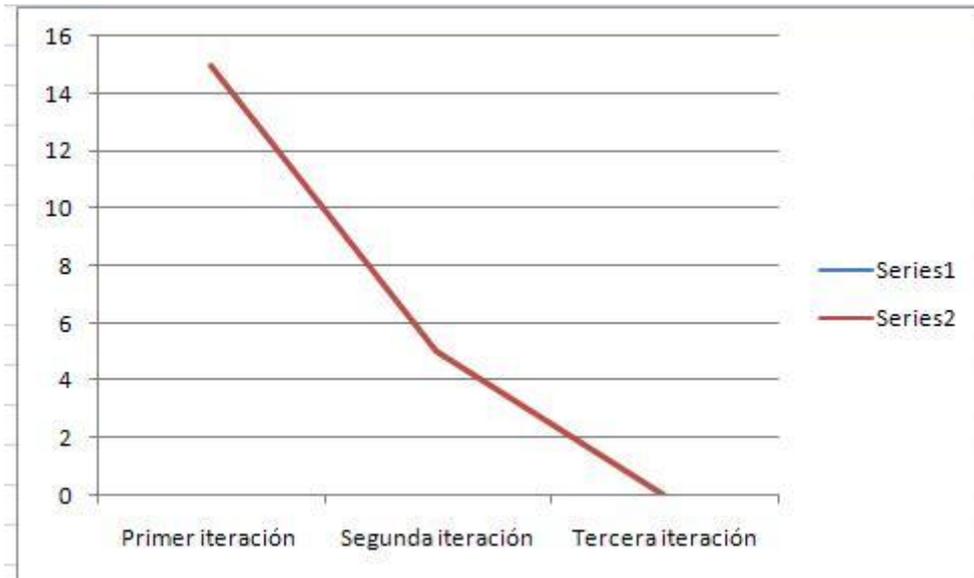


Figura 28: Comportamiento de las no conformidades.

4.8: Conclusiones

Concluido este capítulo se puede afirmar que el diagrama de componentes es una herramienta muy importante y versátil cuando se refiere a modelar la vista estática de un sistema, y ayuda a determinar qué componentes pueden compartirse entre diferentes partes funcionales. El modelo de despliegue genera una organización de los dispositivos que se necesitan y su distribución física. Los casos de prueba por su parte colaboran con el objetivo de lograr un sistema completamente funcional y libre de errores.

Conclusiones

Conclusiones

- Se logró dar cumplimiento al desarrollo de un sistema que permita gestionar la información relacionada con la participación y los resultados de los estudiantes en las diferentes actividades extradocentes.
- La aplicación fue desarrollada siguiendo la metodología de desarrollo AUP, utilizando UML para la especificación, construcción y documentación del sistema y Visual Paradigm para la creación de los artefactos que se generaron durante el ciclo de vida del software, permitiendo que el trabajo de desarrollara con una mayor eficiencia.
- Con los requisitos funcionales quedaron claras todas las funcionalidades que debe soportar el sistema, para dar cumplimiento a las necesidades y expectativas del cliente.

Recomendaciones

Recomendaciones

- Desplegar el software en las condiciones especificadas en los requisitos.
- Extender el negocio a otras áreas de la FEU para el futuro desarrollo de la aplicación con el objetivo de incorporarle nuevas funcionalidades.

Bibliografía Referenciada

Bibliografía Referenciada

(AUP), The Agile Unified Process. *The Agile Unified Process (AUP).* The Agile Unified Process (AUP). [Online] [Cited: febrero 19, 2010.] <http://www.ambyssoft.com/unifiedprocess/agileUP.html>.

Alvarez, Miguel Angel. *Desarrolloweb.com.* La tecnología Java para la creación de páginas web con programación en el servidor. [Online] [Cited: febrero 17, 2010.] <http://www.desarrolloweb.com/articulos/831.php>.

Bartle, Phil. *INFORMACIÓN PARA LA GESTIÓN y gestión de la información.* INFORMACIÓN PARA LA GESTIÓN y gestión de la información. [Online] [Cited: febrero 19, 2010.] <http://www.scn.org/mpfc/modules/mon-miss.htm>.

Chamizo, Mario Yoandry. Portal de Arquitectura del software de la facultad 8.

Crespo, César. *Introducción a Hibernate.* Introducción a Hibernate. [Online] [Cited: febrero 12, 2010.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=hibernate>.

desarrolloweb.com. *desarrolloweb.com.* [Online] [Cited: febrero 12, 2010.] <http://www.desarrolloweb.com/articulos/436.php>.

González, MsC. Yanetsys Sarduy Domínguez y Lic. Pedro Urra. *Sistemas de gestión de contenidos: En busca de una plataforma ideal.* Sistemas de gestión de contenidos: En busca de una plataforma ideal. [Online] [Cited: febrero 19, 2010.] http://bvs.sld.cu/revistas/aci/vol14_4_06/aci11406.htm.

Guevara, Humberto Rivero. 2008. Análisis, diseño e implementación del módulo Aprehensión del SIIPOL. 2008.

Hondares, Yeleyne Maure Díaz y Yaimara Granados. 2009. Análisis, Diseño e Implementación del módulo EXPERTICIAS CRIMINALÍSTICAS v2.0 del Sistema de Investigación e Información Policial. 2009.

IDE, Desarrollo de aplicaciones multiplataforma con NetBeans. *Desarrollo de aplicaciones multiplataforma con NetBeans IDE.* [Online] [Cited: febrero 10, 2010.] http://www.sun.com/emrkt/innercircle/newsletter/latam/0207latam_feature.html.

Insight. *Insight.* [Online] [Cited: febrero 10, 2010.] <http://www.insight.com.uy/insight.html>.

java, Spring:framework de. *Spring:framework de java.* [Online] [Cited: febrero 12, 2010.] <http://sentidoweb.com/2006/12/26/spring-framework-de-java.php>.

León, Lic. Oscar Santana. *Sistema de Gestión Consular.* Sistema de Gestión Consular. [Online] [Cited: febrero 19, 2010.] <http://espejos.unesco.org.uy/simplac2002/Ponencias/Gobierno%20en%20%EDnea/EL022.doc>.

Lockhart, Thomas. *Tutorial de PostgreSQL.* Tutorial de PostgreSQL. [Online] [Cited: febrero 12, 2010.] <http://sdi.bcn.cl/desarrollo/doctos/PostgreSQL%20-%20Tutorial.pdf>.

Mark Grand, Wiley. Patterns in Java.

Bibliografía Referenciada

Acosta., Marcia Leidis Rodríguez Guerra. Alcides Aguilera. Análisis, Diseño e Implementación del Sistema Estudio de Contrarios para el Voleibol.

Nicot, Liumila Laffita. 2008. "Gestión de la participación y los resultados obtenidos por los estudiantes en las actividades extracurriculares". 2008.

P.Charvat, Jason. *Rational Rose simplifica el desarrollo de software.* Rational Rose simplifica el desarrollo de software. [Online] [Cited: febrero 12, 2010.]
http://translate.google.com/cu/translate?hl=es&langpair=en%7Ces&u=http://articles.techrepublic.com.com/5100-10878_11-1049781.html..

Page, Ventajas y desventajas del Personal Home. *Ventajas y desventajas del Personal Home Page.* Ventajas y desventajas del Personal Home Page. [Online] [Cited: febrero 19, 2010.]
[http://www.creargratisunapaginaweb.com/PHP/Ventajas-y-desventajas-del-Personal-Home-Page-4/.](http://www.creargratisunapaginaweb.com/PHP/Ventajas-y-desventajas-del-Personal-Home-Page-4/)

Porta, Santiago. *Frameworks y herramientas.* Frameworks y herramientas. [Online] [Cited: febrero 12, 2010.] <http://www.santiagoporta.com.ar/site/frameworks-y-herramientas.html>.

Quiroga, Lic. Lourdes Aja. 2002. *Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones.* [Online] agosto 8, 2002. [Cited: FEBRERO 10, 2010.]
http://bvs.sld.cu/revistas/aci/vol10_5_02/aci04502.htm.

Ramírez, Iván. "Aplicación para desarrollo en Java". "Aplicación para desarrollo en Java". [Online] [Cited: febrero 10, 2010.] [http://netbeans-ide.softonic.com/.](http://netbeans-ide.softonic.com/)

Rivalta, Mario Yoandry Chamizo. Portal de Arquitectura del software de la facultad 8.

Sánchez, Adriel Alejandro Aliaga benavides y Eduardo Alfonso. Análisis, Diseño e Implementación de los submódulos Denuncia y Control de Investigación pertenecientes al Módulo de Investigación Penal del Sistema de Investigación e Información Policial(SIIPOL).

Silberschatz, Abraham. *FUNDAMENTOS DE BASES DE DATOS cuarta edición.* FUNDAMENTOS DE BASES DE DATOS cuarta edición. [Online] [Cited: febrero 12, 2010.]
<http://bibliotecavirtualeive.files.wordpress.com/2008/08/fundamentos-de-bases-de-datos.pdf>.

softonic, Equipo de. "La mejor alternativa gratuita a Oracle y a SQL Server". "La mejor alternativa gratuita a Oracle y a SQL Server". [Online] [Cited: febrero 10, 2010.] [http://mysql.softonic.com/.](http://mysql.softonic.com/)

software, Sitio de descargas de. 2007. *Sitio de descargas de software.* Sitio de descargas de software. [Online] Mar 05, 2007. [Cited: febrero 10, 2010.]
[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(Iglesia_Anglicana\)_%5BMac_OS_X_cuenta_14717_p/.](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(Iglesia_Anglicana)_%5BMac_OS_X_cuenta_14717_p/)

Bibliografía

Bibliografía

Acosta., Marcia Leidis Rodríguez Guerra. Alcides Aguilera. Análisis, Diseño e Implementación del Sistema Estudio de Contrarios para el Voleibol.

Agüero, Martín. 2007. Introducción a Spring Framework. 2007.

Case, Herramientas. Herramientas Case. Herramientas Case. [Online] [Cited: febrero 17, 2010.] <http://www.slideshare.net/guest15d257/herramientas-case-508428>.

Eclipse. [Online] [Cited: enero 12, 2010.] <http://www.eclipse.org/>.

Ecuador, Pontificada Universidad Católica del. HERRAMIENTAS CASE.

El lenguaje de Programación Java.

Guerrero, Luis A. Taller de UML. s.l. : Universidad de Chile Departamento de Ciencias de la Computación.

Java, Tecnología. Tecnología Java. Tecnología Java. [Online] [Cited: febrero 17, 2010.] <http://www.slideshare.net/Dryope/tecnologia-java-420996>.

2005. Métodos de prueba de software. [Online] octubre 13, 2005. <http://pruebasoftware.blogcindario.com/2005/10/00001-metodos-de-prueba-del-software.html>.

Pressman, Roger. Ingeniería de Software un Enfoque Práctico. 2006.

Sánchez, Adriel Alejandro Aliaga benavides y Eduardo Alfonso. Análisis, Diseño e Implementación de los submódulos Denuncia y Control de Investigación pertenecientes al Módulo de Investigación Penal del Sistema de Investigación e Información Policial(SIIPOL). .

Serrano, Javier. UML - Lenguaje Unificado de Modelado. UML - Lenguaje Unificado de Modelado. [Online] [Cited: febrero 16, 2010.] <http://programando.foroactivo.com.es/umd-200901-ingenieria-del-software-f3/uml-lenguaje-unificado-de-modelado-t9.htm>.

Glosario de Términos

Glosario de términos

- CU: Abreviatura de la palabra caso de uso.
- DCA: Diagrama de clases del análisis.
- CASE: Ingeniería de Software Asistida por Ordenador.
- UML: Lenguaje unificado de modelado.
- DAO: Data Access Object, traducido al español como Objeto de Acceso a Datos, es un patrón de diseño de clases que permite a quien lo aplique, suministrar una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos.
- AUP: Proceso Ágil Unificado.
- J2EE: Java 2 Enterprise Edition, define un estándar para el desarrollo de aplicaciones empresariales multicapa diseñado por Sun Microsystems. Simplifica las aplicaciones empresariales basándolas en componentes modulares y estandarizados Bytecode: es un código intermedio más abstracto que el código máquina. Habitualmente es tratado como un fichero binario que contiene un programa ejecutable.
- GPL2: Licencia Pública General.
- Hibernate: Framework de Mapeo - Objeto – Relacional para la plataforma java, con una versión para .NET que permite mapear los atributos de las clases con las entidades de las base de datos, abstrayendo al programador del trabajo con el paradigma relacional.

Anexos

eyeOS | Web Desktop - Web OS - Web Office - your files and applications everywhere! - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://eyeos.org/en/?section=spanish

Getting Started Latest Headlines

Open Source Platform

EYEOS HOMEPAGE | PROFESSIONAL SERVICES | DEVELOPERS COMMUNITY
ENGLISH | CATALÀ | DEUTSCH | ESPAÑOL | FRANÇAIS | PORTUGUÉS

Home Concept Blog Applications Demo Downloads Tour Forums Wiki Use eyeOS

Welcome
The concept
Download eyeOS 1.6!

MOBILE SYNC YOUR COMPUTER

Introducing eyeOS 1.6 Gala Sync

What is eyeOS?

For users
eyeOS is a new kind of Operating System, where everything resides on a web browser. With eyeOS, you will have your desktop, applications and files always with you, from your home, your college, your office or your neighbour's house. Just open a web browser, connect to your eyeOS System and access your personal desktop and all your stuff just like you left it last time.

For developers
eyeOS introduces a new way of developing web applications with the eyeOS Toolkit, a class library to help the application development and a group of functions to interact with the system.

Use eyeOS!

Download
Get the latest eyeOS and build your own server. eyeOS is Open Source!

Sign up for free!
If you just want to use eyeOS, you can use the eyeOS Public Server.

Donations
eyeOS is Open Source, and everybody can help.

Community

Inicio Alejandro 25, Córdoba... eyeOS | Web Desko... http://026430631.us... ES 15:02

Anexo 1: Ejemplo de EyeOS

Anexos



Anexo 2: Vista de la ventana principal del Sistema.



Anexo 3: Ejemplo del uso de las TIC en la UCI "Akademos".

Anexos

Anexo 4: Ejemplo del uso de las TIC en la UCI” Entorno Virtual de Aprendizaje (EVA)”.

Caso de Uso:	Registrar participación en juegos inter-años
Actores:	Estudiante
Trabajadores:	Jurado(profesores de educación física), Estudiante FEU de deporte
Resumen:	El caso de uso da comienzo cuando el estudiante comunica su disposición de participar en los juegos inter-año y se lo hace llegar al estudiante de la FEU encargado de guardar los datos del estudiante, así como el año al que pertenece .Este estudiante de la FEU le hace llegar los datos a los diferentes jefes de brigada que es el que debe tener el control de la participación de los estudiantes, en todas las actividades y registra la participación del estudiante. Luego el estudiante participa en los juegos y los profesores EF en dependencia de su participación dan un resultado, que es registrado por el estudiante de la FEU y este se lo hace a los diferentes jefes de brigada finalizando así el caso de uso.
Precondiciones:	--

Anexo 5: Caso de uso Registrar participación en juegos inter-años

Anexos

Caso de Uso:	Registrar participación en juegos inter-facultades
Actores:	Estudiante
Trabajadores:	Jurado(profesores de educación física), Estudiante FEU de deporte
Resumen:	Si el resultado obtenido por el estudiante es bueno y este emite su disposición para participar en los juegos inter-facultades da inicio el caso de uso. Luego de la decisión del estudiante, el encargado por parte de la FEU recoge los datos del estudiante y facultad a la que pertenece y estos datos se lo hace llegar a todos los jefes de brigada quienes a su vez registran los datos del estudiante. Durante la participación del estudiante los profesores dan un resultado que es el que obtiene el estudiante durante los juegos y el estudiante de la FEU registra los resultados y este se los hace llegar a todos los jefes de brigada dando fin al caso de uso.
Precondiciones:	--

Anexo 6: Caso de uso Registrar participación en juegos inter-facultades

Caso de Uso:	Audicionar
Actores:	
Trabajadores:	Jurado(coordinadores del centro cultural)
Resumen:	El caso de uso da inicio cuando el estudiante se presenta a la audición. Una vez presentado este realiza el número u obra que presentara a un jurado. Luego de haber visto la presentación del estudiante o el trabajo del estudiante el jurado decide si puede o no presentarse al festival terminando así el caso de uso.
Precondiciones:	--

Anexo 7: Caso de uso Audicionar.

Anexos

Caso de Uso:	Registrar participación en Actividades de Conocimiento(AC)
Actores:	Estudiante
Trabajadores:	Comisión(Estudiante de FEU),jurado(profesores)
Resumen:	<p>El caso de uso da inicio cuando el estudiante informa su disposición para participar en alguno de estos eventos y se lo hace saber a la comisión, la que pide al estudiante sus datos, para el caso de que participe en Mi web x Cuba, JCE, SJM un resumen del trabajo y en el caso que participe en alguna copa el equipo al que pertenece y con esto incluye la participación en una de las comisiones del evento. El caso de uso finaliza cuando el jurado informa el resultado obtenido por el estudiante a la comisión organizadora y este se lo hace llegar a los jefes de brigada.</p> <ul style="list-style-type: none"> - Si el estudiante va participar en mi Web x Cuba ir al caso de usos registrar participación en mi Web x Cuba. - Si el estudiante va participar en Jornada Científica Estudiantil ir al caso de usos registrar participación en Jornada Científica Estudiantil - Si el estudiante va participar en Seminario Juvenil Martiano ir al caso de usos registrar participación en Seminario Juvenil Martiano. - Si el estudiante va participar en Copa IS ir al caso de usos registrar participación en Copa IS. - Si el estudiante va participar en Copa Java ir al caso de usos registrar participación en Copa Java. - Si el estudiante va participar en Copa Pascal ir al caso de usos registrar participación en Copa Pascal.
Precondiciones:	-

Anexo 8: Caso de uso Registrar participación en Actividades de Conocimiento (AC).

Anexos

Caso de Uso:	Registrar participación en Mi Web x Cuba(inter-facultad)
Actores:	Estudiante
Trabajadores:	Comisión(Estudiante de FEU),jurado(profesores),Jurado(coordinadores del centro cultural)
Resumen:	El caso de uso da inicio cuando el estudiante solicita participar en mi Web x Cuba y se lo hace saber a la comisión, la que pide al estudiante sus datos y un resumen del trabajo que presentara y con esto incluye la participación en una de las comisiones del evento. El caso de uso finaliza cuando el jurado informa el resultado a la comisión organizadora los resultados obtenidos por el estudiante y este se los hace llegar a los diferentes jefes de brigadas.
Precondiciones:	Para participar en evento posterior obtener (1er lugar o relevante).

Anexo 9: Caso de uso Registrar participación en Mi Web x Cuba (inter-facultad).

Caso de Uso:	Registrar participación en Jornada Científica Estudiantil(inter-facultad)
Actores:	Estudiante
Trabajadores:	Comisión(Estudiante de FEU),jurado(profesores)
Resumen:	El caso de uso da inicio cuando el estudiante informa su disposición para participar en Jornada Científica Estudiantil y se lo hace saber a la comisión, la que pide al estudiante sus datos y un resumen del trabajo que presentará y con esto incluye la participación en una de las comisiones del evento. El caso de uso finaliza cuando el jurado informa el resultado obtenido por el estudiante y se lo hace llegar a la comisión y este a los diferentes jefes de brigadas.
Precondiciones:	Para participar en evento posterior obtener (1er lugar o relevante).

Anexo 10: Caso de uso Registrar participación en Jornada Científica Estudiantil (inter-facultad).

Anexos

Caso de Uso:	Registrar participación en Seminario Juvenil Martiano(inter-facultad)
Actores:	Estudiante
Trabajadores:	Comisión(Estudiante de FEU),jurado(profesores)
Resumen:	El caso de uso da inicio cuando el estudiante informa su decisión de participar en Seminario Juvenil Martiano y se lo hace saber a la comisión, la que pide al estudiante sus datos y un resumen del trabajo que presentara y con esto incluye la participación en una de las comisiones del evento. El caso de uso finaliza cuando el jurado informa el resultado obtenido por el estudiante y se lo hace llegar a la comisión y este a los diferentes jefes de brigadas.
Precondiciones:	Para participar en evento posterior obtener (1er lugar o relevante).

Anexo 11: Caso de uso Registrar participación en Seminario Juvenil Martiano (inter-facultad).

Caso de Uso:	Registrar participación en Copa Pascal(inter-facultad)
Actores:	Estudiante
Trabajadores:	Comisión(Estudiante de FEU),jurado(profesores)
Resumen:	El caso de uso da inicio cuando el estudiante hace saber su decisión de participar en la copa pascal que se realizara y para ello le hace saber la decisión a la comisión quien pide al estudiante sus datos y el equipo al que pertenece. La comisión registra los datos del estudiante y al equipo, este participa en el evento el jurado informa del resultado obtenido por el estudiante a la comisión registra el resultado y participación y este a los diferentes jefes de brigadas finalizando así el caso de uso.
Precondiciones:	Para participar en evento posterior obtener (1er lugar o relevante).

Anexo 12: Caso de uso Registrar participación en Copa Pascal (inter-facultad).

Caso de Uso:	Registrar participación en Copa ISW(inter-facultad)
Actores:	Estudiante
Trabajadores:	Comisión(Estudiante de FEU),jurado(profesores)

Anexos

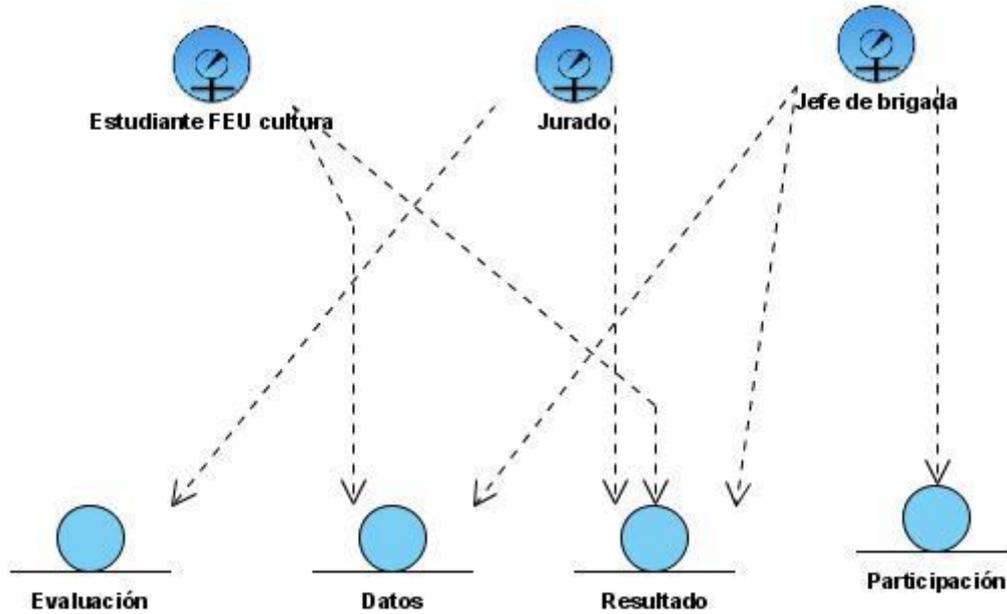
Resumen:	El caso de uso da inicio cuando el estudiante hace saber su decisión de participar en la copa IS que se realizara y para ello le hace saber la decisión a la comisión quien pide al estudiante sus datos y el equipo al que pertenece. La comisión registra los datos del estudiante y el equipo, este participa en el evento y el jurado informa del resultado obtenido por el estudiante a la comisión y este a los diferentes jefes de brigadas finalizando así el caso de uso.
Precondiciones:	Para participar en evento posterior obtener (1er lugar o relevante).

Anexo 13: Caso de uso Registrar participación en Copa ISW (inter-facultad).

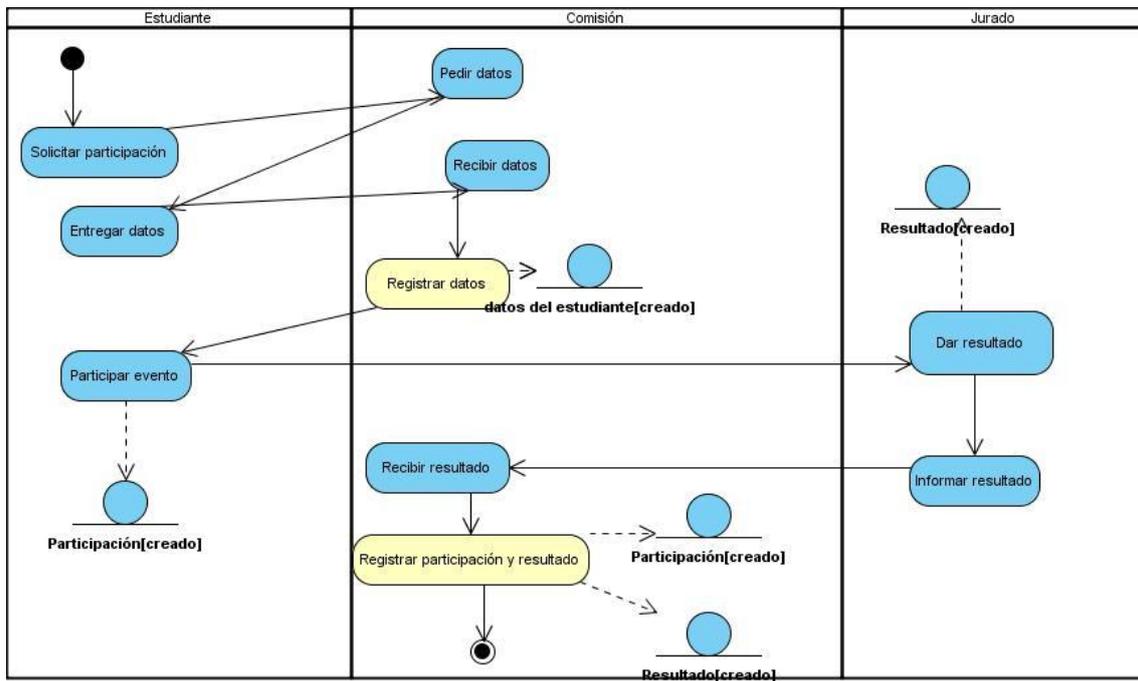
Caso de Uso:	Registrar participación en Copa Java
Actores:	Estudiante
Trabajadores:	Comisión(Estudiante de FEU),jurado(profesores)
Resumen:	El caso de uso da inicio cuando el estudiante hace saber su decisión de participar en la copa java que se realizará y para ello le hace saber la decisión a la comisión quien pide al estudiante sus datos y el equipo al que pertenece. La comisión registra los datos del estudiante y el equipo, este participa en el evento y el jurado informa del resultado obtenido por el estudiante a la comisión y a los diferentes jefes de brigadas finalizando así el caso de uso.
Precondiciones:	Para participar en evento posterior obtener (1er lugar o relevante).

Anexo 14: Caso de uso Registrar participación en Copa Java.

Anexos



Anexo 15: Diagrama de objetos.



Anexo 16: Diagrama de actividades Registrar participación en Actividades de Conocimiento.

Caso de Uso:	Autenticar usuario
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario introduce los datos que se le

Anexos

	piden para acceder a la aplicación, estos se verifican y finaliza dándole los permisos y permitiéndole el acceso a la aplicación.
Precondiciones:	El usuario ha sido validado.
Referencias	R1.1,R1.2,R1.3
Prioridad	Crítico
Flujo Normal de Eventos	
Sección ""	
Acción del Actor	Respuesta del Sistema
1-El usuario entra su usuario y contraseña.	1.1. El sistema inscripta la contraseña. 1.2. Si los datos son correctos se dan los privilegios de acceso terminando así el caso de uso. 1.4. Muestra un mensaje satisfactorio.
Flujos Alternos	
Acción del actor	Respuesta del sistema
Acción 1	1.2. En caso de no existir los permisos se envía un mensaje de aviso.
Prototipo de interfaz	
Poscondiciones	Se habilitan las funcionalidades según los privilegios.

Anexo 17: Caso de uso Autenticar usuario.

Caso de Uso:	Gestionar usuario
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador solicita realizar la gestión de un usuario. Este puede registrar, modificar o eliminar un usuario, para ello debe realizar la búsqueda del mismo, dado un criterio de búsqueda, luego de esto selecciona de la lista mostrada al usuario que desea realizarle algún cambio, al realizar esta acción deberá presionar el botón aceptar para guardar los cambios en la base de datos.
Precondiciones:	El usuario ha sido identificado como administrador.
Referencias	R2.1,R2.2,R2.3
Prioridad	Crítico
Flujo Normal de Eventos	
Sección ""	
Acción del Actor	Respuesta del Sistema

Anexos

1-El administrador selecciona la opción de gestionar usuario.	2. El sistema le brinda las opciones referentes a los usuarios. <ul style="list-style-type: none"> ✓ Registrar usuario. ✓ Modificar usuario. ✓ Eliminar usuario.
3. El administrador elige la acción a realizar.	4. <ul style="list-style-type: none"> ✓ Si elige Registrar usuario debe ir a la sección Registrar usuario. ✓ Si elige Modificar usuario debe ir a la sección Modificar usuario. ✓ Si elige Eliminar usuario debe ir a la sección Eliminar usuario.
Sección “Registrar usuario”	
	1. El sistema muestra una interfaz para introducir los datos que serán registrados, mostrando un formulario con los campos generales que se deben introducir.
2. El administrador introduce el nombre del usuario, privilegio, usuario y contraseña.	3. El sistema busca el usuario y muestra sus datos.
	4. El sistema registra el nuevo usuario terminando así el caso de uso.
	5. Se muestra un mensaje satisfactorio.
Flujos Alternos	
Acción del actor	Respuesta del sistema
Acción 3	Si el usuario ya existe debe emitir un mensaje de que no existe el usuario y no realiza el registro.
Sección “Modificar usuario”	
	1. El sistema muestra la interfaz de modificar usuario.
2. El administrador introduce el usuario de la persona a la que le quiere modificar los datos.	3. El sistema busca al usuario y muestra sus datos.
4. El administrador realiza los cambios que	5. El sistema verifica que no hayan

Anexos

le interesan y ordena actualizar dichas modificaciones.	introducido datos incoherentes y hace los cambios.
	6. El sistema actualiza los datos en la base de datos terminando así el caso de uso.
	7. Se muestra un mensaje satisfactorio.
Flujos alternos	
Acción del actor	Respuesta del sistema
Acción 5	Si el administrador introduce algún dato incorrecto, emite error” ha entrado datos incorrectos”.
Prototipo de interfaz	
Sección “Eliminar usuario”	
	1. El sistema muestra la interfaz de eliminar usuario, la cual tiene un listado con todos los usuarios.
	2. El sistema brinda la opción de mostrar los usuarios por privilegios.
3. El administrador escoge la opción a realizar.	4. <ul style="list-style-type: none"> ✓ Si escoge mostrar los usuarios por privilegio para poder eliminarlos, ir a paso 5. ✓ Si desea eliminar directamente cuando se muestran todos los usuarios, ir a paso 7.
5. El administrador selecciona el privilegio del cual quiere mostrar los usuarios.	6. Muestra los usuarios que pertenecen a ese privilegio.
7. El administrador selecciona el usuario que desea eliminar y ordena que se elimine.	8. El sistema elimina el usuario terminando así el caso de uso.
	9. Se muestra un mensaje satisfactorio.
Flujos alternos	
Acción del actor	Respuesta del sistema
Acción 5	Si selecciona todo el sistema muestra todos

Anexos

	los usuarios.
Prototipo de interfaz	
Poscondiciones	Fueron registrados los datos del usuario. Fueron modificados los datos del usuario. Fue creada o eliminada una instancia usuario.

Anexo 18: Caso de uso Gestionar usuario.

Caso de Uso:	Gestionar estudiante
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador solicita realizar la gestión de un estudiante. Este puede registrar, modificar o eliminar un estudiante, para ello debe realizar la búsqueda del mismo, dado un criterio de búsqueda, luego de esto selecciona de la lista mostrada al usuario que desea realizarle algún cambio, al realizar esta acción deberá presionar el botón aceptar para guardar los cambios en la base de datos.
Precondiciones:	El usuario ha sido identificado como administrador.
Referencias	R15.1,R15.2,R15.3
Prioridad	Crítico

Flujo Normal de Eventos

Sección ""

Acción del Actor	Respuesta del Sistema
1-El administrador selecciona la opción de gestionar estudiante.	2. El sistema le brinda las opciones referentes a los usuarios. <ul style="list-style-type: none"> ✓ Registrar estudiante. ✓ Modificar estudiante. ✓ Eliminar estudiante.
3. El administrador elige la acción a realizar.	4. <ul style="list-style-type: none"> ✓ Si elige Registrar estudiante debe ir a la sección Registrar estudiante. ✓ Si elige Modificar estudiante. debe ir a la sección Modificar estudiante. ✓ Si elige Eliminar estudiante debe ir a la sección Eliminar estudiante.

Anexos

Sección "Registrar estudiante"	
	1. El sistema muestra una interfaz para introducir los datos que serán registrados, mostrando un formulario con los campos generales que se deben introducir.
2. El administrador introduce el nombre del estudiante, privilegio, usuario, contraseña y grupo.	3. El sistema busca el estudiante y muestra sus datos.
	4. El sistema registra el nuevo estudiante terminando así el caso de uso.
	5. Se muestra un mensaje satisfactorio.
Flujos Alternos	
Acción del actor	Respuesta del sistema
Acción 3	Si el estudiante ya existe debe emitir un mensaje de que no existe el estudiante y no realiza el registro.
Sección "Modificar estudiante"	
	1. El sistema muestra la interfaz de modificar estudiante.
2. El administrador introduce el usuario de la persona a la que le quiere modificar los datos.	3. El sistema busca al usuario y muestra sus datos.
4. El administrador realiza los cambios que le interesan y ordena actualizar dichas modificaciones.	5. El sistema verifica que no hayan introducido datos incoherentes y hace los cambios.
	6. El sistema actualiza los datos en la base de datos terminando así el caso de uso.
	7. Se muestra mensaje satisfactorio.
Flujos alternos	
Acción del actor	Respuesta del sistema
Acción 5	Si el administrador introduce algún dato incorrecto, emite error" ha entrado datos incorrectos".
Prototipo de interfaz	
Sección "Eliminar estudiante"	
	1. El sistema muestra la interfaz de eliminar estudiante, la cual tiene un listado con todos los estudiantes.

Anexos

	2. El sistema brinda la opción de mostrar los estudiantes por privilegios.
3. El administrador escoge la opción a realizar.	4. ✓ Si escoge mostrar los estudiantes por privilegio para poder eliminarlos, ir a paso 5. ✓ Si desea eliminar directamente cuando se muestran todos los usuarios, ir a paso 7.
5. El administrador selecciona el privilegio del cual quiere mostrar los estudiantes.	6. Muestra los estudiantes que pertenecen a ese privilegio.
7. El administrador selecciona el estudiante que desea eliminar y ordena que se elimine.	8. El sistema elimina el estudiante terminando así el caso de uso.
	9. Se muestra un mensaje satisfactorio.
Flujos alternos	
Acción del actor	Respuesta del sistema
Acción 5	Si selecciona todo, el sistema muestra todos los estudiantes.
Prototipo de interfaz	
Poscondiciones	Fueron registrados los datos del estudiante. Fueron modificados los datos del estudiante. Fue creada o eliminada una instancia estudiante.

Anexo 19: Caso de uso Gestionar estudiante.

Caso de Uso:	Gestionar AC
Actores:	Estudiante FEU AC
Resumen:	El caso de uso se inicia cuando el estudiante FEU solicita realizar la gestión de una AC. Este puede registrar, modificar o eliminar una AC, para ello debe realizar la búsqueda de las AC, dado un criterio de búsqueda, luego de esto selecciona de la lista mostrada la AC que desea realizarle algún cambio, al realizar esta acción deberá presionar el botón aceptar para guardar los cambios en la base de datos.
Precondiciones:	El usuario ha sido identificado como estudiante FEU AC.

Anexos

Referencias	R8.1,R8.2,R8.3
Prioridad	Crítico
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1-El estudiante FEU AC selecciona la opción de gestionar AC.	2. El sistema le brinda las opciones referentes a los usuarios. <ul style="list-style-type: none"> ✓ Registrar AC. ✓ Modificar AC. ✓ Eliminar AC.
3. El estudiante elige la acción a realizar.	4. <ul style="list-style-type: none"> ✓ Si elige Registrar AC debe ir a la sección Registrar AC. ✓ Si elige Modificar AC debe ir a la sección Modificar AC. ✓ Si elige Eliminar AC debe ir a la sección Eliminar AC.
Sección “Registrar AC”	
	1. El sistema muestra una interfaz para introducir los datos que serán registrados, mostrando un formulario con los campos generales que se deben introducir.
2. El estudiante FEU AC introduce los datos de la AC.	3. El sistema busca los datos.
	4. El sistema registra el nuevo usuario terminando así el caso de uso.
	5. Se muestra un mensaje satisfactorio.
Flujos Alternos	
Acción del actor	Respuesta del sistema
Acción 3	Si la AC ya existe debe emitir un mensaje de que existe la AC y no realiza el registro.
Sección “Modificar AC”	
	1. El sistema muestra la interfaz de modificar AC.
2. El estudiante FEU AC introduce los datos	3. El sistema muestra la información.

Anexos

de la actividad a modificar.	
4. El estudiante FEU AC realiza los cambios que le interesan y ordena actualizar dichas modificaciones.	5. El sistema verifica que no hayan introducido datos incoherentes y realiza la modificación.
	6. El sistema actualiza los datos en la base de datos terminando así el caso de uso.
	7. Se muestra un mensaje satisfactorio.
Flujos alternos	
Acción del actor	Respuesta del sistema
Acción 5	Si el estudiante FEU AC introduce algún dato incorrecto, emite error” ha entrado datos incorrectos”.
Prototipo de interfaz	
Sección “Eliminar AC”	
	1. El sistema muestra la interfaz de eliminar AC, la cual tiene un listado con todos las AC.
	2. El sistema pide datos de la actividad que va a eliminar.
3. El estudiante FEU AC introduce datos.	4. El sistema elimina la actividad terminando así el caso de uso.
	5. Se muestra un mensaje satisfactorio.
Flujos alternos	
Acción del actor	Respuesta del sistema
Prototipo de interfaz	
Poscondiciones	Fueron registrados los datos de la AC. Fueron modificados los datos de la actividad. Fue eliminada una actividad.

Anexo 20: Caso de uso Gestionar AC.

Caso de Uso:	Gestionar participación en AC
Actores:	Estudiante FEU AC
Resumen:	El caso de uso se inicia cuando el estudiante de la FEU necesita gestionar la participación de un estudiante en actividades de conocimiento (registrar participación o eliminar participación) y realiza

Anexos

	una búsqueda de las actividades de conocimiento sobre la que desea trabajar filtrándola por la fecha de realización de esta. Si las actividades seleccionadas tiene alguna realización en la fecha entrada el sistema muestra un listado con los participantes en dicha actividad (nombre, apellidos).Luego el estudiante de la FEU selecciona una de las opciones disponibles (registrar participación o eliminar participación).
Precondiciones:	El usuario ha sido identificado como estudiante FEU AC.
Referencias	R7.1,R7.2
Prioridad	Crítico
Flujo Normal de Eventos	
Sección ""	
Acción del Actor	Respuesta del Sistema
1-El estudiante FEU de AC accede al sistema.	2. El sistema muestra un formulario con las actividades disponibles y un campo de entrada para la fecha.
3. El estudiante FEU de AC selecciona la actividad a buscar e introduce la fecha por la que desea filtrar.	4. El sistema muestra un listado con todos los participantes en la actividad.
5. El estudiante de la FEU elige la acción a realizar.	6. <ul style="list-style-type: none"> ✓ Si elige registrar participación en AC debe ir a la Sección Registrar Participación en AC. ✓ Si elige eliminar participación en AC debe Sección Eliminar Participación en AC.
Sección "Registrar participación en AC"	
	1. El sistema muestra una interfaz.
2. El estudiante FEU AC selecciona el/los estudiantes.	3. El sistema obtiene el/los estudiantes y registra la participación terminando así el caso de uso.
	4. Se muestra un mensaje satisfactorio.
Flujos Alternos	
Acción del actor	Respuesta del sistema

Anexos

Sección "Eliminar participación en AC"	
	1. El sistema muestra la interfaz de eliminar participación en AC.
2. El estudiante FEU escoge lo que desea eliminar y presiona aceptar.	3. El sistema procede a obtener el estudiante y a la eliminación de la participación, y a la actualización de la base de datos, terminando así el caso de uso.
	4. Se muestra un mensaje satisfactorio.
Flujos alternos	
Acción del actor	Respuesta del sistema
Prototipo de interfaz	
Poscondiciones	Se actualiza el registro, eliminación de las actividades de conocimiento.

Anexo 21: Caso de uso Gestionar participación en AC.

Caso de Uso:	Gestionar resultados en AC
Actores:	Estudiante FEU AC
Resumen:	El caso de uso se inicia cuando el estudiante de la FEU necesita gestionar los resultados obtenidos por un estudiante en las actividades de conocimiento (registrar un nuevo resultado o modificar resultado existente) y realiza una búsqueda de las actividades de conocimiento sobre el que desea trabajar filtrándola por la fecha de realización de esta. Si la actividad seleccionada tiene alguna realización en la fecha entrada el sistema muestra un listado con los participantes en los estas actividades (nombre y apellidos y resultado obtenido por el estudiante). Luego el estudiante de la FEU selecciona una de las opciones disponibles (registrar resultado o modificar resultado).
Precondiciones:	El usuario ha sido identificado como estudiante FEU AC.
Referencias	R6.1, R6.2
Prioridad	Crítico
Flujo Normal de Eventos	
Sección ""	
Acción del Actor	Respuesta del Sistema
1-El estudiante FEU de AC accede a la	2. El sistema muestra un formulario con los

Anexos

interfaz que el corresponde.	nombres de las actividades disponibles y un campo de entrada para la fecha.
3. El estudiante FEU de AC selecciona la actividad a buscar e introduce la fecha por la que desea filtrar.	4. El sistema muestra un listado con todos los participantes en la actividad así como las opciones de registrar y modificar resultado.
5. El estudiante de la FEU elige la acción a realizar.	6. <ul style="list-style-type: none"> ✓ Si elige registrar resultado en AC debe ir a la Sección Registrar Resultado en AC. ✓ Si elige modificar resultado en AC debe Sección Modificar Resultado en AC.
Sección “Registrar resultado en AC”	
	1. El sistema muestra una interfaz.
2. El estudiante FEU AC introduce datos de la actividad a seleccionar.	3. El sistema obtiene la actividad y el estudiante y muestra un formulario con los resultados. <ul style="list-style-type: none"> ✓ Relevante ✓ Destacado ✓ Mención
4. El estudiante FEU AC selecciona los resultados.	5. El sistema registra el resultado en la actividad de conocimiento terminando así el caso de uso.
	6. Se muestra un mensaje satisfactorio.
Flujos Alternos	
Acción del actor	Respuesta del sistema
Sección “Modificar resultado en AC”	
	1. El sistema muestra la interfaz de modificar resultado en AC.
2. El estudiante FEU AC introduce los datos del usuario que desea modificar el resultado.	3. El sistema busca al usuario y muestra los datos.
4. El estudiante FEU realiza los cambios que le interesan y ordena actualizar dichas modificaciones.	5. El sistema verifican que no hayan datos incoherentes y modifica los datos.
	6. El sistema actualiza los cambios en la base

Anexos

	de datos terminando así el caso de uso.
	7. Se muestra un mensaje satisfactorio.
Flujos alternos	
Acción del actor	Respuesta del sistema
Prototipo de interfaz	
Poscondiciones	Se actualiza el registro y modificación de los resultados de las actividades de conocimiento.

Anexo 22: Caso de uso Gestionar resultados en AC.

Caso de Uso:	Realizar búsqueda de estudiante
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario introduce los datos para realizar la búsqueda del estudiante
Precondiciones:	El usuario ha sido validado.
Referencias	R11.
Prioridad	Crítico
Flujo Normal de Eventos	
Sección ""	
Acción del Actor	Respuesta del Sistema
1-El usuario entra los datos del estudiante a buscar.	1.1. El sistema busca los datos y los obtiene, mostrándolos así, y terminando el caso de uso. 1.2. Muestra un mensaje satisfactorio.
Flujos Alternos	
Acción del actor	Respuesta del sistema
Acción 1.	1.1. En caso de no existir estos datos emitir un mensaje de aviso.
Prototipo de interfaz	
Poscondiciones	-

Anexo 23: Caso de uso Realizar búsqueda de estudiante.

Caso de Uso:	Gestionar reunión
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario necesita gestionar reuniones (registrar una nueva reunión).
Precondiciones:	El usuario ha sido validado.

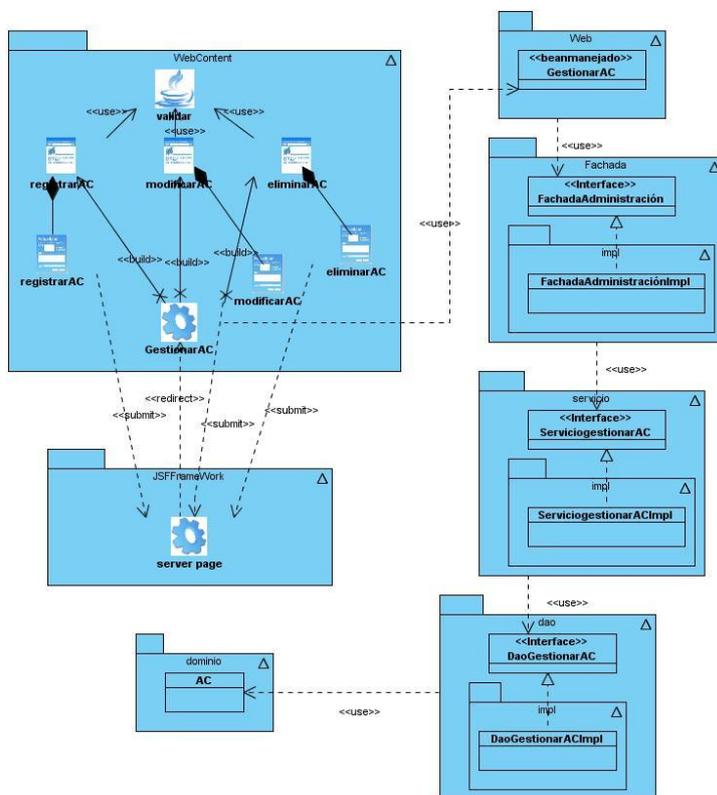
Anexos

Referencias	R6.1,R6.2
Prioridad	Crítico
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1-El usuario accede a la interfaz que el corresponde.	2. El sistema muestra la opción de seleccionar el orden del día.
3. El jefe de brigada elige el orden del día.	<p>4. El sistema muestra la opción de tomar la asistencia de los estudiantes. De ahí el jefe de brigada tiene tres opciones.</p> <ul style="list-style-type: none"> ✓ Si elige orden del día debe ir a la Sección Orden del día. ✓ Si elige acuerdos debe ir a la Sección Acuerdos. ✓ Si elige temas de la brigada debe ir a la Sección Temas de la brigada.
Sección “Orden del Día”	
1. El jefe de brigada selecciona el estudiante para insertar su opinión.	2. El sistema registra la opinión del estudiante.
	3. El sistema muestra una vista previa donde te muestra todos los estudiantes que han opinado respecto a un tema determinado, terminando así el caso de uso.
	4. Se muestra un mensaje satisfactorio.
Flujos Alternos	
Acción del actor	Respuesta del sistema
Sección “Acuerdos”	
1. El jefe de brigada selecciona adicionar nuevo acuerdo.	2. El sistema brinda la opción de adicionar un nuevo acuerdo y seleccionar además el responsable.
3. El jefe de brigada después de haber	4. El sistema muestra para eliminar un nuevo

Anexos

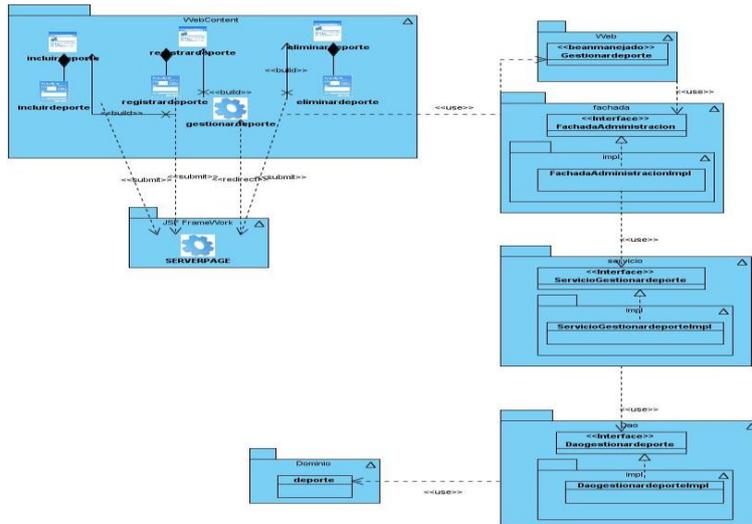
adicionado el nuevo acuerdo lo puede eliminar.	acuerdo terminando así el caso de uso.
	4. Se muestra un mensaje satisfactorio
Sección "Temas de la brigada"	
1. El jefe de brigada selecciona el estudiante para insertar su tema.	2. El sistema registra el tema del estudiante.
	3. El sistema muestra una vista previa donde te muestra todos los estudiantes que han dado un tema, terminando así el caso de uso.
	4. Se muestra un mensaje satisfactorio.
Flujos alternos	
Acción del actor	Respuesta del sistema
Prototipo de interfaz	
Poscondiciones	Se actualizan los acuerdos, temas de la brigada y orden del día.

Anexo 24: Caso de uso Gestionar reunión.

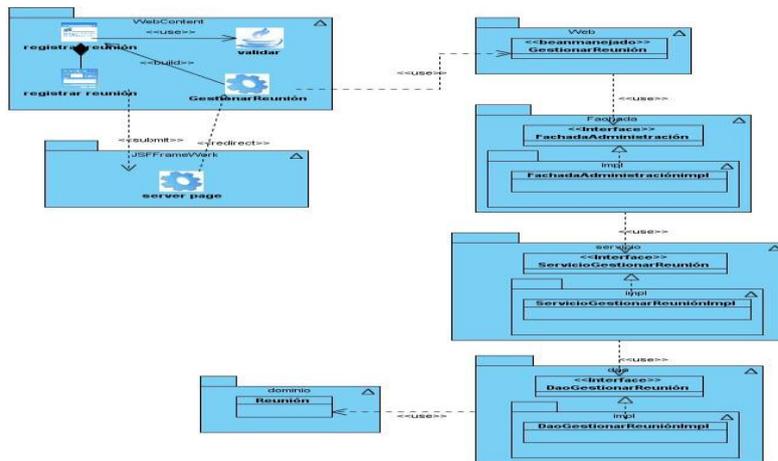


Anexo 25: Diagrama del diseño Gestionar AC.

Anexos



Anexo 26: Diagrama del diseño Gestionar deporte.



Anexo 27: Diagrama del diseño Gestionar reunión.

Entrada	Resultados	Condiciones
El administrador selecciona la opción Insertar estudiante.	El sistema muestra una interfaz con los datos requeridos para insertar un nuevo estudiante.	
El administrador introduce	El sistema envía un	No se registren valores

Anexos

<p>los datos del estudiante dejando algún campo vacío.</p>	<p>mensaje informando que debe llenar todos los campos.</p> <p>Ejemplo: “El campo nombre completo está vacío o tiene caracteres incorrectos”.</p>	<p>vacíos en los campos de entrada para insertar un nuevo estudiante.</p>
<p>El administrador introduce los datos siguientes:</p> <p>Nombre :Yadira</p> <p>Segundo nombre:</p> <p>Primer apellido: Cervantes</p> <p>Segundo Apellido: García</p> <p>Carnet:87011701234</p> <p>Cargo: Jefe de actividades</p>	<p>El sistema inserta el nuevo estudiante, confirmando la operación. Ejemplo: “Insertado correctamente”.</p>	<p>No exista el estudiante en la aplicación.</p>
<p>El administrador introduce los datos siguientes:</p> <p>Nombre :Yadira</p> <p>Segundo nombre:</p> <p>Primer apellido: Cervantes</p> <p>Segundo Apellido: García</p> <p>Carnet:87011701234</p> <p>Cargo: Jefe de actividades</p>	<p>El sistema informa que el estudiante que se pretende insertar ya está registrado, enviando un mensaje. Ejemplo: “ El estudiante ya se encuentra registrado”</p>	<p>Existe el estudiante en la aplicación.</p>
<p>El administrador selecciona la opción de Modificar estudiante.</p>	<p>El sistema muestra una interfaz con un listado de estudiante.</p>	

Anexos

<p>El administrador selecciona el estudiante a modificar y modifica alguno de los datos siguientes:</p> <p>Nombre :Yadira</p> <p>Segundo nombre:</p> <p>Primer apellido: Cervantes</p> <p>Segundo Apellido: García</p> <p>Carnet:87011701234</p> <p>Cargo: Jefe de actividades</p> <p>Y presiona el botón Modificar</p>	<p>El sistema envía un mensaje de que fue modificado correctamente.</p> <p>Ejemplo: “Modificado correctamente.”</p>	<p>Los datos estén correctos.</p>
<p>El administrador selecciona la opción de Eliminar estudiante.</p>	<p>El sistema muestra una interfaz con una lista de estudiante.</p>	
<p>El administrador selecciona el estudiante a eliminar y presiona el botón Eliminar.</p>	<p>El sistema envía una pregunta para garantizar que no se realice una operación errada. ¿El estudiante está a punto de ser eliminado?.Desea continuar?(Si, No)</p>	<p>Sea presionado el botón Eliminar.</p>
<p>El administrador presiona el botón Si de la pregunta lanzada posteriormente.</p>	<p>El sistema envía mensaje que el estudiante fue eliminado correctamente.</p> <p>Ejemplo:” Eliminado correctamente.”</p>	<p>Sea presionado el botón Si.</p>

Anexo 28: Caso de prueba Gestionar estudiante.

Anexos

Entrada	Resultados	Condiciones
El administrador selecciona la opción Insertar Actividad.	El sistema muestra una interfaz con los datos requeridos para insertar una nueva actividad.	
El administrador introduce los datos de la actividad dejando algún campo vacío.	El sistema envía un mensaje informando que debe llenar todos los campos. Ejemplo: "Existen campos vacíos que son obligatorios".	No se registren valores vacíos en los campos de entrada para insertar una nueva actividad.
El administrador introduce los datos siguientes: Nombre: Reunión de brigada Seleccione Tipo de actividad :Organizativa	El sistema inserta la nueva actividad, confirmando la operación. Ejemplo: "Insertada correctamente".	No exista la actividad en la aplicación.
El administrador introduce los datos siguientes: Nombre: Reunión de brigada Seleccione Tipo de actividad :Organizativa	El sistema informa que la actividad que se pretende insertar ya está registrado, enviando un mensaje. Ejemplo: " La actividad ya se encuentra registrado"	Existe la actividad en la aplicación.
El administrador selecciona la opción de Modificar actividad.	El sistema muestra una interfaz.	

Anexos

<p>El administrador selecciona la actividad a modificar y modifica el campo siguiente:</p> <p>Nombre :Reunión de brigada</p> <p>Y presiona el botón Modificar</p>	<p>El sistema envía un mensaje de que fue modificado correctamente.</p> <p>Ejemplo: “La acción se ha efectuado satisfactoriamente.”</p>	<p>Los datos estén correctos.</p>
<p>El administrador selecciona la opción de Eliminar actividad.</p>		
<p>El administrador selecciona la actividad a eliminar y presiona el botón Eliminar.</p>	<p>El sistema envía una pregunta para garantizar que no se realice una operación errada. ¿La actividad está a punto de ser eliminado? ¿Desea continuar?(Si, No)</p>	<p>Sea presionado el botón Eliminar.</p>
<p>El administrador presiona el botón Si de la pregunta lanzada posteriormente.</p>	<p>El sistema envía mensaje que la actividad fue eliminada correctamente.</p> <p>Ejemplo:” Eliminado correctamente.”</p>	<p>Sea presionado el botón Si.</p>

Anexo 29: Caso de prueba Gestionar actividad.

Entrada	Resultados	Condiciones
<p>El administrador selecciona la opción Insertar Usuario.</p>	<p>El sistema muestra una interfaz con los datos requeridos para insertar un nuevo usuario.</p>	
<p>El administrador introduce</p>	<p>El sistema envía un</p>	<p>No se registren valores</p>

Anexos

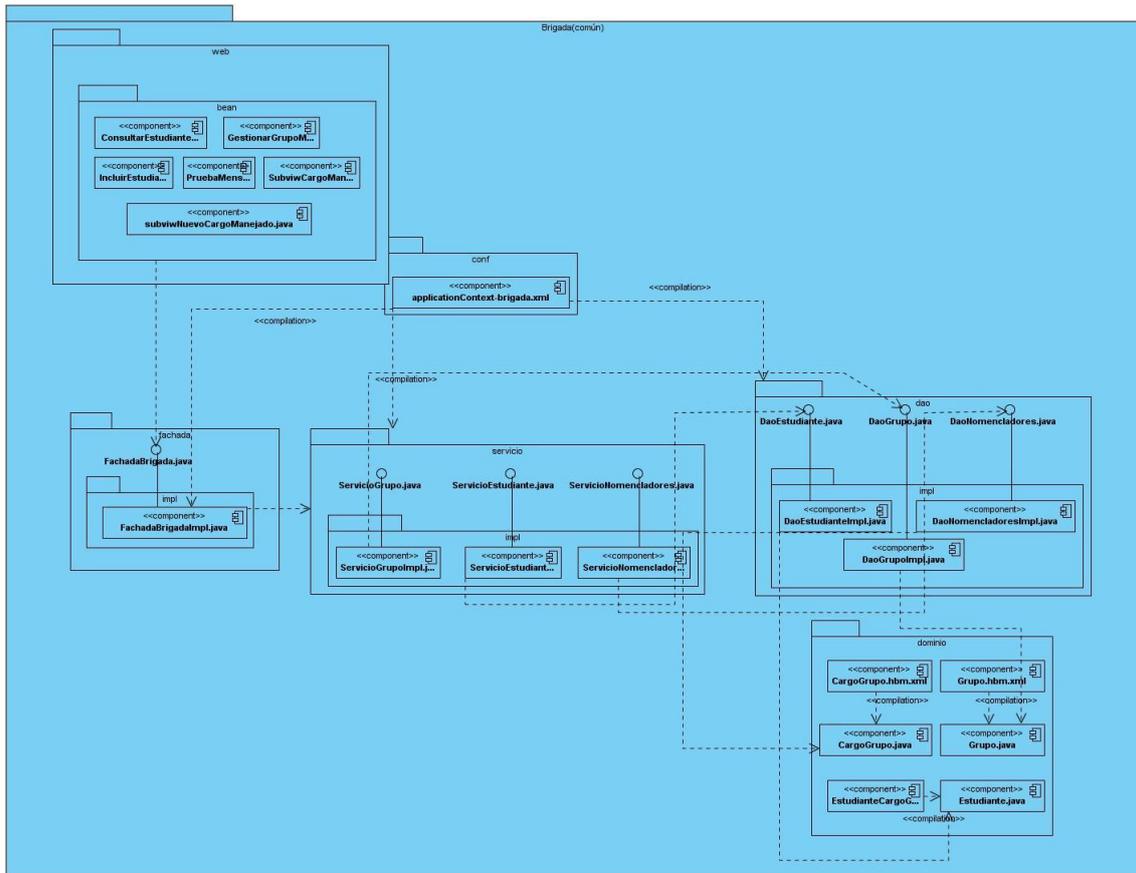
<p>los datos del usuario dejando algún campo vacío.</p>	<p>mensaje informando que debe llenar todos los campos.</p> <p>Ejemplo: “El campo nombre completo está vacío o tiene caracteres incorrectos”.</p>	<p>vacíos en los campos de entrada para insertar un nuevo usuario.</p>
<p>El administrador introduce los datos siguientes:</p> <p>Nombre :Yadira</p> <p>Segundo nombre:</p> <p>Primer apellido: Cervantes</p> <p>Segundo Apellido: García</p> <p>Carnet:87011701234</p> <p>Cargo: Jefe de actividades</p>	<p>El sistema inserta el nuevo usuario, confirmando la operación. Ejemplo: “Insertado correctamente”.</p>	<p>No exista el usuario en la aplicación.</p>
<p>El administrador introduce los datos siguientes:</p> <p>Nombre :Yadira</p> <p>Segundo nombre:</p> <p>Primer apellido: Cervantes</p> <p>Segundo Apellido: García</p> <p>Carnet:87011701234</p> <p>Cargo: Jefe de actividades</p>	<p>El sistema informa que el usuario que se pretende insertar ya está registrado, enviando un mensaje. Ejemplo: “ El usuario ya se encuentra registrado”</p>	<p>Existe el usuario en la aplicación.</p>
<p>El administrador selecciona la opción de Modificar Usuario.</p>	<p>El sistema muestra una interfaz con un listado de usuarios.</p>	

Anexos

<p>El administrador selecciona el usuario a modificar y modifica alguno de los datos siguientes:</p> <p>Nombre :Yadira</p> <p>Segundo nombre:</p> <p>Primer apellido: Cervantes</p> <p>Segundo Apellido: García</p> <p>Carnet:87011701234</p> <p>Cargo: Jefe de actividades</p> <p>Y presiona el botón Modificar</p>	<p>El sistema envía un mensaje de que fue modificado correctamente.</p> <p>Ejemplo: “Modificado correctamente.”</p>	<p>Los datos estén correctos.</p>
<p>El administrador selecciona la opción de Eliminar Usuario.</p>	<p>El sistema muestra una interfaz con una lista de usuarios.</p>	
<p>El administrador selecciona el usuario a eliminar y presiona el botón Eliminar.</p>	<p>El sistema envía una pregunta para garantizar que no se realice una operación errada. ¿El usuario está a punto de ser eliminado? ¿Desea continuar?(Si, No)</p>	<p>Sea presionado el botón Eliminar.</p>
<p>El administrador presiona el botón Si de la pregunta lanzada posteriormente.</p>	<p>El sistema envía mensaje que el usuario fue eliminado correctamente.</p> <p>Ejemplo:” Eliminado correctamente.”</p>	<p>Sea presionado el botón Si.</p>

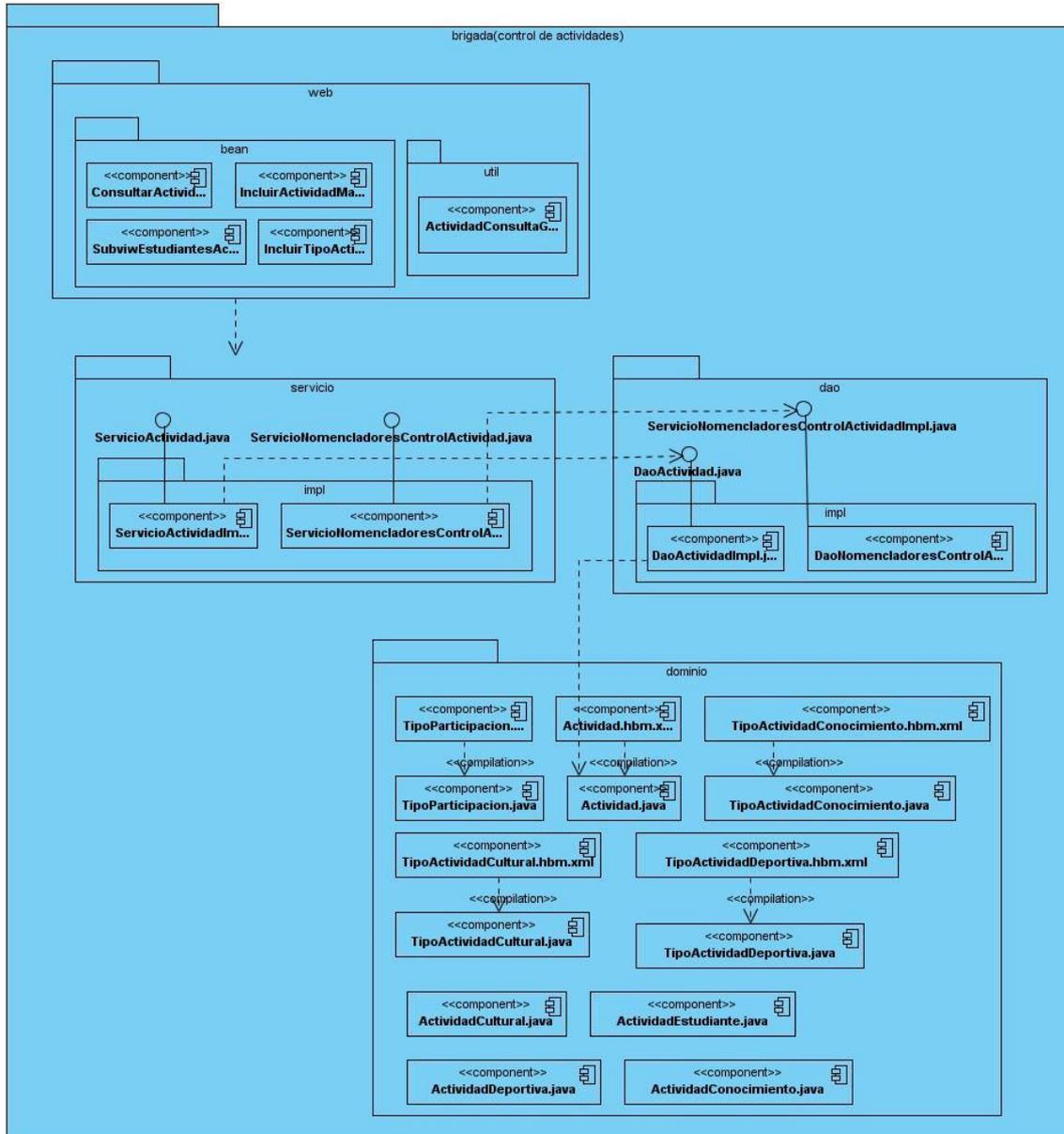
Anexo 30: Caso de prueba Gestionar usuario.

Anexos



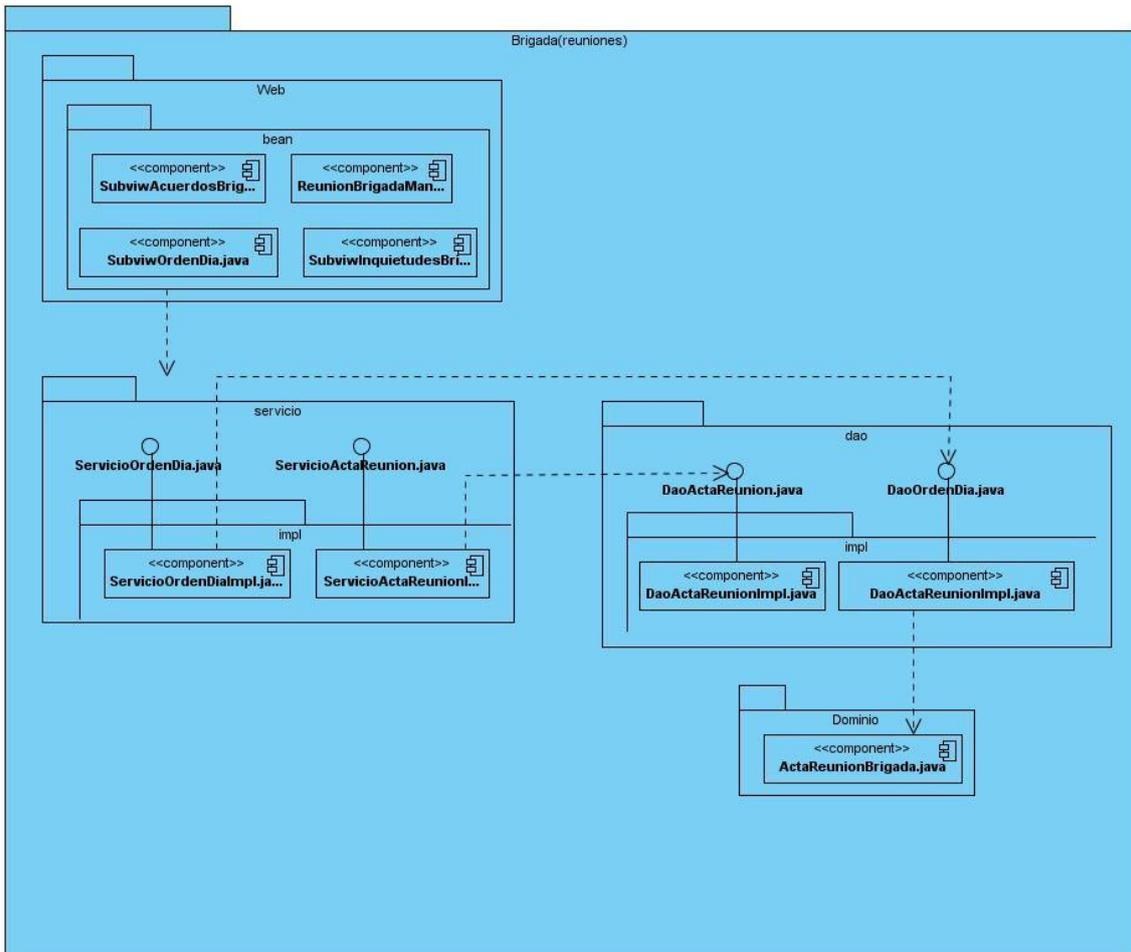
Anexo 31: Diagrama de componentes paquete Brigada (común).

Anexos



Anexo 32: Diagrama de componentes paquete Brigada (control de actividades).

Anexos



Anexo 33: Diagrama de componentes paquete Brigada (reuniones).