

*Universidad de Sancti Spíritus “José Martí”
Facultad de Ingeniería
Carrera Ingeniería Informática*



*Trabajo de diploma para optar por el título de Ingeniería
Informática.*

*Título: “Aplicación Web para contribuir al proceso de enseñanza-
aprendizaje en la carrera de Cultura Física de la Universidad de
Sancti Spíritus”*

Autor: Dayana Cancio Espinaco.

Tutora: Ing. Irina Machado Mutis.

Consultante: Ing. Yainel García Alfonso.

Sancti Spíritus, Cuba

2013

Dedicatoria

*A mi familia, en especial a mi mamá, mi hermano, mi padrastro y mi novio
por ser las personas que han estado siempre a mi lado dándome todo su
apoyo y cariño.*

Agradecimientos

- ✓ *A mi familia por todo el amor que me brindan y por estar siempre a mi lado.*
- ✓ *A mi mamá por ser un ejemplo para mí.*
- ✓ *A mi novio por el amor tan lindo que me regala todos los días.*
- ✓ *A mi amiga Diana por estar siempre dispuesta a ayudarme y su amistad tan sincera y especial.*
- ✓ *A Maruchi por ser una de las mejores personas que conocí en la universidad.*
- ✓ *A mi tutora Irina por su apoyo y dedicación.*
- ✓ *A Nane por la ayuda tan importante que me brindó.*
- ✓ *A mis compañeros de aula en especial a Dunelvis y Alexander.*
- ✓ *A todos mis profesores por los conocimientos y experiencia que me aportaron.*

A todos, muchas Gracias

La Carrera de Cultura Física en la Universidad de Sancti Spíritus actualmente presenta problemas en cuanto al proceso de enseñanza-aprendizaje en la asignatura Metodología de la Investigación, es por ello que el objetivo de esta investigación es desarrollar una Aplicación Web para contribuir al proceso de enseñanza-aprendizaje de la asignatura Metodología de la Investigación para los estudiantes de Cultura Física en la Uniss.

Las etapas llevadas a cabo en la elaboración del software fueron las planteadas por el Proceso Unificado de Desarrollo (RUP), utilizándose como lenguaje el “Lenguaje Unificado de Modelado” (UML) y como herramienta para el análisis y diseño: el Rational Rose Enterprise Edition. El sistema se implementó haciendo uso del framework de PHP Symfony 2.2.0, el cual incluye facilidades de desarrollo para aplicaciones utilizando el patrón Modelo Vista Controlador. Se empleó PHP 5.3.5 como lenguaje de programación, lo que proporcionó un código eficiente; para la persistencia de los datos se utilizó MySQL 5.5.8 por considerarse un Sistema Gestor de Bases de Datos muy poderoso y como servidor web se escogió Apache 2.2.17, por la seguridad y estabilidad que presenta.

Physical Education career from Sancti Spíritus University has currently presented difficulties in the teaching learning process related to the subject of Methodology of Investigation that is why this paper aims at designing a web site to make the content accessible for the professors and students.

According to the Unified Developing Process (UDP) different phases were planned for the web site design and "The Unified Model Language" (UML) was the model language used. For the analysis and design of the web the Rational Rose Enterprise Edition was implemented. The system was carried out using Symphony 2.2.0, PHP framework, which includes development facilities for applications using as a pattern the Sight Model Controller (SMC). PHP 5.3.5 was also used as the program language and MySQL 5.5.8 was stated to be considered as a base powerful data manager system and Apache 2.2.17 was selected as the web server because of the security and stability it provides.

Introducción	1
Capítulo I: Fundamentación teórica y metodológica que sustentan el desarrollo de una Aplicación Web que contribuya al proceso de enseñanza-aprendizaje de la asignatura metodología de la investigación.....	6
Introducción.....	6
1.1 El proceso de enseñanza-aprendizaje.....	6
1.1.1 Conceptualización y análisis de los términos que componen dicho proceso. 6	
1.2 Universidad de Sancti Spíritus “José Martí Pérez”	7
1.3 Facultad de Cultura Física	8
1.4 Metodologías utilizadas	8
1.4.1 Metodologías Tradicionales	8
1.4.2 Metodologías Ágiles.....	10
1.5 Arquitecturas para el desarrollo de sitios web	11
1.6 Lenguajes de programación utilizados para el desarrollo de sitios web.....	15
1.7 Frameworks	16
1.8 Servidor Web Apache.....	17
1.9 Sistemas gestores de Bases de Datos	18
1.10 Herramientas de desarrollo	20
Conclusiones.....	21
Capítulo II: Descripción de la aplicación propuesta para contribuir al proceso de enseñanza-aprendizaje de la asignatura metodología de la investigación para los estudiantes de Cultura Física de la Uniss.	21
Introducción.....	21
2.1 Modelo del negocio	21
2.1.1 Identificación de los procesos del negocio	21
2.1.2 Reglas del negocio	22
2.1.3 Modelo de casos de uso del negocio	22
2.1.4 Actores del negocio	22
2.1.5 Diagrama de casos de uso del negocio	22
2.1.6 Trabajadores del negocio.....	23
2.1.7 Descripción de los Casos de Uso del negocio.	23
2.1.8 Diagrama de actividades.....	24
2.1.9 Modelo de objetos del negocio.....	24
2.2 Requerimientos	25

2.2.1 Requerimientos funcionales	25
2.2.2 Requerimientos no funcionales	27
2.3 Descripción del sistema propuesto	30
2.3.1 Modelo de casos de uso del sistema	30
2.3.2 Actores del sistema.....	30
2.3.3 Casos de uso del sistema	30
2.3.4 Diagrama de casos de uso del sistema.....	31
2.4 Conclusiones.....	35
Capítulo III: Implementación de una Aplicación Web que contribuya al proceso de enseñanza-aprendizaje de la asignatura metodología de la investigación para los estudiantes de Cultura Física de la Uniss.	36
Introducción.....	36
3.1 Diagrama de Clases Web.....	36
3.2 Diseño de la base de datos	36
3.2.1 Diagrama de clases persistentes	36
3.2.2 Modelo de datos	37
3.3 Principios de diseño	38
3.3.1 Diseño de la interfaz del sistema	38
3.3.2 Tratamiento de errores.....	38
3.3.3 Concepción general de la ayuda.....	38
3.3.4 Seguridad	38
3.4 Modelo de implementación.....	39
3.4.1 Diagrama de despliegue.	39
3.4.2 Diagrama de componentes	39
Conclusiones.....	41
Conclusiones	42
Recomendaciones	43
Bibliografía.....	44
Anexos	47

Índice de Figuras

Figura 1: Esquema de funcionamiento de la arquitectura cliente/servidor.....	12
Figura 2: Arquitectura en tres capas.	13
Figura 3: Esquema Modelo-Vista-Controlador	14
Figura 4: Diagrama de casos de uso del negocio.	23
Figura 5: Diagrama de casos de uso del sistema por paquetes.	33
Figura 6: Diagrama de casos de uso del sistema: Paquete seguridad.	33
Figura 7: Diagrama de casos de uso del sistema: Paquete gestión.	34
Figura 8: Diagrama de casos de uso del sistema: Paquete reportes.....	35
Figura 9: Diagrama de casos de uso del sistema: Paquete ejercicios.....	34
Figura 10: Diagrama de clases persistentes.	37
Figura 11: Diagrama de Despliegue.....	39
Figura 12: Diagrama de Componentes.	40

Índice de Tablas

Tabla 1: Actor del negocio.	22
Tabla 2: Trabajador del negocio.	23
Tabla 3: Descripción de los casos de uso del negocio.	24
Tabla 4: Actores del sistema.	30
Tabla 5: Casos de uso del sistema por paquetes.	32
Tabla 6: Descripción de los componentes.	40

Introducción

La educación en Cuba comenzó siendo una tarea que se realizaba por el sector privado, el primer maestro de Cuba fue el sacerdote católico Padre Miguel Velázquez, nombrado maestro en la Catedral de Santiago en 1544. A partir de entonces, la Iglesia Católica echó sobre sus hombros la responsabilidad de impartir la poca instrucción que se ofreció durante casi dos siglos a los niños de edad escolar que vivían en la Isla. La educación comenzó a tomar algún impulso a partir de la fundación de la Sociedad Económica de Amigos del País, que tuvo lugar en La Habana en 1793. También contribuyó a incrementar la educación la llegada a la Isla de algunas Órdenes Religiosas dedicadas a la enseñanza. Ya en 1840, el Gobierno Colonial empezó a mostrar algún interés en la educación, por lo que se establecieron Escuelas Normales para formar maestros. A consecuencia de estas medidas, y del constante estímulo de las Sociedades Económicas, el número de las escuelas públicas continuó creciendo y atrayendo un mayor número de estudiantes.

Los eventos de mayor significación para la enseñanza en Cuba durante el período colonial fueron: la Fundación de la Real y Pontificia Universidad de San Jerónimo (la Universidad de La Habana), en 1728; la Creación del Seminario de San Carlos y San Ambrosio, en 1769; donde realizaron su gran labor educativa los Padres José Agustín Caballero y Félix Varela. El primero de ellos concibió la idea de llevar a cabo una reforma educativa de vastas proporciones, pero al Padre Varela es a quien le corresponde el mérito de haber transformado y modernizado la educación, no sólo en su cátedra de Filosofía del Seminario, sino en todos los niveles de la enseñanza en Cuba. Además, realizó una labor educativa extraordinaria, y formó a toda una generación de jóvenes, de donde salieron los más grandes líderes intelectuales del siglo XIX cubano.

Poco a poco en la Isla la educación fue cobrando auge pero todavía la situación era bastante deficiente durante la época previa a la Revolución ya que el sector educativo nunca gozó de amplios recursos. Con el triunfo de la Revolución Cubana se dio un impulso fundamental al sector educacional. En 1961 se organizó una campaña nacional de alfabetización. Esto hizo que en la práctica se erradicara el analfabetismo en Cuba. De igual manera, la obligatoriedad de la educación se extendió al sexto grado. En la actualidad, la enseñanza en Cuba es gratuita a todos los niveles desde los centros escolares primarios hasta las universidades.

Resulta innegable el incremento cada vez mayor de las tecnologías de la información (TI) en las disímiles esferas de la sociedad a nivel mundial. El desarrollo impetuoso de la ciencia y la tecnología ha llevado a la sociedad a entrar al nuevo milenio, inmerso en lo que se ha dado en llamar “era de la información”.

Las nuevas Tecnologías de la Información y de la Comunicación (NTIC) han evolucionado espectacularmente en los últimos años. Esta nueva fase de desarrollo va a tener gran impacto en la organización de la enseñanza y el proceso de aprendizaje. La acomodación del entorno educativo a este nuevo potencial y la adecuada utilización didáctica del mismo supone un reto sin precedentes.

La aplicación de los medios de enseñanza permite, entre otras cosas: mejorar la calidad de las clases y adquirir los conocimientos de una manera más fácil. Se demostró en muchas ocasiones por la ciencia pedagógica, que la computación favorece la asimilación de los contenidos. (García Suárez & López Delgado, 2007)

La Universidad de Sancti Spiritus (Uniss) comienza como Filial Universitaria e inicia sus actividades en el curso 1976-1977 en las aulas del Centro Escolar Serafín Sánchez solamente con dos grupos de alumnos, uno de la especialidad de agronomía y otro de la especialidad de control económico. En el curso 1980-1981 se le entrega el actual edificio y posteriormente en el curso 94-95 surge la Sede Universitaria que por el desarrollo alcanzado es nombrada Universidad (Uniss) en el año 2010.

Este es un centro donde las nuevas Tecnologías de la Información y las Comunicaciones también han tomado gran importancia en lo referido al apoyo del proceso de enseñanza-aprendizaje siendo la facultad de cultura física que comienza a formar parte de la Sede en el curso 94-95 una de las facultades que se plantea aprovechar estas tecnología para mejorar dicho proceso. En sus aulas se forman los Licenciados en Cultura Física y Deportes del territorio que tanto han aportado al desarrollo deportivo y los éxitos alcanzados por la provincia en eventos Nacionales e Internacionales.

Para la Uniss constituye un reto insertarse en el creciente auge que tienen hoy en día las tecnologías de la información y las comunicaciones (TIC) y su vínculo con la pedagogía, teniendo en cuenta las facilidades que puede brindar una herramienta informática y la necesidad que existe de mejorar el proceso de enseñanza-aprendizaje específicamente en la asignatura Metodología de la Investigación.

Los estudiantes de cultura física presentan dificultades en la asimilación de los contenidos de la asignatura Metodología de la Investigación debido a diferentes factores que les impiden la constancia en el estudio. Estos estudiantes en su mayoría son atletas de alto rendimiento, o sea, son atletas que tienen que dedicar la mayor parte del tiempo al entrenamiento físico, por otra parte reciben clases en las aulas una vez a la semana lo que trae consigo poca ejercitación de los contenidos. No tienen acceso a Internet y solo cuentan con la bibliografía básica que se utiliza en la asignatura. Además no cuentan con ejercicios complementarios que les permitan consolidar los conocimientos adquiridos.

Se plantea como **Problema Científico**:

¿Cómo contribuir al proceso de enseñanza-aprendizaje en la asignatura Metodología de la Investigación de la carrera de Cultura Física en la Uniss?

El **objetivo general** es desarrollar una Aplicación Web que contribuya al proceso de enseñanza-aprendizaje de la asignatura Metodología de la Investigación para los estudiantes de Cultura Física en la Uniss.

Para dar cumplimiento al objetivo general se plantean las siguientes **preguntas científicas**:

1. ¿Qué fundamentos teóricos y metodológicos sustentan el desarrollo de una Aplicación Web que contribuya al proceso de enseñanza-aprendizaje de la asignatura Metodología de la Investigación?
2. ¿Cómo diseñar una Aplicación Web que contribuya al proceso de enseñanza-aprendizaje de la asignatura Metodología de la Investigación para los estudiantes de Cultura Física de la Uniss?
3. ¿Cómo implementar una Aplicación Web que contribuya al proceso de enseñanza-aprendizaje de la asignatura Metodología de la Investigación para los estudiantes de Cultura Física de la Uniss?

Para dar respuesta a las preguntas científicas se plantean las siguientes **tareas de investigación**:

1. Análisis de los fundamentos teóricos y metodológicos que sustentan el desarrollo de una Aplicación Web que contribuya al proceso de enseñanza-aprendizaje de la asignatura Metodología de la Investigación.

2. Diseño de una Aplicación Web que contribuya al proceso de enseñanza-aprendizaje de la asignatura Metodología de la Investigación para los estudiantes de Cultura Física de la Uniss.
3. Implementación de una Aplicación Web que contribuya al proceso de enseñanza-aprendizaje de la asignatura Metodología de la Investigación para los estudiantes de Cultura Física de la Uniss.

Estructura del documento:

El documento consta de una introducción, tres capítulos, conclusiones recomendaciones, bibliografías y anexos

Capítulo I: Fundamentación teórica y metodológica que sustentan el desarrollo de una Aplicación Web que contribuya al proceso de enseñanza-aprendizaje de la asignatura Metodología de la Investigación. En este capítulo se plasma toda la fundamentación teórica del trabajo. Además se hace un estudio de las diferentes metodologías, como las metodologías tradicionales y metodologías ágiles, arquitecturas, lenguajes de programación, sistemas gestores de bases de datos y framework existentes, describiendo sus características y por qué se seleccionaron.

Capítulo II: Diseño de una Aplicación Web que contribuya al proceso de enseñanza-aprendizaje de la asignatura Metodología de la Investigación para los estudiantes de Cultura Física de la Uniss. En este capítulo se describen los elementos que componen el proceso estudiado, utilizando la metodología RUP. Se determinan los actores y trabajadores del negocio, los casos de uso, requerimientos funcionales y no funcionales, los casos de uso del sistema así como sus correspondientes diagramas y descripciones.

Capítulo III: Implementación de una Aplicación Web que contribuya al proceso de enseñanza-aprendizaje de la asignatura Metodología de la Investigación para los estudiantes de Cultura Física de la Uniss. En este capítulo se plantean los detalles relacionados con la implementación del sistema que se propone a través de los diagramas de clases del diseño, el diagrama de las clases persistentes, el modelo de datos y la descripción de las tablas. Se describen los principios de diseño usados en la interfaz de la aplicación, la concepción general de la ayuda, cómo se maneja la seguridad y el tratamiento de las excepciones.

Capítulo I: Fundamentación teórica y metodológica que sustentan el desarrollo de una Aplicación Web que contribuya al proceso de enseñanza-aprendizaje de la asignatura metodología de la investigación.

Introducción

El presente capítulo incluye la base de la fundamentación teórica del tema a desarrollar. Describe los principales conceptos asociados al problema así como la misión y visión de la Universidad de Sancti Spíritus y la facultad de Cultura Física. Se realiza además un estudio de las distintas metodologías, arquitecturas y lenguajes de programación que posteriormente se utilizarán en la implementación del software.

1.1 El proceso de enseñanza-aprendizaje

Enseñanza y aprendizaje forman parte de un único proceso que tiene como fin la formación del estudiante. (Hernández, 1989)

1.1.1 Conceptualización y análisis de los términos que componen dicho proceso.

En el proceso de enseñanza-aprendizaje nos resulta difícil ver estos dos términos por separados debido a la dependencia que existe entre ambos ya que uno no tiene sentido sin el otro, pero esto no significa que no podamos definir dichos términos por separados. La enseñanza es una actividad realizada conjuntamente mediante la interacción de 3 elementos: un profesor o docente, uno o varios alumnos o discentes y el objeto desconocimiento. Por su parte el aprendizaje es el proceso a través del cual se adquieren nuevas habilidades, destrezas, conocimientos, conductas o valores como resultado del estudio. Podemos afirmar que existe un sujeto que conoce (el que puede enseñar), y otro que desconoce (el que puede aprender). El que puede enseñar, quiere enseñar y sabe enseñar (el profesor); el que puede aprender quiere y sabe aprender (el alumno). Ha de existir pues una disposición por parte de alumno y profesor. Aparte de estos agentes, están los contenidos, esto es, lo que se quiere enseñar o aprender (elementos curriculares) y los procedimientos o instrumentos para enseñarlos o aprenderlos (medios). Cuando se enseña algo es para conseguir alguna meta (objetivos). Por otro lado, el acto de enseñar y aprender acontece en un marco determinado por ciertas condiciones físicas, sociales y culturales (contexto). (Hernández, 1989)

De acuerdo con lo expuesto, podemos considerar que el **proceso de enseñar** es el acto mediante el cual el profesor muestra o suscita contenidos educativos (conocimientos, hábitos, habilidades) a un alumno, a través de unos medios, en función de unos objetivos y dentro de un contexto. (Hernández, 1989)

El **proceso de aprender** es el proceso complementario de enseñar. Aprender es el acto por el cual un alumno intenta captar y elaborar los contenidos expuestos por el profesor, o por cualquier otra fuente de información. Él lo alcanza a través de unos medios (técnicas de estudio o de trabajo intelectual). Este proceso de aprendizaje es realizado en función de unos objetivos, que pueden o no identificarse con los del profesor y se lleva a cabo dentro de un determinado contexto. (Hernández, 1989)

1.2 Universidad de Sancti Spíritus “José Martí Pérez”

Los principios de universalización en Sancti-Spíritus los comenzó a materializar la Universidad Central de las Villas (UCLV) en el año 1969, manifestándose a través de trabajos experimentales e investigaciones en arroz en el área del Sur del Jíbaro y de la vinculación, en los planes de desarrollo de la ganadería, de los alumnos de quinto año de medicina veterinaria. (Portal de la Universidad de Sancti Spíritus “José Martí Pérez”)

Una vez fundada la Filial Universitaria de Sancti-Spíritus, en el curso 1976-1977, comienza sus actividades en los locales del Centro Escolar Serafín Sánchez. El curso 1977 -1978 se inició con un aumento en el número de carreras y se siguieron utilizando diversas aulas de diferentes escuelas del territorio para impartir las clases y ya en el curso 1980-1981 comienza a radicar en el actual edificio Docente I. Debido a los méritos alcanzados, a los requerimientos de la sociedad espirituana y con el compromiso de lograr una mayor preparación de sus futuros profesionales, se convierte de Filial en Centro Universitario y posteriormente en Universidad (UNISS) en el año 2010.

La Universidad de Sancti Spíritus “José Martí Pérez”, como Universidad Territorial Universalizada Sustentable, tiene como **misión** la formación de manera continua en la sociedad, profesionales integrales, revolucionarios, comprometidos con la solución trascendente de las necesidades del territorio y la nación en el desarrollo y producción científica, tecnológica, artística; fieles a las tradiciones e historia y con la voluntad de innovación; en el logro de una calidad para todos, que se geste desde la formación contextualizada y una gestión compartida, en correspondencia con la política del PCC,

programas de la Revolución y los fundamentos del ideario martiano, con un claustro integralmente preparado.

Su **visión** es consolidar la comunidad universitaria como fiel exponente de los principios revolucionarios, los valores patrios y el socialismo. El claustro está altamente comprometido con la Revolución, en correspondencia con los valores de la Organización. Se alcanzan niveles cualitativamente superiores en la formación integral de los estudiantes, en correspondencia con las demandas de fuerza laboral calificada universitaria en la provincia. Se aumenta la calidad de las carreras en todos los tipos de curso, comprobable a través del sistema de evaluación y acreditación de carreras universitarias que posee el Ministerio de Educación Superior.

1.3 Facultad de Cultura Física

La Facultad de la Cultura Física y el Deporte (UCCFD) es el centro de nivel superior adscrito al Instituto Nacional de Deportes, Educación Física y Recreación (INDER) encargado de dirigir la formación de profesionales y especialistas de alto nivel científico técnico en la rama de la Cultura Física y el Deporte. Tiene como misión formar profesionales competentes y comprometidos con el proyecto socialista cubano, universalizar los estudios de la Cultura Física y el Deporte en todo el territorio promoviendo la formación profesional y el postgrado, como un proceso continuo, la investigación científica y una cultura general integral, para lograr profesionales altamente capacitados que sepan poner bien en alto el Deporte y la Cultura Física cubana en cualquier parte del mundo, para ello cuenta con un grupo de profesionales altamente capacitados y comprometidos con el proceso revolucionario. Su visión es ser una Universidad acreditada de excelencia en todos sus procesos sustantivos. Ser líder en la aplicación consecuente de la política científica y tecnológica que facilite la satisfacción de la demanda de Ciencia e Innovación Tecnológica del INDER. Ser centro de referencia de la Cultura Física y el Deporte en el ámbito Provincial y Nacional.

1.4 Metodologías utilizadas

1.4.1 Metodologías Tradicionales

✓ Proceso Unificado Racional (RUP)

Proceso Racional Unificado: (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje

Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos(Booch, Rumbaugh & Jacobson, 2006).

Principales características

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software
- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Es un producto de Rational (IBM) que se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

Lenguaje Unificado de Modelado (UML)

El Lenguaje de Modelado Unificado (UML: Unified Modeling Language) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de los 80 y principios de los 90. El UML es llamado un lenguaje de modelado, no un método.

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML 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 reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. En otras palabras, es el lenguaje en el que está descrito el modelo.

1.4.2 Metodologías Ágiles

✓ XP- eXtreme Programming

XP es la primera metodología ágil y la que le dio conciencia al movimiento actual de metodologías ágiles. De la mano de Kent Beck, XP ha conformado un extenso grupo de seguidores en todo el mundo. (Calderón, Sarah y Valverde 2007)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el 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. (Calderón, Sarah y Valverde 2007)

✓ Scrum

El proceso conocido como Scrum se formaliza en el año 1995. Aunque surgió como modelo para el desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de software.

La intención de Scrum es la de maximizar la realimentación sobre el desarrollo pudiendo corregir problemas y mitigar riesgos de forma temprana. Su uso se está extendiendo cada vez más dentro de la comunidad de Metodologías Ágiles, siendo combinado con otras como XP para completar sus carencias. Cabe mencionar que Scrum no propone el uso de ninguna práctica de desarrollo en particular; sin embargo, es habitual emplearlo como un framework ágil de administración de proyectos que puede ser combinado con cualquiera de las metodologías mencionadas. (Calderón, Sarah y Valverde 2007)

✓ Crystal Clear

Alistair Cockburn es el propulsor detrás de la serie de metodologías Crystal. Las mismas presentan un enfoque ágil, con gran énfasis en la comunicación, y cierta tolerancia que la hace ideal en los casos en que sea inaplicable la disciplina.

Los siete valores o propiedades de Crystal Clear son: (Calderón, Sarah y Valverde 2007)

- Entrega frecuente. Consiste en entregar software a los clientes con frecuencia, no solamente en compilar el código.
- Comunicación osmótica. Todos juntos en el mismo cuarto.

- Mejora reflexiva. Tomarse un pequeño tiempo (unas pocas horas por algunas emanaciones o una vez al mes) para pensar bien qué se está haciendo, cotejar notas, reflexionar, discutir.
- Seguridad personal. Hablar cuando algo molesta.
- Foco. Saber lo que se está haciendo y tener la tranquilidad y el tiempo para hacerlo.
- Fácil acceso a usuarios expertos. Se le da gran importancia al contacto directo con expertos en el desarrollo de un proyecto.
- Ambiente técnico con prueba automatizada e integración frecuente. Muchos equipos ágiles compilan e integran varias veces al día.

1.5 Arquitecturas para el desarrollo de sitios web

✓ **Arquitectura cliente/servidor**

La arquitectura Cliente/Servidor es una infraestructura versátil, modular y basada en mensajes que mejora la portabilidad, la interoperabilidad y la escalabilidad de los sistemas (Méndez Cáceres, 2005).

La tecnología Cliente/Servidor es el procesamiento cooperativo de la información por medio de un conjunto de procesadores, en el cual múltiples clientes, distribuidos geográficamente, solicitan requerimientos a uno o más servidores centrales.

Desde el punto de vista funcional, se puede definir la computación Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información de forma transparente aún en entornos multiplataforma. Se trata pues, de la arquitectura más extendida en la realización de Sistemas Distribuidos.

Elementos Principales

Cliente

Un cliente es todo proceso que reclama servicios de otro, o sea, es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor. Se lo conoce con el término front-end. Éste normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario.

Servidor

Un servidor es todo proceso que proporciona un servicio a otros. Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se lo conoce con el término back-end. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.

El Esquema de funcionamiento de un Sistema Cliente/Servidor sería:

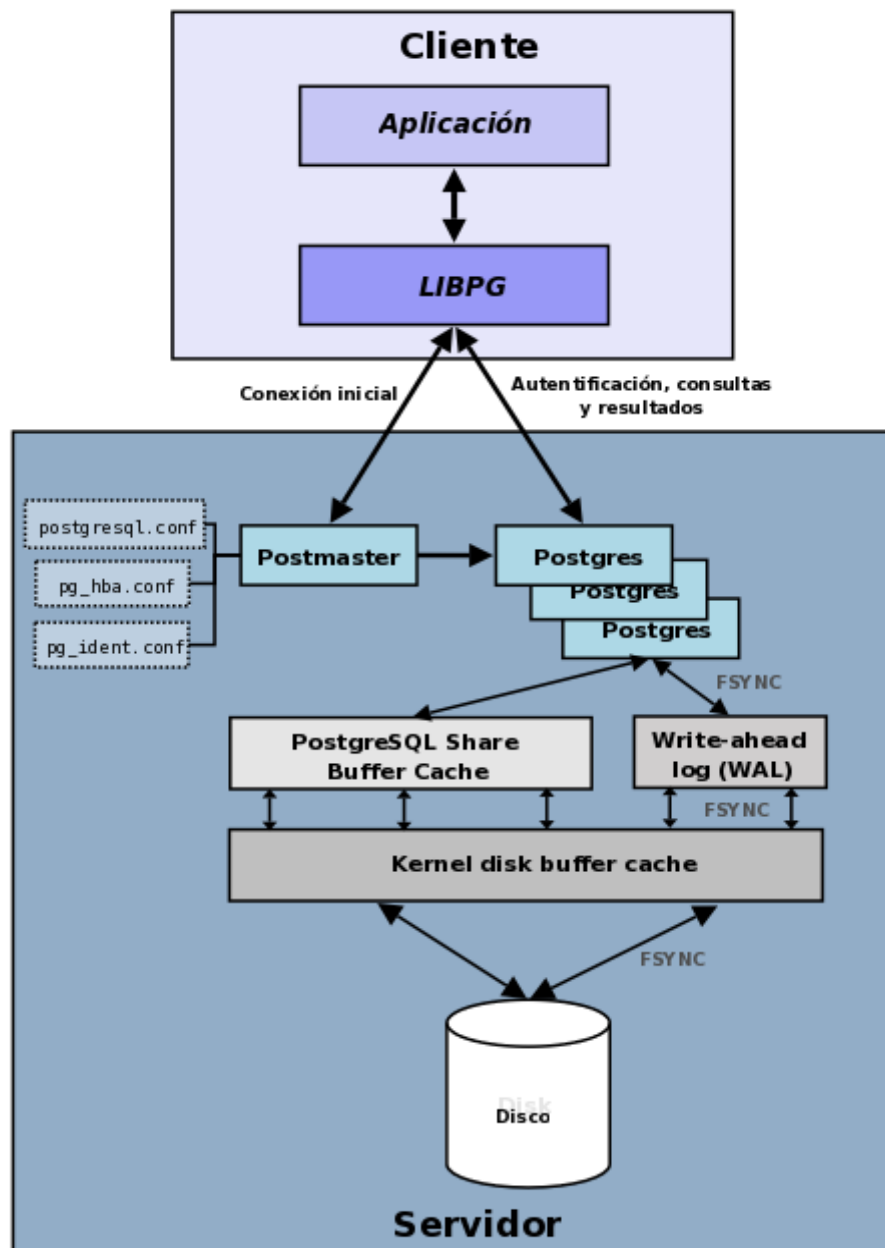


Figura 1: Esquema de funcionamiento de la arquitectura cliente/servidor

✓ Arquitectura de tres capas

La arquitectura de tres capas o programación de tres capas consiste literalmente en separar un proyecto en Capa de Presentación, Capa de Negocio y Capa de Datos. Esto permite distribuir el trabajo de creación de una aplicación por niveles.

Ventajas de esta Arquitectura

- El desarrollo se puede llevar a cabo en varios niveles.
- Desarrollos paralelos (en cada capa).
- Aplicaciones más robustas debido al encapsulamiento.
- En caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.
- Mantenimiento y soporte más sencillo (es más sencillo cambiar un componente que modificar una aplicación monolítica).
- Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad)
- Alta escalabilidad. La principal ventaja de una aplicación distribuida bien diseñada es su buen escalador, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware.

El diseño más usado en la actualidad es el de tres capas (Truel, 2006):

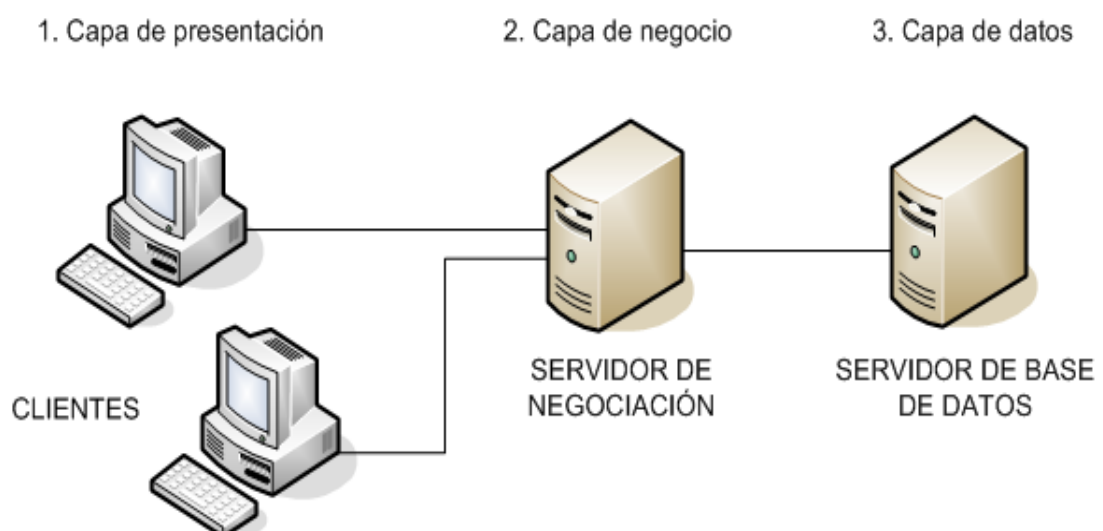


Figura 2: Arquitectura en tres capas.

Fuente: Introducción a la arquitectura en tres capas.

✓ Modelo- vista-controlador (MVC)

Modelo-Vista-Controlador es un patrón para el desarrollo del software que se basa en separar los datos (por un lado), la interfaz del usuario (por otro) y la lógica interna (por un último lado). Es mayormente usado en aplicaciones web, dónde la vista es la página HTML, el modelo es el Sistema de Gestión de Base de Datos y la lógica interna, y el controlador es el responsable de recibir los eventos y darles solución.

Si se analizan estos tres elementos por separado vemos que: **Modelo:** Es la representación de la información en el sistema. Trabaja junto a la vista para mostrar la información al usuario y es accedido por el controlador para añadir, eliminar, consultar o actualizar datos. **Vista:** Es la presentación al modelo en un formato adecuado para que el usuario pueda interactuar con él, casi siempre es la interfaz de usuario. **Controlador:** Es el elemento más abstracto. Recibe, trata y responde los eventos enviados por el usuario o por la propia aplicación. Interactúa tanto con el modelo como con la vista.

Ventajas de este patrón según (Martínez, 2007):

Este modelo de arquitectura presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa; lo cual nos permite implementarlos por separado.
- Debido a que cada parte es independiente de la otra los cambios o actualizaciones pueden ser realizados sin afectar la aplicación como un todo.
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

Figura tomada de (Martínez, 2007) que representa el esquema o patrón MVC:

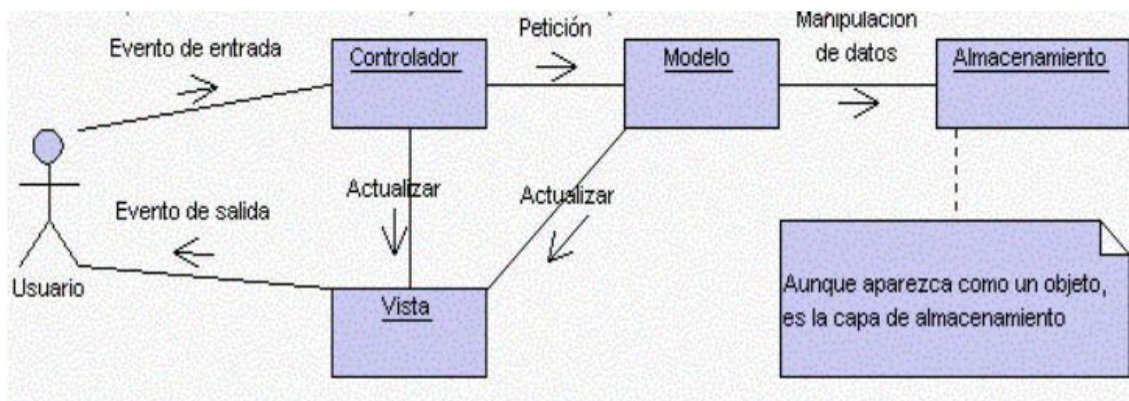


Figura 3: Esquema Modelo-Vista-Controlador

1.6 Lenguajes de programación utilizados para el desarrollo de sitios web.

✓ HTML

Desde el surgimiento de internet se han publicado sitios web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas de Hipertexto). Desarrollado por el World Wide Web Consortium (W3C). Este lenguaje se caracteriza por ser un lenguaje de fácil aprendizaje, estático, lo admiten todos los exploradores, pero la interpretación de cada navegador puede ser diferente (Webmaster, 2009).

✓ PHP

Es un lenguaje de programación utilizado para la creación de sitio web. PHP es un acrónimo recursivo que significa “PHP Hypertext Pre-processor”, (inicialmente se llamó Personal Home Page). Surgió en 1995, desarrollado por PHP Group. Es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas.

Ventajas de PHP (Pérez Valdés D., 2007):

- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Tiene capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

✓ JavaScript

Este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications. Utilizado principalmente en páginas web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código JavaScript. El código JavaScript puede ser integrado dentro de las páginas web. Los script tienen capacidades limitadas, por razones de seguridad, el código visible por cualquier usuario y el código deben descargarse completamente.

Ventajas de JavaScript (Pérez Valdés D., 2007):

- Lenguaje de scripting seguro y fiable.
- Los script tienen capacidades limitadas, por razones de seguridad.
- El código JavaScript se ejecuta en el cliente.

1.7 Frameworks

Un framework, en el desarrollo de software es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Provee una estructura y una metodología de trabajo. (Gutiérrez, 2008)

Un framework representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. Es decir, un framework está específicamente orientado a resolver o extender una solución general de un dominio de problemas. (Tadei, 2009)

✓ **Symfony:**

Symfony está más bien enfocado a desarrolladores avanzados cuyo objeto principal son aplicaciones de nivel corporativo. Este framework open-source posee una gran cantidad de propiedades y puede hacer de todo, pero su mayor falla es que es un poco más lento que otros frameworks. (eWebmaster, 2008)

Está diseñado con el objetivo de optimizar la creación de las aplicaciones Web, con el uso de sus características. Posee una librería de clases que permiten reducir el tiempo de desarrollo. Symfony está desarrollado en PHP5, se puede utilizar en plataformas *nix (Unix, Linux) y Windows. Requiere de una instalación, configuración y líneas de comando, incorpora el patrón MVC, soporta AJAX, plantillas y un gran número de bases de datos. (Valdés, 2007)

✓ **CodeIgniter:**

CodeIgniter es un programa o aplicación Web desarrollada en PHP para la creación de cualquier tipo de aplicación Web bajo PHP. Es un producto de código libre, libre de uso para cualquier aplicación. (Álvarez, 2009)

Como cualquier otro framework, CodeIgniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones Web. Marca una manera específica de codificar las páginas Web y clasificar sus diferentes scripts, lo que permite que el código esté organizado y sea más fácil de crear y mantener. CodeIgniter implementa el proceso de desarrollo llamado Modelo Vista Controlador.

Contiene ayudas para la creación de aplicaciones PHP avanzadas, que hacen que el proceso de desarrollo sea más rápido. A la vez, define una arquitectura de desarrollo que hará que se programe de una manera más ordenada y contiene diversas herramientas que ayudan a hacer aplicaciones más versátiles y seguras.

1.8 Servidor Web Apache

Un servidor de páginas Web es un programa que permite acceder a páginas Web alojadas en un ordenador. Hoy en día Apache es el servidor web más utilizado del mundo, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Es un software de código abierto que funciona sobre cualquier plataforma. Desde su origen ha evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad, surgió en abril de 1996 y ya en julio del 2002 era utilizado por el 57% de los sitios Web de Internet. (Rocha, 2008)

Características del Apache:

- ✓ Es una tecnología gratuita y de código abierto, lo que proporciona transparencia en todo el proceso de instalación.
- ✓ Es prácticamente universal por su disponibilidad en multitud de sistemas operativos.
- ✓ Posee una alta configurabilidad en la creación y gestión de logs, de este modo es posible tener un mayor control sobre lo que sucede en el servidor.
- ✓ Fácil integración con varios lenguajes de programación como: Java, Perl y especialmente PHP. Dicha relación ha dado lugar al desarrollo de aplicaciones como el APPSERV y XAMPP, los cuales instalan el Apache y el PHP configurados para su uso. (Vedora Willock, 2006)

1.9 Sistemas gestores de Bases de Datos

Un Sistema Gestor de Bases de Datos (SGBD) o DBMA (Data Base Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.

Algunos ejemplos de (SGBD) son:

✓ **SQL Server**

SQL Server es un Sistema de Gestión de Bases de Datos Relacionales (SGBDR), compuesto por un conjunto de elementos, que se integran con el sistema operativo Windows NT, para proporcionar un entorno avanzado de procesamiento de datos, dentro de una arquitectura cliente-servidor. Fue desarrollado por Microsoft y permite, como su propio nombre indica, la gestión de un entorno de bases de datos relacional. SQL Server abarca, tanto el área de diseño, como el de la de administración, proporcionando una interfaz bastante amigable con el usuario.

SQL Server utiliza una extensión (dialecto) al SQL estándar, que se denomina Transact SQL. Esto quiere decir que soporta el SQL de ANSI (American National Standards Institute), pero además se le han añadido ciertas funciones adicionales, no contempladas en el estándar, y que son específicas para este producto, es decir, si ejecutamos una sentencia del conjunto adicional (Transact SQL) en otro SGBDR, éste no la entendería.

✓ **MySQL**

MySQL es un servidor de base de datos relacional que soporta el bien conocido lenguaje de base de datos SQL (Structured Query Language). Es un sistema cliente-servidor. Desarrollado, distribuido y soportado por MySQL AB.

Es software de código abierto lo que significa que es posible para cualquiera usarlo y modificarlo.

Las principales características de MySQL

Interno y portabilidad:

- Escrito en C y C++. Probado con un amplio rango de distintos compiladores.
- Trabaja en múltiples plataformas.

- Con APIs para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby y Tel.
- Completamente multi-hilos. Puede utilizar varios CPUs si están disponibles.

Tipos de datos:

- Posee numerosos tipos de datos: enteros con y sin signo, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, y ENUM.
- Records de longitud fija y variable.

✓ Oracle

Corría no solo en sistemas Digital VAX/VMS. También en UNIX y otras plataformas. En 1995 Oracle adquiere la habilidad de correr sobre 30 plataformas. En la actualidad, además de mantener el soporte de múltiples plataformas, tiene un núcleo Oracle que data de mediados de los 80s que incluye desarrollo de software complementario y herramientas de toma de decisiones (business intelligence) entre otros.

Características de la base de datos Oracle

PL/SQL

- Es la extensión de SQL que utiliza Oracle.
- Puede ser utilizado para la creación de procedimientos almacenados, triggers, control de ciclos, sentencias condicionales y manejo de errores.
- Se pueden compilar y guardar procedimientos PL/SQL en la BD.

Servicios web.

- La base de datos puede funcionar como un proveedor de servicios web implementados a través de XML DB.
- Los servicios web habilitan a SQL para enviar consultas y recibir resultados como XML o invocar funciones o paquetes de funciones PL/SQL y recibir resultados.

✓ PostgreSQL

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD(Berkeley Software Distribution) y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

La última serie de producción es la 9.1. Sus características técnicas la hacen una de las bases de datos más potentes y robustos del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.

Características

- Atomicidad (Indivisible) es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- Consistencia es la propiedad que asegura que sólo se empieza aquello que se puede acabar.
- Aislamiento es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que dos transacciones sobre la misma información nunca generará ningún tipo de error.
- Durabilidad es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.
- Corre en casi todos los principales sistemas operativos como: Linux, Unix, Windows.
- Altamente adaptable a las necesidades del cliente.
- Soporte nativo para los lenguajes más populares del medio: PHP, C, C++, Perl.

1.10 Herramientas de desarrollo

✓ **Dreamweaver**

Dreamweaver es la herramienta de diseño de páginas Web más avanzada, tal como se ha afirmado en muchos medios. Aunque sea un experto programador de HTML el usuario que lo maneje, siempre se encontrarán en este programa razones para utilizarlo, sobre todo en lo que a productividad se refiere. (Álvarez, 2010)

Cuenta con una amplia gama de herramientas que posibilitan la creación de sitios web desde los más sencillos hasta los más complejos, permitiendo utilizar casi todos los recursos web. Ayuda a la creación de páginas web dinámicas apoyadas en Bases de Datos. Permite escribir código script para extender las capacidades de las páginas web creadas con nuevos comportamientos. Soporta varias tecnologías del servidor entre las que se incluye el PHP.

✓ Rational Rose

Para apoyar el trabajo de la metodología RUP, ha sido desarrollada por la Compañía norteamericana Rational Corporation la herramienta CASE (Computer Assisted Software Engineering) Rational Rose desde el año 2000. Esta herramienta integra todos los elementos que propone la metodología para cubrir el ciclo de vida de un proyecto. Esta herramienta CASE propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. (Rodríguez Febles, 2005)

Conclusiones

Se realizó un análisis sobre las diferentes metodologías y tecnologías existentes en la actualidad, que permitió considerar a RUP como la metodología más apropiada para el desarrollo del proyecto y a UML como lenguaje de modelado de la aplicación Web, para el desarrollo de la misma utilizó PHP como lenguaje de programación, Symfony como framework para PHP, servidor Web Apache y PostgreSQL para el almacenamiento de los datos, considerándose estas las más adecuadas para la investigación por ser sistemas multiplataforma, de libre acceso y por las múltiples facilidades que juntas brindan al programador.

Capítulo II: Descripción de la aplicación propuesta para contribuir al proceso de enseñanza-aprendizaje de la asignatura metodología de la investigación para los estudiantes de Cultura Física de la Uniss.

Introducción

En este capítulo se hace un estudio del modelo del negocio, el cual permite comprender mejor la problemática después de determinar los actores del negocio, los trabajadores y los casos de uso, conformándose el diagrama de casos de uso y la expansión de los mismos. Además se exponen las reglas del negocio a tener en cuenta, se realizan diagramas de actividades y modelos de objeto.

También se dejan plasmados los casos de uso del sistema, sus actores, diagramas de casos de uso del sistema así como la descripción de cada uno según lo propuesto por la metodología RUP.

2.1 Modelo del negocio

El modelado de negocio es una técnica para comprender los procesos de negocio de la organización. Además de identificar los casos de uso y las entidades del negocio relevantes que el software debe soportar, de forma que se puede modelar solo lo necesario para que se comprenda el contexto. (Jacobson, Booch, & Rumbaugh, 2000)

El modelo del negocio está soportado por dos tipos de modelos de UML: modelo de casos de uso y modelo de objetos. (Jacobson, Booch, & Rumbaugh, 2000)

A continuación se describe el proceso de negocio que se lleva a cabo en la asignatura de metodología de la investigación de la carrera de Cultura Física en la Universidad de Sancti Spíritus, mediante los artefactos propuestos por la metodología RUP y modelados por el lenguaje UML.

2.1.1 Identificación de los procesos del negocio

Cuando se hable de procesos de negocio se puede decir que son un grupo de tareas relacionadas lógicamente que se llevan a cabo en una determinada secuencia y forma, y que emplean los recursos de la organización para dar resultados que apoyen sus objetivos. (Rumbaugh, Booch, & Jacobson, 2006)

A partir del planteamiento anterior se identificó el siguiente proceso del negocio.

En la universidad de Sancti Spíritus, la carrera de Cultura Física cuenta con un departamento donde radican los profesores de la asignatura metodología de la investigación, cuando los estudiantes necesitan solicitar alguna bibliografía ya sea un libro una tesis u otro documento relacionado con la asignatura se dirigen a este departamento y allí su profesor le entrega la bibliografía solicitada o se la envía a través del correo electrónico.

2.1.2 Reglas del negocio

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio (Jacobson, Booch, &Rumbaugh, 2006).

Partiendo de lo planteado anteriormente fueron identificadas las siguientes reglas:

- ✓ El usuario que solicite información debe ser alumno de la carrera de Cultura Física de la Universidad de Sancti Spíritus.
- ✓ Los estudiantes solo pueden pedir artículos bibliográficos a sus profesores correspondientes.

2.1.3 Modelo de casos de uso del negocio

El modelo de Casos de Uso del Negocio es el encargado de describir los procesos de una empresa utilizando los casos de uso y los actores, en correspondencia, a su vez, con los procesos del negocio y los clientes (Jacobson, Booch, &Rumbaugh, 2000).

2.1.4 Actores del negocio

Se considera actor del negocio a cualquier individuo, grupo, entidad, organización, máquina o sistema de información externo; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados (Jacobson, Booch, &Rumbaugh, 2000)

Actor del Negocio	Justificación
Estudiante	Interesado en consultar la bibliografía.

Tabla 1: Actor del negocio.

2.1.5 Diagrama de casos de uso del negocio

Los diagramas de casos de uso se utilizan para especificar las funcionalidades y el comportamiento de un sistema mediante su interacción con los usuarios y otros

sistemas. O sea es un diagrama que muestra la relación entre los actores y los casos de uso en un sistema (Jacobson, Booch, & Rumbaugh, 2000).

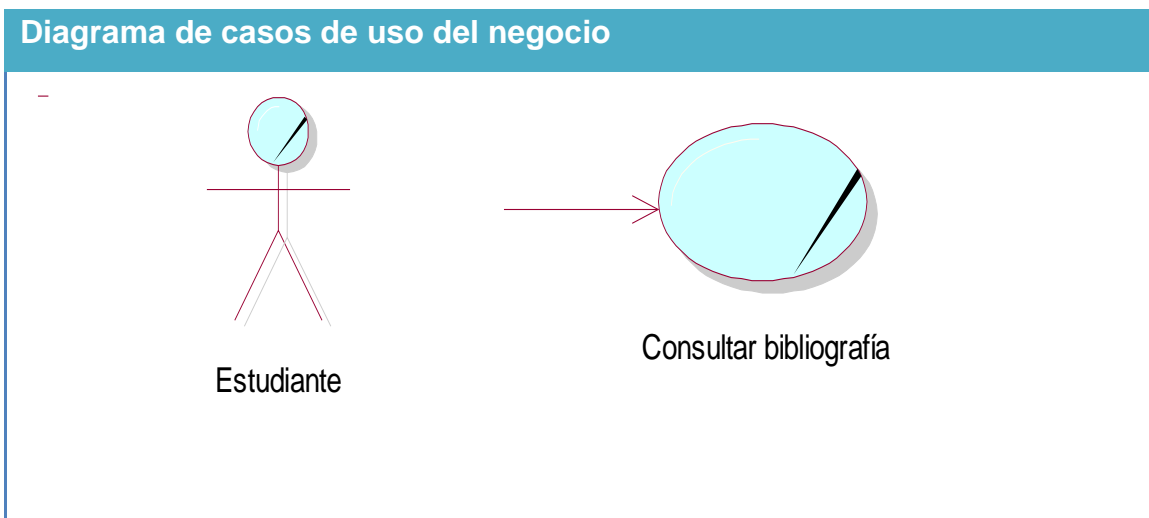


Figura 4: Diagrama de casos de uso del negocio.

2.1.6 Trabajadores del negocio

Un trabajador es una abstracción de una persona o grupo de personas, una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio, manipulando entidades del mismo y representando un rol (González, 2005).

Basado en el concepto se define el trabajador del negocio, como se muestra en la tabla siguiente:

Trabajador del Negocio	Justificación
Profesor	Es el encargado de brindarles la información a los estudiantes.

Tabla 2: Trabajador del negocio.

2.1.7 Descripción de los Casos de Uso del negocio.

Caso de uso del negocio	Consultar información
Actor	Estudiante.
Propósito	Obtener bibliografía.
Resumen: Este caso de uso comienza cuando el estudiante solicita una bibliografía específica. El profesor busca la misma, si la encuentra se la entrega y se termina el caso de uso, en caso contrario, realiza una búsqueda en Intranet, Internet u otras fuentes y conforma así un documento que satisfaga la necesidad del estudiante, se la entrega y culmina el caso de uso.	

Curso Normal de los eventos	
Acción del Actor	Respuesta del negocio
1. El estudiante solicita bibliografía.	
	2. El profesor busca la bibliografía solicitada, si la encuentra se la entrega al estudiante, si no ir a Subsección 1.
3. El estudiante recibe la bibliografía y se termina el caso de uso.	
	Subsección 1.
	2.1 Si el profesor no cuenta con la bibliografía que está solicitando el estudiante realiza una búsqueda en Intranet, Internet u otras fuentes y se la entrega a este.
3. El estudiante recibe la bibliografía y finaliza el caso de uso.	
Prioridad	Alta

Tabla 3: Descripción de los casos de uso del negocio.

2.1.8 Diagrama de actividades

Un diagrama de actividades es un diagrama que muestra el flujo de actividad a actividad; los diagramas de actividad tratan la vista dinámica de un sistema. Un caso especial de diagrama de estados (aquellos diagramas que tratan la vista dinámica de un sistema) en el cual todos o casi todos los estados son estados de acción y en el cual todas o casi todas las transiciones son disparadas por la terminación de las acciones en los estados origen. (Rumbaugh, Booch, & Jacobson, 2006). El diagrama de actividad se encuentra en el [Anexo1](#).

2.1.9 Modelo de objetos del negocio

Un modelo de objetos del negocio es un modelo interno a un negocio. Describe como cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio y unidades de trabajo. (Jacobson, Booch, & Rumbaugh, 2006)

Una entidad del negocio representa algo, que los trabajadores toman, inspeccionan, manipulan, producen o utilizan en un caso de uso del negocio. (Jacobson, Booch, & Rumbaugh, 2006). El modelo de objetos del negocio se encuentra en el [Anexo2](#).

2.2 Requerimientos

2.2.1 Requerimientos funcionales

Los requerimientos funcionales expresan una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara lo que debe hacer el sistema, siempre basándose en las necesidades de los usuarios (González, 2005).

1-Autenticarse

2-Cambiar contraseña

3-Gestionar libros

3.1- Insertar libro

3.2- Modificar libro

3.3- Eliminar libro

4-Gestionar tesis

4.1- Insertar tesis

4.2- Modificar tesis

4.3- Eliminar tesis

5-Gestionar documentos actuales

5.1- Insertar documentos actuales

5.2- Modificar documentos actuales

5.3- Eliminar documentos actuales

6-Gestionar concepto

6.1- Insertar concepto

6.2- Modificar concepto

6.3- Eliminar concepto

7-Gestionar ejemplos

7.1- Insertar ejemplos

7.2- Modificar ejemplos

7.3- Eliminar ejemplos

8-Gestionar estructura para trabajos investigativos

8.1- Insertar estructura

8.2- Modificar estructura

8.3- Eliminar estructura

9-Gestionar curiosidades

9.1- Insertar curiosidades

9.2- Modificar curiosidades

9.3- Eliminar curiosidades

10-Gestionar ejercicios de seleccionar

10.1- Insertar ejercicio

10.2- Modificar ejercicio

10.3- Eliminar ejercicio

11-Gestionar ejercicios de completar

11.1- Insertar ejercicio

11.2- Modificar ejercicio

11.3- Eliminar ejercicio

12-Gestionar ejercicios de verdadero o falso

12.1- Insertar ejercicio

12.2- Modificar ejercicio

12.3- Eliminar ejercicio

13- Gestionar usuario

13.1- Insertar usuario

13.2- Modificar usuario

13.3- Eliminar usuario

14-Mostrar libros

15- Mostrar tesis

16- Mostrar documentos actuales

17- Mostrar conceptos

- 18- Mostrar ejemplos**
- 19- Mostrar estructura investigativa**
- 20- Mostrar curiosidades**
- 21- Resolver ejercicio de seleccionar**
- 22- Resolver ejercicio de completar**
- 23- Resolver ejercicio de verdadero o falso**
- 24- Evaluar respuestas**
- 25- Mostrar respuesta correcta**
- 26- Descargar ficheros**
 - 26.1- Descargar libros
 - 26.2- Descargar tesis
 - 26.3- Descargar documentos actuales
 - 26.4- Descargar conceptos
 - 26.5- Descargar ejemplos
 - 26.6- Descargar estructura investigativa

2.2.2 Requerimientos no funcionales

Los requerimientos no funcionales describen las restricciones del sistema o del proceso de desarrollo; no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida, en cuanto a prestaciones, atributos de calidad y la representación de datos que se utiliza en la interfaz del sistema. (Soto, 2006)

✓ Requerimientos de Apariencia o Interfaz Externa

La interfaz del sistema debe ser a través de una página Web dinámica y personalizada de acuerdo al tipo de usuario que acceda al sistema. Estará diseñada de modo tal que el usuario pueda tener en todo momento el control de la aplicación, lo que le permitirá ir de un punto a otro dentro de ella con gran facilidad. Se cuidará porque la aplicación sea lo más interactiva posible.

✓ Requerimientos de Usabilidad

El sistema es de libre acceso, pero a la aplicación diseñada para el mantenimiento del mismo solo tendrán acceso los usuarios definidos por la Administración de Red de la Universidad de Sancti Spíritus.

Serán el administrador del sitio y los estudiantes pertenecientes a la carrera de Cultura Física de la Universidad de Sancti Spíritus los principales usuarios de este software.

✓ **Requerimientos de Soporte**

Se requiere un servidor de bases de datos con soporte para grandes volúmenes de información, velocidad de procesamiento y tiempo de respuesta rápido en accesos concurrentes. Se documentará la aplicación para garantizar su soporte.

El administrador tendrá la responsabilidad de mantener actualizada la aplicación.

Las pruebas del sistema se realizarán en la Universidad de Sancti Spíritus. Dichas pruebas permitirán evaluar en la práctica la funcionalidad y las ventajas de este nuevo producto.

✓ **Requerimientos de Portabilidad**

La plataforma seleccionada para desarrollar la aplicación fue Windows, pero puede ser ejecutada desde otras plataformas que soporten el lenguaje PHP y MySQL, por ejemplo Linux.

✓ **Requerimientos de Seguridad**

Debe garantizar la conectividad e integridad de los datos almacenados a través de la red. Esto está garantizado por el Sistema Operativo.

Debe garantizar la confidencialidad para proteger la información de acceso no autorizado. Esto estará garantizado por el Sistema Gestor de Base de Datos.

El sistema impondrá un estricto control de acceso que permitirá a cada usuario tener disponible solamente las opciones relacionadas con su actividad.

La información deberá estar disponible a los usuarios en todo momento, limitada solamente por las restricciones que estos tengan de acuerdo con la política de seguridad del sistema.

✓ **Confiabilidad**

Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario.

El sistema en casos de fallos debe garantizar que las pérdidas de información sean mínimas.

✓ **Requerimientos de Ayuda y Documentación en Línea**

El sistema contará con una ayuda que explicará de manera clara y sencilla al usuario todas las funcionalidades del sistema.

La ayuda quedará conformada por un menú general que le facilitará al usuario poder ir de un lugar a otro sin perderse.

✓ **Requerimientos de Software**

Se debe disponer de un sistema operativo compatible. El sistema debe poderse ejecutar en entornos Windows y Linux.

Del lado del servidor se utilizará Apache como servidor Web, PHP 5.2.6 o superior como lenguaje de programación, MySQL como gestor de base de datos para garantizar la integridad y consistencia de la información.

Del lado del cliente se requiere un navegador Mozilla Firefox 3.0 o superior.

✓ **Requerimientos de Hardware**

Para la puesta en práctica del proyecto se requieren máquinas con los siguientes requisitos:

Se requiere de una máquina que funcione como servidor, la cual debe tener como mínimo:

- Un Procesador Pentium III con 1 GHz de frecuencia o superior
- 512 Mb de Memoria RAM
- 40 GB de Disco Duro.
- Navegador Web.
- Periférico de comunicación: Módem y tarjeta de red.

Las computadoras situadas en los puestos de trabajo de los usuarios requerirán como mínimo:

- Un Procesador Pentium III.
- 128 Mb de Memoria RAM.
- Deben estar conectadas en red con el servidor a través de una tarjeta de red de 100 Mbps.

✓ **Restricciones en el diseño y la implementación**

Se utilizarán herramientas de desarrollo que garanticen la calidad de todo el ciclo de desarrollo del producto.

2.3 Descripción del sistema propuesto

2.3.1 Modelo de casos de uso del sistema

El modelado de Casos de Uso es la técnica más simple y que emplean los desarrolladores de software para modelar los requisitos del sistema desde la perspectiva del usuario. El modelo de casos de uso consiste en actores y casos de uso. Los actores representan usuarios y otros sistemas que interactúan con el sistema y los casos de uso representan el comportamiento del sistema, los escenarios que el sistema atraviesa en respuesta a un estímulo desde un actor (Popkin Software and System, 2005).

2.3.2 Actores del sistema

Un actor no es más que un rol que juega un usuario de Caso de Uso cuando interactúan con estos casos de uso. Los actores representan a terceros fuera del sistema que colaboran con el mismo. Una vez que se han identificado los actores del sistema, se ha identificado el entorno externo del sistema. (Jacobson, Booch, & Rumbaugh, 2006).

Teniendo en cuenta lo planteado anteriormente se definieron los siguientes actores del sistema:

Actores del sistema	Descripción
Administrador	Es la persona con acceso total al sistema y el encargo de mantenerlo actualizado.
Estudiante	Es la persona que utiliza el sistema para adquirir conocimientos. Este tiene acceso limitado al sistema. Puede revisar toda la información teórica, realizar los ejercicios, revisar la ayuda y reportes del sistema en cualquier momento.

Tabla 4: Actores del sistema.

2.3.3 Casos de uso del sistema

Cada forma en que los actores usan el sistema se representa con un Caso de Uso. Los Casos de Uso son “fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. Un Caso de Uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia (Jacobson, 2006).

Para este sistema se definieron los siguientes casos de uso:

- 1- Autenticarse
- 2- Cambiar contraseña
- 3- Gestionar libros
- 4- Gestionar tesis
- 5- Gestionar documentos actuales
- 6- Gestionar concepto
- 7- Gestionar ejemplos
- 8- Gestionar estructura para trabajos investigativos
- 9- Gestionar curiosidades
- 10- Gestionar ejercicios de seleccionar
- 11- Gestionar ejercicios de completar
- 12- Gestionar ejercicios de verdadero o falso
- 13- Gestionar usuario
- 14- Mostrar libros
- 15- Mostrar tesis
- 16- Mostrar documentos actuales
- 17- Mostrar conceptos
- 18- Mostrar ejemplos
- 19- Mostrar estructura investigativa
- 20- Mostrar curiosidades
- 21- Resolver ejercicio de seleccionar
- 22- Resolver ejercicio de completar
- 23- Resolver ejercicio de verdadero o falso
- 24- Evaluar respuestas
- 25- Mostrar respuesta correcta
- 26- Descargar ficheros

2.3.4 Diagrama de casos de uso del sistema

Para facilitar el trabajo con los casos de uso y la organización de los elementos se agrupan los casos de uso en paquetes, los cuales se muestran a continuación.

Nombre del paquete	Funcionalidades	Criterio de agrupamiento	de Actores
Seguridad	1- Autenticarse 2- Cambiar contraseña	Funcionalidades (Requerimientos funcionales)	Administrador

Ejercicios	<ol style="list-style-type: none"> 1- Resolver ejercicio de seleccionar 2- Resolver ejercicio de completar 3- Resolver ejercicio de verdadero o falso 	Funcionalidades (Requerimientos funcionales)	Estudiante
Gestión	<ol style="list-style-type: none"> 1- Gestionar libros 2- Gestionar tesis 3- Gestionar documentos actuales 4- Gestionar concepto 5- Gestionar ejemplos 6- Gestionar estructura para trabajos investigativos 7- Gestionar curiosidades 8- Gestionar usuario 9- Gestionar ejercicios de seleccionar 10- Gestionar ejercicios de completar 11- Gestionar ejercicios de verdadero o falso 	Funcionalidades (Requerimientos funcionales)	Administrador
Reportes	<ol style="list-style-type: none"> 1- Mostrar libros 2- Mostrar tesis 3- Mostrar documentos actuales 4- Mostrar conceptos 5- Mostrar ejemplos 6- Mostrar estructura investigativa 7- Mostrar curiosidades 8- Evaluar respuestas 9- Mostrar respuesta correcta 10- Descargar ficheros 	Funcionalidades (Requerimientos funcionales)	Estudiante

Tabla 5: Casos de uso del sistema por paquetes.

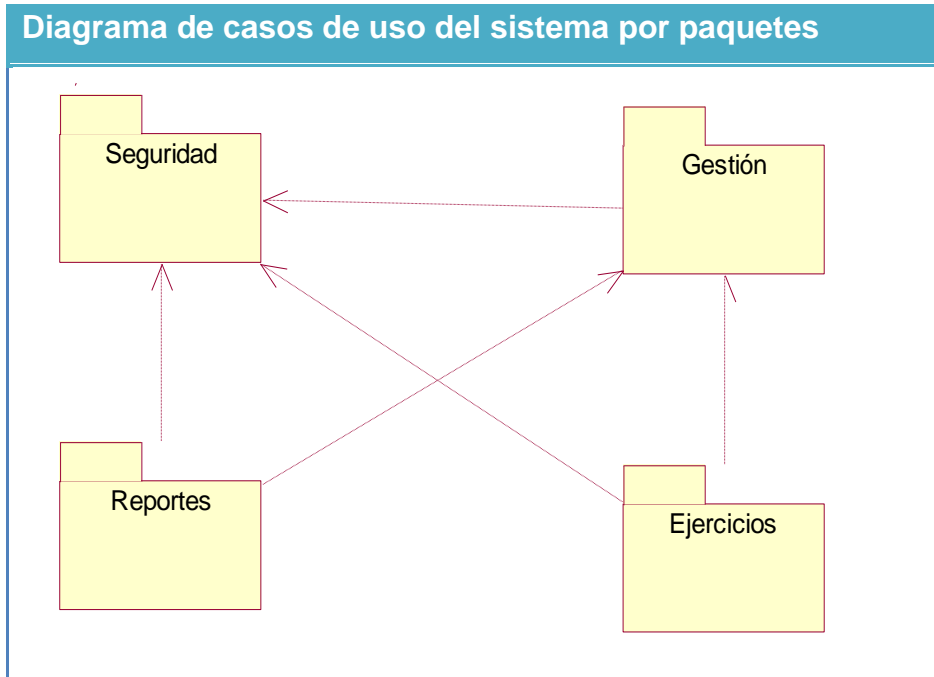


Figura5: Diagrama de casos de uso del sistema por paquetes.

Diagrama de casos de uso del sistema: Paquete seguridad.

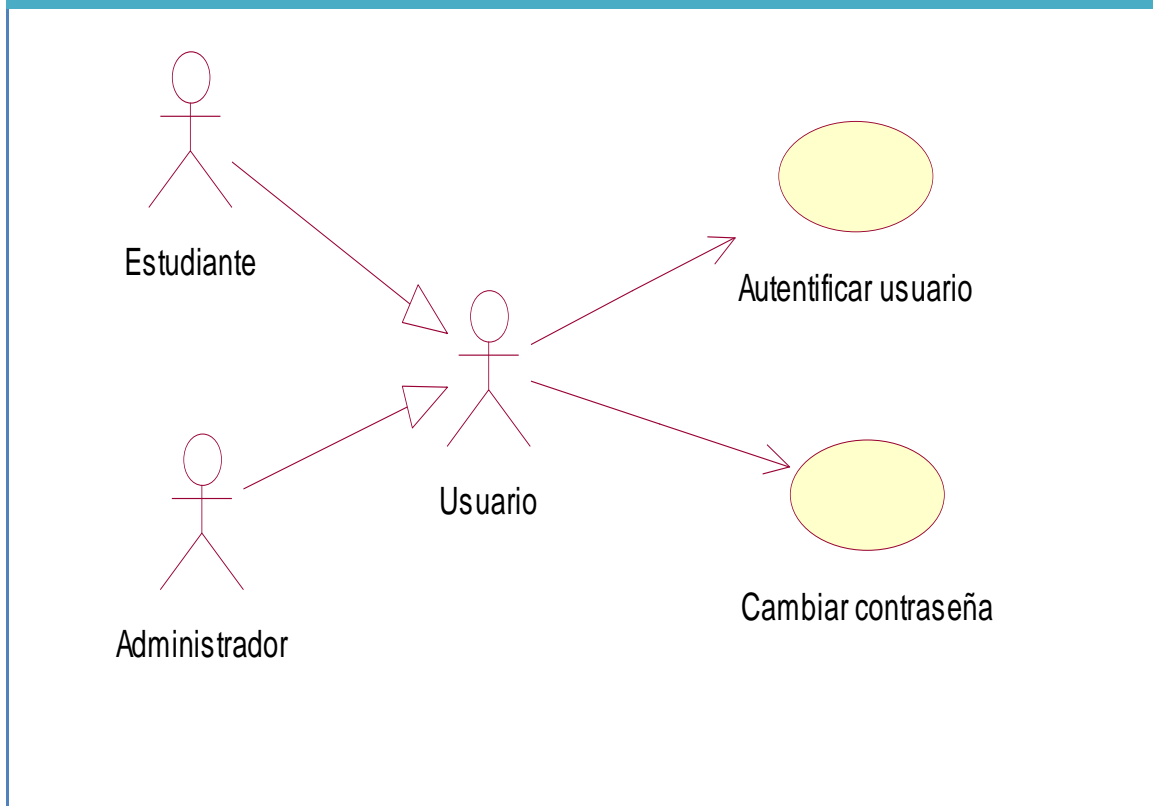


Figura 6: Diagrama de casos de uso del sistema: Paquete seguridad.

La descripción de los casos de del sistema del paquete seguridad se encuentran en el [Anexo3](#).

Diagrama de casos de uso del sistema: Paquete ejercicios.

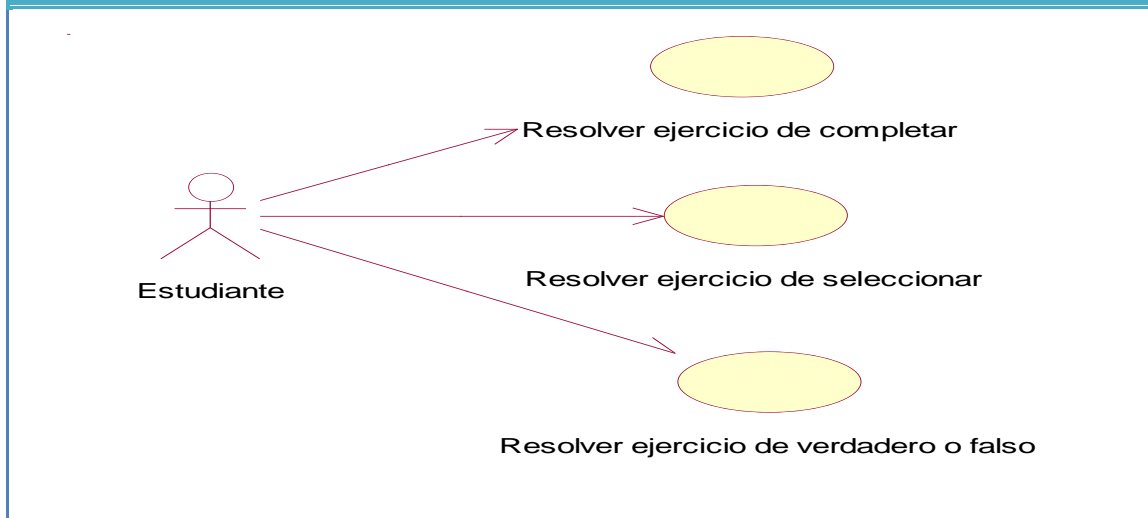


Figura 7: Diagrama de casos de uso del sistema: Paquete ejercicios.

La descripción de los casos de del sistema del paquete ejercicios se encuentran en el [Anexo4](#).

Diagrama de casos de uso del sistema: Paquete gestión.

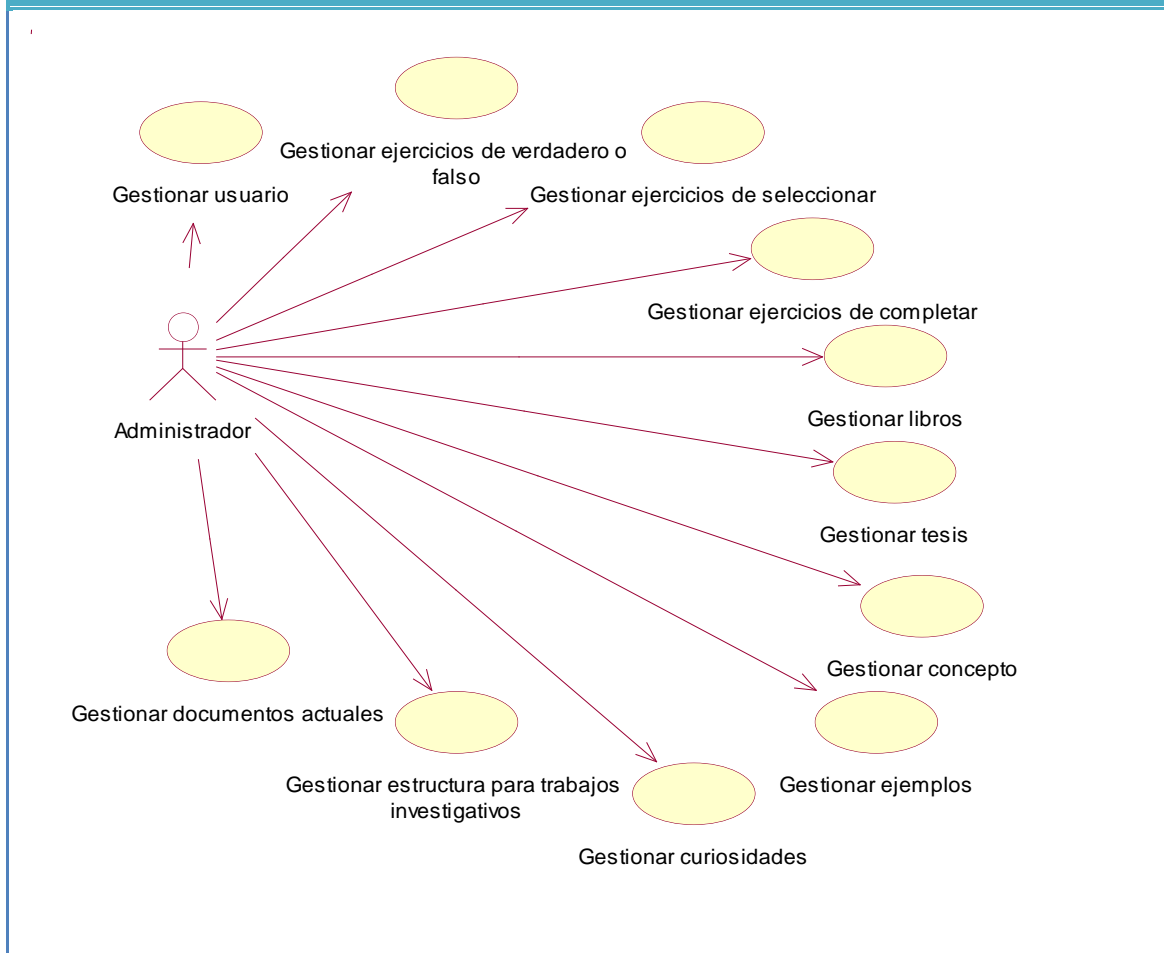


Figura 8: Diagrama de casos de uso del sistema: Paquete gestión.

La descripción de los casos de del sistema del paquete gestión se encuentran en el [Anexo5.](#)

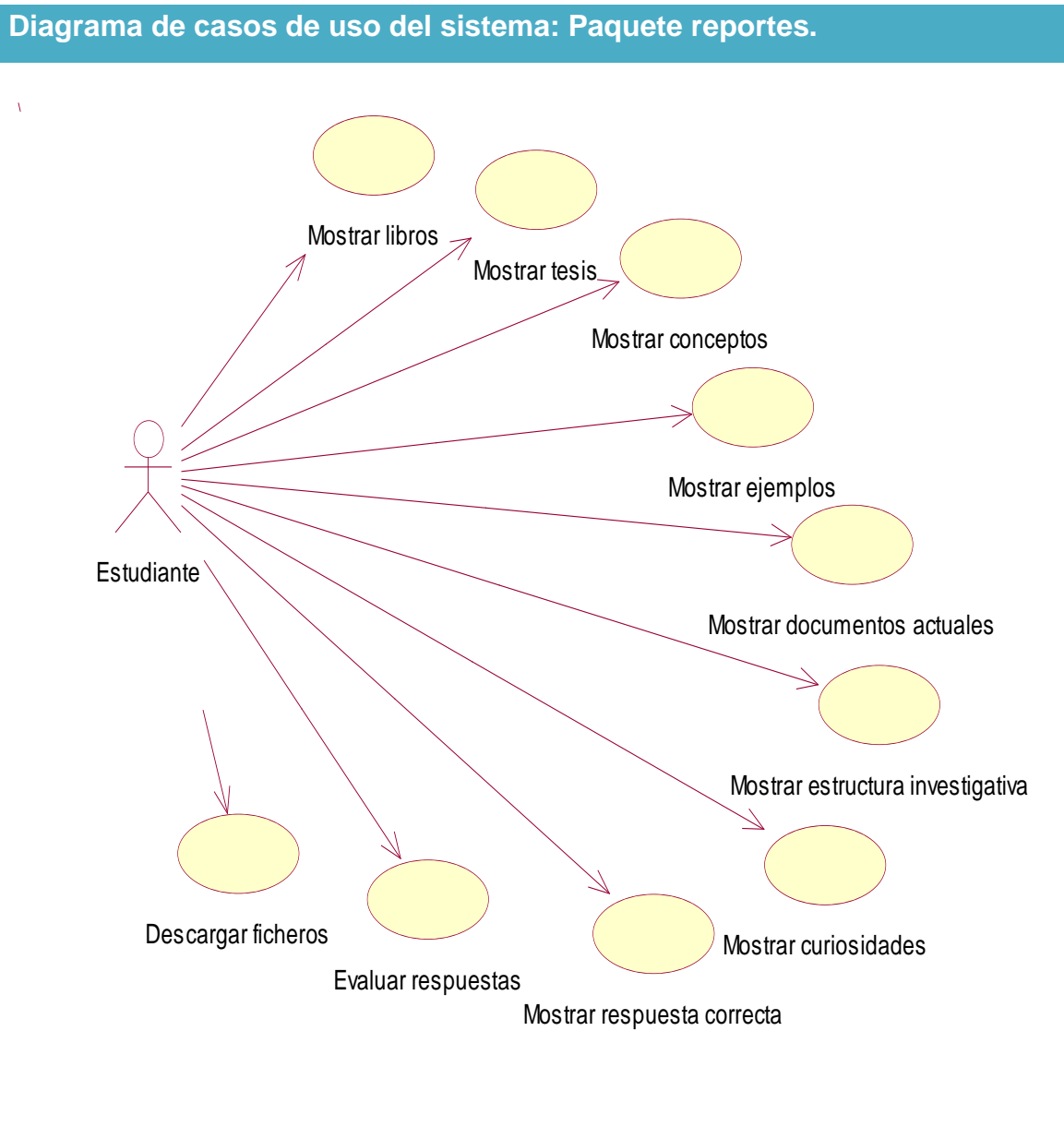


Figura 9: Diagrama de casos de uso del sistema: Paquete reportes.

La descripción de los casos de del sistema del paquete reportes se encuentran en el [Anexo6.](#)

2.4 Conclusiones

En este capítulo queda definido el modelo de objetos del problema y los principales conceptos asociados a este. Tras lograr la modelación del dominio de los procesos actuales se ha logrado una mejor comprensión de los problemas que se tienen que

resolver. Se definieron los principales requerimientos funcionales y no funcionales del sistema, así como los actores y su relación con estos, se definieron los paquetes de casos de uso que permiten agruparlos según su funcionalidad y por último se realiza una descripción de cada caso de uso.

Capítulo III: Implementación de una Aplicación Web que contribuya al proceso de enseñanza-aprendizaje de la asignatura metodología de la investigación para los estudiantes de Cultura Física de la Uniss.

Introducción

El diseño es el primer paso en la fase de desarrollo de cualquier producto o sistema de ingeniería y la importancia del mismo se puede resumir en una palabra: calidad. Durante esta etapa se precisa el cumplimiento de los requerimientos funcionales y el desarrollo de un software con calidad.

En este capítulo queda plasmado el diagrama de componente y de despliegue, así como el modelo de datos a partir del diagrama de clases persistentes y por último se hará referencia a los estándares de diseño y de programación que se tienen en cuenta.

3.1 Diagrama de Clases Web

El diagrama de clases de diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Contiene: clases, asociaciones y atributos; interfaces, con sus operaciones y constantes; métodos; información sobre los tipos de atributos; navegabilidad y dependencias. A diferencia del modelo conceptual, un diagrama de este tipo contiene las definiciones de las entidades del software en vez de conceptos del mundo real. (Larman, 2004)

Los diagramas de clases del diseño fueron elaborados a partir de los diferentes casos de uso del sistema, estos se encuentran en el [Anexo 7](#).

3.2 Diseño de la base de datos

Cuando se va a realizar el diseño de la base de datos para un sistema dado, es necesario determinar los datos que se deben tomar en cuenta y las dependencias funcionales existentes entre ellos. Esto se obtiene luego de realizada la etapa de análisis del sistema y partiendo de lo obtenido en ésta (Mato, 2006).

3.2.1 Diagrama de clases persistentes

El diagrama del modelo lógico de datos o diagrama de clases persistentes, muestra las clases capaces de mantener su valor en el espacio y en el tiempo (Jacobson, Booch & Rumbaugh, 2006).

Diagrama de clases persistentes



Figura 10: Diagrama de clases persistentes.

3.2.2 Modelo de datos

El modelo físico de datos, representa la estructura o descripción física de las tablas de la base de datos, obtenido a partir del modelo lógico de datos (Rumbaugh, Booch, & Jacobson, 2006). Este modelo se encuentra en el [Anexo 8](#).

3.3 Principios de diseño

El objetivo de diseñar es producir un modelo o representación de una entidad que va a construir posteriormente. El diseño es el lugar donde se fomenta la calidad en el desarrollo del software, nos proporciona representaciones de software en las que se puede valorar la calidad.

3.3.1 Diseño de la interfaz del sistema

La interfaz del software se concibe con la mayor sencillez posible, de manera que el trabajo con el sistema sea fácil y ameno. Se utilizan colores agradables a la vista y una letra legible para su rápida comprensión, además se utiliza un lenguaje conocido por el usuario, sin emplear términos informáticos.

El software consta de un menú que aparece de manera permanente en el cual se encuentran todas las funcionalidades del sistema, el usuario podrá acceder a ellas en dependencia de los permisos que tenga.

3.3.2 Tratamiento de errores

El sistema propuesto valida constantemente de la información, con el propósito de minimizar las posibilidades de introducir errores por parte del usuario. En caso de errores se le comunica al usuario a través de un mensaje de alerta en un lenguaje fácil de comprender.

3.3.3 Concepción general de la ayuda

El sistema cuenta con una ayuda capaz de explicar al usuario cómo es la manipulación y funcionamiento del software. La ayuda está formada por un menú interno que contiene todas las funcionalidades del sistema. El usuario podrá navegar por cada uno de ellos y consultarlo en cualquier momento ya que se encuentra ubicada en el menú principal de la aplicación.

3.3.4 Seguridad

El sistema mantiene un fuerte mecanismo de seguridad, basado en un nombre de usuario y contraseña para el acceso al mismo. Las personas encargadas de actualizar cualquier información presente en la base de datos del sistema, tendrán nombres de usuario y contraseñas únicas, evitando que esta acción pueda ser realizada por cualquier otro usuario. Además los usuarios del sistema tendrán solamente permiso de acceso a las funcionalidades correspondientes a su rol.

La consistencia de los datos es otro aspecto que se toma en consideración, y para ello el sistema cuenta con formularios validados, con funciones del lenguaje PHP que

garantizan que la información que se registre en la base de datos sea totalmente consistente e íntegra.

3.4 Modelo de implementación

El modelo de implementación denota la implementación del sistema en términos de componentes y subsistemas de implementación. Describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración, y modularización disponibles en el entorno de la implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros. (Méndez Cáceres, 2005).

3.4.1 Diagrama de despliegue.

El diagrama de despliegue muestra la forma en que los componentes se desplegarán en el sistema, en este se detalla las especificaciones de la red, del servidor y los requisitos de hardware. Un diagrama de despliegue es un diagrama que muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. (Jacobson, Booch, & Rumbaugh, 2006)

A continuación se muestra el Diagrama de Despliegue de la aplicación:

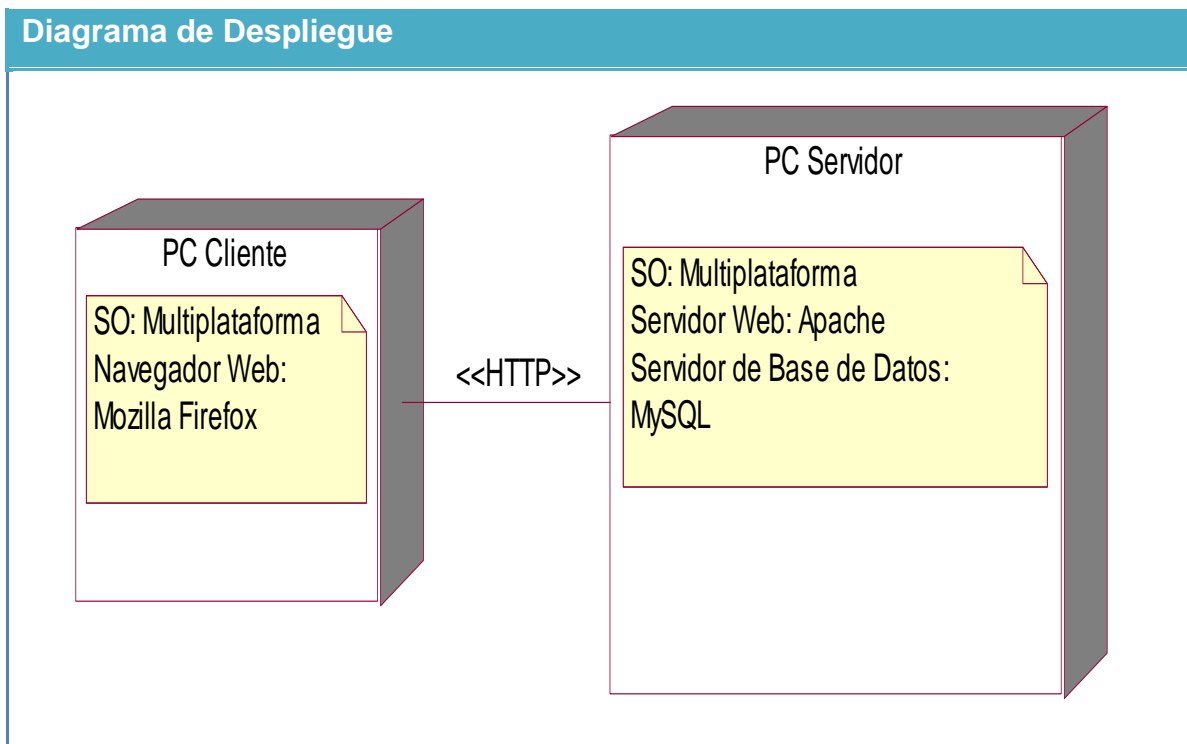


Figura 11: Diagrama de Despliegue.

3.4.2 Diagrama de componentes

Un diagrama de componentes muestra un conjunto de elementos del modelo, se utiliza para modelar la vista estática de un sistema, muestra la organización y dependencias

lógicas entre los componentes del software ya sean bibliotecas, ejecutables o componentes binarios.

La siguiente figura muestra el diagrama de componentes:

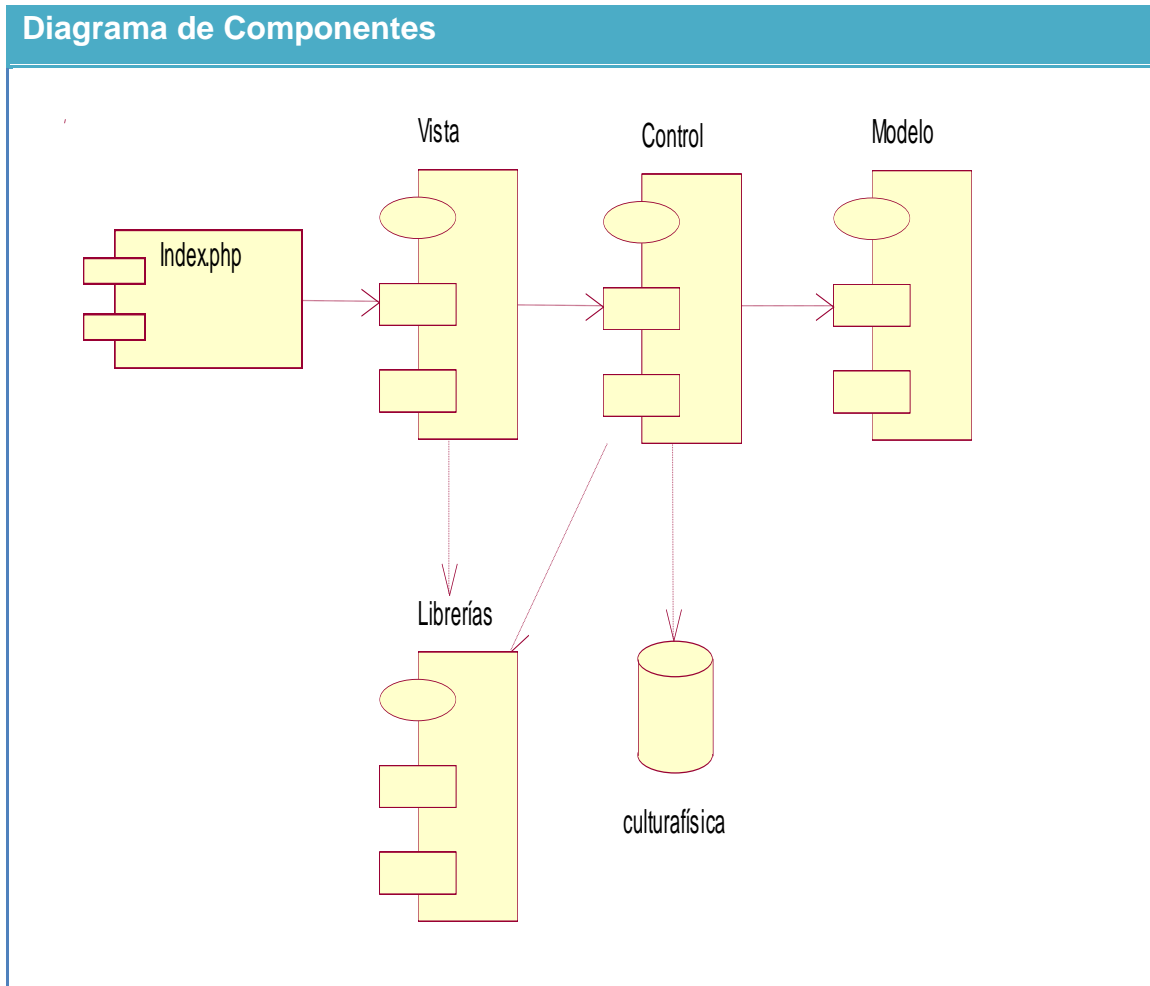


Figura 12: Diagrama de Componentes.

Componente	Descripción
Index.php	Este componente representa el ejecutable de la aplicación.
Vista	Paquete que agrupa los ficheros de las clases de interfaz del usuario.
Control	Paquete que agrupa los ficheros de las clases controladoras que actúan de intermediarias entre las clases del paquete vista y las del modelo.
Modelo	Este paquete contiene los ficheros de las clases de acceso a datos.
cultura física	Representa la base de datos de la aplicación.

Tabla 6: Descripción de los componentes.

Conclusiones

En el presente capítulo se mostraron los resultados de la etapa de implementación del sistema. Se desarrollaron los diagramas de clases, el diseño de la base de datos, el diagrama de despliegue y el de componentes. Además, se describieron los principios de diseño seguidos, específicamente, estándares de la interfaz de usuario, el formato de reportes, el tratamiento de excepciones, la concepción de la ayuda y los estándares de codificación.

Conclusiones

- 1- El estudio de los fundamentos teóricos y metodológicos para la elaboración de un software que contribuya al proceso de enseñanza-aprendizaje de la asignatura Metodología de la Investigación permitió determinar que la metodología RUP es la adecuada para el análisis, diseño e implementación del sistema, lo cual posibilitó la adecuada documentación de la solución propuesta.
- 2- Se diseñó la aplicación web utilizando la metodología RUP con el Lenguaje Unificado de Modelado (UML), además se empleó el patrón modelo-vista-controlador, lo que permite la reutilización y mantenimiento del código.
- 3- Se implementó la aplicación web utilizando el lenguaje de programación PHP y MySQL como Sistema Gestor de Base de Datos con funcionalidades que se ajustan a las necesidades del cliente.

Recomendaciones

- 1- Agregar como nueva funcionalidad a la aplicación, el cálculo del promedio alcanzado por los estudiantes según las notas obtenidas durante la realización de los ejercicios hasta el momento.
- 2- Agregarle al sistema ejercicios de mayor complejidad.

Bibliografía

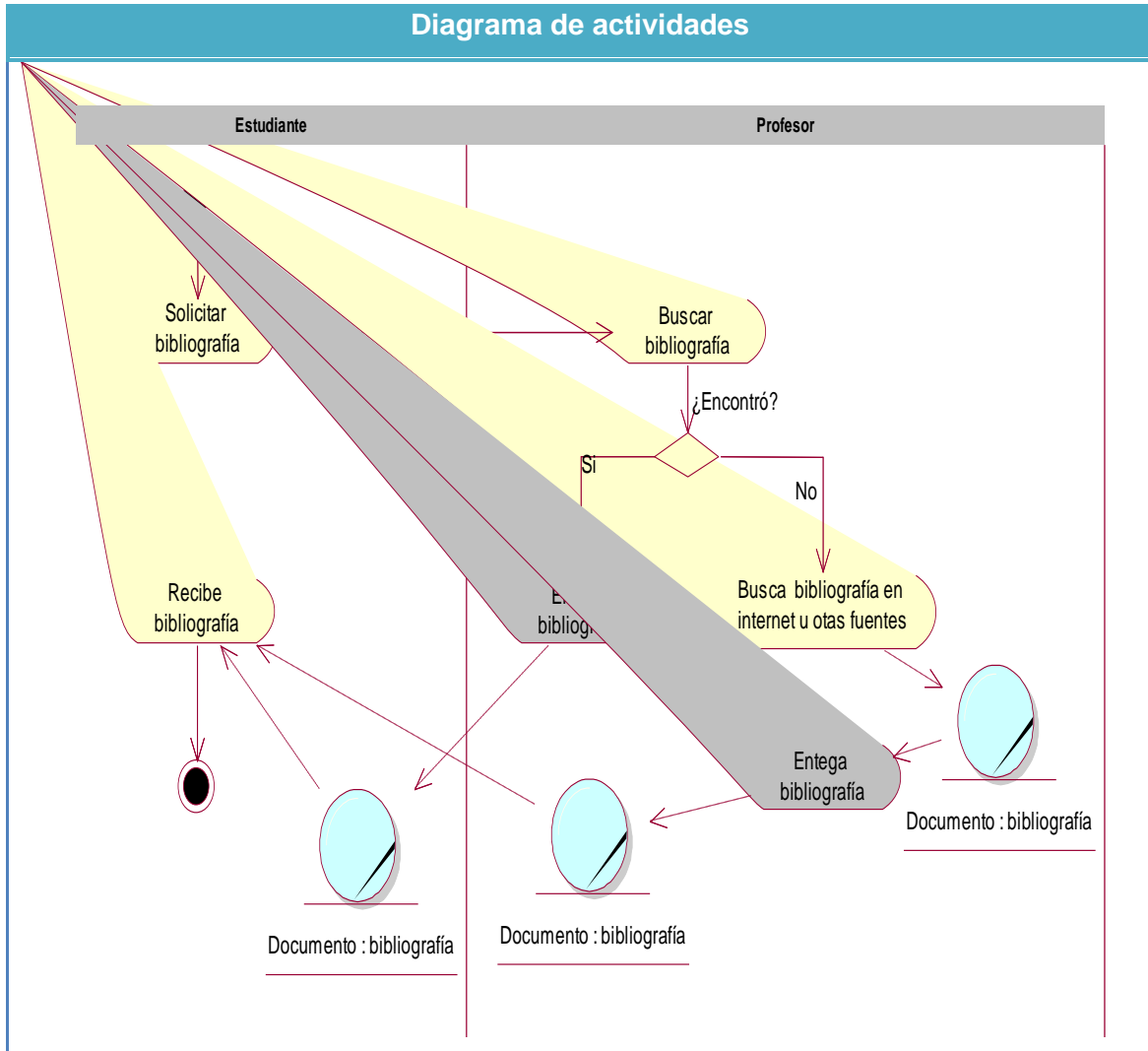
- ✓ *Alejo Machado, O. (2006).* Portal de Psicología de la Universidad de Cienfuegos.
- ✓ *Álvarez, M. A. (25 de Enero de 2010).* DesarrolloWeb.com. Recuperado el 11 de Febrero de 2013 de, <http://www.desarrolloweb.com/articulos/332.php>
- ✓ *Álvarez, M. A. (23 de Noviembre de 2009).* CodeIgniter. Recuperado el 8 de Marzo de 2013, de DesarrolloWeb.com: <http://www.desarrolloweb.com/articulos/codeigniter.html>
- ✓ *Amaro Calderón, Sarah Dámaris, Valverde Rebaza. Jorge Carlos (2007).* Metodologías Ágiles. Universidad Nacional de Trujillo. Perú
- ✓ *Diana Palliotto, Gabriel Romano.* Universidad Nacional de Santiago del Estero – Facultad de Ciencias Exactas y Tecnologías Santiago del Estero, Argentina. <http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>
- ✓ *elWebmaster.com. (9 de Agosto de 2009).* Frameworks PHP recomendados, guía para principiantes. Recuperado el 16 de Abril de 2013, de elWebmaster.com : <http://www.elwebmaster.com/>
- ✓ *García Suárez, L., & López Delgado, L. (2007).* Medios para aprender a aprender.
- ✓ *González, A. (2005).* Modelamiento del negocio. Centro de Estudios de Ingeniería de Sistemas (CEIS).
- ✓ *Gutiérrez, P. P. (Julio de 2008).* Desarrollo ágil con Yupp PHP Framework A/C. Recuperado el 22 de Febrero de 2013. <http://www.slideshare.net/pablitox/yupp-php-framework-presentation>
- ✓ *Hernández 1989.* Recuperado el 25 de enero de 2013, de <http://www.infor.uva.es/~descuder/docencia/pd>
- ✓ *Jacobson, I., Booch, G., & Rumbaugh, J. (2000).* El Proceso Unificado de Desarrollo. Madrid: Addison Wesley.
- ✓ *Jacobson, I., Booch, G., & Rumbaugh, J. (2006).* El Proceso Unificado de Desarrollo de Software. Félix Varela.
- ✓ *Larman, Craig (2004).* UML y Patrones. Introducción al análisis y diseño orientado a objetos. La Habana: Editorial Félix Varela.

- ✓ *Mato García, R. M. (2006)*. Sistema de Base de Datos. La Habana: Félix Varela.
- ✓ *Martínez, P. (2007)*. Patrones. Extraído el 14 Enero, 2013, de Patrón "Modelo-Vista-Controlador":
<http://www.proactiva-calidad.com/java/patrones/mvc.html>
- ✓ *Méndez Cáceres, (2005)*. Generador de Patrones de Navegación Aplicando Web Log Mining bibliografía 62 en Cliente/Servidor recuperado el 5 de febrero de 2013, en www.tamps.cinvestav.mx/~vjsosa/thesis.html
- ✓ *Popkin Software and System. (2005)*. Recuperado el 10 de abril de 2013, de Modelado de sistemas con UML:
<http://es.tldp.org/tutoriales/doc-modelado-sistemas-UML-multiple-tml/c124.html>
- ✓ *Pérez Valdés, D. (7 de noviembre de 2007)*. Los diferentes lenguajes de programación para la web. Extraído el 14 de Enero, 2013, de <http://www.tufuncion.com/diferentes-lenguajes-programacion>
- ✓ *Rodríguez Febles, J. (2005)*. Sistema Informático para la Gestión Integral de Comedores (SISCOMED). CUJAE. (UH): Trabajo de Diploma.
- ✓ *Soto, L. (4 de Marzo de 2006)*. Especificación de requerimientos. Obtenido de <http://mitecnologico.com/Main/EspecificacionesDeRequerimientos>
- ✓ *Tadei, L. (22 de Septiembre de 2009)*. Frameworks en PHP y otros lenguajes también. Recuperado el 22 de Febrero de 2013, de PEGASUS Blog:
<http://blog.pegasusnet.com.ar/2009/09/frameworks-en-php-y-otros-lenguajes.html>
- ✓ *Truel, A. (2006)*. Recuperado el 5 de febrero de 2013, de Introducción a la arquitectura de capas:
<http://www ldc.usb.ve/~teruel/ci3715/clases/arqCapas.html>
- ✓ *Valdés, D. P. (31 de Julio de 2007)*. Los Frameworks de PHP agilizan tu trabajo. Recuperado el 16 de abril de 2013, de Maestros del Web.:
<http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>
- ✓ *Webmaster. (2009)*. HTML, lenguaje de Hipertexto. Extraído el 11 Febrero, 2013, de Mundo Webmaster:
<http://mundowebmaster.wordpress.com/2009/02/21/html-lenguaje-de-hipertexto/>

- ✓ *Rocha, D. (2008)*. Sistema de Gestión de Indicadores de consumo de combustible en el transporte automotor de pasajeros de la Empresa Provincial de Cienfuegos. Universidad de Cienfuegos Carlos Rafael Rodríguez., Cienfuegos, Cuba.
- ✓ *Vedora Willock, V. (2006)*. Sistema de gestión de información estudiantes becarios extranjeros. Cienfuegos, Cuba.

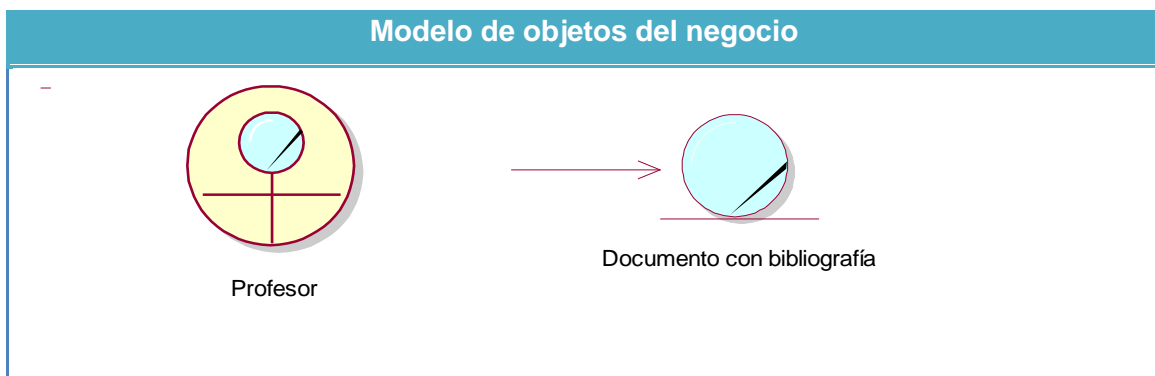
Anexos

Anexo 1: Diagrama de actividades.



Anexo 1.1: Diagrama de actividades.

Anexo 2: Modelo de objetos.



Anexo 2.1: Modelo de objetos del negocio.

Anexo 3: Descripción de los casos de uso: Paquete Seguridad.

CU# 1 Autenticarse	
Actores	Administrador (inicia)
Propósito	Entrar al software.
Resumen: El caso de uso comienza cuando el administrador necesita entrar al software, introduce su nombre de usuario y contraseña, si está registrado y sus datos son correctos tiene acceso al software con todos los permisos, en caso contrario se le informa a través de un mensaje. El caso de uso finaliza cuando se ejecuta la operación antes descrita.	
Referencias	RF- 1
Prototipo	Anexo 9

[Anexo 3.1: Descripción del caso de uso: Autenticarse.](#)

CU# 2 Cambiar contraseña	
Actores	Administrador (inicia)
Propósito	Mantener la seguridad del sistema y la confiabilidad de los datos al ser la contraseña del administrador únicamente de su conocimiento.
Resumen: El caso de uso comienza cuando el administrador solicita el cambio de su contraseña, introduce los datos pertinentes y de ser correctos el sistema guarda la nueva contraseña, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.	
Referencias	RF- 2
Prototipo	

[Anexo 3.2: Descripción del caso de uso: Cambiar contraseña.](#)

Anexo 4: Descripción de los casos de uso: Paquete Ejercicios.

CU# 3 Resolver ejercicio de Verdadero o Falso	
Actores	Estudiante (inicia)
Propósito	Resolver los ejercicios propuestos por el profesor.
Resumen: El caso de uso comienza cuando el estudiante decide entrar al sitio para resolver los ejercicios, selecciona el menú cuestionario y una vez dentro de este el ejercicio que desea resolver, luego responde las preguntas y al presionar el botón enviar cuestionario el sistema emite la evaluación obtenida y las respuestas correctas. El caso de uso finaliza cuando se ejecutan las acciones antes descritas.	
Referencias	RF- 22
Prototipo	

[Anexo 4.1: Descripción del caso de uso: Resolver ejercicio de verdadero o falso.](#)

CU# 4 Resolver ejercicio de Seleccionar	
Actores	Estudiante (inicia)
Propósito	Resolver los ejercicios propuestos por el profesor.
Resumen: El caso de uso comienza cuando el estudiante decide entrar al sitio para resolver los ejercicios, selecciona el menú cuestionario y una vez dentro de este el ejercicio que desea resolver, luego responde las preguntas y al presionar el botón enviar cuestionario el sistema emite la evaluación obtenida y las respuestas correctas. El caso de uso finaliza cuando se ejecutan las acciones antes descritas.	
Referencias	RF- 20
Prototipo	

Anexo 4.2: Descripción del caso de uso: Resolver ejercicio de Seleccionar.

CU# 5 Resolver ejercicio de Completar	
Actores	Estudiante (inicia)
Propósito	Resolver los ejercicios propuestos por el profesor.
Resumen: El caso de uso comienza cuando el estudiante decide entrar al sitio para resolver los ejercicios, selecciona el menú cuestionario y una vez dentro de este el ejercicio que desea resolver, luego responde las preguntas y al presionar el botón enviar cuestionario el sistema emite la evaluación obtenida y las respuestas correctas. El caso de uso finaliza cuando se ejecutan las acciones antes descritas.	
Referencias	RF- 21
Prototipo	

Anexo 4.3: Descripción del caso de uso: Resolver ejercicio de Completar.

Anexo 5: Descripción de los casos de uso: Paquete Gestión.

CU# 6 Gestionar libros	
Actores	Administrador (inicia)
Propósito	Insertar un nuevo libro, modificar o eliminar alguno ya existente.
Resumen: El caso de uso se inicia cuando el administrador necesita insertar un nuevo libro, modificar o eliminar uno ya existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.	
Referencias	RF- 3
Prototipo	

Anexo 5.1: Descripción del caso de uso: Gestionar libros.

CU# 7 Gestionar tesis	
Actores	Administrador (inicia)
Propósito	Insertar una tesis, modificar o eliminar alguna ya existente.
Resumen: El caso de uso se inicia cuando el administrador necesita insertar una tesis más actual en el sitio, modificar o eliminar alguna de las que se encuentran a disposición de los estudiantes, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.	
Referencias	RF- 4
Prototipo	

Anexo 5.2: Descripción del caso de uso: Gestionar tesis.

CU# 8 Gestionar documentos actuales	
Actores	Administrador (inicia)
Propósito	Insertar un nuevo documento de interés para los estudiantes, modificar o eliminar alguno ya existente.
Resumen: El caso de uso se inicia cuando el administrador necesita insertar en el sitio un nuevo documento de interés para los estudiantes, modificar o eliminar alguno de los que se encuentran a disposición de estos, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.	
Referencias	RF- 5
Prototipo	Anexo12

Anexo 5.3: Descripción del caso de uso: Gestionar documentos actuales.

CU# 9 Gestionar concepto	
Actores	Administrador (inicia)
Propósito	Insertar un nuevo concepto de interés para los estudiantes, modificar o eliminar alguno ya existente.
Resumen: El caso de uso se inicia cuando el administrador necesita insertar un nuevo concepto de interés para los estudiantes, modificar o eliminar alguno de los que se encuentran a disposición de estos, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.	
Referencias	RF- 6
Prototipo	Anexo 11

Anexo 5.4: Descripción del caso de uso: Gestionar concepto.

CU# 10 Gestionar ejemplo	
Actores	Administrador (inicia)
Propósito	Insertar un nuevo ejemplo asociado a un concepto, modificar o eliminar alguno ya existente.
Resumen: El caso de uso se inicia cuando el administrador necesita insertar un nuevo ejemplo a un concepto determinado, modificar o eliminar alguno de los que ya existen, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.	
Referencias	RF- 7
Prototipo	

Anexo 5.5: Descripción del caso de uso: Gestionar ejemplo.

CU# 11 Gestionar estructura para trabajos investigativos	
Actores	Administrador (inicia)
Propósito	Insertar una nueva estructura para trabajos investigativos, modificar o eliminar alguna ya existente.
Resumen: El caso de uso se inicia cuando el administrador necesita insertar una nueva estructura para trabajos investigativos, modificar o eliminar alguna ya existente, luego introduce los datos pertinentes, de ser correctos el sistema ejecuta la acción, en caso contrario emite un mensaje de error. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.	
Referencias	RF- 8
Prototipo	

Anexo 5.6: Descripción del caso de uso: Gestionar estructura para trabajos investigativos.

CU# 12 Gestionar curiosidades	
Actores	Administrador (inicia)
Propósito	Insertar, actualizar y eliminar una curiosidad.
Resumen: El caso de uso se inicia cuando el administrador accede al sistema para insertar, modificar o eliminar una curiosidad. Para modificar los datos de una curiosidad se selecciona y se actualiza el campo deseado. Para eliminar se debe seleccionar la curiosidad que se desea borrar. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.	
Referencias	RF- 9
Prototipo	

Anexo 5.7: Descripción del caso de uso: Gestionar curiosidad.

CU# 13 Gestionar ejercicios de seleccionar	
Actores	Administrador (inicia)
Propósito	Insertar, actualizar y eliminar un ejercicio de selección.
Resumen: El caso de uso se inicia cuando el administrador accede al sistema para insertar, modificar o eliminar un ejercicio de selección. Para modificar un ejercicio de completar se muestran los existentes, se marca el que desea modificar y realiza los cambios. Cuando el administrador desea eliminar un ejercicio de selección se muestran los ejercicios que existen, marca el que desea eliminar y pulsa el botón eliminar. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.	
Referencias	RF- 10
Prototipo	

Anexo 5.8: Descripción del caso de uso: Gestionar ejercicio de seleccionar.

CU# 14 Gestionar ejercicios de completar	
Actores	Administrador (inicia)
Propósito	Insertar, actualizar y eliminar un ejercicio de completar.
Resumen: El caso de uso se inicia cuando el administrador accede al sistema para insertar, modificar o eliminar un ejercicio de completar. Para modificar un ejercicio de completar se muestran los existentes, marca el que desea modificar y realiza los cambios. Cuando el administrador desea eliminar un ejercicio de selección se muestran los ejercicios que existen, marca el que desea eliminar y pulsa el botón eliminar. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.	
Referencias	RF- 11
Prototipo	

Anexo 5.9: Descripción del caso de uso: Gestionar ejercicio de completar.

CU# 15 Gestionar ejercicios de verdadero o falso	
Actores	Administrador (inicia)
Propósito	Insertar, actualizar y eliminar un ejercicio de verdadero o falso.
Resumen: El caso de uso se inicia cuando el administrador accede al sistema para insertar, modificar o eliminar un ejercicio de verdadero o falso. Para modificar un ejercicio de verdadero o falso se muestran los existentes, marca el que desea modificar y realiza los cambios. Cuando el administrador desea eliminar un ejercicio de verdadero o falso se muestran los ejercicios que existen, marca el que desea eliminar y pulsa el botón eliminar. El caso de uso finaliza cuando se ejecuta una de	

las operaciones antes descritas.	
Referencias	RF- 12
Prototipo	Anexo 13

Anexo 5.10: Descripción del caso de uso: Gestionar ejercicio de verdadero o falso.

CU# 16	Gestionar usuario
Actores	Administrador (inicia)
Propósito	Insertar, actualizar y eliminar un usuario.
Resumen: El caso de uso se inicia cuando el administrador accede al sistema para insertar, modificar o eliminar un usuario. Para modificar un usuario se muestran los existentes, marca el que desea modificar y realiza los cambios. Cuando el administrador desea eliminar un usuario se muestran los que existen, marca el que desea eliminar y pulsa el botón eliminar. El caso de uso finaliza cuando se ejecuta una de las operaciones antes descritas.	
Referencias	RF- 13
Prototipo	

Anexo 6: Descripción de los casos de uso: Paquete Reportes.

CU# 17	Mostrar libros
Actores	Estudiante (inicia)
Propósito	Permitir a los estudiantes consultar los libros de su interés.
Resumen: El caso de uso se inicia cuando el estudiante accede al menú principal del Sitio y selecciona la opción Libros, una vez allí selecciona el que desea consultar. El caso de uso finaliza cuando el estudiante recibe la información.	
Referencias	RF-14
Prototipo	Anexo 15

Anexo 6.1: Descripción del caso de uso: Mostrar libros.

CU# 18	Mostrar tesis
Actores	Estudiante (inicia)
Propósito	Permitir a los estudiantes consultar las tesis de su interés.
Resumen: El caso de uso se inicia cuando el estudiante accede al menú principal del Sitio y selecciona la opción Tesis, una vez allí selecciona la que desea consultar. El caso de uso finaliza cuando el estudiante recibe la información.	
Referencias	RF-15
Prototipo	

Anexo 6.2: Descripción del caso de uso: Mostrar tesis.

CU# 19 Mostrar documentos actuales	
Actores	Estudiante (inicia)
Propósito	Permitir a los estudiantes consultar los documentos de su interés.
Resumen: El caso de uso se inicia cuando el estudiante accede al menú principal del Sitio y selecciona la opción Documentos, una vez allí selecciona el que desea consultar. El caso de uso finaliza cuando el estudiante recibe la información.	
Referencias	RF-16
Prototipo	

Anexo 6.3: Descripción del caso de uso: Mostrar documentos actuales.

CU# 20 Mostrar conceptos	
Actores	Estudiante (inicia)
Propósito	Permitir a los estudiantes consultar los conceptos de su interés.
Resumen: El caso de uso se inicia cuando el estudiante accede al menú principal del Sitio y selecciona la opción Conceptos, una vez allí selecciona el que desea consultar. El caso de uso finaliza cuando el estudiante recibe la información.	
Referencias	RF-17
Prototipo	Anexo 24

Anexo 6.4: Descripción del caso de uso: Mostrar conceptos.

CU# 21 Mostrar ejemplos	
Actores	Estudiante (inicia)
Propósito	Permitir a los estudiantes consultar los ejemplos de su interés.
Resumen: El caso de uso se inicia cuando el estudiante accede al menú principal del Sitio y selecciona la opción Conceptos, una vez allí selecciona la opción ejemplos y consulta el de su interés. El caso de uso finaliza cuando el estudiante recibe la información.	
Referencias	RF-18
Prototipo	

Anexo 6.5: Descripción del caso de uso: Mostrar ejemplos.

CU# 22 Mostrar estructura investigativa	
Actores	Estudiante (inicia)
Propósito	Permitir a los estudiantes consultar la estructura investigativa de su interés.
Resumen: El caso de uso se inicia cuando el estudiante accede al menú principal del Sitio y selecciona la opción Investigaciones, una vez allí selecciona la que desea	

consultar. El caso de uso finaliza cuando el estudiante recibe la información.	
Referencias	RF-19
Prototipo	

Anexo 6.6: Descripción del caso de uso: Mostrar estructura investigativa.

CU# 23	Mostrar curiosidades
Actores	Estudiante (inicia)
Propósito	Permitir a los estudiantes consultar las curiosidades de su interés.
Resumen: El caso de uso se inicia cuando el estudiante accede al menú principal del Sitio y selecciona la opción Curiosidades, una vez allí selecciona la que desea consultar. El caso de uso finaliza cuando el estudiante recibe la información.	
Referencias	RF-23
Prototipo	

Anexo 6.7: Descripción del caso de uso: Mostrar curiosidades.

CU# 24	Evaluar respuestas
Actores	Estudiante (inicia)
Propósito	Permitir a los estudiantes obtener una evaluación.
Resumen: El caso de uso se inicia cuando los estudiantes acceden al menú Cuestionario y una vez que responden los ejercicios presionan el botón enviar cuestionario, luego el sistema le muestra su evaluación. El caso de uso finaliza cuando el estudiante recibe la información.	
Referencias	RF-24
Prototipo	

Anexo 6.8: Descripción del caso de uso: Evaluar respuestas.

CU# 25	Mostrar respuesta correcta
Actores	Estudiante (inicia)
Propósito	Permitir a los estudiantes obtener la respuesta correcta.
Resumen: El caso de uso se inicia cuando los estudiantes acceden al menú Cuestionario y una vez que responden los ejercicios presionan el botón enviar cuestionario, luego el sistema le muestra la evaluación obtenida con la respuesta correcta. El caso de uso finaliza cuando el estudiante recibe la información.	
Referencias	RF-25
Prototipo	

Anexo 6.9: Descripción del caso de uso: Mostrar respuesta correcta.

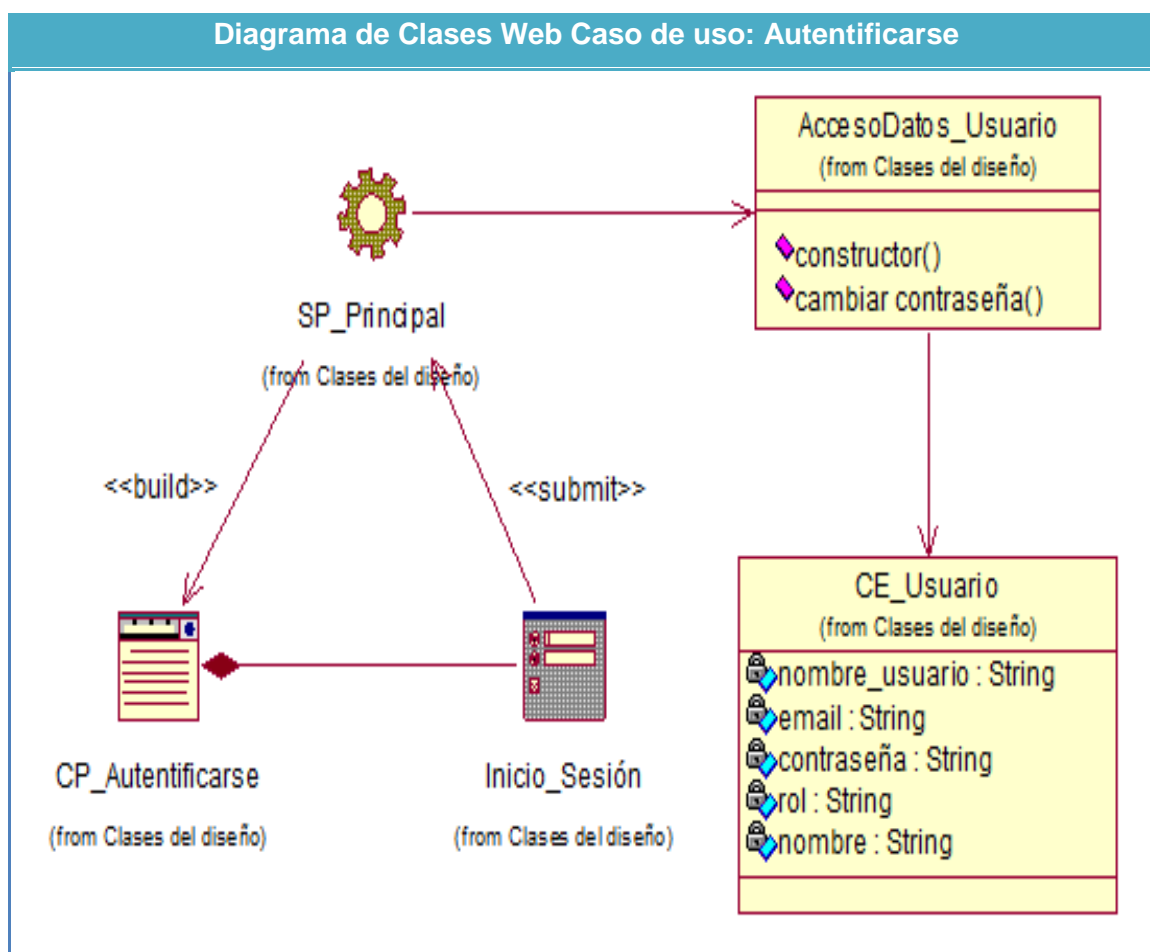
CU# 26	Descargar ficheros
Actores	Estudiante (inicia)

Propósito	Permitir a los estudiantes obtener de forma física los documentos o bibliografía que necesite.
Resumen:	El caso de uso se inicia cuando el estudiante accede al menú donde se encuentra el fichero a descargar, una vez allí selecciona Descargar y luego la ruta en la que desea que se guarde el fichero. El caso de uso finaliza cuando el estudiante recibe la información.
Referencias	RF-26
Prototipo	

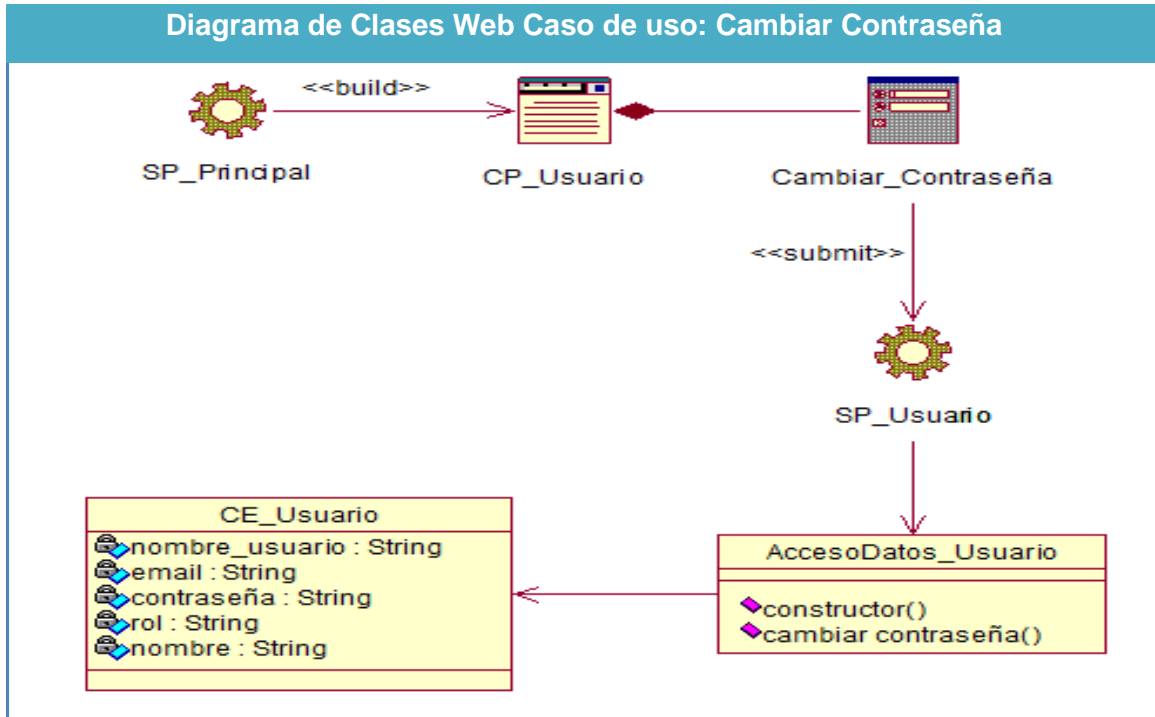
Anexo 6.10: Descripción del caso de uso: Descargar ficheros.

Anexo 7: Diagramas de Clases del Diseño.

Diagramas de Clases Web: Paquete seguridad.

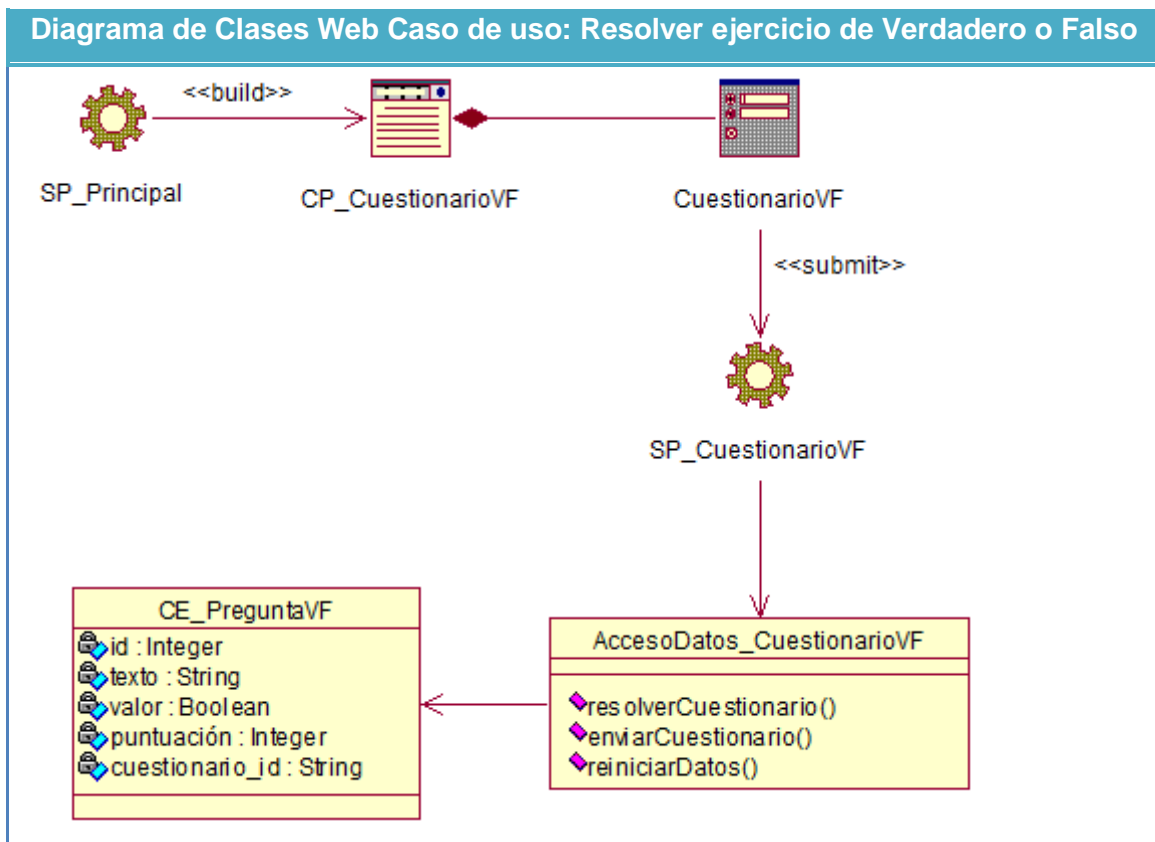


Anexo 7.1: Diagrama de clases web caso de uso: Autenticarse.

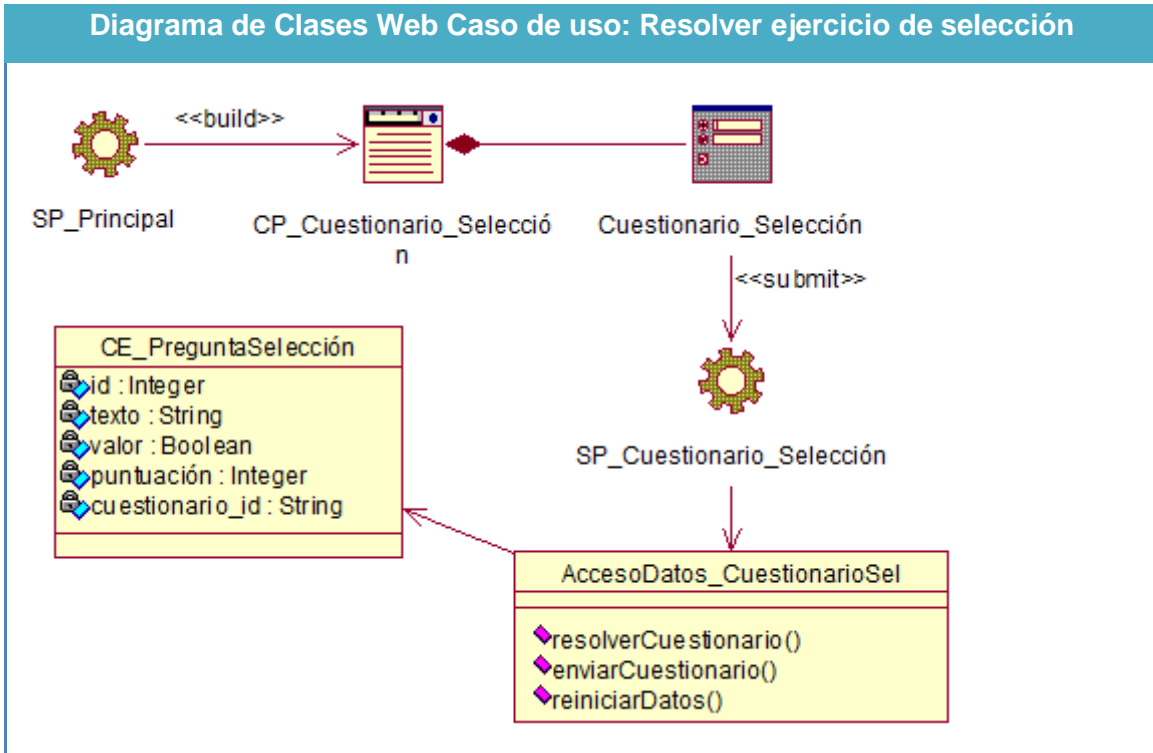


Anexo 7.2: Diagrama de clases web caso de uso: Cambiar contraseña.

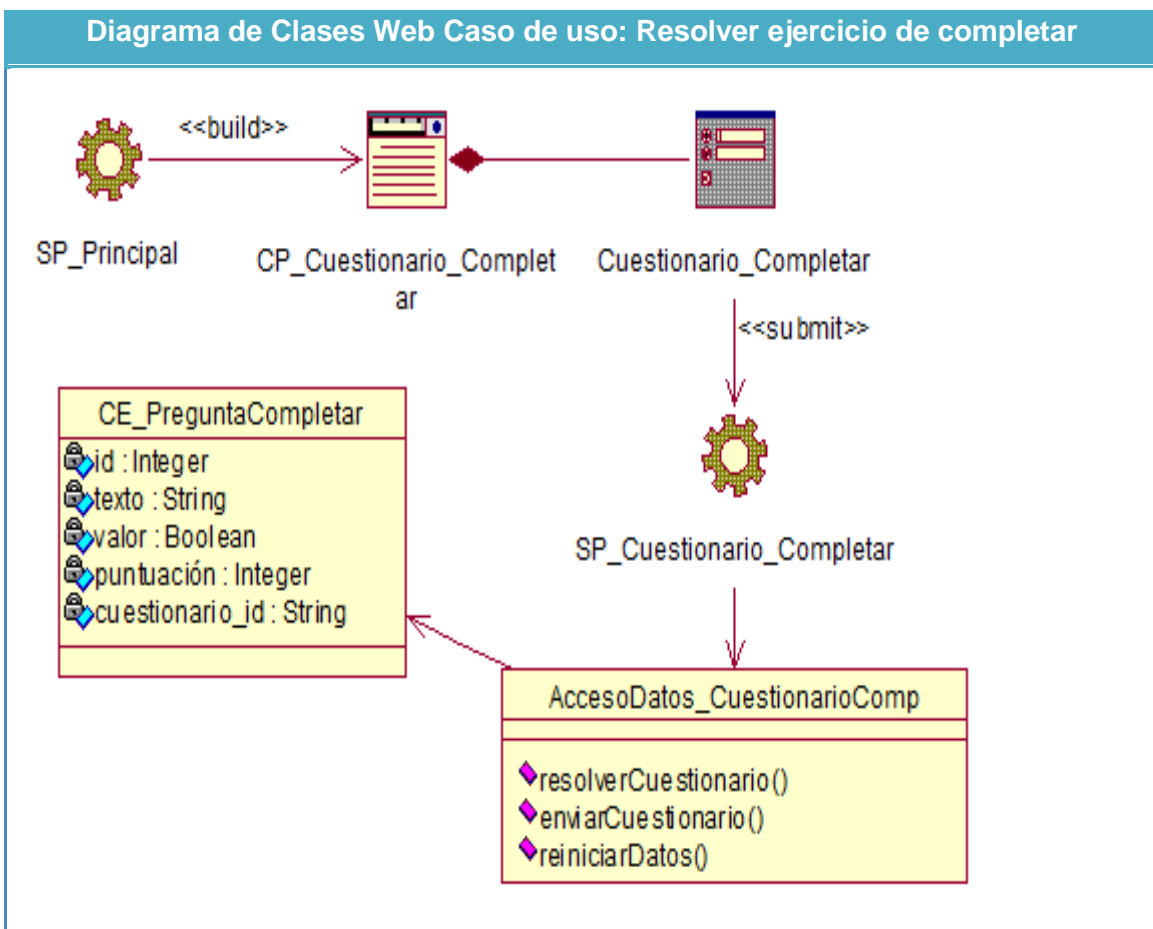
Diagramas de Clases Web: Paquete Ejercicios.



Anexo 7.3: Diagrama de clases web caso de uso: Resolver ejercicio de verdadero o falso.

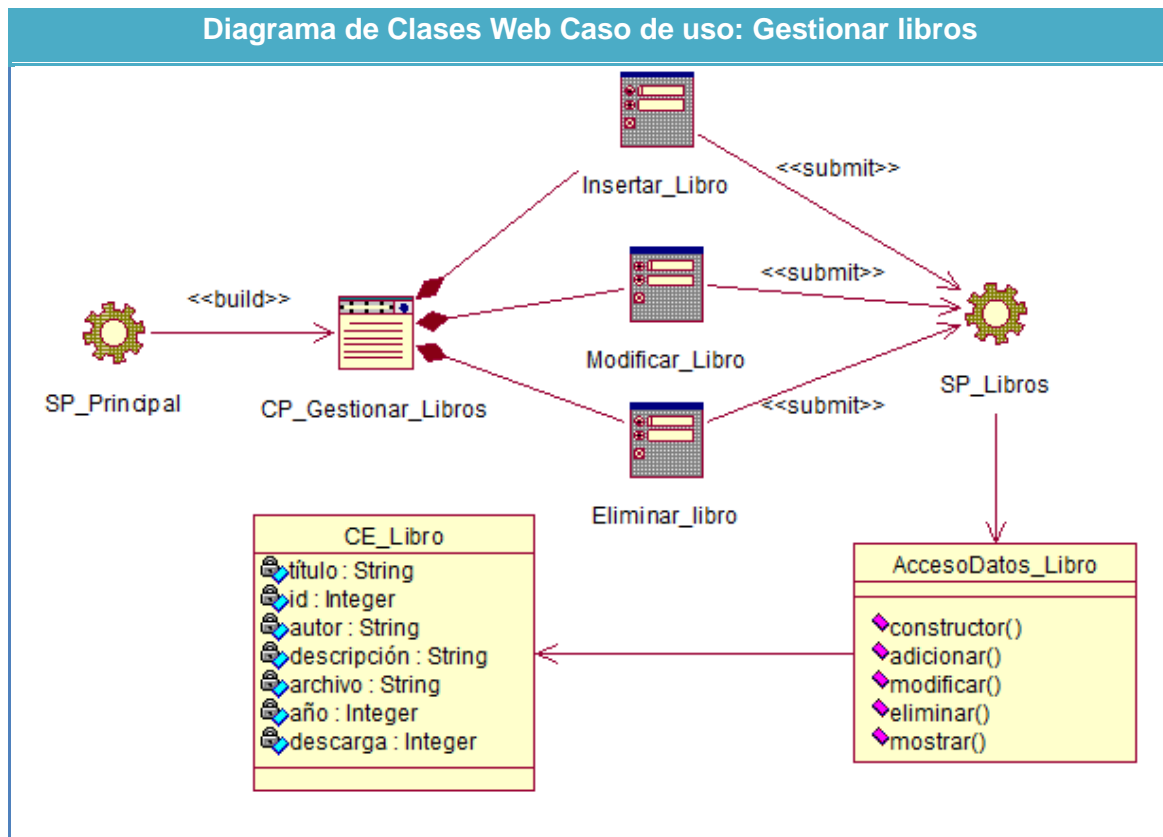


Anexo 7.4: Diagrama de clases web caso de uso: Resolver ejercicio de selección

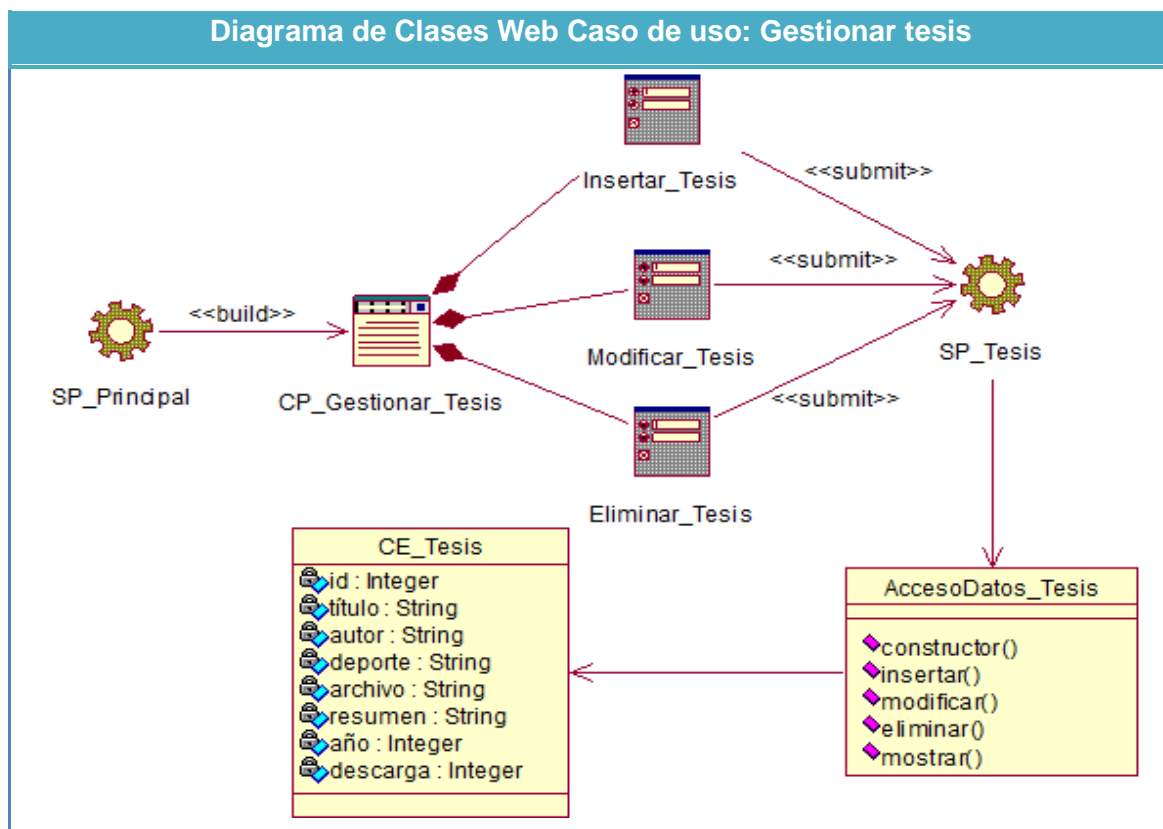


Anexo 7.5: Diagrama de clases web caso de uso: Resolver ejercicio de completar

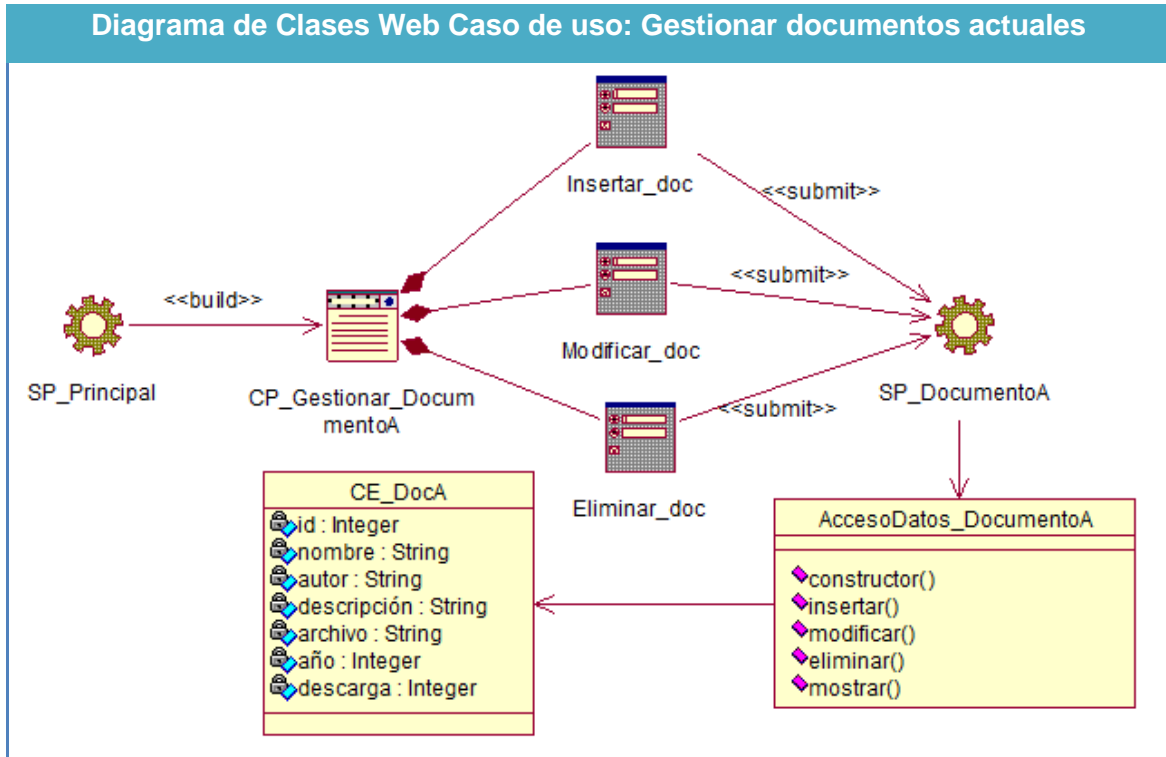
Diagramas de Clases Web: Paquete Gestión.



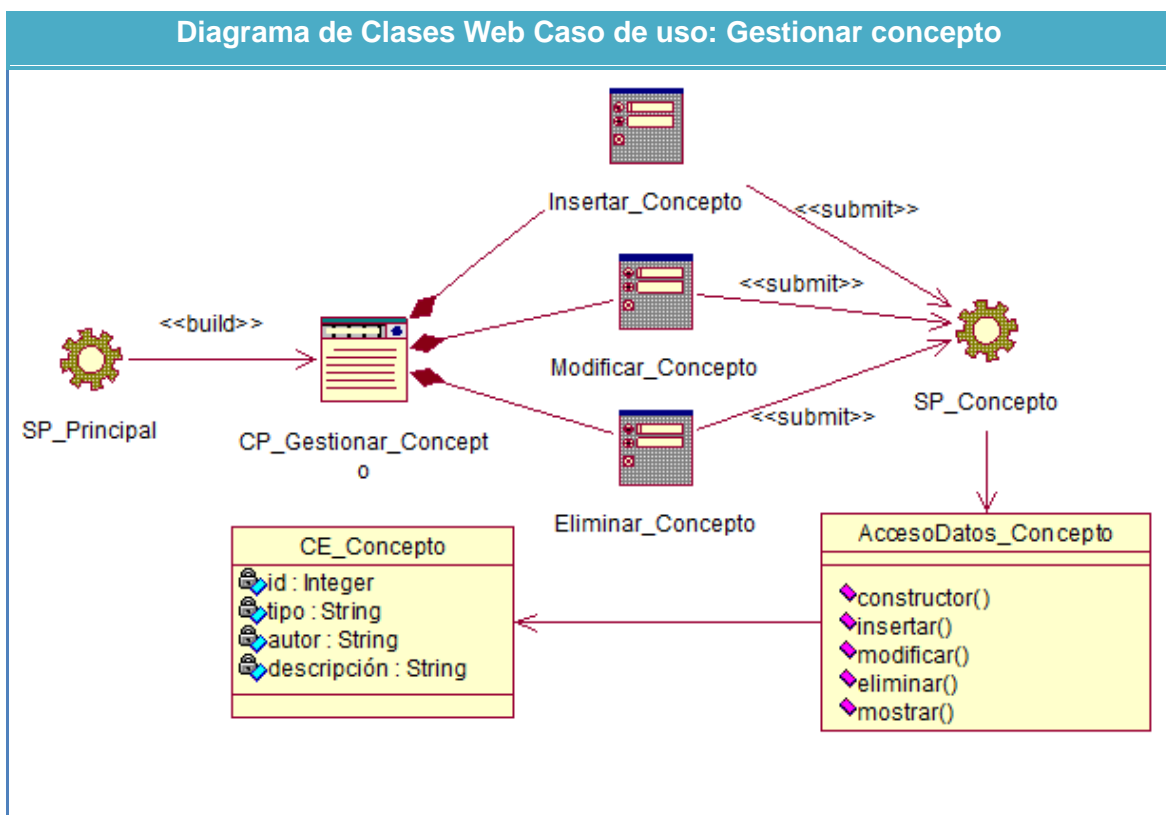
Anexo 7.6: Diagrama de clases web caso de uso: Gestionar libros.



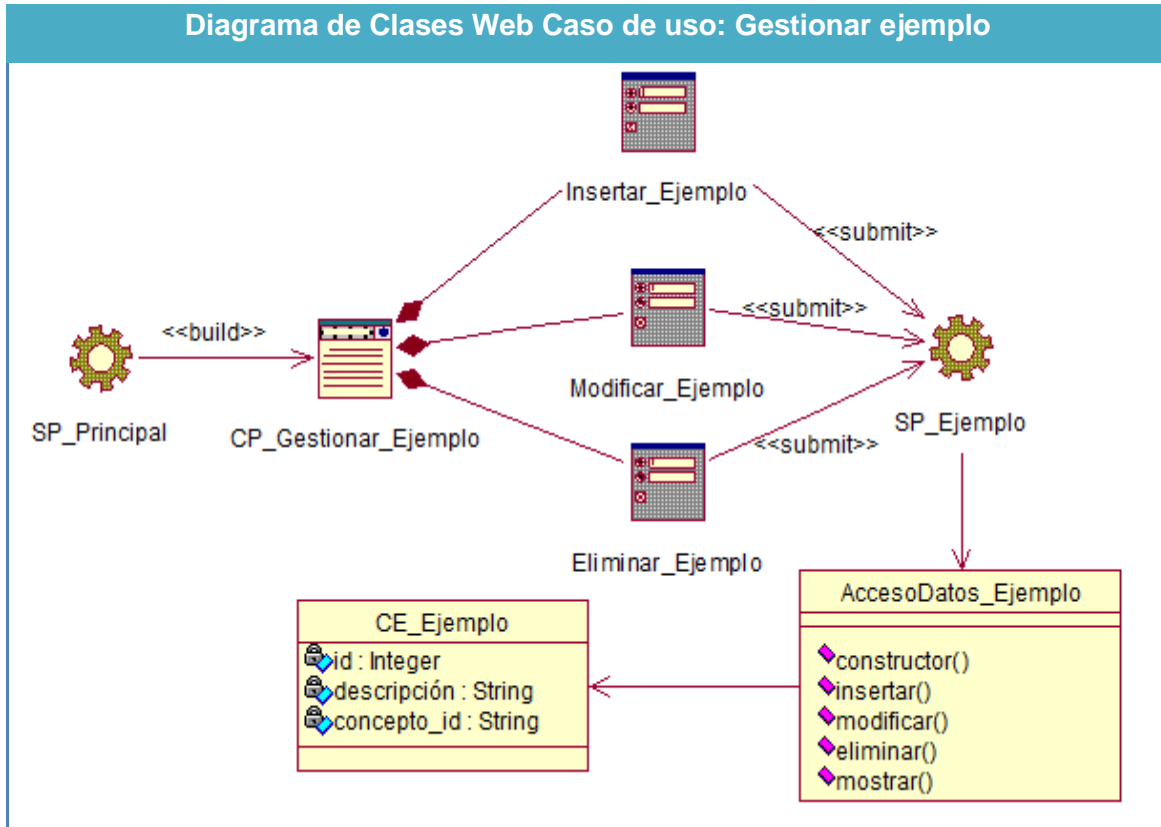
Anexo 7.7: Diagrama de clases web caso de uso: gestionar tesis.



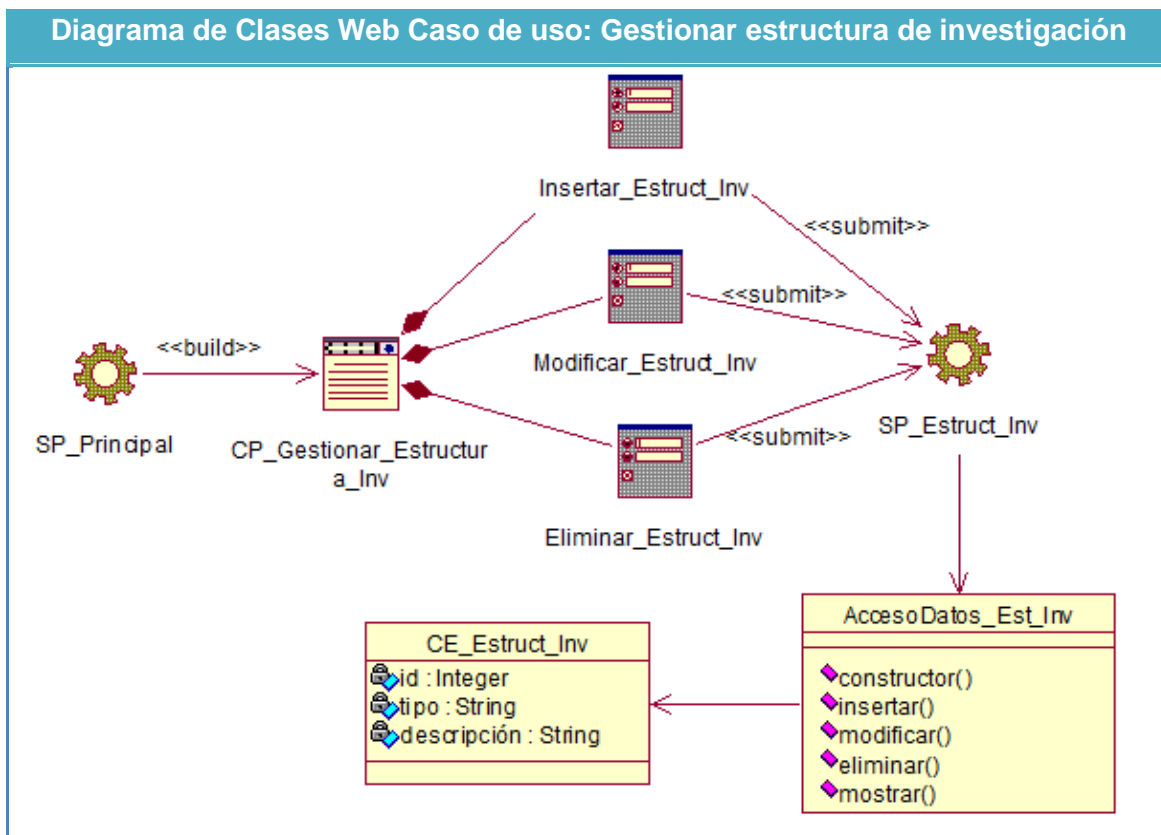
Anexo 7.8: Diagrama de clases web caso de uso: Gestionar documentos actuales.



Anexo 7.9: Diagrama de clases web caso de uso: Gestionar concepto.

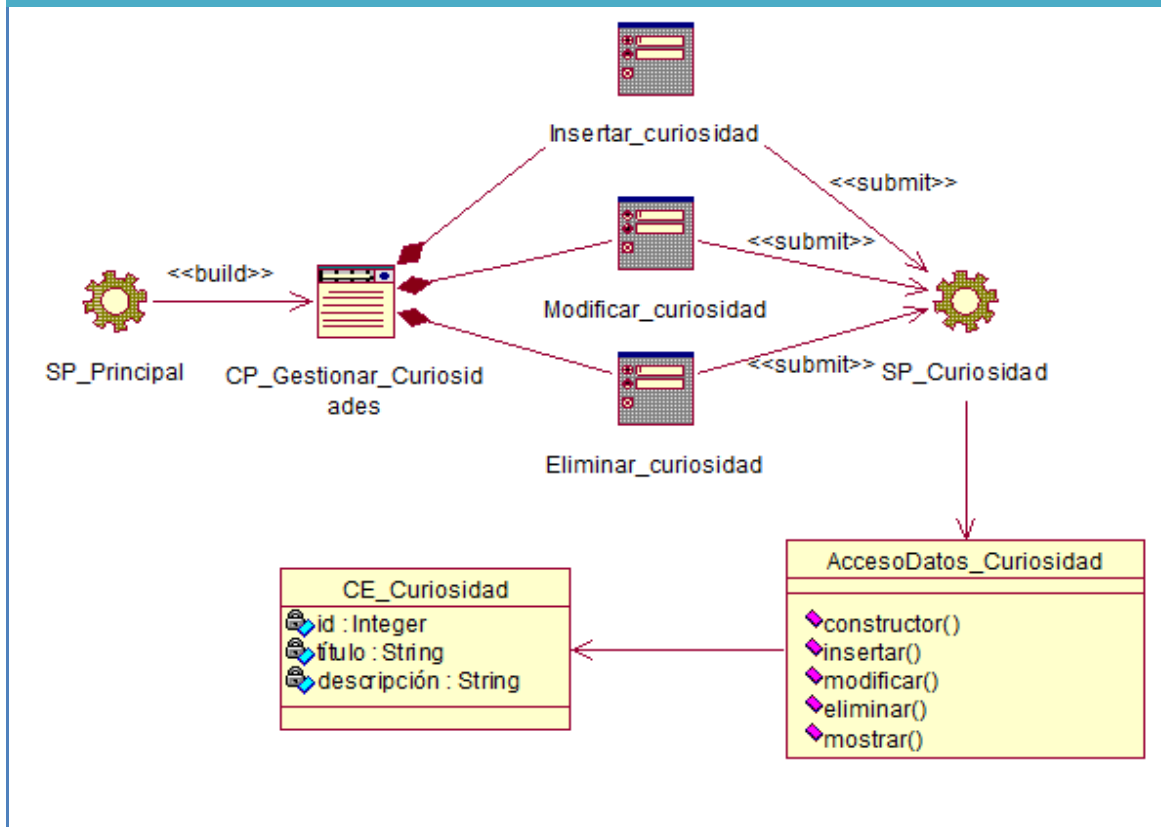


Anexo 7.10: Diagrama de clases web caso de uso: Gestionar ejemplo.



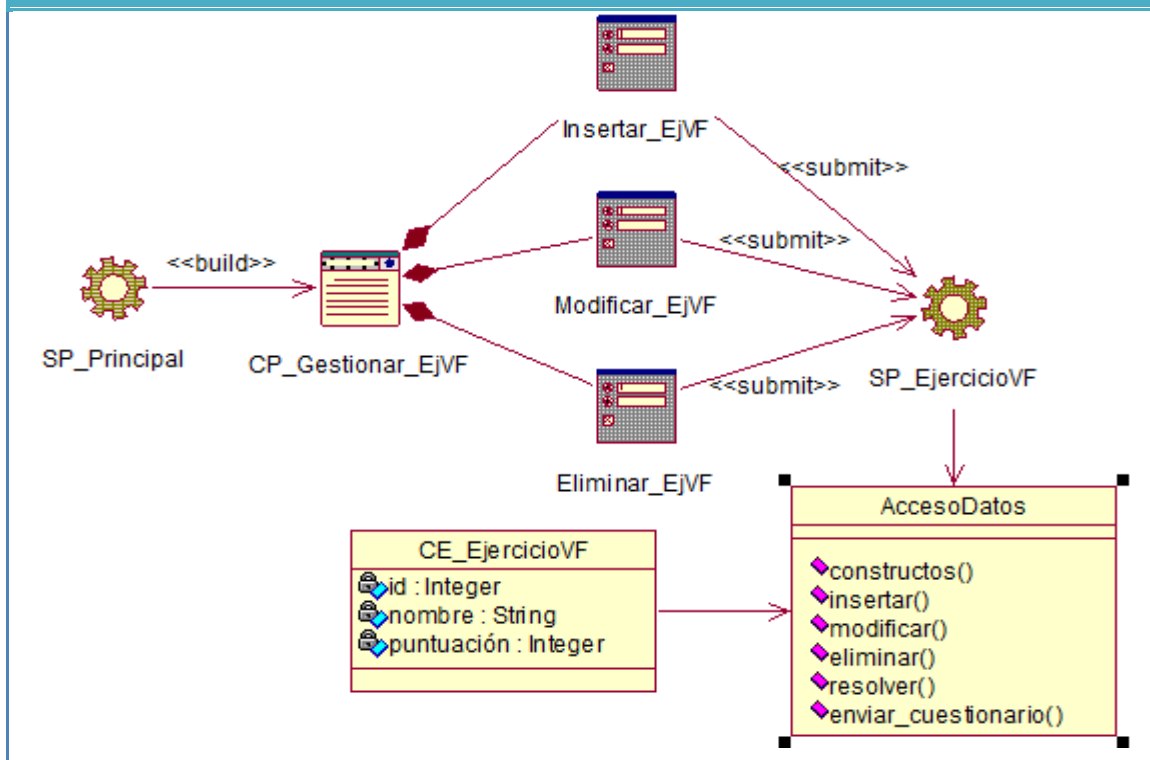
Anexo 7.11: Diagrama de clases web caso de uso: Gestionar estructura de investigación

Diagrama de Clases Web Caso de uso: Gestionar curiosidades

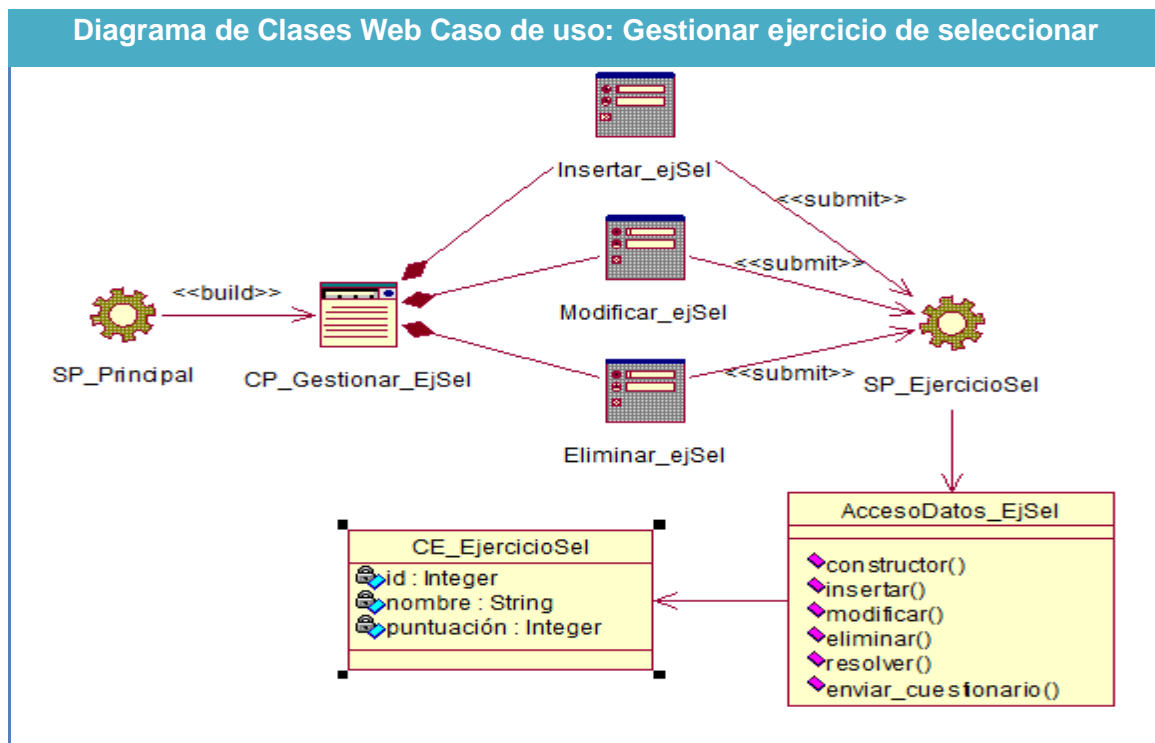


Anexo 7.12: Diagrama de clases web caso de uso: Gestionar curiosidades.

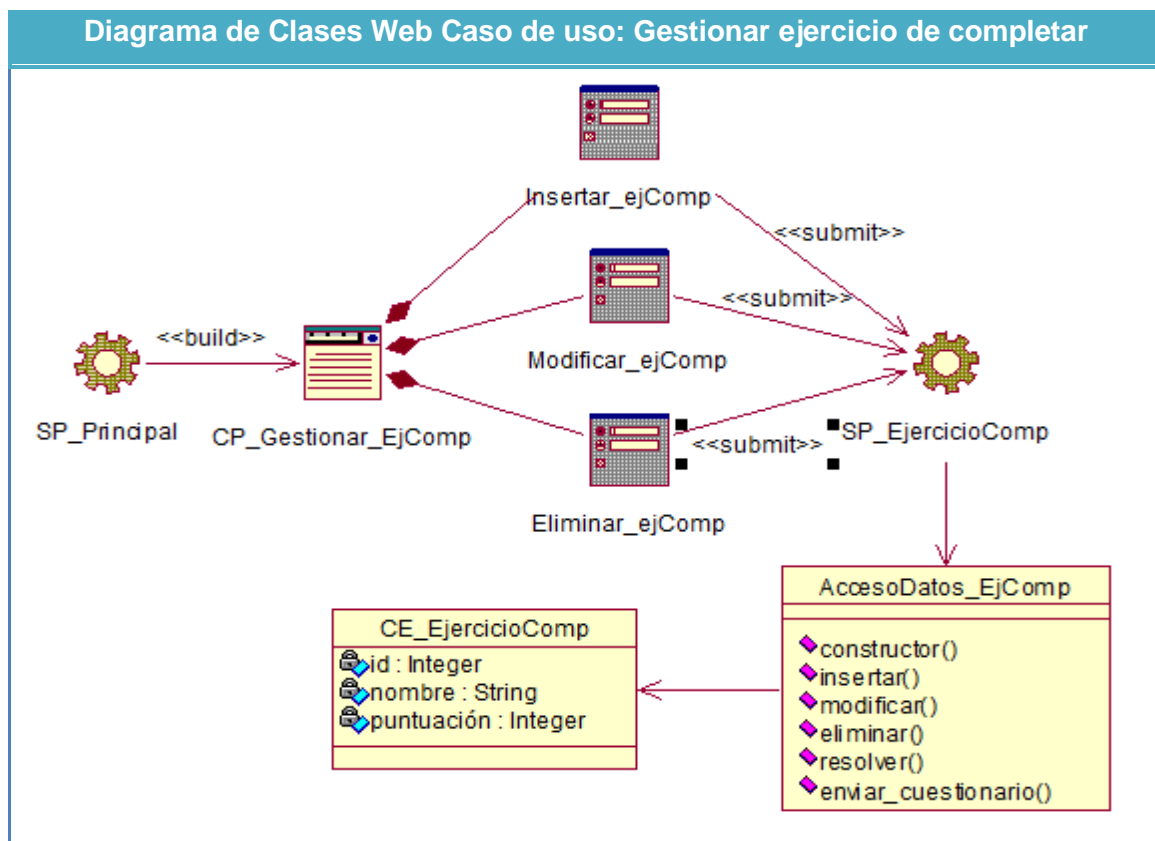
Diagrama de Clases Web Caso de uso: Gestionar ejercicio de verdadero y falso



Anexo 7.13: Diagrama de clases web caso de uso: Gestionar ejercicio de verdadero y falso.

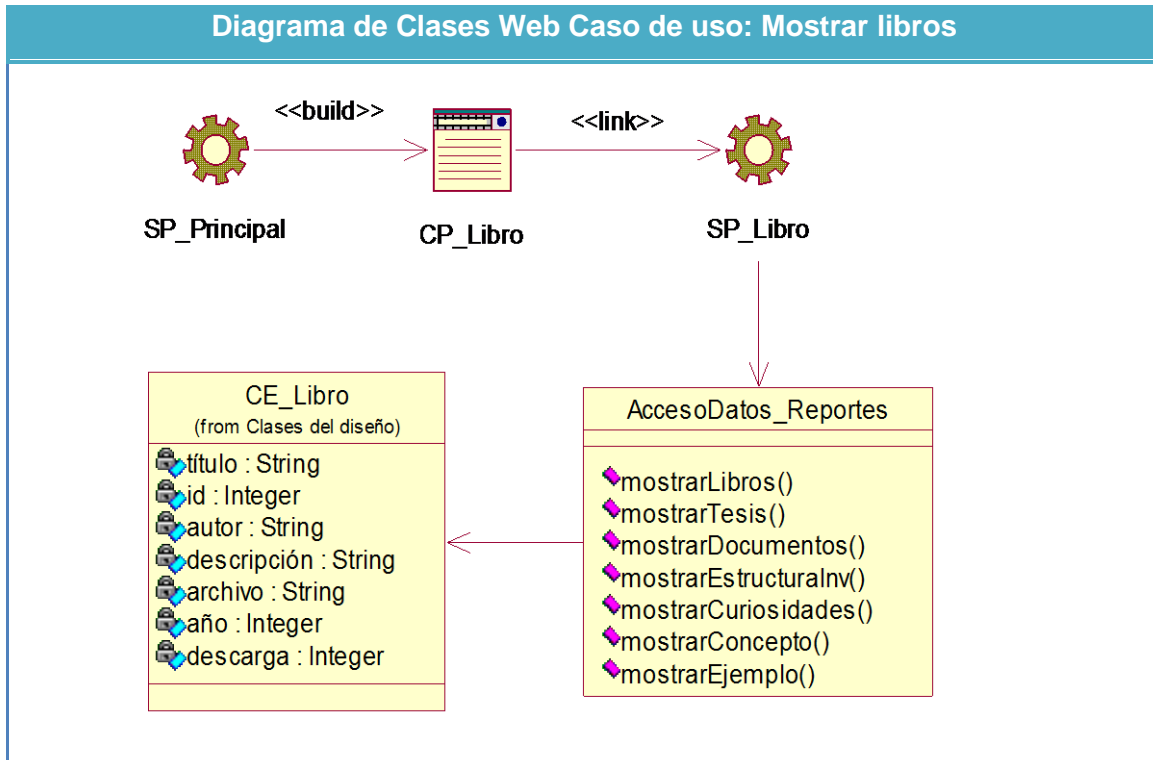


Anexo 7.14: Diagrama de clases web caso de uso: Gestionar ejercicio de seleccionar.

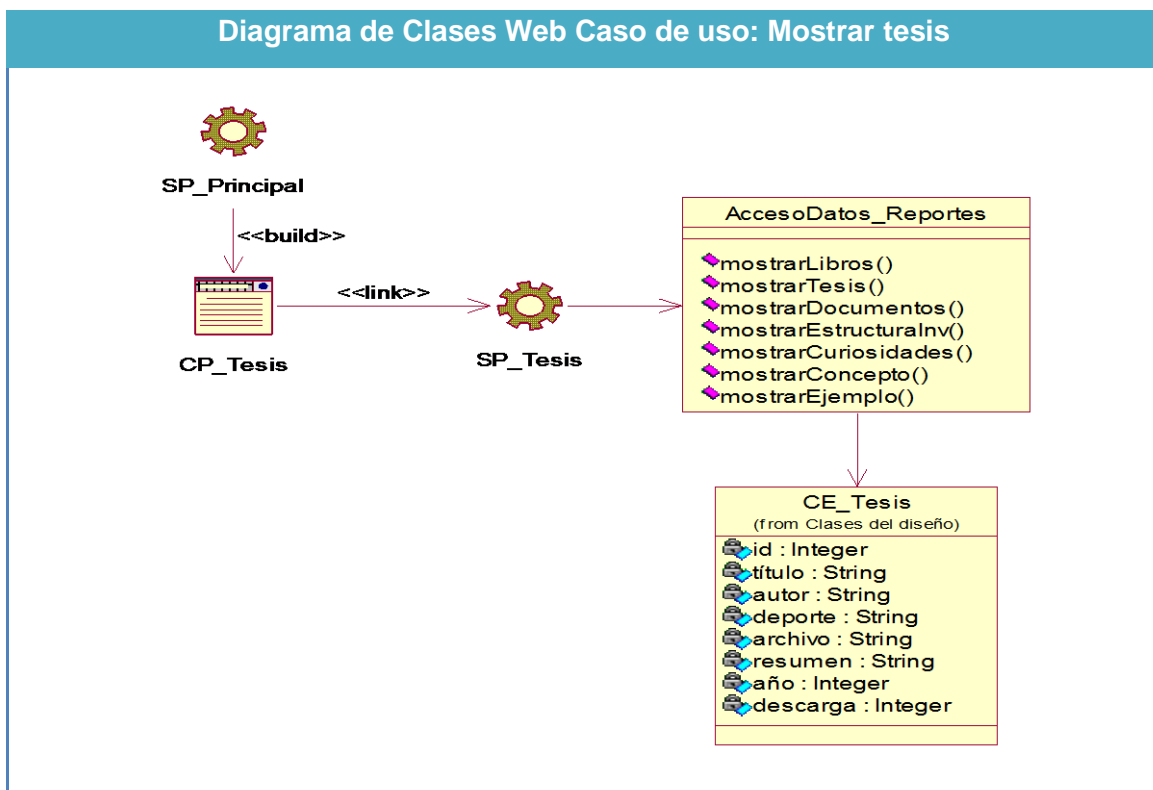


Anexo 7.15: Diagrama de clases web caso de uso: Gestionar ejercicio de completar.

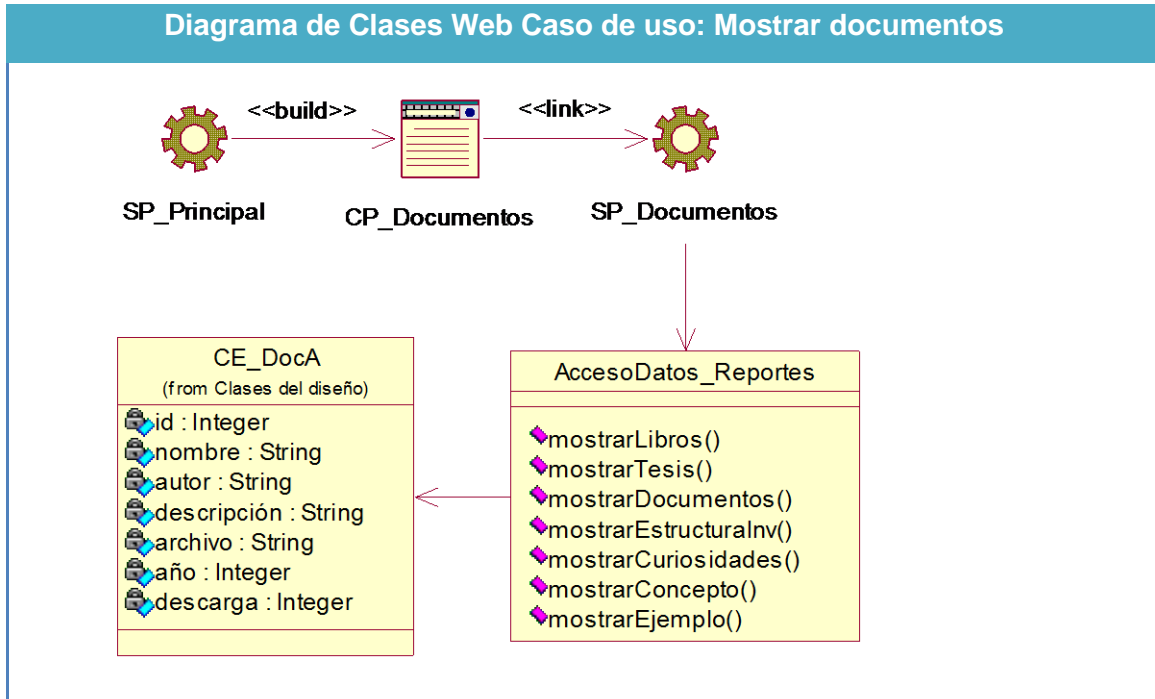
Diagramas de Clases Web: Paquete Reportes.



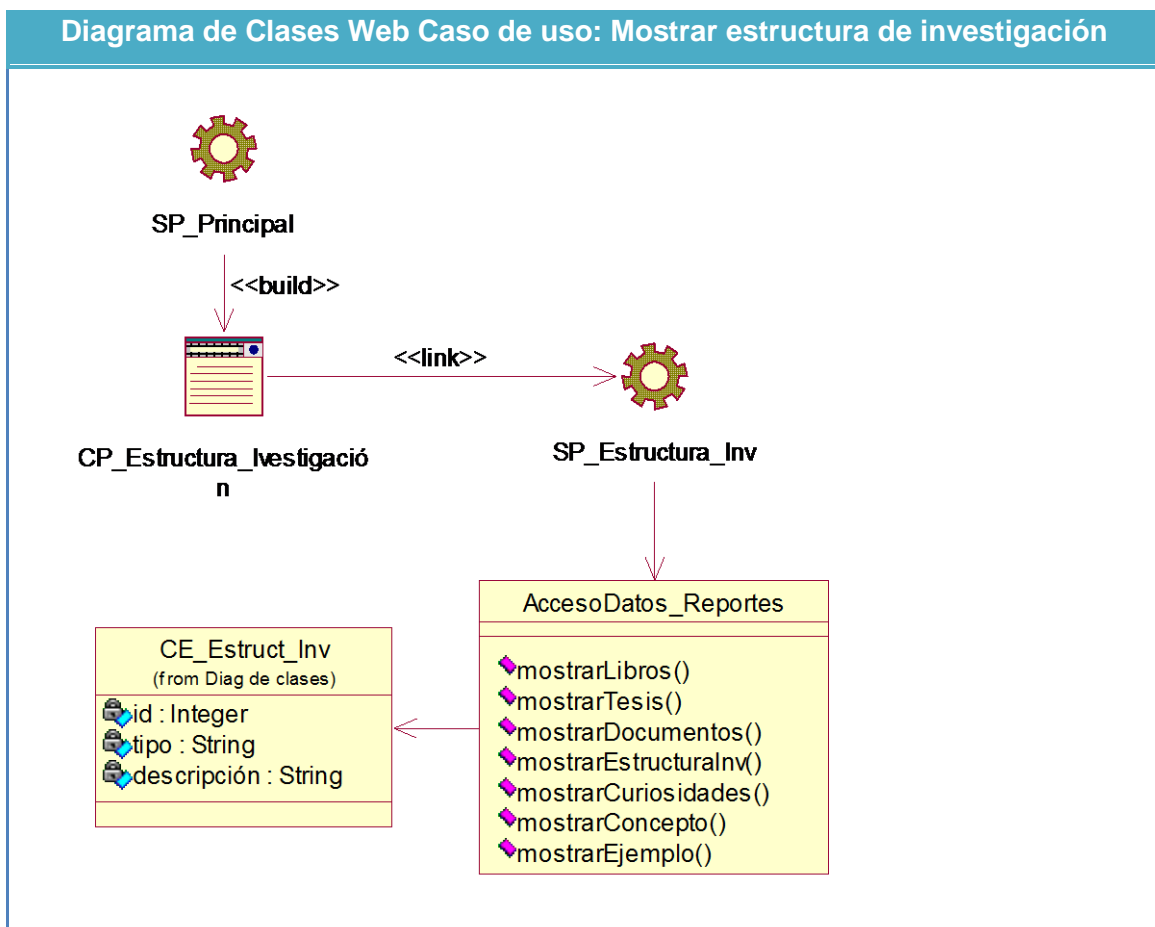
Anexo 7.16: Diagrama de clases web caso de uso: Mostrar libros.



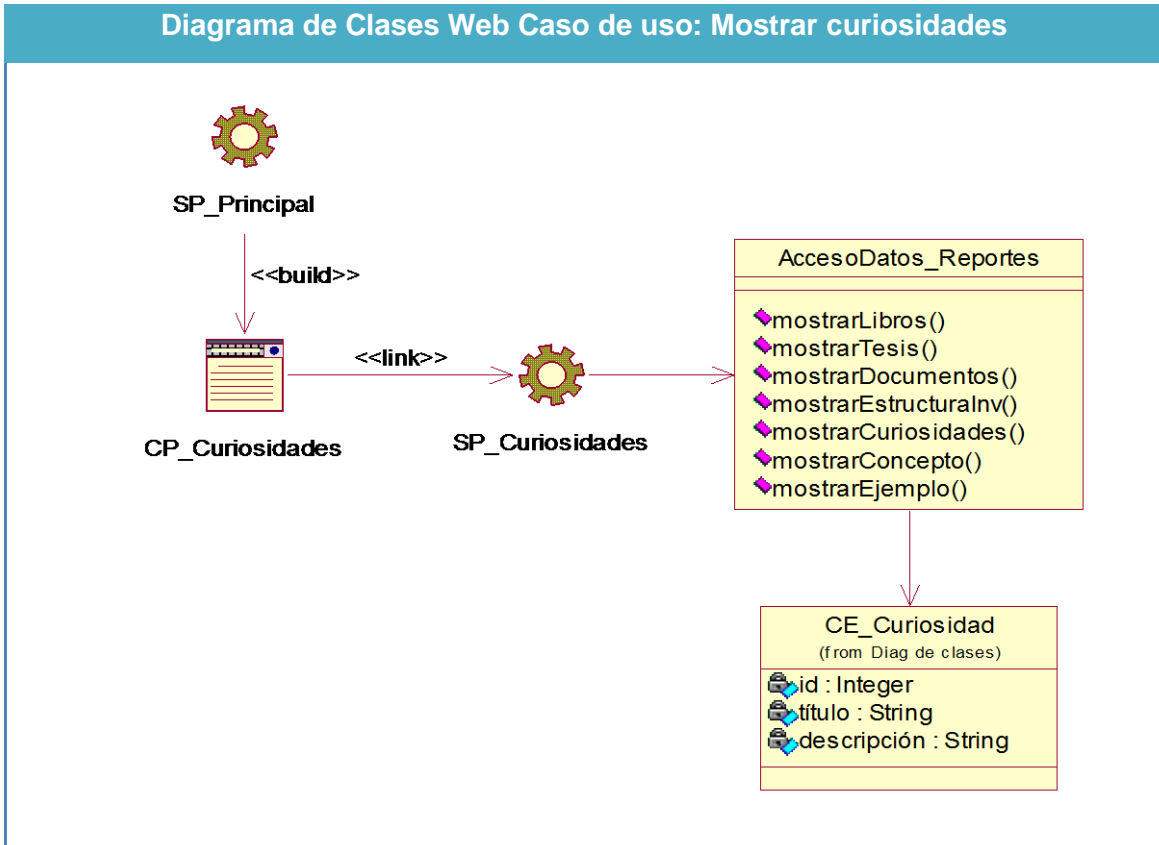
Anexo 7.17: Diagrama de clases web caso de uso: Mostrar tesis.



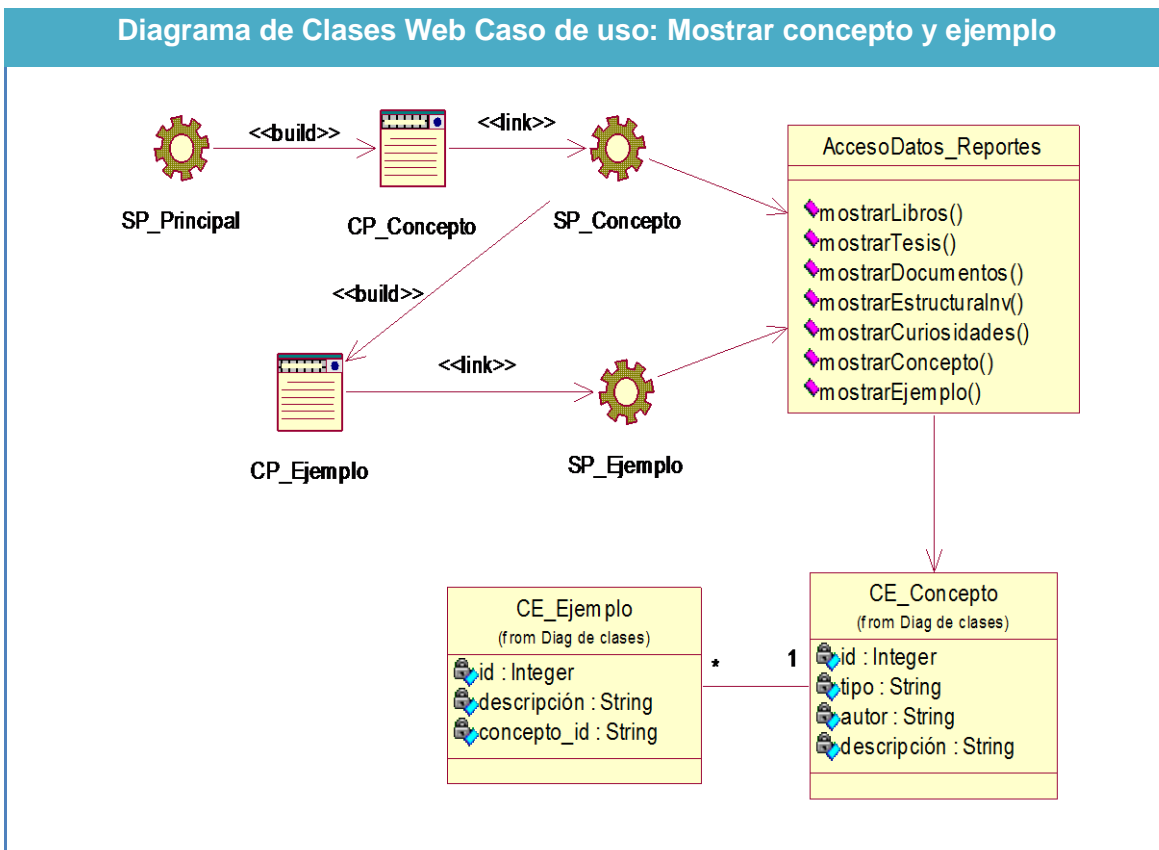
Anexo 7.18: Diagrama de clases web caso de uso: Mostrar documentos.



Anexo 7.19: Diagrama de clases web caso de uso: Mostrar estructura de investigación.

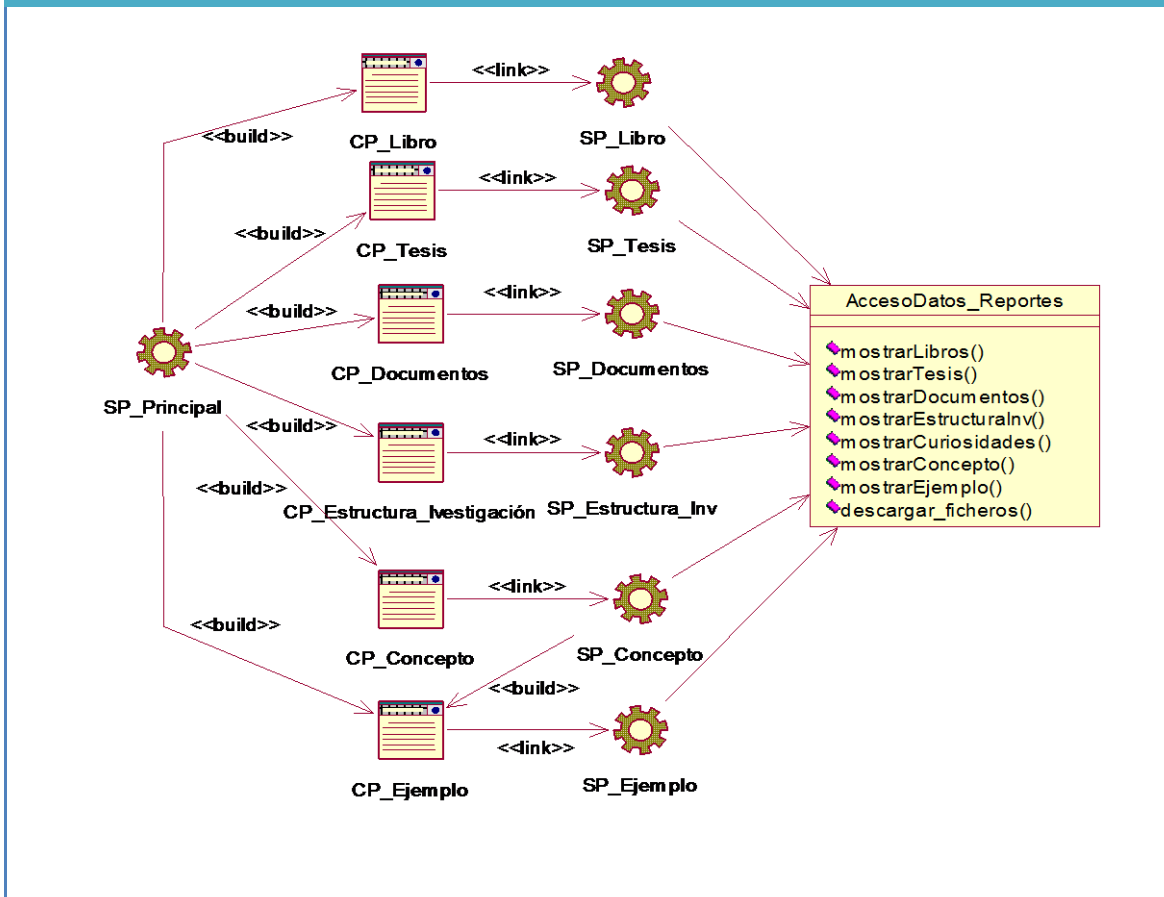


Anexo 7.20: Diagrama de clases web caso de uso: Mostrar curiosidades.



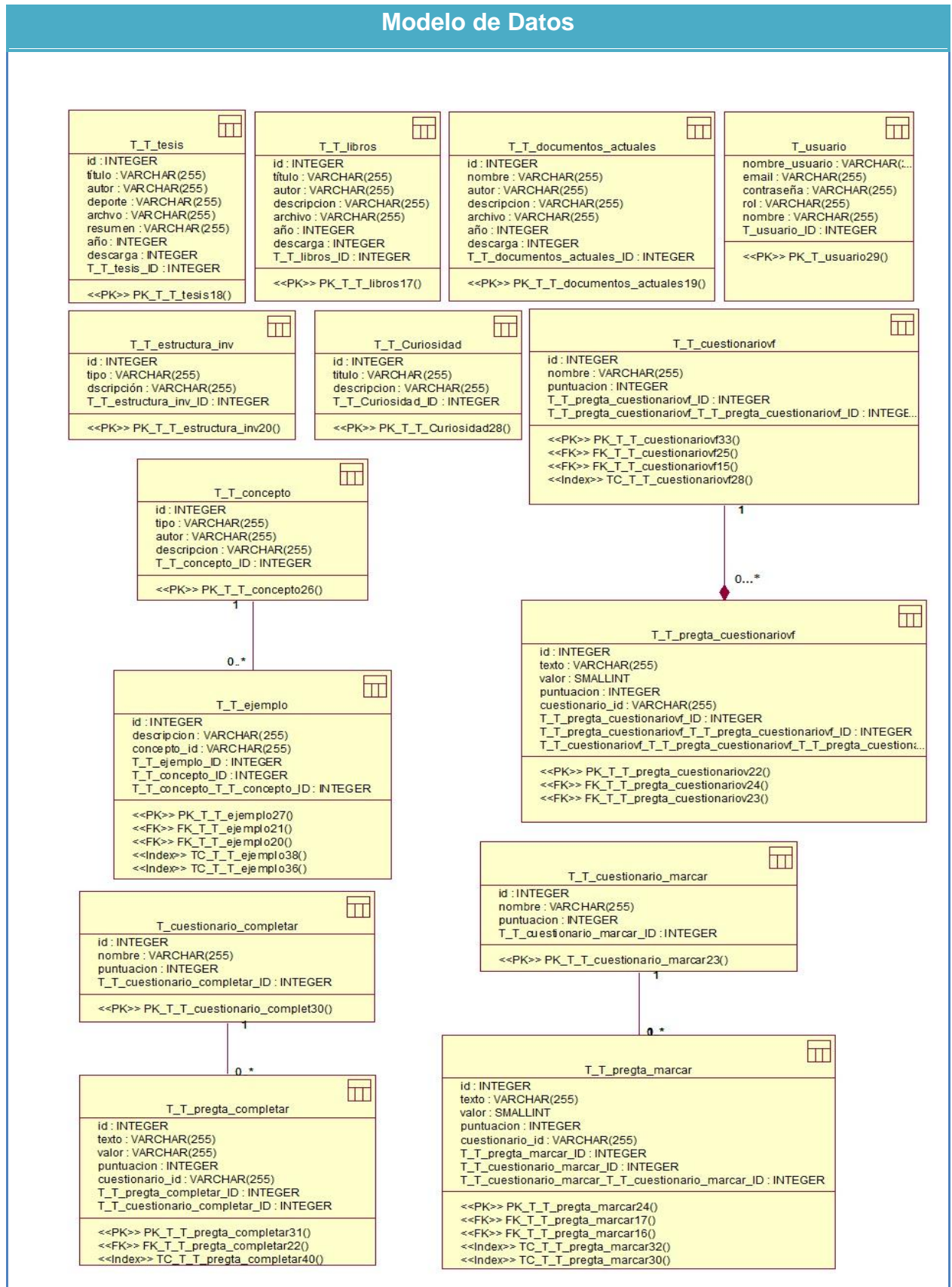
Anexo 7.21: Diagrama de clases web caso de uso: Mostrar concepto y ejemplo.

Diagrama de Clases Web Caso de uso: Descargar ficheros



Anexo 7.22: Diagrama de clases web caso de uso: Descargar ficheros.

Anexo 8: Modelo de Datos.



Anexo 8: Modelo de Datos

Anexo 9: Ventana de la aplicación: Principal.

UNISS

Inicio Libros Tesis Documentos Cuestionarios Conceptos Curiosidades Estructuras de investigación

Gloria del deporte cubano
Yuliesky Gourriel

¿Qué podemos ofrecerles a estos jóvenes?

La oportunidad de adquirir una carrera universitaria en el terreno de la educación física y del deporte, que les permita más tarde vivir decorosamente como profesores del deporte, investigadores de la actividad y formadores de nuevos atletas...

Siguiete >

UNISS UNIVERSIDAD SANCTI SPIRITUS

Libros, Tesis, Documentos...
Usuarios de la UNISS

Este sitio web fue desarrollado con el objetivo de hacer pública y accesible para todos la información y la novedad científica que aporta la carrera de Licenciatura en Cultura Física a la Universidad.

Ejercicios y Cuestionarios...
Estudiantes de Cultura Física

Cuenta con un apartado académico donde los estudiantes de la carrera de cultura física miden sus conocimientos y habilidades mediante un sistema de ejercicios o cuestionarios en línea.

Panel de Administración
Profesores de la carrera

Los profesores de la facultad, mediante un panel de control crean los ejercicios que funcionaran en línea y que estarán disponibles en cualquier momento para quien tenga interés en realizarlos.

@usuario @estudiante @profesor

Uniss Todos los derechos reservados Ir arriba

Anexo 10: Ventana de la aplicación: Autenticarse.

Formulario de acceso

Usuario
Administrador

Contraseña
●●●●●●●●

Entrar

Anexo 11: Ventana de la aplicación: Gestionar concepto.

Anexo 12: Ventana de la aplicación: Gestionar documento.

Anexo 13: Ventana de la aplicación: Gestionar ejercicio de verdadero y falso.

Panel de administración 4/06/2013 John Doe Salir

Cuestionario: algun orto nombre para probar
Puntuación total: 20
Nueva Pregunta:

texto:

Valor:
Seleccione el valor

Puntuación:

Menú de administración

- Usuarios
- Libros
- Tesis
- Documentos
- Conceptos
- Curiosidades
- Estructura de Investigación

Ejercicios

- Verdadero y Falso

Anexo 14: Ventana de la aplicación: Mostrar concepto.

Panel de administración 4/06/2013 John Doe Salir

Listado de conceptos

Tipo	Autor	Ejemplos	Editar	Eliminar
concepto 26	autor 12	Ejemplos (6)	Editar	Eliminar
concepto 19	autor 2	Ejemplos (4)	Editar	Eliminar
concepto 23	autor 6	Ejemplos (1)	Editar	Eliminar
--Tipo de concepto	Autor del concepto	Ejemplos (0)	Editar	Eliminar
--bnmbnm	bnmbnm	Ejemplos (0)	Editar	Eliminar

+ Nuevo concepto

Menú de administración

- Usuarios
- Libros
- Tesis
- Documentos
- Conceptos
- Curiosidades
- Estructura de Investigación

Ejercicios

- Verdadero y Falso

Uniss Todos los derechos reservados Ir arriba

Anexo 15: Ventana de la aplicación: Mostrar libros.

Panel de administración 4/06/2013 John Doe Salir

Listado de libros

Título	Autor	Descargas			
Quelque livre	Antonie revelliere	6	Descargar	Editar	Eliminar
Quelque livre	Antonie revelliere	0	Descargar	Editar	Eliminar

+ Nuevo libro

Menú de administración

- Usuarios
- Libros
- Tesis
- Documentos
- Conceptos
- Curiosidades
- Estructura de Investigación

Ejercicios

- Verdadero y Falso

Uniss Todos los derechos reservados Ir arriba

Anexo 16: Ventana de la aplicación: Mostrar ejercicios de verdadero o falso.

Panel de administración 4/06/2013 John Doe Salir

Listado de cuestionarios

Nombre	Profesor	Puntuación Total			
Funcionamiento del cuerpo humano.	Profesor	100	Preguntas	Editar	Eliminar
Otra cosa	Profesor	30	Preguntas	Editar	Eliminar
Un nombre para este cuestionario	Profesor	50	Preguntas	Editar	Eliminar
algun orto nombre para probar	Profesor	20	Preguntas	Editar	Eliminar

+ Nuevo cuestionario

Menú de administración

- Usuarios
- Libros
- Tesis
- Documentos
- Conceptos
- Curiosidades
- Estructura de Investigación

Ejercicios

- Verdadero y Falso

Uniss Todos los derechos reservados Ir arriba

Anexo 17: Ventana de la aplicación: Mostrar usuarios

The screenshot shows the UNISS administration interface. At the top, there is a navigation bar with the UNISS logo and a series of icons representing various sports. Below this, a header bar displays the current page ('Panel de administración'), date ('4/06/2013'), user name ('John Doe'), and a 'Salir' button.

The main content area is divided into two columns. The left column, titled 'Listado de usuarios', contains a table with the following data:

Nombre	Usuario	Correo	Administrador	
John Doe	john	john123@gmail.com		Editar Eliminar
Nemecio Capote	nemecio	nemecio@gmail.com		Editar Eliminar
Yainel García Alfonso	nane	nane@gmail.com	X	Editar Eliminar
Zinedine Zidane	zidane	zidane@gmail.com		Editar Eliminar
Diego Armando Maradona	diego	diego@gmail.com		Editar Eliminar

Below the table, there is a '+ Nuevo usuario' link. The right column, titled 'Menú de administración', lists several menu items: Usuarios, Libros, Tesis, Documentos, Conceptos, Curiosidades, Estructura de Investigación, Ejercicios, and Verdadero y Falso.

At the bottom of the page, there is a footer with the text 'Uniss Todos los derechos reservados' on the left and 'Ir arriba' on the right.