

**Universidad de las Ciencias Informáticas
Facultad 1**



**Sistema para la Gestión de la Información Curricular de los
Profesores**

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Andy Pérez Silva

Tutores: Ing. Yoan Sánchez Mesa
Ing. Lizandra Arza Pérez

Ciudad de la Habana, Junio del 2009

Declaración de Autoría

Declaro que soy el único autor del trabajo titulado:

_____ Sistema para la Gestión de la Información Curricular de los Profesores _____

y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Andy Pérez Silva

Yoan Sanchez Mesa

Datos de Contacto

Tutores:

Ing. Yoan Sánchez Mesa (ysanchezm@uci.cu)

Instructor recién graduado. Facultad 1.

Ing. Lizandra Arza Pérez (lizandra@uci.cu)

Decana Facultad 4.

“La grandeza de un hombre no se mide por el terreno que ocupan sus pies, sino por el horizonte que descubren sus ojos”.

José Martí

Agradecimientos

A mi mamá Rossmary, mi papá Julito, por darme la vida y estar conmigo en todo momento, haciéndome ver las cosas malas y celebrando mis victorias.

A mi novia Annarelys por estar siempre a mi lado en las buenas y en las malas, y soportarme durante todo este tiempo.

A mis abuelos Mama Cary, Aba y Abo, por darme siempre su apoyo incondicional.

A mis amigos Tike, Guille y Albertón, porque siempre estuvieron ahí cuando los necesité, brindándome su amistad y apoyo en los momentos difíciles.

A mis tutores Yoan y Lizandra por su ayuda en el desarrollo de mi tesis.

A Waldo, por su apoyo y por ayudarme a mí y a mi mamá.

A Erlán, porque parte de lo que soy se lo debo a él.

Muy especialmente a Alberto, que sin su ayuda no hubiese podido llegar a donde estoy.

Dedicatoria

A Mita, por darme la vida y enseñarme a conducirme siempre por el camino correcto, y porque lo que he sido, soy y seré, es gracias a ella. Gracias por permitirme el orgullo de ser tu hijo.

Resumen

En la actualidad, es muy común escuchar acerca de la importancia de la información, en ocasiones sin conocer si es una afirmación válida sobre un acontecimiento. Para muchas personas es evidente que la información puede llevar a fines diversos y que no necesariamente tienen que estar relacionados con un hecho científico. Aunque la información es un término muy amplio y general, puede decirse que “es *la acción y efecto de informar o informarse*”¹, es un fenómeno que proporciona significado o sentido a las cosas, e indica mediante códigos y conjuntos de datos, los modelos del pensamiento humano.

Para los profesionales que son generadores de conocimiento, investigadores, productores, etc. se requiere de un sistema para guardar y generar información de interés para la comunidad, que además puede ser reproducida y entendida por una lógica formal del pensamiento, refiriéndose como comunidad en este caso al marco de la Universidad de las Ciencias Informáticas que es el punto en cuestión. En la Universidad de las Ciencias Informáticas existe un número considerable de profesores de los cuales se hace necesario poder gestionar determinado tipo de información.

Este trabajo consiste en la elaboración de un sistema que sea capaz de gestionar la información curricular de los profesores de la Facultad 1. En la Universidad, se han presentado trabajos en diferentes eventos, relacionados con el tema de la gestión de la información de los profesores, pero no se han logrado desarrollar completamente, de manera que puedan brindar este servicio.

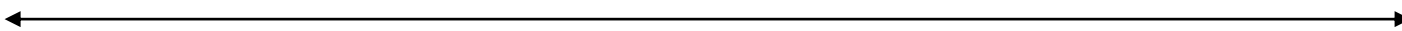
El principal objetivo de este trabajo es el desarrollo de una Aplicación Web, que tenga como características el ser una aplicación segura, agradable para el usuario, y que sea capaz de gestionar la información curricular de los profesores, de forma más rápida y fácil, incluyendo información que en un momento determinado, sirve para la toma de decisiones, así como hacer menos difícil el trabajo para los encargados de manejar esta información.

¹ Concepto de información (Real Academia de la Lengua Española, 1936)

Índice

Introducción.....	1
Situación Problemática.....	3
Métodos de Investigación:.....	5
Capítulo 1: Fundamentación teórica.....	7
Introducción	7
1.1 Conceptos Fundamentales.....	7
1.1.1 Currículum vitae	7
1.1.2 Gestión de Información	8
1.1.3 Gestión de Información en la Informática	8
1.1.4 Información	8
1.2 Sistemas para la Gestión de Currículos.....	9
1.2.1 Sistema de gestión curricular Institución Universitaria Esumer	9
1.2.2 CurrículumFácil	9
1.2.3 NitroCV.....	9
1.3 Herramientas y tecnologías posibles a utilizar.....	10
1.3.1 Lenguajes para el desarrollo Web	11
1.3.1.1 Lenguajes del lado del servidor	11
1.3.1.1.1 Java.....	12
1.3.1.1.2 Hypertext Preprocessor (PHP).....	12
1.3.1.2 Lenguajes del lado del cliente.....	15
1.3.1.2.1 Hypertext Markup Language (HTML).....	15
1.3.1.2.2 JavaScript.....	16
1.3.1.2.3 Asynchronous JavaScript And XML (AJAX)	16
1.3.2 Frameworks	18
1.3.2.1 Extend JS (Ext).....	18
1.3.2.2 CodeIgniter.....	19
1.3.3 Content Management System (CMS)	20
1.3.3.1 Drupal.....	20
1.3.3.2 Joomla.....	20
1.3.4 Sistemas Gestores de Base de Datos (SGBD).....	21
1.3.4.1 MySQL	21
1.3.4.1 PostgreSQL.....	22
1.3.5 Metodologías de desarrollo del software	23
1.3.5.1 Programación Extrema (XP)	24
1.3.5.2 SCRUM	25
1.3.5.3 Proceso Unificado del Software (RUP)	25
1.3.6 Posibles Herramientas CASE a utilizar para el modelado	28
1.3.6.1 Rational Rose	28
1.3.6.2 Visual Paradigm	28

1.3.7 Lenguaje de Modelado Utilizado.....	29
1.3.7.1 Lenguaje Unificado de Modelado (UML).....	29
1.3.8 Servidor de aplicaciones WEB	30
1.3.8.1 Apache	30
1.3.8.2 Wamp.....	31
1.3.9 Patrón de arquitectura	31
1.3.10 Entorno de desarrollo integrado (IDE).	32
Conclusiones.....	33
Capítulo 2: Características del Sistema.....	35
Introducción.	35
2.1 Información que se maneja	35
2.1.1 Información personal del profesor	36
2.1.2 Información del currículo	36
2.2 Propuesta del Sistema	37
2.3 Modelo de Dominio	37
2.3.1 Glosario de términos del modelo de dominio	38
2.3.2 Diagrama del Modelo del Dominio	38
2.4 Especificación de Requerimientos de Software.....	39
2.4.1 Requerimientos Funcionales (RF)	40
2.4.1 Requerimientos No Funcionales.....	43
2.5 Descripción del los Actores del Sistema	45
2.6 Descripción Breve de los Casos de Uso	46
2.7 Diagrama de Casos de Uso del Sistema	49
Conclusiones.....	50
Capítulo 3: Análisis y Diseño del Sistema	51
Introducción	51
3.1 Diagrama de Clases del Análisis	51
3.2 Modelo de diseño	55
3.2.1 Patrones de diseño	55
3.2.1.1 Patrón Modelo - Vista - Controlador.....	55
3.2.1.2. Patrón Alta cohesión.....	56
3.2.2. Diagrama de secuencia.....	56
3.2.3 Diagrama de Clases Web.....	58
3.2.4 Descripción de las Clases del Diseño.....	64
3.2.4.1 Descripción de Clases del Modelo.....	64
3.2.4.2 Descripción de las Clases Controladoras	68



3.2.5 Diseño de la Base de Datos.	77
3.2.5.1 Diagrama Entidad-Relación de la Base de Datos.	77
3.2.5.2. Descripción de las tablas de la Base de Datos.	81
3.2.6. Definiciones de diseño.	94
3.2.7. Tratamiento de errores.	94
3.2.8. Seguridad.	96
3.2.9. Interfaz.	96
3.2.10. Estándares de codificación.	96
3.3.10.1 Declaraciones:	97
3.3.10.2 Espacios en blanco.	97
3.2.11. Concepción de la ayuda.	97
Conclusiones.....	97
Capítulo 4: Implementación	99
Introducción	99
4.1. Modelo de despliegue.....	99
4.2. Modelo de Implementación.	100
4.3. Diagrama de componentes.....	101
Conclusiones.....	106
Conclusiones generales.....	107
Recomendaciones.....	108
Bibliografía referenciada	109
Bibliografía consultada	115
Anexos.....	117
Anexo1. Descripción ampliada de los casos de uso del sistema.	117
Anexo 2. Diagrama de Clases Web.....	146
Anexo 3. Descripción de Clases del Modelo.....	160
Anexo 4. Descripción de Clases Controladoras.....	174
Anexo 5. Diagramas de secuencia.	189
Glosario de Términos	194

Índice de figuras.

Figura 1 Modelo de dominio o conceptual.....	39
Figura 2 Diagrama de casos de uso del sistema.....	50

Figura 3 DCA "Autenticar usuario".	52
Figura 4 DCA "Gestionar usuarios".	52
Figura 5 DCA "Gestionar datos personales".	53
Figura 6 DCA "Gestionar currículum".	53
Figura 7 DCA "Generar reportes".	54
Figura 8 DCA "Cambiar contraseña".	54
Figura 9 DCA "Realizar búsqueda".	54
Figura 10 DCA "Gestionar nomencladores".	55
Figura 11 Diagrama de secuencia "Autenticar".	57
Figura 12 Diagrama de secuencia "Gestionar usuario".	58
Figura 13 DCW "Autenticar usuario".	59
Figura 14 DCW "Gestionar usuarios".	60
Figura 15 DCW "Gestionar datos personales".	61
Figura 16 DCW "Generar reportes".	62
Figura 17 DCW "Gestionar búsqueda".	63
Figura 18 DER Relación rol-usuario-profesor-curriculo.	78
Figura 19 DER Currículo-Información del currículum (parte 1).	79
Figura 20 DER Relación Currículum-Información del currículum (parte 2).	80
Figura 21 Mensaje de error. Validación en la Página Cliente.	95
Figura 22 Mensaje de error. Validación en la Página Servidora.	95
Figura 23 Modelo de Despliegue.	99
Figura 24 Modelo de Implementación.	100
Figura 25 Diagrama de Componentes. Autenticarse.	101
Figura 26 Diagrama de componentes. Realizar Búsqueda.	102
Figura 27 Diagrama de componentes. Generar Reportes.	103
Figura 28 Diagrama de componentes. Gestionar Currículum.	103
Figura 29 Diagrama de componentes. Gestionar Nomencladores.	104
Figura 30 Diagrama de componentes. Gestionar Datos Personales.	104
Figura 31 Diagrama de componentes. Gestionar Usuarios.	105
Figura 32 Diagrama de componentes. Gestionar contraseña.	105

Figura 33 DCW "Cambiar contraseña"	146
Figura 34 DCW "Gestionar nomenclador categoría docente"	146
Figura 35 DCW "Gestionar nomenclador categoría científica"	147
Figura 36 DCW "Gestionar nomenclador idioma"	148
Figura 37 DCW "Gestionar nomenclador tipo de evento"	148
Figura 38 DCW "Gestionar categoría científica"	149
Figura 39 DCW "Gestionar categoría docente"	150
Figura 40 DCW "Gestionar cursos de postgrado impartidos"	150
Figura 41 DCW "Gestionar cursos de postgrado recibidos"	151
Figura 42 DCW "Gestionar curso pregrado impartidos"	152
Figura 43 DCW "Gestionar eventos"	152
Figura 44 DCW "Gestionar idiomas"	153
Figura 45 DCW "Gestionar investigaciones"	154
Figura 46 DCW "Gestionar misiones"	155
Figura 47 DCW "Gestionar otra información"	156
Figura 48 DCW "Gestionar productos patentados"	156
Figura 49 DCW "Gestionar proyectos"	157
Figura 50 DCW "Gestionar publicaciones"	158
Figura 51 DCW "Gestionar responsabilidades ocupadas"	159
Figura 52 DCW "Gestionar tesis tutoradas"	159
Figura 53 Diagrama de secuencia "Realizar búsqueda"	189
Figura 54 Diagrama de secuencia "Cambiar contraseña"	190
Figura 55 Diagrama de secuencia "Generar Reportes"	190
Figura 56 Diagrama de secuencia "Gestionar información del currículum"	191
Figura 57 Diagrama de secuencia "Gestionar datos personales"	192
Figura 58 Diagrama de secuencia "Gestionar nomencladores"	193

Índice de tablas.

Tabla 1 Descripción de los usuarios del sistema	46
Tabla 2 Breve descripción del caso de uso "Autenticar"	46

Tabla 3 Breve Descripción del caso de uso "Gestionar Usuario"	47
Tabla 4 Breve Descripción del caso de uso "Gestionar Datos Personales".....	47
Tabla 5 Breve Descripción del caso de uso "Gestionar Currículo".	47
Tabla 6 Breve Descripción del caso de uso "Generar Reportes".....	48
Tabla 7 Breve Descripción del caso de uso "Cambiar Contraseña".	48
Tabla 8 Breve Descripción del caso de uso "Gestionar Búsqueda".....	49
Tabla 9 Breve Descripción del caso de uso "Gestionar Nomencladores".....	49
Tabla 10 Descripción de la clase del modelo "Usuariomodel".....	64
Tabla 11 Descripción de la clase del modelo "Profesormodel".....	66
Tabla 12 Descripción de la clase del modelo "Busquedamodel".....	67
Tabla 13 Descripción de la clase del modelo "Reportemodel".....	68
Tabla 14 Descripción de la clase controladora "Profesor".....	68
Tabla 15 Descripción de la clase controladora "Usuario".....	69
Tabla 16 Descripción de la clase controladora "Administrador".....	69
Tabla 17 Descripción de la clase controladora "Directivo".....	70
Tabla 18 Descripción de la clase controladora "Autenticar usuario".....	70
Tabla 19 Descripción de la clase controladora "Gestionar usuarios".....	71
Tabla 20 Descripción de la clase controladora "Gestionar datos personales".....	72
Tabla 21 Descripción de la clase controladora "Gestionar búsqueda".....	75
Tabla 22 Descripción de la clase controladora "Generar reportes".....	76
Tabla 26 Descripción de la tabla "profesor".....	81
Tabla 27 Descripción de la tabla "usuario".....	82
Tabla 28 Descripción de la tabla "rol".....	82
Tabla 29 Descripción de la tabla "curriculo".....	82
Tabla 30 Descripción de la tabla "categoría_científica".....	83
Tabla 31 Descripción de la tabla "categoría_docente".....	83
Tabla 32 Descripción de la tabla "categoria_docente_curriculo".....	83
Tabla 33 Descripción de la tabla "co_autor".....	84
Tabla 34 Descripción de la tabla "curriculo_categoria_científica".....	85
Tabla 35 Descripción de la tabla "curriculo_curso_pregrado".....	85

Tabla 36	Descripción de la tabla "curriculo_evento".....	86
Tabla 37	Descripción de la tabla "curriculo_investigación".....	86
Tabla 38	Descripción de la tabla "curriculo_mision".....	86
Tabla 39	Descripción de la tabla "curriculo_responsabilidad_ocupada".....	87
Tabla 40	Descripción de la tabla "curso_postgrado".....	87
Tabla 41	Descripción de la tabla "curso_postgrado_Impartido".....	88
Tabla 42	Descripción de la tabla "curso_postgrado_Recibido".....	88
Tabla 43	Descripción de la tabla "dominio_idioma".....	89
Tabla 44	Descripción de la tabla "evento".....	89
Tabla 45	Descripción de la tabla "idioma".....	90
Tabla 46	Descripción de la tabla "investigacion".....	90
Tabla 47	Descripción de la tabla "mision".....	90
Tabla 48	Descripción de la tabla "otra_informacion".....	91
Tabla 49	Descripción de la tabla "producto_patentado".....	91
Tabla 50	Descripción de la tabla "proyecto".....	92
Tabla 51	Descripción de la tabla "proyecto_curriculo".....	92
Tabla 52	Descripción de la tabla "publicacion".....	92
Tabla 53	Descripción de la tabla "responsabilidad_ocupada".....	93
Tabla 54	Descripción de la tabla "tesis_tutorada".....	93
Tabla 55	Descripción de la tabla "tipo_evento".....	94
Tabla 56	Descripción ampliada del caso de uso "Autenticar".....	118
Tabla 57	Descripción ampliada del caso de uso "Gestionar Usuario".....	123
Tabla 58	Descripción ampliada del caso de uso "Gestionar Datos Personales".....	128
Tabla 59	Descripción ampliada del caso de uso "Gestionar Currículo".....	135
Tabla 60	Descripción ampliada del caso de uso "Generar Reportes".....	137
Tabla 61	Descripción ampliada del caso de uso "Cambiar contraseña".....	139
Tabla 62	Descripción ampliada del caso de uso "Gestionar Búsqueda".....	142
Tabla 63	Descripción ampliada del caso de uso "Gestionar Nomencladores".....	145
Tabla 64	Descripción de la Clase del Modelo "Catcientificamodel".....	160
Tabla 65	Descripción de la Clase del Modelo "Catdocentemodel".....	161

Tabla 66	Descripción de la Clase del Modelo "Cursopostimpartidomodel".	162
Tabla 67	Descripción de la Clase del Modelo "Cursopostrecibidomodel".	163
Tabla 68	Descripción de la Clase del Modelo "Cursopregradomodel".	164
Tabla 69	Descripción de la Clase del Modelo "Eventomodel".	165
Tabla 70	Descripción de la Clase del Modelo "Idiomasmodel".	166
Tabla 71	Descripción de la Clase del Modelo "Investigacionmodel".	167
Tabla 72	Descripción de la Clase del Modelo "Misionmodel".	168
Tabla 73	Descripción de la Clase del Modelo "Tesismodel".	169
Tabla 74	Descripción de la Clase del Modelo "Otrainformacionmodel."	169
Tabla 75	Descripción de la Clase del Modelo "Productomodel".	170
Tabla 76	Descripción de la Clase del Modelo "Proyectomodel".	171
Tabla 77	Descripción de la Clase del Modelo "Publicacionmodel".	172
Tabla 78	Descripción de la Clase del Modelo "Responsabilidadmodel".	173
Tabla 79	Descripción de la Clase Controladora "Gestionar Categoría Científica".	174
Tabla 80	Descripción de la Clase Controladora "Gestionar Categoría Docente".	175
Tabla 81	Descripción de la Clase Controladora "Gestionar cursos de postgrado".	176
Tabla 82	Descripción de la Clase Controladora "Gestionar cursos de pregrado impartidos".	177
Tabla 83	Descripción de la Clase Controladora "Gestionar responsabilidades ocupadas".	178
Tabla 84	Descripción de la Clase Controladora "Gestionar eventos".	179
Tabla 85	Descripción de la Clase Controladora "Gestionar idiomas".	179
Tabla 86	Descripción de la Clase Controladora "Gestionar investigación".	180
Tabla 87	Descripción de la Clase Controladora "Gestionar misiones cumplidas".	181
Tabla 88	Descripción de la Clase Controladora "Gestionar otra información".	182
Tabla 89	Descripción de la Clase Controladora "Gestionar productos patentados".	183
Tabla 90	Descripción de la Clase Controladora "Gestionar proyectos".	183
Tabla 91	Descripción de la Clase Controladora "Gestionar publicaciones".	184
Tabla 92	Descripción de la Clase Controladora "Gestionar tesis tutoradas".	185
Tabla 93	Descripción de la Clase Controladora "Gestionar nomenclador categoría científica".	186
Tabla 94	Descripción de la Clase Controladora "Gestionar nomenclador categoría docente".	186
Tabla 95	Descripción de la Clase Controladora "Gestionar nomenclador idioma".	187

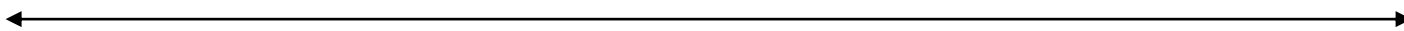


Tabla 96 Descripción de la Clase Controladora "Gestionar tipo de evento". 188

Introducción

Globalmente existe una necesidad en todas las empresas e instituciones de gestionar de forma eficaz la información, recurso al cual no se atendía debidamente, se descuidaba y no se trataba con la importancia que requiere. La información es muy importante, debido a que produce un efecto en el conocimiento del receptor, además, permite generar puntos de vista, análisis y discusiones de situaciones de la realidad. El grado de utilidad de la información depende estrictamente de la importancia que se de a la misma, la comprensión, la aprehensión, y el uso que se haga de ella.

Para garantizar el almacenamiento de tan importante recurso, el hombre ha ideado gran cantidad de métodos a lo largo de la historia, y a pesar del avance existente en la tecnología en el mundo actual, el documento escrito sigue siendo la forma de almacenamiento de información que prevalece. Desde la creación de las primeras computadoras, décadas atrás, el hombre las ha utilizado para intereses personales y sociales, creando aplicaciones para satisfacer esos intereses. Debido a la necesidad de trabajar con grandes volúmenes de información, el mundo del software se ha desarrollado, dando lugar a sistemas que permiten la gestión y presentación de la misma.

La gestión de información, en el mundo de la informática, no es más que un conjunto de actividades que aseguran una correcta información para la toma de decisiones, incluyendo la designación de las personas que preparan y reciben los datos, los métodos y medios que pueden emplear, la periodicidad, y la garantía de fiabilidad y confidencialidad. (Theodore, 1990)

La Universidad de las Ciencias Informáticas (UCI) es una de las universidades con mayor capacidad tecnológica instalada en el país, aspirando a ser una Ciudad Digital, donde se encuentren automatizada la mayor cantidad de procesos, incluyendo aquellos que se encargan de gestionar la información, por lo que se hace necesaria la informatización, al más alto nivel, de los procesos que actualmente se llevan a cabo de forma manual, y de esta manera, a través del uso de las nuevas tecnologías que brinda la universidad, lograr una mayor disponibilidad, funcionalidad, potencialidad y versatilidad de estos procesos. El recurso humano es un ente fundamental en los procesos que se desarrollan en la universidad, estos recursos humanos son profesores, trabajadores y estudiantes. Y como pilares fundamentales del desarrollo del centro, es de suma importancia gestionar y utilizar de forma adecuada la información relacionada con estos recursos humanos, sus actividades docentes, de superación y extracurriculares. Además, debido a que la Universidad realiza constantes cambios, se gestionan eventos continuamente, y en algunas

ocasiones se hace necesario impartir cursos, postgrados, u optar por una categorización más alta o una ubicación en determinada posición, y para tomar las decisiones de quiénes serán los seleccionados para realizar estas actividades, se necesita tener en cuenta el currículum² del profesor, para tomar la mejor decisión posible, de acuerdo con las cualidades del mismo.

Igualmente, existe la necesidad de gestionar determinado tipo de información de los profesores que no se puede medir o cuantificar. Las actividades extracurriculares en la Universidad son muy variadas y aportan profesionalidad e integralidad al currículum del profesor, sin embargo este tipo de información no se gestiona eficientemente. Por esto es necesario un sistema automatizado que permita gestionar toda la información de los profesores, de las actividades profesionales, de superación y extracurriculares en las que participa, esta información es el currículum enriquecido del profesor que se convierte en una herramienta importante para la toma de decisiones para los directivos de la docencia y la producción en la facultad.

“La función esencial de la gestión de información universitaria es la de supervisar, controlar y garantizar el flujo correcto de la información hacia todas las instancias que conforman la estructura de la institución, con el fin de minimizar, en lo posible, los problemas típicos que suelen presentarse en estas organizaciones, como lo es: la sobrecarga de papeles con la consecuente escasez de información, la resistencia al cambio, los celos por la información, la poca claridad en la propiedad de la información, los retrasos o extemporaneidad en el suministro de la información y la coexistencia de diferentes estilos para manejar este valioso recurso”. (Librería Científica Electrónica)

² currículum: Es la acepción singular en español del latín *currículum*. (Wikipedia La Enciclopedia Libre) (ver 1.1.1 Currículum vitae)

Situación Problemática


En la Facultad 1 de la Universidad de las Ciencias Informáticas no existe un sistema que registre la información curricular de los profesores, lo cual hace que el trabajo con la misma se haga de forma manual, esto implica que gestionar esta información se haga bastante engorroso. Una vez gestionada la información, se hace sumamente difícil el acceso a la misma, ya que se encuentra almacenada en documentos de Microsoft Word, ya sean digitales o impresos, o en alguna hoja de cálculo de Excel, y en varias ocasiones distribuidas en diferentes lugares. De esta manera, también es un problema que no existe un dispositivo de almacenamiento digital de esta información, y debe ser almacenada de esa manera, muchas veces perdiéndose información por descuidos o deterioro de documentos. Esto hace que la información no esté fácilmente accesible para los directivos de la facultad para la toma de decisiones que requieren asignación de recursos humanos, por lo que esta asignación en ocasiones no es la óptima o más adecuada de acuerdo al desempeño y conocimiento de los profesores.

A partir de la situación existente, surge la necesidad de crear un sistema que sea capaz de satisfacer estas necesidades. Teniendo en cuenta lo anteriormente planteado, se puede extraer el siguiente **problema científico**: *¿Cómo garantizar la eficiencia en la gestión de la información curricular de los profesores?* Según el problema científico identificado, puede definirse como **objeto de estudio** de la investigación, la gestión de la información, y el **campo de acción** de la misma es la gestión de información curricular de los profesores de la Facultad 1 de la Universidad de las Ciencias Informáticas. Para dar solución al problema científico planteado se define como **objetivo general** de la investigación: *Desarrollar un sistema que permita el control de la información curricular referente a los profesores*, y para dar cumplimiento al objetivo general de la investigación, se definen los siguientes **objetivos específicos**:

- Definir las tecnologías de desarrollo y metodologías a utilizar para el desarrollo de la aplicación.
- Realizar el análisis del sistema que permita controlar la información curricular de los profesores.
- Diseñar el sistema que permita controlar la información curricular de los profesores.
- Implementar el sistema que permita controlar la información curricular de los profesores.

Para lograr alcanzar estos objetivos, se trazan las siguientes tareas:

- Caracterizar las tecnologías existentes que permiten el desarrollo de sistemas informáticos.

- 
- Identificar y caracterizar las metodologías de desarrollo a utilizar.
 - Realizar el levantamiento de los requerimientos que debe cumplir el sistema.
 - Realizar las descripciones textuales de los CUS.
 - Realizar los diagramas de clases del diseño.
 - Realizar los diagramas de secuencia.
 - Realizar diagrama de despliegue.
 - Realizar los diagramas de componentes³.
 - Implementar el sistema.

El presente trabajo de diploma está formado por 4 capítulos:

- **Capítulo 1: Fundamentación Teórica.**

Este capítulo contiene una base teórica para entender el problema a solucionar. Conceptos fundamentales, técnicas, tendencias, metodologías y herramientas utilizadas son explicados además en este capítulo, así como los elementos tenidos en cuenta para definir las que se utilizan en el desarrollo del trabajo. Además se incluye el estado del arte del tema.

- **Capítulo 2: Características del sistema.**

En este capítulo se describen los procesos que es necesario automatizar, para poder entender mejor las funcionalidades que tendrá el sistema. Se hará una descripción del modelo de dominio, así como la descripción de los requerimientos funcionales y no funcionales que deberá tener el sistema.

- **Capítulo 3: Análisis y diseño del Sistema.**

En este capítulo se muestran los diagramas de clases del análisis y del diseño, con sus correspondientes diagramas interacción por cada caso de uso del sistema. Se incluye el diseño de la base de datos, así como las definiciones de diseño y la forma en que se tratarán los errores.

³ Componente: Parte modular de un sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces y proporciona la realización de los mismos.

Capítulo 4: Implementación En este capítulo se describe la implementación del sistema, para cumplimentar los requisitos especificados para el mismo, incluye el diagrama de componentes y diagrama de despliegue, se definirán los componentes a implementar, así como del modelo de implementación en el cual se describen las partes de la aplicación desde el punto de vista de la arquitectura.

Para realizar este trabajo es necesario llevar a cabo determinados métodos de investigación:

Métodos de Investigación:

Analítico-Sintético: Este método permite el análisis de toda la información acerca de las tecnologías, metodologías y herramientas posibles a ser utilizadas en el desarrollo del sistema, de esta manera se puede definir mejor las mismas, sintetizando sus características, y analizando la viabilidad de cada una.

Análisis Histórico-Lógico: Este método se utiliza para analizar la historia de la gestión de la información, los principales conceptos que giran alrededor de la misma, los sistemas que ya existen sobre el tema, además, para estudiar los procesos de registro de información.

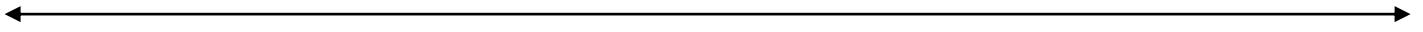
Hipotético-Deductivo: A partir de la hipótesis planteada en el trabajo, se investiga toda la información asociada al tema, para proponer una solución al problema existente.

Modelación: Una vez realizada la investigación, y analizada la información, este método es muy útil para realizar los modelos correspondientes al ciclo de vida del sistema, esto permite facilidades a la hora de cumplir con las tareas de análisis y diseño de los procesos que intervendrán en la aplicación, así como para implementar el sistema.

Observación: Este método tiene relación con la mayoría de las tareas propuestas a ser realizadas. Se puede observar cómo funciona el proceso de gestión del currículo de los profesores y los principales problemas cuando se realiza la misma. Para la realización de las pruebas, también es útil este método, debido a que se registra la información resultante de las mismas.

Experimento: Se utiliza una vez terminada la implementación, para realizar las pruebas de unidad. Y en la realización de las pruebas, para experimentar con datos de entrada que el sistema no este preparado para recibir, y de esta manera, poder alcanzar la calidad del mismo que se desea.

Entrevista: Se realizaron diferentes entrevistas para realizar el levantamiento de requisitos, así como



para entender el proceso de gestión de la información curricular de los profesores y cómo funciona este en la facultad 1.

Capítulo 1: Fundamentación teórica

Introducción

Este capítulo contiene la información acerca del estudio del estado del arte del trabajo, así como el análisis de las herramientas y tecnologías posibles a usar para desarrollar el sistema, la definición de cuáles serán las seleccionadas para realizar tal propósito. Se definen además, algunos conceptos importantes, necesarios para realizar y entender el trabajo, como por ejemplo, los conceptos de información, currículum vitae⁴, gestión, gestión de la información, entre otros.

1.1 Conceptos Fundamentales

1.1.1 Currículum vitae

El currículum vitae es una frase del latín que significa carrera de la vida, que por simplificación, a nivel mundial, se utiliza solo el término currículum. Esto no es más que el conjunto de experiencias, laborales, educacionales y vivenciales de una persona, así como la relación de títulos, honores, cargos, trabajos realizados, datos biográficos, etc., que califican a una persona o individuo. Currículum es la acepción en singular de currículum. (Definicion.de)

Existen varios tipos de currículum vitae, como por ejemplo, el cronológico, el inverso y el temático o funcional. El cronológico es cuando la información curricular es ordenada de la más antigua a la más reciente, este método, tiene como propiedad principal que resalta la evolución seguida del individuo, resalta la estabilidad y la evolución ascendente de la persona. El inverso es el que permite destacar la experiencia reciente, que en algunas ocasiones, y para algunas empresas es la información más importante a considerar. El temático o funcional no es otro que cuando se ordena el currículum en bloques temáticos, y tiene como ventaja que al no seguir un orden cronológico, se pueden abordar los puntos más importante de la información curricular de un individuo, y omitir aquella información que indique errores de recorrido. (González Pérez, y otros) (Reyes Miranda, y otros)

Cuando se elabora un currículum, se deben tener en cuenta algunos aspectos importantes, que no pueden faltar dentro de la información del currículum, como son: los datos personales, el historial, tanto

⁴ Currículum Vitae: proviene del Latín y significa “carrera de la vida” (Definicion.de) (ver 1.1.1 Currículum vitae)

académico como profesional, cualidades profesionales y personales, por ejemplo, los idiomas que habla, el conocimiento de informática, historia, y otros datos de interés. Para la elaboración del currículum es necesario tener un orden y limpieza adecuados, ya que se está redactando la información personal, que será leída por otras personas.

1.1.2 Gestión de Información

La gestión de información es el proceso que se encarga de suministrar los recursos necesarios para la toma de decisiones, así como para mejorar los procesos, productos y servicios de una organización. La gestión de la información no es un proceso que se pueda llamar nuevo, es algo que preocupó al hombre desde el inicio de los tiempos, desde que el hombre se dio cuenta que aprovechar solamente su experiencia lo limitaba con respecto a aquellos que aprovechaban las experiencias colectivas, y más aún, se trazaban estrategias para poder adquirir más conocimiento. (Universidad Michoacana de San Nicolás de Hidalgo)

Como concepto puede decirse que la gestión de la información no es más que un conjunto de actividades relacionadas con la obtención de la información adecuada, a un precio adecuado, en el tiempo y lugar adecuados, para tomar la decisión adecuada. (Woodman, 1985)

1.1.3 Gestión de Información en la Informática

Es un conjunto de actividades que aseguran una correcta información para la toma de decisiones, incluyendo la designación de las personas que preparan y reciben los datos, los métodos y medios que pueden emplear, la periodicidad, y la garantía de fiabilidad y confidencialidad. Consiste en la transmisión de una estructura o de un orden estructural de una computadora a otra, o de una parte a otra dentro de la misma computadora, otra vez sin tomar en cuenta al significado de la información. Así, la informática en sus albores sostiene que pueden transmitirse estructuras sin significado y que lo realmente importante es la transmisión de dichas estructuras. (Theodore, 1990)

1.1.4 Información

La información puede entenderse como la significación que adquieren los datos como resultado de un proceso consciente e intencional de tres elementos: los datos del entorno, los propósitos y el contexto de aplicación, así como la estructura de conocimiento del sujeto. (Angulo, 2005)

1.2 Sistemas para la Gestión de Currículos

1.2.1 Sistema de gestión curricular Institución Universitaria Esumer.

La Institución Universitaria ESUMER (Escuela Superior de Mercadeo) de Medellín, gracias al convenio con la Universidad de Pinar del Río de Cuba, creó el Sistema de Gestión Curricular para programas de pregrado y ciclos propedéuticos; que se convirtió en la carta de navegación para el desarrollo de los procesos básicos de la educación superior: Docencia, investigación y extensión.

Este Sistema se ejecuta desde el 2003 y propone la definición de una estrategia para la instrumentación del sistema de gestión del proceso curricular como aporte práctico, y de esta forma pueda ser introducida en los procesos académicos y administrativos de las instituciones de educación superior. (Ministerio de Educación Nacional Republica de Colombia)

1.2.2 CurrículumFácil

CurrículumFácil es un programa de diseño y redacción de currículos con las siguientes características:

- Propone el tipo de currículo, adecuándose a la situación personal.
- Sugiere frases y expresiones predeterminadas para definir habilidades.
- Ayuda durante el proceso de creación.
- Sugerencia de formas de redacción.
- Sugerencias de modelos de cartas de presentación.
- Incluye más de 25.000 modelos y plantillas de currículos.
- Gestión de envío de currículos y cartas de presentación.
- Aprobado por miles de usuarios a nivel mundial.

1.2.3 NitroCV

NitroCV es una aplicación que brinda ayuda para confeccionar y distribuir currículos. Permite crear un Currículum Vitae, publicarlo en Internet, enviarlo por correo electrónico, utilizar plantillas a la hora de crear

el currículum, almacenar un número ilimitado de currículum, entre otras funcionalidades.

NitroCV requiere para empezar crear un currículum vitae nuevo, una vez hecho se introducen todos los datos, basándose en unas pautas que el software tiene definidas, luego se selecciona una plantilla y se crea el currículum, siendo este publicado, creándose una página en Internet con el Currículum Vitae, funcionando al instante y con la posibilidad de ser actualizado, por otro lado también brinda la posibilidad de enviar el currículum directamente desde el programa NitroCV (requiere parámetros de configuración del correo electrónico que se utilice).

De manera general se puede decir que estos sistemas son usados con gran aceptación, tanto en el mundo como en nuestro país y en nuestro sistema de educación, porque realizan la gestión curricular de manera eficiente, llevando a cabo un conjunto amplio de funcionalidades, pero son demasiado generales, y para realizar la gestión de la información curricular de los profesores en la Universidad de las Ciencias Informáticas se necesita un sistema que gestione específicamente la información que se requiere en un currículum de un profesor de la UCI. Por lo que se hace necesaria la creación de un sistema que sea capaz de realizar acciones con el objetivo de gestionar la información curricular de los profesores de la Universidad, y más específicamente, de la facultad 1.

1.3 Herramientas y tecnologías posibles a utilizar

Para desarrollar el sistema se tiene que definir las herramientas y tecnologías a utilizar. Con este objetivo, se hace necesario realizar un estudio de las mismas, analizando sus características, ventajas y desventajas, así como las tendencias actuales en el desarrollo de sistemas, para definir cuáles serán las que se utilizarán para desarrollar el sistema. Teniendo en cuenta la política de desarrollo de la Facultad 1, se analizarán las herramientas y tecnologías de desarrollo que sean libres o permitan implementaciones sobre software libre⁵.

Una aplicación web es un software diseñado para automatizar procesos cuyo resultado se entrega a través de Internet o una Intranet, siendo un sistema Web donde la entrada de datos del usuario y la navegación que realice, afectan al estado del sistema. Una aplicación de este tipo brinda diferentes facilidades, como la conectividad en cualquier instante, la posibilidad de migrar de sistema operativo sin

⁵ Software Libre: (en inglés free software) es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

tener que afectar el funcionamiento de la aplicación, no se requiere gran cantidad de recursos, solo una computadora y un navegador, pueden utilizar la aplicación miles de usuarios al mismo tiempo, se requiere solo de un conocimiento básico de informática, tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de escritorio, no necesitan ser descargadas, instaladas o configuradas, por todas estas ventajas que brindan las aplicaciones web y debido a que el perfil de desarrollo de la Facultad 1 es el Desarrollo de Aplicaciones Web, se decidió realizar una aplicación de este tipo para resolver el problema existente.

1.3.1 Lenguajes para el desarrollo Web

El desarrollo de aplicaciones Web ha tenido un auge en todo el mundo gracias a las ventajas que las mismas ofrecen a empresas e instituciones. Para el desarrollo de estas existen numerosos lenguajes de programación que se dividen en dos grupos, el primer grupo abarca los lenguajes que corren en el lado del cliente y el segundo los lenguajes que corren en el lado del servidor. Las técnicas de desarrollo Web y lenguajes del lado del cliente más utilizados son HTML⁶ y JavaScript⁷, mientras que los lenguajes de programación del lado del servidor más usados en software libre son Java⁸ y PHP⁹.

1.3.1.1 Lenguajes del lado del servidor

Un lenguaje del lado del servidor es un lenguaje de programación que es reconocido, ejecutado e interpretado en el servidor, al cual el usuario no tendrá acceso, que procesa los datos mediante la interpretación de un script¹⁰ en el servidor para generar páginas web dinámicamente como respuesta, y

6 HTML: HyperText Markup Language(Lenguaje de Marcas de Hipertexto) es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con diferentes objetos. Es un lenguaje que hace posible presentar información en páginas web. (HTML.net) (ver 1.3.1.2.1 Hypertext Markup Language (HTML))

7JavaScript: Es un lenguaje de programación del lado del cliente que es interpretado en el navegador. (Maestros del web) (ver 1.3.1.2.2 JavaScript)

8 Java: Es un lenguaje de programación independiente de plataforma desarrollado por Sun Microsystems (Alvarez) (ver 1.3.1.1.1 Java)

9 PHP : Hypertext Preprocessors un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. (Van Der Henst, 2005) (ver 1.3.1.1.2 Hypertext Preprocessor (PHP))

10 Script: Es un programa usualmente simple, que generalmente se almacena en un archivo de texto plano.

que es independiente del navegador. Para definir un lenguaje del lado del servidor a utilizar se deben tener en cuenta la complejidad del mismo, si el lenguaje es capaz de resolver el problema, la explotación que hagan de la base de datos, los recursos que requiera, la velocidad de solución de problemas, entre otras características. (Álvarez) (de la Torre)

1.3.1.1.1 Java

Java es un lenguaje de programación sencillo, de propósito general, independiente de la plataforma de desarrollo, orientado a objetos, con una sintaxis muy parecida a la de C¹¹ o C++¹². Fue creado por la compañía Sun Microsystems a principios de los 90. Su principal ventaja es la independencia de la plataforma. Como principales desventajas de Java se encuentran que el rendimiento de una aplicación hecha en este lenguaje depende de la eficiencia del compilador, o de la Máquina Virtual de Java (JVM¹³). Java necesita que este programa esté instalado en la computadora, además, sus entornos de desarrollo consumen una gran cantidad de recursos de la computadora. (Gosling, y otros, 1999)

Características.

- Alto rendimiento.
- Indiferente a la arquitectura.
- Robusto.
- Interpretado y compilado a la vez
- Seguro.
- Distribuido.

1.3.1.1.2 Hypertext Preprocessor (PHP)

PHP fue ideado en el año 1994 por Rasmus Lerdorf. Sus primeras versiones, que no eran distribuidas a

11 C: Es un lenguaje de programación creado en 1972 por Kenneth L. Thompson, orientado a la implementación de Sistemas operativos. (Wikipedia La enciclopedia libre)

12 C++: Es un lenguaje de programación imperativo orientado a objetos derivado del C. (Ceballos Sierra, 2003)

13 JVM: Java Virtual Machine (Máquina Virtual de Java). Es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario. (Sun Microsystems)

los clientes fueron usadas en las páginas web que utilizaba para controlar el acceso a su currículum. La primera versión que estuvo disponible para el público, en 1995, se llamaba "Herramientas para páginas web personales"(Personal Home Page Tools).

PHP es un lenguaje de programación interpretado, que es ejecutado del lado del servidor, es utilizado para generar páginas web dinámicas, embebido en páginas html, y ejecutado en el servidor, aunque puede ser utilizado para realizar aplicaciones con una interfaz gráfica, utilizando las debidas extensiones del lenguaje. La mayoría de su sintaxis ha sido obtenida del lenguaje de programación C, Java y Perl¹⁴, aunque tiene algunas sintaxis características específicas de sí mismo. Es un lenguaje que tiene como objetivo principal permitir a los desarrolladores la generación dinámica de páginas web. Al ser ejecutado en el lado del servidor, nos brinda la posibilidad de acceder a los recursos que se encuentren en el servidor, como por ejemplo, una base de datos. PHP es un lenguaje independiente del navegador, o sea, el navegador no tiene que obligatoriamente soportar el lenguaje, pero para que las páginas PHP funcionen, el servidor donde se encuentran las páginas debe soportarlo.

Características.

Debido a su condición de ser Software Libre posee una gran cantidad de características, que lo hacen una potente herramienta para la creación de páginas web dinámica, y que lo convierten en uno de los lenguajes para generar páginas web más usado a nivel mundial. Dentro de las múltiples características de PHP se pueden definir como fundamentales las que siguen:

- Soporta una gran cantidad de bases de datos: MySQL¹⁵, PostgreSQL¹⁶, Oracle¹⁷, entre otras.
- Integrado con disímiles bibliotecas externas, brinda la posibilidad de generar documentos en formato PDF¹⁸, hasta el análisis de código XML¹⁹.

¹⁴ Perl: (Practical Extraction and Report Language) es un lenguaje de propósito general, interpretado, orientado a objeto, y enfocado al tratamiento de información y su presentación por pantalla u otro medio. (GENEURA)

¹⁵ MySQL: es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. (Schumacher, y otros) (ver 1.3.4.1 MySQL)

¹⁶ PostgreSQL: Es un Sistema de Gestión de Bases de Datos Objeto-Relacionales, considerado como la base de datos de código abierto más avanzada del mundo. (eAprende.com) (ver 1.3.4.1 PostgreSQL)

¹⁷ Oracle: Es una herramienta cliente/servidor para la gestión de Bases de Datos. (desarrolloweb.com)

-
- Provee al usuario una solución simple y universal para las paginaciones dinámicas fácilmente programable.
 - Su código es más fácil de mantener y poner al día que en otros lenguajes.
 - Es soportado por una gran comunidad de desarrolladores, como producto de código abierto²⁰ que es, teniendo la ayuda de un grupo grande de programadores, permitiendo esto encontrar y reparar los fallos de funcionamiento.
 - El código se actualiza constantemente con extensiones y mejoras continuas del lenguaje.
 - A través del uso de PHP, se pueden realizar diferentes funcionalidades, como por ejemplo el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.
 - Ofrece gran seguridad, debido a que puede acceder a ficheros, ejecutar comandos y abrir conexiones de red en el servidor.
 - Existen diferentes opciones de configuración para controlar su funcionamiento.
 - Muy fácil de aprender.
 - Soporte multiplataforma de diversos Sistemas Operativos, así como servidores http²¹ y de base de datos.
 - Se distribuye de forma gratuita bajo una licencia gratuita.
 - Posee la capacidad de expandir su potencial, a través de la gran cantidad de módulos, llamados ext's o extends.
 - Posee una gran documentación en su sitio oficial, y en otros sitios de la comunidad de desarrollo.

¹⁸ PDF (del inglés Portable Document Format) es un formato de documento portátil de almacenamiento de documentos. (Adobe y PDF)

¹⁹ XML: eXtensible Markup Language, o lenguaje de anotación extensible, es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium. (GENEURA)

²⁰ código abierto: Es un software que pone a disposición de cualquier usuario su código fuente. (Informática Hoy)

²¹ http: *HyperText Transfer Protocol* es el protocolo usado las transferencias de datos de la Web. (Sun MycroSystems)

- Soporta programación orientada a objetos.
- Posee una biblioteca nativa de funciones sumamente amplia e incluida.

Una vez analizados los lenguajes de programación del lado del servidor, es necesario hacer una selección de cuál utilizar en el desarrollo del sistema. Para la implementación del sistema se usará PHP ya que posee una gran comunidad de desarrollo, la cual implementa constantemente mejoras en su código, en la Universidad de las Ciencias Informáticas este lenguaje es muy utilizado por un gran número de programadores, además de que el mismo es software libre, permite técnicas de programación orientada a objetos, puede ser ejecutado en la mayoría de los sistemas operativos, permite la combinación con un gran número de servidores de bases de datos entre los cuales se encuentran los posibles a utilizar, lo cual le es de gran beneficio para los programadores, y fundamentalmente, porque el servidor Apache, en el cual es interpretado el código de PHP es uno de los más rápidos del mundo y porque PHP es un lenguaje pensado para dar solución rápida a los problemas. Por su parte, Java es un lenguaje más pesado, pensado para realizar proyectos complejos y que no requieran de una solución inmediata, además, el código de Java puede llegar a ser redundante, debido a las frecuentes declaraciones de tipos y conversiones de tipo manual, sus entornos de desarrollo consumen menos recursos que los de Java, y es necesario tener la máquina virtual instalada para que corra una determinada aplicación.

1.3.1.2 Lenguajes del lado del cliente

Un lenguaje del lado del cliente es un lenguaje que es reconocido e interpretado por el navegador, al cual el cliente tiene acceso pudiendo observar el código en cualquier momento.

1.3.1.2.1 Hypertext Markup Language (HTML)

Hypertext Markup Language es un lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertexto, es el formato estándar de las páginas Web, gracias a Internet y a los navegadores como Internet Explorer, Opera, Firefox o Netscape²² se ha convertido en uno de los formatos más populares que existen para la construcción de documentos. Para crear una página Web se pueden utilizar varios programas especializados en la creación de páginas, como por ejemplo, el Microsoft FrontPage²³,

22 Internet Explorer, Opera, Firefox y Netscape: Herramientas para navegar en internet.

23 Microsoft FrontPage :Es una herramienta de construcción y edición de páginas web

Macromedia Dreamweaver²⁴ o cualquier editor de textos básico.

Características.

- Información por hipertexto: Diversos elementos (texto o imágenes) de la información que se muestra en la pantalla están vinculados con otras informaciones que pueden ser de otras fuentes.
- Gráfico: En la pantalla aparece simultáneamente texto, imágenes e incluso sonidos.
- Global: Se puede acceder a él desde cualquier tipo de plataforma, usando cualquier navegador y desde cualquier parte del mundo.
- Pública: Toda su información está distribuida en miles de ordenadores que ofrecen su espacio para almacenarla. Toda esta información es pública y toda puede ser obtenida por el usuario.
- Dinámica: La información, aunque esta almacenada, puede ser actualizada por el que la publicó sin que el usuario deba actualizar su soporte técnico.
- Independiente: Dada la inmensa cantidad de fuentes, es independiente y libre.

1.3.1.2.2 JavaScript

JavaScript es un lenguaje de programación del lado del cliente, que es interpretado, lo cual significa que no requiere compilación. Es utilizado generalmente en páginas Web, siendo embebido dentro del código HTML o similares. La mayoría de los navegadores más usados son capaces de interpretar el código JavaScript incluidos en las páginas Web. Este lenguaje es basado en prototipos, pues las clases nuevas se generan clonando las clases base (prototipo), y extendiendo sus funcionalidades.

El JavaScript es sumamente utilizado en el mundo del desarrollo Web por ser muy versátil y potente, tanto para la realización de pequeñas tareas como para la gestión de complejas aplicaciones. Con su utilización podemos evitar la sobrecarga del lado del servidor ya que funciones como la validación de formularios, que antes se hacían del lado del servidor, se pueden realizar con este lenguaje y de una manera rápida.

1.3.1.2.3 Asynchronous JavaScript And XML (AJAX)

²⁴ Macromedia Dreamweaver: Es una herramienta de diseño de páginas web.

El término AJAX se presentó por primera vez en el artículo "Ajax: A New Approach to Web Applications" publicado por Jesse James Garrett el 18 de Febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación web que estaba apareciendo.

Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.

Las tecnologías que forman AJAX son:

- XHTML²⁵ y CSS²⁶, para crear una presentación basada en estándares.
- DOM²⁷, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT²⁸ y JSON²⁹, para el intercambio y la manipulación de información.
- XMLHttpRequest³⁰, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano, realizándose una creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

25 XHTML: eXtensible Hypertext Markup Language Es el lenguaje de marcado pensado para sustituir a HTML como estándar. (Sun Microsystems)

26 CSS: Cascading Style Sheets Es el lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML. (WebEstilo)

27 DOM: Document Object Model Es una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML. (Sun Microsystems)

28 XSLT: Extensible Stylesheet Language Transformations: Es un estándar que presenta una forma de transformar documentos XML en otros formatos. (Sun Microsystems)

29 JSON: JavaScript Object Notation: Es un formato ligero para el intercambio de datos. (JSON)

30 XMLHttpRequest: Es una interfaz empleada para realizar peticiones a servidores WEB. (Sun Microsystems)

1.3.2 Frameworks

Un framework es un conjunto de funciones, procedimientos y herramientas destinadas a la construcción de un determinado tipo de aplicaciones de manera generalista, es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Los frameworks hacen el trabajo más fácil, presentando funcionalidades y componentes reutilizables, que brindan mayor eficiencia a la hora de desarrollar una aplicación.

1.3.2.1 Extend JS (Ext)

Ext JS es un framework libre y de código abierto de presentación para PHP. Ext basa toda su funcionalidad en JavaScript a través de diferentes librerías. Presenta ventanas, paneles, tablas, componentes de almacenamiento y agrupamiento, seleccionadores de fechas, mensajes emergentes y un sinnúmero de utilidades que son creadas en tiempo de ejecución, creándose además todos los objetos HTML, a través del uso intenso del DOM.

Dentro de las principales características de Ext, se encuentran:

- Controles de entrada de campos y áreas de texto.
- Campos de fecha, con seleccionador de fechas incluido.
- Campos numéricos.
- Listbox y combobox.
- Control de edición HTML.
- Control de Grids³¹.
- Control de árbol.
- Paneles.
- Barra de herramientas.
- Menús con estilo de aplicaciones de escritorio.

Ventajas:

- La orientación a objetos intensa hacen modular todos sus scripts.
- El diseño está completamente separado de la funcionalidad.

- Funciones comunes como validación, comboboxes editables, ventanas arrastrables (con minimizar y maximizar), paneles editables, son muy fáciles de implementar.
- Amplia documentación, así como también una gran comunidad de desarrollo.

Este framework se utilizará para la realización de las interfaces que interactuarán con los usuarios, debido a las funcionalidades que brinda, y que es una tecnología nueva que proporciona facilidad de desarrollo para aplicaciones web, así como la rapidez del manejo de datos, debido a que utiliza la tecnología Ajax para el envío y recibo de datos.

1.3.2.2 CodeIgniter.

CodeIgniter es un framework libre y de código abierto para desarrollar aplicaciones web utilizando el lenguaje PHP. Basado en el patrón de desarrollo Modelo Vista Controlador (MVC), permite que las páginas web tengan la menor cantidad de código interpretado, debido a que la presentación está separada del código PHP. El modelo representa la estructura de datos, que contendrán funciones para añadir, modificar o eliminar datos, así como cargar los diferentes formularios y tablas. La vista es la información mostrada al usuario, la cual será una interfaz diseñada con Ext. El controlador es el intermediario entre la vista y el modelo. Además, CodeIgniter permite la creación de aplicaciones web sin la necesidad de tener un modelo, solamente con vistas y clases controladoras.

CodeIgniter es un framework verdaderamente ágil y rápido en la solución de problemas debido a que el núcleo necesita muy pocas librerías, a diferencia de otros frameworks que necesitan demasiados recursos. Las librerías adicionales son cargadas dinámicamente cuando se solicitan, basándose en la necesidad de un proceso determinado. Además, cuenta con librerías que permiten realizar tareas comunes en el desarrollo de aplicaciones web, como el acceso a la base de datos, enviar correos electrónicos, validaciones de datos, el mantenimiento de sesiones, manipular imágenes entre otras.

CodeIgniter cuenta con una amplia comunidad de desarrolladores, y además con una excelente guía de usuario donde se explican detalladamente cómo se realizan los procesos en el mismo, y cómo utilizar cada funcionalidad. Es un framework ideal para aplicaciones web que no cuenten con un negocio muy complejo y que necesiten solución rápida de problemas.

³¹ Grid: Tabla en la que se muestran los datos usando Ext.

CodeIgniter se utilizará para el manejo de los procesos de la aplicación. Debido a que se rige por el patrón MVC, se pueden definir fácilmente las vistas, los modelos y el controlador correspondiente del sistema, es un framework de fácil entendimiento, con una guía de usuario muy amplia, es una tecnología nueva que es libre y además, cuenta con las librerías, procedimientos y funcionalidades necesarias para resolver el problema en cuestión.

1.3.3 Content Management System (CMS)

Los CMS son muy usados actualmente en el desarrollo de aplicaciones web, debido a que brindan un conjunto amplio de funcionalidades, que se encuentran ubicadas en módulos dentro de los CMS. Por lo tanto, como posibles herramientas a utilizar para el desarrollo del sistema, también se analizarán los CMS de código abierto, y debido a que son los más usados a nivel mundial, en Cuba y específicamente, en la UCI, se analizarán Drupal y Joomla.

1.3.3.1 Drupal

Drupal es un poderoso CMS ampliamente conocido en el mundo entero debido a la seguridad que brinda, así como por la calidad en su código. Este CMS, en lugar de almacenar el contenido en archivos que sean estáticos en los ficheros del servidor, de forma fija, el contenido textual de las páginas se almacena en una base de datos, así como de otras configuraciones, haciendo de Drupal un sistema para la administración de contenido muy estable, dinámico y de actualización continua.

Drupal es un Sistema de Administración de Contenido de código abierto, por lo tanto, tiene funcionalidades incorporadas, así como un conjunto de módulos disponibles que son añadidos libremente, incluyendo la administración de listas de correo, sistemas de comercio electrónico y galería de fotos. Permite a los usuarios la publicación de artículos, fotos y diversos archivos, además de contar con servicios como foros, encuestas, boletines de noticias, votaciones, administración de usuarios permisos y blogs. En diversas ocasiones Drupal es usado para la creación o modificación de diferentes sitios Web, como son los portales de la comunidad, sitios de discusión, corporativos, blogs personales, sitios sociales del establecimiento de una red, etc.

1.3.3.2 Joomla

Joomla es un Sistema de Gestión de Contenido de código abierto, que ha sido desarrollado bajo la licencia GPL, y que es uno de los más usados a nivel mundial para desarrollar sitios web, debido a su

característica de permitir la creación de estos sitios de forma dinámica e interactiva. Su principal objetivo es brindar una solución a las necesidades de todos y cada uno de los integrantes de un proyecto.

Joomla permite implementar componentes como salas de conversación y encuestas. Incluye hacer caché de páginas para mejorar el rendimiento, permite además realizar indexamiento Web, soporta RSS, incluye versiones imprimibles de páginas, flash con noticias, búsqueda en el sitio Web, calendarios e internacionalización del lenguaje. Utiliza además, diferentes formas de comunicación para su desarrollo, como son: encuentros por medio de IRC, foros de Discusión, wikis, blogs y listas de correo.

No se utilizará un CMS debido a que no existe un módulo para diseñar la entrada de datos adecuadamente, y conforme con la información que se requiere en un currículo. Para ello, se debería crear un módulo para el CMS, lo cual se torna complicado, ya que se dificulta el aprendizaje del funcionamiento interno de los CMS, siendo más fácil de aprender el lenguaje PHP y los frameworks propuestos, además de que con esta tecnología se pueden personalizar más fácilmente los componentes y las interfaces que interactuarán con el usuario.

1.3.4 Sistemas Gestores de Base de Datos (SGBD)

Los sistemas gestores de bases de datos son un tipo de software muy específico que sirve de interfaz entre el usuario, las aplicaciones y la base de datos. Están compuestos por un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Su propósito general es manejar de forma sencilla y ordenada los datos que posteriormente se convertirán en información relevante para el manejo correcto de información. Los objetivos fundamentales de los SGBD son: independencia de los datos y los programas de aplicación, minimización de la redundancia, integración y sincronización de las bases de datos, integridad de los datos, seguridad y protección de los datos, facilidad de manipulación de la información y control centralizado, entre los más usados mundialmente están: Oracle, PostgreSQL y MySQL.

1.3.4.1 MySQL

Cuenta con más de seis millones de instalaciones, es multi-hilo, multiusuario, rápido y bastante robusto. Es propiedad de la empresa MySQL AB, que en el año 2008 pasó a ser una subsidiaria de Sun

MicroSystem. Por un lado se ofrece bajo la GNU GPL³² para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Es un SGBD muy utilizado en el mundo, y es muy popular por estar ampliamente ligado a PHP. Funciona sobre múltiples plataformas, incluyendo GNU/Linux, Mac OS X, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows.

Características.

- Mediante la verificación basada en el host y tráfico de contraseñas encriptado ofrece un sistema de contraseñas y privilegios seguro, denotando cualidades de seguridad altamente confiables.
- Es capaz de soportar gran cantidad de datos.

1.3.4.1 PostgreSQL

Es un SGBD relacional de software libre y orientado a objetos, liberado bajo la licencia BSD³³. Posee las características de los más potentes sistemas comerciales como Oracle o SQL Server, con la ventaja de que su licencia es gratuita. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales. Dicha comunidad es denominada PostgreSQL Global Development Group (PGDG). Tiene más de 15 años de desarrollo activo y se ha ganado la reputación de ser confiable y mantener la integridad de los datos. Se ejecuta en la mayoría de los sistemas operativos más utilizados en el mundo como Linux, varias versiones de UNIX y Windows.

Ventajas:

- Alta concurrencia.

³² GNU GPL: GNU *General Public License* (Licencia Pública General de GNU): Licencia que se aplica a programas software libre, para protegerlos de intentos de apropiación que restrinjan la libertad de libre distribución, modificación y uso a los usuarios. (Wikipedia La enciclopedia libre) (GNU Operating System)

³³ Licencia BSD: permite el uso y distribución de PostgreSQL sin costo, tanto para aplicaciones comerciales como no comerciales. (PostgreSQL)

-
- Amplia variedad de tipos nativos
 - Multiplataforma
 - Extensible
 - Estabilidad y Confiabilidad
 - Instalación ilimitada
 - Diseñado para ambientes de alto volumen

De los dos SGBD analizados, teniendo en cuenta la política de desarrollo de la Facultad 1 de desarrollar sistemas con herramientas software libre, y además que desde el año 2008, la empresa MySQL AB pasó a ser una subsidiaria de la empresa productora de software privativo Sun Microsystems, y ésta, a su vez, de Oracle Corporation, desde abril de 2009, desarrollando MySQL en un esquema de licenciamiento dual, se define PostgreSQL como SGBD para desarrollar el sistema.

1.3.5 Metodologías de desarrollo del software

La Ingeniería del Software es una tecnología multicapa en la que se pueden definir los métodos, el proceso y las herramientas. La Ingeniería del Software surge con el principal objetivo de crear y mantener aplicaciones software, aplicando tecnologías y prácticas computacionales. Su constante desarrollo ha derivado la realización de varias tareas en este campo, como son: análisis de requisitos, especificación, diseño y arquitectura, programación, prueba, documentación y mantenimiento.

Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto. El proceso de desarrollo del software, define el conjunto de actividades precisas para convertir los requisitos de los usuarios en el conjunto seguro y resistente de artefactos que componen un producto de software. Existen dos vertientes fundamentales de acuerdo con el método de desarrollo, los cuales están dirigidos a beneficiar la labor de los desarrolladores de software: los métodos ágiles y los métodos pesados.

Las metodologías ágiles proponen mejorar la calidad del producto software a través de la comunicación inmediata y directa, mientras que las metodologías pesadas proponen que sea a través del orden y la documentación.

Entre las metodologías de desarrollo de software más utilizadas a nivel mundial, en nuestro país, y específicamente en la UCI, se pueden encontrar:

- eXtreme Programing (XP)
- Rational Unified Process (RUP)
- SCRUM

Para desarrollar el sistema que se quiere desarrollar es necesario definir la metodología a utilizar, y es por ello que hay que analizar cada una de ellas, teniendo en cuenta sus características, ventajas y desventajas, y su adecuación al desarrollo del sistema.

1.3.5.1 Programación Extrema (XP)

XP, del inglés eXtreme Programing, es una metodología ágil, que se centra en el aumento de las relaciones interpersonales para lograr un desempeño exitoso en el desarrollo del software, donde el trabajo en equipo es muy importante, preocupándose por la existencia de un ambiente de trabajo óptimo, y porque los desarrolladores realicen un buen aprendizaje. Se basa en el trabajo orientado directamente al objetivo, teniendo en cuenta para esto la velocidad de reacción para la implementación y para los cambios que puedan surgir durante el desarrollo del proceso, además se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Es el más destacado de los procesos ágiles de desarrollo de software.

XP minimiza el riesgo de fallo del proceso manteniendo dentro del equipo a un representante competente del cliente, este representante es quién responderá a todas las preguntas y dudas que surjan por parte del equipo de desarrollo durante el proceso, de forma que no se retrase la toma de decisiones. XP define tiene cuatro variables principales, que son: Coste, Tiempo, Calidad y Alcance, y como fases se definen: Planificación, Diseño, Desarrollo y Pruebas. Para su uso, XP define varios roles, como son: Programador, Cliente, Encargado de Pruebas, Encargado de seguimiento, Entrenador, Consultor y Gestor.

XP se basa en historias de uso (UseStories), estas historias las escribe el cliente o su representante dentro de equipo y describen los escenarios claves del funcionamiento del software, a partir de estas se generan las entregas (releases) entre el equipo y el cliente.

Una característica fundamental de XP, es que el código siempre se produce en parejas, parejas que van

cambiando constantemente para lograr así que todo el equipo sepa y pueda modificar según necesidades el código generado, esto logra en el equipo que los integrantes aprendan entre sí y compartan todo el código.

1.3.5.2 SCRUM

SCRUM define un marco para la gestión de proyectos, que se ha utilizado exitosamente durante los últimos diez años. Dentro de sus principales características se encuentran:

- Equipos auto dirigidos.
- Utiliza reglas para crear un entorno ágil de administración de proyectos.
- No prescribe prácticas específicas de ingeniería.
- Los requerimientos se capturan como ítems de la lista reserva del producto.
- El producto se construye en una serie de sprints de un mes de duración.
- Usado para proyectos complejos con requerimientos cambiantes.
- Basado en un control de proceso empírico.

1.3.5.3 Proceso Unificado del Software (RUP)

El Proceso Unificado del Software (RUP), es una metodología de desarrollo de software, pesada y orientada a objetos, es un marco de trabajo genérico que puede ser especializado para una gran variedad de software para distintas áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.

RUP está basado en componentes interconectados a través de interfaces y utiliza el UML³⁴ como lenguaje de modelado de procesos, es dirigido por casos de uso³⁵, centrado en la arquitectura e iterativo e incremental. Dirigido por casos de uso, porque los casos de uso reflejan lo que los usuarios futuros

³⁴UML: Unified Modeling Language(Lenguaje Unificado de Modelado): Es un lenguaje de modelado para visualizar, especificar, construir y documentar un sistema. (ver 1.3.7 Lenguaje de Modelado Utilizado)

³⁵ Caso de uso: Es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. (Jacobson, y otros, 2004)

desean y necesitan, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos, a partir de ahí, los casos de uso guían el proceso de desarrollo. Centrado en la arquitectura, porque la arquitectura muestra una visión común del sistema completo, en la que el equipo del proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes producirlos económicamente. Iterativo e incremental, porque RUP propone que cada fase se desarrolle en iteraciones, y cada iteración tiene que proponerse un incremento en el proceso de desarrollo del software.

RUP define como sus principales elementos:

Trabajadores (quién): Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto con un equipo.

Actividades (cómo): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.

Artefactos (qué): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

Flujo de Actividades (cuándo): Secuencia de actividades realizadas por los trabajadores y que produce un resultado de valor observable.

Las actividades son agrupadas en grupos lógicos, donde se definen 9 flujos de trabajo fundamentales, de los cuales solo los primeros 6 son considerados flujos de ingeniería, y los 3 restantes y últimos como flujos de apoyo. Estos flujos son: Modelado de Negocio, Requerimientos, Análisis y Diseño, Implementación, Pruebas, Despliegue, Configuración y Administración de Cambios, Administración de Proyecto y Entorno. Además, RUP define cuatro fases importantes en el desarrollo, las mismas son: Conceptualización (Concepción o Inicio), la cual describe el negocio y delimita el proyecto, se definen sus alcances con la identificación de los casos de uso del sistema; Elaboración, donde se define la arquitectura que tendrá el sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen; Construcción, donde se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario, se obtienen una o más entregas del producto que han pasado las pruebas y por último, pero no menos importante, la fase de Transición, en la cual el release ya está listo para su instalación en condiciones reales.

En RUP existen algunos principios fundamentales de desarrollo de software que son:

Adaptar el proceso:

Lo que se refiere a que los procesos deben de adaptarse al tamaño de los proyectos o de la organización.

Balancear prioridades:

Se debe de encontrar un balance que satisfaga los deseos de todos.

Colaboración entre equipos:

Los proyectos de desarrollo de software no son llevados a cabo por una sola persona, sino varias o varios equipos de desarrollo, los cuales deben de contar con una buena comunicación para que esto les permita coordinar esfuerzos.

Demostrar valor iterativamente:

Los proyectos aunque sea de manera interna deben de entregarse de manera iterada en cada iteración se analiza el avance, estabilidad, calidad del producto.

Elevar el nivel de abstracción:

Esto previene a los ingenieros de software ir directamente de los requerimientos del cliente a la codificación, un nivel alto de abstracción permite discusiones sobre diversos niveles de arquitectura, los cuales se pueden acompañar por representaciones visuales de la arquitectura como por ejemplo utilizando UML.

Enfocarse a la calidad:

El control de calidad deberá de ser llevado a cabo no solo al final de cada iteración sino a todo lo largo de toda la producción.

En el desarrollo del sistema no se utilizará la metodología XP debido a que no se cuenta con la participación directa del cliente dentro del equipo de desarrollo, además, porque no se cuenta con el personal necesario para realizar la programación en pareja, dos de los principales principios de XP. SCRUM no se utilizará debido a que no prescribe prácticas específicas de ingeniería, lo cual se interpone a la organización de los procesos a automatizar, a la hora de implementar, además, porque no brinda la organización del trabajo que se requiere, ni la documentación ideal una vez terminado el producto. Para el

desarrollo del sistema se utilizará la metodología RUP, debido a la organización que brinda, así como la documentación que genera en cada iteración. Esta metodología, brinda una excelente organización y planificación, al dividir cada etapa por flujos de trabajo, donde los casos de uso y la arquitectura son los pilares fundamentales, realizando diferentes iteraciones que mejoran constantemente el producto.

1.3.6 Posibles Herramientas CASE a utilizar para el modelado

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas sirven de ayuda en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, calculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. (Clase Teórico Práctica # 1 Ingeniería de Software I, 2008)

1.3.6.1 Rational Rose

Racional Rose es una herramienta CASE utilizada para el modelado visual. Permite completar los flujos fundamentales de RUP, incluye además herramientas de ingeniería inversa y generación de código. Está entre las mejores herramientas para traducir requisitos de alto nivel a una arquitectura basada en componentes. Se ha convertido en una de las mejores opciones por la notación estándar que brinda para especificar, visualizar y construir productos software y sistemas, debido a que está en la avanzada en cuanto al desarrollo de UML.

1.3.6.2 Visual Paradigm

El Visual Paradigm es un software gratuito para el modelado UML. Presenta características gráficas muy cómodas para el usuario, haciendo más fácil el modelado de diagramas que sigue el estándar de UML, los cuales son: Diagramas de clase, Casos de Uso, Comunicación, Secuencia, Estado, Actividad, Componentes, etc.

Ventajas de Visual Paradigm:

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.

-
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
 - Capacidades de ingeniería directa e inversa.
 - Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
 - Disponibilidad de múltiples versiones, para cada necesidad.
 - Disponibilidad en múltiples plataformas.
 - Producto de calidad.
 - Soporta aplicaciones Web.
 - Varios idiomas.
 - Generación de código para Java y exportación como HTML.
 - Fácil de instalar y actualizar.
 - Compatibilidad entre ediciones.

Teniendo en cuenta la política de desarrollo de la Facultad 1, se define como herramienta Case para el desarrollo del sistema, Visual Paradigm, debido a las características antes mencionadas, y a que una de sus características principales de ser un software libre para el modelado.

1.3.7 Lenguaje de Modelado Utilizado

Un lenguaje de modelado es un lenguaje para modelar sistemas y procesos, de manera que se pueda comunicar las ideas a otros desarrolladores, representar de forma más simplificada la realidad y servir de apoyo en los procesos de análisis de un problema. Para facilitar este modelado, se realiza una abstracción y se plasma en una notación gráfica.

1.3.7.1 Lenguaje Unificado de Modelado (UML)

UML, del inglés Unified Modeling Language, es un lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software. Permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos, haciéndolo el estándar de la industria, sus creadores son los autores de las tres metodologías orientadas a objetos más populares de la primera generación del análisis y diseño orientado a objetos. La finalidad de UML es describir modelos de sistemas, ya sea del

mundo real o del mundo del software, basándose en la metodología orientada a objetos. UML está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Permite establecer la serie de requerimientos y estructuras necesarias para diseñar un sistema de software previo al proceso de escribir código. Para el modelado del sistema se utilizará este lenguaje de modelado, por ser el más usado en nuestro país y en la UCI, y por ser el que más se utiliza a nivel universal, debido a la gran cantidad de ventajas que presenta.

1.3.8 Servidor de aplicaciones WEB

Un servidor web es un programa que implementa el protocolo HTTP y está diseñado para transferir hipertextos, páginas web o páginas HTML, textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

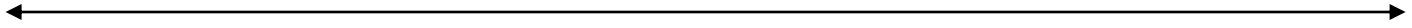
Es un programa que se ejecuta continuamente en un ordenador, manteniéndose a la espera de peticiones por parte de un cliente y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

1.3.8.1 Apache

El servidor HTTP Apache es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1, esta particularidad de correr sobre casi todos los Sistemas Operativos lo hace prácticamente universal.

Apache es un servidor altamente configurable de diseño modular. Tiene una alta configurabilidad en la creación y gestión de logs, ya que permite la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor. Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos. Entre sus características más importantes se destacan:

- Es multiplataforma
- Es un servidor de web conforme al protocolo HTTP/1.1



- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API³⁶ de programación de módulos, para el desarrollo de módulos específicos.
- Basado en hebras en la versión 2.0
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta
- Extensible: Gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

1.3.8.2 Wamp

Esta herramienta no es más que una suite de servidores para aplicaciones web que contiene al servidor web Apache, fácil de usar y que además permite crear aplicaciones web con el servidor Apache, PHP y base de datos MySQL. Además, contiene herramientas para administrar fácilmente la base de datos, como son PHPMyAdmin y SQLiteManager.

Con esta herramienta se puede administrar fácilmente los servicios de Apache y MySQL, se puede dar acceso a usuarios externos o solo a usuarios locales, permitiendo cambiar de estado (online u offline). Además, permite administrar la configuración del servidor, acceder a los registros de actividades, a los archivos de configuración y crear alias.

Esta es la herramienta a utilizar para el desarrollo del sistema, ya que contiene al servidor apache, así como las herramientas necesarias para administrar las aplicaciones y la base de datos.

1.3.9 Patrón de arquitectura

Para desarrollar el sistema es necesario además de las herramientas y las tecnologías, definir la

³⁶ API (Application Programming Interface): Interfaz de Programación de Aplicaciones, es el conjunto de funciones, procedimientos o métodos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. (Sun Microsystems)

arquitectura que se utilizará. En este caso para el desarrollo del sistema de gestión de información curricular de los profesores se define el uso del patrón de arquitectura Modelo Vista Controlador (MVC), debido a que el framework principal que se utilizara (CodeIgniter) utiliza este patrón para la comunicación de las diferentes capas de su aplicación, y a que el sistema será una aplicación web, siendo MVC un patrón ideal para aplicaciones de este tipo.

MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el SGBD y el controlador representa la lógica de negocio.

- Modelo: Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.
- Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

1.3.10 Entorno de desarrollo integrado (IDE).

Como entorno de desarrollo se utilizará Zend Studio for Eclipse, el cual es un entorno de desarrollo que combina la tecnología Zend Studio con las herramientas de desarrollo de PHP con Eclipse³⁷, siendo una de las herramientas más potentes para el desarrollo de aplicaciones Web a nivel mundial. Esta herramienta fue diseñada por desarrolladores de PHP.

Este entorno de desarrollo presenta las siguientes características:

- Refactorización de código.
- Generación de código.
- Asistentes de ayuda al usuario.

³⁷ Eclipse: Es un entorno de desarrollo integrado de código abierto, usado para desarrollar entornos de desarrollo integrados. (IBM)


- Cubrimiento de código.
- Soporta pruebas de unidad de PHP.
- Editor de PHP con formato avanzado.
- Soporta JavaScript.
- Soporta HTML incluyendo HTML WYSIWYG³⁸.
- Soporta Control de Versiones con Historia Local.
- Soporte de diferentes frameworks, plantillas de código, vistas MVC, entre otros.
- Acceso al ecosistema de plug-ins³⁹ de Eclipse.
- Soporta desarrollo multi-lenguaje.
- Presenta herramientas de migración.
- Mecanismo de actualización automática.

Conclusiones

En este capítulo se analizaron los antecedentes de los sistemas de gestión de información curricular, tanto los nacionales, como los extranjeros, que son los antecedentes del futuro sistema que se quiere implementar. Se llegó a la conclusión que los sistemas existentes no se adecúan a lo que se quiere hacer en la facultad 1, por lo que es necesario la concepción e implementación de un sistema que sea capaz de gestionar correctamente este tipo de información, información que es de gran utilidad para la toma de decisiones. Además, se analizaron las diferentes metodologías y tecnologías posibles a utilizar para el desarrollo del sistema. Y teniendo en cuenta la política de la Facultad sobre el uso de Software Libre, el sistema será desarrollado utilizando el lenguaje PHP como lenguaje del lado del servidor, para la presentación se utilizará el framework EXT JS, y CodeIgniter como framework para manejar las vistas, los

³⁸ WYSIWYG: *What You See Is What You Get*. (Lo que ves es lo que obtienes) Son tipos de herramientas que permiten escribir la página sobre una vista preliminar similar a la de un procesador de textos. (Creating Web 2.0-Enabled Communities With Sun Java System Portal Server)

³⁹ Plug-in: Es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande. (masadelante.com)



controladores y los modelos. Como servidor web se utilizará el Servidor Web Apache, a través del entorno de desarrollo web WAMP, y como servidor de base de datos el PostgreSQL. Para la ingeniería se utilizará la metodología de desarrollo RUP, como herramienta CASE el Visual Paradigm Enterprise Edition, para la modelación de los procesos se utilizará el lenguaje de modelado UML, como patrón de arquitectura se utilizará MVC y como entorno de desarrollo el Zend Studio for Eclipse.

Capítulo 2: Características del Sistema

Introducción.

En este capítulo se describen formalmente el problema científico y la situación problemática, su entorno y el objetivo de la automatización. Se propone una solución al problema, presentando la propuesta del sistema, y describiéndose además los artefactos generados en la modelación del negocio, levantamiento de requerimientos, específicamente: modelo de dominio o conceptual, descripción de los requerimientos funcionales y no funcionales, y definición de actores⁴⁰ y casos de uso del sistema. El resultado de este capítulo es de gran importancia para el posterior análisis y el diseño del sistema.

2.1 Información que se maneja

A la hora de tomar decisiones para asignar una tarea determinada, es necesario conocer la trayectoria de los profesores, en su carrera como profesional de la educación. Para esto se hace necesario un currículum vitae, que contenga la información necesaria para, basándose en la misma, decidir quién es la persona adecuada para cada una de las tareas que tiene la facultad, tanto en la producción, como en la investigación y formación.

El currículo de un recién graduado se va llenando a medida que se realicen actividades en su vida docente, sin embargo, los profesores que vienen de otras instituciones ya traen una trayectoria profesional, y a veces esta información es importante y no se tiene en cuenta. Por lo que se hace necesario que el sistema controle determinado tipo de información, que será útil en la toma de decisiones, y que formará parte del currículo enriquecido del profesor.

El sistema manejará la información curricular de los profesores, la cual es amplia y muy variada, por lo que se dividió en dos grupos, la información personal del profesor, que además de los datos personales, contendrá una breve descripción profesional del profesor, y la información curricular que contendrá toda la información de las actividades curriculares en las que el profesor ha estado inmerso. La información que manejará el sistema es la siguiente:

⁴⁰ actores: todo lo que interactúa con el sistema, pueden ser personas u otros sistemas.

2.1.1 Información personal del profesor

- 1- Nombre.
- 2- Apellidos.
- 3- Fecha de nacimiento.
- 4- Especialidad de graduación.
- 5- Año de graduación.
- 6- Dirección Particular.
- 7- Teléfono.
- 8- Correo electrónico.
- 9- Breve descripción profesional.

2.1.2 Información del currículum

- 1- Categoría Docente.
- 2- Categoría Científica.
- 3- Cursos de Pregrado Impartidos.
- 4- Cursos de Postgrado.
 - 4.1 Impartidos.
 - 4.2 Recibidos.
- 5- Tesis Tutoradas.
- 6- Publicaciones Nacionales.
- 7- Publicaciones Internacionales.
- 8- Eventos Internacionales.
- 9- Eventos Nacionales.
- 10- Idiomas.
- 11- Participación en Proyectos Nacionales.

- 12- Participación en Proyectos Internacionales.
- 13- Misiones Cumplidas.
- 14- Productos Patentados.
- 15- Premios Obtenidos en Eventos Deportivos y Culturales.
- 16- Premios Obtenidos en eventos de Ciencia y Técnica.
- 17- Investigaciones.
- 18- Responsabilidades ocupadas.
- 19- Asociaciones Profesionales.
- 20- Otros.

2.2 Propuesta del Sistema

El sistema propuesto debe manejar la información curricular de los profesores de manera eficiente, permitiendo al profesor crear su currículum, y una vez que este sea creado, actualizarlo cada vez que lo estime conveniente. Debe garantizar la seguridad de la información manejada, ya que esta puede servir para tomar decisiones importantes en la Universidad. Debe además, permitir generar reportes con la información guardada en el currículum de los profesores, así como la búsqueda de profesores con determinadas características.

El sistema además, debe garantizar la eficiencia de su funcionamiento, realizando las tareas requeridas en tiempo mínimo para evitar esperas innecesarias por parte de los usuarios. El sistema presentará al usuario una interfaz amigable y fácil de usar, respetando al máximo la experiencia del usuario, siendo esta interfaz diseñada para ser utilizada por usuarios de poca experiencia informática, permitiendo el aprendizaje rápido de la misma.

2.3 Modelo de Dominio

Debido a que no existe un proceso de negocio bien definido con respecto al currículum de los profesores en la UCI, se lleva a cabo el modelo de dominio o modelo conceptual, donde se explican los principales conceptos y eventos que suceden en el entorno en el que se trabaja. De esta manera se particionan estos conceptos, realizando un diccionario visual de los términos importantes del dominio.

2.3.1 Glosario de términos del modelo de dominio

Universidad: Es la institución donde se encuentra la Facultad (UCI).

Facultad: Es específicamente la entidad donde se desarrolla el análisis del currículo de los profesores de la Facultad 1.

Profesor RG: Es el profesor recién egresado de la Universidad, que cuando comienza su trabajo, su currículo se encuentra en cero.

Profesor: Es un profesor que no es graduado de la UCI, y que tiene una hoja de vida desarrollada antes de entrar a la UCI, además, puede ser un profesor que solicite una plaza para trabajar en la UCI.

Currículo: Contiene la información curricular de los profesores.

Departamento: Puede ser un profesor o conjunto de profesores con un cargo dentro de un departamento.

Decano: Es el principal dirigente de la Facultad.

Planificación: Persona o grupo de personas que se encargan de la planificación docente de la facultad.

Postgrado: Es una persona o conjunto de personas que se encargan de organizar los cursos de postgrado que pasan los profesores.

2.3.2 Diagrama del Modelo del Dominio

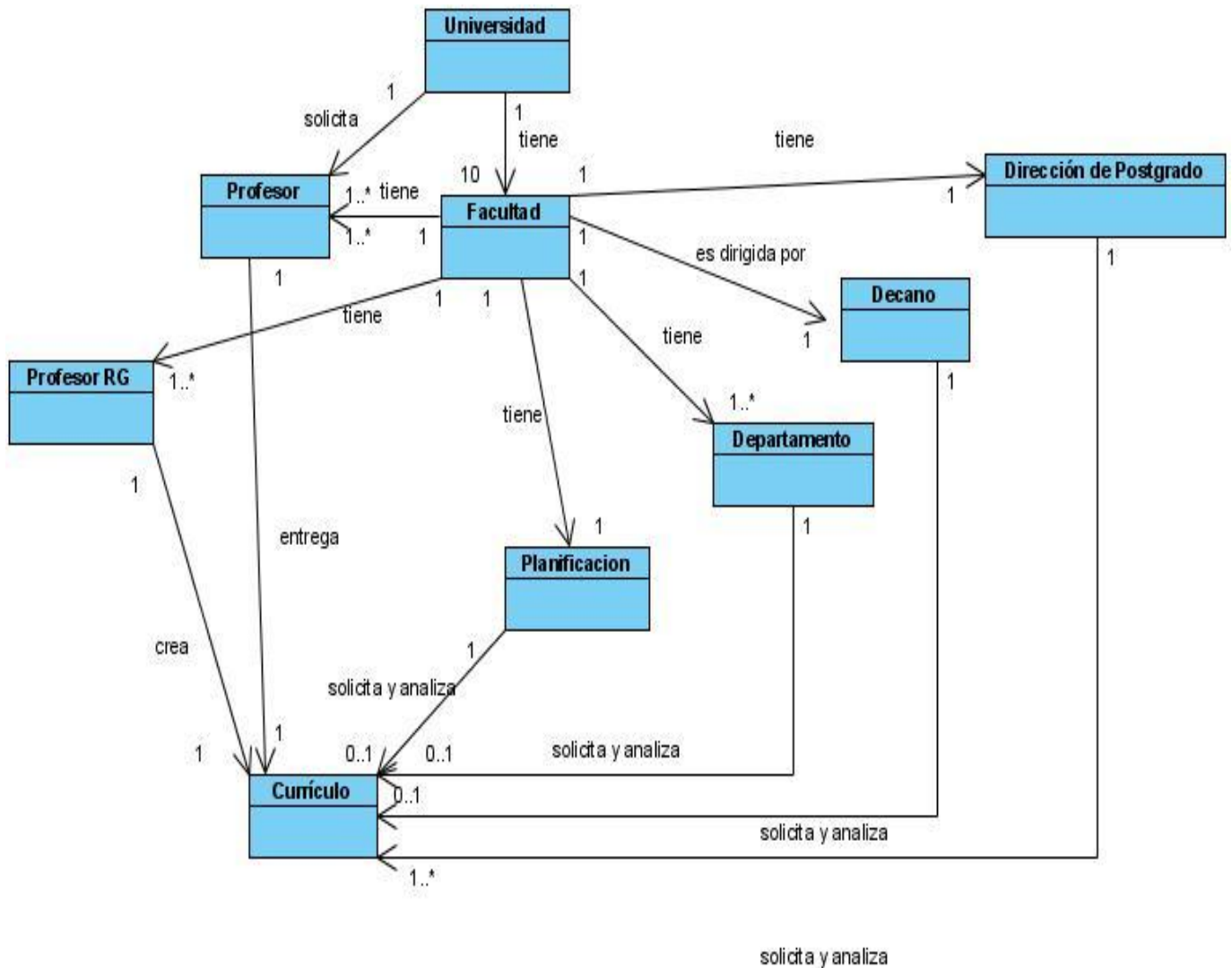


Figura 1 Modelo de dominio o conceptual

2.4 Especificación de Requerimientos de Software

La IEEE⁴¹ Standard Glossary of Software Engineering Terminology define un requerimiento como una condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Un requerimiento no es más que esa condición que debe ser alcanzada o poseída por un sistema o

⁴¹ IEEE: (Institute of Electrical and Electronics Engineers). El Instituto de Ingenieros Eléctricos y Electrónicos es una asociación técnico-profesional mundial dedicada a la estandarización.

componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente. Por lo tanto, los requerimientos de software le brindan al usuario la posibilidad de que el sistema cumpla las condiciones que le interesen, siendo estos una manera de verificar la calidad del producto. Por esto, el levantamiento de requerimientos se hace fundamental a la hora de desarrollar cualquier sistema. Estos requerimientos se dividen en dos grupos, los requerimientos funcionales, que son las capacidades o condiciones que debe cumplir el sistema, y los no funcionales que son cualidades o propiedades que el producto debe tener. (Conferencia 4 Ingeniería de Software I, 2008)

2.4.1 Requerimientos Funcionales (RF)

RF1. Autenticación.

RF1.1. Permitir el acceso a los usuarios autorizados.

RF1.2. Denegar el acceso a los usuarios no autorizados.

RF2. Gestionar Usuarios.

RF2.1. Adicionar un nuevo usuario.

RF2.2. Modificar un usuario existente.

RF2.3. Eliminar un usuario.

RF3. Gestionar datos personales del profesor.

RF3.1. Adicionar datos personales del profesor.

RF3.2. Modificar datos personales del profesor.

RF3.3. Eliminar datos personales del profesor.

RF3.4. Insertar una imagen personal del profesor.

RF4. Gestionar currículum del profesor.

RF4.1. Gestionar categoría docente.

RF4.1.1. Adicionar datos de la categoría docente.

RF4.1.2. Modificar datos de la categoría docente.

RF4.1.3. Eliminar datos de la categoría docente.

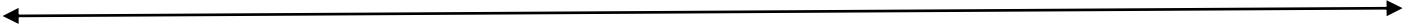
RF4.2. Gestionar categoría científica.

RF4.2.1. Adicionar datos de la categoría científica.

RF4.2.2. Modificar datos de la categoría científica.

- RF4.2.3. Eliminar datos de la categoría científica.
- RF4.3. Gestionar misiones cumplidas.
 - RF4.3.1. Adicionar datos de las misiones cumplidas.
 - RF4.3.2. Modificar datos de las misiones cumplidas.
 - RF4.3.3. Eliminar datos de las misiones cumplidas.
- RF4.4. Gestionar eventos.
 - RF4.4.1. Adicionar datos de los eventos.
 - RF4.4.2. Modificar datos de los eventos.
 - RF4.4.3. Eliminar datos de los eventos.
- RF4.5. Gestionar tesis tutoradas.
 - RF4.5.1. Adicionar datos de las tesis tutoradas.
 - RF4.5.2. Modificar datos de las tesis tutoradas.
 - RF4.5.3. Eliminar datos de las tesis tutoradas.
- RF4.6. Gestionar participación en proyectos.
 - RF4.6.1. Adicionar datos del proyecto.
 - RF4.6.2. Modificar datos de un proyecto.
 - RF4.6.3. Eliminar datos del proyecto.
- RF4.7. Gestionar investigaciones realizadas.
 - RF4.7.1. Adicionar datos de las investigaciones realizadas.
 - RF4.7.2. Modificar datos de las investigaciones realizadas.
 - RF4.7.3. Eliminar datos de las investigaciones realizadas.
- RF4.8. Gestionar cursos de postgrado.
 - RF4.8.1. Gestionar cursos de postgrado impartidos.
 - RF4.8.1.1. Adicionar datos del curso de postgrado impartido.
 - RF4.8.1.2. Modificar datos del curso de postgrado impartido.
 - RF4.8.1.3. Eliminar datos del curso de postgrado impartido.
 - RF4.8.2. Gestionar cursos de postgrado recibidos.
 - RF4.8.2.1. Adicionar datos del curso de postgrado recibido.
 - RF4.8.2.2. Modificar datos del curso de postgrado recibido.
 - RF4.8.2.3. Eliminar datos del curso de postgrado recibido.

-
- RF4.9. Gestionar dominio de idiomas.
 - RF4.9.1. Adicionar datos del idioma.
 - RF4.9.2. Modificar datos del idioma.
 - RF4.9.3. Eliminar datos del idioma.
 - RF4.10. Gestionar publicaciones realizadas.
 - RF4.10.1. Adicionar datos de las publicaciones.
 - RF4.10.2. Modificar datos de las publicaciones.
 - RF4.10.3. Eliminar datos de las publicaciones.
 - RF4.11. Gestionar productos patentados.
 - RF4.11.1. Adicionar datos de los productos patentados.
 - RF4.11.2. Modificar datos de los productos patentados.
 - RF4.11.3. Eliminar datos de los productos patentados.
 - RF4.12. Gestionar responsabilidades ocupadas.
 - RF4.12.1. Adicionar datos de las responsabilidades ocupadas.
 - RF4.12.2. Modificar datos de las responsabilidades ocupadas.
 - RF4.12.3. Eliminar datos de las responsabilidades ocupadas.
 - RF4.13. Gestionar otra información.
 - RF4.13.1. Adicionar otra información.
 - RF4.13.2. Modificar otra información.
 - RF4.13.3. Eliminar otra información.
 - RF5. Cambiar contraseña anterior.
 - RF6. Realizar búsqueda avanzada.
 - RF6.1. Buscar por nombre.
 - RF6.2. Buscar por cualquiera de los dos apellidos.
 - RF6.3. Buscar por categoría científica.
 - RF6.4. Buscar por categoría docente.
 - RF7. Obtener el currículum de un profesor.
 - RF8. Obtener los datos personales de un profesor.
 - RF9. Gestionar nomencladores.
 - RF9.1. Gestionar nomenclador idioma.



RF9.1.1. Adicionar nuevo idioma.

RF9.1.2. Modificar idioma existente.

RF9.1.3. Eliminar idioma.

RF9.2. Gestionar nomenclador categoría científica.

RF9.2.1. Adicionar nueva categoría científica.

RF9.2.2. Modificar categoría científica existente.

RF9.2.3. Eliminar categoría científica.

RF9.3. Gestionar nomenclador categoría docente.

RF9.3.1. Adicionar categoría docente.

RF9.3.2. Modificar categoría docente existente.

RF9.3.3. Eliminar categoría docente.

RF9.4. Gestionar nomenclador tipo de evento.

RF9.4.1. Adicionar tipo de evento.

RF9.4.2. Modificar tipo de evento.

RF9.4.3. Eliminar tipo de evento.

RF10. Generar reportes.

RF10.1. Reporte de profesores que han impartido o recibido un curso de postgrado dado.

RF10.2. Reporte de profesores graduados en una determinada especialidad.

RF10.3. Reporte de profesores con una determinada categoría docente.

RF10.4. Reporte de profesores con una determinada categoría científica.

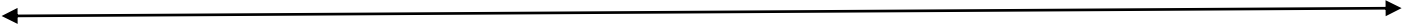
RF10.5. Reporte de profesores que han impartido una asignatura determinada.

RF11. Generar documento Reporte en formato pdf.

2.4.1 Requerimientos No Funcionales

Apariencia o Interfaz:

- El sistema debe contar con un entorno amigable que esté identificado con la facultad.
- Debe presentar una interfaz sencilla y fácil de usar aún por usuarios que posean poco conocimiento informático.
- Las imágenes y colores deben identificar el negocio donde se implantará el sistema.



Usabilidad:

- El sistema debe ser fácil de aprender.
- Debe presentar una ayuda para su mejor uso.

Seguridad:

- Garantizar que todos sus servicios sean usados por el personal autorizado.
- Garantizar que la información sea modificada y vista solamente por el personal con permiso para ello.

Disponibilidad:

- Permitir a los usuarios conectarse desde cualquier computadora dentro de la Universidad.
- Disponible 24 horas.

Portabilidad:

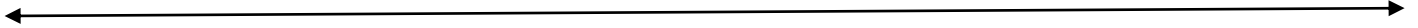
- Múltiple plataforma.
- El sistema podrá ser visualizado en diferentes sistemas operativos como Windows 98(o superiores) y Linux.

Rendimiento:

- El sistema debe brindar una rápida respuesta a las solicitudes que hagan los usuarios.
- El sistema debe presentar pocas imágenes en las páginas para garantizar una respuesta más rápida.

Hardware:

- Se requiere para el servidor una computadora PC P4, con un procesador de velocidad 1.60Hz o más, con una memoria RAM de 256 MB o mayor, y 5 GB o más de espacio en el disco duro.
- Al ser una aplicación Web, se necesita un navegador, como Internet Explorer, Mozilla Firefox, Opera, etc.
- Para acceder al sistema se requiere solamente de una computadora con un navegador instalado.



Restricciones de Diseño e Implementación:

- Para implementar el sistema se usará el lenguaje PHP.
- El sistema gestor de base de datos será postgres.

Software:

- El navegador debe tener el intérprete de JavaScript habilitado.

2.5 Descripción del los Actores del Sistema

Los actores del sistema son aquellas personas que interactuarán directamente con el mismo. Para el desarrollo de la aplicación se definen cuatro tipos de usuarios que son los que tendrán acceso al sistema, los cuales se describen a continuación.

Actores	Justificación
Usuario	Es el usuario que tiene permiso para realizar la búsqueda de profesores, ver su información personal y su currículum, así como para generar el documento con esta información. Además, puede cambiar su propia contraseña. Pueden ser estudiantes, trabajadores, profesores, etc.
Profesor	Es el cliente principal del sistema. Tiene los permisos de los usuarios, y además gestionar su propia información personal y su currículum, así como generar el documento de su propio currículum. Pueden ser cualquier profesor de la facultad.
Directivo	Es un profesor con permisos adicionales. Es la persona que tiene acceso a los reportes de la aplicación. Un directivo puede ser el Jefe de Departamento, el Jefe de Postgrado, el Jefe del Departamento Central, Director de Formación, el

	Jefe del Colectivo, el Decano.
Administrador	Cuenta con los permisos de los usuarios, y además, gestiona los nomencladores y los usuarios.

Tabla 1 Descripción de los usuarios del sistema

2.6 Descripción Breve de los Casos de Uso

Con el objetivo de crear un modelo de las funcionalidades deseadas para el sistema y su entorno, y debido a que se utilizan como entrada esencial para las actividades de análisis, diseño y prueba, se definen una serie de casos de uso del sistema, realizándose el diagrama de casos de uso del sistema y una descripción detallada de cada uno de ellos. La descripción ampliada de los casos de uso del sistema muestra una descripción detallada de los mismos. Debido a que es mucha y muy variada la información del currículo de un profesor en la UCI, los casos de uso Gestionar Nomencladores y Gestionar Currículo, serán descritos de forma genérica. (Ver Anexo1. Descripción ampliada de los casos de uso del sistema.)

CU-1	Autenticar.
Actor	Usuario, Profesor, Directivo, Administrador.
Descripción	Permite o limita el acceso a los usuarios y se inicia cuando un usuario quiere acceder al sistema.
Referencia	RF 1, RF 1.1, RF 1.2

Tabla 2 Breve descripción del caso de uso "Autenticar".

CU-2	Gestionar Usuario.
Actor	Administrador.
Descripción	Permite al administrador crear, modificar y eliminar usuarios que podrán acceder al sistema.

Referencia	RF 2, RF 2.1, RF 2.2, RF 2.3
-------------------	------------------------------

Tabla 3 Breve Descripción del caso de uso "Gestionar Usuario"

CU-3	Gestionar Datos Personales.
Actor	Profesor, Directivo.
Descripción	Permite a los profesores o directivos crear una ficha con su información personal, brindándole la posibilidad de adicionar, modificar o eliminar estos datos.
Referencia	RF 3, RF 3.1, RF 3.2, RF 3.3, RF 3.4

Tabla 4 Breve Descripción del caso de uso "Gestionar Datos Personales".

CU- 4	Gestionar Currículo
Actor	Profesor, Directivo
Descripción	Permite al profesor crear, modificar o eliminar información de su currículo.
Referencia	RF 4, RF 4.1, RF 4.1.1, RF 4.1.2, RF 4.1.3, RF 4.2, RF 4.3, RF 4.4, RF 4.5, RF 4.6, RF 4.7, RF 4.8, RF 4.9, RF 4.10, RF 4.11, RF 4.12, RF 4.13

Tabla 5 Breve Descripción del caso de uso "Gestionar Currículo".

CU-5	Generar Reportes.
-------------	-------------------

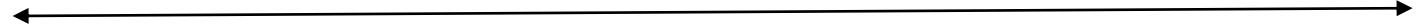
Actor	Directivo.
Descripción	Permite a los directivos generar reportes concernientes a la información de los datos personales introducidos por los profesores, así como de los datos del currículo.
Referencia	RF 11, RF 11.1, RF 11.2, RF 11.3, RF 11.4, RF 11.5

Tabla 6 Breve Descripción del caso de uso "Generar Reportes".

CU-6	Cambiar Contraseña
Actor	Usuario, Administrador, Profesor, Directivo.
Descripción	Permite a todos los usuarios cambiar su contraseña una vez que accedan al sistema y sus cuentas hayan sido creadas por el administrador.
Referencia	RF 5

Tabla 7 Breve Descripción del caso de uso "Cambiar Contraseña".

CU-7	Gestionar Búsqueda.
Actor	Usuario, Administrador, Profesor, Directivo.
Descripción	Permite a los usuarios realizar una búsqueda de profesores por diferentes criterios de búsqueda, y una vez mostrador los resultados, brinda la opción de ver el currículo y la ficha con los datos



	personales del profesor, así como imprimir el currículum del profesor que seleccione.
Referencia	RF 6, RF 6.1, RF 6.2, RF 6.3, RF 6.4, RF 7, RF 8

Tabla 8 Breve Descripción del caso de uso "Gestionar Búsqueda".

CU-8	Gestionar Nomencladores
Actor	Administrador
Descripción	Permite al administrador adicionar, modificar o eliminar nomencladores como categoría científica, docente, tipos de eventos e idiomas.
Referencia	RF 9, RF 9.1, RF 9.1.1, RF 9.1.2, RF 9.1.3, RF 9.2, RF 9.3, RF 9.4, RF 10

Tabla 9 Breve Descripción del caso de uso "Gestionar Nomencladores".

2.7 Diagrama de Casos de Uso del Sistema

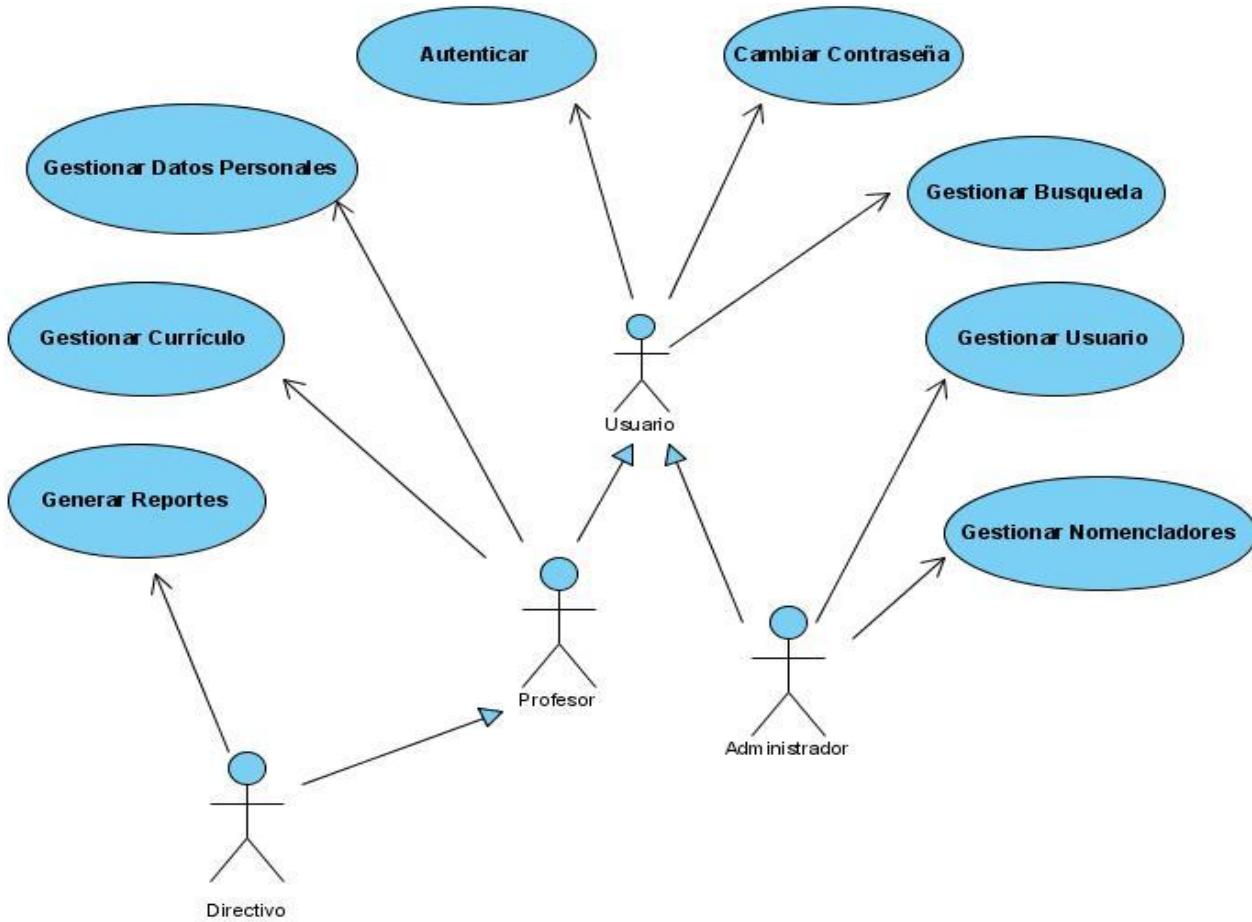


Figura 2 Diagrama de casos de uso del sistema.

Conclusiones

En este capítulo se identificaron los principales conceptos que se usarán para desarrollar el sistema, así como los tipos más importantes de objetos que existen y los eventos que suceden en el entorno en el que estará el sistema. Se obtuvieron los requerimientos funcionales y los no funcionales de la aplicación. Partiendo de los requisitos del sistema, posteriormente fueron presentados los casos de uso del sistema y sus relaciones con los actores del sistema. Realizando una detallada descripción de los casos de uso. Los artefactos generados en este capítulo son la base para comenzar el análisis y el diseño.

Capítulo 3: Análisis y Diseño del Sistema

Introducción

En este capítulo se abordan los temas referentes al análisis y el diseño del sistema. A partir del resultado obtenido en el capítulo anterior se realiza el análisis, donde se consigue una comprensión más precisa de los requisitos, de manera que puedan ser refinados y estructurados, proporcionando una visión general del sistema, describiéndose en el diagrama de clases del análisis. Además, se obtiene el modelo de diseño, el cual incluye el diseño de la base de datos, el diagrama de clases del diseño, con sus respectivos diagramas de interacción. Se describen detalladamente las clases del diseño y las tablas de la base de datos. Además, se explican los patrones de diseño que se utilizan, y cómo se lleva a cabo el tratamiento de errores, la seguridad, el diseño de interfaz y cómo estará concebida la ayuda. El resultado de este capítulo es muy importante porque sirve de entrada para la implementación.

3.1 Diagrama de Clases del Análisis

Con el objetivo de refinar y estructurar los requisitos obtenidos con anterioridad, profundizándose en el dominio de la aplicación, y para obtener una mayor comprensión del problema para modelar la solución, se realiza el análisis, cuyo principal resultado es el diagrama de clases del análisis (DCA). Para desarrollar el sistema, fue necesario hacer este diagrama, el cual representa los conceptos en el dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas. Los diagramas de clases del análisis resultantes se muestran a continuación:

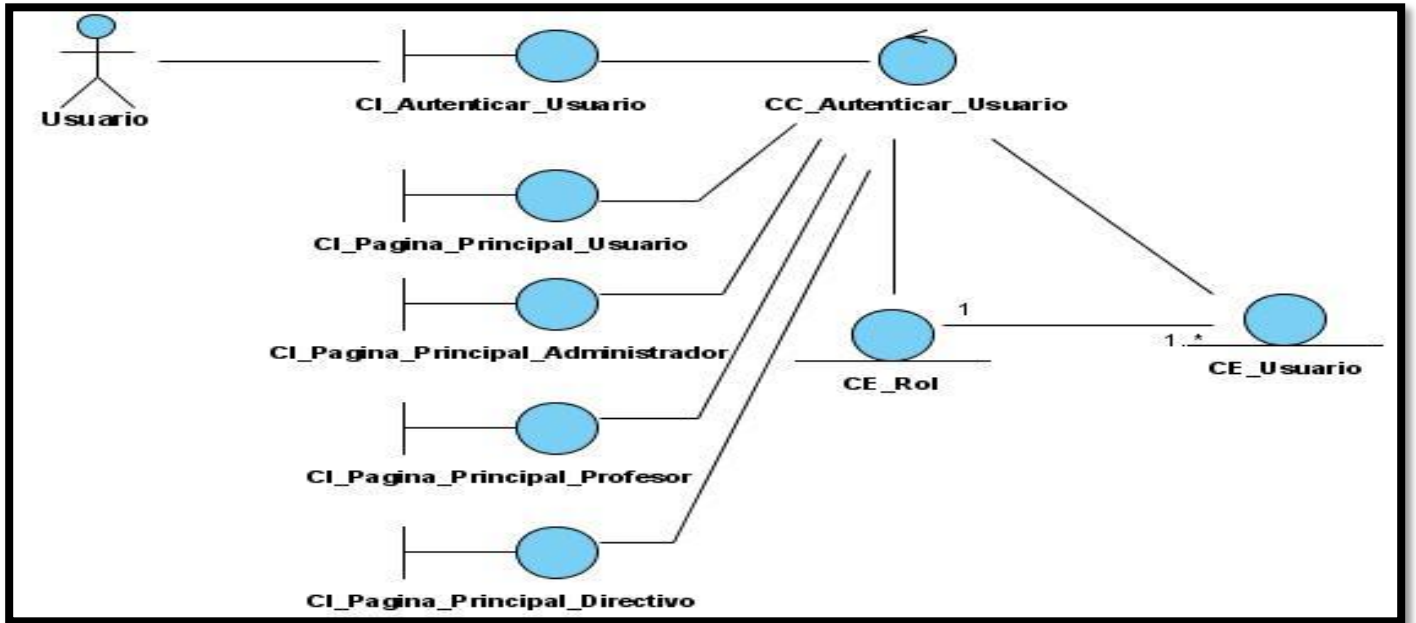


Figura 3 DCA "Autenticar usuario".

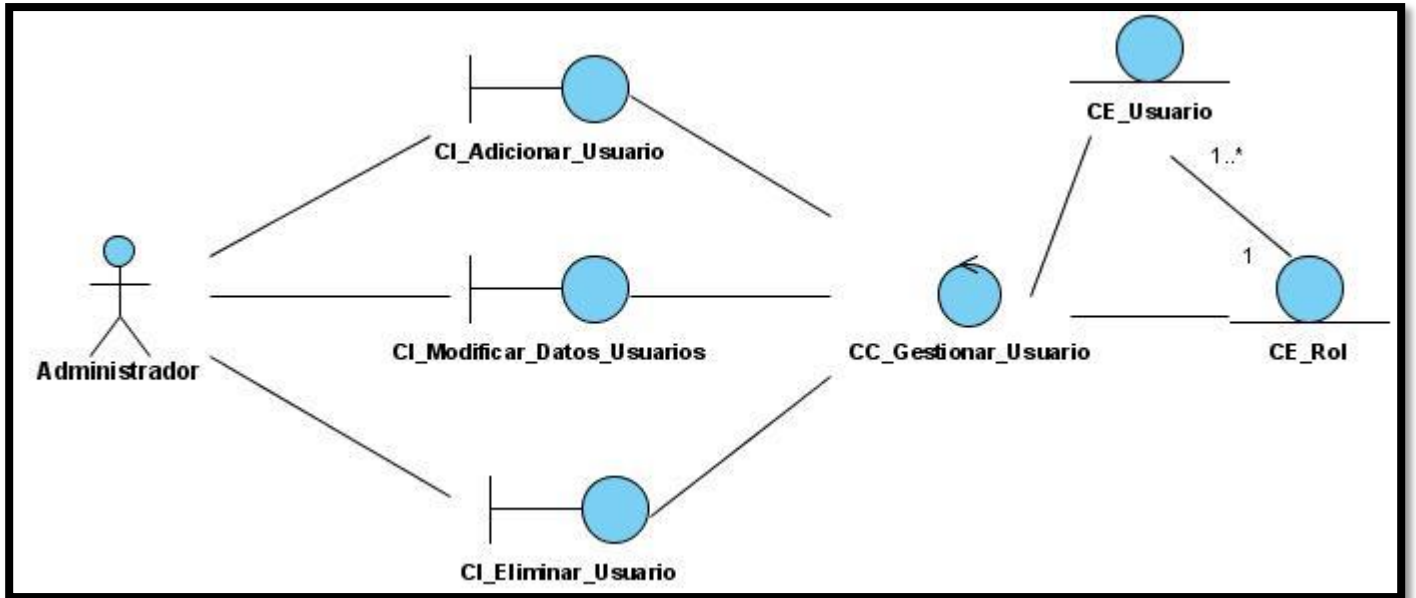


Figura 4 DCA "Gestionar usuarios".

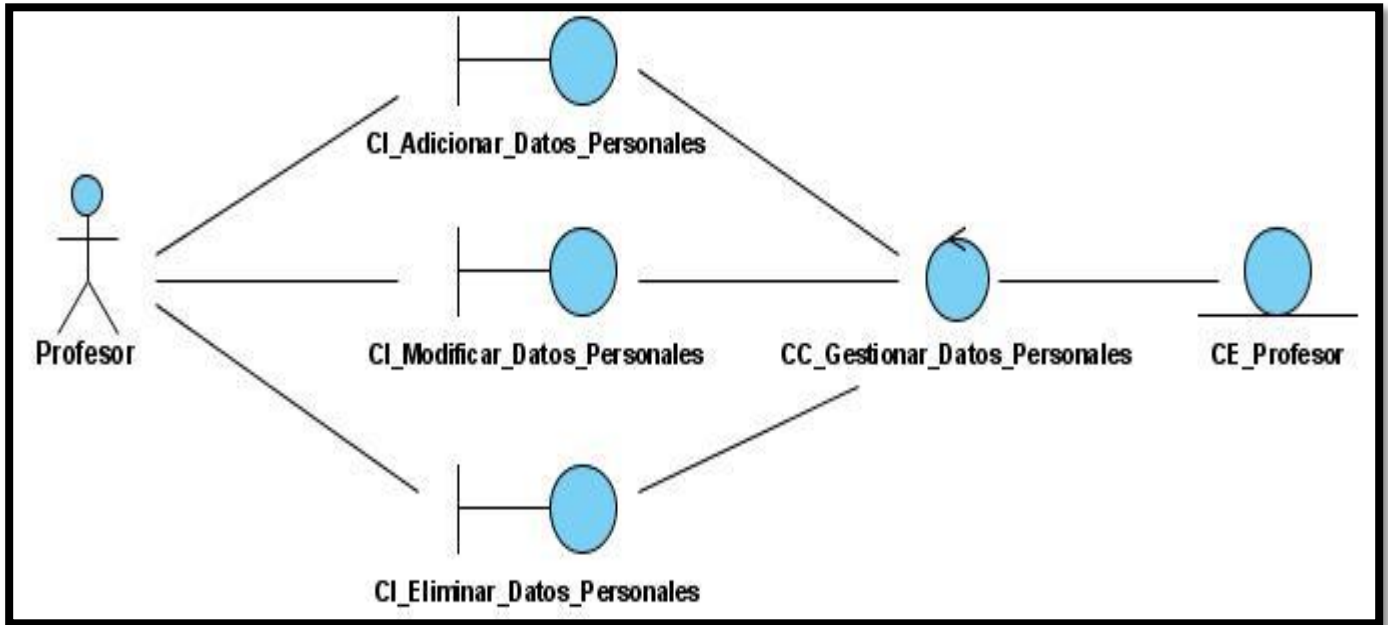


Figura 5 DCA "Gestionar datos personales".

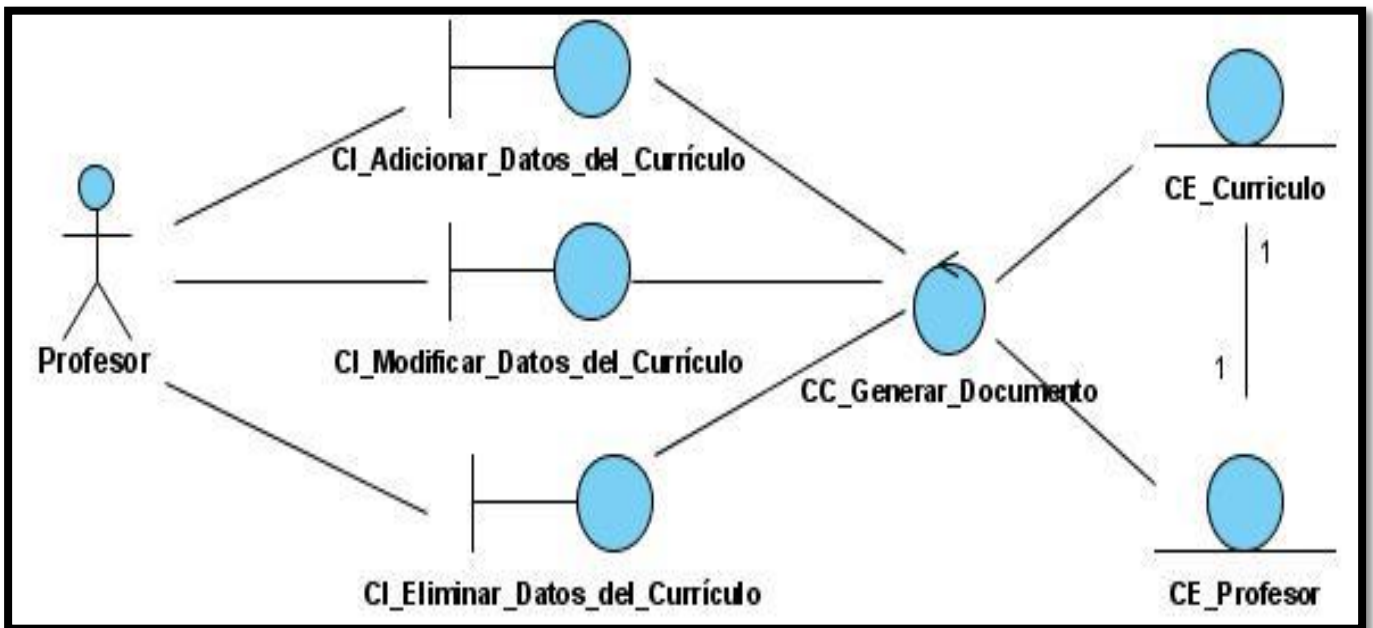


Figura 6 DCA "Gestionar currículum".

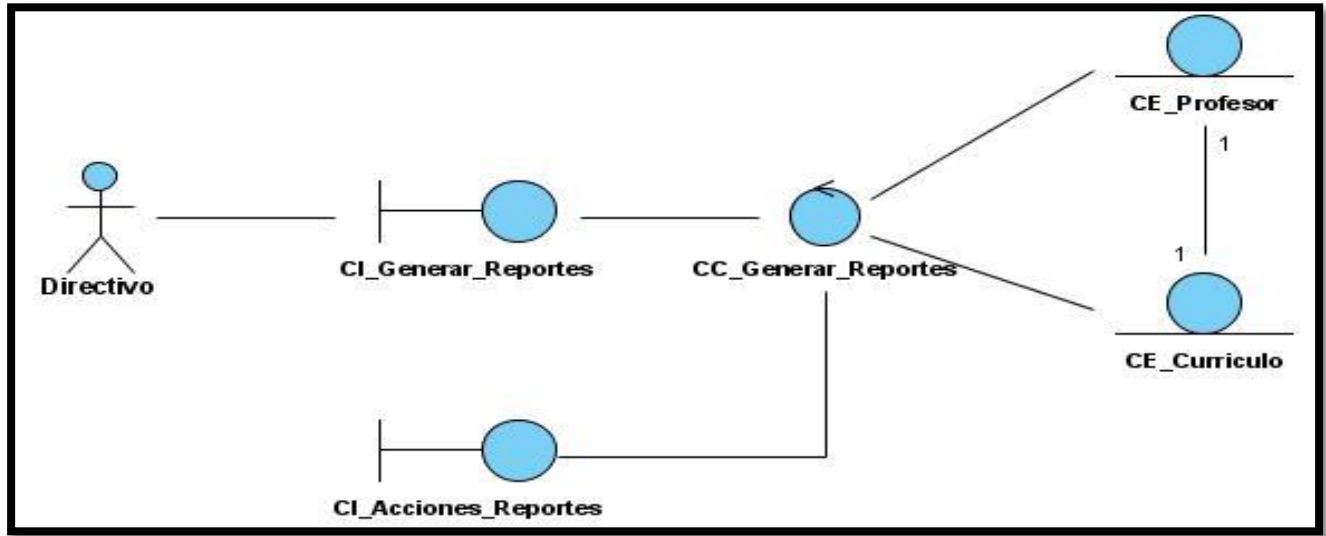
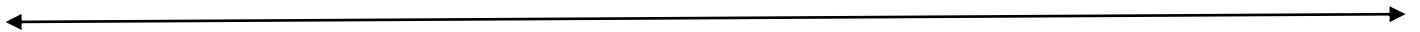


Figura 7 DCA "Generar reportes".

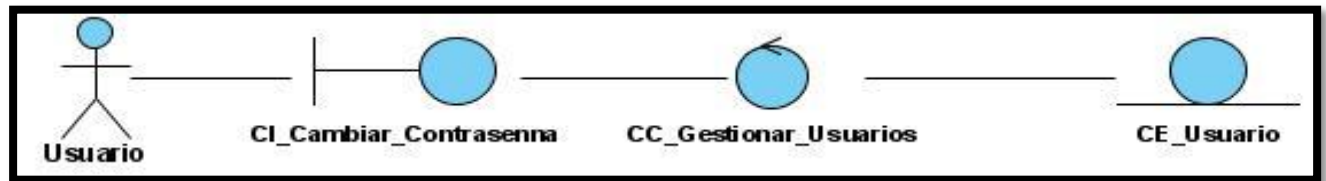


Figura 8 DCA "Cambiar contraseña".

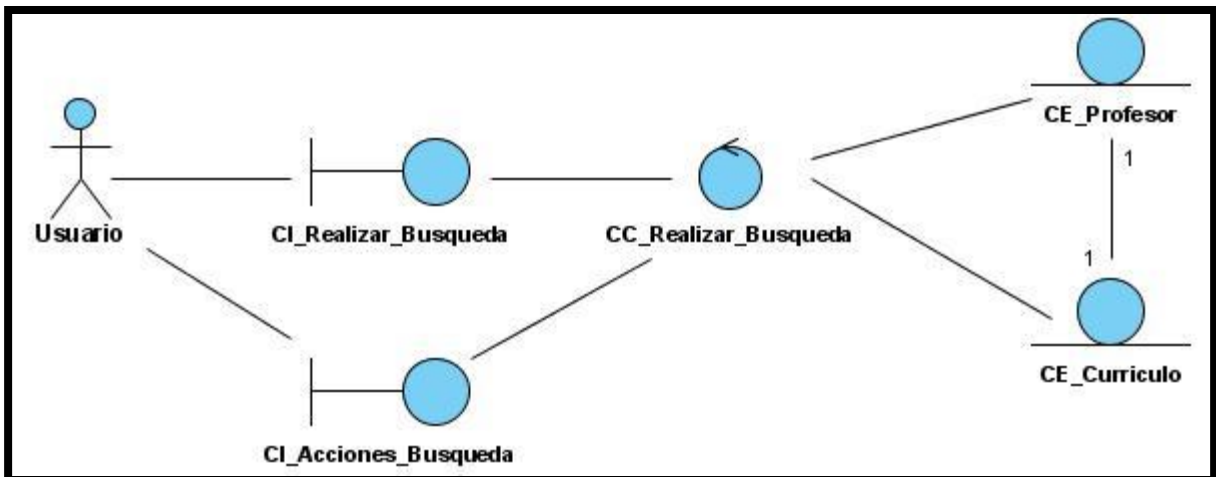


Figura 9 DCA "Realizar búsqueda".

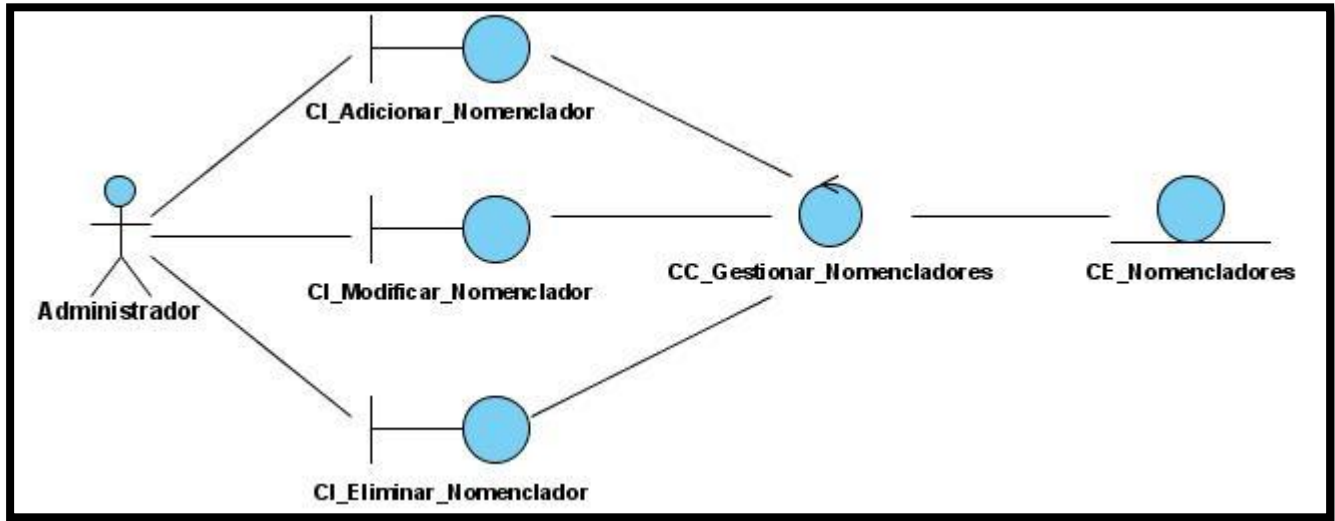


Figura 10 DCA "Gestionar nomencladores".

3.2 Modelo de diseño

En el diseño se modela el sistema y se encuentra su forma (incluida la arquitectura) para que soporte todos los requerimientos, incluyendo los no funcionales y sus restricciones. Una entrada esencial en el diseño es el resultado del análisis, que proporciona una comprensión detallada de los requisitos.

3.2.1 Patrones de diseño

Los patrones de diseño son soluciones a problemas ya conocidos que ayudan a un mejor rendimiento, desarrollo y mantenimiento del software donde se apliquen, debido a esto se estudió la posibilidad de aplicarlos en el desarrollo del sistema.

3.2.1.1 Patrón Modelo - Vista - Controlador

Este patrón separa el modelo, los datos, la vista, la interfaz y el controlador que se encarga de manejar y adaptar los datos para la interfaz. Este patrón facilita el trabajo, logrando un determinado avance en el desarrollo del sistema, ya que se pueden implementar cada una de estas partes por separado, lo que agiliza la elaboración del mismo, haciéndose el código más viable, entendible y reutilizable. Se propone utilizar el patrón MVC además, porque CodeIgniter lo implementa por su definición, aprovechándolo al máximo.

3.2.1.2. Patrón Alta cohesión

La cohesión no es más que una medida de la fuerza que une a las responsabilidades de una clase, siendo el patrón alta cohesión muy importante a la hora de desarrollar el sistema como patrón de diseño a seguir. Las clases con alta cohesión mejoran la claridad y la facilidad de uso, mediante su uso se simplifica el mantenimiento, se genera un bajo acoplamiento entre clases, y son mucho más fáciles de reutilizar. De esta manera, mediante el uso de este patrón, cada clase tendrá asociada la responsabilidad que le corresponde de acuerdo a su comportamiento.

3.2.2. Diagrama de secuencia.

Los diagramas de secuencia son importantes en el diseño de cualquier sistema, porque muestran gráficamente las interacciones del actor y de las operaciones a que dan origen. Siguiendo esta idea, y con ese objetivo, se realizaron los diagramas de secuencia del sistema.

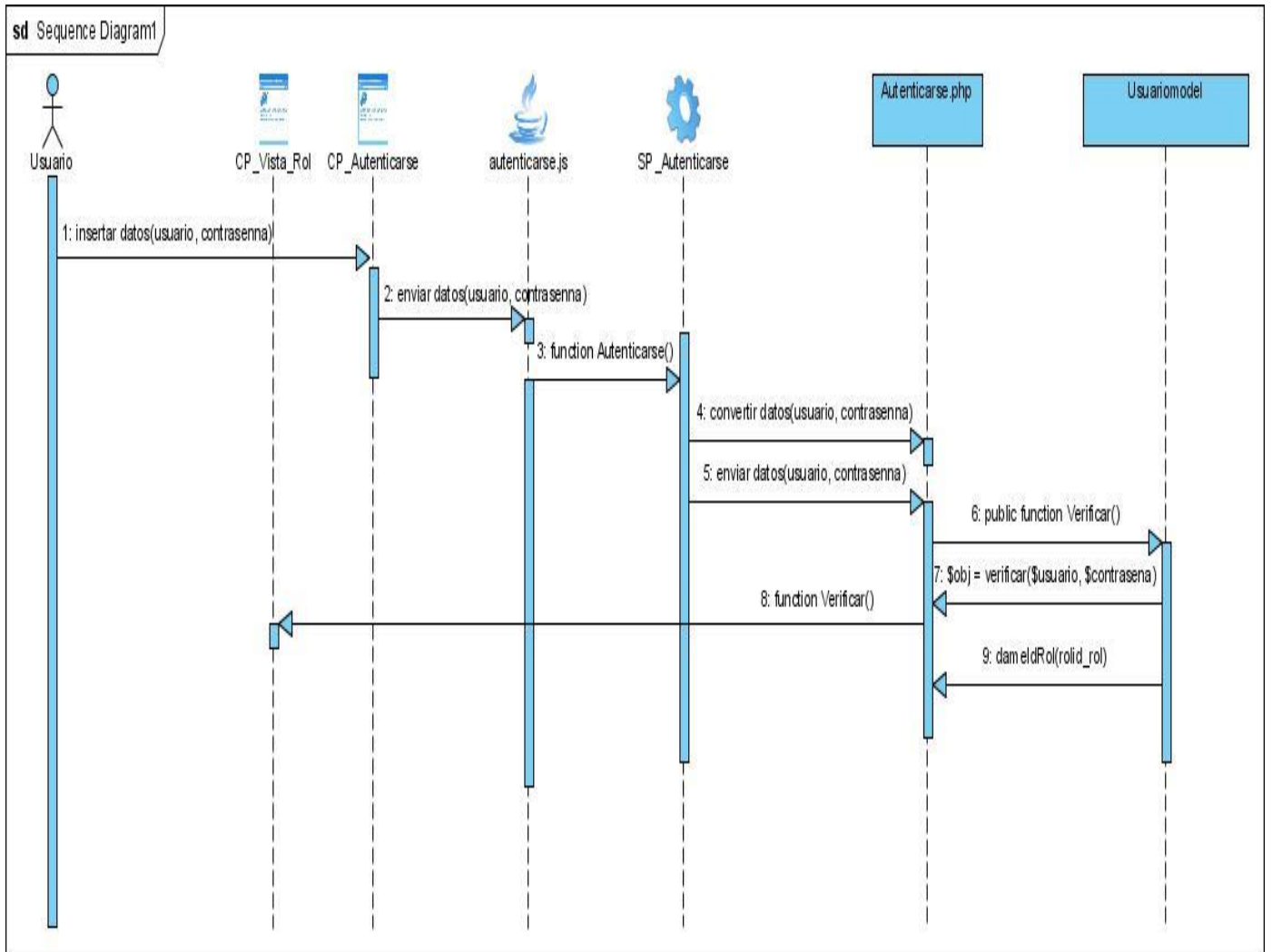


Figura 11 Diagrama de secuencia "Autenticar".

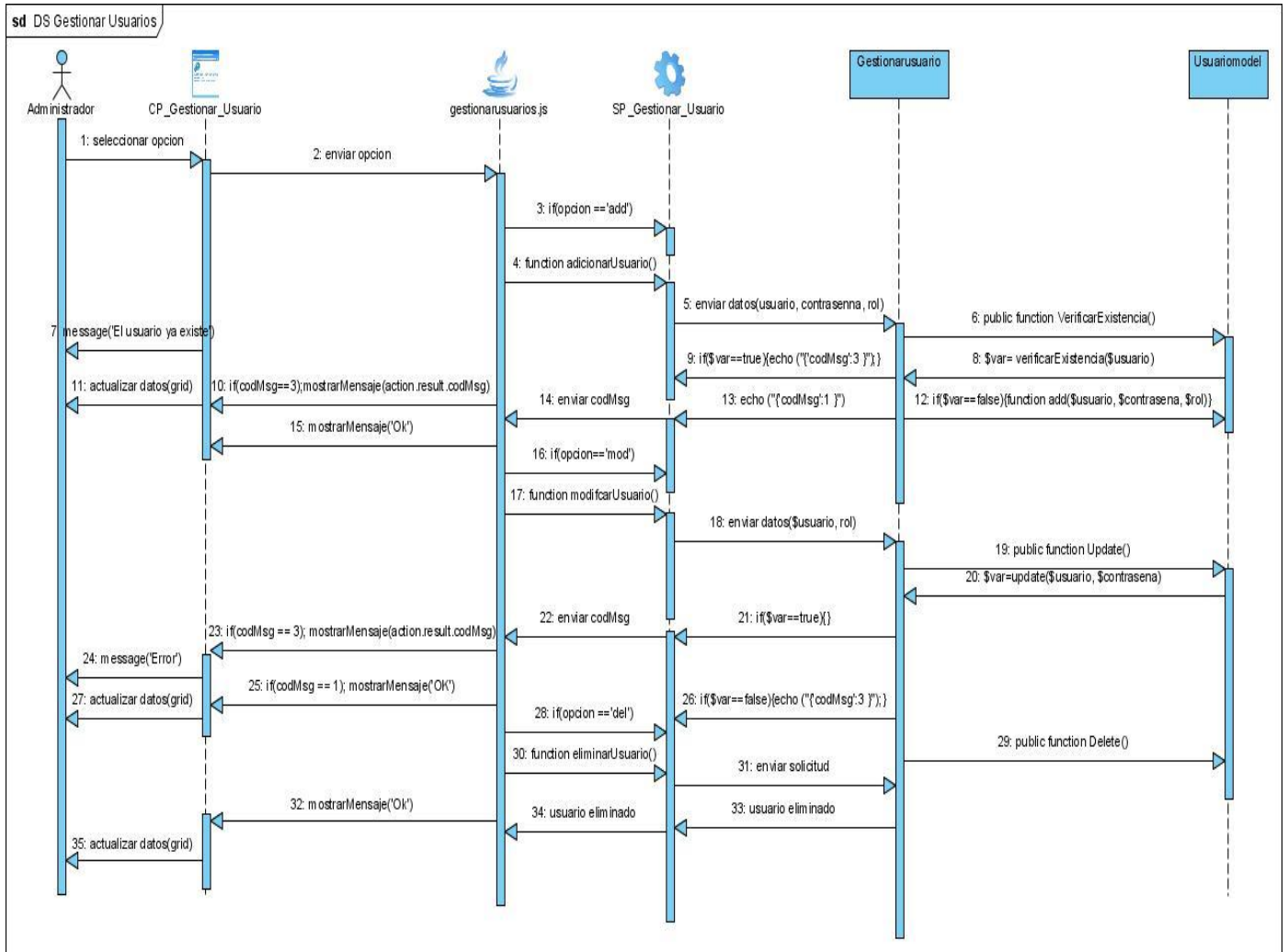


Figura 12 Diagrama de secuencia "Gestionar usuario".

Para el desarrollo eficiente del sistema, y específicamente del capítulo 4, en el flujo de trabajo de implementación, se realizaron diagramas de secuencia por cada clase obtenida del análisis. Las mismas se encuentran en el Anexo 5. Diagramas de secuencia.

3.2.3 Diagrama de Clases Web

El Diagrama de Clases Web (DCW) es donde se muestran las relaciones entre las clases del modelo, las controladoras y las vistas, así como sus atributos y operaciones y las relaciones entre las mismas.

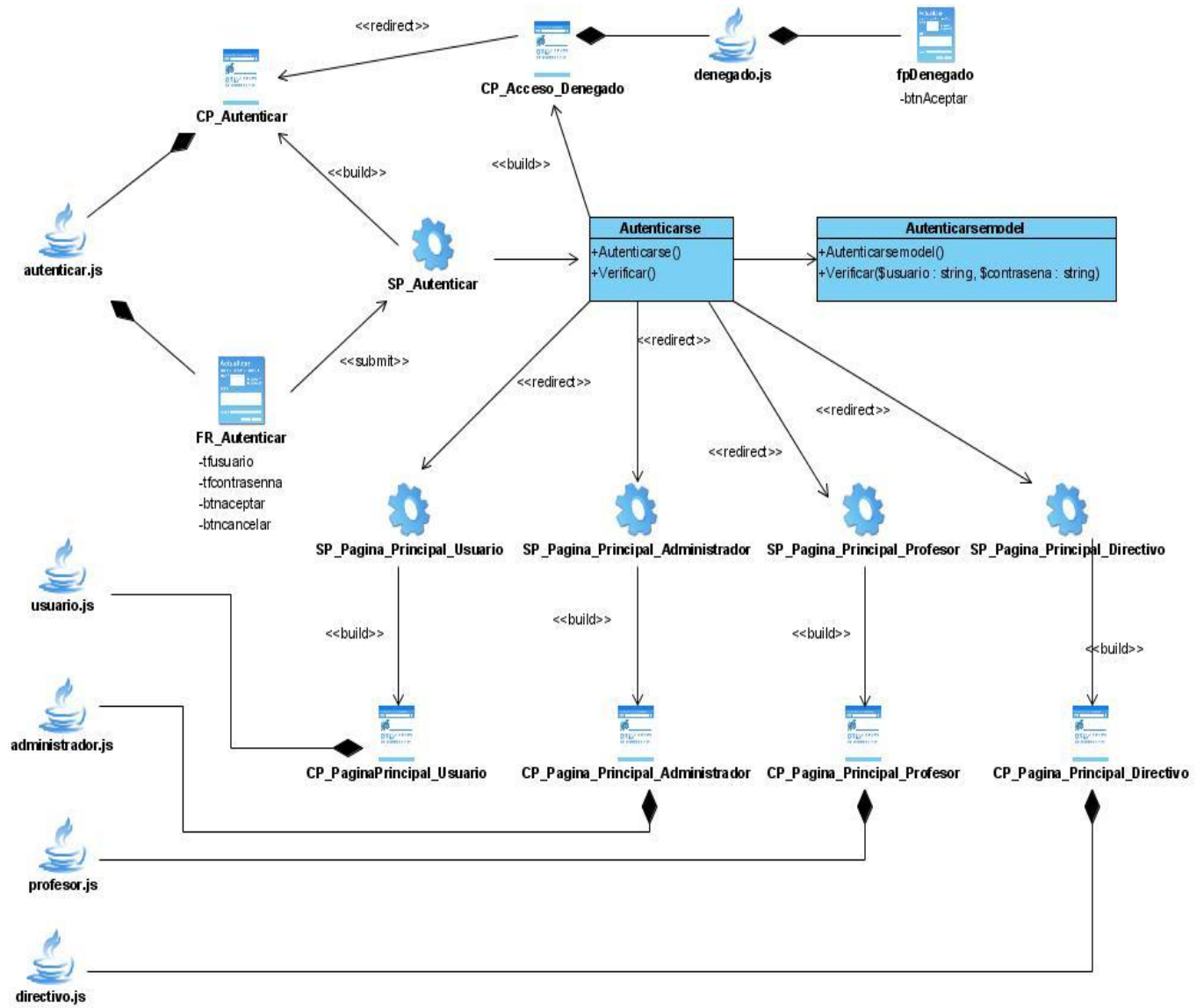


Figura 13 DCW "Autenticar usuario".

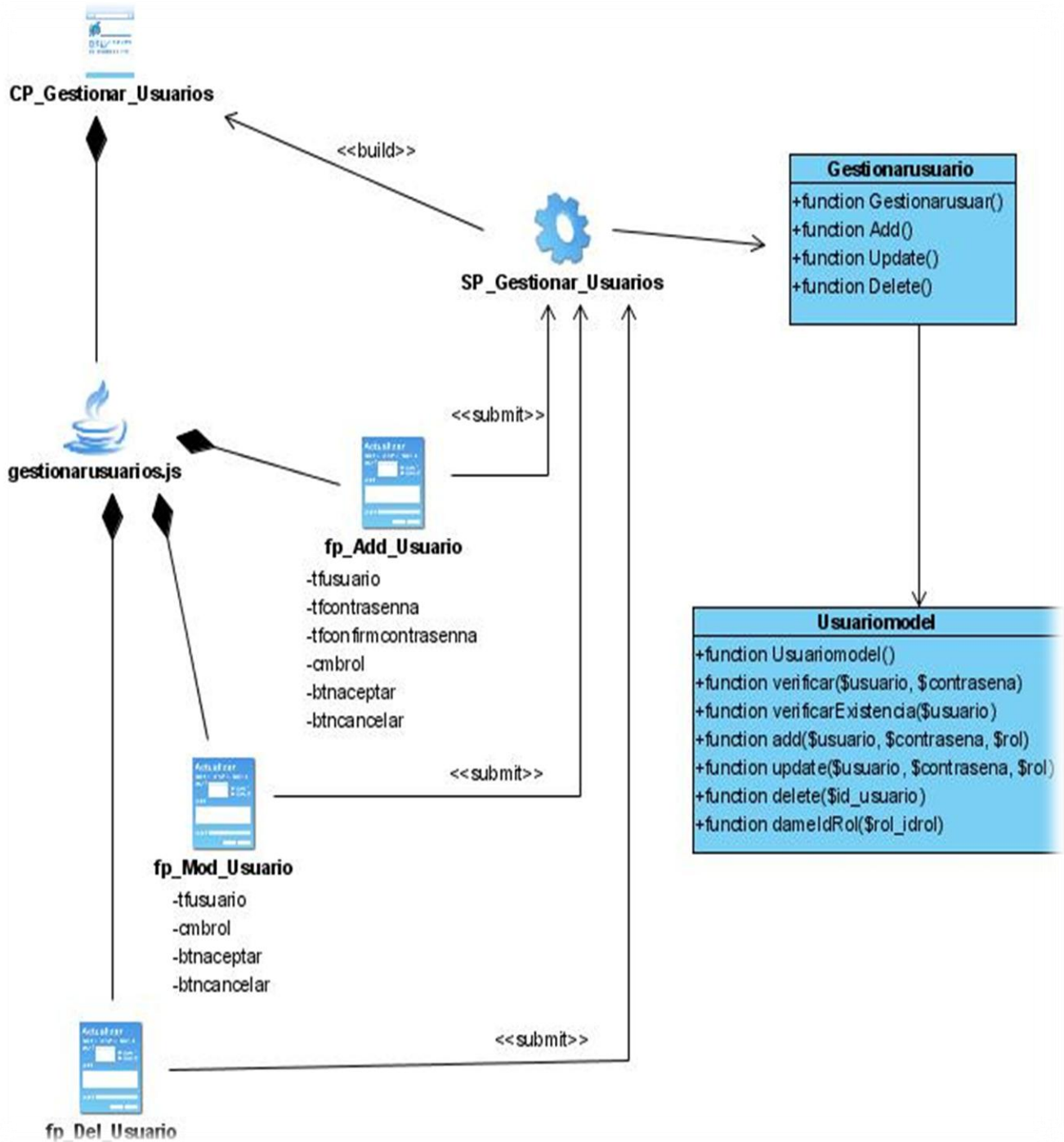


Figura 14 DCW "Gestionar usuarios"

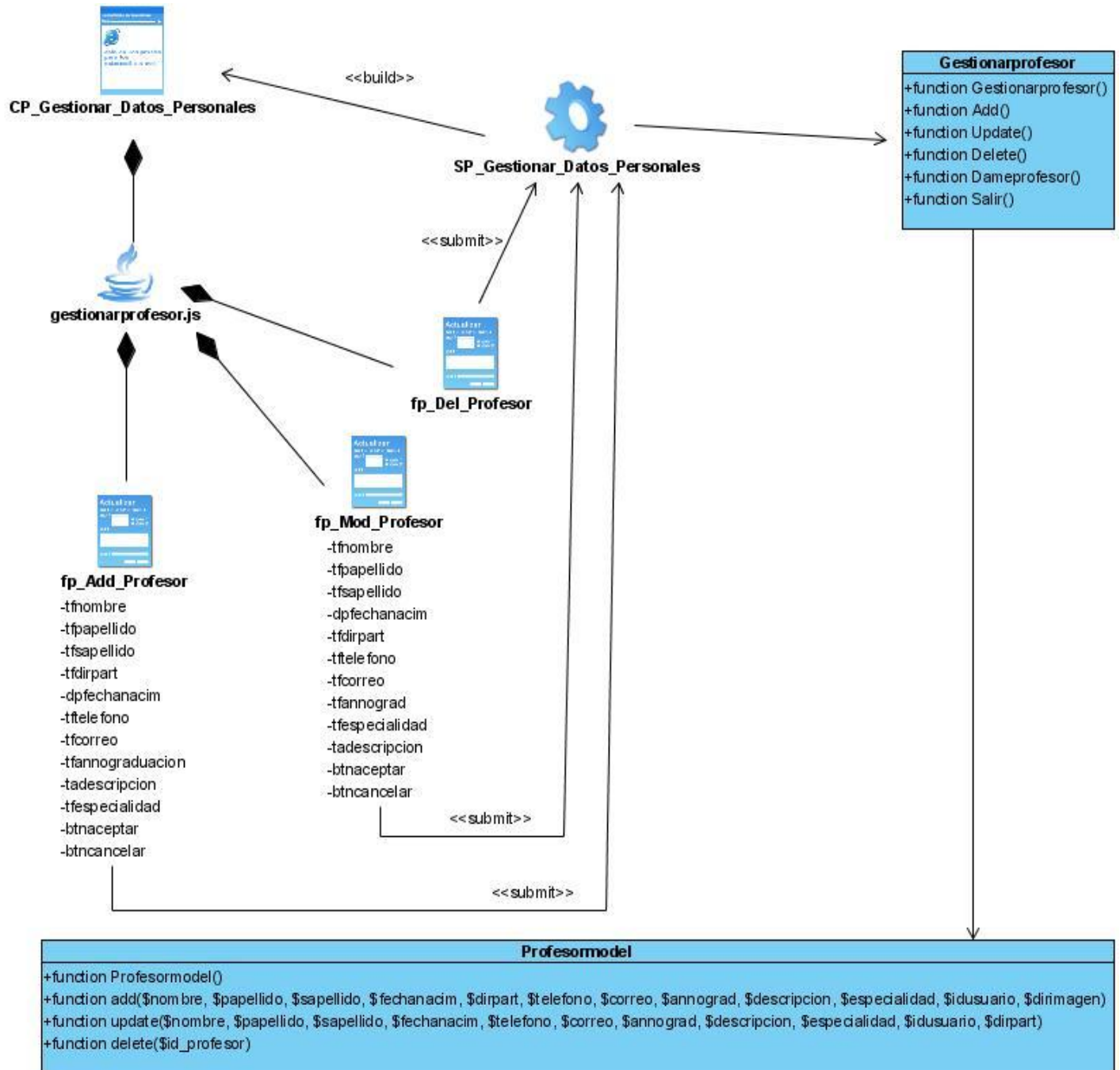


Figura 15 DCW "Gestionar datos personales".

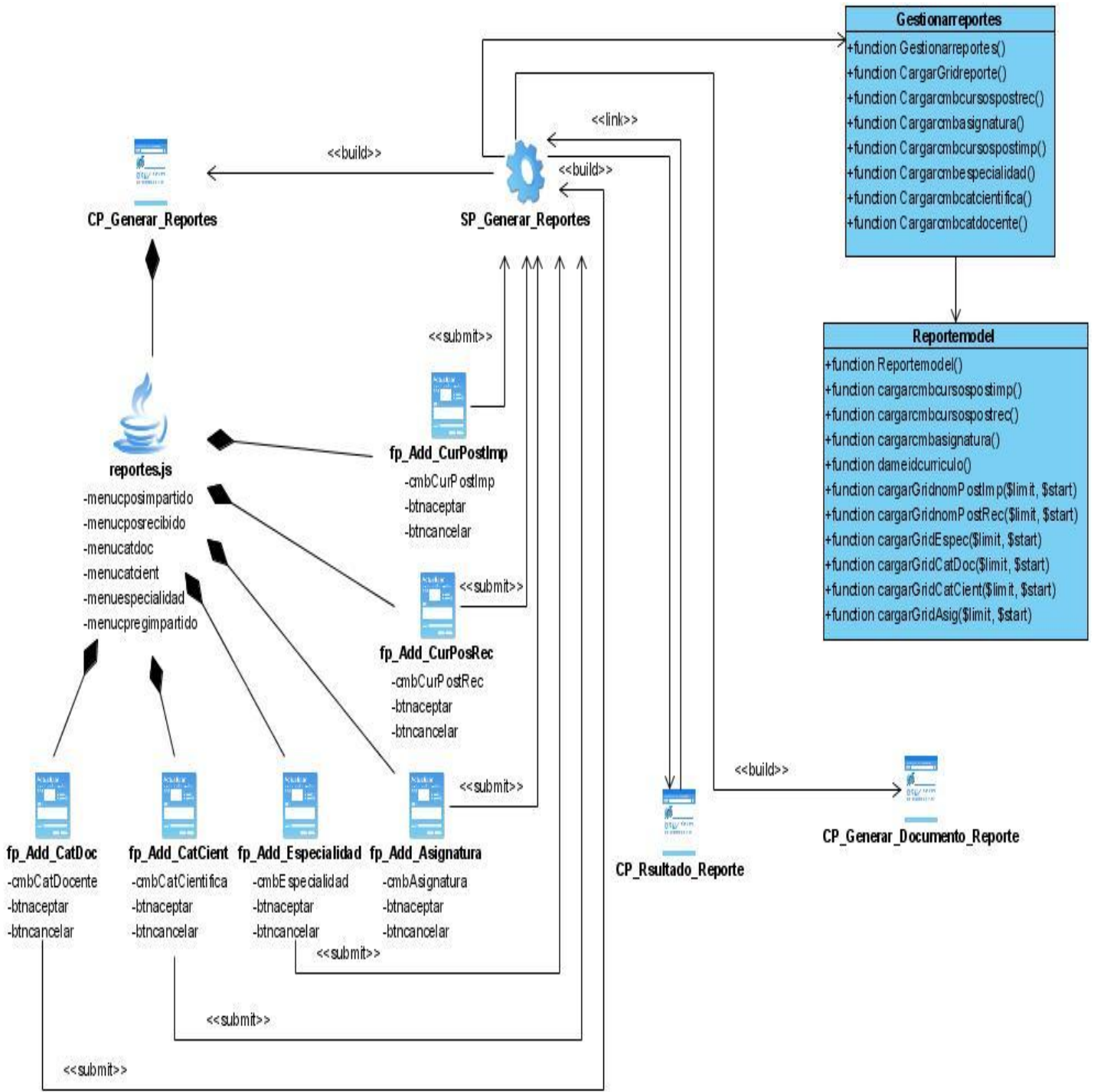
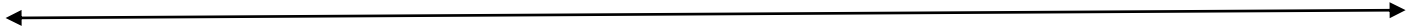


Figura 16 DCW "Generar reportes".

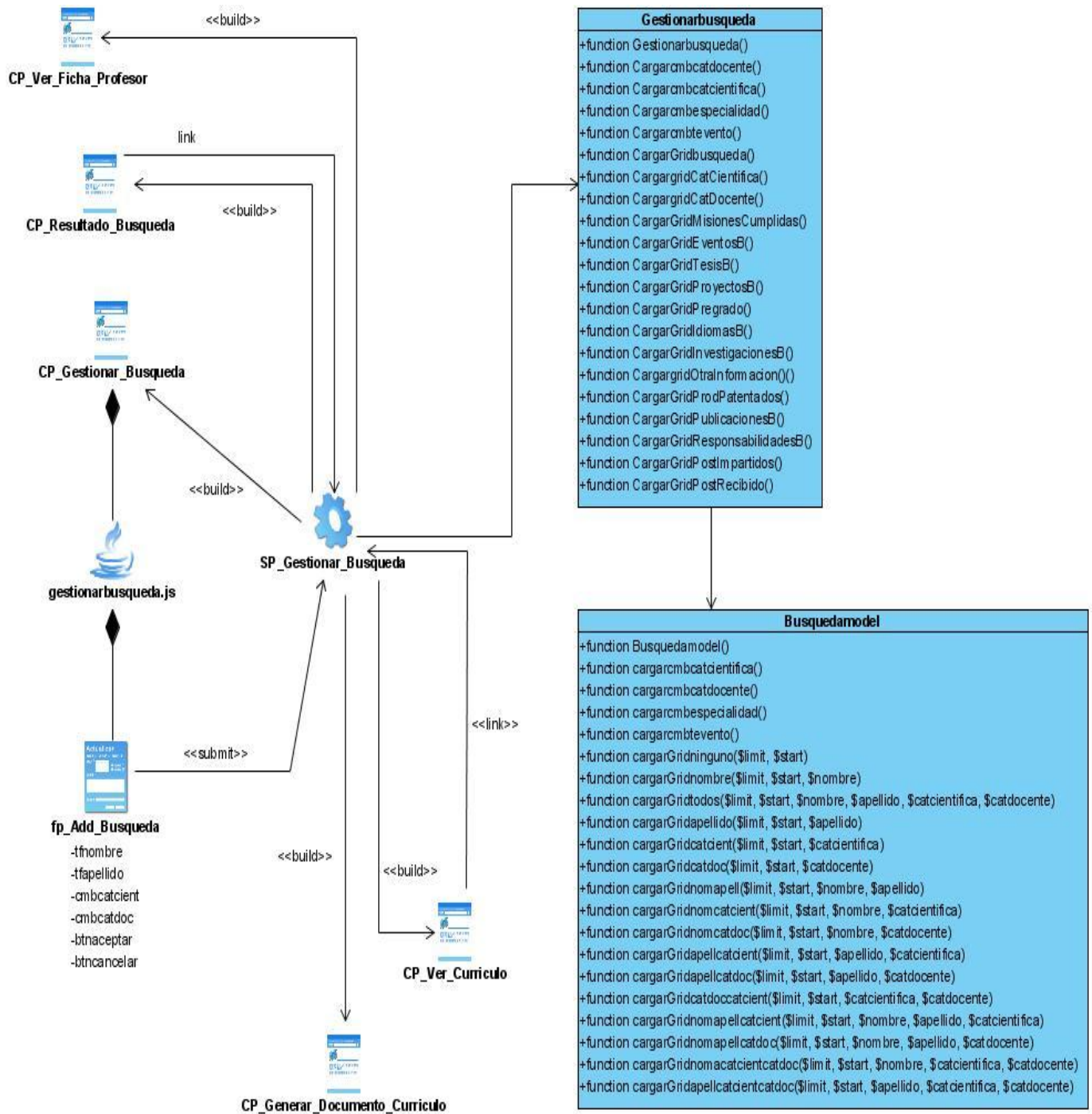


Figura 17 DCW "Gestionar busqueda".

Los demás Diagramas de Clases Web se encuentran en el Anexo 2. Diagrama de Clases Web..

3.2.4 Descripción de las Clases del Diseño

3.2.4.1 Descripción de Clases del Modelo

Nombre: Usuariomodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo
Nombre:	Descripción:
Usuariomodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
verificar(\$usuario,\$contrasena)	Verifica que los datos de los usuarios sean correctos.
verificarExistencia(\$usuario)	Verifica si el usuario existe en la base de datos.
dameldRol(\$rol_idrol)	Devuelve el id del rol.
add (\$usuario, \$contrasena, \$rol)	Adiciona un usuario en la base de datos.
cargarComborol()	Devuelve el listado de los roles.
cargarGridusuario(\$limit,\$start)	Devuelve el listado de usuarios.
cargarformusuario(\$id_usuario)	Devuelve los datos del usuario.
update (\$id_usuario, \$usuario, \$rol)	Modifica el usuario en la base de datos.
delete(\$id_usuario)	Elimina un usuario de la base de datos.

Tabla 10 Descripción de la clase del modelo "Usuariomodel".

Nombre: Profesormodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo
Nombre:	Descripción:
Profesormodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
add (\$nombre, \$papellido, \$sapellido, \$fechanacim, \$dirpart, \$telefono, \$correo, \$annograd, \$descripcion, \$especialidad, \$idusuario, \$dirimagen)	Adiciona los datos personales del profesor.
idprofesor()	Devuelve el id del profesor.
cargarform(\$id_profesor)	Devuelve los datos personales del profesor para el formulario de modificar.
update (\$nombre, \$papellido, \$sapellido, \$fechanacim, \$dirpart, \$telefono, \$correo, \$annograd, \$descripcion, \$especialidad, \$idusuario, \$dirimagen, \$idprofesor)	Modifica los datos personales del profesor en la base de datos.
delete(\$id_profesor)	Elimina el profesor de la base de datos.
dameidcurriculo(\$idprofesor)	Devuelve el id del currículo asociado al profesor.
updateimage(\$idprofesor)	Modificar imagen personal.

update (\$id_usuario, \$usuario, \$rol)	Modifica el usuario en la base de datos.
delete(\$id_usuario)	Elimina un usuario de la base de datos.

Tabla 11 Descripción de la clase del modelo "Profesormodel".

Nombre: Busquedamodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo
Nombre:	Descripción:
Busquedamodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
cargarcmcatdocente()	Devuelve el listado de las categorías docentes.
cargarcmcatcientifica()	Devuelve el listado de las categorías científicas.
cargarcmbspecialidad()	Devuelve el listado de las especialidades.
cargarcmbevento()	Devuelve el listado de los eventos.
dameidcurriculo(\$idprofesor)	Devuelve el id del currículo.
cargarGridvacio(\$limit , \$start)	Carga el grid vacío.
cargarGridtodos(\$limit, \$start, \$nombre, \$apellido,	Devuelve el resultado de la búsqueda.

\$catcientifica, \$catdocente)	
cargarGridnombre(\$limit, \$start, \$nombre)	Devuelve un listado con los nombres.
cargarGridapellido(\$limit, \$start, \$apellido)	Devuelve un listado con los nombres.

Tabla 12 Descripción de la clase del modelo "Busquedamodel".

Nombre: Reportemodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo
Nombre:	Descripción:
Reportemodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
cargarcmbcursospost ()	Devuelve el listado de los cursos de postgrado.
cargarcmbasignatura ()	Devuelve el listado de las asignaturas impartidas.
dameidcurriculo(\$idprofesor)	Devuelve el id del currículo asociado al profesor.
cargarGridnomPost (\$limit, \$start, \$cursoPost)	Devuelve los datos de los cursos de postgrado.
cargarGridEspe(\$limit, \$start, \$especialidad)	Devuelve los datos de las especialidades.



function cargarGridCatDoc(\$limit, \$start , \$catdoc)	Devuelve los datos de las categorías docentes.
cargarGridCatCient(\$limit, \$start, \$catcient)	Devuelve los datos de las categorías científicas.
cargarGridAsig(\$limit, \$start, \$asignatura)	Devuelve los datos de las asignaturas.

Tabla 13 Descripción de la clase del modelo "Reportemodel".

Las demás descripciones de las clases del modelo se encuentran en el Anexo 3. Descripción de Clases del Modelo.

3.2.4.2 Descripción de las Clases Controladoras

Nombre: Profesor.	
Tipo de clase: Controladora.	
Atributo:	Tipo
Nombre:	Descripción:
Profesor ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
index()	Carga la vista del profesor.

Tabla 14 Descripción de la clase controladora "Profesor".

Nombre: Usuario.
Tipo de clase: Controladora.

Atributo:	Tipo
Nombre:	Descripción:
Usuario ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
index()	Carga la vista del usuario.

Tabla 15 Descripción de la clase controladora "Usuario".

Nombre: Administrador.	
Tipo de clase: Controladora.	
Atributo:	Tipo
Nombre:	Descripción:
Administrador ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
index()	Carga la vista del administrador.

Tabla 16 Descripción de la clase controladora "Administrador".

Nombre: Directivo.
Tipo de clase: Controladora.



Atributo:	Tipo
Nombre:	Descripción:
Directivo ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
index()	Carga la vista del directivo.

Tabla 17 Descripción de la clase controladora "Directivo".

Nombre: Autenticarse.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$_model	Objeto de tipo Usuariomodel.
Nombre:	Descripción:
Autenticarse ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Verificar()	Verifica que los datos del usuario sean correctos, y una vez que estos sean verificados, re direcciona la aplicación hacia la vista que le corresponda según su usuario.

Tabla 18 Descripción de la clase controladora "Autenticar usuario".

Nombre: Gestionarusuario.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objeto de tipo Usuariomodel.
Nombre:	Descripción:
Gestionarusuario ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add()	Adiciona un usuario nuevo en la base de datos.
Cargarcmbrol()	Carga el combo con los roles existentes.
CargarGridusuario()	Carga los datos de los usuarios en el grid.
Cargarformusuario()	Cargar los datos de los usuarios en el formulario de modificar usuario.
Update()	Modifica un usuario.
Delete()	Elimina un usuario.

Tabla 19 Descripción de la clase controladora "Gestionar usuarios".

Nombre: Gestionarprofesor.	
Tipo de clase: Controladora.	
Atributo:	Tipo

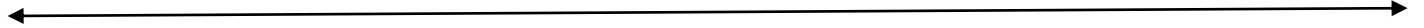
\$ _model	Objetode de tipo Profesormodel.
Nombre:	Descripción:
Gestionarprofesor ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add()	Adiciona los datos personales de un nuevo profesor en la base de datos.
Dameimg ()	Retorna la imagen personal del profesor.
DameimgForm ()	Modifica la imagen personal del profesor y la muestra en el formulario actualizada.
Guardarfoto()	Guarda la imagen personal del profesor en la base de datos.
Update()	Modifica los datos personales de un profesor.
Delete()	Elimina los datos personales de un profesor.
CargarGridprofesor()	Carga los datos personales del profesor en el grid.
Cargarformprofesor()	Carga los datos del formulario para modificar los datos personales del profesor
Salir()	Destruye la sesión creada inicialmente.

Tabla 20 Descripción de la clase controladora "Gestionar datos personales".

Nombre: Gestionarbusqueda.
Tipo de clase: Controladora.

Atributo:	Tipo
\$_model	Objetode de tipo Busquedamodel.
\$_modelp	Objeto de tipo Profesormodel.
\$_modelcf	Objeto de tipo Catcientificamodel.
\$_modelcd	Objeto de tipo Catdocentemodel.
\$_modelm	Objeto de tipo Misionmodel.
\$_modele	Objeto de tipo Eventomodel.
\$_modelt	Objeto de tipo Tesismodel.
\$_modelproy	Objeto de tipo Proyectomodel.
\$_modelcp	Objeto de tipo Cursospregradomodel.
\$_modeli	Objeto de tipo Idiomasmodel.
\$_modelinv	Objeto de tipo Investigacionmodel.
\$_modelo	Objeto de tipo Otrainformacionmodel.
\$_modelprod	Objeto de tipo Productomodel.
\$_modelpub	Objeto de tipo Publicacionmodel.
\$_modelr	Objeto de tipo Responsabilidadmodel.
\$_modelposi	Objeto de tipo Cursospostimpartidomodel.
_modelposr	Objeto de tipo Cursospostrecibidomodel.

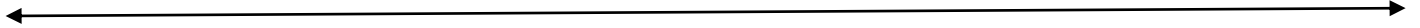
Nombre:	Descripción:
Gestionarbusqueda ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Cargarcmbcatdocente()	Carga el combo de las categorías docentes.
Cargarcmbcatcientifica ()	Carga el combo de las categorías científicas.
Cargarcmbespecialidad()	Carga el combo de las especialidades.
Cargarcmbevento()	Carga el combo de los tipos de eventos.
CargarGridbusqueda ()	Carga el grid con el resultado de la búsqueda.
CargargridCatCientifica ()	Carga el grid de la categoría científica.
CargargridCatDocente ()	Carga el grid de la categoría docente.
CargarGridMisionesCumplidas ()	Carga el grid de las misiones cumplidas.
CargarGridEventosB ()	Carga el grid de los eventos.
CargarGridProyectosB ()	Carga el grid de los proyectos.
CargarGridPregrado ()	Carga el grid de los cursos de pregrado.
CargarGridIdiomasB ()	Carga el grid de los idiomas.
CargarGridInvestigacionesB ()	Carga el grid de las investigaciones.
CargargridOtraInformacion ()	Carga el grid con otras informaciones.
CargarGridProdPatentados ()	Carga el grid de los productos patentados.



CargarGridPublicacionesB ()	Carga el grid de las publicaciones.
CargarGridResponsabilidadesB ()	Carga el grid de las responsabilidades.
CargarGridPostImpartidos ()	Carga el grid de los cursos de postgrado impartidos
CargarGridPostRecibido ()	Carga el grid de los cursos de postgrado recibidos.

Tabla 21 Descripción de la clase controladora "Gestionar búsqueda".

Nombre: Gestionarreportes.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objetode de tipo Busquedamodel.
\$ __modelp	Objeto de tipo Profesormodel.
\$ __modelr	Objeto de tipo Reportesmodel.
Nombre:	Descripción:
Gestionarreportes ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Cargarcmbcatdocente()	Carga el combo de las categorías docentes.
Cargarcmbcatcientifica ()	Carga el combo de las categorías científicas.
Cargarcmbespecialidad()	Carga el combo de las especialidades.



Cargarcmbcursospostimp ()	Carga el combo de los cursos de postgrado impartidos.
Cargarcmbcursospostrec ()	Carga el combo de los cursos de postgrado recibidos.
Cargarcmbasignatura ()	Carga el combo de los cursos de pregrado impartidos.
CargarGridreporte()	Carga el grid del resultado del reporte.

Tabla 22 Descripción de la clase controladora "Generar reportes".

Para ver las descripciones de las demás clases controladoras, ver

Anexo 4. Descripción de Clases *Controladoras*.

3.2.5 Diseño de la Base de Datos.

En la base de datos es donde se almacenarán todos los datos con los cuales trabaja el sistema. En el diseño de la base de datos se deben tener en cuenta las tablas, índices, vistas, restricciones, procedimientos almacenados y cualquier otro elemento que se necesite para almacenar, recuperar y eliminar objetos persistentes, siendo de gran importancia elaborar el diagrama de entidad-relación, el cual muestra las relaciones que existen entre las tablas de la base de datos.

3.2.5.1 Diagrama Entidad-Relación de la Base de Datos.

Debido a la extensión del diagrama Entidad-Relación, el mismo fue dividido en 3 partes para su mejor análisis y entendimiento. La primera parte, muestra la relación existente entre las tablas usuario, rol, currículo y profesor. En las 2 siguientes se muestran las relaciones existentes entre la tabla currículo y las demás tablas que conforman como un todo la información que tendrá el mismo.

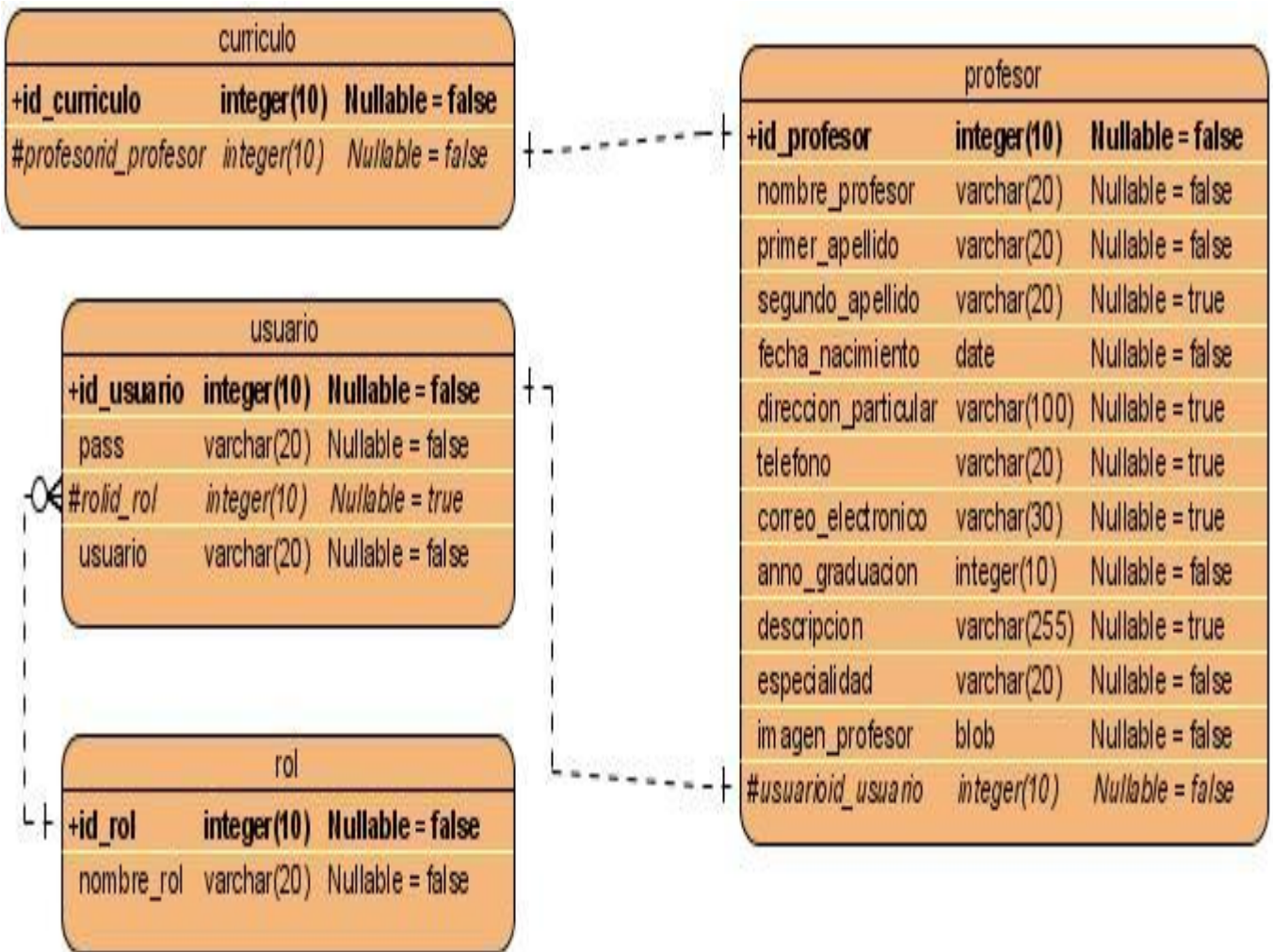


Figura 18 DER Relación rol-usuario-profesor-curriculo.

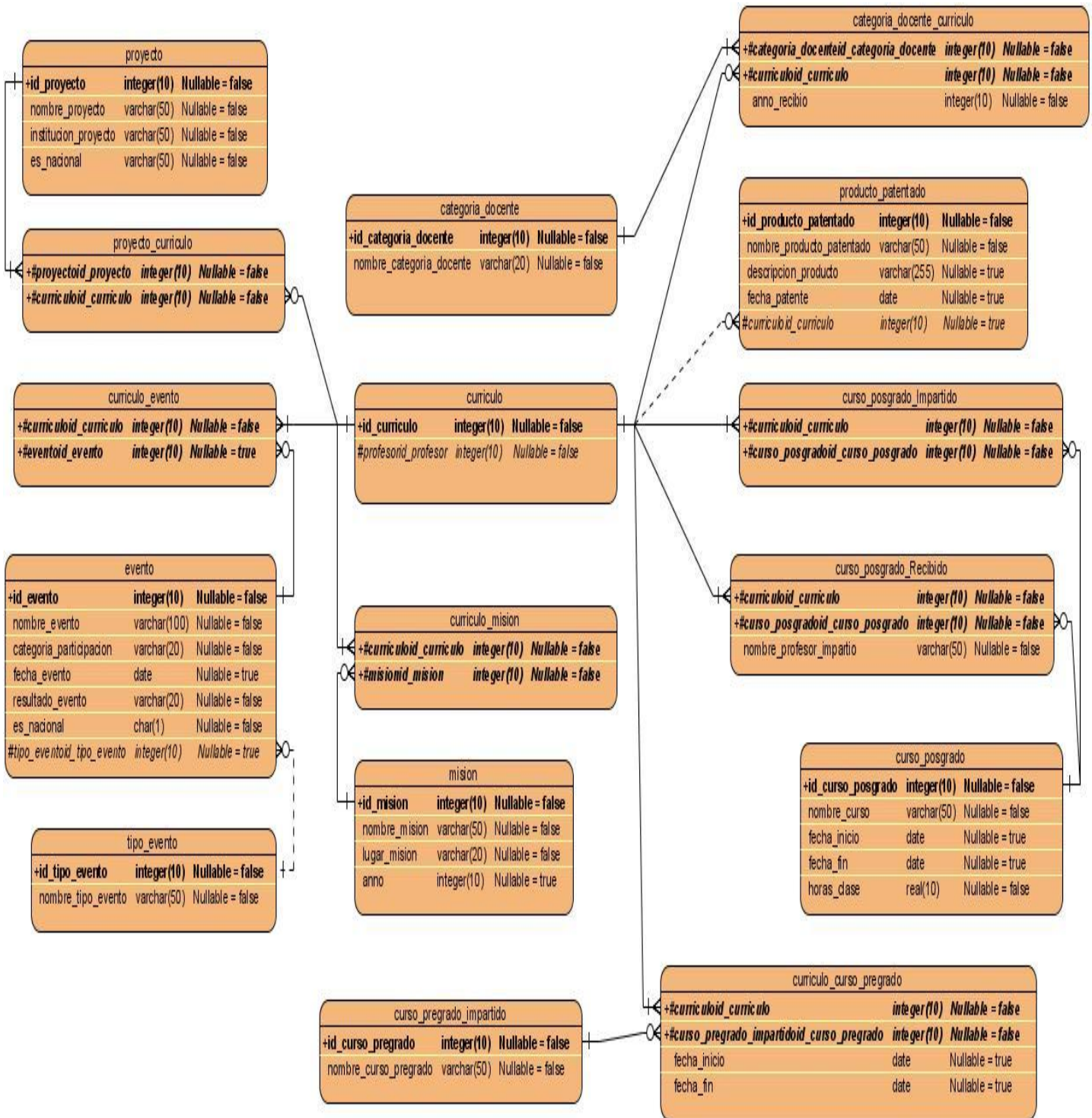


Figura 19 DER Currículo-Información del currículum (parte 1).

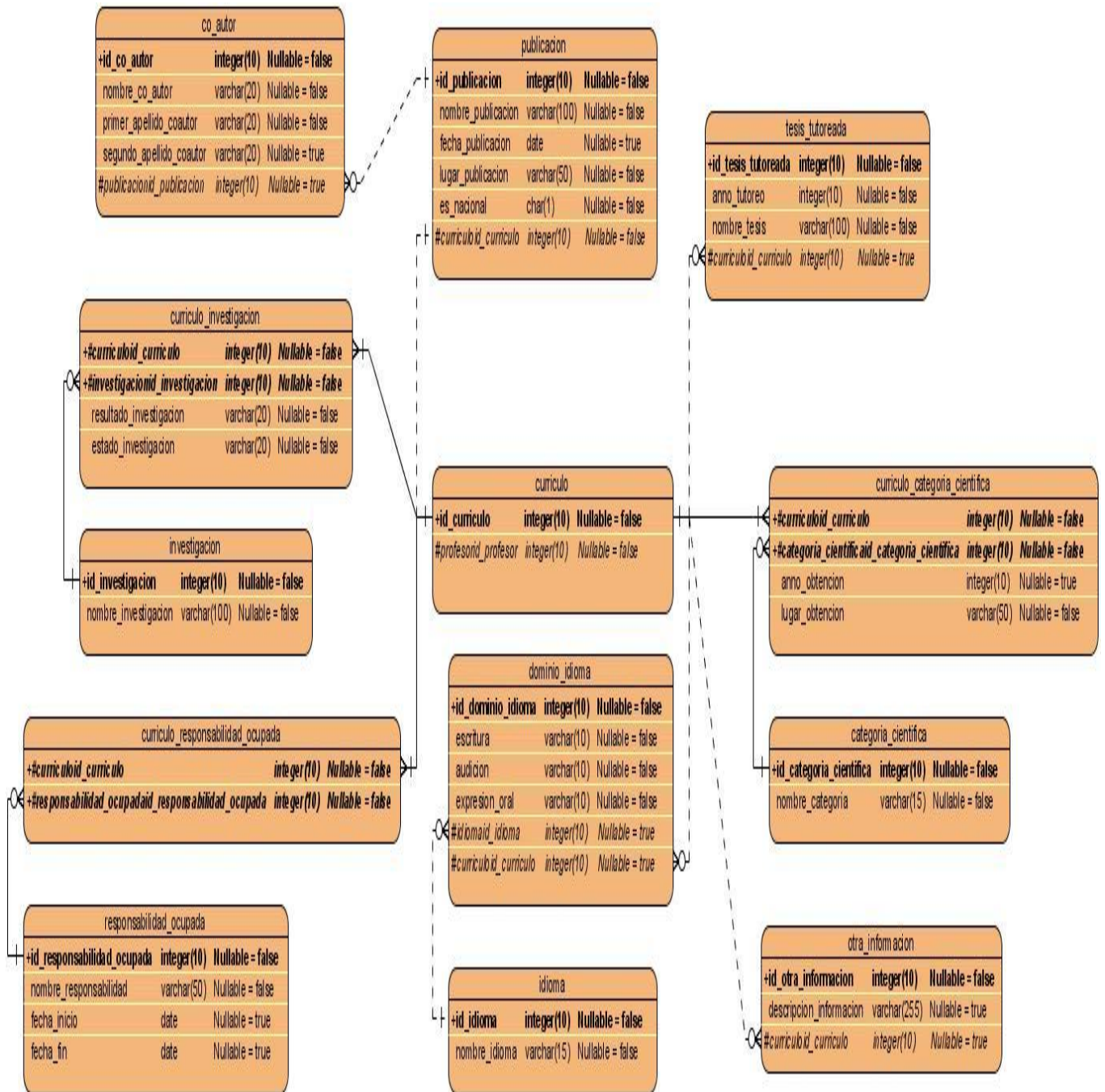
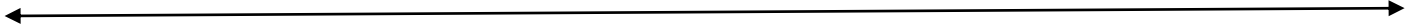


Figura 20 DER Relación Currículo-Información del currículo (parte 2).



3.2.5.2. Descripción de las tablas de la Base de Datos.

Nombre: profesor.		
Descripción: La tabla almacena los datos personales del profesor.		
Atributo	Tipo	Descripción
id_profesor	integer	Es el id por el que se identificará al profesor. Es la llave primaria de la tabla.
nombre_profesor	varchar	Nombre del profesor.
primer_apellido	varchar	Primer apellido del profesor.
fecha_nacimiento	date	Fecha de nacimiento del profesor.
direccion_particular	varchar	Dirección particular del profesor.
telefono	varchar	Teléfono del profesor.
correo_electronico	varchar	Correo electrónico del profesor.
anno_graduacion	integer	Año de graduación del profesor.
descripcion	text	Breve descripción profesional.
especialidad	varchar	Especialidad en la que se graduó el profesor.
imagen_profesor	bytea	Imagen personal para el currículum del profesor.
usuarioid_usuario	integer	El id de usuario que tendrá el profesor.

Tabla 23 Descripción de la tabla "profesor".

Nombre: usuario.		
Descripción: La tabla almacena los datos del usuario.		
Atributo	Tipo	Descripción
id_usuario	integer	Es el id por el que se identifica el usuario.
usuario	varchar	Nombre de usuario.

pass	varchar	Contraseña del usuario.
rolid_rol	integer	Es el id del rol asignado al usuario.

Tabla 24 Descripción de la tabla "usuario".

Nombre: rol.		
Descripción: La tabla almacena los roles existentes en la aplicación.		
Atributo	Tipo	Descripción
id_rol	integer	Es el id por el que se identifica el rol.
nombre_rol	varchar	Nombre del rol.

Tabla 25 Descripción de la tabla "rol".

Nombre: curriculum.		
Descripción: La tabla almacena el id del profesor asociado al currículum y el id del currículum. Sirve como puente entre el profesor y la información del currículum asociado.		
Atributo	Tipo	Descripción
id_curriculo	integer	Es el id del curriculum.
profesorid_profesor	integer	Profesor al que le pertenece el curriculum.

Tabla 26 Descripción de la tabla "curriculum".

Nombre: categoria_cientifica.		
Descripción: Es un nomenclador con los nombres de las categorías científicas existentes.		

Atributo	Tipo	Descripción
id_categoria_cientifica	integer	Id de la categoría científica.
nombre_categoria	varchar	Nombre de la categoría científica

Tabla 27 Descripción de la tabla "categoría_científica".

Nombre: categoría_docente.		
Descripción: Almacena los datos del nomenclador categoría docente.		
Atributo	Tipo	Descripción
id_categoria_docente	integer	Es el id la categoría docente.
nombre_categoria_docente	varchar	Nombre de la categoría docente.

Tabla 28 Descripción de la tabla "categoría_docente".

Nombre: categoría_docente_curriculo.		
Descripción: La tabla se genera a partir de la relación entre el currículo y la categoría docente.		
Atributo	Tipo	Descripción
categoria_docenteid_categoria_docente	integer	Es el id del nomenclador categoría docente que le es asignada a la categoría.
curriculoid_curriculo	integer	Id del currículo al que pertenece la categoría.
anno_recibió	integer	Año en que recibió la categorización docente.

Tabla 29 Descripción de la tabla "categoría_docente_curriculo".

Nombre: co_autor.		
Descripción: Almacena los datos de un posible co-autor de una publicación.		
Atributo	Tipo	Descripción
id_co_autor	integer	Es el id del co-autor.
nombre_co_autor	varchar	Nombre del co-autor.
primer_apellido_co_autor	varchar	Primer apellido del co-autor.
segundo_apellido_co_autor	varchar	Segundo apellido del co-autor.
publicacionid_publicacion	integer	Es el id asociado de la publicación.

Tabla 30 Descripción de la tabla "co_autor".

Nombre: curriculum_categoria_cientifica.		
Descripción: Es la relación entre la categoría científica y el currículum, y contiene los datos de la categoría científica específicos para un profesor.		
Atributo	Tipo	Descripción
curriculoid_curriculo	integer	Es el id del currículum al que pertenece la categoría científica.
categoria_cientificaid_categoria_cientifica	integer	Id del nomenclador categoría científica asociado.
anno_obtencion	integer	Año de obtención de la categoría científica.

lugar_obtencion	varchar	Lugar en que obtuvo la categoría científica.
especialidad	varchar	Especialidad en la que recibió la categoría científica.

Tabla 31 Descripción de la tabla "curriculo_categoria_cientifica".

Nombre: curriculo_curso_pregrado.		
Descripción: Es la relación que existe entre el currículum y los cursos de pregrado, contiene los datos específicos para un profesor.		
Atributo	Tipo	Descripción
curriculoid_curriculo	integer	Es el id del currículum al que pertenece el curso.
curso_pregrado_impardoid_curso_pregrado	integer	Id del curso de pregrado asociado.
fecha_inicio	date	Fecha de comienzo del curso.
fecha_fin	date	Fecha de culminación del curso.

Tabla 32 Descripción de la tabla "curriculo_curso_pregrado".

Nombre: curriculo_evento.		
Descripción: Es la relación entre el currículum y los eventos, contiene los datos de los eventos específicos para un profesor.		
Atributo	Tipo	Descripción
curriculoid_curriculo	integer	Es el id del currículum al que pertenece el evento.

eventoid_evento	integer	Id del evento asociado.
-----------------	---------	-------------------------

Tabla 33 Descripción de la tabla "curriculo_evento".

Nombre: curriculo_investigacion.		
Descripción: Es la relación entre el currículo y las investigaciones, contiene los datos de las investigaciones específicos para un profesor.		
Atributo	Tipo	Descripción
curriculoid_curriculo	integer	Es el id del currículo al que pertenece la investigación.
investigacionid_investigacion	integer	Id de la investigación asociada.
resultado_investigacion	varchar	Resultado de la investigación.
estado_investigacion	varchar	Estado de la investigación.

Tabla 34 Descripción de la tabla "curriculo_investigación".

Nombre: curriculo_mision.		
Descripción: Es la relación entre el currículo y las misiones cumplidas, contiene los datos de las misiones específicos para un profesor.		
Atributo	Tipo	Descripción
curriculoid_curriculo	integer	Es el id del currículo al que pertenece la misión.
misionnid_mision	integer	Id de la misión asociada.

Tabla 35 Descripción de la tabla "curriculo_mision".

Nombre: curriculum_responsabilidad_ocupada.		
Descripción: Es la relación entre el currículo y las responsabilidades ocupadas, contiene los datos de las responsabilidades específicos para un profesor.		
Atributo	Tipo	Descripción
curriculoid_curriculo	integer	Es el id del currículo al que pertenece la responsabilidad.
responsabilidad_ocupadaid_responsabilidad_ocupada	integer	Id de la responsabilidad asociada.

Tabla 36 Descripción de la tabla "curriculum_responsabilidad_ocupada".

Nombre: curso_postgrado.		
Descripción: Almacena los datos de los cursos de postgrado que hayan sido impartidos o recibidos.		
Atributo	Tipo	Descripción
id_curso_postgrado	integer	Id del curso de postgrado.
nombre_curso	varchar	Nombre del curso de postgrado.
fecha_inicio	date	Fecha en que comenzó el curso.
fecha_fin	date	Fecha en que culminó el curso.
horas_clases	real	Cantidad de horas clase que tiene el curso.

Tabla 37 Descripción de la tabla "curso_postgrado".

Nombre: curso_postgrado_impartido.		
Descripción: Almacena los datos específicos si el profesor impartió el curso.		

Atributo	Tipo	Descripción
curriculoid_curriculo	integer	Es el id del currículo al que pertenece el curso.
curso_postgradoid_curso_postgrado	integer	Id del curso de postgrado asociado.

Tabla 38 Descripción de la tabla "curso_postgrado_Impartido".

Nombre: curso_postgrado_Recibido.		
Descripción: Almacena los datos específicos si el profesor recibió el curso.		
Atributo	Tipo	Descripción
curriculoid_curriculo	integer	Es el id del currículo al que pertenece el curso.
curso_postgradoid_curso_postgrado	integer	Id del curso de postgrado asociado.
profesor_impertio	varchar	Nombre del profesor que impartió el curso.

Tabla 39 Descripción de la tabla "curso_postgrado_Recibido".

Nombre: dominio_idioma.		
Descripción: Almacena los datos referentes al dominio de idiomas.		
Atributo	Tipo	Descripción
id_dominio_idioma	integer	Id del dominio del idioma.
escritura	varchar	Descripción del dominio de la escritura.
audicion	varchar	Descripción del dominio de la audición.
expresion_oral	varchar	Descripción del dominio de la expresión oral.

idiomaid_idioma	integer	Id del nombre del idioma.
curriculoid_curriculo		Id del currículo al que pertenece.

Tabla 40 Descripción de la tabla "dominio_idioma".

Nombre: evento.		
Descripción: Almacena los datos de los eventos.		
Atributo	Tipo	Descripción
id_evento	integer	Id del evento.
nombre_evento	varchar	Nombre del evento.
tipo_eventoid_tipo_evento	integer	Id del tipo de evento asociado.
categoria_participacion	varchar	Categoría de la participación en el evento.
es_nacional	varchar	Nacionalidad del evento, nacional o internacional.
resultado_evento	varchar	Resultado alcanzado en el evento.
fecha_evento	date	Fecha en que se realizó el evento.

Tabla 41 Descripción de la tabla "evento".

Nombre: idioma.		
Descripción: Almacena los datos del nomenclador idioma.		
Atributo	Tipo	Descripción
id_idioma	integer	Id del idioma.
nombre_evento	varchar	Nombre del idioma.



Tabla 42 Descripción de la tabla "idioma".

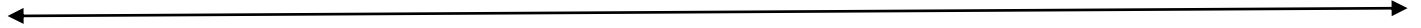
Nombre: investigacion.		
Descripción: Almacena los datos de las investigaciones realizadas.		
Atributo	Tipo	Descripción
id_investigacion	integer	Id de la investigación.
nombre_investigacion	varchar	Nombre de la investigación.

Tabla 43 Descripción de la tabla "investigacion".

Nombre: mision.		
Descripción: Almacena los datos de las misiones cumplidas.		
Atributo	Tipo	Descripción
id_mision	integer	Id de la misión.
nombre_mision	varchar	Nombre de la misión.
lugar_mision	varchar	Lugar en la que se cumplió la misión.
anno	integer	Año en el que se cumplió la misión.

Tabla 44 Descripción de la tabla "mision".

Nombre: otra_informacion.		
Descripción: Almacena los datos de informaciones adicionales del currículum.		
Atributo	Tipo	Descripción
id_otra_mision	integer	Id de la información.



descripcion_información	text	Descripción de la información.
curriculoid_curriculo	integer	Id del currículo al cual pertenece.

Tabla 45 Descripción de la tabla "otra_informacion".

Nombre: producto_patentado.		
Descripción: Almacena los datos de los productos patentados.		
Atributo	Tipo	Descripción
id_producto_patentado	integer	Id de la información.
nombre_producto_patentado	varchar	Nombre del producto.
descripcion_producto	text	Id del currículo al cual pertenece.
fecha_patente	date	Fecha en que consiguió la patente del producto.
curriculoid_curriculo	integer	Id del currículo al que pertenece.

Tabla 46 Descripción de la tabla "producto_patentado".

Nombre: proyecto.		
Descripción: Almacena los datos de los proyectos.		
Atributo	Tipo	Descripción
id_proyecto	integer	Id del proyecto.
nombre_proyecto	varchar	Nombre del proyecto.
institucion_proyecto	varchar	Institución con la cual se realizó el proyecto.
es_nacional	varchar	Nacionalidad del proyecto, nacional o internacional.



Tabla 47 Descripción de la tabla "proyecto".

Nombre: proyecto_curriculo.		
Descripción: Representa la relación entre el currículum y los proyectos, y guarda los datos específicos de los proyectos en los que ha participado un profesor.		
Atributo	Tipo	Descripción
proyectoid_proyecto	integer	Id del proyecto asociado.
curriculoid_curriculo	integer	Id del currículum asociado.

Tabla 48 Descripción de la tabla "proyecto_curriculo".

Nombre: publicacion.		
Descripción: Almacena los datos de las publicaciones.		
Atributo	Tipo	Descripción
id_publicacion	integer	Id de la publicación.
nombre_publicacion	varchar	Nombre de la publicación.
lugar_publicacion	varchar	Lugar en la que se realizó la publicación.
fecha_publicacion	date	Fecha en la que se realizó la publicación.
es_nacional	varchar	Nacionalidad de la publicación, nacional o internacional.
curriculoid_curriculo		Id del currículum al que pertenece.

Tabla 49 Descripción de la tabla "publicacion".

Nombre: responsabilidad_ocupada.

Descripción: Almacena los datos de las responsabilidades ocupadas.		
Atributo	Tipo	Descripción
id_responsabilidad_ocupada	integer	Id de la responsabilidad.
nombre_responsabilidad	varchar	Nombre de la responsabilidad.
fecha_inicio	date	Fecha en la que inició la responsabilidad.
fecha_fin	date	Fecha en la que terminó la responsabilidad.

Tabla 50 Descripción de la tabla "responsabilidad_ocupada".

Nombre: tesis_tutorada.		
Descripción: Almacena los datos de las tesis tutoradas.		
Atributo	Tipo	Descripción
id_tesis_tutoreada	integer	Id de la tesis tutorada.
anno_tutoreo	integer	Nombre de la tesis tutorada.
nombre_tesis	varchar	Nombre de la tesis tutorada.
curriculoid_curriculo	integer	Id del currículo al que pertenece.

Tabla 51 Descripción de la tabla "tesis_tutorada".

Nombre: tipo_evento.		
Descripción: Almacena los nombres de los tipos de eventos del nomenclador tipo de evento.		
Atributo	Tipo	Descripción
id_tipo_evento	integer	Id del tipo de evento.



nombre_tipo_evento	varchar	Nombre del tipo de evento.
--------------------	---------	----------------------------

Tabla 52 Descripción de la tabla "tipo_evento".

3.2.6. Definiciones de diseño.

El diseño es un elemento muy importante para desarrollar cualquier aplicación, el mismo debe basar sus características en el usuario con el cual interactúa. En este caso en particular son directivos, profesores, estudiantes y trabajadores de la facultad 1, los cuales en ocasiones no tienen amplios conocimientos de Informática, por ello, para desarrollar este sistema se ha decidido apoyarse en varios principios de diseño universal que garantizan el correcto aprovechamiento del mismo.

Simple e Intuitivo: debe eliminar complejidades innecesarias, ser consistente con las expectativas e intuición del usuario, que presente la información acorde con su importancia y que proporcione avisos eficaces durante la realización de una tarea.

Información perceptible: proporcionar contraste suficiente entre la información esencial y sus alrededores, ampliar la legibilidad de la información esencial.

Con tolerancia al error: proporcionar advertencias sobre peligros y errores, proporcionar características seguras de interrupción, disponer de elementos para minimizar los riesgos y errores.

Que exija poco esfuerzo físico: permitir que el usuario mantenga una posición corporal neutra, minimizar las acciones repetitivas, minimizar el esfuerzo físico continuado.

Tamaño y espacio para el acceso y uso: proporcione una línea de visión clara hacia los elementos importantes tanto para un usuario sentado como de pie, el alcance de cualquier componente debe ser confortable para cualquier usuario sentado o de pie.

3.2.7. Tratamiento de errores.

Los errores serán tratados de dos formas diferentes, aunque las dos son sumamente importantes para el correcto funcionamiento del Sistema. La primera es usando las validaciones que brinda el framework Ext, las cuales serán hechas en las Página Clientes para que el Sistema no se ejecute usando datos inconsistentes, mostrando mensajes de alerta instantáneos, de manera que el usuario sepa en el mismo instante cuál es la causa del error, siendo ésta forma la más aceptada generalmente por ellos (ver Figura 21 Mensaje de error. Validación en la Página Cliente.). La segunda forma ocurre en la Páginas

Servidoras, validando la entrada de datos incorrectos, y en caso de que se filtre un dato erróneo, se mostrarán mensajes embebidos en las Páginas Clientes, informando las causas que los originan (ver Figura 22 Mensaje de error. Validación en la Página Servidora.).

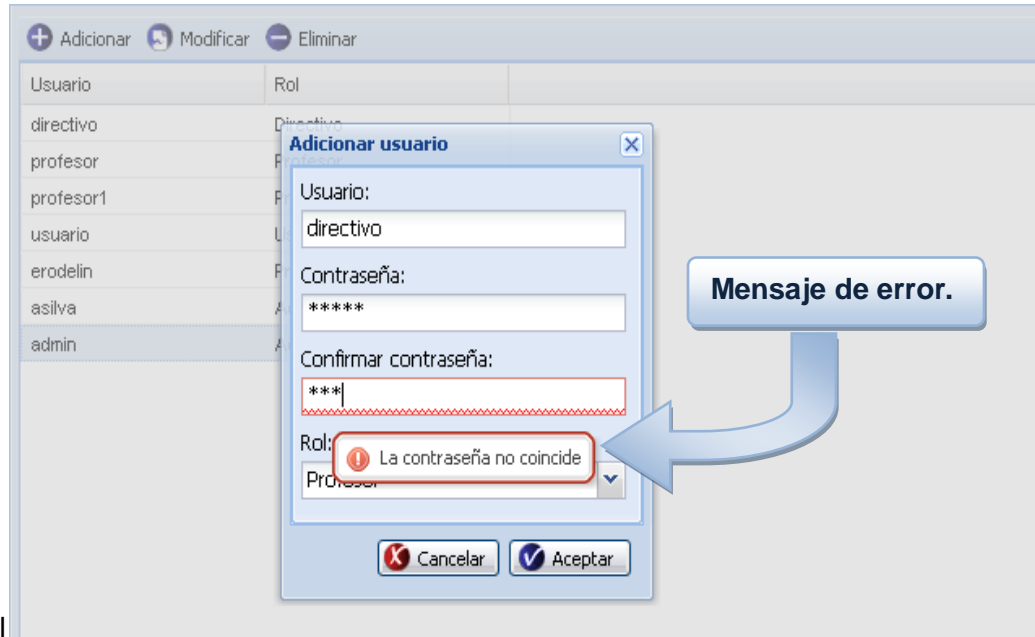


Figura 21 Mensaje de error. Validación en la Página Cliente.

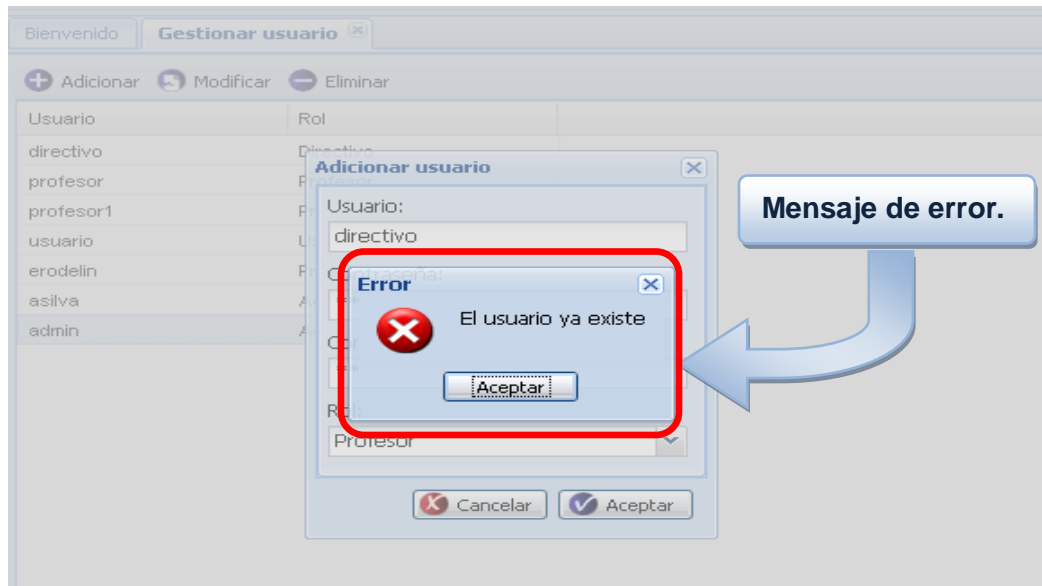


Figura 22 Mensaje de error. Validación en la Página Servidora.

3.2.8. Seguridad.

La seguridad del Sistema se basa fundamentalmente en el empleo de la autenticación como una acción de obligatorio y estricto cumplimiento para todos los usuarios que quieran tener acceso a la aplicación. De esta manera, cada usuario contará con un nombre de usuario y una contraseña, la cual será creada por el administrador, pero podrá ser modificada por el usuario de manera eventual, con el objetivo de garantizar además la seguridad de la información del usuario. De esta manera se garantiza evitar el uso incorrecto del Sistema. Además, solamente el administrador del Sistema podrá crear un nuevo usuario, y una vez creado, asignarle un rol.

La utilización de roles es otra técnica que se utiliza con el propósito de garantizar la seguridad, donde cada rol tendrá acceso a determinadas funcionalidades de acuerdo a su desempeño. Por otra parte, el Sistema implementa un mecanismo de encriptación utilizando MD5 para la encriptación de las contraseñas, de manera que las mismas viajan por la red, y son almacenadas en la base de datos encriptadas.

3.2.9. Interfaz.

La interfaz está confeccionada siguiendo el esquema Cabecera-Navegador-Contenido, con el objetivo de lograr un diseño consistente. Se utilizan las tablas (grids) para mostrar la información que se maneja. Se utiliza por otro lado la hoja de estilos que ofrece el framework Ext. Predominan tonalidades de color azul claro combinadas con el color blanco y azul oscuro para lograr una interfaz agradable a la vista del usuario.

El color predominante es el azul, un color que da mucho juego en las composiciones gráficas en general y en las aplicaciones web en concreto. Es utilizado ampliamente como color corporativo, por la seriedad y confianza que inspira y admite buenas gradaciones, pudiendo ser el color predominante en una página. El diseño simétrico de la interfaz sugiere estabilidad, equilibrio, resultando estético, ordenado, atractivo y agradable de contemplar. La simetría usada no es del todo exacta, ya que una simetría demasiado perfecta hace parecer las composiciones artificiales y premeditadas. Pequeñas variaciones en la distribución simétrica dan ese toque de ruptura que hace su contemplación más amena y natural.

3.2.10. Estándares de codificación.

El estándar de codificación que se utilizará para lograr la organización y un fácil entendimiento del código

si en un futuro fuera necesario, es el siguiente:

3.3.10.1 Declaraciones:

Variables explícitas. Ejemplo: var \$nombre_profesor; var \$primer_apellido_profesor.

La declaración de Clases se hará de la siguiente manera:

- Para las clases del modelo, comenzarán con el nombre con mayúscula, seguido por la palabra model. Ejemplo: Profesormodel.
- Para las clases controladoras, comenzarán con la palabra gestionar con mayúscula, seguido por el nombre. Ejemplo: Gestionarprofesor.
- Para las vistas, serán el nombre de la vista con mayúscula. Ejemplo: Profesor.
- Los métodos en las clases controladoras comenzarán con letra mayúscula. Ejemplo: Add().
- Los métodos en las clases del modelo comenzarán con letra minúscula. Ejemplo add().

3.3.10.2 Espacios en blanco

Espacios en blanco entre operadores lógicos-aritméticos y sus operandos.


```
if($datos == -1)
```

3.2.11. Concepción de la ayuda.

Esta aplicación va dirigida principalmente a profesores de la UCI, aunque tendrán acceso al mismo, estudiantes y trabajadores, y todos tienen un dominio de la informática que no se puede considerar mínimo. Por lo que la ayuda estará centrada en explicar las acciones a tomar, así como los problemas que se puedan presentar, más allá de los problemas técnicos de la aplicación o conceptos ambiguos.

Conclusiones

En el capítulo se desarrolló la modelación del sistema, realizándose los diagramas de clases del análisis y el diseño, con sus respectivos diagramas de interacción, específicamente de secuencia, así como el diseño de la base de datos. Además, se describieron detalladamente las clases del diseño y las tablas de la base de datos, así como los aspectos de seguridad, interfaz, patrones de diseño, definiciones de diseño. Este capítulo es de suma importancia para el desarrollo del sistema, porque el resultado del



mismo, que son los artefactos generados, constituyen la entrada para el flujo de trabajo de implementación, debido a que brinda una vista de la arquitectura del sistema, y describe detalladamente las acciones a tomar en el siguiente flujo de trabajo.



Capítulo 4: Implementación

Introducción

En el presente Capítulo se parte del resultado del flujo de trabajo Análisis y Diseño, y se implementa el sistema en términos de componentes. En él se muestra el diagrama de despliegue en el cual se representan los nodos del sistema, basados en las características y funcionalidades que debe cumplir. También se muestra en el modelo de implementación los diagramas de los componentes que contiene la aplicación.

4.1. Modelo de despliegue.

En el modelo de despliegue se representa la configuración de los tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes. En el mismo se muestra una computadora (PC) cliente desde la cual el usuario accederá al sistema. Una PC Servidor Web donde estará emplazada la aplicación, una PC Servidor de Base de Datos en la cual estará ubicada la base de datos que utilizará el sistema y una Impresora para que el profesor pueda imprimir su currículum, y el directivo además los reportes generados por la aplicación. Para mejor comprensión ver Figura 23 Modelo de Despliegue.

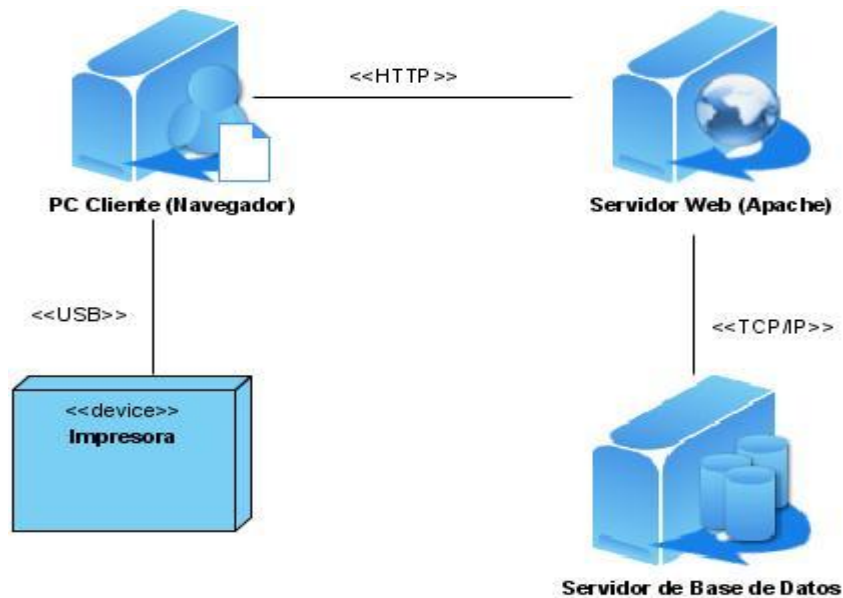


Figura 23 Modelo de Despliegue.

4.2. Modelo de Implementación.

El Modelo de Implementación es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. (Ver Figura 24 Modelo de Implementación.)

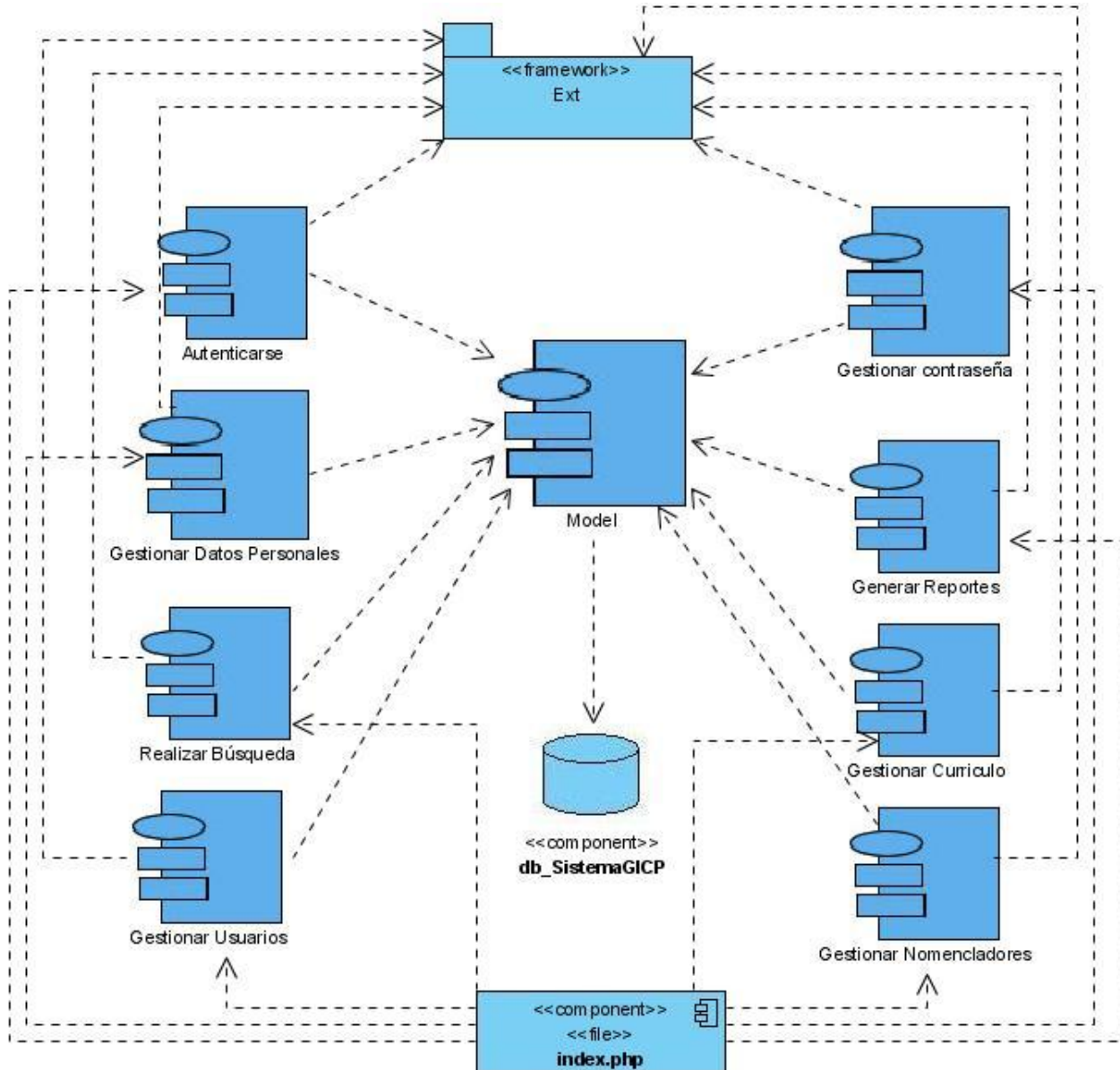


Figura 24 Modelo de Implementación.

4.3. Diagrama de componentes.

Un componente es una parte física y reemplazable de un sistema, que consiste en la implementación física de un conjunto de otros elementos lógicos, como pueden ser clases y colaboraciones. En el diagrama de componentes del sistema, se expresan las dependencias que existen entre los mismos para desarrollar la aplicación.

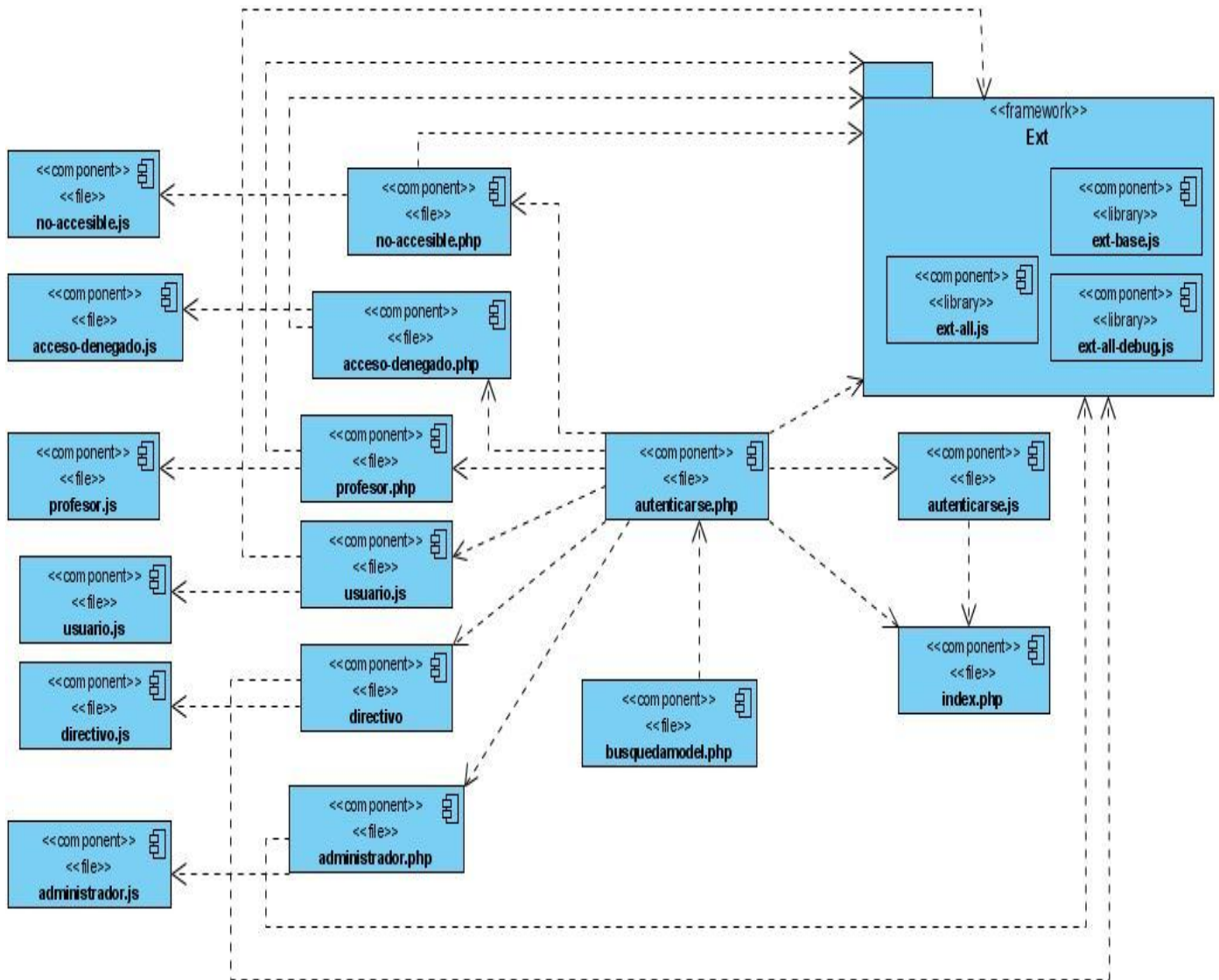


Figura 25 Diagrama de Componentes. Autenticarse.

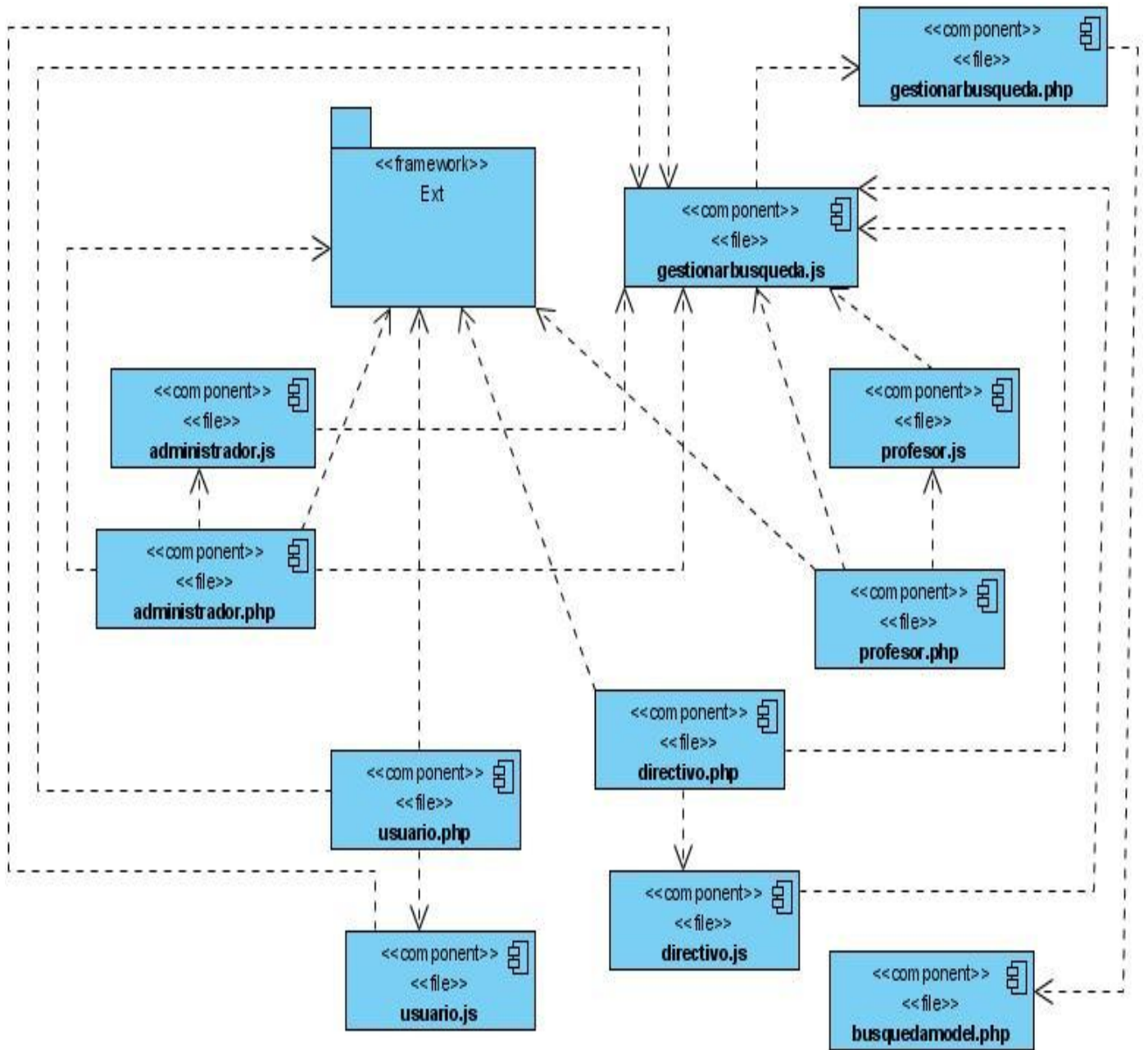


Figura 26 Diagrama de componentes. Realizar Búsqueda.

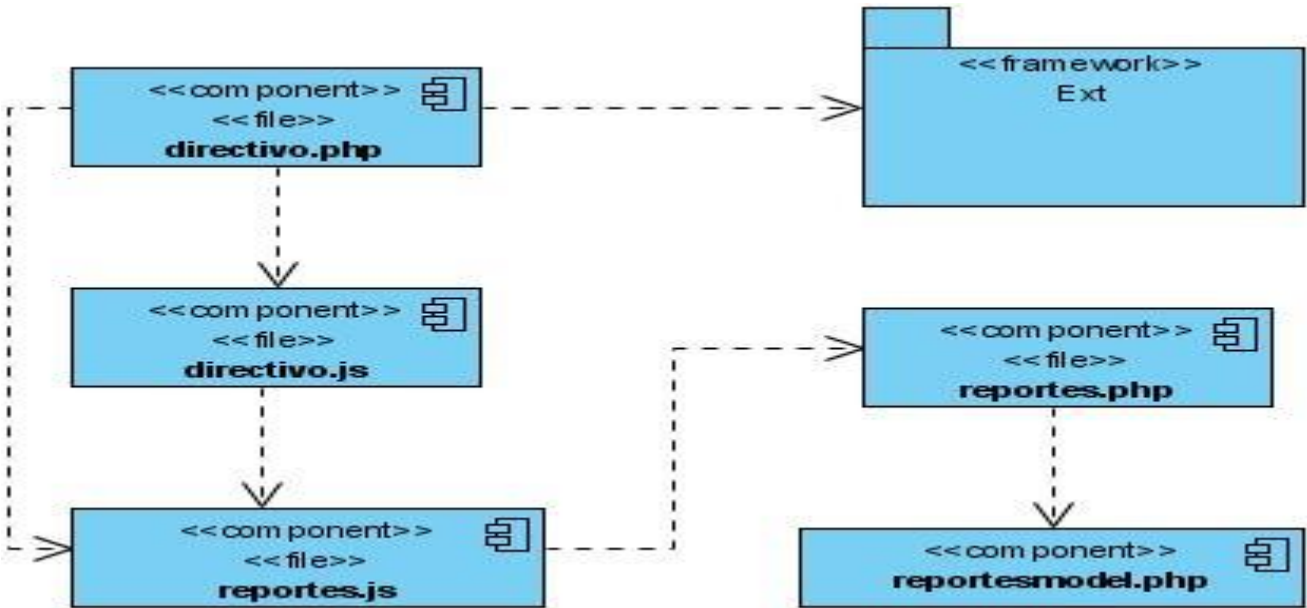


Figura 27 Diagrama de componentes. Generar Reportes.

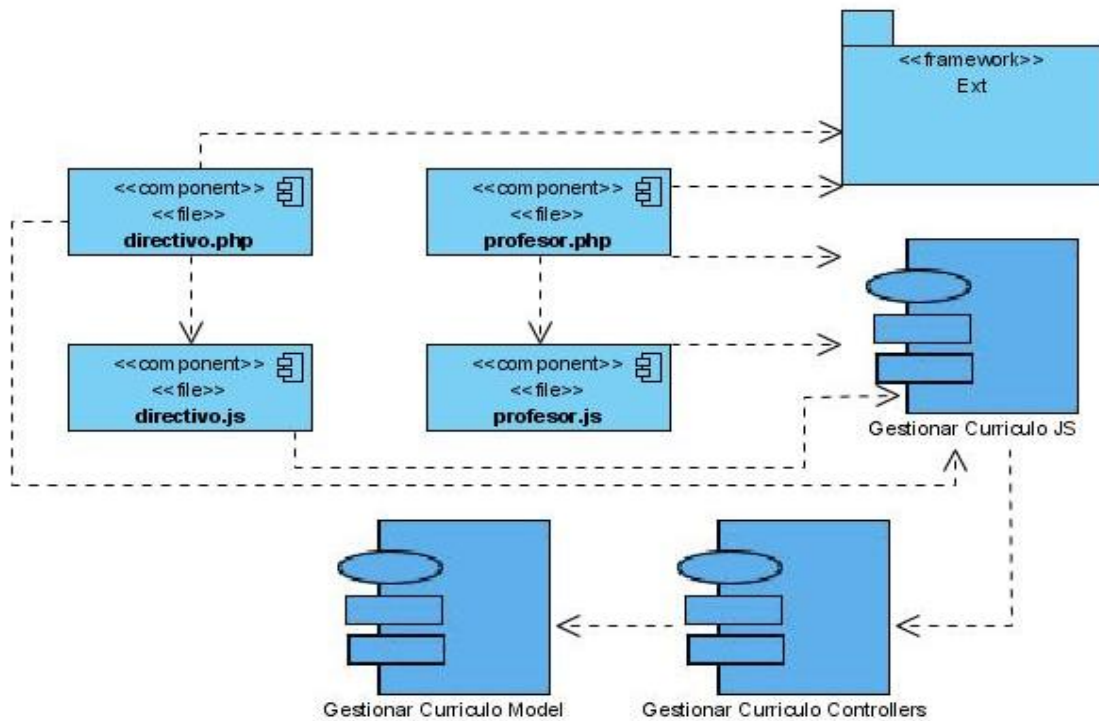


Figura 28 Diagrama de componentes. Gestionar Currículo.

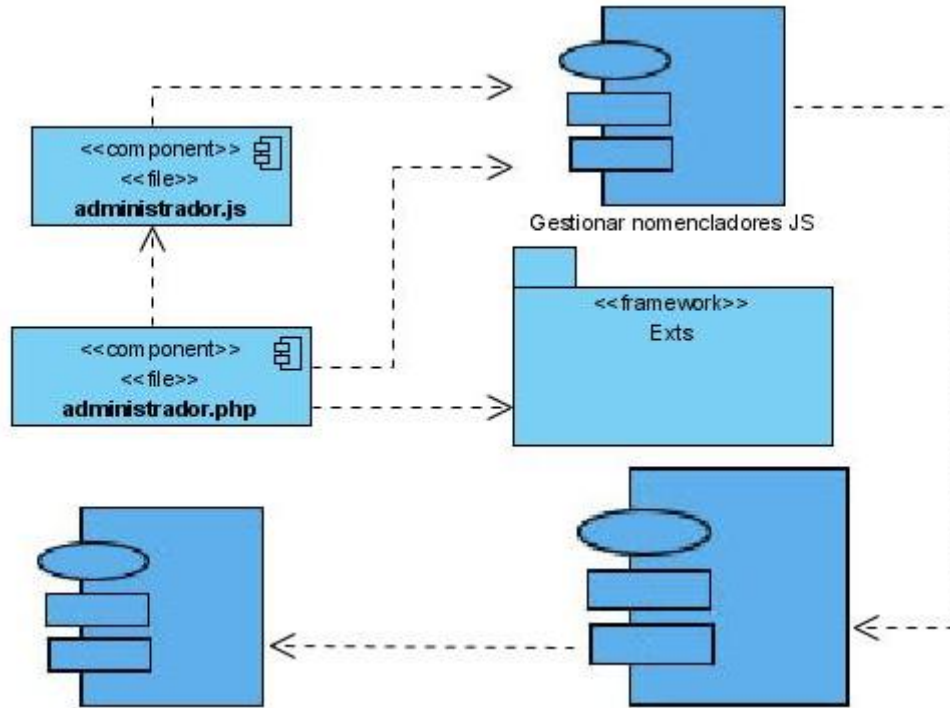


Figura 29 Diagrama de componentes. Gestionar Nomencladores.

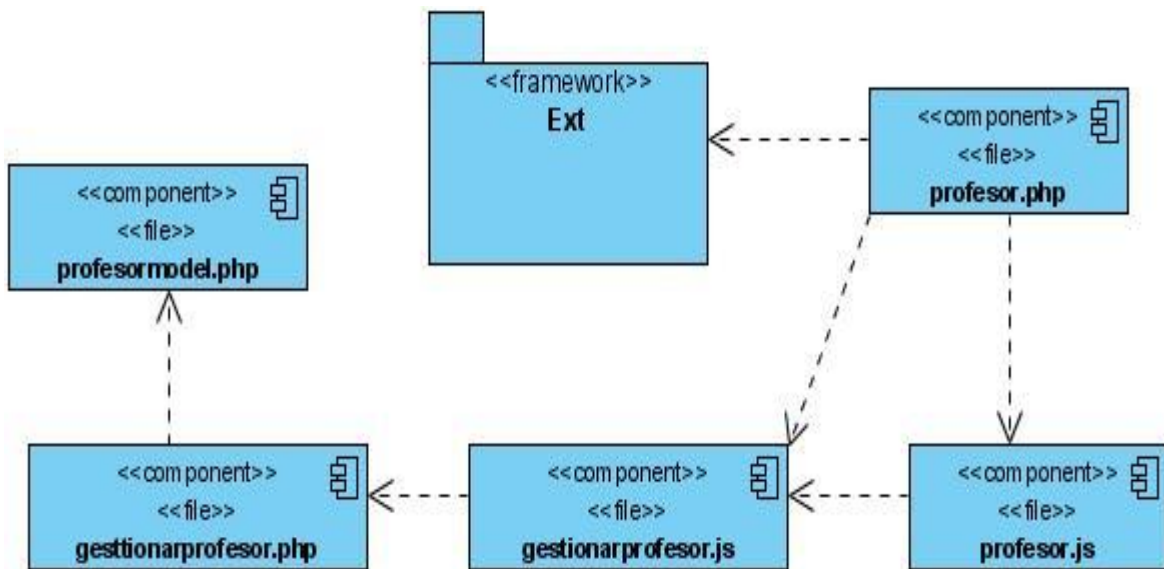


Figura 30 Diagrama de componentes. Gestionar Datos Personales.

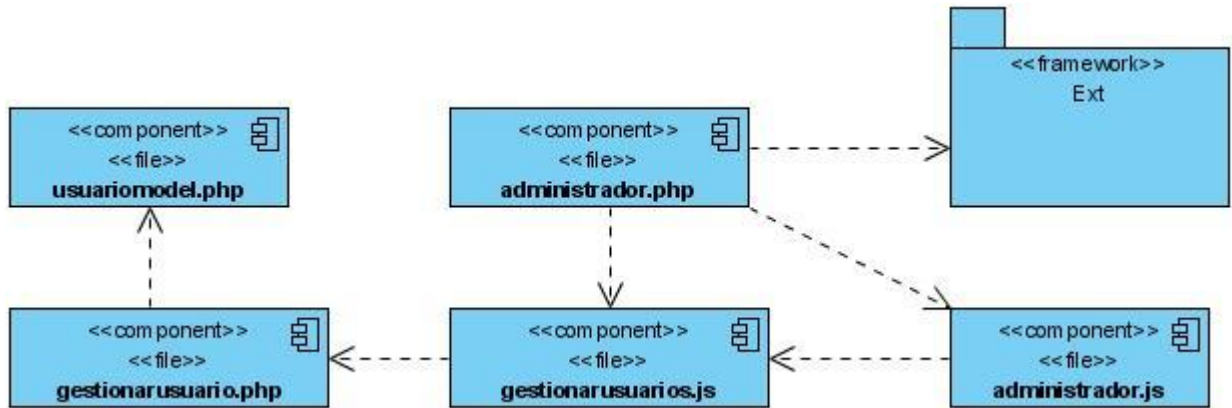


Figura 31 Diagrama de componentes. Gestionar Usuarios.

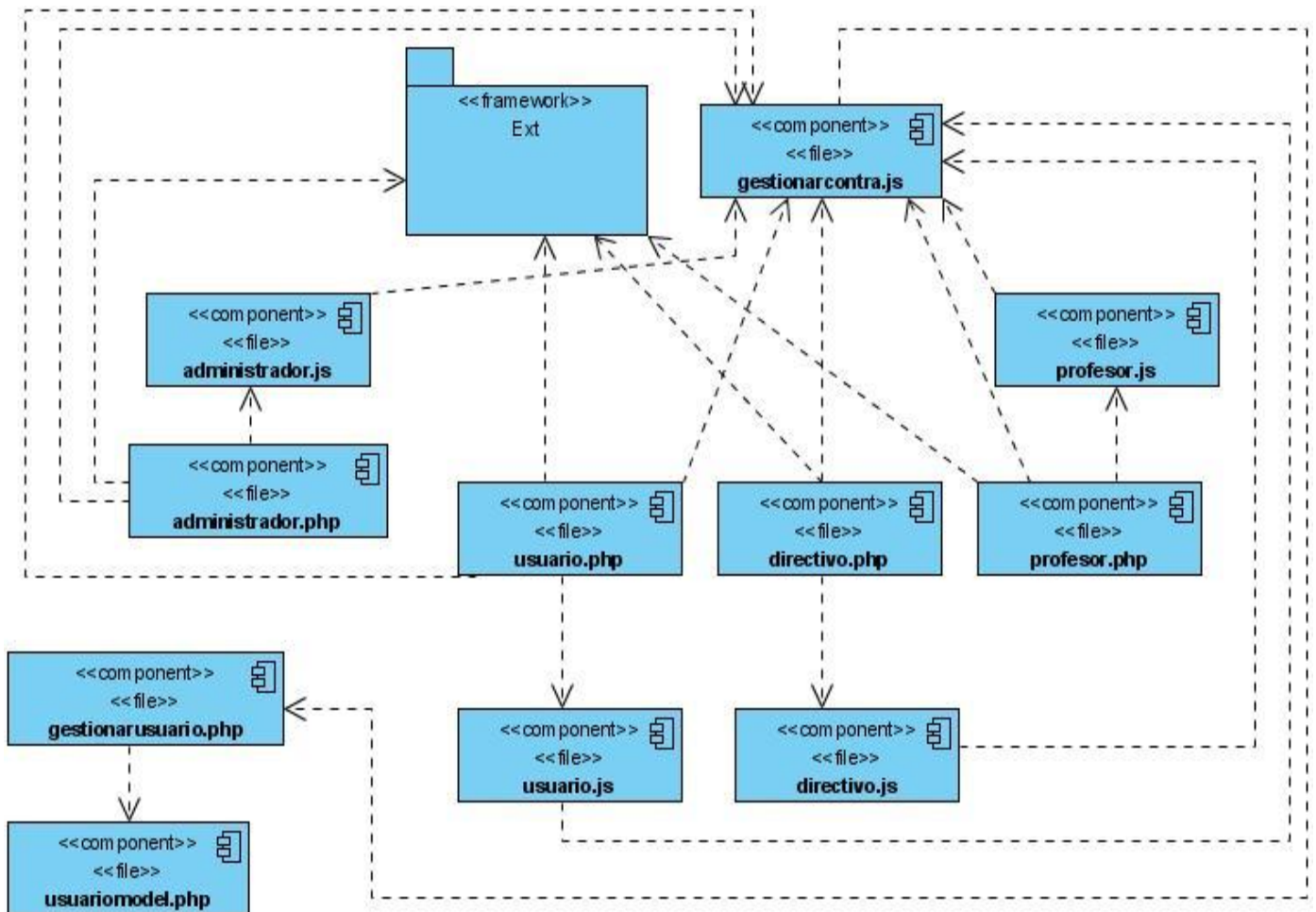


Figura 32 Diagrama de componentes. Gestionar contraseña.

Conclusiones

En este capítulo se ha plasmado mediante el diagrama de despliegue los nodos que se necesitan para la utilización del sistema, lo que se traduce en una PC servidora para emplazar la aplicación, una PC servidora de base de datos, para ubicar la base de datos del sistema, las PC clientes desde donde el usuario podrá acceder al mismo desde cualquier ubicación dentro de la Universidad, y una impresora con la cuál los usuarios podrán imprimir el currículum, así como los directivos imprimir el resultado de los reportes realizados. Además, se puso en evidencia la relación de dependencia que existe entre los diferentes componentes con los que cuenta la aplicación, mediante los diagramas de componentes. Este capítulo es de gran importancia, porque como resultado se obtiene el modelo de implementación, el cual es fundamental para implementar el sistema.

Conclusiones generales.

Debido a la importancia que tiene la información curricular de los profesores de la UCI, surge en la facultad 1 la necesidad de un sistema que sea capaz de gestionar este tipo de información. Para el desarrollo del mismo, se hizo un análisis de la situación existente en el mundo y en Cuba de los sistemas que llevan a cabo estas acciones, llegando a la conclusión que los que tratan el tema, no resuelven el problema en cuestión. Por lo tanto fue necesario desarrollar un sistema para gestionar esta información.

Con este objetivo, se identificaron los principales conceptos envueltos en el problema, y además se caracterizaron las herramientas y tecnologías para el desarrollo de software posibles a utilizar. Teniendo en cuenta las tendencias actuales, la necesidad del uso de software libre, y las ventajas de cada una de ellas, se seleccionó PHP como lenguaje de programación, CodeIgniter como framework de aplicación web, Ext JS como framework de presentación, PostgreSQL como sistema gestor de base de datos, y Apache como servidor web. Como metodología para el desarrollo de software se seleccionó RUP, utilizando UML como lenguaje de modelado.

Se realizó el análisis y diseño de la propuesta del sistema, lográndose identificar las funcionalidades necesarias, y la forma en que estas debían ser implementadas. Una vez hecho esto, se desarrolló el sistema automatizado, logrando abarcar en él todas las actividades que contribuyen a la correcta gestión de la información curricular de los profesores de la facultad 1, obteniéndose de esta forma una herramienta que facilita, viabiliza y apoya este proceso de forma eficiente.

La utilización de este sistema traerá como resultado que el profesor no realice ningún esfuerzo al crear su currículo, modificarlo, o entregarlo. Además, se contará con un dispositivo de almacenamiento para guardar toda esta información, que podrá ser consultado en cualquier momento para la toma de alguna decisión que requiera de la misma, logrando tener una hoja de vida de uno de los recursos humanos más importantes de la Universidad, el profesor.

Recomendaciones

- Integrar los frameworks utilizados con otros frameworks que hagan más fácil el trabajo.
- Incluir la gestión de los roles por funcionalidades.
- Extender el uso de la aplicación a nivel de Universidad.
- Gestionar autenticación a nivel de dominio.
- Exportar el currículum en un documento en formato pdf.
- Realizar pruebas que permitan evaluar el diseño y la implementación de la solución.

Bibliografía referenciada

Adobe y PDF. Adobe y PDF. *Adobe y PDF.* [Online] [Cited: Enero 12, 2009.] <http://www.adobe.com/es/products/acrobat/adobepdf.html>.

Alvarez, Miguel Angel. desarrolloweb.com. *Que es Java.* [Online] [Cited: Febrero 22, 2009.] <http://www.desarrolloweb.com/articulos/497.php>.

Álvarez, Rubén. desarrolloweb.com. *desarrolloweb.com.* [Online] [Cited: Enero 8, 2009.] <http://www.desarrolloweb.com/articulos/243.php>.

Angulo, Marcial N. 2005. *Información: Una nueva propuesta conceptual.* México : Ciencias Informáticas, 2005.

Apache. Apache. *HTTP Server Project.* [Online] [Cited: Marzo 1, 2009.] <http://httpd.apache.org/>.

Caceres, Ana Mercedes. 2006. [Online] 07 21, 2006. [Cited: 03 05, 2009.] Facultad de ingeniería, Universidad de Don Bosco. <http://www.galeon.com/rcruz0423/docs/case.pdf>.

Ceballos Sierra, Francisco Javier. 2003. *Enciclopedia Del Lenguaje C++.* s.l. : Ra-Ma, 2003.

Clase Teórico Práctica # 1 Ingeniería de Software I. **Clase Teórico Práctica # 1 Ingeniería de Software I. 2008.** Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2008.

Company, Zend The PHP. [Online] [Cited: Marzo 3, 2009.] <http://www.zend.com/products/studio/>.

Conallen, Jim. 1999. *Building Web Applications with UML.* 1999.

Conferencia 4 Ingeniería de Software I. **Conferencia 4 Ingeniería de Software I. 2008.** Ciudad de la Habana : s.n., 2008. p. 27.

CRANE, DAVE, PASCARELLO, ERIC and JAMES, DARREN. 2005. *Ajax in action.* 2005.

Creating Web 2.0-Enabled Communities With Sun Java System Portal Server. **Mueller, Tom.** Sun MicroSystem.

- de la Torre, Anníbal.** Lenguajes del lado servidor o cliente. *Lenguajes del lado servidor o cliente*. [Online] [Cited: Enero 6, 2009.] http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
- Definicion.de.** Definicion.de. *Definición de currículum*. [Online] [Cited: Febrero 20, 2009.] <http://definicion.de/curriculum/>.
- desarrolloweb.com.** desarrolloweb.com. *desarrolloweb.com*. [Online] [Cited: Enero 23, 2009.] <http://www.desarrolloweb.com/articulos/840.php>.
- eAprende.com.** eAprende.com. *eAprende.com*. [Online] [Cited: Enero 25, 2009.] <http://www.eaprende.com/gestor-de-basededatos-mysql-postresql-sqlite.html>.
- García Mato, Lic. Rosa María. 1999.** *Diseño de bases de datos*. 1999.
- García Mosquera, Julio Cesar and Pérez La Rosa, Yunesti. 2007.** *Sistema de Control para la Gestión Legal de Software Educativo*. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2007. p. 100.
- GENEURA.** Geneura. *CURSO: Programación Básica en Lenguaje Perl*. [Online] [Cited: Enero 12, 2009.] <http://geneura.ugr.es/CUR/PERL/>.
- . GENEURA. [Online] [Cited: Enero 23, 2009.] <http://geneura.ugr.es/~jmerelo/xml/>.
- GNU Operating System.** GNU Operating System. [Online] [Cited: Febrero 25, 2009.] <http://www.gnu.org/licenses/licenses.es.html>.
- . La definición de software libre. [Online] [Cited: Febrero 26, 2009.] <http://www.gnu.org/philosophy/free-sw.es.html>.
- González Pérez, Dra. Miriam, et al.** *Currículum y Formación Profesional*. Ciudad de la Habana : s.n. Centro de Estudios para el Perfeccionamiento de la Educación Superior. Universidad de la Habana.
- Gosling, James, et al. 1999.** *The Java language specification*. s.l. : Addison-Wesley, 1999.

-
- Grupo Ingeniería del Software y Sistemas de Información (ISSI).** 2003. *Metodologías Ágiles en el Desarrollo de Software*. Alicante : s.n., 2003.
- Hollingsworth, David.** The Workflow Reference Model. Workflow Management Coalition. [Online] <http://www.wfmc.org/standards/docs/tc003v11.pdf>..
- HTML.net.** HTML.net. *Tutoriales sobre HTML y CSS*. [Online] [Cited: Febrero 21, 2009.] <http://es.html.net/tutorials/html/lesson2.asp>.
- IBM.** IBM. *IBM*. [Online] [Cited: Febrero 15, 2009.] <http://www.ibm.com/developerworks/opensource/library/os-eclipse.html>.
- Informática Hoy.** Informática Hoy. *Qué es Código Abierto*. [Online] [Cited: Febrero 20, 2009.] <http://www.informatica-hoy.com.ar/software-libre-gnu/Que-es-Codigo-Abierto.php>.
- Jacobson, Ivar and Booch, Grady y Rumbaugh, James.** 2004. *El Proceso Unificado de Desarrollo del Software*. Ciudad de la Habana : Félix Varela, 2004.
- Java en castellano.** Java en castellano. *Java en castellano*. [Online] [Cited: Enero 6, 2009.] <http://www.programacion.com/java/>.
- JSON.** Introducción a JSON. *Introducción a JSON*. [Online] [Cited: Febrero 25, 2009.] <http://www.json.org/json-es.html>.
- Larman, Craig.** *UML y Patrones*.
- Letelier Torres, Patricio.** *Desarrollo de Software Orientado a Objeto usando UML*. Valencia : Departamento Sistemas Informáticos y Computación.
- Librería Científica Electrónica.** Librería Científica Electrónica. [Online] [Cited: Febrero 10, 2009.] http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S1012-15872003000100006&lng=es&nrm=iso.
- Maestros del web.** Maestros del web. *¿Qué es JavaScript?* [Online] [Cited: Febrero 27, 2009.] www.maestrosdelweb.com/editorial/¿que-es-javascript.

masadelante.com. masadelante.com. *masadelante.com*. [Online] [Cited: Febrero 27, 2009.] <http://www.masadelante.com/faq-plug-in.htm>.

Ministerio de Educación Nacional Republica de Colombia. Ministerio de Educación Nacional Republica de Colombia. *CVN - Centro Virtual de Noticias*. [Online] [Cited: Febrero 21, 2009.] <http://www.mineducacion.gov.co/cvn/1665/article-135782.html>.

Moreno Rodríguez, Carlos. Derecho de Internet. *Las esferas socioculturales del Software Libre*. [Online] [Cited: Febrero 15, 2009.] <http://www.derecho-internet.org/node/293>.

PostgreSQL. PostgreSQL. *Kit de Prensa de PostgreSQL 8.3*. [Online] [Cited: Febrero 27, 2009.] <http://www.postgresql.org/about/press/presskit83.html.es>.

PostgreSQL-es.org. *Portal en español sobre PostgreSQL*. [Online] [Cited: Febrero 27, 2009.] http://www.postgresql-es.org/sobre_postgresql.

R. de Soto, Adolfo and Cuervo Fernández, Eva. 2006. *Nuevas Tendencias en Sistemas de Información: Procesos y Servicios*. 2006. pp. 129-158.

Ramos, Germán, Regourd, Jorge and Rivademar, Juan J. *Tesis de graduación de la Licenciatura en Informática*. Facultad de Ciencias Exactas, Universidad Nacional de La Plata. La Plata : s.n. Tesis.

Real Academia de la Lengua Española. 1936. *Diccionario de la Lengua Española*. España : s.n., 1936.

Reyes Miranda, MsC. Daniel and Segredo Pérez, Dra. Alina María. Cuba bvs Biblioteca virtual de salud. *Revista Cubana de Medicina General Integral*. [Online] [Cited: Febrero 21, 2009.]

Schumacher, Robin and Lentz, Arjen. Dispelling the Myths. *Dispelling the Myths*. [Online] [Cited: Enero 20, 2009.] <http://dev.mysql.com/tech-resources/articles/dispelling-the-myths.html>.

Sun MicroSystem. MySQL. *The world's most popular open source database*. [Online] [Cited: Febrero 25, 2009.] <http://www.mysql.com/why-mysql/>.

Sun Microsystem. Sun Microsystem. *Sun Developer Network (SDN)*. [Online] [Cited: Enero 6, 2009.] <http://java.sun.com/features/1998/05/birthday.html>.

Sun MicroSystem. Sun Network Developer. [Online] [Cited: Febrero 23, 2009.] <http://java.sun.com/xml/docs/api>.

Sun MicroSystems. *Developing AJAX Applications*. s.l. : Sun Microsystems.

—. Sun Developer Network (SDN). [Online] [Cited: Febrero 21, 2009.] <http://java.sun.com/docs/books/jvms/>.

—. Sun Developer Networks. [Online] [Cited: Febrero 22, 2009.] <http://java.sun.com/developer/Books/javaserverpages/ProJSP/Chap12.pdf>.

Sun MycroSystems. [Online] <http://www.sun.com/>.

—. Sun Microsystems. *Sun Developer Networks*. [Online] [Cited: Febrero 23, 2009.] <http://developers.sun.com/mobility/midp/ttips/HTTPost/>.

Theodore, Roszak. 1990. *El culto a la información: el folklore de los ordenadores*. México, Grijalbo : s.n., 1990.

Universidad Michoacana de San Nicolás de Hidalgo. Coordinación de Innovación Educativa. [Online] [Cited: Febrero 22, 2009.] <http://dieumsnh.qfb.umich.mx/gesinfo/>.

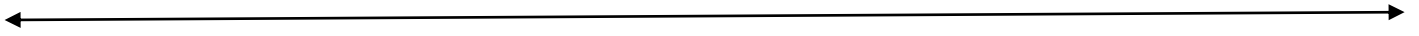
Van Der Henst, Christian. 2005. *Introducción al PHP*. España : s.n., 2005.

WebEstilo. WebEstilo. *Usabilidad, programación y mucho más*. [Online] [Cited: Febrero 25, 2009.] <http://www.webestilo.com/css/>.

Wikipedia La enciclopedia libre. *Wikipedia La enciclopedia libre*. [Online] [Cited: Febrero 10, 2009.] <http://es.wikipedia.org/wiki/Framework>.

Wikipedia La enciclopedia libre. [Online] [Cited: Marzo 1, 2009.] <http://es.wikipedia.org/wiki/AJAX>.

Wikipedia La enciclopedia libre. Wikipedia La enciclopedia libre. *Wikipedia La enciclopedia libre*. [Online] [Cited: Enero 10, 2009.] es.wikipedia.org/wiki/Script.



—. Wikipedia La enciclopedia libre. [Online] [Cited: Enero 10, 2009.] [es.wikipedia.org/wiki/C_\(lenguaje_de_programación\)](http://es.wikipedia.org/wiki/C_(lenguaje_de_programación)).

—. Wikipedia La enciclopedia libre. [Online] [Cited: Febrero 22, 2009.] <http://es.wikipedia.org/wiki/GPL>.

Wikipedia La Enciclopedia Libre. Wikipedia La Enciclopedia Libre. *Wikipedia La Enciclopedia Libre*. [Online] [Cited: Febrero 20, 2009.] [es.wikipedia.org/wiki/Currículo_\(educación\)](http://es.wikipedia.org/wiki/Currículo_(educación)).

Zend The PHP Company. [Online] [Cited: Marzo 3, 2009.] <http://www.zend.com/en/products/studio/features#develop>.

Bibliografía consultada

Acosta Guevara, Teosvany and García Ojito, Yadián. 2008. *Análisis y diseño de los procesos de Apertura y Control de los Despachos de Valores en las Oficinas Postales Telegráficas de Venezuela.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2008. p. 107, Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Angulo, Marcial N. 2005. *Información: Una nueva propuesta conceptual.* México : Ciencias Informáticas, 2005.

CodeIgniter, Guía de Usuario. CodeIgniter. *CodeIgniter.* [Online] [Cited: Febrero 15, 2009.] http://codeigniter.com/user_guide/.

CRANE, DAVE, PASCARELLO, ERIC and JAMES, DARREN. 2005. *Ajax in action.* 2005.

Creating Web 2.0-Enabled Communities With Sun Java System Portal Server. **Mueller, Tom.** Sun Microsystems.

Easyjob. Ayuda CurriculumFacil.

González Pérez, Dra. Miriam, et al. *Currículum y Formación Profesional.* Ciudad de la Habana : s.n. Centro de Estudios para el Perfeccionamiento de la Educación Superior. Universidad de la Habana.

Gosling, James, et al. 1999. *The Java language specification.* s.l. : Addison-Wesley, 1999.

Grupo Ingeniería del Software y Sistemas de Información (ISSI). 2003. *Metodologías Ágiles en el Desarrollo de Software.* Alicante : s.n., 2003

Jacobson, Ivar and Booch, Grady y Rumbaugh, James. 2004. *El Proceso Unificado de Desarrollo del Software.* Ciudad de la Habana : Félix Varela, 2004.

Macías Rodríguez, Juan and Vazquez Rodríguez, Annarelys. 2007. *Estudio de Sistemas de Gestión de Contenidos basados en el lenguaje PHP.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2007. p. 57.

NitroCV. Ayuda NitroCV.

Pressman, R. 2000. *Ingeniería del software: Un enfoque práctico.* McGraw Hill : s.n., 2000.

Pressman, S. 2002. *Ingeniería de Software: Un enfoque Práctico.* Quinta edición. 2002.

Vega Millares, Sissi and Martínez Lozano, Leonardo Andrés. 2008. *Modelación del sitio Web para la gestión de la información en la Agencia Cubana de Noticias.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2008. p. 86, Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Anexos

Anexo1. Descripción ampliada de los casos de uso del sistema.

Caso de Uso	
CU-1	Autenticar
Propósito.	Permitir el acceso a los usuarios.
Actor: Usuario	
Resúmen: El caso de uso se inicia cuando el usuario quiere acceder al sistema, una vez introducidos los datos, el sistema los verifica y le permite o no el acceso, con sus respectivos privilegios. El caso de uso finaliza cuando la operación es realizada con éxito.	
Referencias.	RF 1, RF 1.1, RF 1.2
Precondiciones.	
Poscondiciones.	1. Permitir el acceso al sistema.
Acción del Usuario.	Respuesta del sistema.
1- El usuario introduce el nombre de usuario y la contraseña.	1.1- El sistema toma esos datos y verifica que los mismos sean correctos. 1.2- Si los datos son correctos, el sistema muestra un mensaje de bienvenida en la página principal y finaliza así el caso de uso.
Flujo Alternativo.	
Acción del Actor.	Respuesta del sistema.



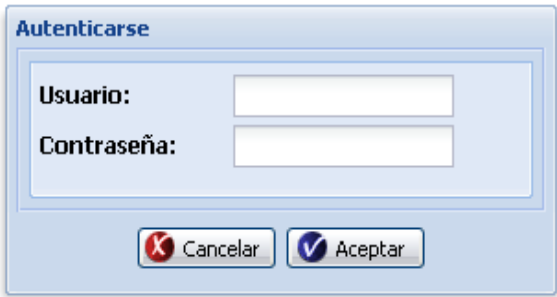
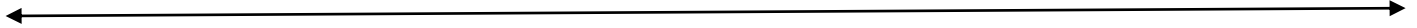
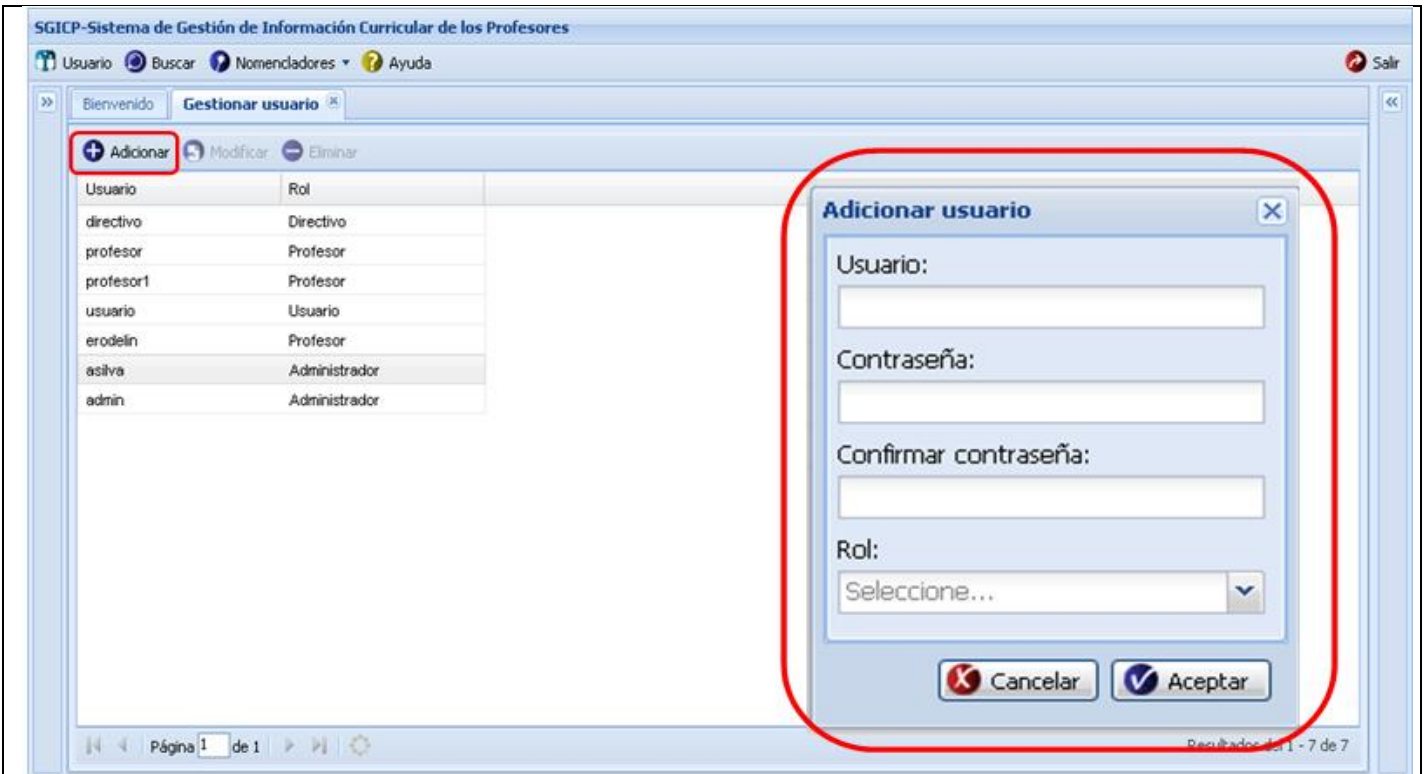
	1.2.1- Si los datos no son correctos el sistema muestra un mensaje de acceso denegado y vuelve al paso 1.
Prototipo de interfaz de usuario.	
	

Tabla 53 Descripción ampliada del caso de uso "Autenticar".

Caso de Uso	
CU-2	Gestionar Usuario.
Propósito.	Adicionar, modificar o eliminar usuarios.
Actor: Profesor	
Resúmen: El caso de uso comienza cuando el Administrador quiere adicionar, modificar o eliminar un usuario y termina cuando la operación que realiza se completa con éxito.	
Referencias.	RF 2, RF 2.1, RF 2.2, RF 2.3
Precondiciones.	<p>1- El usuario debe estar autenticado como Administrador.</p> <p>2- Para modificar o eliminar debe existir al</p>



	menos un usuario.
Poscondiciones.	1- Será creado, modificado o eliminado un usuario.
Escenario “Adicionar usuario”.	
Acción del Usuario.	Respuesta del sistema.
1. El usuario Administrador selecciona adicionar usuario.	2. El sistema muestra un formulario con los datos necesarios para crear un usuario nuevo.
3. El Administrador llena el formulario correspondiente.	4. El sistema comprueba que el usuario no exista ya en la base de datos. Si existe, va al flujo alterno 1. 5. El sistema crea un usuario nuevo, mostrando un mensaje satisfactorio, terminando así el caso de uso.
Flujo Alterno 1.	
Acción del Usuario.	Respuesta del sistema.
	4.1 Si el usuario ya existe, el sistema muestra un mensaje de error, volviendo al paso 3.
Prototipo de interfaz de usuario.	



Escenario “Modificar usuario”.

Acción del Usuario	Respuesta del Sistema
1. Selecciona un usuario de la lista.	2. El sistema habilita la opción de modificar el usuario.
3. Selecciona la opción de modificar el usuario.	4. El sistema muestra los datos anteriores del usuario, brindándole al Administrador la opción de modificarlos.
5. Modifica los datos y acepta.	6. El sistema muestra un mensaje de confirmación. El sistema comprueba que los datos sean correctos, si no son correctos, ir al flujo alterno 1.

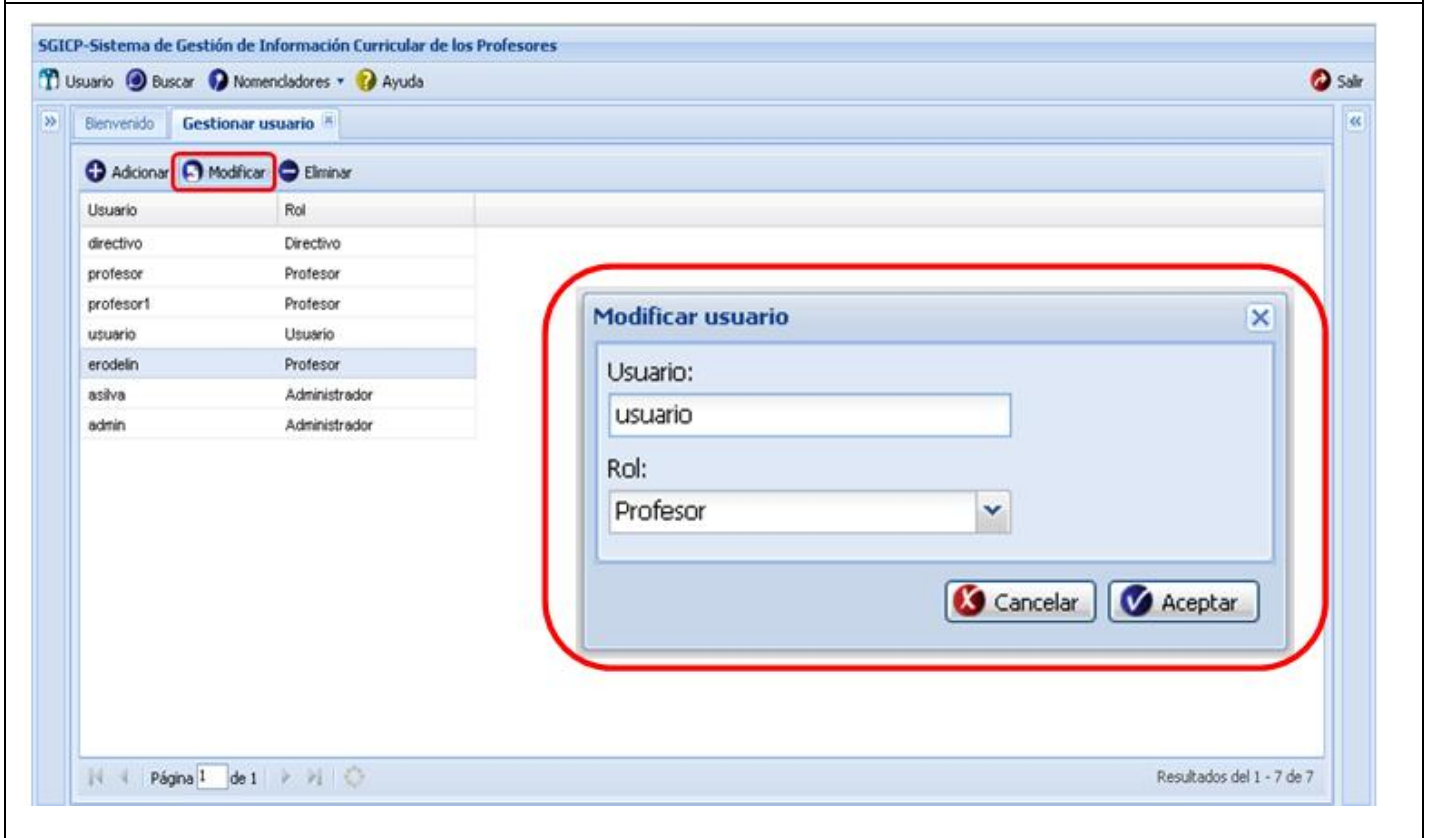


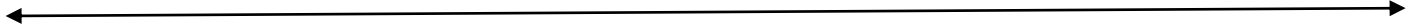
	7- El sistema modifica los datos mostrando un mensaje satisfactorio, terminando así el caso de uso.
--	---

Flujo Alterno 1.

Acción del Actor.	Respuesta del sistema.
	6.1. Si el usuario no confirma la acción, vuelve al paso 1.
	6.2. Si los datos no son correctos, el sistema muestra un mensaje de error volviendo al paso 3.

Prototipo de interfaz de usuario.





Prototipo de interfaz de usuario.	
Escenario “Eliminar usuario”.	
Acción del Usuario	Respuesta del Sistema
1- Selecciona un usuario.	2- El sistema habilita la opción de eliminar el usuario.
3- Selecciona la opción eliminar usuario.	4- El sistema muestra un mensaje de confirmación. Si el Administrador confirma la acción, elimina el usuario y muestra un mensaje satisfactorio, terminando así el caso de uso, si no, ir al flujo alterno 1.
Flujo alterno 1.	
Acción del Usuario	Respuesta del Sistema
	4.1- Si el Administrador no confirma la acción, vuelve al paso 1.
Prototipo de interfaz de usuario.	

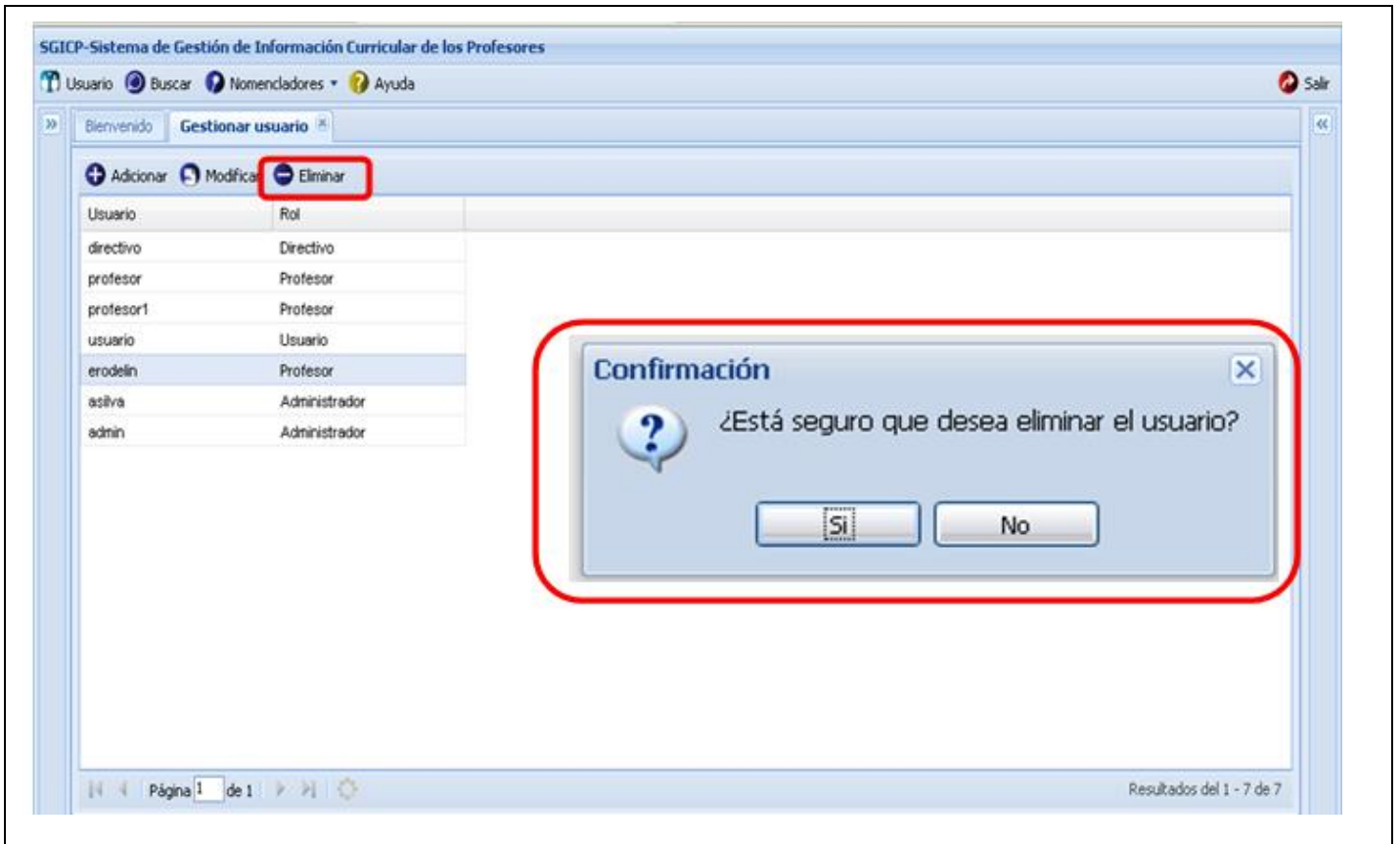


Tabla 54 Descripción ampliada del caso de uso "Gestionar Usuario".

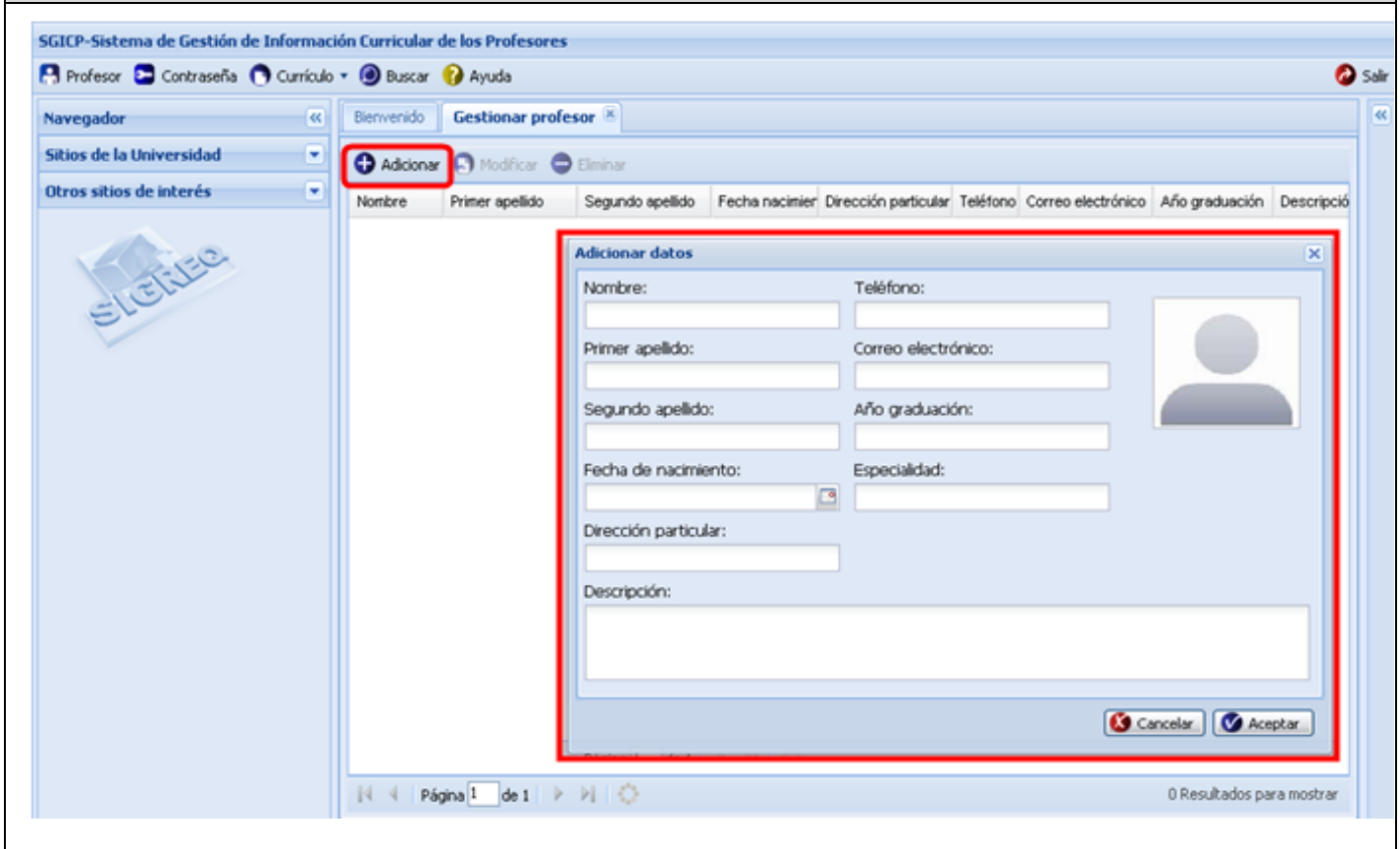
Caso de Uso	
CU-3	Gestionar Datos Personales.
Propósito.	Adicionar, modificar o eliminar datos personales del profesor.
Actor: Profesor, Directivo.	
Resúmen: El caso de uso inicia cuando el profesor o el directivo, desean crear su currículum, y para eso debe llenar una serie de datos personales. Estos datos pueden ser modificados o eliminados posteriormente.	

Referencias.	RF 3, RF 3.1, RF 3.2, RF 3.3, RF 3.4
Precondiciones.	1- El usuario debe estar autenticado como profesor o directivo.
Poscondiciones.	1- Serán guardados los datos personales. 2- Se podrá iniciar la creación del currículum.
Escenario “Adicionar datos personales”.	
Acción del Usuario.	Respuesta del sistema.
1. Selecciona gestionar datos personales.	2. El sistema muestra la opción de adicionar datos personales.
3. Selecciona la opción adicionar datos personales.	4- El sistema muestra un formulario con los datos requeridos.
5 – Introduce los datos personales y acepta.	6-El sistema verifica que los datos sean correctos, y que los datos obligatorios sean llenados, si no es así, ir al flujo alterno 1. 7-El sistema adiciona los datos en la base de datos, mostrando un mensaje de éxito, terminando así el caso de uso.
Flujo Alterno 1.	
Acción del Usuario.	Respuesta del sistema.
	6.1 Si los datos obligatorios no han sido llenados, el sistema muestra un mensaje con el error, solicitándole al usuario llenarlos, volviendo al paso 5.



6.2. Si los datos no son correctos, el sistema muestra un mensaje, volviendo al paso 5.

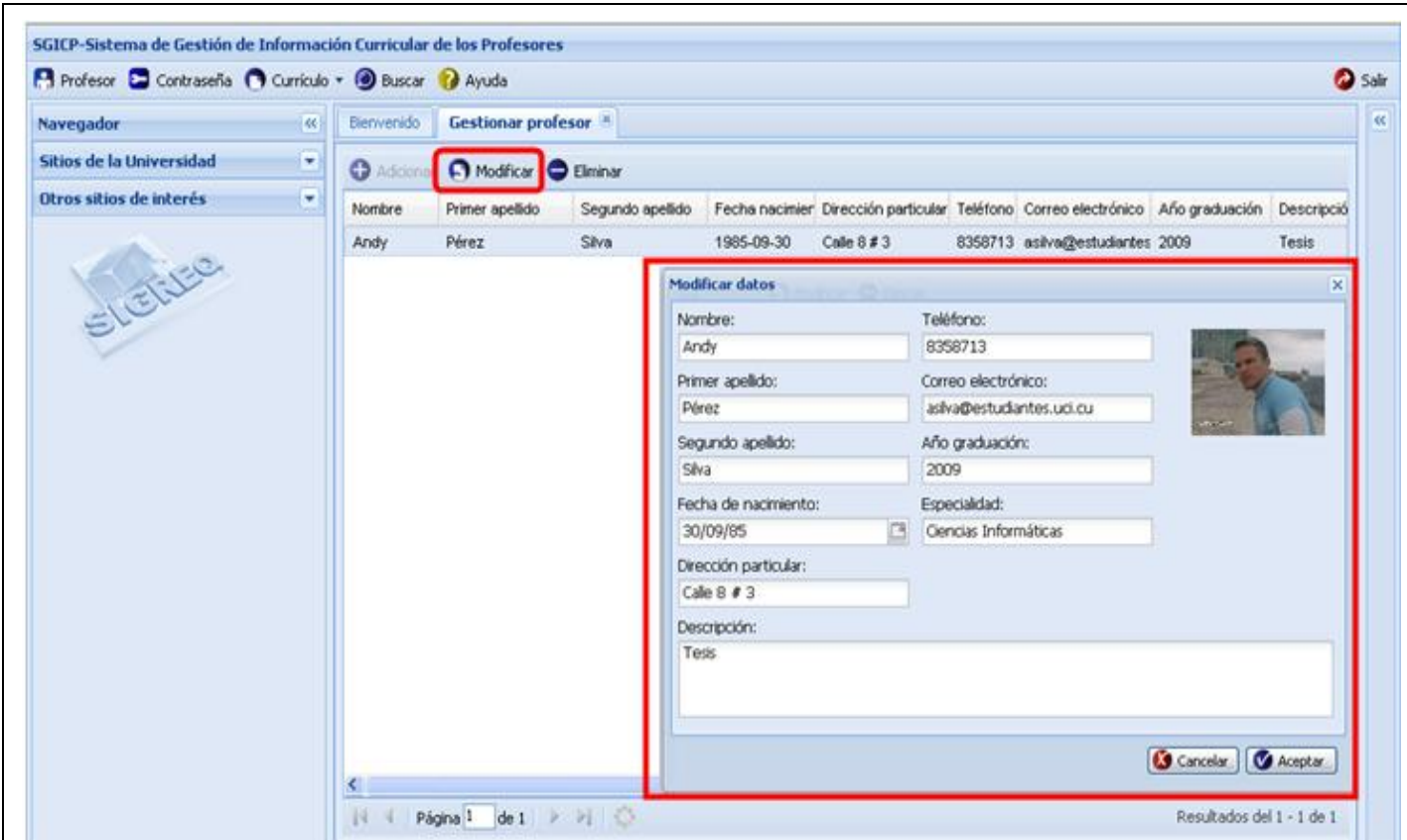
Prototipo de interfaz de usuario.



Escenario “Modificar datos personales”.

Acción del Usuario	Respuesta del Sistema
1. Selecciona gestionar datos personales.	2. El sistema muestra la ventana con los datos.
3. Selecciona los datos personales.	4. El sistema habilita la opción modificar datos personales.

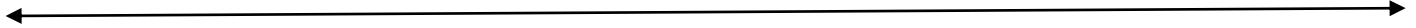
5. Selecciona la opción modificar datos personales.	6. El sistema muestra un formulario con los datos anteriores brindándole al usuario la posibilidad de modificarlos
7. Modifica los datos y acepta.	<p>8. El sistema comprueba que los datos sean correctos y que los datos obligatorios sean llenados, si no es así, ir al flujo alterno 1.</p> <p>9. El sistema modifica los datos, mostrando un mensaje satisfactorio y terminando así el caso de uso.</p>
Flujo Alterno 1.	
Acción del Actor.	Respuesta del sistema.
	<p>8.1 Si los datos obligatorios no han sido llenados, el sistema muestra un mensaje con el error, solicitándole al usuario llenarlos, volviendo al paso 7.</p> <p>8.2. Si los datos no son correctos, el sistema muestra un mensaje, volviendo al paso 7.</p>
Prototipo de interfaz de usuario.	



Escenario “Eliminar datos personales”.

Acción del Usuario	Respuesta del Sistema
1. Selecciona gestionar datos personales.	2. El sistema muestra la ventana con los datos.
3. Selecciona los datos personales.	4. El sistema habilita la opción eliminar datos personales.
5. Selecciona la opción eliminar datos personales.	6. El sistema muestra mensaje de confirmación de la acción, si el usuario acepta, el sistema elimina los datos y muestra un mensaje satisfactorio, terminando así el caso de uso, si no, ir al flujo alterno 1.

Flujo Alterno 1.



Acción del Actor.	Respuesta del sistema.
	6.1 Si el usuario cancela la acción, volver al paso 1.

Prototipo de interfaz de usuario.

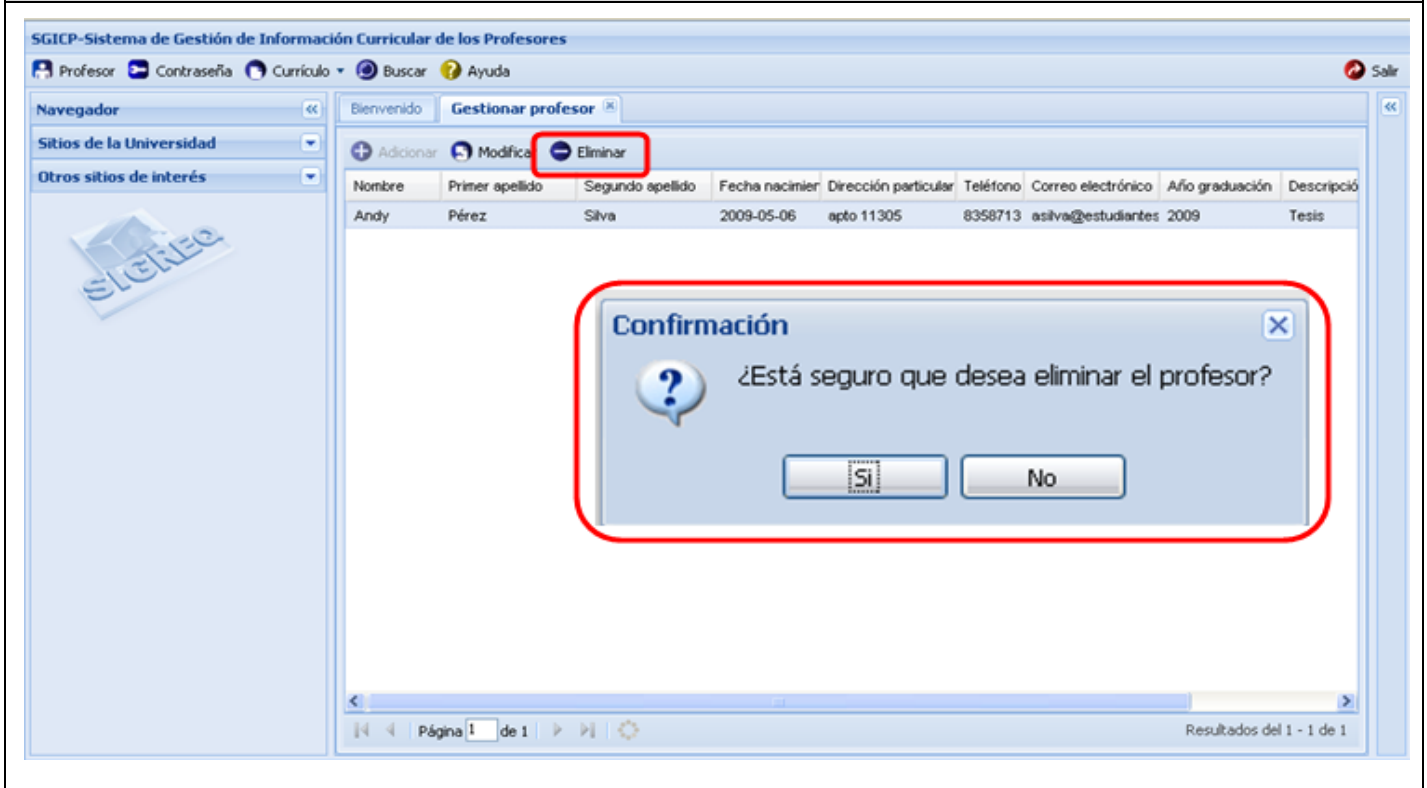
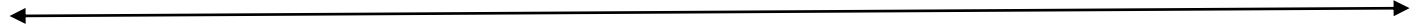


Tabla 55 Descripción ampliada del caso de uso "Gestionar Datos Personales".

Caso de Uso	
CU-4	Gestionar Currículo
Propósito.	Adicionar, modificar y eliminar los datos del currículo.
Actor: Profesor, Directivo.	
Resúmen: El caso de uso inicia cuando el usuario Profesor o Directivo accede al sistema para crear,	

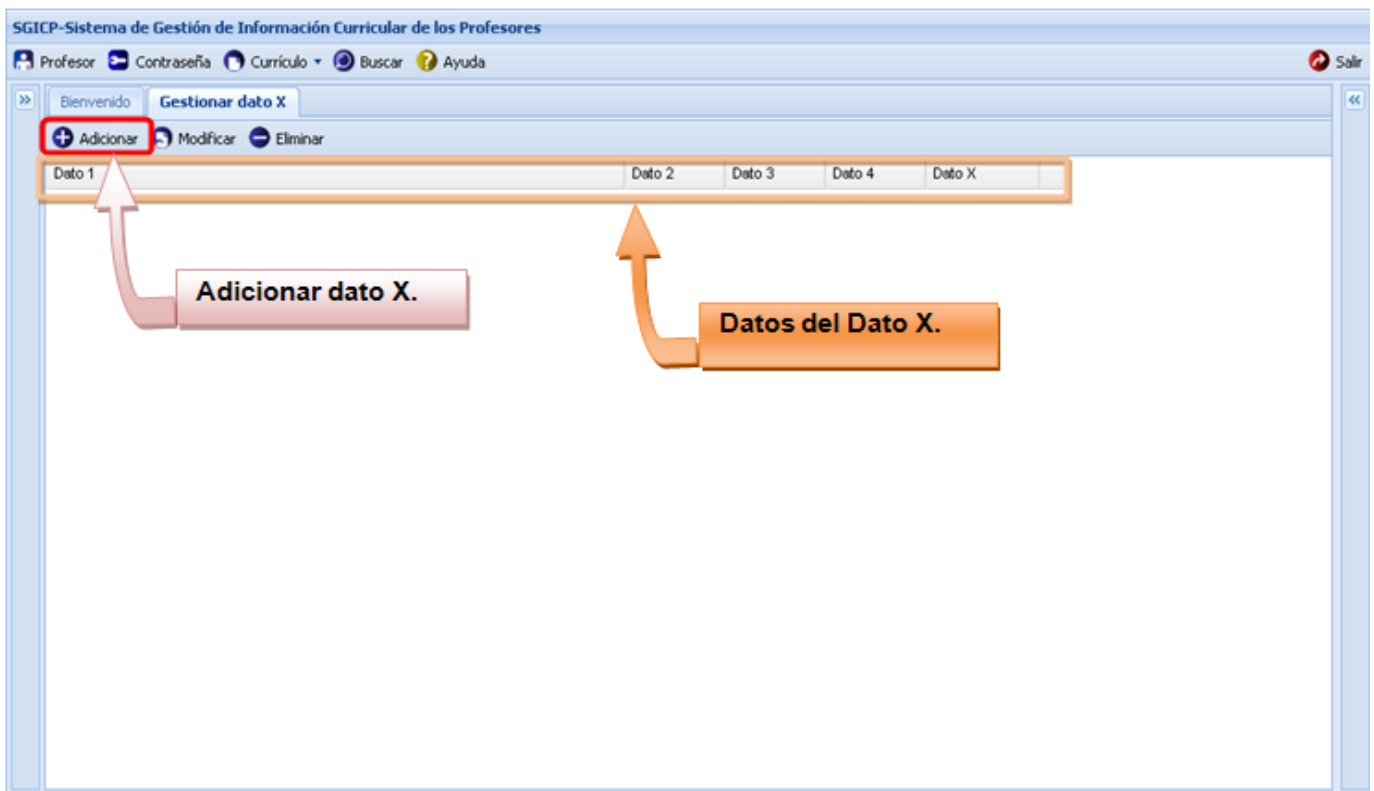


<p>modificar o eliminar datos del currículum. Cuando lleva a cabo la operación u operaciones que desee, termina el caso de uso.</p>	
<p>Referencias.</p>	<p>RF 4, RF 4.1, RF 4.1.1, RF 4.1.2, RF 4.1.3, RF 4.2, RF 4.3, RF 4.4, RF 4.5, RF 4.6, RF 4.7, RF 4.8, RF 4.9, RF 4.10, RF 4.11, RF 4.12, RF 4.13</p>
<p>Precondiciones.</p>	<p>3- El usuario debe estar autenticado como profesor o como directivo.</p> <p>4- Los datos personales del profesor o directivo deben estar guardados.</p>
<p>Poscondiciones.</p>	<p>1- Los datos del currículum serán adicionados, modificados o eliminados.</p>
<p>Escenario “Adicionar Dato X”.</p>	
<p>Acción del Usuario.</p>	<p>Respuesta del sistema.</p>
<p>1. Selecciona adicionar Dato X.</p>	<p>2- El sistema muestra los datos que deben ser llenados, con toda la información concerniente al Dato X.</p>
<p>3. El profesor llena los datos correspondientes y acepta.</p>	<p>4. El sistema comprueba que los datos obligatorios sean llenados, y que los datos introducidos sean correctos, en caso contrario ir a flujo alternativo 1.</p> <p>5. Envía los datos mostrando un mensaje satisfactorio terminando así el caso de uso.</p>
<p>Flujo alternativo 1.</p>	
<p>Acción del Usuario.</p>	<p>Respuesta del sistema.</p>



- 4.1- Si los datos obligatorios no han sido llenados, el sistema muestra un mensaje de error, volviendo al paso 3.
- 4.2- Si los datos introducidos son incorrectos, el sistema muestra un mensaje, volviendo al paso 3.

Prototipo de interfaz de usuario.

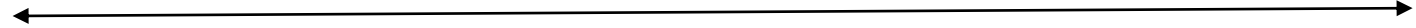




<p>Adicionar categoría científica</p> <p>Categoría científica: <input type="text" value="Seleccione..."/></p> <p>Especialidad: <input type="text"/></p> <p>Lugar de obtención: <input type="text"/></p> <p>Año de obtención: <input type="text"/></p> <p>Cancelar Aceptar</p>	<p>Adicionar categoría docente</p> <p>Categoría docente: <input type="text" value="Seleccione..."/></p> <p>Año en que la recibió: <input type="text"/></p> <p>Cancelar Aceptar</p>	<p>Adicionar misión</p> <p>Nombre de la misión: <input type="text"/></p> <p>Lugar donde cumplió la misión: <input type="text"/></p> <p>Año en que cumplió la misión: <input type="text"/></p> <p>Cancelar Aceptar</p>	<p>Adicionar datos</p> <p>Título de la tesis: <input type="text"/></p> <p>Año que tutoró: <input type="text"/></p> <p>Cancelar Aceptar</p>
<p>Adicionar eventos</p> <p>Nombre de evento: <input type="text"/></p> <p>Nacionalidad evento: <input type="text" value="Seleccione..."/></p> <p>Categoría de participación: <input type="text"/></p> <p>Fecha de evento: <input type="text"/></p> <p>Resultado de evento: <input type="text"/></p> <p>Tipo evento: <input type="text" value="Seleccione..."/></p> <p>Cancelar Aceptar</p>	<p>Adicionar proyecto</p> <p>Nombre del proyecto: <input type="text"/></p> <p>Institución: <input type="text"/></p> <p>Nacionalidad del proyecto: <input type="text" value="Seleccione..."/></p> <p>Cancelar Aceptar</p>	<p>Adicionar investigación</p> <p>Nombre de la investigación: <input type="text"/></p> <p>Resultado: <input type="text"/></p> <p>Estado: <input type="text"/></p> <p>Cancelar Aceptar</p>	
<p>Adicionar curso postgrado recibido</p> <p>Nombre del curso: <input type="text"/></p> <p>Horas clases: <input type="text"/></p> <p>Fecha inicio: <input type="text"/></p> <p>Fecha fin: <input type="text"/></p> <p>Nombre del profesor que lo impartió: <input type="text"/></p> <p>Cancelar Aceptar</p>	<p>Adicionar curso postgrado impartido</p> <p>Nombre del curso: <input type="text"/></p> <p>Horas clases: <input type="text"/></p> <p>Fecha inicio: <input type="text"/></p> <p>Fecha fin: <input type="text"/></p> <p>Cancelar Aceptar</p>	<p>Adicionar curso pregrado impartido</p> <p>Nombre del curso: <input type="text"/></p> <p>Fecha inicio: <input type="text"/></p> <p>Fecha fin: <input type="text"/></p> <p>Imagen del P-1: <input type="text" value="Seleccione la imagen"/></p> <p>Cancelar Aceptar</p>	<p>Adicionar idioma</p> <p>Nombre Idioma: <input type="text" value="Seleccione..."/></p> <p>Escritura: <input type="text"/></p> <p>Audición: <input type="text"/></p> <p>Expresión oral: <input type="text"/></p> <p>Cancelar Aceptar</p>
<p>Adicionar publicación</p> <p>Nombre de la publicación: <input type="text"/></p> <p>Lugar de la publicación: <input type="text"/></p> <p>Fecha de la publicación: <input type="text"/></p> <p>Nacionalidad de la publicación: <input type="text" value="Seleccione..."/></p> <p>Nombre del co-autor: <input type="text"/></p> <p>Primer apellido del co-autor: <input type="text"/></p> <p>Segundo apellido del co-autor: <input type="text"/></p> <p>Cancelar Aceptar</p>	<p>Adicionar datos</p> <p>Nombre del producto: <input type="text"/></p> <p>Fecha de obtención de la patente: <input type="text"/></p> <p>Descripción del producto: <input type="text"/></p> <p>Cancelar Aceptar</p>	<p>Adicionar responsabilidad</p> <p>Responsabilidad: <input type="text"/></p> <p>Fecha inicio: <input type="text"/></p> <p>Fecha fin: <input type="text"/></p> <p>Cancelar Aceptar</p>	



Escenario “Modificar Dato X”	
Acción del Usuario	Respuesta del Sistema
1. Selecciona modificar Dato X.	2. El sistema muestra un formulario con los datos anteriores del Dato X, brindándole al usuario la posibilidad de modificarlos.
3. Modifica los datos correspondientes y acepta.	4- Comprueba que los datos obligatorios sean llenados y que sean correctos, sino ir flujo alterno 1. 5-Modifica los datos, mostrando un mensaje satisfactorio, terminando así el caso de uso.
Flujo alterno 1.	
Acción del Usuario.	Respuesta del sistema.
	4.1- Si los datos obligatorios no han sido llenados, el sistema muestra un mensaje de error, volviendo al paso 3. 4.2- Si los datos son incorrectos, el sistema muestra un mensaje de error, volviendo al paso 3.



The image displays the SGICP-Sistema de Gestión de Información Curricular de los Profesores interface. At the top, there is a navigation bar with options like 'Profesor', 'Contraseña', 'Currículo', 'Buscar', 'Ayuda', and 'Salir'. Below this, a 'Gestionar dato X' section contains 'Adiciona', 'Modificar', and 'Eliminar' buttons. A table below shows columns for 'Dato 1', 'Dato 2', 'Dato 3', 'Dato 4', and 'Dato X'. An orange arrow points from the 'Modificar' button to the 'Dato X' column, labeled 'Modificar datos del Dato X.'. A red arrow points from the 'Dato X' column to a callout box labeled 'Datos del Dato X.'. Below the main interface are seven modal windows for adding different types of data: 'Adicionar categoría científica', 'Adicionar categoría docente', 'Adicionar misión', 'Adicionar datos', 'Adicionar eventos', 'Adicionar proyecto', and 'Adicionar investigación'. Each modal contains specific input fields and 'Cancelar'/'Aceptar' buttons.



Escenario "Eliminar Dato X"

Acción del Usuario.	Respuesta del Sistema
1- Selecciona eliminar Dato X. 3- Acepta la acción.	2. El sistema muestra un mensaje de confirmación de la acción. 4- Si el usuario acepta el sistema elimina el Dato X y muestra un mensaje satisfactorio, terminando así el caso de uso, si no acepta, ir al flujo alterno 1.
<p>Flujo Alterno 1.</p>	
Acción del Actor.	Respuesta del sistema.
	4.1- Si el usuario no acepta la acción, el sistema le brinda la posibilidad de volver al paso 1.

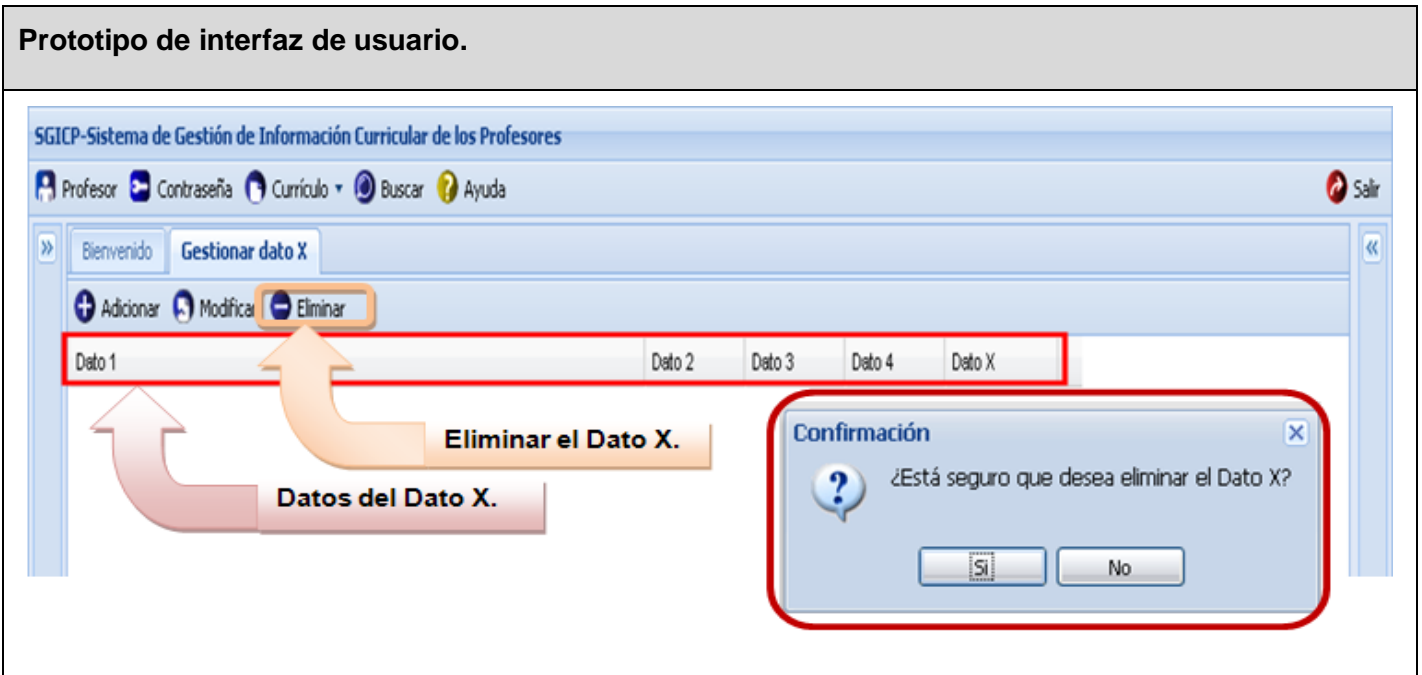


Tabla 56 Descripción ampliada del caso de uso "Gestionar Currículo"

Caso de Uso	
CU-5	Generar Reportes.
Propósito.	Generar listados de los profesores de acuerdo con un criterio.
Actor: Directivo.	
Resúmen: El caso de uso inicia cuando el directivo quiere generar listados de los profesores de acuerdo con diferentes criterios. Se obtiene un listado de acuerdo a un criterio introducido de interés para el directivo, pudiendo este generar el listado a un documento en formato pdf, terminando el caso de uso.	
Referencias.	RF 11, RF 11.1, RF 11.2, RF 11.3, RF 11.4, RF 11.5
Precondiciones.	1- El usuario debe estar autenticado como directivo.
Poscondiciones.	1- Se obtiene el listado de los profesores que cumplan

	con los criterios.
Acción del Usuario.	Respuesta del sistema.
1- Selecciona la opción generar reportes.	2- El sistema muestra los criterios para generar los reportes.
3- Selecciona el criterio para generar el listado.	4- El sistema le solicita al usuario que introduzca los datos.
5- Introduce los datos y acepta.	6- El sistema muestra un listado con los profesores que cumplen con el criterio seleccionado. Si no encuentra ningún profesor, ir al flujo alterno 1. 7- El sistema brinda la opción de generar el listado a un documento.
8- El usuario selecciona la opción generar el documento, de no ser así, ir al flujo alterno 1.	9- El sistema genera el documento, terminando así el caso de uso.
Flujo Alterno 1.	
Acción del Actor.	Respuesta del sistema.
8.1- Si el usuario no desea generar el documento, cancela y puede volver al paso 3.	6.1- El sistema muestra un listado vacío, permitiendo al usuario volver a introducir un criterio distinto, volviendo al paso 3.

Prototipo de interfaz de usuario.

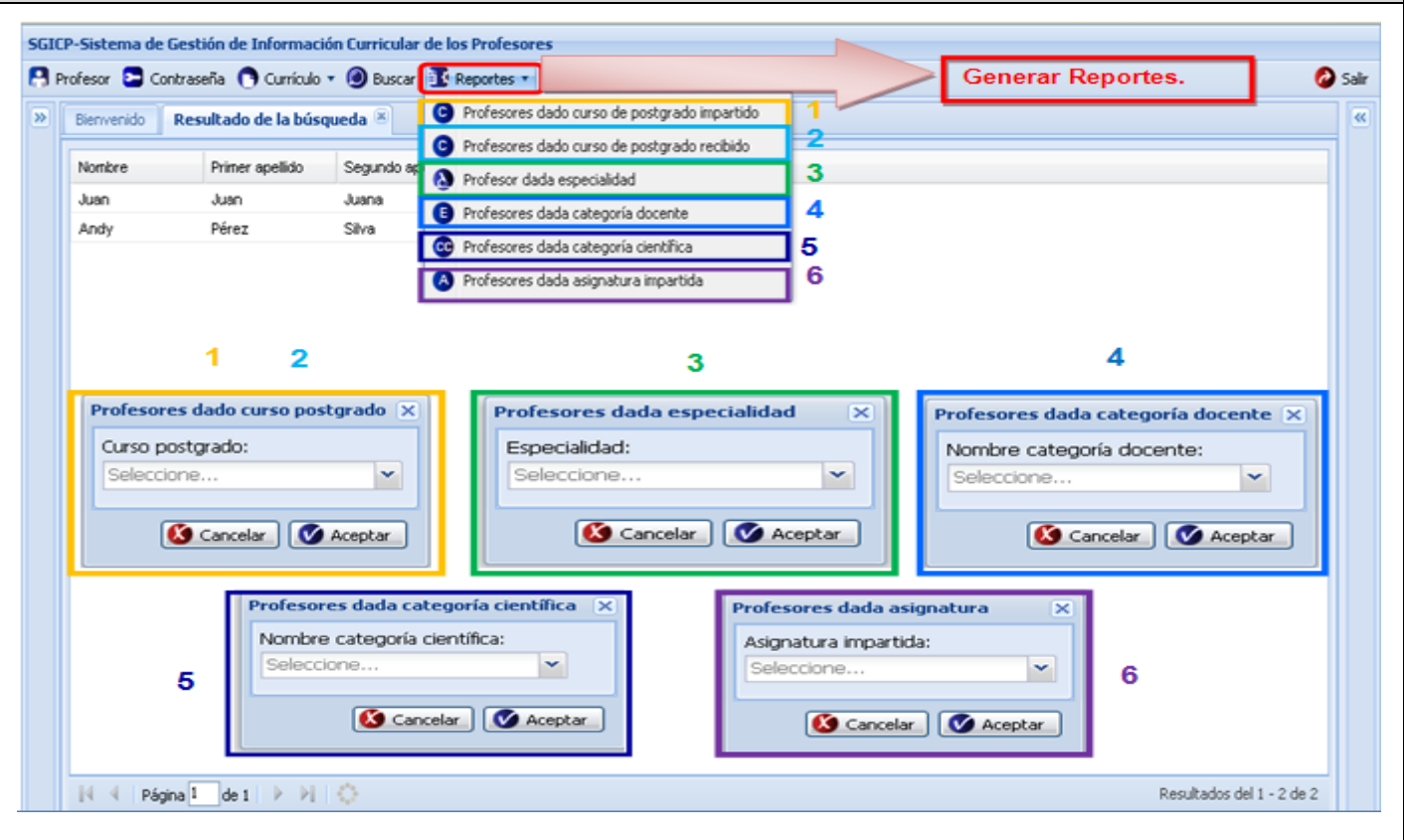


Tabla 57 Descripción ampliada del caso de uso "Generar Reportes"

Caso de Uso	
CU-6	Cambiar Contraseña.
Propósito.	El propósito de este caso de uso es brindarles seguridad a los usuarios, permitiéndoles cambiar su propia contraseña, una vez que sean creadas sus cuentas y accedan al sistema, de manera que solo ellos conozcan su contraseña.
Actor: Usuario, Administrador, Profesor, Directivo.	



Resumen: El caso de uso comienza cuando el usuario solicita cambiar su contraseña. El sistema le permite comprobar que los datos sean correctos, cambia la contraseña, terminando así el caso de uso.	
Referencias.	RF 5
Precondiciones.	1- El usuario debe estar autenticado.
Poscondiciones.	2- El usuario modificará su contraseña.
Acción del Usuario.	Respuesta del sistema.
1- Selecciona la opción cambiar contraseña.	2- El sistema muestra un formulario con el usuario, y los campos con la contraseña anterior, la contraseña nueva y la confirmación.
3- Llena el formulario con los datos requeridos y acepta.	4- El sistema comprueba que los datos obligatorios sean llenados, si no es así, ir al flujo alterno 1. 5- El sistema comprueba que la contraseña anterior coincida, si no es así, ir al flujo alterno 2. 6- El sistema realiza el cambio de contraseña, mostrando un mensaje satisfactorio, terminando así el caso de uso.
Flujo Alterno 1.	
Acción del Actor.	Respuesta del sistema.
	4.1- El sistema muestra un mensaje para que el usuario llene los datos obligatorios, volviendo al paso 3.
Flujo Alterno 2.	
Acción del Usuario.	Respuesta del sistema.



5.1- El sistema muestra un mensaje con el error, volviendo al paso 3.

Prototipo de interfaz de usuario.

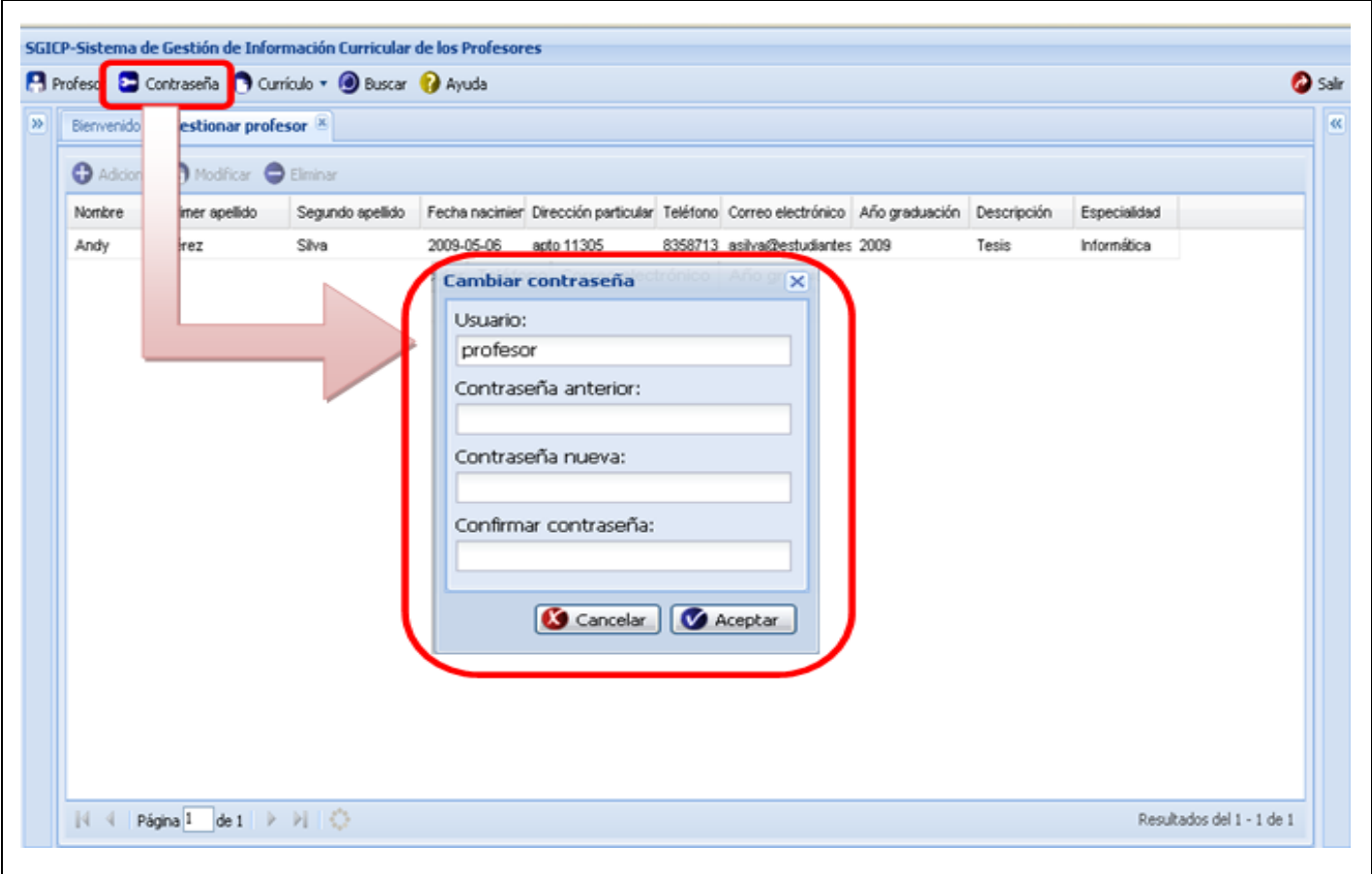



Tabla 58 Descripción ampliada del caso de uso "Cambiar contraseña"

Caso de Uso	
CU-7	Gestionar búsqueda.
Propósito.	Realizar una búsqueda avanzada de los profesores, para poder ver su currículum y datos personales.

Actor: Usuario, Administrador, Profesor, Directivo.	
Resumen: El caso de uso comienza cuando un usuario quiere buscar un profesor determinado. El sistema realiza una búsqueda siguiendo varios criterios de búsqueda. El resultado es un listado de profesores de los cuales se podrá obtener el currículum y los datos personales, y además, generar el documento con el currículum.	
Referencias.	RF 6, RF 6.1, RF 6.2, RF 6.3, RF 6.4, RF 7, RF 8
Precondiciones.	1- El usuario debe estar autenticado.
Poscondiciones.	1- Se obtendrá un resultado de búsqueda, además, la información personal y curricular del profesor, así como un documento con el currículum del mismo.
Acción del Usuario.	Respuesta del sistema.
1- Selecciona la opción realizar búsqueda avanzada.	2- El sistema muestra un formulario con los criterios por los que el usuario puede buscar.
3- Introduce el criterio de búsqueda.	4- El sistema realiza la búsqueda, mostrando el resultado en una página. Si el usuario no introduce ningún criterio, o si no hay ningún resultado de la búsqueda, ir al flujo alternativo 1.
5- Selecciona un profesor.	6- El sistema habilita las opciones de ver ficha del profesor y ver currículum.
7- Selecciona ver ficha del profesor, para ver los datos personales. Si selecciona ver currículum, ir al flujo alternativo 2.	8- El sistema muestra los datos personales del profesor, terminando así el caso de uso.
Flujo Alternativo 1.	



Acción del Actor.	Respuesta del sistema.
	4.1- Si el usuario no introduce ningún criterio de búsqueda, el sistema muestra un listado de todos los profesores existentes.
Flujo Alternativo 2.	
7.1- Selecciona la opción ver currículum.	7.2- El sistema muestra el currículum del profesor, brindándole al usuario la opción de generar el documento currículum en formato pdf.
7.3- Selecciona generar el documento currículum.	7.4- El sistema genera el documento currículum, terminando así el caso de uso.

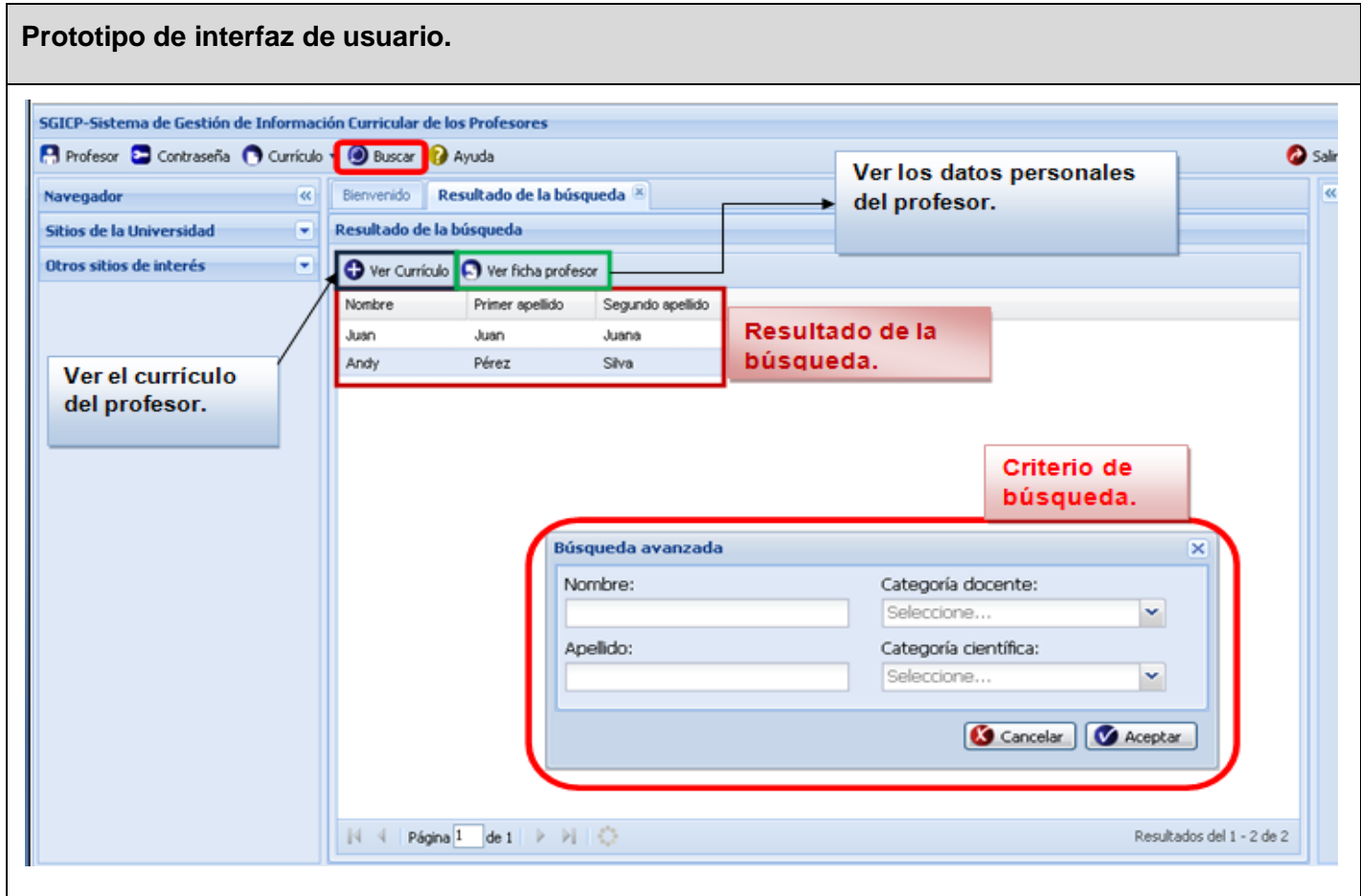
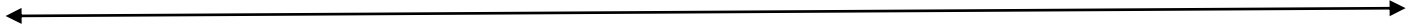


Tabla 59 Descripción ampliada del caso de uso "Gestionar Búsqueda"

Caso de Uso	
CU-8	Gestionar Nomencladores.
Propósito.	Adicionar, modificar o eliminar un nomenclador de la aplicación.
Actor: Administrador.	
Resumen: El caso de uso inicia cuando el usuario Administrador desea adicionar, modificar o eliminar un	

nomenclador. El sistema realiza estas operaciones terminando así el caso de uso.	
Referencias.	RF 9, RF 9.1, RF 9.1.1, RF 9.1.2, RF 9.1.3, RF 9.2, RF 9.3, RF 9.4, RF 10
Precondiciones.	1- El usuario debe estar autenticado como administrador.
Poscondiciones.	1- Se adicionará, modificará o eliminará un nomenclador.
Escenario “Adicionar Nomenclador X”.	
Acción del Usuario.	Respuesta del sistema.
1- Selecciona la opción adicionar nomenclador.	2- El sistema muestra un formulario con los datos requeridos.
3- Llena los datos y acepta.	4- El sistema comprueba que los datos obligatorios sean llenados y que los datos sean correctos, en caso contrario, ir al flujo alterno 1. 5- El sistema adiciona el nomenclador, mostrando un mensaje satisfactorio, terminando así el caso de uso.
Flujo Alterno 1.	
Acción del Actor.	Respuesta del sistema.
	4.1- Si los datos obligatorios no son llenados, el sistema muestra un mensaje de error, volviendo al paso 3. 4.2- Si los datos no son correctos, el sistema muestra un mensaje de error, volviendo al paso 3.
Escenario “Modificar Nomenclador X”.	
Acción del Usuario.	Respuesta del sistema.



1- El usuario selecciona la opción modificar nomenclador.	2- El sistema muestra un formulario con los datos requeridos para modificar el nomenclador.
3- Llena los datos y acepta.	6- El sistema comprueba que los datos obligatorios sean llenados y que los datos sean correctos, en caso contrario, ir al flujo alterno 1. 4- El sistema adiciona el nomenclador, mostrando un mensaje satisfactorio, terminando así el caso de uso.
Flujo Alterno 1.	
Acción del Actor.	Respuesta del sistema.
	4.1- Si los datos obligatorios no son llenados, el sistema muestra un mensaje de error, volviendo al paso 3. 4.2- Si los datos no son correctos, el sistema muestra un mensaje de error, volviendo al paso 3.
Escenario “Nomenclador X”.	
Acción del Usuario.	Respuesta del sistema.
1- El usuario selecciona la opción eliminar nomenclador.	2- El sistema muestra un mensaje de confirmación de la acción. 3- Si el usuario acepta la acción, el sistema elimina el nomenclador, terminando así el caso de uso, si no, ir al flujo alterno 1.
Flujo Alterno 1.	
Acción del Actor.	Respuesta del sistema.
	3.1- Si el usuario no acepta la acción, el sistema cierra el



mensaje, dándole al usuario la posibilidad de ir al paso 1.

Prototipo de interfaz de usuario.

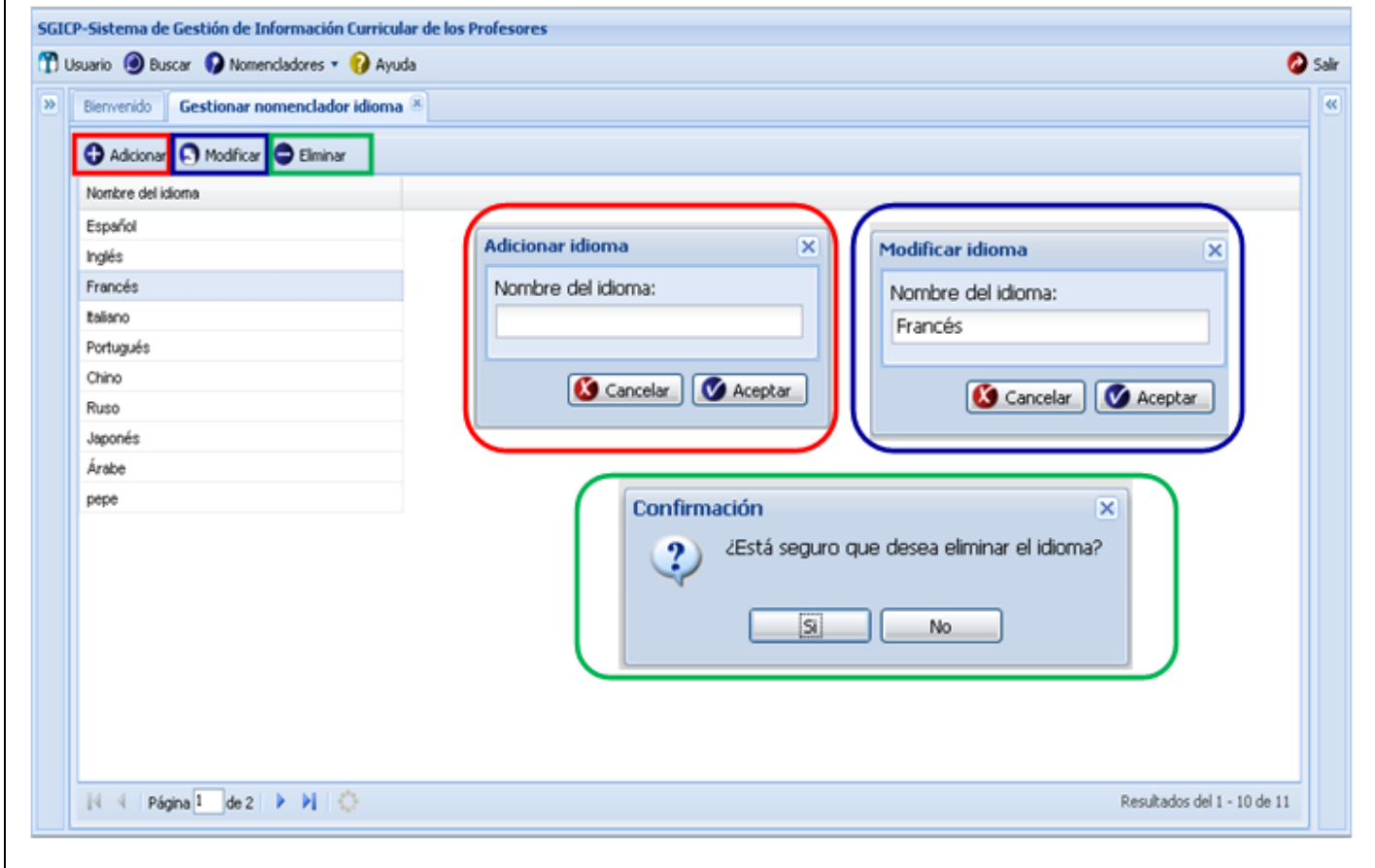


Tabla 60 Descripción ampliada del caso de uso "Gestionar Nomencladores".

Anexo 2. Diagrama de Clases Web.

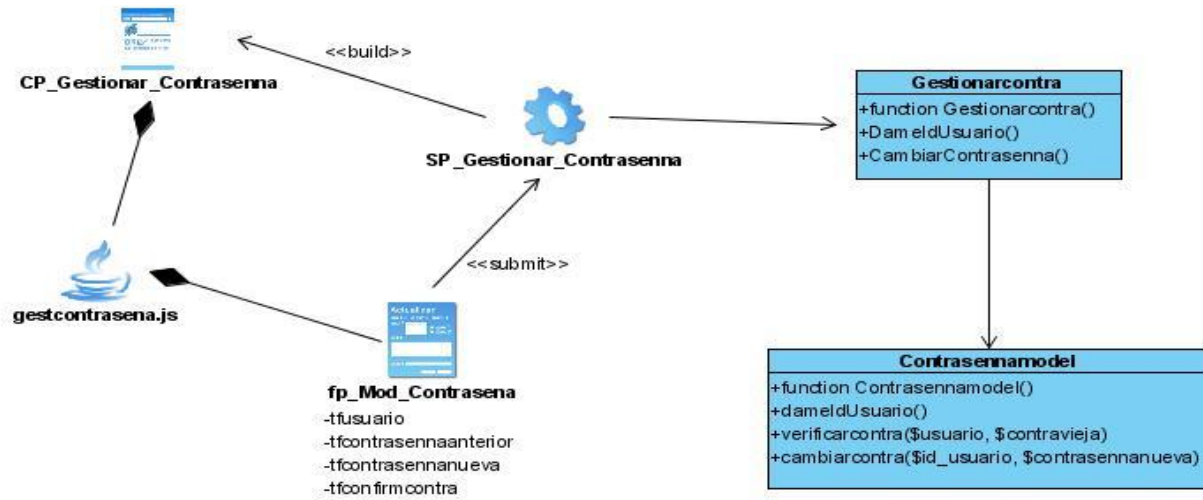


Figura 33 DCW "Cambiar contraseña".

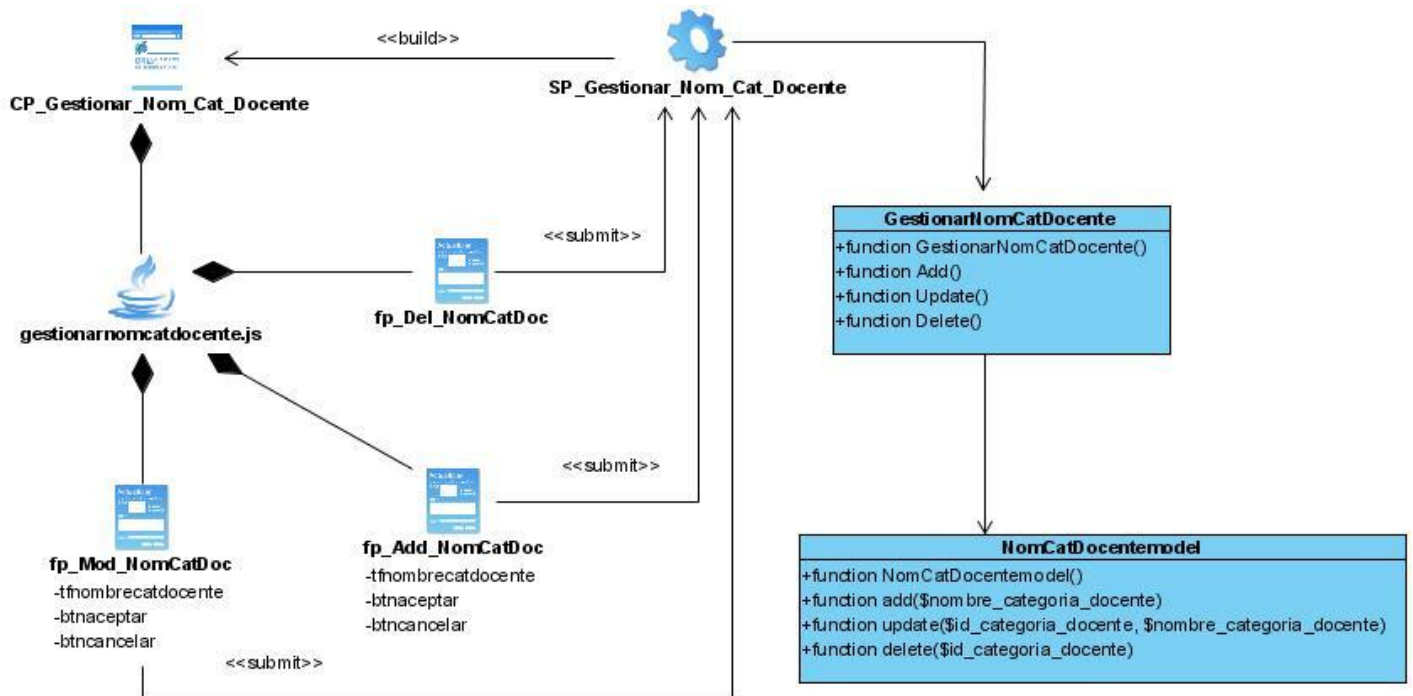


Figura 34 DCW "Gestionar nomenclador categoría docente".

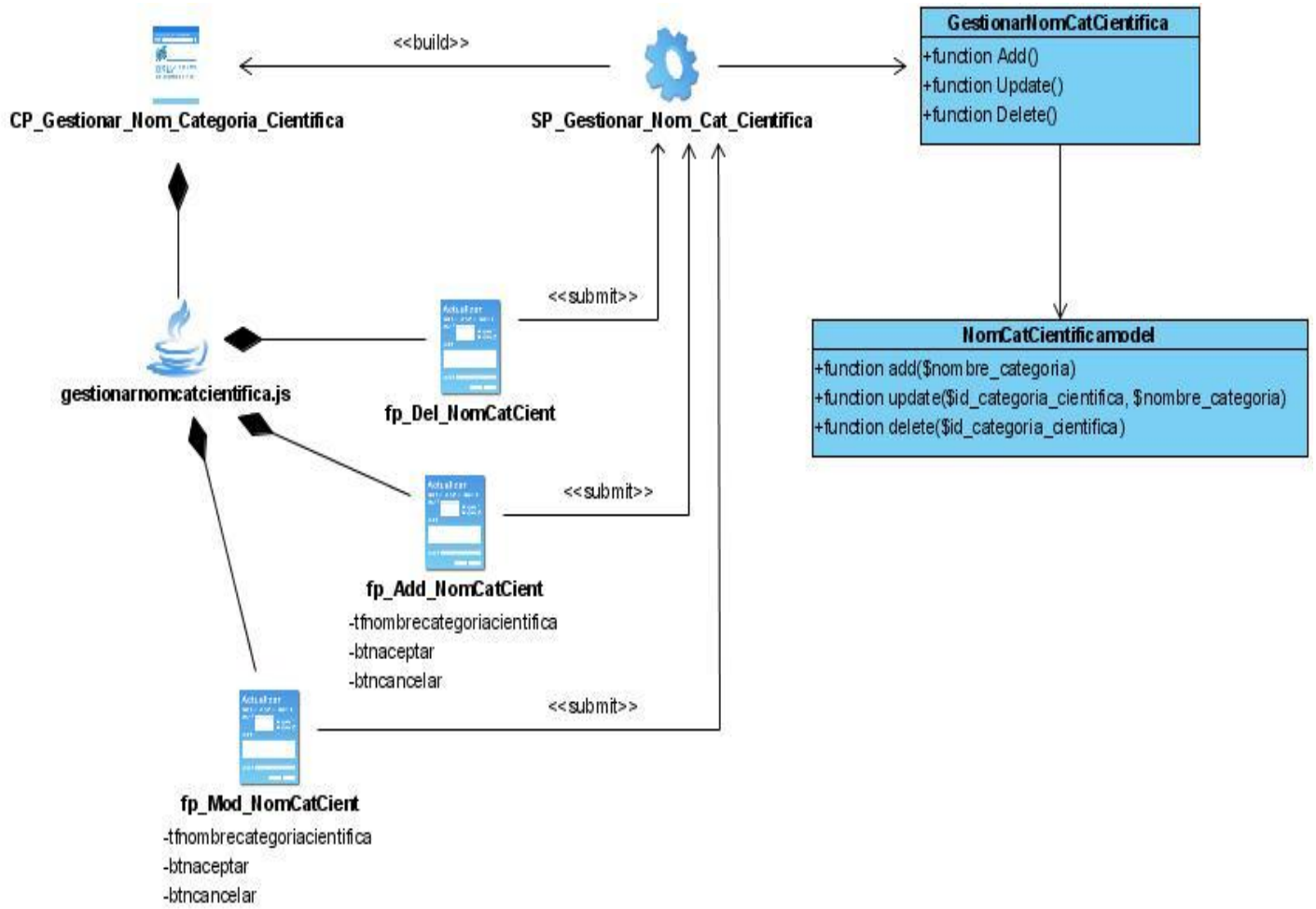


Figura 35 DCW "Gestionar nomenclador categoría científica".

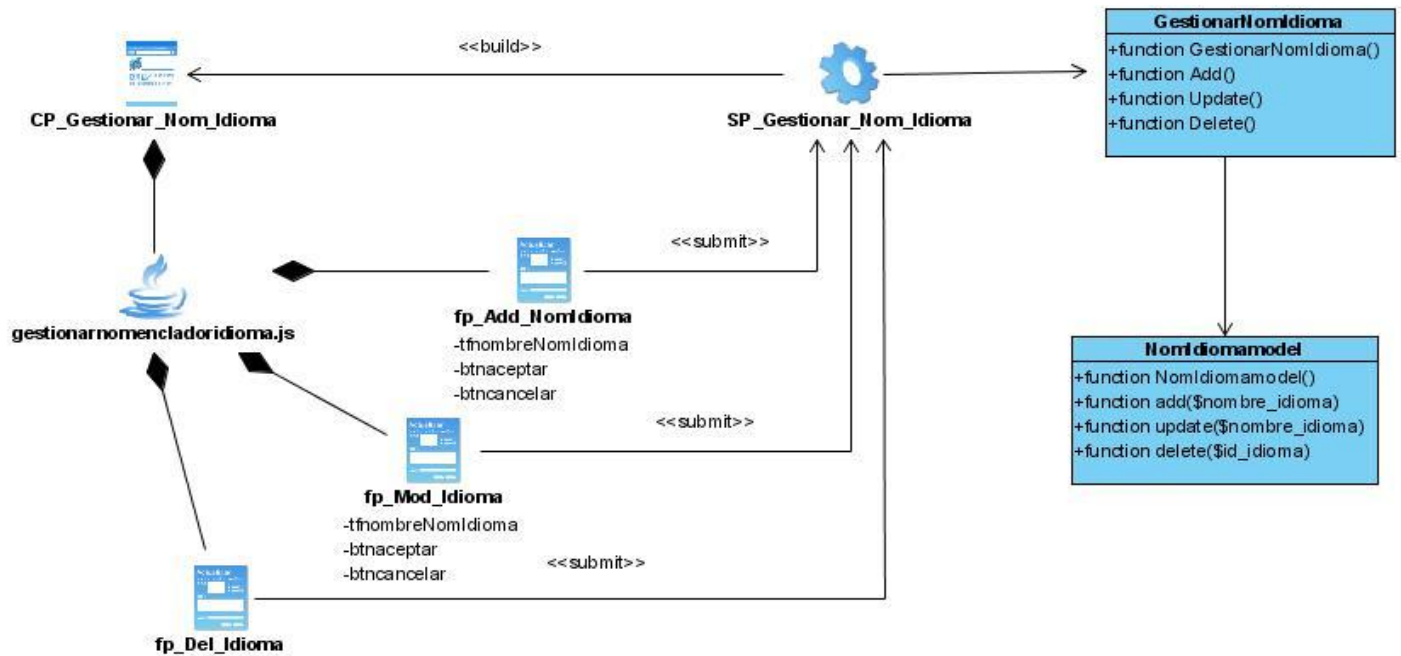


Figura 36 DCW "Gestionar nomenclador idioma".

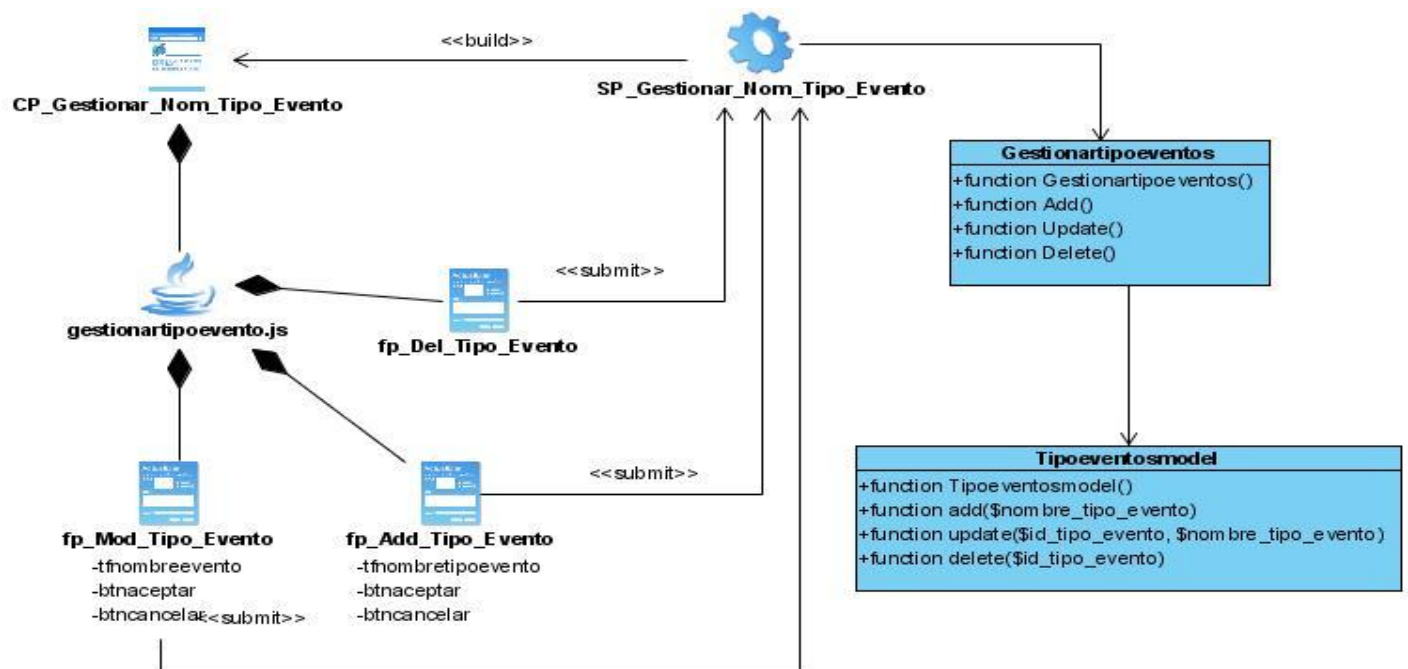


Figura 37 DCW "Gestionar nomenclador tipo de evento".

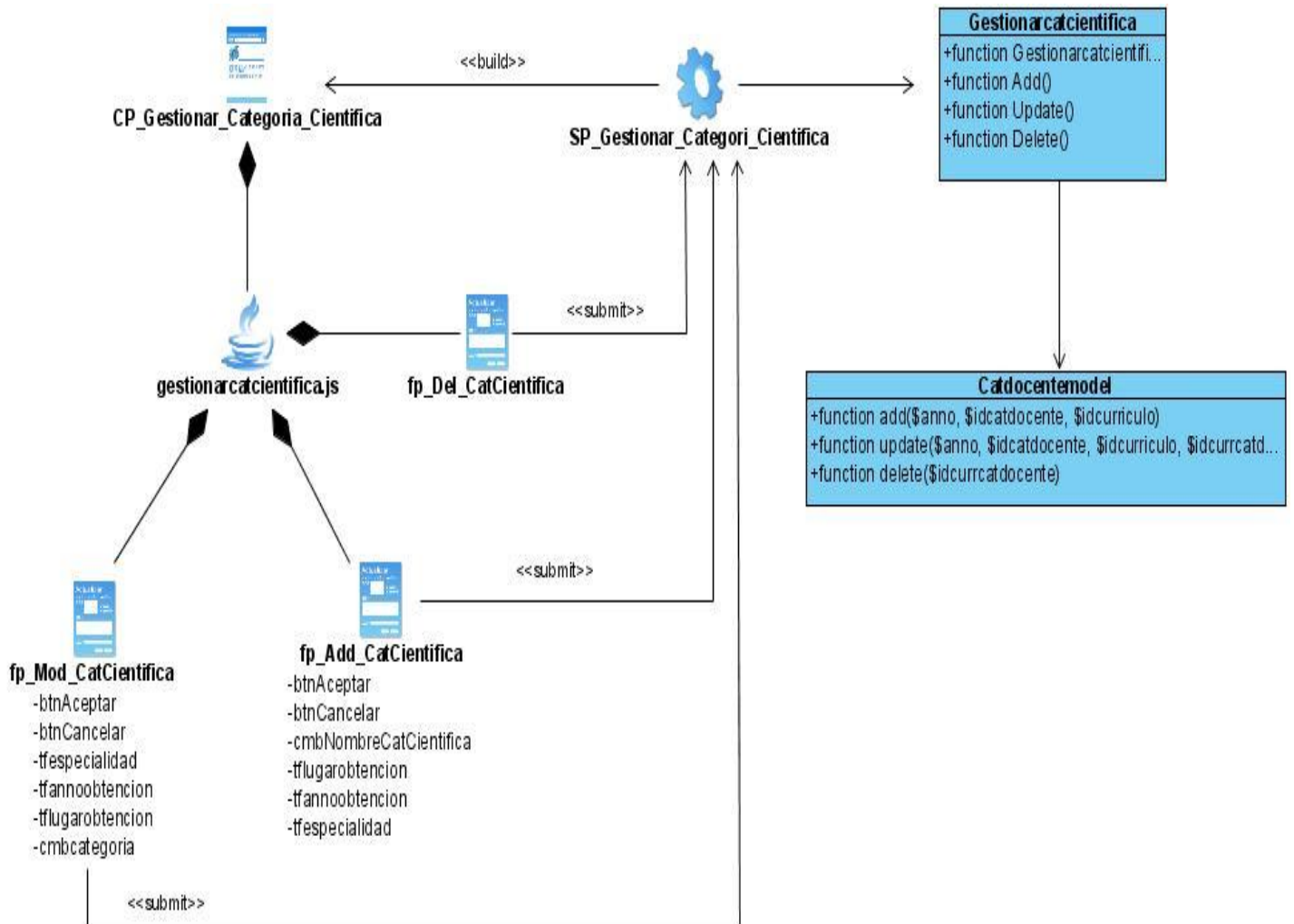


Figura 38 DCW "Gestionar categoría científica".

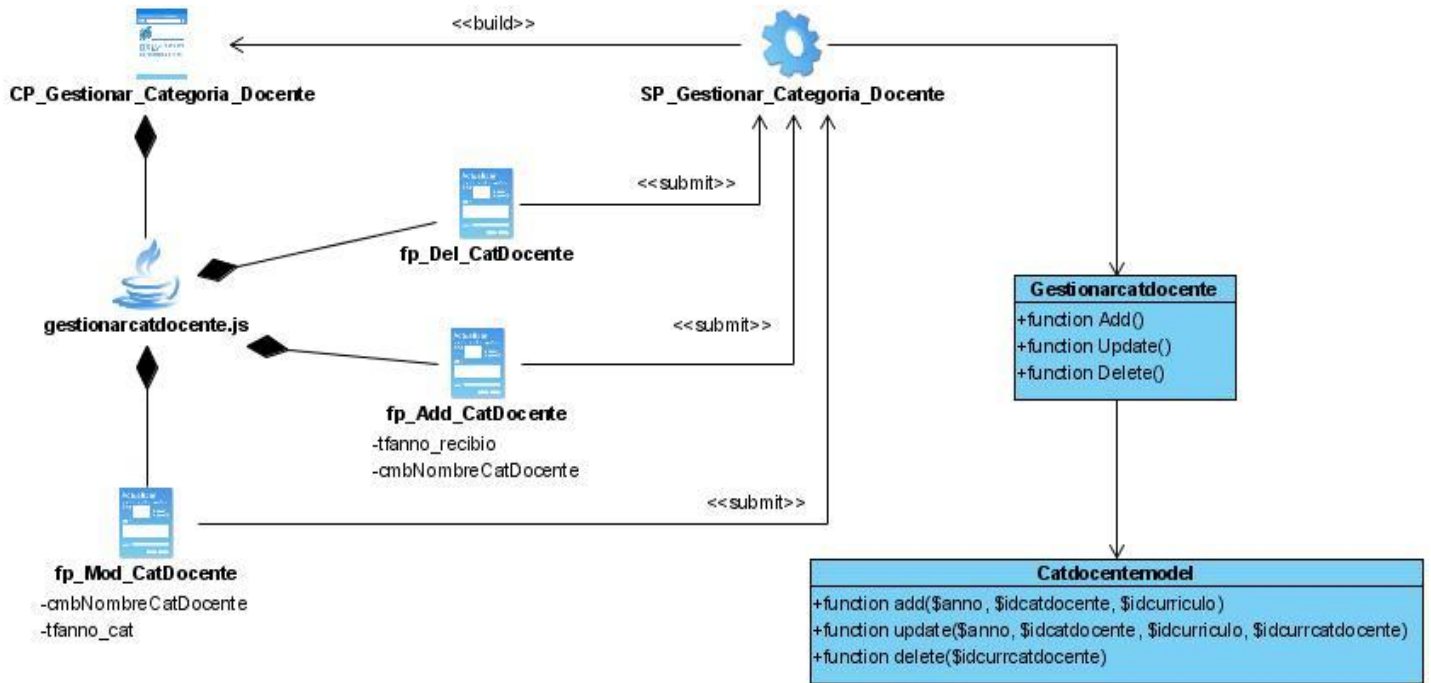


Figura 39 DCW "Gestionar categoría docente".

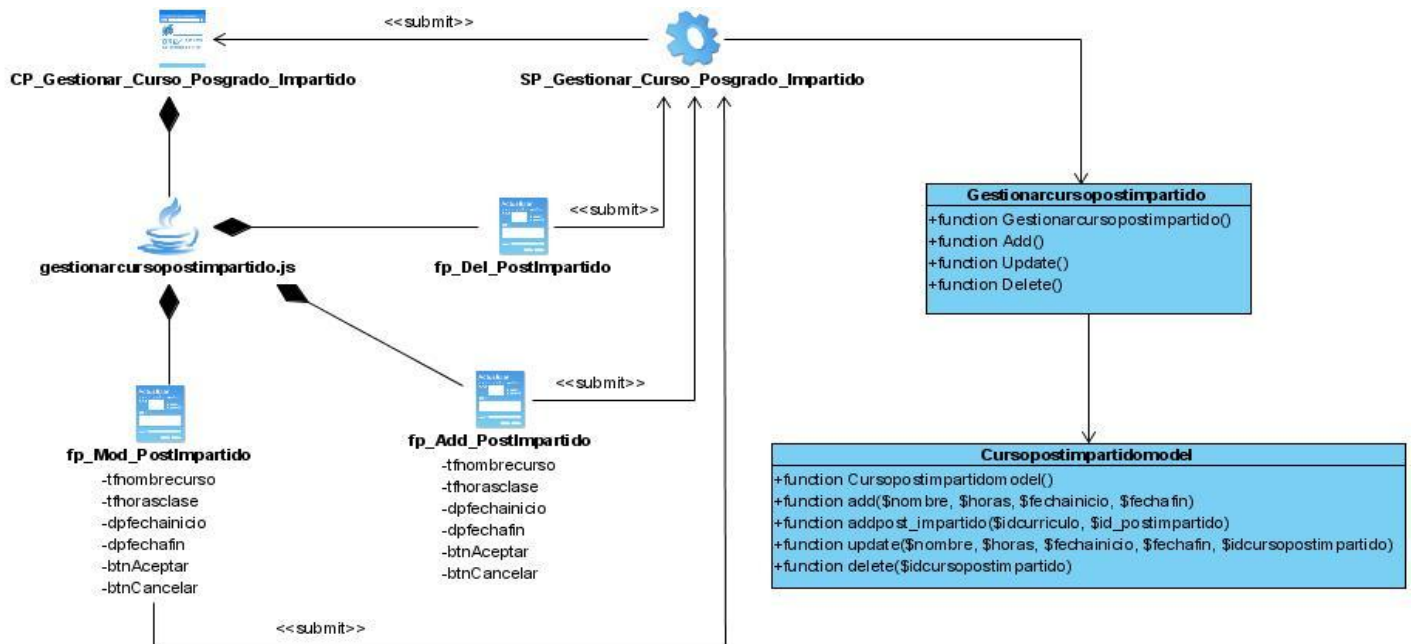


Figura 40 DCW "Gestionar cursos de postgrado impartidos".

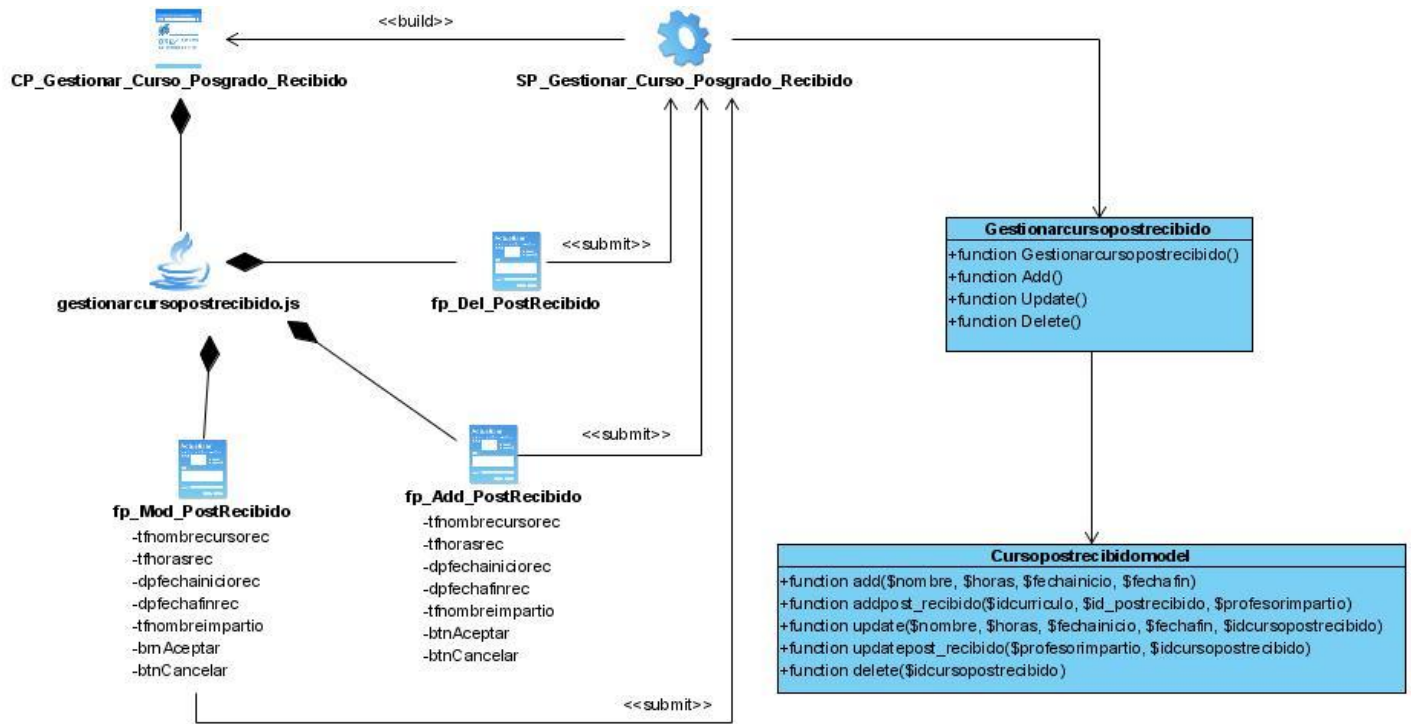


Figura 41 DCW "Gestionar cursos de postgrado recibidos".

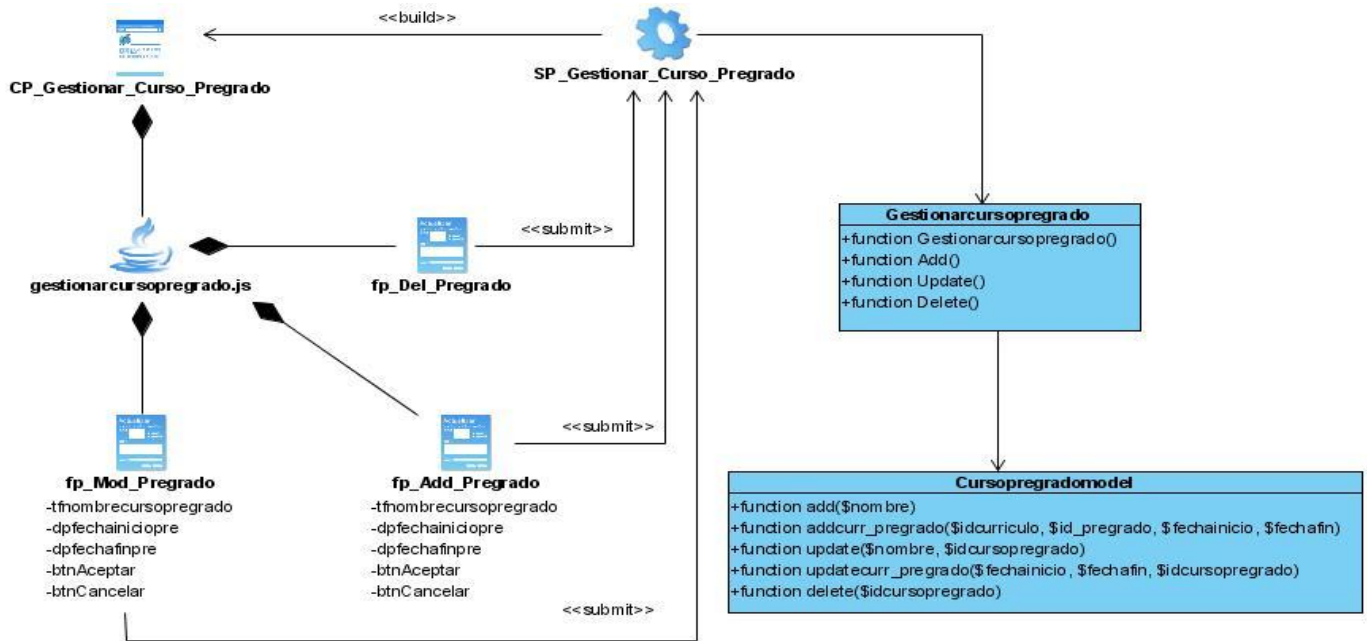


Figura 42 DCW "Gestionar curso pregrado impartidos".

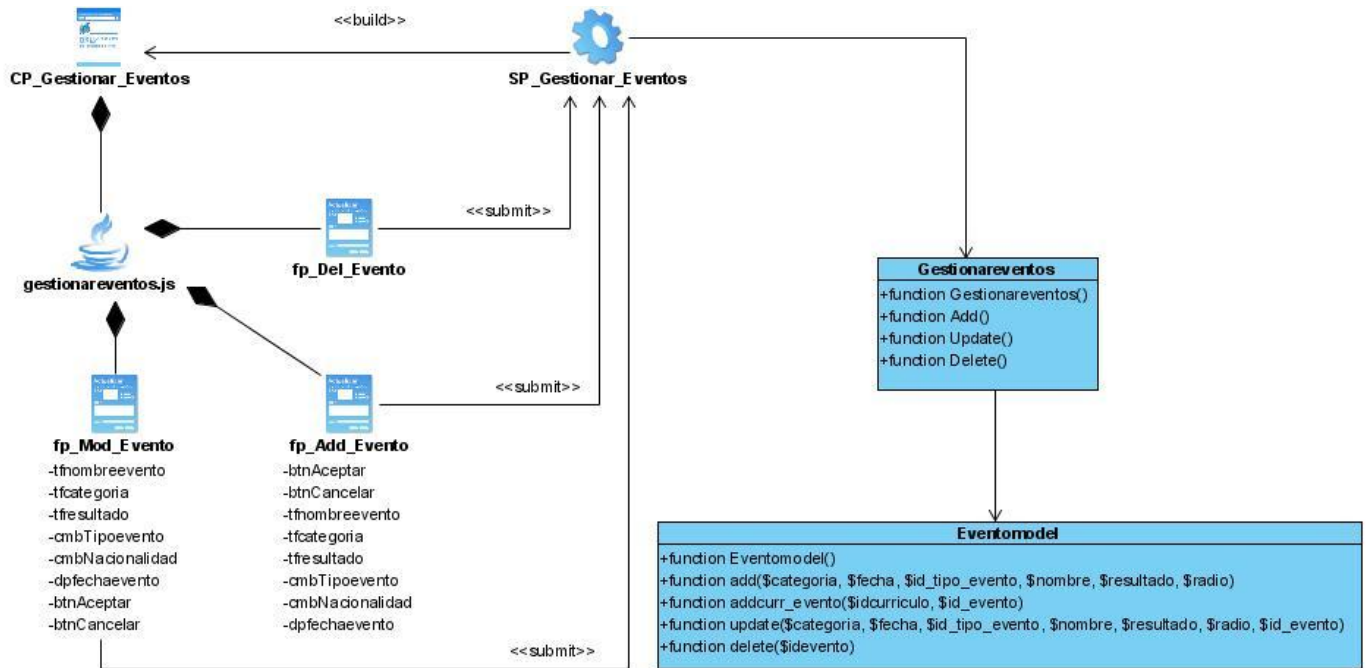


Figura 43 DCW "Gestionar eventos".

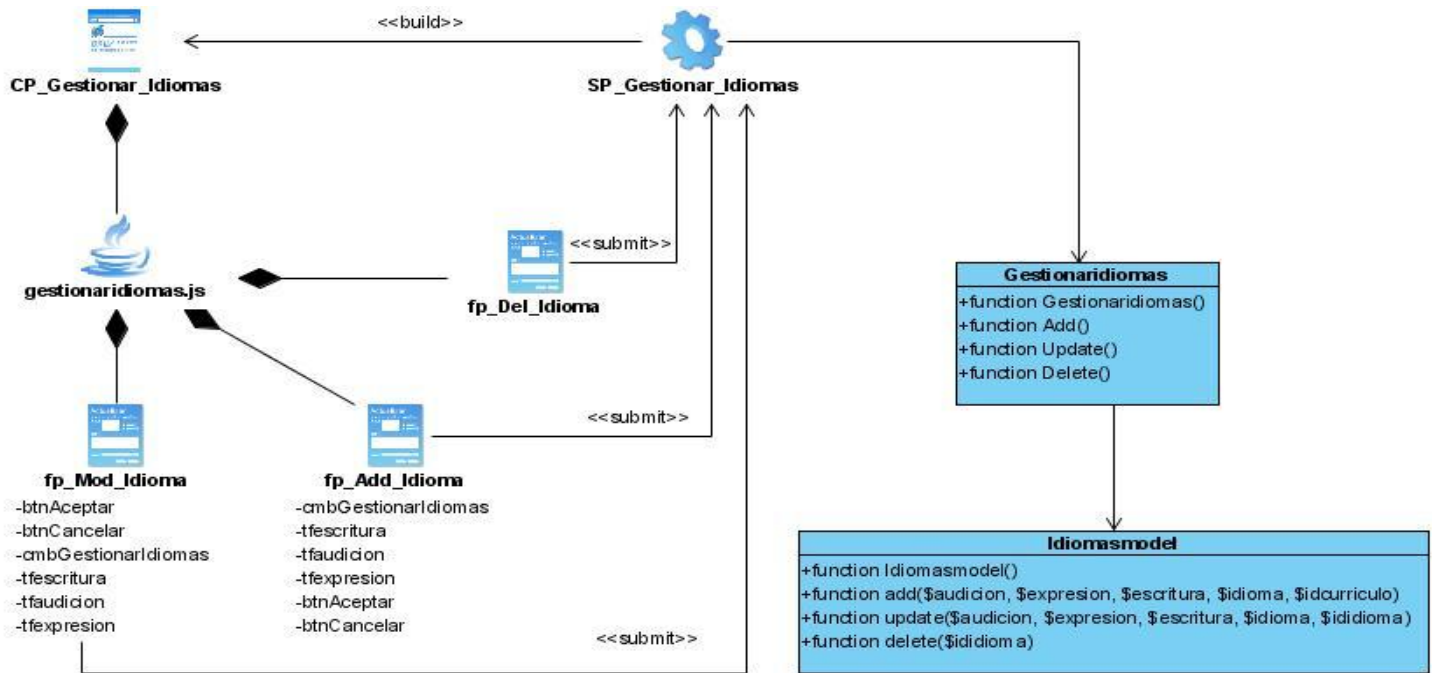


Figura 44 DCW "Gestionar idiomas".

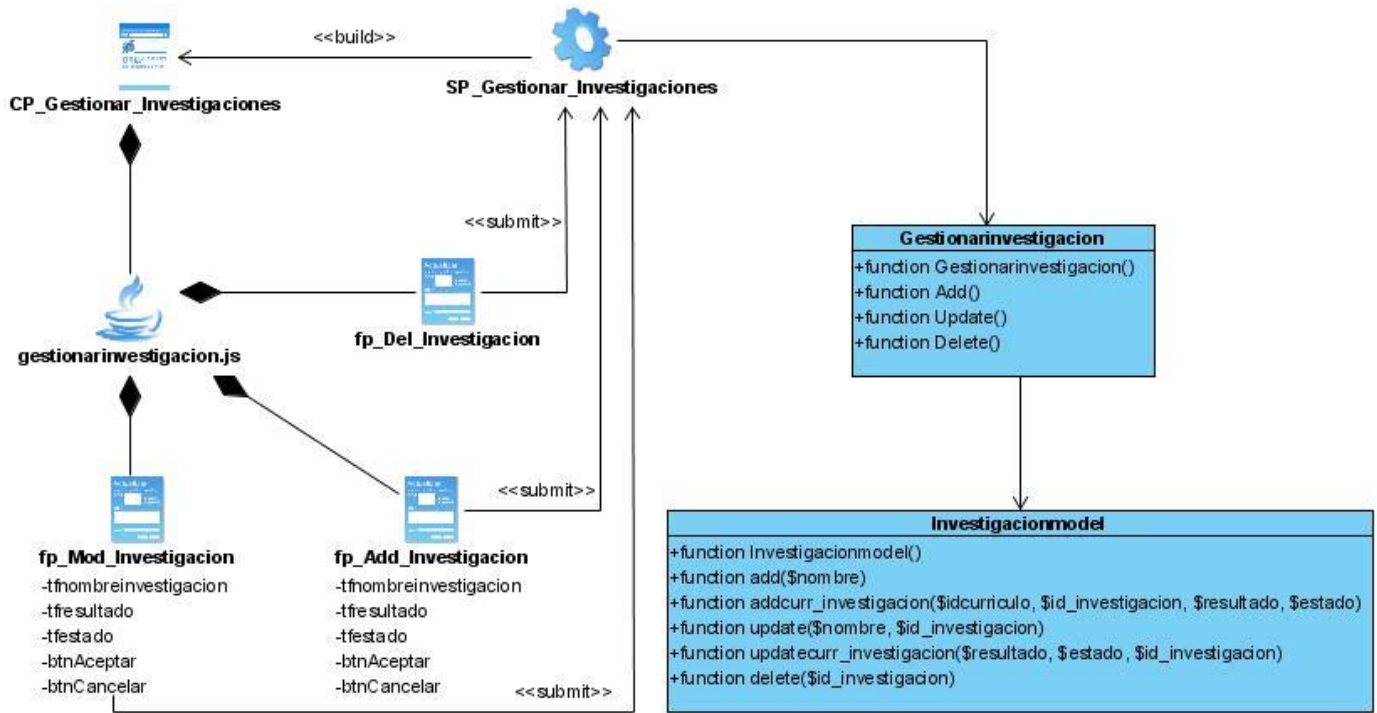


Figura 45 DCW "Gestionar investigaciones".

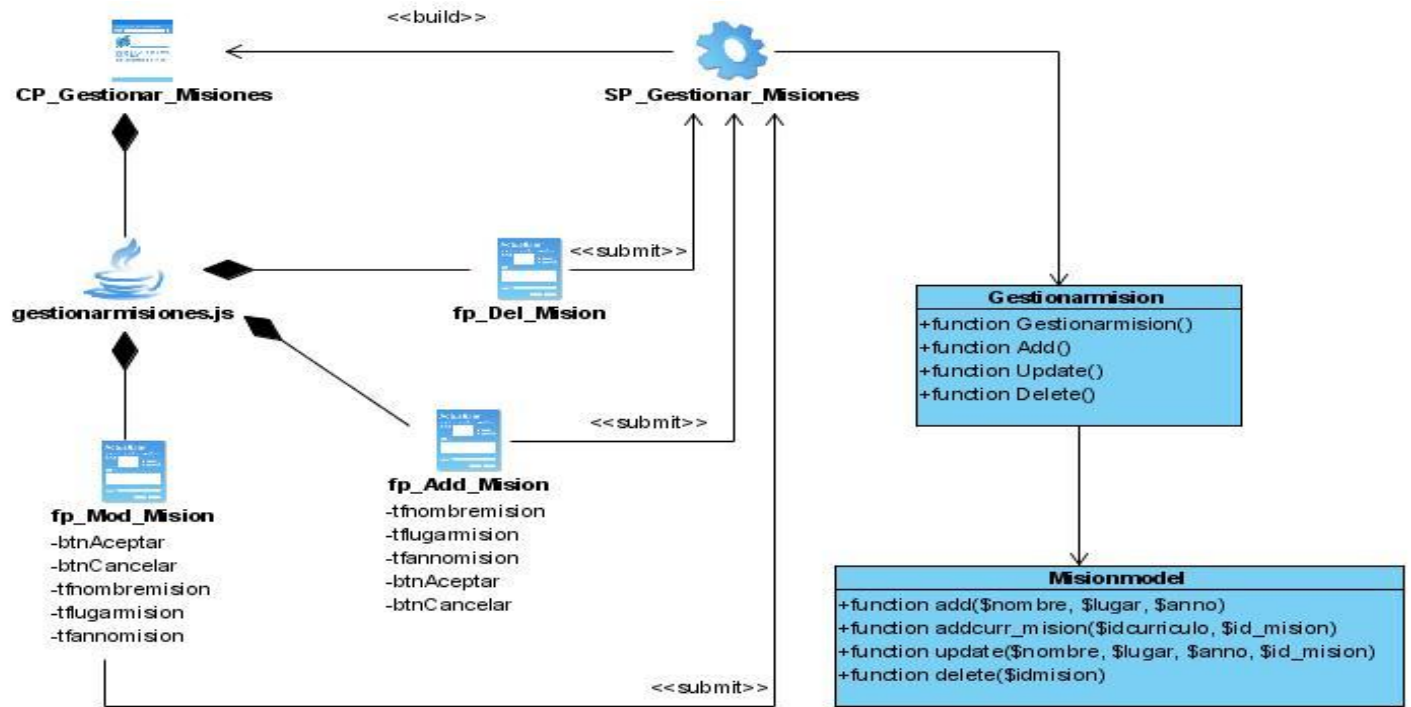


Figura 46 DCW "Gestionar misiones".

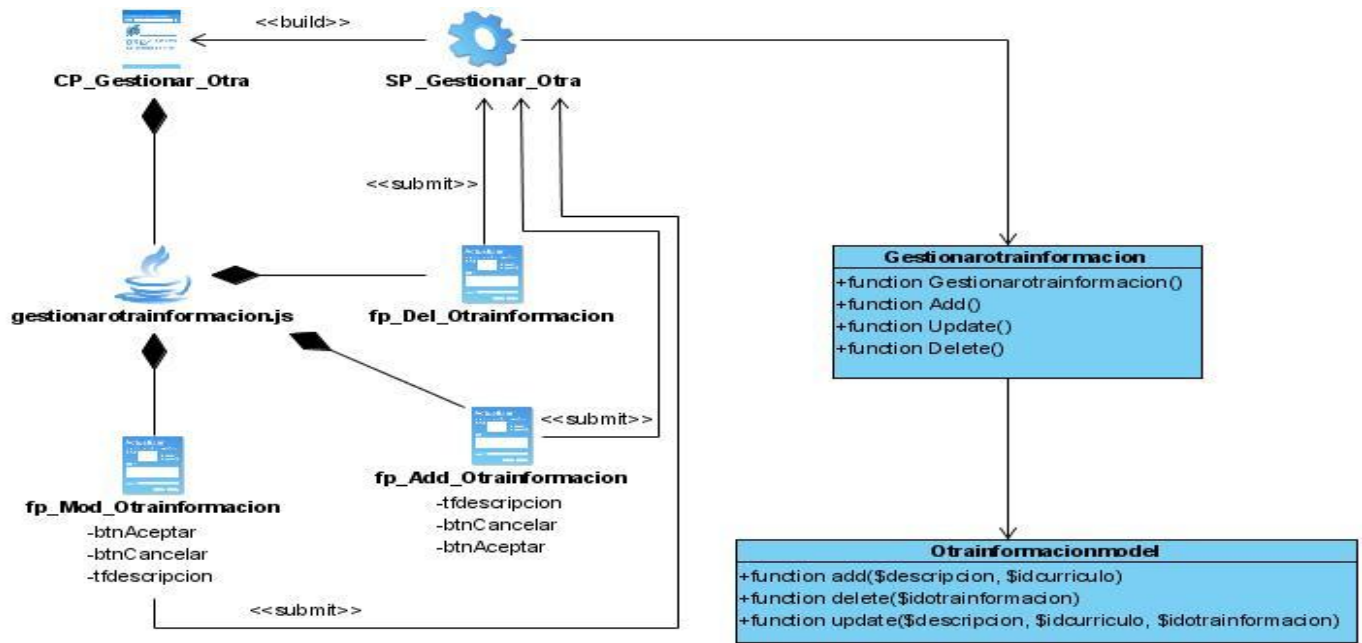


Figura 47 DCW "Gestionar otra información".

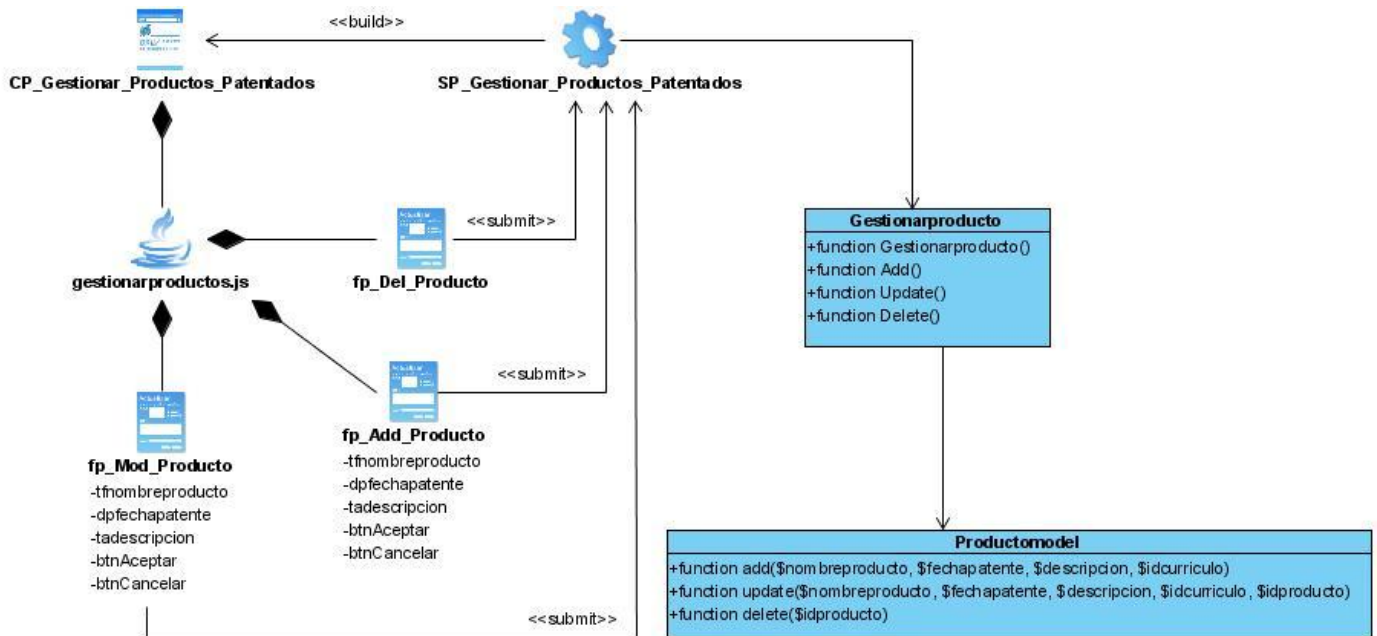


Figura 48 DCW "Gestionar productos patentados".

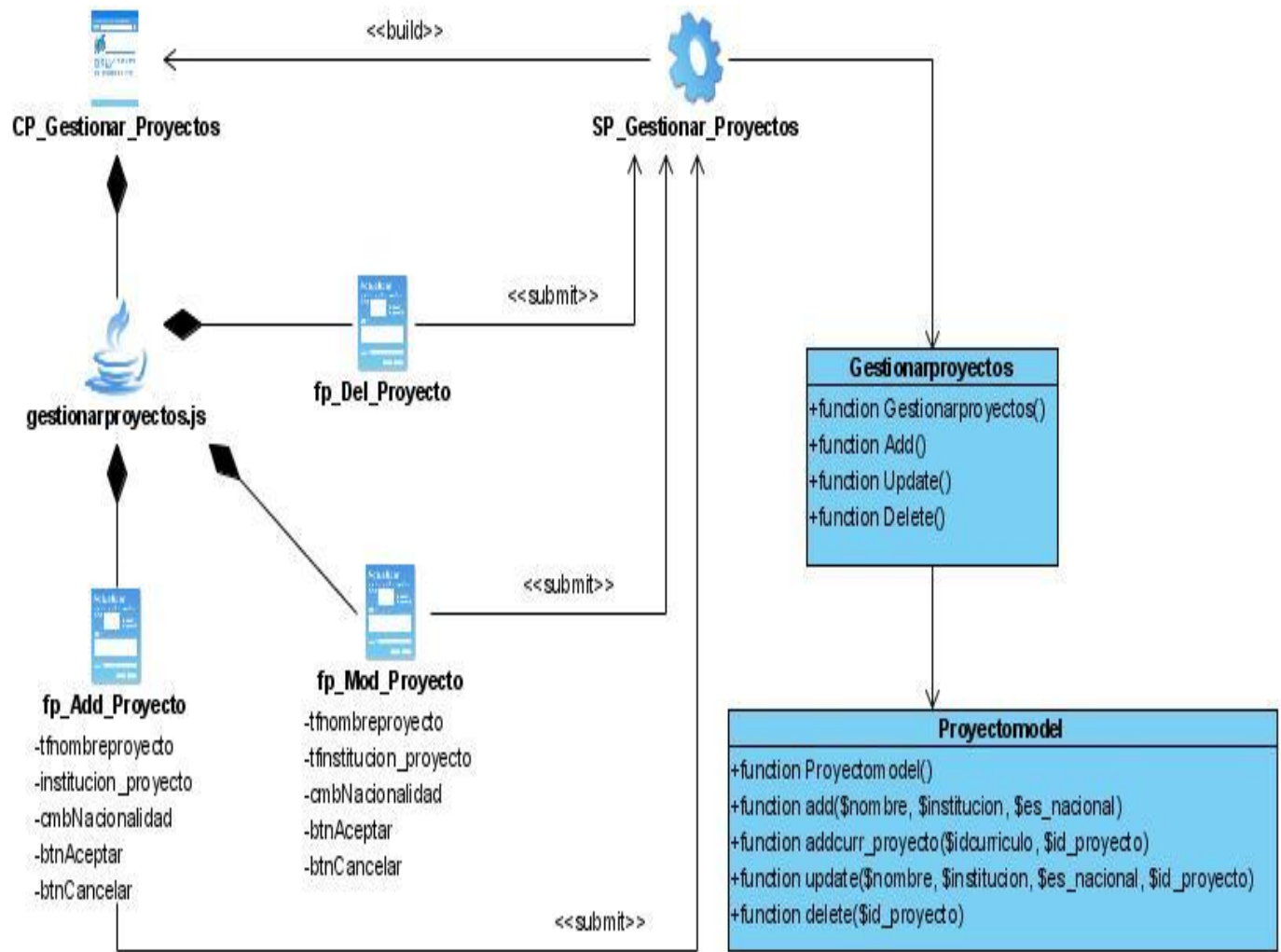


Figura 49 DCW "Gestionar proyectos".

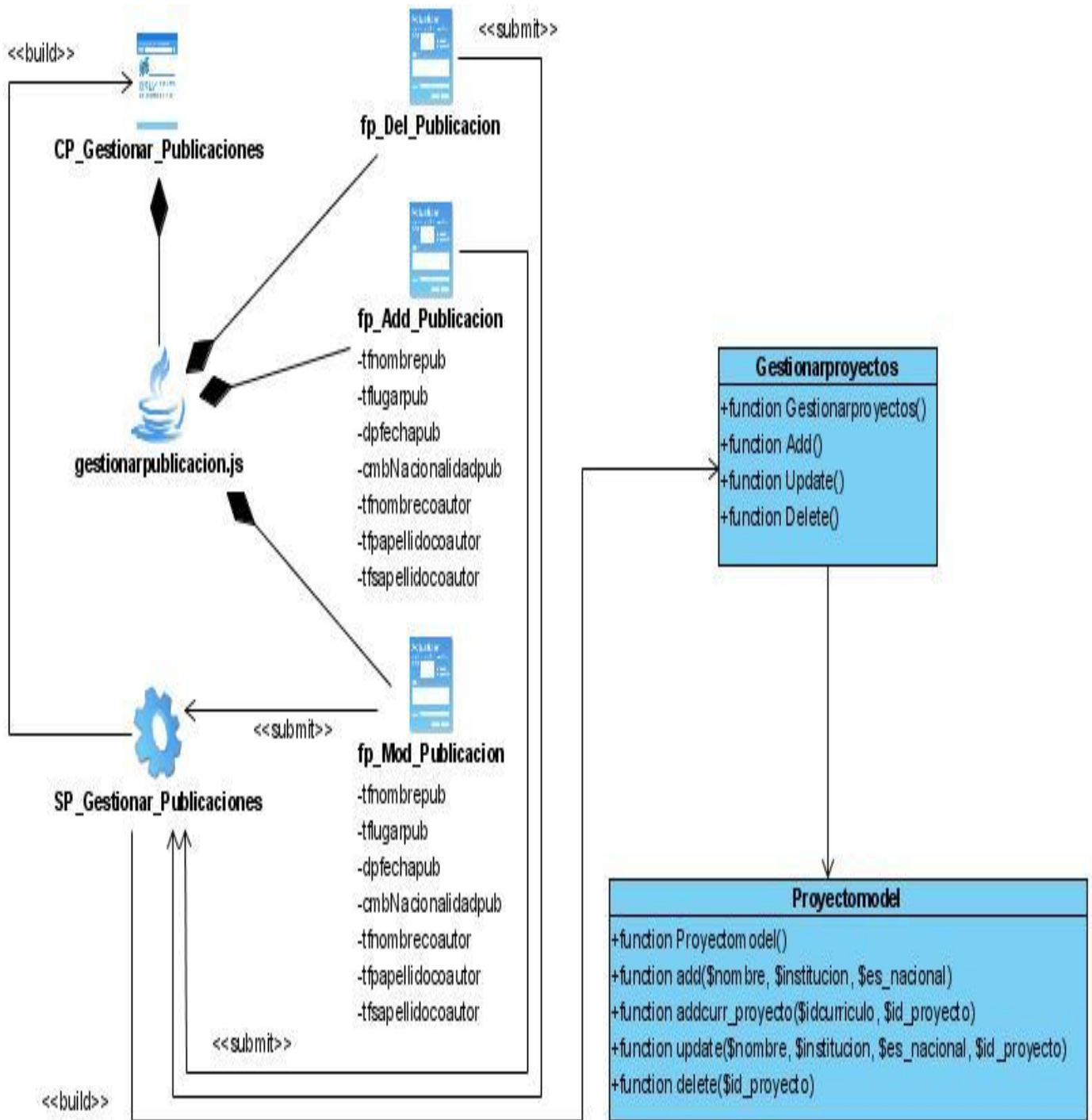


Figura 50 DCW "Gestionar publicaciones".

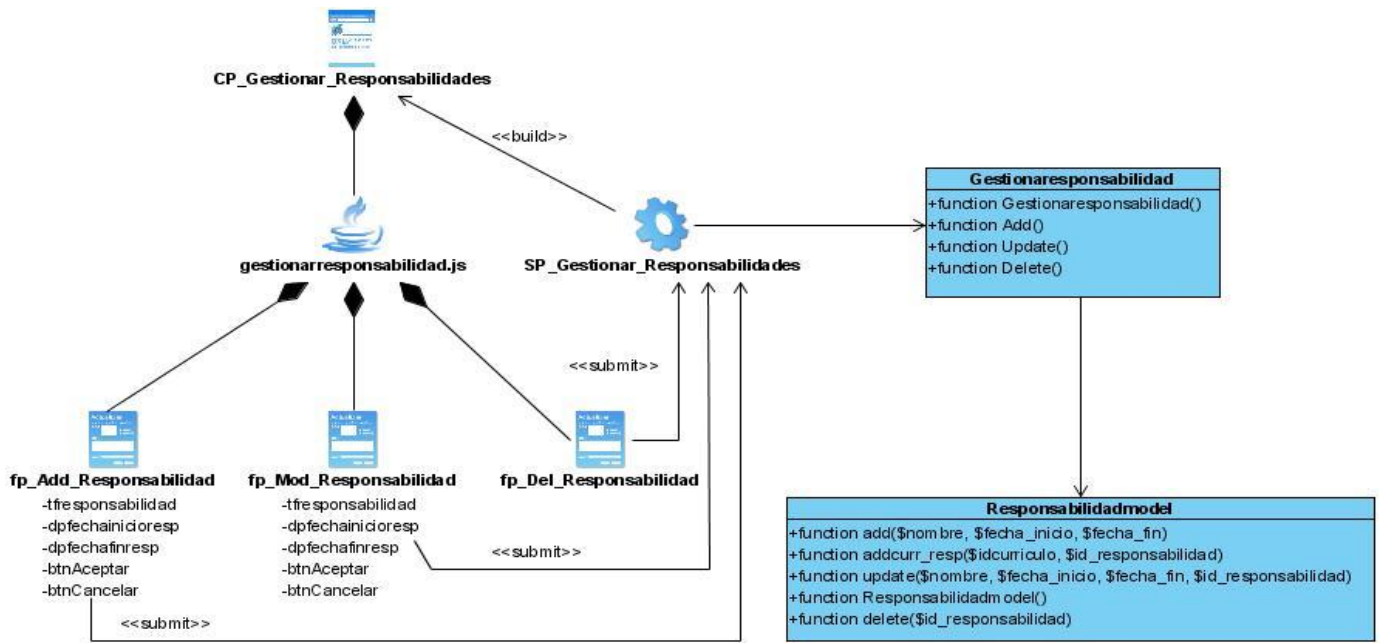


Figura 51 DCW "Gestionar responsabilidades ocupadas".

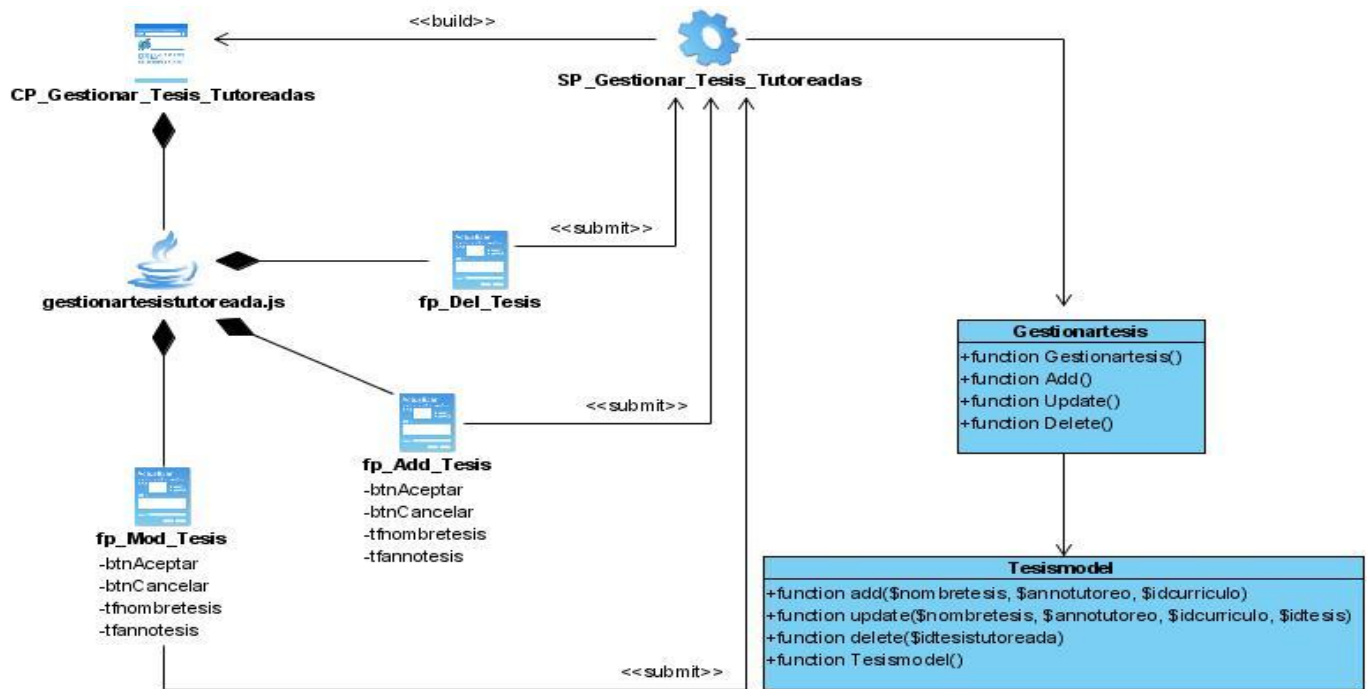


Figura 52 DCW "Gestionar tesis tutoradas".

Anexo 3. Descripción de Clases del Modelo.

Nombre: Catcientificamodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo
Nombre:	Descripción:
Catcientificamodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
cargarCombocatcientifica()	Devuelve el listado de las categorías científicas.
add (\$anno, \$idcatcientifica, \$lugar, \$idcurriculo, \$especialidad)	Adiciona los datos de una categoría científica en la base de datos.
cargarGridcatcientifica (\$limit, \$start, \$idcurriculo)	Devuelve un listado con los datos de la categoría científica.
cargarformcatcientifica (\$id_curr_cat_cientifica)	Devuelve la categoría científica del usuario.
update (\$anno, \$idcatcientifica, \$lugar, \$idcurriculo,\$especialidad,\$idcurratcientifica)	Modifica los datos de la categoría científica.
delete (\$idcurratcientifica)	Elimina una categoría científica de la base de datos.

Tabla 61 Descripción de la Clase del Modelo "Catcientificamodel".

Nombre: Catdocentemodel.

Tipo de clase: Modelo.	
Atributo:	Tipo
Nombre:	Descripción:
Catdocentemodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
cargarCombocatdocente ()	Devuelve el listado de las categorías docentes.
add (\$anno, \$idcatdocente, \$idcurriculo)	Añade un usuario en la base de datos.
cargarGridcatdocente (\$limit, \$start, \$idcurriculo)	Devuelve un listado con los datos de la categoría docente.
cargarformcatdocente (\$id_curr_cat_docente)	Devuelve la categoría docente del usuario.
update (\$anno, \$idcatdocente, \$idcurriculo, \$idcurratdocente)	Modifica los datos de la categoría docente.
delete (\$idcurratdocente)	Elimina los datos de una categoría docente de la base de datos.

Tabla 62 Descripción de la Clase del Modelo "Catdocentemodel".

Nombre: Cursopostimpartidomodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo

Nombre:	Descripción:
Cursopostimpartidomodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
add (\$nombre, \$horas,\$fechainicio,\$fechafin)	Añade un curso de postgrado impartido en la tabla curso_postgrado.
addpost_impartido(\$sidcurriculo,\$sid_postimpartido)	Añade un curso de postgrado en la tabla curso_postgrado_impertido.
cargarGridpostimpartido (\$limit, \$start, \$sidcurriculo)	Devuelve un listado con los datos de los cursos de postgrado impartidos.
cargarformpostimpartido (\$sidcursopostimpartido)	Devuelve los datos de un curso de postgrado impartido.
update (\$nombre, \$horas,\$fechainicio, \$fechafin,\$sidcursopostimpartido)	Modifica los datos de los cursos de postgrado impartidos.
delete (\$sidcursopostimpartido)	Elimina los datos de un curso de postgrado impartido.

Tabla 63 Descripción de la Clase del Modelo "Cursopostimpartidomodel".

Nombre: Cursopostrecibidomodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo

Nombre:	Descripción:
Cursopostrecibidomodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
add (\$nombre,\$horas,\$fechainicio,\$fechafin)	Adiciona un curso de postgrado recibido en la tabla curso_postgrado.
addpost_recibido(\$idcurriculo,\$ id_postrecibido, \$profesorimpartio)	Adiciona un curso de postgrado en la tabla curso_postgrado_Recibido.
cargarGridpostrecibido (\$limit, \$start, \$idcurriculo)	Devuelve un listado con los datos de los cursos de postgrado recibidos.
cargarformpostrecibido (\$idcursopostimpartido)	Devuelve los datos de los cursos de postgrado recibidos del usuario.
update(\$nombre, \$horas, \$fechainicio,\$fechafin,\$idcursopostrecibido)	Modifica los datos de los cursos de postgrado en la tabla curso_postgrado.
updatepost_recibido(\$profesorimpartio,\$idcursopostrecibido)	Modifica los datos en la tabla cursos_postgrado_Recibido
delete (\$idcursopostrecibido)	Elimina los datos de un curso de postgrado recibido.

Tabla 64 Descripción de la Clase del Modelo "Cursopostrecibidomodel".

Nombre: Cursopregradomodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo

Nombre:	Descripción:
Cursopregradomodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
add (\$nombre)	Adiciona un curso de pregrado en la tabla curso_pregrado.
addcurr_pregrado \$idcurriculo,\$id_pregrado, \$fechainicio, \$fechafin)	Adiciona un curso de pregrado en la tabla curso_pregrado_impartido.
cargarGridpregrado (\$limit, \$start, \$idcurriculo)	Devuelve un listado con los datos de los cursos de pregrado impartidos.
cargarformpregrado (\$idcursopregrado)	Devuelve los datos de los cursos de pregrado impartidos por el usuario.
update(\$nombre, \$idcursopregrado)	Modifica los datos de los cursos de regrado en la tabla curso_pregrado_impertido.
updatepost_ updatecurr_pregrado(\$fechainicio,\$fechafin,\$idcursopregrado)	Modifica los datos en la tabla currículu_cursos_pregrado
delete (\$idcursopregrado)	Elimina los datos de un curso de pregrado.

Tabla 65 Descripción de la Clase del Modelo "Cursopregradomodel".

Nombre: Eventomodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo

Nombre:	Descripción:
Eventomodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
cargarComboevento()	Devuelve el listado de los tipos de eventos.
add (\$categoria, \$fecha,\$sid_tipo_evento,\$nombre,\$resultado,\$radio)	Adiciona un evento.
addcurr_evento(\$sidcurriculo,\$sid_evento)	Adiciona un evento en la tabla currícul_evento.
cargarGrideventos (\$limit, \$start, \$sidcurriculo)	Devuelve un listado con los datos de los eventos.
cargarformevento (\$sid_evento,\$sidcurriculo)	Devuelve los datos de los eventos del usuario.
update(\$categoria, \$fecha,\$sid_tipo_evento,\$nombre,\$resultado,\$radio,\$i d_evento)	Modifica los datos de los eventos.
delete (\$sidevento)	Elimina un evento.

Tabla 66 Descripción de la Clase del Modelo "Eventomodel".

Nombre: Idiomasmodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo

Nombre:	Descripción:
Idiomamodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
add (\$audicion,\$expresion,\$escritura,\$idioma,\$idcurriculo)	Adiciona un dominio de idioma.
cargarGrididioma (\$limit, \$start, \$idcurriculo)	Devuelve un listado con los datos de los dominios de idiomas.
cargarformidioma \$(idioma)	Devuelve los datos de los datos de los idiomas que dominan usuario.
update (\$audicion,\$expresion,\$escritura,\$idioma,\$ididioma)	Modifica los datos de los dominios de idiomas.
delete (\$ididioma)	Elimina un dominio de idioma.

Tabla 67 Descripción de la Clase del Modelo "Idiomamodel".

Nombre: Investigacionmodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo
Nombre:	Descripción:
Investigacionmodel ()	Constructor de la clase. Hereda de la clase

	Model del framework CodeIgniter.
add (\$nombre)	Adiciona una investigación.
addcurr_investigacion(\$idcurriculo,\$id_investigacion,\$ resultado,\$estado)	Adiciona una investigación en la tabla currícul_investigación.
cargarGridinvestigacion(\$limit , \$start, \$idcurriculo)	Devuelve un listado con los datos de las investigaciones.
cargarforminvestigacion(\$id_investigacion)	Devuelve los datos de las investigaciones de un usuario.
update (\$nombre, \$id_investigacion)	Modifica los datos de una investigación.
updatecurr_investigacion(\$resultado,\$estado,\$id_investigacion)	Modifica los datos de una investigación en la tabla currícul_investigación.
delete(\$id_investigacion)	Elimina los datos de una investigación.

Tabla 68 Descripción de la Clase del Modelo "Investigacionmodel".

Nombre: Misionmodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo
Nombre:	Descripción:
Misionmodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.

add (\$nombre,\$lugar,\$anno)	Adiciona una misión.
addcurr_mision(\$idcurriculo,\$id_mision)	Adiciona una misión en la tabla currículum_mision.
cargarGridmision(\$limit , \$start, \$idcurriculo)	Devuelve los datos de las misiones.
cargarformmision(\$id_mision,\$idcurriculo)	Devuelve la misión a modificar.
update (\$nombre,\$lugar,\$anno,\$id_mision)	Modifica una misión.
delete(\$idmision)	Elimina una misión.

Tabla 69 Descripción de la Clase del Modelo "Misionmodel".

Nombre: Tesismodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo
Nombre:	Descripción:
Tesismodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
add (\$nombretesis,\$annotutoreo,\$idcurriculo)	Adiciona una tesis tutorada.
cargarGridtesis(\$limit , \$start, \$idcurriculo)	Devuelve los datos de las tesis tutoradas.
delete(\$idtesistutoreada)	Elimina una tesis.
cargarformtesis(\$id_tesis)	Devuelve los datos de una tesis tutorada.

update (\$nombretesis,\$annotutoreo,\$idcurriculo,\$idtesis)	Modifica los datos de una tesis tutorada.
---	---

Tabla 70 Descripción de la Clase del Modelo "Tesismodel".

Nombre: Otrainformacionmodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo
Nombre:	Descripción:
Otrainformacionmodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
add (\$descripcion,\$idcurriculo)	
cargarGridotrainformacion(\$limit ,\$start, \$idcurriculo)	
delete(\$idotrainformacion)	
cargarformotrainformacion(\$idotrainformacion)	
update (\$descripcion,\$idcurriculo,\$idotrainformacion)	

Tabla 71 Descripción de la Clase del Modelo "Otrainformacionmodel."

Nombre: Productomodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo



Nombre:	Descripción:
Productomodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
add (\$nombreproducto,\$fechapatente,\$descripcion,\$idcurriculo)	
cargarGridproducto(\$limit , \$start, \$idcurriculo)	
delete(\$idproducto)	
cargarformproducto(\$id_producto)	
update (\$nombreproducto,\$fechapatente,\$descripcion,\$idcurriculo,\$idproducto)	

Tabla 72 Descripción de la Clase del Modelo "Productomodel".

Nombre: Proyectomodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo
Nombre:	Descripción:
Proyectomodel ()	Constructor de la clase. Hereda de la clase

	Model del framework CodeIgniter.
add (\$nombre,\$institucion,\$ses_nacional)	Adiciona un proyecto nuevo.
addcurr_proyecto(\$idcurriculo,\$id_proyecto)	Adiciona un proyecto en la tabla proyecto_curriculo.
cargarGridproyecto(\$limi , \$start, \$idcurriculo)	Devuelve los datos de los proyectos.
cargarformproyecto(\$id_proyecto)	Devuelve los datos de un proyecto.
update (\$nombre,\$institucion,\$ses_nacional,\$id_proyecto)	Modifica los datos de un proyecto.
delete(\$id_proyecto)	Elimina un proyecto.


Tabla 73 Descripción de la Clase del Modelo "Proyctomodel".

Nombre: Publicacionmodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo
Nombre:	Descripción:
Publicacionmodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
add (\$nombrep,\$fecha,\$lugar,\$nacionalidad,\$idcurriculo)	Adiciona una publicación.
addco_autor(\$nombrecoautor,\$papellidocoautor,\$sap	Adiciona un co-autor.

ellidocoautor,\$sid_publicacion)	
cargarGridpublicacion(\$limit,\$start, \$sidcurriculo)	Devuelve los datos de las publicaciones.
cargarformpublicacion(\$sid_publicacion)	Devuelve los datos de una publicación.
update (\$nombrep,\$fecha,\$lugar,\$nacionalidad,\$sid_publicacion)	Modifica una publicación.
updateco_autor (\$nombrecoautor,\$papellidocoautor,\$sapellidocoautor,\$sid_publicacion)	Modifica un co-autor.
delete(\$sid_publicacion)	Elimina una publicación.

Tabla 74 Descripción de la Clase del Modelo "Publicacionmodel".

Nombre: Responsabilidadmodel.	
Tipo de clase: Modelo.	
Atributo:	Tipo
Nombre:	Descripción:
Responsabilidadmodel ()	Constructor de la clase. Hereda de la clase Model del framework CodeIgniter.
add (\$nombre,\$fecha_inicio, \$fecha_fin)	Adiciona una responsabilidad.
addcurr_resp(\$sidcurriculo,\$sid_responsabilidad)	Adiciona una responsabilidad en la tabla



	currículo_responsabilidad.
cargarGridresponsabilidad(\$limit,\$start, \$idcurriculo)	Devuelve los datos de las responsabilidades.
cargarformresponsabilidad(\$id_responsabilidad)	Devuelve los datos de una responsabilidad.
update (\$nombre,\$fecha_inicio, \$fecha_fin,\$id_responsabilidad)	Modifica una responsabilidad.
delete(\$id_responsabilidad)	Elimina responsabilidad.

Tabla 75 Descripción de la Clase del Modelo "Responsabilidadmodel".

Anexo 4. Descripción de Clases Controladoras.

Nombre: Gestionarcaticientifica.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objetode de tipo Catcientificamodel.
\$ __modelp	Objeto de tipo Profesormodel.
Nombre:	Descripción:
Gestionarcaticientifica ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Añade una nueva categoría científica.
CargarGridcaticientifica ()	Carga el grid con los datos de las categorías científicas.
Cargarformcaticientifica ()	Carga el formulario con los datos de las categorías científicas que serán modificados.
Cargarcmbnombrecaticientifica ()	Carga el combo con los nombres de las categorías científicas.
Update ()	Modifica una categoría científica.
Delete ()	Elimina una categoría científica.

Tabla 76 Descripción de la Clase Controladora "Gestionar Categoría Científica".

Nombre: Gestionarcatdocente.

Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objetode de tipo Catdocentemodel.
\$ __modelp	Objeto de tipo Profesormodel.
Nombre:	Descripción:
Gestionarcatdocente ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Adiciona una nueva categoría docente.
CargarGridcatdocente ()	Carga el grid con los datos de la categoría docente.
Cargarformcatdocente ()	Carga el formulario con los datos de la categoría docente que serán modificados.
Cargarcmbnombrecatdocente ()	Carga el combo con los nombres de la categoría docente.
Update ()	Modifica la categoría docente.
Delete ()	Elimina la categoría docente.

Tabla 77 Descripción de la Clase Controladora "Gestionar Categoría Docente".

Nombre: Gestionarcursopostgrado.
Tipo de clase: Controladora.

Atributo:	Tipo
\$ _model	Objetode de tipo Cursopostgradomodel.
\$ __modelp	Objeto de tipo Profesormodel.
Nombre:	Descripción:
Gestionarcursopostgrado ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Adiciona un nuevo curso de postgrado.
CargarGridpostgrado()	Carga el grid con los datos del curso de postgrado.
Cargarformpostgrado ()	Carga el formulario con los datos del curso de postgrado.
Update ()	Modifica un curso de postgrado.
Delete ()	Elimina un curso de postgrado.

Tabla 78 Descripción de la Clase Controladora "Gestionar cursos de postgrado".

Nombre: Gestionarcursopregrado.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objetode de tipo Cursopregradomodel.
\$ __modelp	Objeto de tipo Profesormodel.
Nombre:	Descripción:

Gestionarcursopregrado ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Añade un nuevo curso de pregrado.
CargarGridpregrado ()	Carga el grid con los datos del curso de pregrado.
Cargarformpregrado ()	Carga el formulario con los datos del curso de pregrado.
Update ()	Modifica un curso de pregrado.
Delete ()	Elimina un curso de pregrado.

Tabla 79 Descripción de la Clase Controladora "Gestionar cursos de pregrado impartidos".

Nombre: Gestionaresponsabilidad.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objeto de tipo Responsabilidadmodel.
\$ __modelp	Objeto de tipo Profesormodel.
Nombre:	Descripción:
Gestionaresponsabilidad ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Añade una nueva responsabilidad.
CargarGridresponsabilidad ()	Carga el grid con los datos de las

	responsabilidades.
Cargarformresponsabilidad ()	Carga el formulario con los datos de las responsabilidades.
Update ()	Modifica una responsabilidad.
Delete ()	Elimina una responsabilidad.

Tabla 80 Descripción de la Clase Controladora "Gestionar responsabilidades ocupadas".

Nombre: Gestionareventos.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objetode de tipo Eventomodel.
\$ __modelp	Objeto de tipo Profesormodel.
Nombre:	Descripción:
Gestionareventos ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Adiciona un nuevo evento.
Cargarcmbnombreevento()	Carga el combo con los tipos de eventos.
CargarGrideventos ()	Carga el grid con los datos de los eventos.
Cargarformevento ()	Carga el formulario con los datos de los eventos.
Update ()	Modifica un evento.

Delete ()	Elimina un evento.
-----------	--------------------

Tabla 81 Descripción de la Clase Controladora "Gestionar eventos".

Nombre: GestionarIdiomas.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objeto de tipo IdiomasModel.
\$ __modelp	Objeto de tipo ProfesorModel.
Nombre:	Descripción:
GestionarIdiomas ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Añade un nuevo idioma.
CargarCambIdiomas ()	Carga el combo con los idiomas.
CargarGridIdioma ()	Carga el grid con los datos de los idiomas.
CargarFormIdioma ()	Carga el formulario con los datos de los idiomas.
Update ()	Modifica un idioma.
Delete ()	Elimina un idioma.

Tabla 82 Descripción de la Clase Controladora "Gestionar idiomas".

Nombre: GestionarInvestigacion.
--

Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objetode de tipo Investigacionmodel.
\$ __modelp	Objeto de tipo Profesormodel.
Nombre:	Descripción:
Gestionarinvestigacion ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Añade una nueva investigación.
CargarGridinvestigacion ()	Carga el grid con los datos de las investigaciones.
Cargarforminvestigacion ()	Carga el formulario con los datos de las investigaciones.
Update ()	Modifica una investigación.
Delete ()	Elimina una investigación.

Tabla 83 Descripción de la Clase Controladora "Gestionar investigación".

Nombre: Gestionarmision.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objetode de tipo Misionmodel.
\$ __modelp	Objeto de tipo Profesormodel.

Nombre:	Descripción:
Gestionarmision ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Añade una nueva misión.
CargarGridmision ()	Carga el grid con los datos de las misiones.
Cargarformmision ()	Carga el formulario con los datos de las misiones.
Update ()	Modifica una misión.
Delete ()	Elimina una misión.

Tabla 84 Descripción de la Clase Controladora "Gestionar misiones cumplidas".

Nombre: Gestionarotrainformacion.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objeto de tipo Otrainformacionmodel.
\$ __modelp	Objeto de tipo Profesormodel.
Nombre:	Descripción:
Gestionarotrainformacion ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Añade una nueva información extra.
CargarGridotrainformacion ()	Carga el grid con los datos de las informaciones

	extra.
Cargarformotrainformacion ()	Carga el formulario con los datos de las informaciones extra.
Update ()	Modifica una información extra.
Delete ()	Elimina una información extra.

Tabla 85 Descripción de la Clase Controladora "Gestionar otra información".

Nombre: Gestionarproducto.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objetode de tipo Productomodel.
\$ __modelp	Objeto de tipo Profesormodel.
Nombre:	Descripción:
Gestionarproducto ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Adiciona un nuevo producto patentado.
CargarGridproducto ()	Carga el grid con los datos de los productos patentados.
Cargarformproducto ()	Carga el formulario con los datos de los productos patentados.
Update ()	Modifica un producto patentado.

Delete ()	Elimina un producto patentado.
-----------	--------------------------------

Tabla 86 Descripción de la Clase Controladora "Gestionar productos patentados".

Nombre: Gestionarproyectos.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objetode de tipo Proyectomodel.
\$ __modelp	Objeto de tipo Profesormodel.
Nombre:	Descripción:
Gestionarproyectos ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Añade un nuevo proyecto.
CargarGridproyecto ()	Carga el grid con los datos de los proyectos.
Cargarformproyecto ()	Carga el formulario con los datos de los proyectos.
Update ()	Modifica un proyecto.
Delete ()	Elimina un proyecto.

Tabla 87 Descripción de la Clase Controladora "Gestionar proyectos".

Nombre: Gestionarpublicacion.
Tipo de clase: Controladora.

Atributo:	Tipo
\$ _model	Objetode de tipo Publicacionmodel.
\$ __modelp	Objeto de tipo Profesormodel.
Nombre:	Descripción:
Gestionarpublicacion ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Adiciona una nueva publicación.
CargarGridpublicacion ()	Carga el grid con los datos de las publicaciones.
Cargarformpublicacion ()	Carga el formulario con los datos de las publicaciones.
Update ()	Modifica una publicación.
Delete ()	Elimina una publicación.

Tabla 88 Descripción de la Clase Controladora "Gestionar publicaciones".

Nombre: Gestionartesis.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objetode de tipo Tesismodel.
\$ __modelp	Objeto de tipo Profesormodel.
Nombre:	Descripción:

Gestionartesis ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Añade una nueva tesis tutorada.
CargarGridtesis ()	Carga el grid con los datos de las tesis tutoradas.
Cargarformtesis ()	Carga el formulario con los datos de las tesis tutoradas.
Update ()	Modifica una tesis tutorada.
Delete ()	Elimina una tesis tutorada.

Tabla 89 Descripción de la Clase Controladora "Gestionar tesis tutoradas".

Nombre: GestionarNomCatCientifica.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objetode de tipo Nomcatcientificamodel.
Nombre:	Descripción:
GestionarNomCatCientifica ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Añade un nuevo nombre de categoría científica.
CargarGridnomcatcientifica ()	Carga el grid con los nombres de las categorías científicas.

Cargarformnomcatcientifica ()	Carga el formulario con los datos de las categorías científicas.
Update ()	Modifica un nombre de categoría científica.
Delete ()	Elimina un nombre de categoría científica.

Tabla 90 Descripción de la Clase Controladora "Gestionar nomenclador categoría científica".

Nombre: GestionarNomCatDocente.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objetode de tipo Nomcatdocentemodel.
Nombre:	Descripción:
GestionarNomCatDocente ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Adiciona un nuevo nombre de categoría docente.
CargarGridnomcatdocente ()	Carga el grid con los nombres de las categorías docentes.
Cargarformnomcatdocente ()	Carga el formulario con los datos de las categorías docentes.
Update ()	Modifica un nombre de categoría docente.
Delete ()	Elimina un nombre de categoría docente.

Tabla 91 Descripción de la Clase Controladora "Gestionar nomenclador categoría docente".

Nombre: GestionarNomIdioma.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objetode de tipo Nomidiomamodel.
Nombre:	Descripción:
GestionarNomIdioma ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Añade un nuevo nombre de idioma.
CargarGridnomidioma ()	Carga el grid con los nombres de los idiomas.
Cargarformnomidioma ()	Carga el formulario con los datos de los idiomas.
Update ()	Modifica un nombre de idioma.
Delete ()	Elimina un nombre de idioma.

Tabla 92 Descripción de la Clase Controladora "Gestionar nomenclador idioma".

Nombre: Gestionartipoeventos.	
Tipo de clase: Controladora.	
Atributo:	Tipo
\$ _model	Objetode de tipo Tipoeventosmodel.
Nombre:	Descripción:

Gestionartipoeventos ()	Constructor de la clase. Hereda de la clase Controller del framework CodeIgniter.
Add ()	Adiciona un nuevo tipo de evento.
CargarGridtipoevento ()	Carga el grid con los nombres de los tipos de eventos.
Cargarformtipoevento ()	Carga el formulario con los datos de los tipos de eventos.
Update ()	Modifica un tipo de evento.
Delete ()	Elimina un tipo de evento.

Tabla 93 Descripción de la Clase Controladora "Gestionar tipo de evento".

Anexo 5. Diagramas de secuencia.

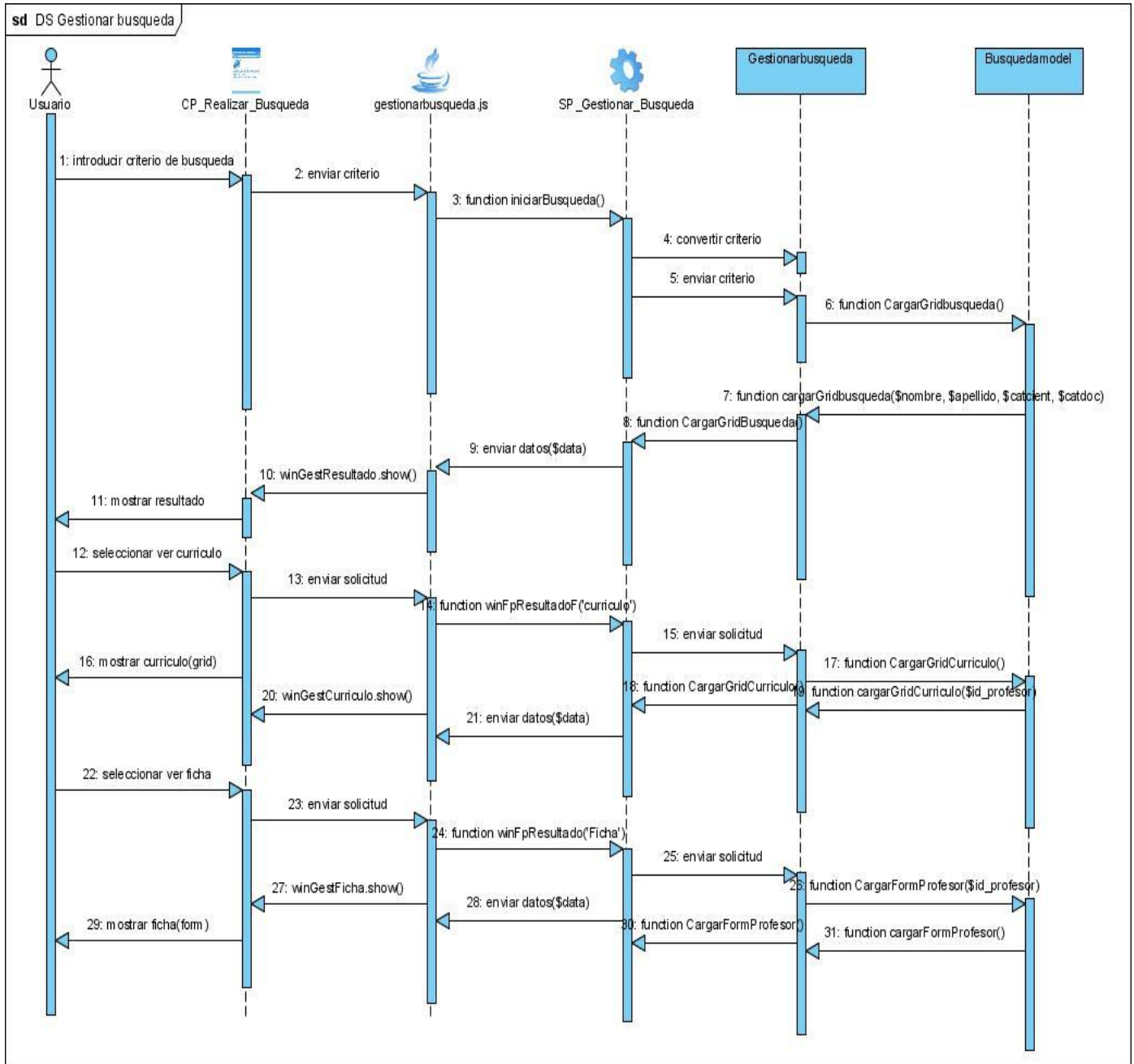


Figura 53 Diagrama de secuencia "Realizar búsqueda".

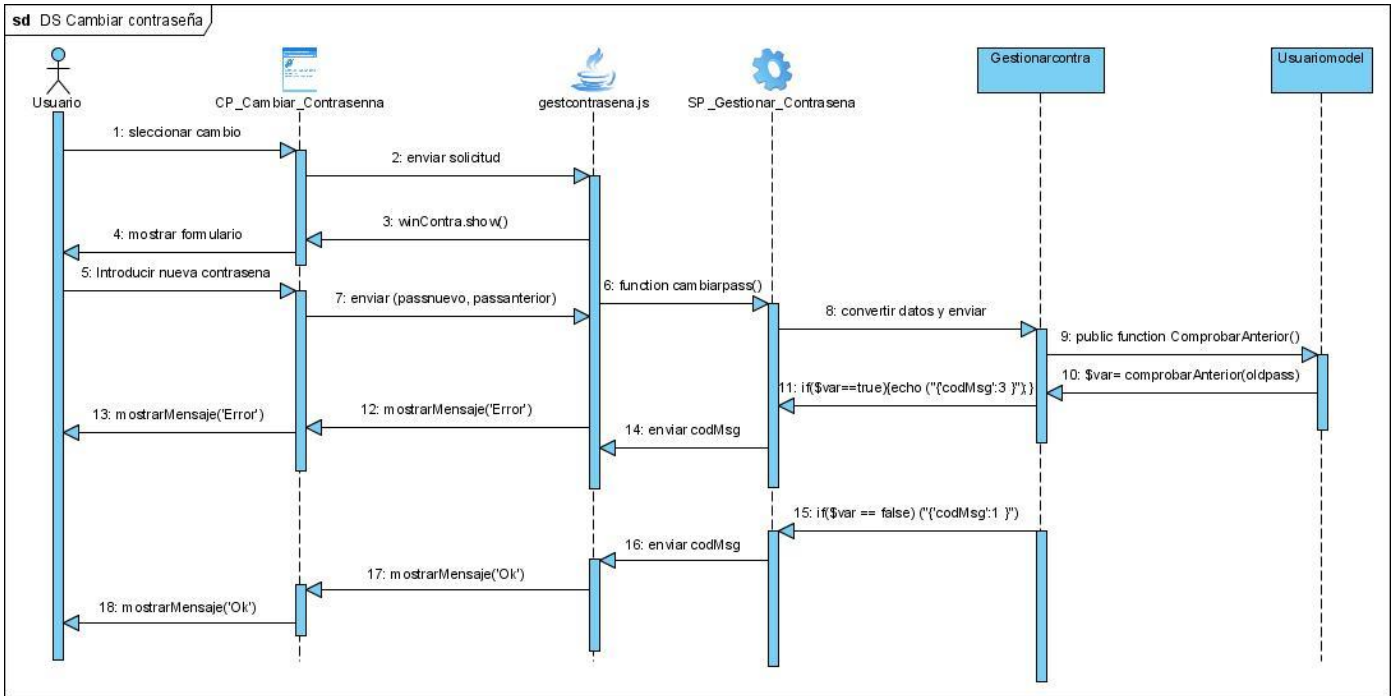


Figura 54 Diagrama de secuencia "Cambiar contraseña".

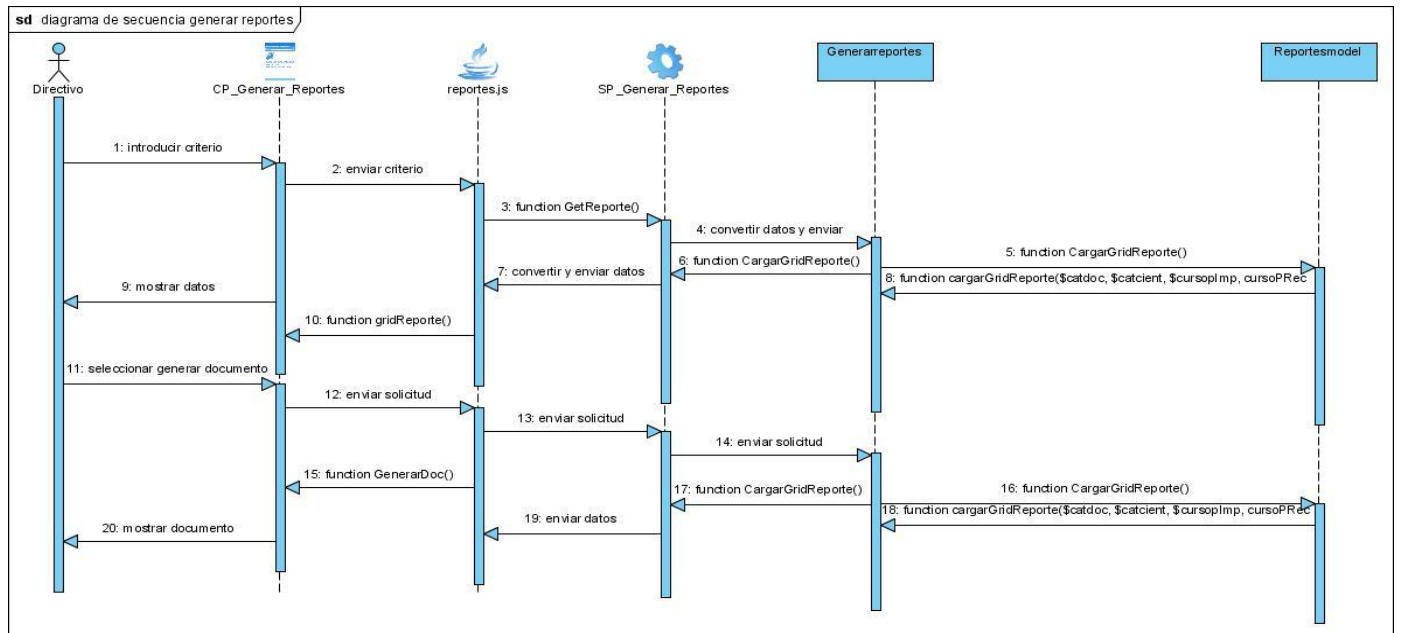


Figura 55 Diagrama de secuencia "Generar Reportes".

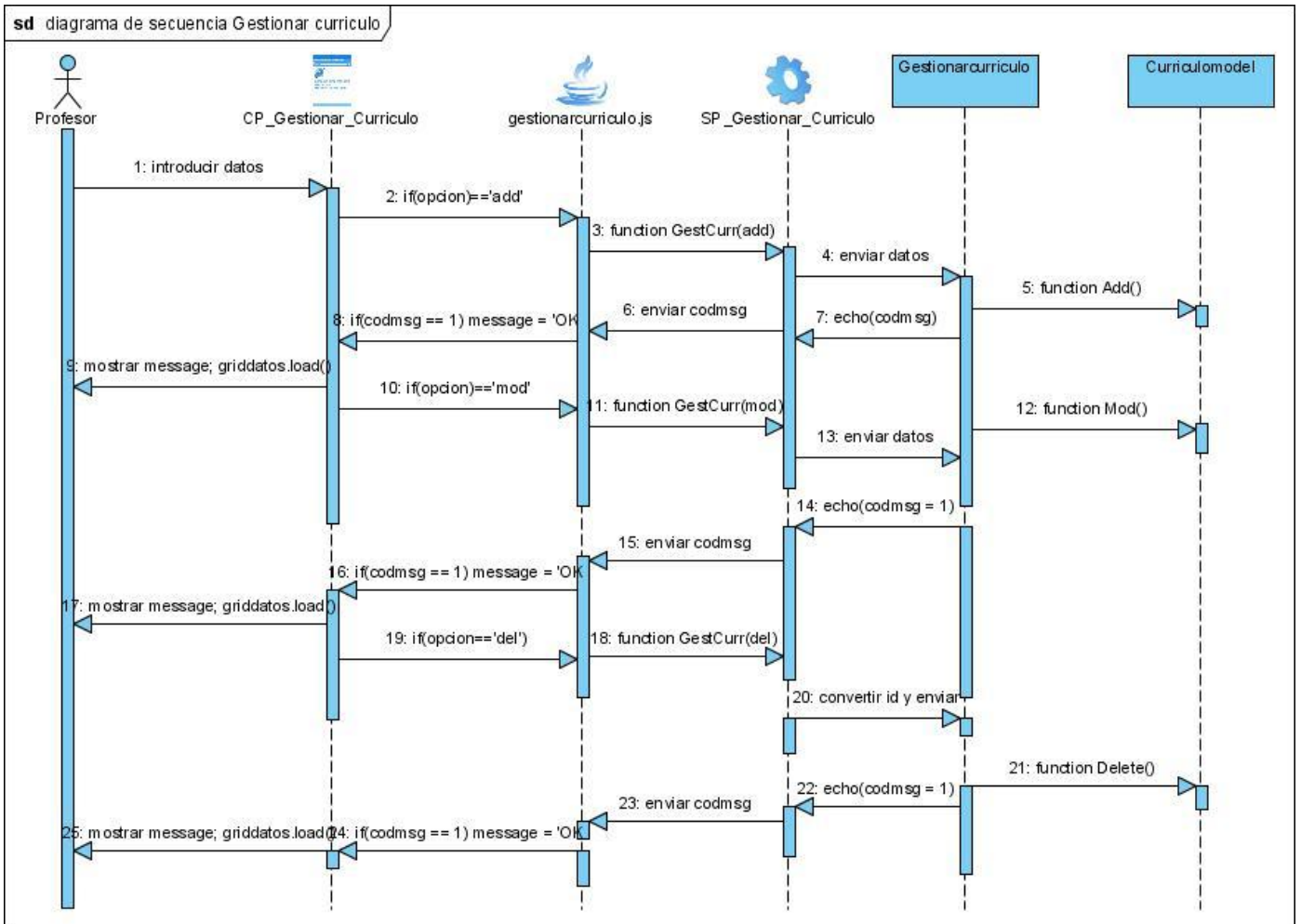


Figura 56 Diagrama de secuencia "Gestionar información del currículo".

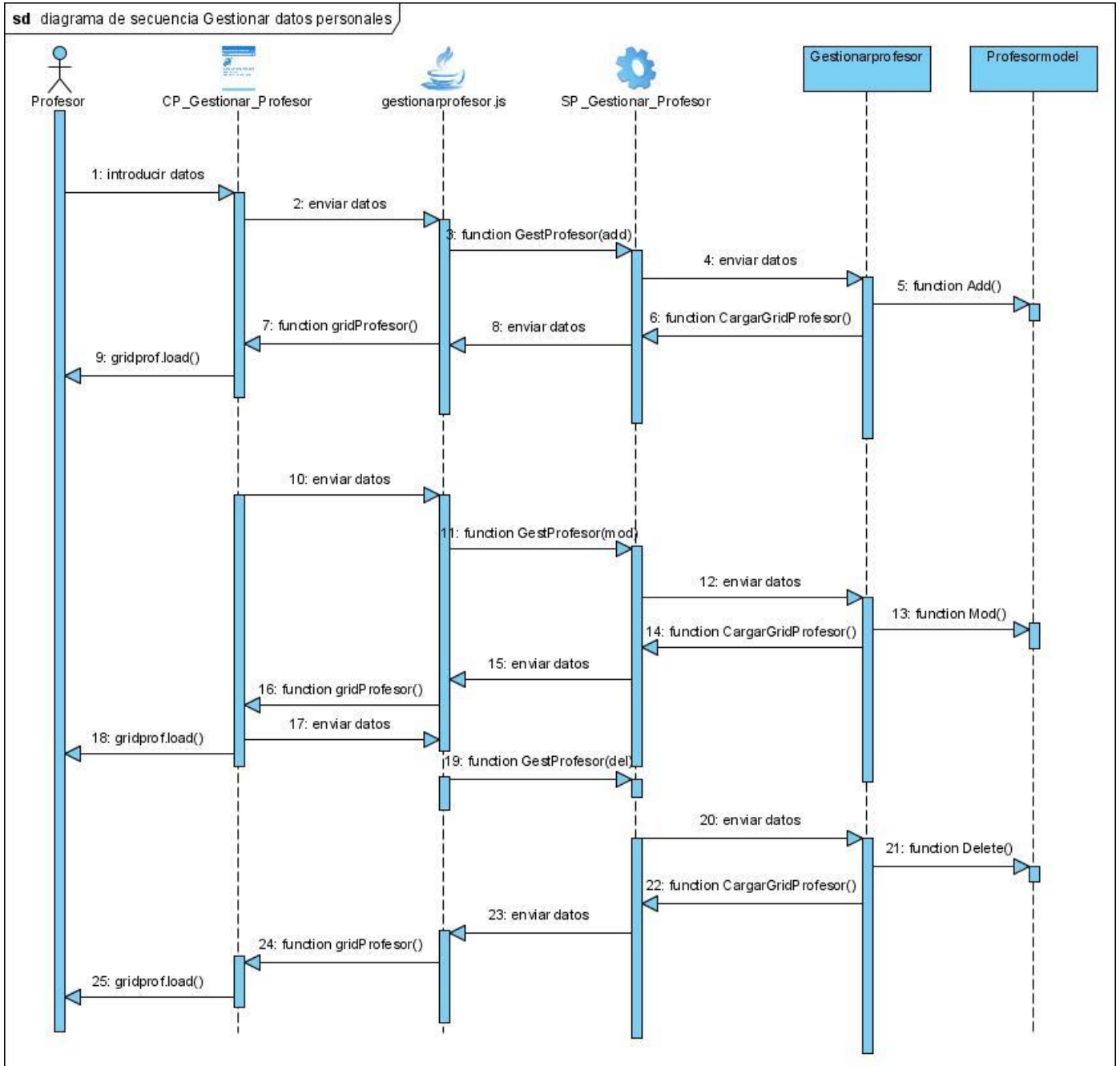


Figura 57 Diagrama de secuencia "Gestionar datos personales".

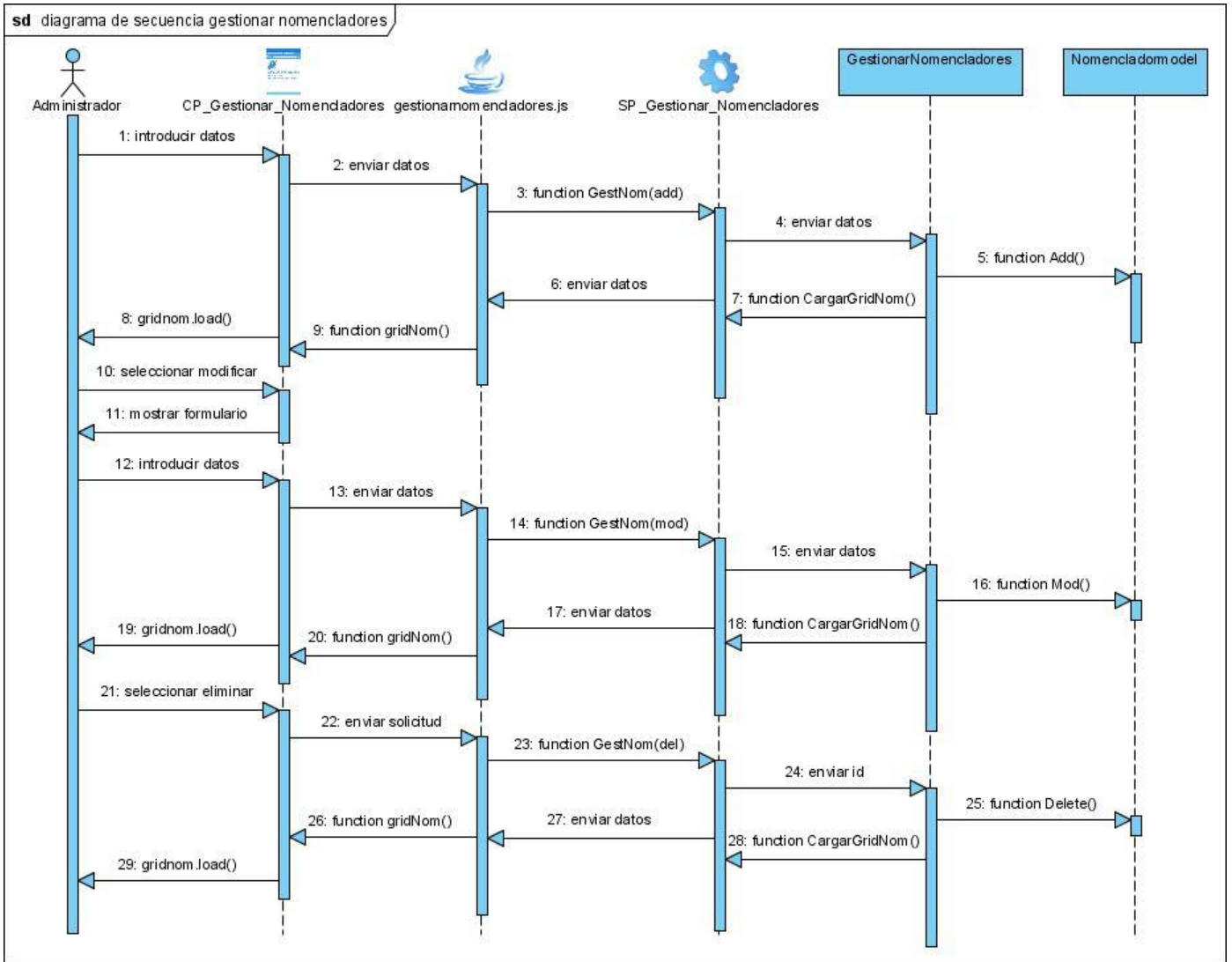


Figura 58 Diagrama de secuencia "Gestionar nomencladores".

Glosario de Términos

Aplicación Web: Es una aplicación que se ejecuta a través de internet o intranet, que contiene un negocio, y que las acciones del usuario modifican el mismo.

Currículum Vitae: Proviene del latín, y quiere decir carrera de la vida, es una acumulación de experiencias profesionales y vivenciales de una persona.

Currículo: Es la abreviación de currículum vitae.

Software libre: Es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, codificado y redistribuido libremente.

HTML: HyperText Markup Language (Lenguaje de Marcas de Hipertexto) Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con diferentes objetos. Es un lenguaje que hace posible presentar información en páginas web.

JavaScript: Es un lenguaje de programación del lado del cliente que es interpretado en el navegador.

Java: Es un lenguaje de programación independiente de plataforma desarrollado por Sun Microsystems.

PHP : Hypertext Preprocessors un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

Script: Es un programa usualmente simple, que generalmente se almacena en un archivo de texto plano.

C: Es un lenguaje de programación creado en 1972 por Kenneth L. Thompson, orientado a la implementación de Sistemas perativos.

C++: Es un lenguaje de programación imperativo orientado a objetos derivado del C.

JVM: Java Virtual Machine (Máquina Virtual de Java). Es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario.

Perl: (Practical Extraction and Report Language) Es un lenguaje de propósito general, interpretado, orientado a objeto, y enfocado al tratamiento de información y su presentación por pantalla u otro medio.

MySQL: Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis

lenguaje PHP.

Framework: Es un conjunto de funciones, procedimientos y herramientas destinadas a la construcción de un determinado tipo de aplicaciones de manera generalista, es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

Content Management System: Sistemas gestores de contenidos para aplicaciones web.

Licencia BSD: Permite el uso y distribución de PostgreSQL sin costo, tanto para aplicaciones comerciales como no comerciales.

XP (Programación Extrema): Es una metodología ágil de desarrollo de software.

SCRUM: Es una metodología ágil de desarrollo de software.

RUP (Proceso Unificado del Software): Es una metodología de desarrollo de software, pesada y orientada a objetos.

UML: Unified Modeling Language (Lenguaje Unificado de Modelado): Es un lenguaje de modelado para visualizar, especificar, construir y documentar un sistema.

Caso de uso: Es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software.

Servidor web: Es un programa que implementa el protocolo HTTP y está diseñado para transferir hipertextos, páginas web o páginas HTML, textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Apache: Es un servidor web.

Wamp: Es un entorno de desarrollo Web que contiene al servidor web Apache.

MVC: Patrón de arquitectura Modelo Vista Controlador.

Zend Studio for Eclipse: es un entorno de desarrollo que combina la tecnología Zend Studio con las herramientas de desarrollo de PHP con Eclipse.

WYSIWYG: *What You See Is What You Get.* (Lo que ves es lo que obtienes) Son tipos de herramientas que permiten escribir la página sobre una vista preliminar similar a la de un procesador de textos.

Plug-in: Es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.

IEEE: (Institute of Electrical and Electronics Engineers). El Instituto de Ingenieros Eléctricos y Electrónicos es una asociación técnico-profesional mundial dedicada a la estandarización.