

Universidad de las Ciencias Informáticas

Facultad 7



**Título: Desarrollo de una herramienta Web para la
generación dinámica de reportes en PHP**

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

Autores: Pavel Manzanares Angulo.

Roberlán Pérez Hernández.

Tutor: Lic. Ernesto Sarduy Alonso.

Ciudad de La Habana, Junio 2008

"Año 50 de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos 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 20 días del mes de Junio del año 2008.

Pavel Manzanares Angulo

Roberlán Pérez Hernández

Lic. Ernesto Sarduy Alonso

DATOS DE CONTACTO

Lic. Ernesto Sarduy Alonso ernestos@uci.cu. Graduado de Ciencias de la Computación en el año 2006, por la Universidad Central de las Villas "Martha Abreus". Comienza a trabajar en el año 2006 como profesor en la Universidad de las Ciencias Informáticas (UCI), impartiendo las asignaturas de Programación II y de Programación IV. Ha pasado numerosos cursos de postgrado en la Universidad de la Habana y en la propia UCI relacionados con su actividad profesional.

AGRADECIMIENTOS.

Agradecemos:

A nuestro Comandante Fidel y Raúl y a la Universidad de Ciencias Informáticas por hacernos partícipes de este proyecto de la Revolución.

A nuestros amigos y compañeros de grupo por brindarnos su apoyo incondicional.

A nuestro tutor de tesis Ernesto Sarduy Alonso por sus consejos, su apoyo y dedicación.

Pavel:

A mis padres Maritza y Boris, hermanos Mariela y Boris que me han dado todo su apoyo y han hecho posibles que hoy se haga realidad este sueño. A mi compañero de tesis "Refor" por haber hecho un trabajo dedicado e intachable para el desarrollo de la tesis. A mis amigos: Raciél, Edwin, Gabriel y César que son como mis hermanos y me han hecho sentir como en casa. A mis compañeros de grupo que siempre me han apoyado cuando lo he necesitado logrando que mi estancia en la UCI haya sido más amena.

Roberlán:

A mis padres, Roberto y María y abuelos, José y Consuelo: por sus preocupaciones, sacrificios y desvelos. A mi hermano Ronel que, a pesar de todo, confía en mí. A mis tíos Mayra y Fidel por tenerme presente, a mis primos. A mi esposa y su familia por su apoyo. A mis abuelos y bisabuelos "Macho", Marta, Eulogio, Isabel que aunque no están con nosotros siempre los llevo presentes. A mi compañero de tesis: Pavel. A mis amistades del preuniversitario: Liester, Osvel, Hayler, Yunier y compañeros de brigada, por los momentos buenos.

DEDICATORIA.

Pavel:

Dedico este trabajo de diploma a mi familia, en especial a mis padres y hermanos por confiar en mí y apoyarme en todos los pasos importantes que he dado en la vida. A mis amigos y demás que de una forma u otra han contribuido a que llegue este día.

Roberlán:

Dedico este trabajo de diploma a mi familia: en especial a mis padres y abuelos por guiarme por el camino correcto, a mi hermano, a mis tíos por tenerme presente a pesar de mi distancia. A mi esposa y su familia por mantener mi felicidad en este lugar distante. A mis amistades por los buenos momentos, el apoyo y la confianza.

RESUMEN

El presente trabajo tiene como objetivo desarrollar una herramienta en el lenguaje PHP que permita la generación dinámica de reportes para el área temática “Sistema de Apoyo a la Salud” en la Universidad de las Ciencias Informáticas.

La aplicación se desarrolló en el lenguaje libre PHP mediante un conjunto de herramientas como NuSphere PhpED, Dreamweaver, EMS SQL Manager y el servidor web: Apache junto al empleo de la metodología Ajax, la tecnología Dojo y el motor de Plantillas Smarty.

Con el desarrollo de la herramienta se obtuvieron logros importantes como: mayor uniformidad y dinamismo en la obtención de reportes. Así como la posibilidad del usuario de abstraerse del conocimiento sobre gestores de base de datos. Facilita la portabilidad a sistemas operativos Linux y Windows 98 o superior, al igual que a los gestores de base de datos PostGresSQL y MySQL. Actualmente se está aplicando la herramienta para generar reportes en el proyecto: *“Sistema para el control de la colaboración médica cubana”*.

PALABRAS CLAVE

Reportes, generación dinámica, herramienta, php, base de datos.

TABLA DE CONTENIDOS

AGRADECIMIENTOS.....	I
DEDICATORIA.....	II
RESUMEN.....	III
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	3
1.1 Sistemas de generación de reportes en el mundo.....	3
1.2 Sistemas de generación de reportes en Cuba.....	3
1.3 Análisis de herramientas de generación de reportes.....	3
1.4 Lenguajes de programación.....	5
1.5 Navegador recomendado: Mozilla Firefox.....	7
1.6 Servidor web: Apache.....	7
1.7 Sistemas gestores de bases de datos.....	8
1.8 Motor de plantillas: Smarty.....	10
1.9 Herramientas.....	11
1.10 Tecnologías y metodologías.....	13
1.11 Arquitectura.....	16
1.12 Metodología de desarrollo: RUP.....	17
1.13 Características e importancia del software a crear.....	18
CAPÍTULO 2: CARACTERÍSTICAS DE LA HERRAMIENTA.....	20
2.1 Objeto de estudio.....	20
2.2 Propuesta de sistema.....	22
2.3 Modelo del negocio.....	22
2.4 Especificación de los requisitos del software.....	23
2.5 Modelo del sistema.....	25
CAPÍTULO 3: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA HERRAMIENTA.....	27
3.1 Análisis.....	27
3.2 Diseño.....	31
3.3 Diagrama de despliegue.....	55
3.4 Diagramas de componentes.....	56
CONCLUSIONES.....	57
RECOMENDACIONES.....	58
BIBLIOGRAFÍA.....	59
REFERENCIAS BIBLIOGRÁFICAS.....	61
ANEXOS.....	62

Anexo 1: Modelo del negocio.....	62
Anexo 2: Modelo del sistema.....	67
Anexo 3. Aval de introducción de resultados.....	89
GLOSARIO.....	90

INTRODUCCIÓN.

Cuba, busca vías de desarrollo apoyándose en el capital humano con que cuenta. Para ello aspira a obtener resultados que beneficien de forma sustancial el trabajo en diferentes sectores de la vida económica y social del país. Uno de los retos fundamentales es lograr que los jóvenes estudiantes desarrollen habilidades que les permitan ayudar al país a insertarse en el dominio de las más modernas técnicas de computación con la menor inversión posible.

Las habilidades adquiridas durante los estudios, permiten en estos momentos, enfrentar trabajos de envergadura, que permitan no sólo su aplicación y desarrollo, sino que contribuyan al ahorro considerable de recursos que habría que invertir en la compra de programas de esta índole. Un ejemplo de ello, son las aplicaciones desarrolladas por los estudiantes de las diferentes áreas temáticas de la Universidad de las Ciencias Informáticas. En el caso del área temática “Sistema de Apoyo a la Salud” (SAS), fue diseñada para apoyar los diferentes sistemas que se desarrollan para el Ministerio de Salud Pública y así garantizar una mayor organización de este sector de la sociedad.

Aunque en todo sistema de gestión de información es de vital importancia la generación de reportes. Actualmente en esta área temática, al igual que en diversos sistemas a nivel mundial, existen problemas a la hora de brindar la información necesaria. Esta carece de dinamismo, las herramientas que permiten el trabajo con la información elaboran estrictamente un tipo de reporte determinado, obligando al usuario a redefinir los requisitos cada vez que necesite añadir uno nuevo.

En el área temática “Sistema de Apoyo a la Salud” (SAS), se desarrollan aplicaciones web que en su mayoría generan reportes. La generación de los mismos se implementa por separado en los diferentes módulos involucrados en el proceso provocando: falta de uniformidad, pérdida de tiempo y falta de dinamismo. Esto ocurre debido a que los actuales reportes utilizan una base de datos predeterminada y sólo se incluyen los atributos previamente establecidos, lo que evidencia la dependencia entre el reporte y el código necesario para generarlo. Estas características traen consigo que cada aplicación tenga un número fijo de reportes a generar. Por tanto, si existe la necesidad de agregar uno nuevo, debe ser programado por el equipo de desarrolladores, en lugar de permitirse que el usuario sea quien lo elabore.

Problema a resolver.

¿Cómo generar de forma dinámica los reportes de los proyectos o sistemas que se desarrollan en el área temática “Sistema de Apoyo a la Salud”?

Objeto de estudio.

Proceso de generación dinámica de reportes para los diversos software de la Universidad de las Ciencias Informáticas.

Campo de acción.

Proceso de generación dinámica de reportes para los diversos software del área temática "Sistema de Apoyo a la Salud"

Objetivo general.

Desarrollar una herramienta en el lenguaje PHP que permita la generación dinámica de reportes para el área temática "Sistema de Apoyo a la Salud".

Tareas propuestas.

Para lograr el cumplimiento satisfactorio de este objetivo se proponen las siguientes tareas:

- ✍ Evaluar las herramientas que realizan funciones análogas en otros sistemas.
- ✍ Aplicar el paradigma orientado a objetos.
- ✍ Obtener los artefactos propios de la fase de análisis en el desarrollo de la herramienta.
- ✍ Obtener los artefactos propios de la fase de diseño en el desarrollo de la herramienta.
- ✍ Diseñar la interfaz gráfica utilizando herramientas acordes con las exigencias del área temática en el desarrollo de aplicaciones Web.
- ✍ Desarrollar un producto en PHP para la generación dinámica de reportes.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se describen los sistemas de generación de reportes en Cuba y en el resto del mundo. Se analizan varias herramientas de generación de reportes. Se caracterizan las tecnologías, lenguajes y herramientas empleados en el desarrollo de la aplicación. Se describe la herramienta, así como se valora la importancia de la misma.

1.1 Sistemas de generación de reportes en el mundo.

Actualmente, en diversos sistemas de generación de reportes a nivel mundial existen problemas con el dinamismo a la hora de brindar la información necesaria. Las herramientas que permiten este trabajo son elaboradas estrictamente para generar un tipo de reporte determinado, lo cual obliga al usuario a redefinir los requisitos cada vez que necesite añadir uno nuevo. Existen múltiples herramientas en el mundo asociadas a este problema, pero en su mayoría tienen un elevado costo de adquisición y son por lo general, orientadas al programador y no al usuario.

1.2 Sistemas de generación de reportes en Cuba.

Cuba, se encuentra en pleno proceso de desarrollo en el campo de la informática y las telecomunicaciones. Para desarrollarlo ha creado un conjunto de instituciones entre ellas la Universidad de las Ciencias Informáticas, que con sólo 5 años de fundada, ha logrado informatizar importantes áreas que inciden de manera trascendental en la calidad de vida de la población, como la medicina y la educación.

A pesar de estos importantes logros, aún no se cuenta con una herramienta que posibilite al usuario explotar la información que posee almacenada en su base de datos, de una manera ágil y sencilla.

1.3 Análisis de herramientas de generación de reportes.

Para lograr una mejor visión y conocimiento del estado del arte en el tema tratado, se realizó un análisis de otras Herramientas de generación de reportes como es el caso de PHP Reports, Crystal Reports y Jasper Report.

Crystal Reports.

Crystal Reports es una herramienta que ha formado parte de Visual Studio desde 1993, y ahora es el estándar de creación de informes de Visual Studio 2005, se incluye en todas sus copias y se integra directamente esta plataforma. Además, incorpora la posibilidad de crear contenido interactivo con calidad de presentación al entorno de Windows. Brinda al usuario, entre otras ventajas, la posibilidad de elegir el tipo de base de datos en que desea trabajar y facilita los cambios en el diseño, es decir, a la hora de editar el informe, en la ventana de vista previa, se pueden cambiar formatos, insertar o eliminar objetos, moverlos, etc.

Esta herramienta sólo utiliza una jerarquía para agrupar los datos, o sea, clasifica los datos según la jerarquía otorgada y así define la estructura de todo el informe. Como consecuencia, si se desea generar reportes con estructura compleja, el usuario está obligado a crear varios subreportes. Además, elabora los reportes en tiempo de diseño ya que está dirigida más bien para el programador y no para el usuario.

Akademos.

La aplicación Akademos, desarrollada en la Universidad de las Ciencias Informáticas en el año 2006, brinda en la actualidad importantes servicios académicos tanto a los departamentos docentes como a los estudiantes. Cuenta con una herramienta de generación de reportes, la cual los genera con eficiencia, pero desde el punto de vista del dinamismo de la generación, se encuentra adaptada solamente al negocio de la gestión académica en la UCI, restringiendo que los reportes sean elaborados con la información contenida en la base de datos específicamente utilizada por el software, lo cual dificulta que esta herramienta sea de propósito general.

Jasper Report

Jasper Report [1]: es una herramienta para la generación de informes. Está implementada en Java y es libre. El funcionamiento consiste en escribir un XML donde se recogen las particularidades del informe. El mismo es tratado por las clases del Jasper y es exportado a otros formatos como: PDF, XML, HTML, XLS o TXT. Jasper Report se integra perfectamente con el JFreeChart que es una librería libre para la generación de todo tipo de graficas. Para generar el XML es recomendable el uso de la

herramienta visual iReport que es un editor gráfico que también está implementado en Java y se integra perfectamente con el Jasper Report.

Para generar un reporte con Jasper Report se emplean una serie de pasos más bien orientados al programador y no al cliente, ya que tiene que trabajar con distintos tipos de ficheros los cuales se muestran a continuación:

- ✍ Generar un fichero .jrxml en el que se configura cómo se quiere el informe
- ✍ Compilar el fichero .jrxml para obtener un fichero .Jasper
- ✍ Rellenar los datos del informe. Esto generará un fichero .jrprint
- ✍ Exportar el fichero .jrprint al formato que se necesite (pdf, etc.). Esto generará el fichero en cuestión.

1.4 Lenguajes de programación.

Lenguaje por parte del cliente.

HTML [2]: (**H**yper**T**ext **M**arkup **L**anguage) Lenguaje de Marcas Hipertextuales. Permite la marcación, fue diseñado con el objetivo de estructurar textos y mostrarlos en el formato estándar de las páginas web o sea hipertexto. Es compatible con navegadores reconocidos como: Internet Explorer, Opera, Firefox, Netscape o Safari. Se ha convertido en uno de los formatos con mayor popularidad y facilidad de aprendizaje que existen en la elaboración de documentos para web.

JavaScript [3]: es un lenguaje que no requiere de la compilación, muy utilizado en páginas web y posee una sintaxis similar a la de los lenguajes Java y C. No se considera un lenguaje orientado a objetos por no disponer de herencia; más bien se basa en prototipos, donde las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

El código JavaScript que se encuentra dentro de las páginas web puede ser interpretado por todos los navegadores. Permite crear páginas web interactivas (DHTML) con relativa facilidad y se utiliza tanto para programación en el lado del cliente (navegadores fundamentalmente).

La extensión para cliente permite acceder a los objetos que utiliza el navegador y el DOM (Document Object Model), la jerarquía de objetos de un documento HTML. Añade soporte para control de eventos, de tal forma que el programa puede interactuar con el usuario.

Lenguaje para la transmisión de datos cliente-servidor: XML.

XML [4]: son las siglas en ingles de Extensible Markup Language (lenguaje de marcas extensible), constituye una mezcla de especificación y lenguaje de programación, desarrollada por el W3C. Es una versión de SGML, diseñado especialmente para los documentos de la web. Permite que los diseñadores creen sus propias etiquetas, posibilitando la definición, transmisión, validación e interpretación de datos entre aplicaciones y organizaciones.

XML posee varias ventajas con respecto a otros lenguajes como:

- ✍ Es extensible, lo que quiere decir que una vez diseñado un lenguaje y puesto en producción, igual es posible extenderlo con la adición de nuevas etiquetas de manera de que los antiguos consumidores de la vieja versión todavía puedan entender el nuevo formato.
- ✍ El analizador es un componente estándar, no es necesario crear un analizador específico para cada lenguaje. Esto posibilita el empleo de uno de los tantos disponibles. De esta manera se evitan errores y se acelera el desarrollo de la aplicación.
- ✍ Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarlo. Mejora la compatibilidad entre aplicaciones.

Lenguaje por parte del servidor.

En la implementación de esta herramienta se decidió utilizar la tecnología montada en la plataforma de desarrollo PHP con el propósito de coincidir con las herramientas que la mayoría de los proyectos desarrollados en el área temática SAS emplean. Lo que permitió aprovechar sus ventajas y la experiencia del personal.

PHP [5]: es un lenguaje de script interpretado en el lado del servidor, se utiliza en la generación de páginas Web dinámicas; es similar al ASP de Microsoft o el JSP de Sun, está embebido en páginas HTML y se ejecuta en el servidor. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl aunque tiene algunas características específicas. Su meta es permitir a los desarrolladores la rápida creación de páginas web dinámicas. Es un potente lenguaje, que se interpreta, tanto en el servidor Web como módulo o ejecutado como un binario CGI; puede acceder a ficheros, ejecutar comandos y abrir comunicaciones de red en el servidor.

Este lenguaje libre tiene características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas:

- ✍ Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- ✍ Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- ✍ Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- ✍ Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- ✍ El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- ✍ Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

1.5 Navegador recomendado: Mozilla Firefox.

Mozilla Firefox [6]: es el navegador web recomendado para la realización de pruebas y posterior despliegue de la propuesta desarrollada ya que es un programa multiplataforma, que está disponible en versiones para Microsoft Windows, Mac OS X y GNU/Linux. El código ha sido portado por terceros a FreeBSD, OS/2, Solaris, SkyOS, BeOS y más recientemente, Windows XP Professional x64 Edition. El código fuente de Firefox está disponible libremente, bajo la triple licencia de Mozilla como un programa libre y de código abierto.

1.6 Servidor web: Apache. [7] [8] [9]

Un **servidor Web** es un programa que permite acceder a páginas Web contenidas en un ordenador. El Servidor Apache: es un sistema de código abierto para plataformas Windows, Unix, Macintosh y otras que implementa el protocolo HTTP. Este servidor posee varias características que lo favorecen como por ejemplo:

- ✍ Es una tecnología gratuita de código fuente abierta, lo cual le atribuye transparencia a este.
- ✍ Corre en varios Sistemas Operativos, lo que provoca que sea una herramienta prácticamente universal.

- ✍ Es un servidor altamente configurable de diseño modular. Permite aumentar fácilmente su capacidad e instalar cualquier módulo para cumplir una función específica. Otra cosa importante es que cualquiera que posea experiencia en la programación de C o Perl puede escribir un módulo para realizar una acción determinada.
- ✍ Permite personalizar la respuesta ante posibles errores. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- ✍ Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log facilitando de este modo el control de las acciones realizadas en el servidor.

1.7 Sistemas gestores de bases de datos.

Los Sistemas de Gestión de Base de Datos [10]: son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. El propósito general de los SGBD es manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información. Existen distintos objetivos que deben cumplir los SGBD:

- ✍ Abstracción de la información. Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
- ✍ Independencia. La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- ✍ Redundancia mínima. Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.
- ✍ Consistencia. En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- ✍ Seguridad. Los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

- ✍ Integridad. Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- ✍ Respaldo y recuperación. Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
- ✍ Control de la concurrencia. En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos y en muchos casos estos accesos ocurren de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.
- ✍ Tiempo de respuesta. Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.

MySQL.

MySQL [11]: es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Posee varias ventajas como:

- ✍ Soporta la mayoría de los comandos del lenguaje SQL (structured query language), el estándar en bases de datos.
- ✍ Acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones.
- ✍ Seguridad: En forma de permisos y privilegios, determinados usuarios tendrán permiso para consulta o modificación de determinadas tablas. Esto permite compartir datos sin que peligre la integridad de la base de datos o protegiendo determinados contenidos.
- ✍ Portabilidad: SQL es también un lenguaje estandarizado, de modo que las consultas hechas usando SQL son fácilmente portables a otros sistemas y plataformas
- ✍ Escalabilidad: es posible manipular bases de datos enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros, y hasta 32 índices por tabla.
- ✍ MySQL está escrito en C y C++ y probado con multitud de compiladores y dispone de APIs para muchas plataformas diferentes.
- ✍ Conectividad: es decir, permite conexiones entre diferentes máquinas con distintos sistemas operativos. Es corriente que servidores Linux o Unix, usando MySQL, sirvan datos para

ordenadores con Windows, Linux, Solaris, etc. Para ello se usa TCP/IP, tuberías, o sockets Unix.

- ✍ Es multihilo, con lo que puede beneficiarse de sistemas multiprocesador.
- ✍ Permite manejar multitud de tipos para columnas.
- ✍ Permite manejar registros de longitud fija o variable.

PostgreSQL.

PostgreSQL [12]: es un sistema de gestión de base de datos relacional orientada a objetos de software libre. Posee varias ventajas como:

- ✍ Ágil navegación y administración de base de datos.
- ✍ Administración sencilla de todos los objetos PostgreSQL.
- ✍ Herramientas de manipulación de datos avanzada.
- ✍ Administración efectiva de seguridad.
- ✍ Excelentes Herramientas visuales y de texto para elaboración de consultas.
- ✍ Acceso al servidor PostgreSQL a través del protocolo HTTP.
- ✍ Conexión vía reenvío de puerto local a través del túnel SSH.
- ✍ Impresionantes opciones de exportación e importación de datos.
- ✍ Poderoso diseñador visual de base de datos.
- ✍ Asistentes fáciles de usar que realizan mantenimiento de tareas de PostgreSQL.

1.8 Motor de plantillas: Smarty.

Smarty [13]: es un motor de plantillas para PHP utilizado para separar el código PHP como capa lógica de negocios de la capa de presentación (HTML/ CSS), es capaz de generar contenidos web mediante la colocación de etiquetas Smarty en un documento, es un software que puede ser utilizado libremente. Es común que en grandes proyectos el rol de diseñador gráfico y el de programador sean cubiertos por personas distintas, sin embargo la programación en PHP tiene la tendencia a combinar estas dos labores en una persona lo que trae consigo grandes dificultades, a la hora de cambiar alguna parte del diseño de la página, pues se tiene que escarbar entre los scripts para modificar la presentación del contenido, Smarty tiene como objetivo solucionar este problema, para ello ofrece un gran número de funcionalidades que permite programar plantillas de forma más organizada, ejemplo de estas son :

- ✍ Expresiones regulares.

- ✍ Bucles, foreach, while.
- ✍ if, elseif, else.
- ✍ Modificadores de variables - Por ejemplo `{ $variable|nl2br }`.
- ✍ Funciones creadas por el usuario.
- ✍ Evaluación de expresiones matemáticas en la plantilla.

1.9 Herramientas.

Rational Rose.

Rational Rose Enterprise Edition [14]: es una herramienta desarrollada por la empresa Rational. Posee paquetes dedicados a todo tipo de situaciones desde elementos para el desarrollo de software a través de UML, la captura de requisitos, hasta el uso de algoritmos para la eficiencia en el desarrollo de sistemas complejos mediante el control del personal. Rational Rose documenta la construcción del diagrama y genera código en la opción del diseñador de C++, de Visual Basic, de Java, de Oracle, de CORBA o del lenguaje de definición de datos.

Dreamweaver 8.

Dreamweaver [15]: es una aplicación para el diseño y programación de páginas web. Es la plataforma de trabajo más utilizada en el sector del diseño y la programación web, debido a sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium.

Tiene facilidades tanto para edición de imágenes como para animación a través de su integración con otras herramientas. Hasta la versión MX, fue duramente criticado por su escaso soporte de los estándares de la web, ya que el código que generaba era con frecuencia sólo válido para Internet Explorer, y no validaba como HTML estándar. Esto se corrigió en la versión 8.

NuSphere PhpED Professional 5.2.

NuSphere PhpED [16]: es un famoso IDE para PHP, HTML, CSS, XML, SMARTY, XHTML y otros. Posee: una combinación equilibrada en la edición de código avanzado y un depurador: fiable,

productivo y rápido en la conectividad de bases de datos, que se ajusta a las necesidades de los desarrolladores. El navegador y explorador de códigos constituyen las principales herramientas para ayudar a los desarrolladores en la escritura de código.

Está diseñado para funcionar con cualquier archivo PHP abierto, independientemente del proyecto actual. No sólo se ejecuta rápido sino que también se puede ejecutar en 18 plataformas: windows/x86, windows/64bit, FreeBSD 6/x86, 6/x86_64 FreeBSD, OpenBSD 4/x86, 4/x86_64 OpenBSD, NetBSD 3/x86, NetBSD 3/x86_64 , Solaris 10/x86, solaris 10/x86_64, Solaris 8 (y superior) / sparc, Solaris 8 (y superior) / UltraSparc, FreeBSD 4/x86, 2.4/x86 linux, linux 2.6/x86, 2.6/x86_64 linux, macosx / x86, macosx / powerpc.

EMS SQL Manager para MySQL.

EMS MySQL Manager [17]: proporciona un completo conjunto de eficaces y potentes herramientas para administrar un servidor MySQL.

A través de su clara interfaz gráfica se puede crear y editar parámetros de una base de datos de forma sencilla. Ofrece la posibilidad de otorgar y administrar privilegios de usuarios, ejecutar scripts SQL, consultas visuales integradas, extraer o imprimir metadatos, importar y exportar datos, etc.

Cambios recientes de EMS MySQL Manager:

- ✍ Soporte completo de MySQL 5.0.
- ✍ Mejorada la interfaz de usuario
- ✍ Diseñador visual de bases de datos
- ✍ Herramientas de gestión de datos
- ✍ Túneles HTTP.
- ✍ Soporte de juegos de caracteres de conexión y de campos
- ✍ Servicio nuevo de cortar tablas
- ✍ Localización romana

EMS SQL Manager para PostgreSQL.

EMS SQL Manager para PostgreSQL [18]: es una poderosa herramienta gráfica para la administración y, de manera general, para el trabajo con servidores PostgreSQL. Funciona con cualquier versión de

PostgreSQL y soporta todas las nuevas características de PostgreSQL incluyendo espacios de tablas, argumentos nombrados en funciones y otras más. Ofrece una gran variedad de herramientas poderosas para usuarios avanzados, tales como diseñador visual de base de datos, constructor visual de consultas y un poderoso editor de objetos binarios para satisfacer todas sus necesidades. PostgreSQL Manager cuenta con una nueva y avanzada interfaz gráfica de usuario con un sistema asistente bastante descriptivo, tan claro en su uso que ni un principiante podría confundirse.

Características principales:

- ✍ Soporte completo de PostgreSQL.
- ✍ Nueva interfaz gráfica de usuario último modelo.
- ✍ Ágil navegación y administración de base de datos.
- ✍ Administración sencilla de todos los objetos PostgreSQL.
- ✍ Herramientas de manipulación de datos avanzada.
- ✍ Administración efectiva de seguridad.
- ✍ Excelentes Herramientas visuales y de texto para elaboración de consultas.
- ✍ Acceso al servidor PostgreSQL a través del protocolo HTTP.
- ✍ Impresionantes opciones de exportación e importación de datos.
- ✍ Poderoso diseñador visual de base de datos.
- ✍ Asistentes fáciles de usar que realizan mantenimiento de tareas de PostgreSQL.

1.10 Tecnologías y metodologías.

Dojo.

Dojo [19]: es más que un conjunto de herramientas, es un Framework que contiene APIs (Applications Program Interface) y widgets (controles) para soportar el desarrollo de aplicaciones Web. Contiene un sistema de empaquetado inteligente, drag and drop APIs, widget APIs, abstracción de eventos, almacenamiento de APIs en el cliente, e interacción de APIs con AJAX. Dojo resuelve asuntos de usabilidad comunes como la navegación y detección del navegador, soportar cambios de URLs para luego regresar a ellas (bookmarking), y la habilidad de degradar cuando AJAX/JavaScript no es completamente soportado en el cliente. Dojo es llamado “la navaja suiza del ejército de las bibliotecas JavaScript”. Proporciona una gama más amplia de opciones en una sola biblioteca y hace un trabajo muy bueno que apoya los nuevos y viejos navegadores.

Características:

1. Múltiples puntos de entrada.
2. Independencia del intérprete.
3. Unifica estándares de codificación.

Ajax.

AJAX (**A**synchronous **J**avaScript **A**nd **X**ML) [20]: es una de las técnicas emergentes para el sector empresarial con mayor impacto en el desarrollo de sistemas interactivos. La misma realiza una función de intermediario entre el servidor y el usuario, anticipando peticiones de datos al servidor de modo que, cuando el usuario efectúa una consulta determinada, AJAX ya tiene listos el resultado y lo muestra directamente, sin necesidad de volver a hacer una petición al servidor y esperar por el resultado. Estas acciones se llevan a cabo a través del uso de pequeñas rutinas escritas en JavaScript que pueden transferir datos entre el navegador y el servidor de manera transparente a los usuarios sin recargar la página.

A continuación la figura 1.1 muestra una imagen donde se puede apreciar la influencia del Ajax en la relación cliente servidor.

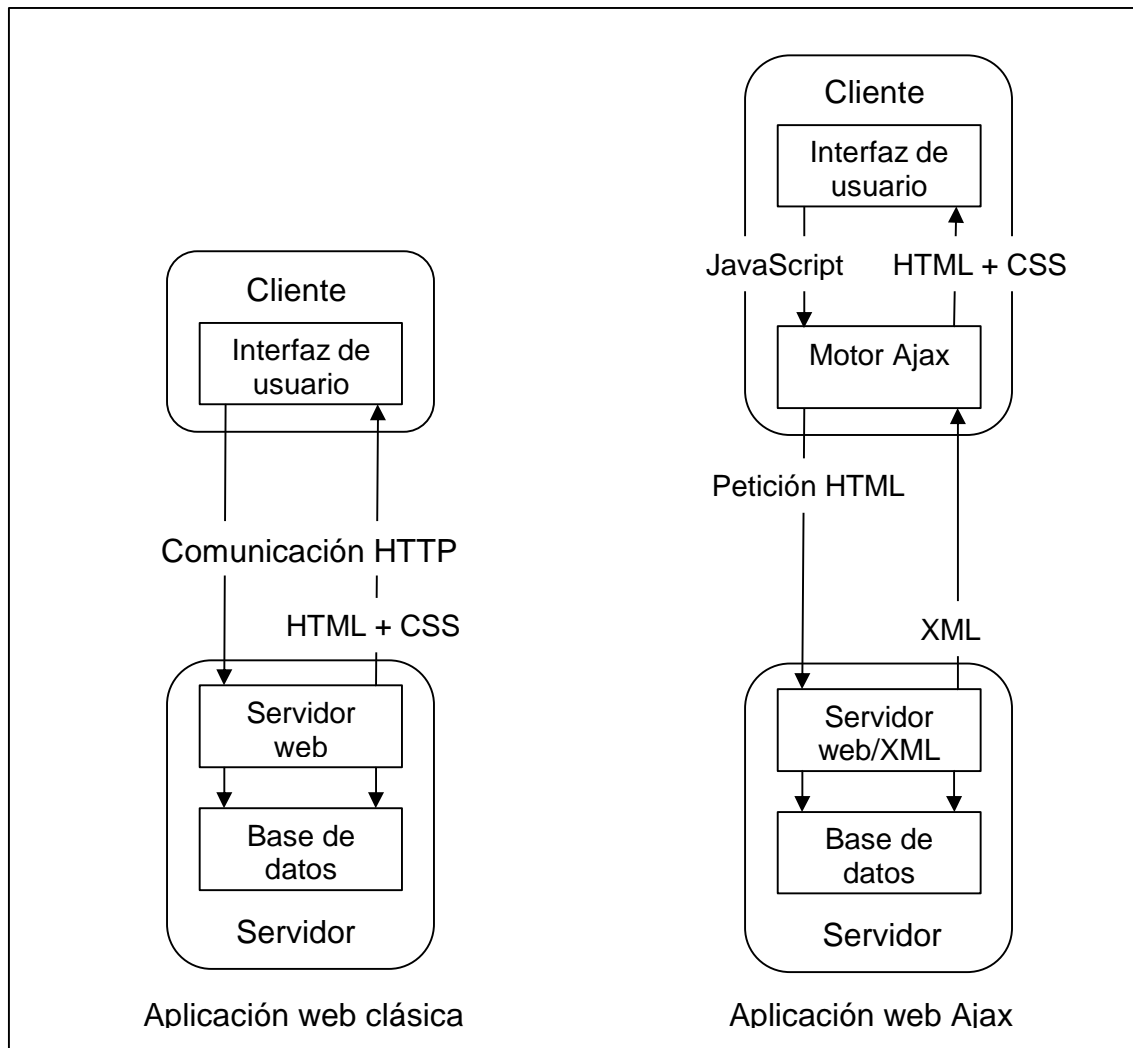


Figura 1.1 Ventajas de la aplicación Ajax.

XSLT o transformaciones XSL

XSLT o Transformaciones XSL [21]: es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML. Las hojas de estilo XSLT (aunque el termino de hojas de estilo no se aplica sobre la función directa del XSLT) realizan la transformación del documento utilizando una o varias reglas de plantilla: unidas al documento fuente a transformar, esas reglas de plantilla alimentan a un procesador de XSLT, el cual realiza las transformaciones deseadas colocando el resultado en un archivo de salida o, como en el caso de una página web, directamente en un dispositivo de presentación, como el monitor de un usuario.

Actualmente, XSLT es un lenguaje funcional muy usado en la edición web, generando páginas HTML o XHTML. La unión de XML y XSLT permite separar contenido y presentación, aumentando así la productividad.

1.11 Arquitectura.

Para seleccionar la arquitectura a utilizar por la aplicación se tuvo en cuenta que la misma sea la apropiada para las aplicaciones web, así como que sea adaptable a las tecnologías proporcionadas por la plataforma PHP y tenga aceptación en el área temática “Sistema de Apoyo a Salud”, se consideró que la arquitectura definida por tres capas cumple con las condiciones planteadas. También se trató de lograr que todo el desarrollo esté soportado sobre software libre, para que la herramienta no herede ningún tipo de limitación en lo relativo a la soberanía tecnológica.

Arquitectura en tres capas.

La arquitectura propuesta para el desarrollo de la aplicación integra tres capas bien definidas las cuales son:

- ✍ Capa de presentación.
- ✍ Capa de lógica de negocio.
- ✍ Capa de acceso a datos.

Capa de presentación:

La capa de presentación contiene una serie de páginas Web que son mostradas de forma dinámica al usuario, siendo para este simples páginas HTML. Existen distintos frameworks que generan estas páginas Web a partir de distintos formatos, en este caso se decidió utilizar Macromedia Dreamweaver 8 que, mediante un conjunto de etiquetas, proporciona una interfaz adecuada a Script PHP propiciados por el servidor de la aplicación.

Capa de negocio:

La lógica de negocio, modelada por un conjunto de script basados en JavaScript, PHP y AJAX, consta de funciones que permiten trabajar con el tratamiento de datos y otras funcionalidades como: validación de entrada de datos, verificación del sistema operativo, conexión a base de datos.

Capa de acceso a datos.

Para controlar el acceso a datos se utilizó, al igual que en la capa de negocio el NuSphere PHP que permite la creación de scripts en PHP para receptor la información y mediante validaciones en JavaScript elaborar consultas que interactúan con la base de datos del cliente.

1.12 Metodología de desarrollo: RUP.

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:

- ✍ Inicio: El Objetivo en esta etapa es determinar la visión del proyecto.
- ✍ Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.
- ✍ Construcción: En esta etapa el objetivo es llegar a obtener la capacidad operacional inicial.
- ✍ Transmisión: El objetivo es llegar a obtener el realce del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, el cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Los elementos del RUP son:

- ✍ Actividades: Son los procesos que se llegan a determinar en cada iteración.
- ✍ Trabajadores: Vienen hacer las personas o entes involucrados en cada proceso.
- ✍ Artefactos: Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

El RUP presenta 3 características que constituyen la esencia de todo el proceso de desarrollo:

- ✍ Dirigido por los casos de uso.
- ✍ Centrado en la arquitectura.
- ✍ Ciclo de vida iterativo.

Ventajas de RUP

El proceso de desarrollo RUP aplica varias de las mejores prácticas en el desarrollo moderno de software en una forma que se adapta a un amplio rango de proyectos y de organizaciones. Provee a cada miembro del equipo, un fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas del desarrollo de software. Esta metodología permite que todos los integrantes de un equipo de trabajo, conozcan y compartan el proceso de desarrollo, una base de conocimientos y los distintos modelos de cómo desarrollar el software utilizando un lenguaje de modelado común: UML.

Otras características o ventajas de la aplicación de esta metodología son las siguientes:

- ✍ Reconoce que las necesidades del usuario y sus requerimientos no se pueden definir completamente al principio.
- ✍ Permite evaluar tempranamente los riesgos en lugar de descubrir problemas en la integración final del sistema.
- ✍ Reduce el costo del riesgo a los costos de un solo incremento.
- ✍ Acelera el ritmo del esfuerzo de desarrollo en su totalidad debido a que los desarrolladores trabajan para obtener resultados claros a corto plazo.
- ✍ Distribuye la carga de trabajo a lo largo del tiempo del proyecto ya que todas las disciplinas colaboran en cada iteración.
- ✍ Facilita la reutilización del código teniendo en cuenta que se realizan revisiones en las primeras iteraciones lo cual además permite que se aprecien oportunidades de mejoras en el diseño.

1.13 Características e importancia del software a crear.

La propuesta posee gran flexibilidad y permite al usuario abstraerse de los conocimientos relacionados con los gestores de bases de datos, pues el mismo solo trabajará con un listado de variables con las cuales se encuentra familiarizado (debe existir un usuario administrador que creará alias de los nombres de los campos de la bases de datos para facilitar la comprensión de los mismos por parte de el usuario común). Además, tendrá la libertad de prescindir de la realización de consultas para elaborar

un reporte (las consultas serán elaboradas por la herramienta a partir de la información introducida por el usuario).

La aplicación es portable en el sentido, de que puede ser ejecutada en Linux, Windows 98 o superior. El usuario podrá confeccionar los reportes de forma dinámica, es decir, tendrá la libertad de: escoger el gestor de base de datos (MySQL ó PostGreSQL), agrupar los datos del reporte, establecer un orden jerárquico según su conveniencia, así como introducir criterios para un atributo seleccionado. Esta herramienta puede significar un paso de avance para el área temática “Sistema de Apoyo a la Salud” ya que, además de las cualidades descritas anteriormente, se logrará estandarizar los reportes de dicha área y los programadores de la misma no tendrán que implementar la generación de los reportes.

En este capítulo se realizó un estudio de los sistemas de generación de reportes en el mundo y en Cuba. Se valoró la importancia del software a crear, fundamentando la necesidad de su desarrollo. Se caracterizaron las principales tecnologías, metodologías, herramientas y lenguajes más utilizados y se determinaron los que son empleados en el desarrollo de la aplicación.

CAPÍTULO 2: CARACTERÍSTICAS DE LA HERRAMIENTA

En este capítulo se hace una propuesta de la herramienta a desarrollar “RPHP” basada en el problema planteado. Se analiza el negocio, se obtiene el Modelo del Negocio y las descripciones de los casos de uso del negocio. Se definen los requerimientos funcionales y los no funcionales, así como se identifican los actores, trabajadores y casos de uso que intervienen en el Negocio y en el Sistema. Se muestra un prototipo de interfaz de usuario así como una serie de tablas y diagramas para la comprensión del Negocio y del Sistema.

2.1 Objeto de estudio.

Problema y situación problemática.

Problema.

¿Cómo generar de forma dinámica los reportes de los proyectos o sistemas que se desarrollan en el área temática “Sistema de Apoyo a la Salud”?

Situación problemática.

En el área temática “Sistema de Apoyo a la Salud” (SAS), se desarrollan aplicaciones web que en su mayoría generan reportes. La generación de los mismos se implementa por separado en los diferentes módulos involucrados en este proceso, provocando falta de uniformidad, pérdida de tiempo, falta de dinamismo, ya que los actuales reportes utilizan una base de datos predeterminada y sólo se incluyen en los mismos los atributos previamente establecidos, lo que evidencia la dependencia entre el reporte y el código necesario para generarlo; estas características traen consigo que cada aplicación tenga un número fijo de reportes a generar y por tanto, ante la necesidad de agregar uno nuevo, este debe ser programado por el equipo de desarrolladores, en lugar de permitírsele al usuario que él mismo sea quien lo elabore.

Objeto de automatización.

La herramienta se utilizará para establecer un estándar en el proceso de generación de reportes, a continuación se detallan los procesos fundamentales que son objeto de automatización:

Proceso de diseñar un reporte.

El proceso de diseñar el reporte es el más importante dentro de la aplicación, el mismo comienza cuando el Usuario de Reporte (puede ser cualquier persona que necesite diseñar un reporte) solicita realizar un reporte a partir de una información disponible (base de datos del cliente), el Gestor de Reporte (cualquier persona que se encarga del manejo de reportes, por ejemplo un secretario, un bibliotecario, un estadístico) le muestra una guía de diseño (libro de patrones de diseño y de ejemplos de reportes diseñados) que le permite al Usuario de Reporte diseñar el reporte, una vez terminado el reporte, el usuario puede guardarlo para mostrarlo posteriormente.

Proceso de obtener un reporte.

El proceso de obtener un reporte le permite al usuario ver y guardar un reporte diseñado anteriormente, el mismo comienza cuando el Usuario de Reporte (puede ser cualquier persona que necesite obtener un reporte determinado) solicita obtener un reporte determinado, el Gestor de Reporte (cualquier persona que se encarga del manejo de reportes, por ejemplo un secretario, un bibliotecario, un estadístico) busca el reporte solicitado en la lista de reportes, si el reporte se encuentra el gestor de reportes muestra el reporte y el usuario lo guarda, si no, el gestor de reportes informa que no se encuentra el reporte solicitado.

Información que se maneja.

La herramienta maneja, gestiona, y elabora reportes de la información contenida en las bases de datos de la cual se desea realizar el reporte, el usuario debe estar previamente autenticado en el sistema, este no podrá modificar el contenido de la base de datos, ya que sólo podrá leer los datos contenidos en ella.

2.2 Propuesta de sistema.

Se propone el diseño de una herramienta que sea flexible, es decir, que el usuario pueda abstraerse de los conocimientos de gestores de bases de datos. La herramienta debe generar de forma dinámica los reportes, es decir, que permita a el usuario autorizado seleccionar el gestor de base datos a utilizar (MySQL ó PostGreSQL) y, mediante un asistente que le oriente los pasos a seguir, elaborar el reporte; dentro de la elaboración del mismo el usuario podrá incluir en el reporte los campos deseados dentro de cualquier tabla de la base de datos accedida, ver, si lo desea las sentencias de las la consultas realizadas, así como establecer un orden jerárquico requerido entre los campos de información.

Los campos sólo pueden ser modificados, es decir el usuario no tiene autorización para cambiar la información contenida en los mismos. Se debe garantizar que se pueda guardar el reporte una vez diseñado. La herramienta debe facilitar la gestión del reporte ya diseñado, es decir, visualizarlo, eliminarlo, exportarlo a otros formatos como PDF, Microsoft Word y Microsoft Excel, mostrar todos lo reportes diseñados así como buscar un reporte determinado.

2.3 Modelo del negocio

Representación de los actores y trabajadores del negocio.

Puede constituir un **actor del negocio** cualquier persona, individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados en este caso se ha definido como actores del negocio el Usuario de Reporte el cual es descrito en el Anexo 1.

Un **trabajador del negocio** define el comportamiento y responsabilidades de un individuo (o grupo de individuos que trabajan en conjunto como un equipo), sistema automatizado o máquina. Realiza las actividades y es propietario de elementos en este caso se ha definido como trabajador del negocio a el Gestor de Reportes el cual es descrito en el Anexo 1.

Diagrama de casos de uso del negocio

Este diagrama muestra la relación entre los actores y los casos de uso del negocio. (Ver Anexo1)

Descripción de los casos de uso del negocio.

Constituye un **caso de uso** el conjunto de secuencia de acciones que un sistema ejecuta y que produce un resultado observable para un actor.

La descripción de los caso de uso de negocio constituye un artefacto correspondiente al flujo de trabajo Modelado del Negocio de RUP llamado “Modelo de Casos de Uso del Negocio”. (Ver Anexo 1)

Diagramas de actividad

Un **diagrama de actividad** es un grafo que contiene distintos estados en que puede hallarse una actividad. (Ver Anexo1)

Representación de las entidades del negocio.

Una **entidad del negocio** representa un contenedor de información, algo físico que se utiliza en el proceso del negocio y que sirve para obtener información o para actualizar la misma.

En este caso el negocio contiene tres entidades; la primera, es “Lista de reportes” la cual representa los reportes ya diseñados, la segunda, es “Reporte” que representa el reporte elaborado por el usuario y la tercera es “Guía de diseño” que representa un conjunto de patrones de diseño que sirven de ejemplo al usuario en la elaboración del reporte. (Ver Anexo 1)

2.4 Especificación de los requisitos del software.

Un **requisito** es una característica que un sistema debe tener para cubrir alguna de las necesidades que lo motivan.

El objetivo de la captura de requisitos es guiar el desarrollo de software hacia el sistema correcto, definiendo objetivos generales concretos de manera tal que tanto el negocio como sus actores se beneficien.

Requisitos funcionales.

Los **requerimientos funcionales** son capacidades o condiciones que el sistema debe cumplir.

Se han definido los siguientes requisitos funcionales:

RF1 Autenticar el usuario en el sistema.

RF1.1 Seleccionar por parte del usuario el gestor de base de datos del cual obtendrá el reporte.

RF2 Favorecer condiciones para la abstracción del usuario a los conocimientos de gestores de bases de datos.

RF3 Elaborar los reportes de forma dinámica mediante un asistente.

RF3.1 Seleccionar las variables para la confección del reporte.

RF3.2 Configurar el estilo del reporte.

RF3.3 Elaborar agrupaciones de datos del reporte.

RF3.4 Favorecer la entrada de condiciones bajo las cuales debe generarse el reporte.

RF3.5 Favorecer la selección de operaciones de cálculo general.

RF3.6 Guardar el reporte diseñado.

RF4 Favorecer condiciones que permitan la gestión de los reportes.

RF4.1 Visualizar un reporte determinado.

RF4.2 Exportar el reporte a formato PDF, Microsoft Excel ó Microsoft Word.

RF4.3 Eliminar un reporte determinado.

RF5 Buscar reportes con una característica determinada.

Requisitos no funcionales.

Los **requisitos no funcionales** son propiedades o cualidades que el producto debe tener.

Se han definido los siguientes Requisitos no Funcionales según su clasificación.

Requisitos de usabilidad.

RNF1 El sistema debe permitir un acceso rápido y sencillo, el mismo podrá ser utilizado por usuarios con poco conocimiento de gestores de base de datos.

Requisitos de rendimiento.

RNF2 Establecer un orden lógico del acceso a las páginas con el objetivo de dar una respuesta más rápida y precisa evitando la sobrecarga de imágenes y objetos que dificulten la rapidez de las respuestas al usuario.

Requisitos de portabilidad.

RNF3 Permitir que el sistema pueda ser ejecutado en Linux, Windows 98 o superior.

Requisitos de disponibilidad.

RNF7 Los usuarios autorizados tendrán acceso a la información en todo momento.

RNF8 El Usuario de Reporte sólo podrá ver los nombres colocados por el Administrador de Reporte.

Requisitos de apariencia.

RNF9 Debe poseer una interfaz acorde al desarrollo de las aplicaciones web actuales.

RNF10 Explorar de manera sencilla.

RNF11 Mostrar las opciones que puede efectuar dentro del sistema según el rol del usuario.

Requisitos legales

RNF12 El empleo del sistema debe estar regido por un manual de normas y procedimientos aprobado por la Dirección de la Universidad y basado en las disposiciones legales vigentes.

2.5 Modelo del sistema.

Actores del sistema

Los actores del sistema generalmente son los trabajadores del negocio o actores del mismo. En este caso se ha definido como actores del sistema el Usuario de Reporte y el Administrador de Reporte para más información. (Ver Anexo 2)

Descripción de los casos de uso arquitectónicamente significativos

Esta descripción se realiza con el objetivo de detallar el flujo de sucesos de cada caso de uso, como se inicia y se termina, las precondiciones y poscondiciones, su interacción con los actores, la prioridad así como los puntos de extensión. (Ver Anexo 2)

Este capítulo posibilitó una mayor comprensión de las características del sistema, en el mismo se elaboraron un conjunto de artefactos que establecen las metas y condiciones que el sistema debe cumplir y que aparecen en los anexos de este trabajo. Además se precisaron los detalles más específicos que facilitan el análisis y diseño del mismo como son el modelo del sistema y el detallado de cada caso de uso.

CAPÍTULO 3: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA HERRAMIENTA

Este capítulo se centra principalmente en la realización del modelo de análisis y diseño del sistema así como en la elaboración de los diagramas de clases y de interacción de los casos de uso detallados en el capítulo anterior. En el mismo se realizan la descripción detallada de las clases.

3.1 Análisis.

El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales.

Diagrama de clases de análisis.

Representa la relación del actor con las clases del análisis así como la interrelación entre las mismas. Las figuras muestran los respectivos diagramas de clase de análisis para cada caso de uso del sistema.

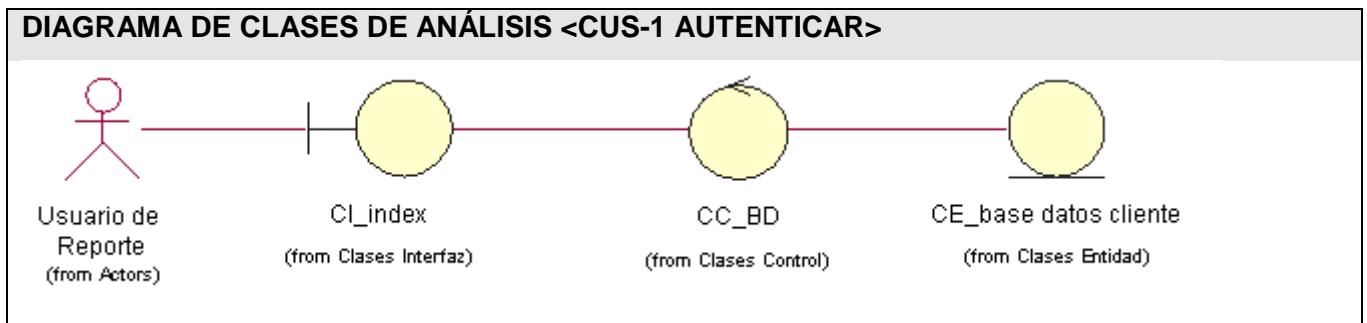


Figura 3.1 Diagrama de clases de Análisis: CUS-1 Autenticar.

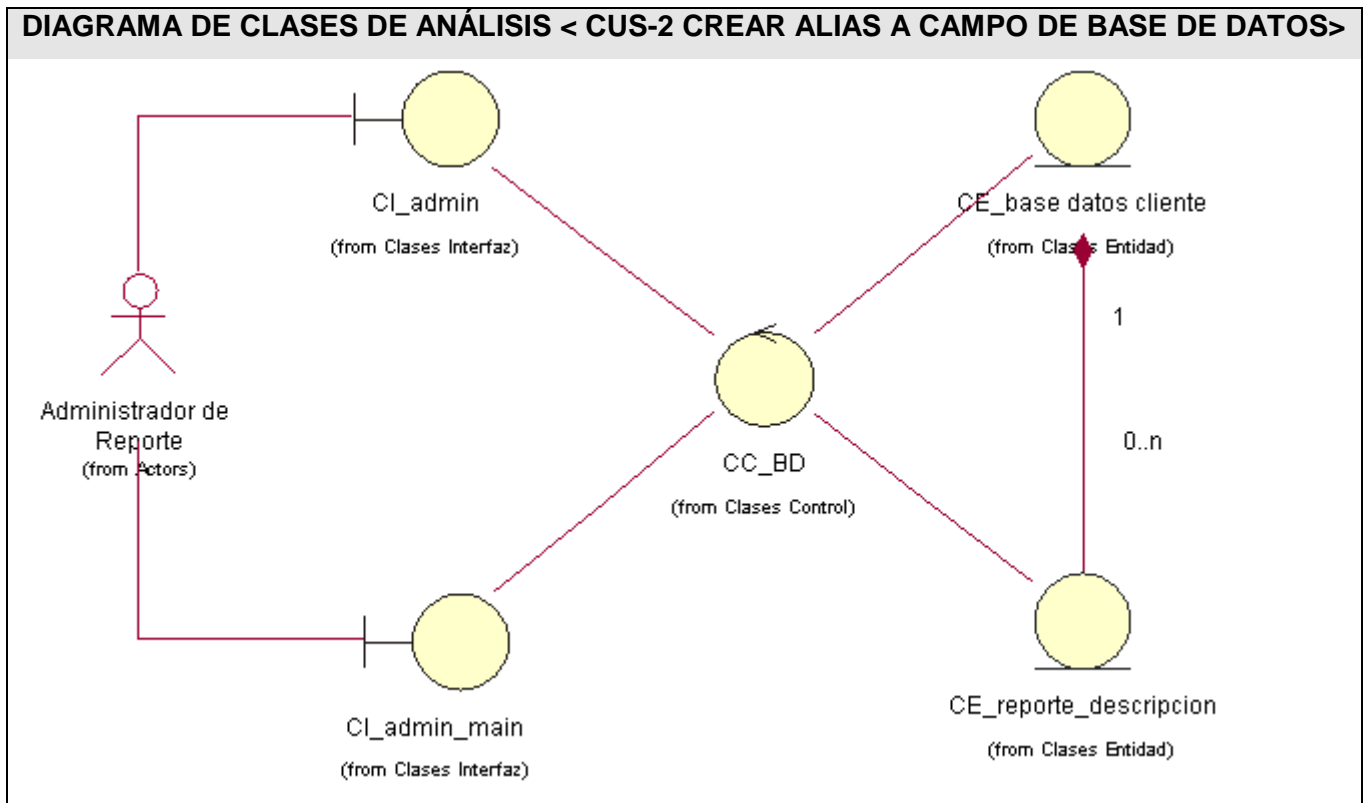


Figura 3.2 Diagrama de clases de Análisis: CUS-2 Crear alias a campo de base de datos.

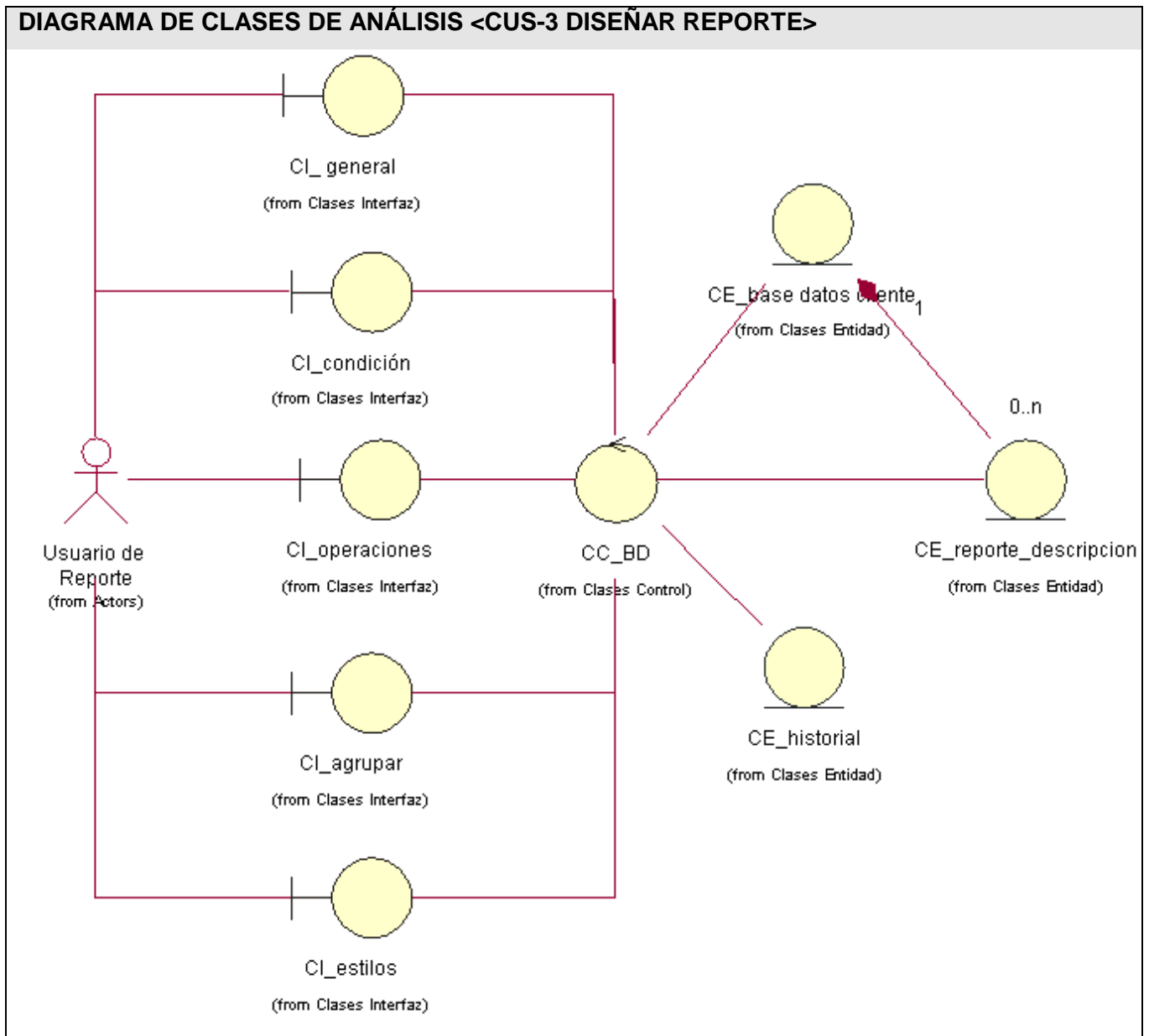


Figura 3.3 Diagrama de clases de Análisis: CUS-3 Diseñar reporte.

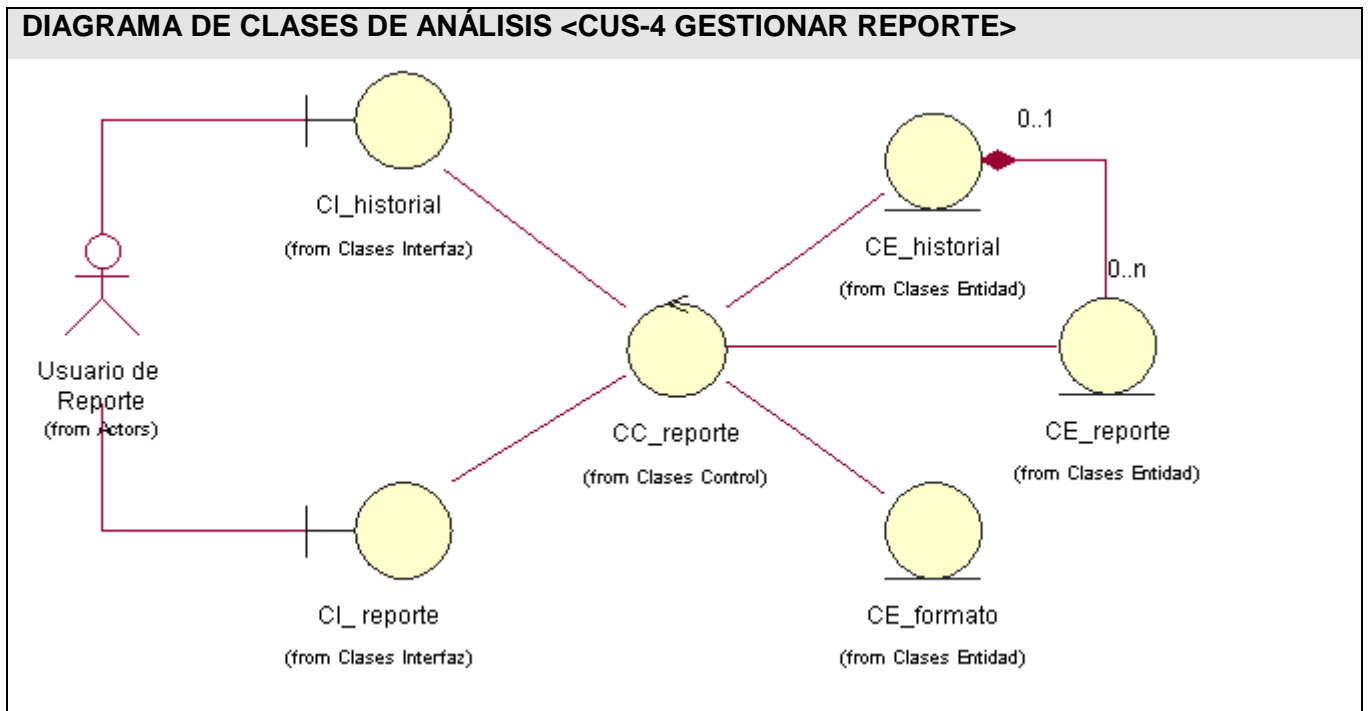


Figura 3.4 Diagrama de clases de Análisis: CUS-4 Gestionar reporte.

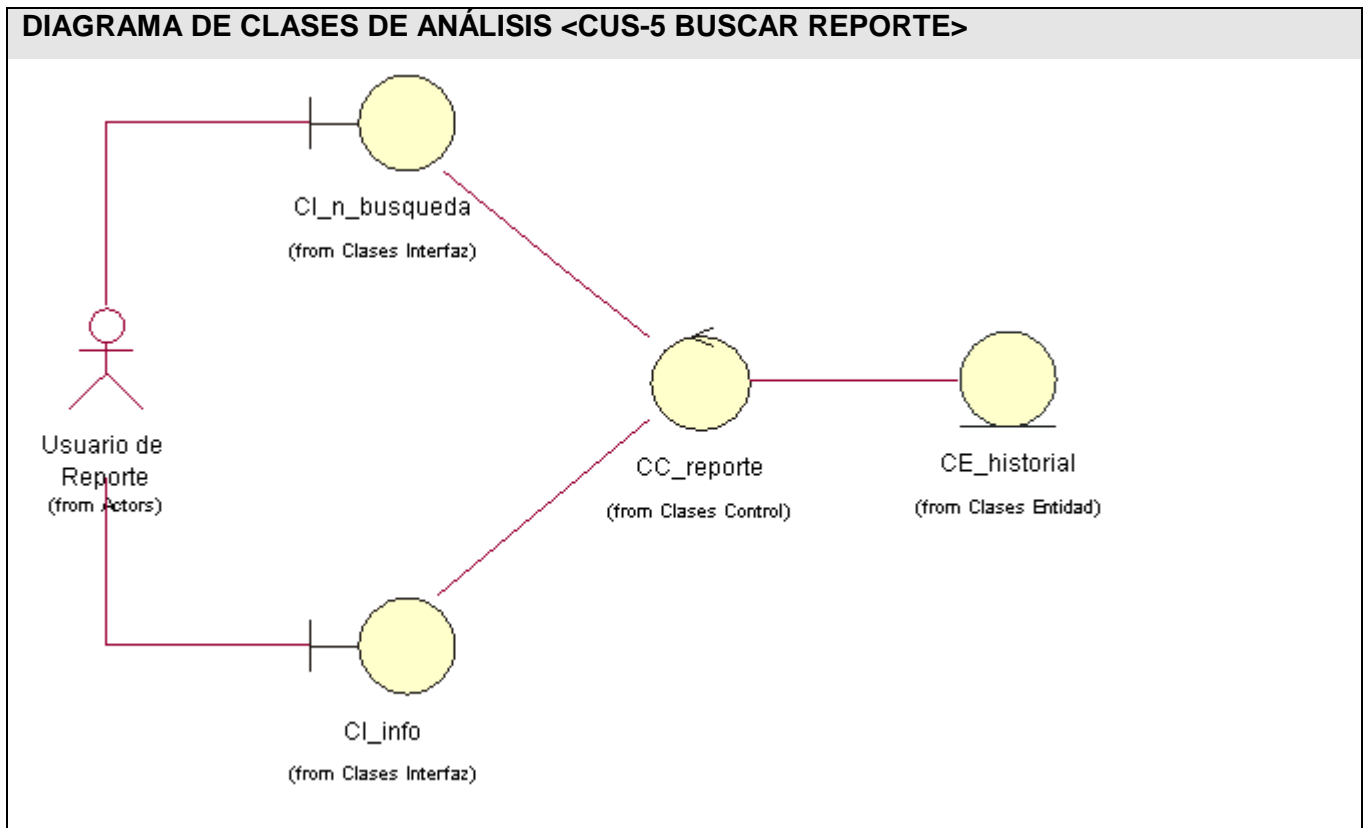


Figura 3.6 Diagrama de clases de Análisis: CUS-5 Buscar Reporte.

3.2 Diseño

Diagrama de clases de diseño.

El diagrama de clases de diseño representa la estructura y organización de las clases que intervienen en el caso de uso desde un punto de vista orientado a la implementación y más específico donde están presentes los métodos a utilizar en cada clase y la forma de comunicación de las mismas.

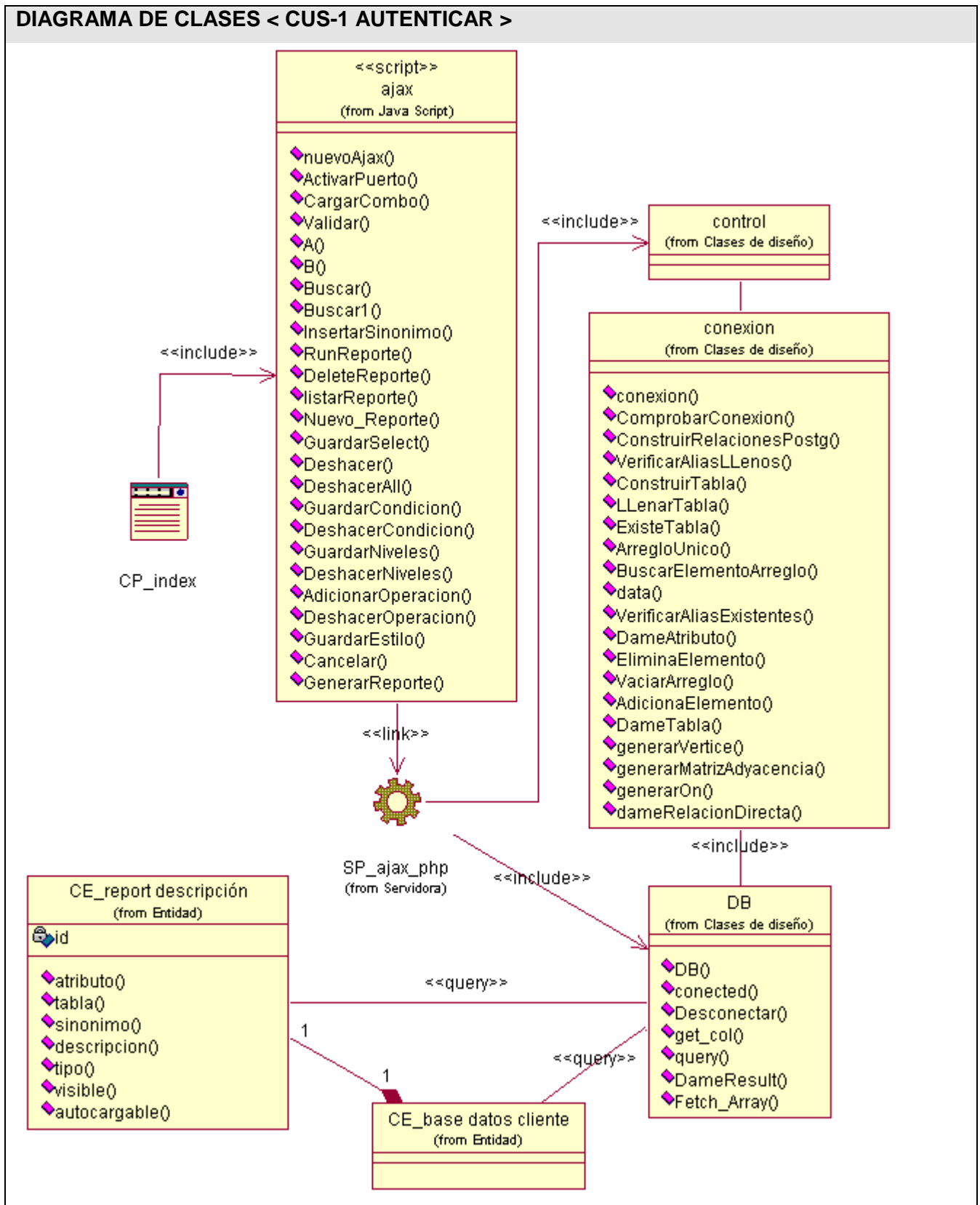


Figura 3.7: Diagrama de clases de diseño: CUS-1 Autenticar.

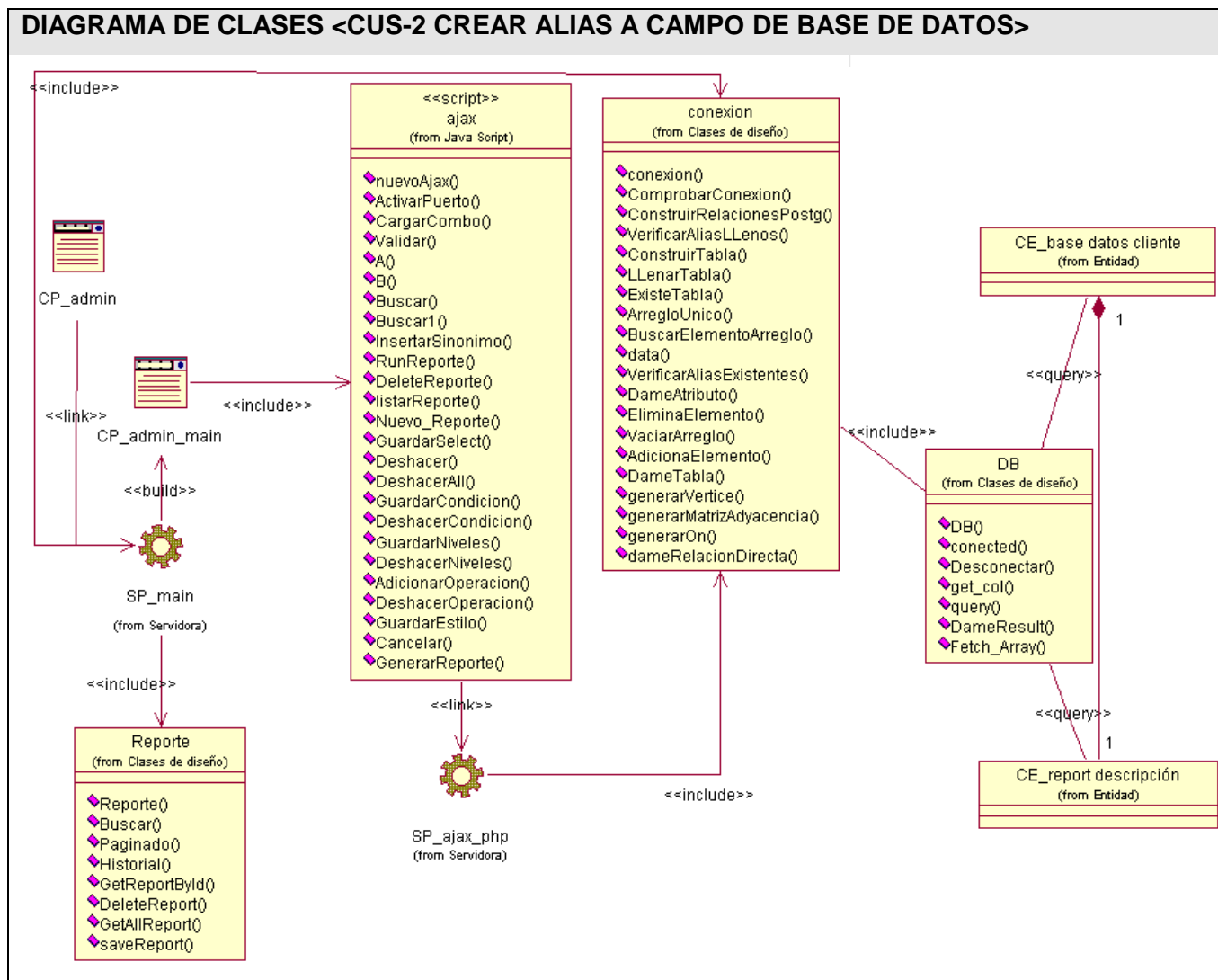


Figura 3.8: Diagrama de clases de diseño: CUS-2 Crear alias a campo de base de datos.

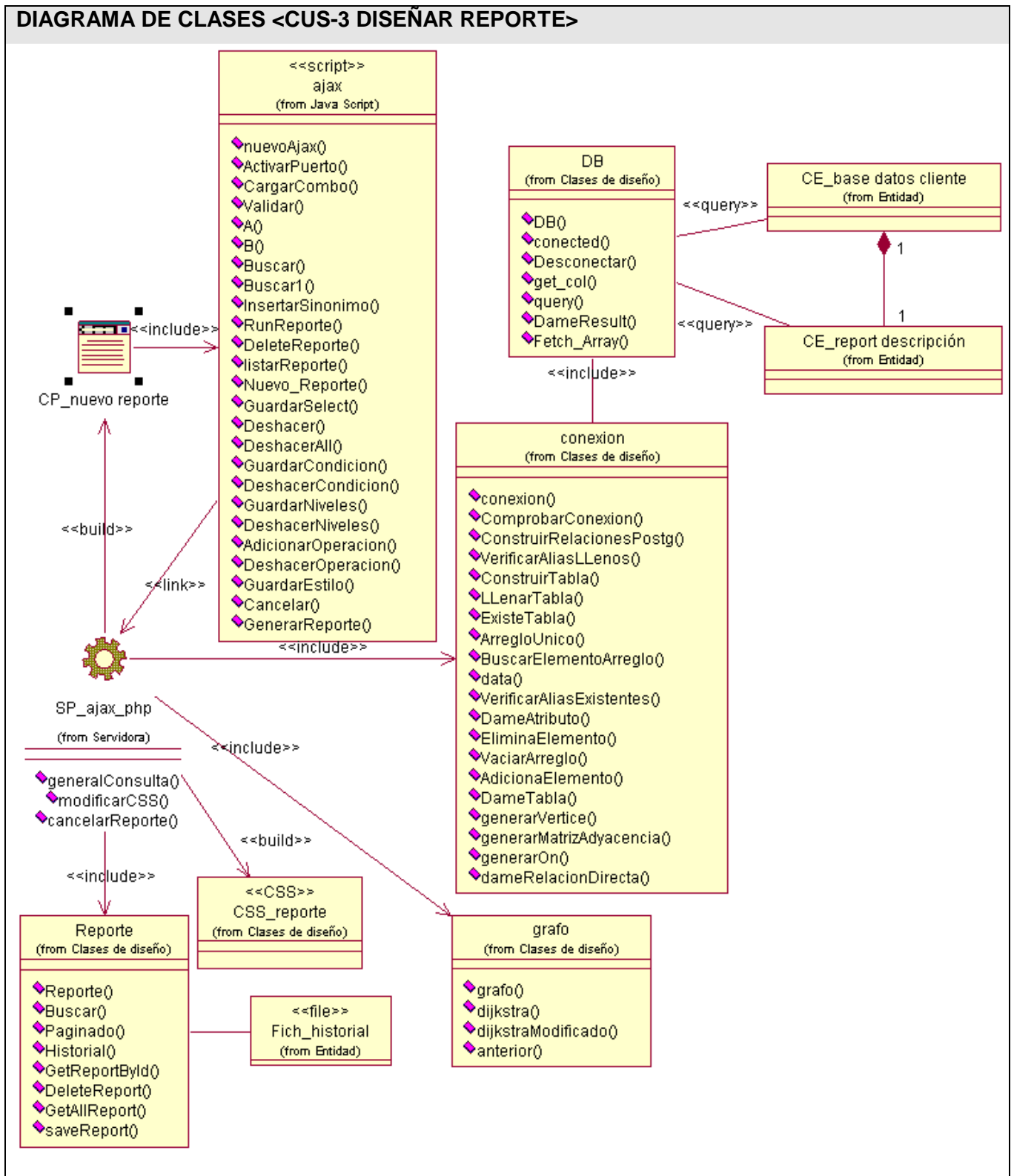


Figura 3.9: Diagrama de clases de diseño: CUS-3 Diseñar reporte.

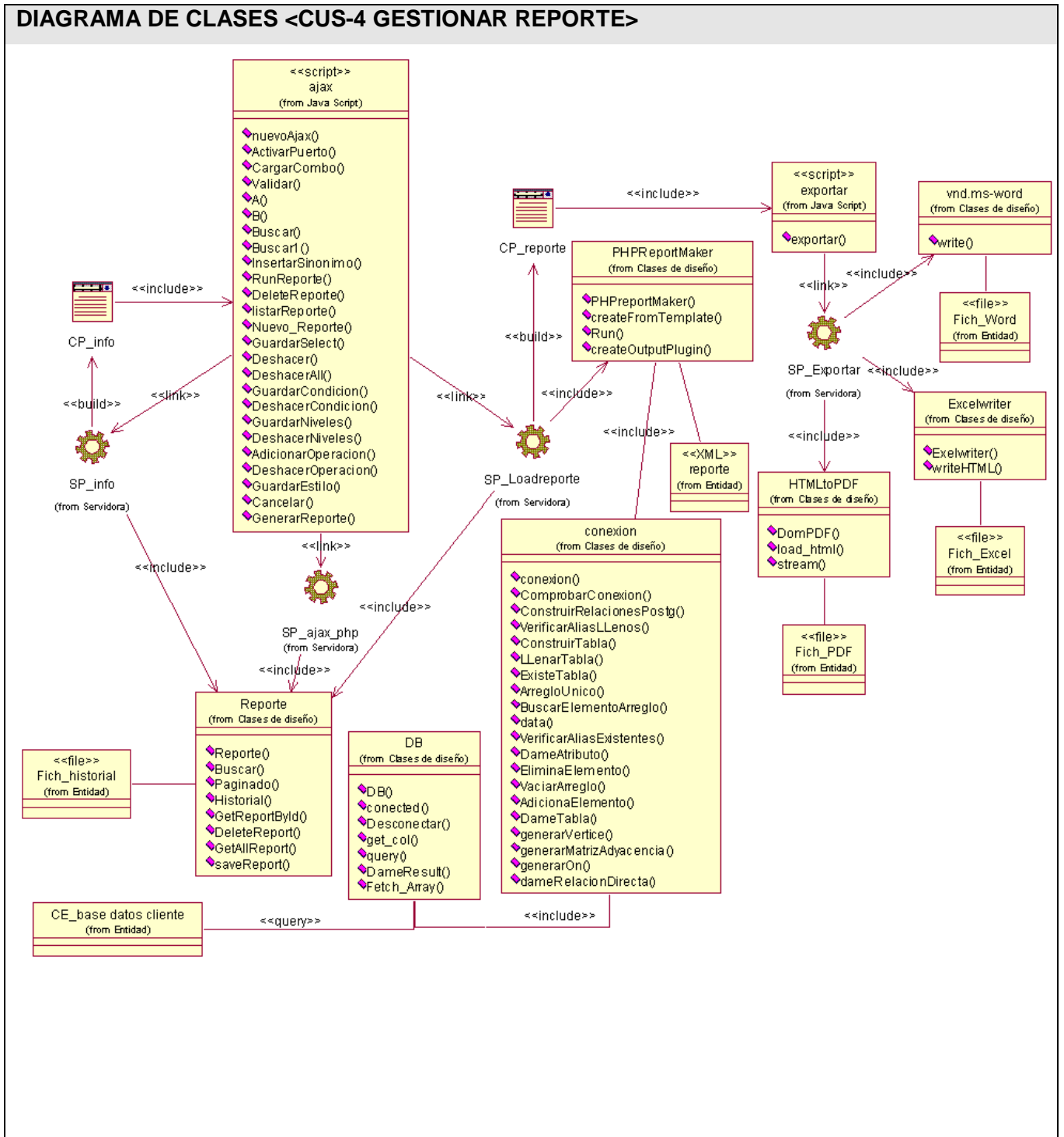


Figura 3.10: Diagrama de clases de diseño: CUS-3 Gestionar reporte.

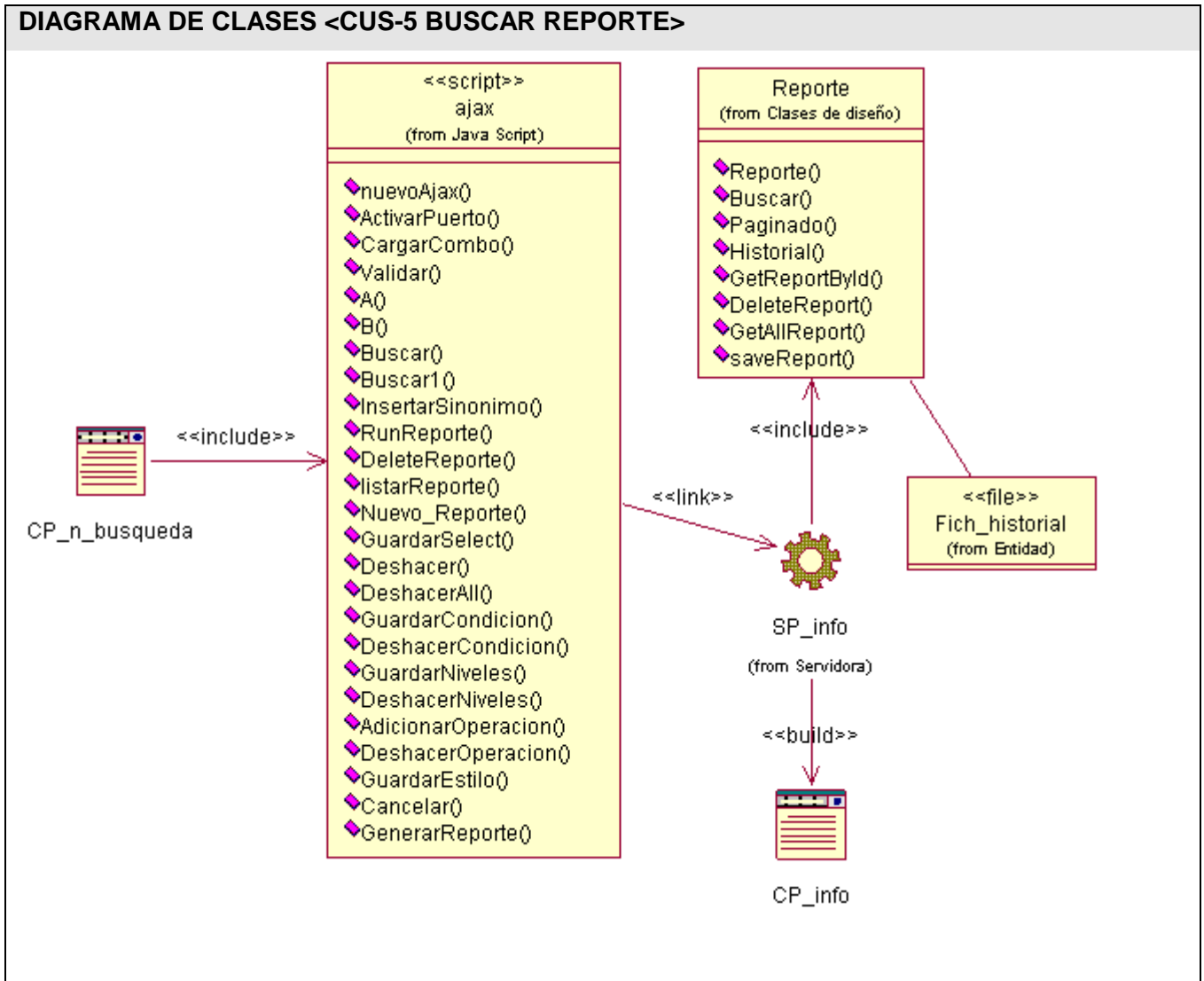


Figura 3.11: Diagrama de clases de diseño: CUS-5 Buscar reporte.

Diagrama de interacción.

El diagrama de secuencia forma parte de la vista del comportamiento del sistema, brinda una especificación de este; representa un conjunto de clases que intercambian mensajes en una iteración organizada a lo largo del tiempo.

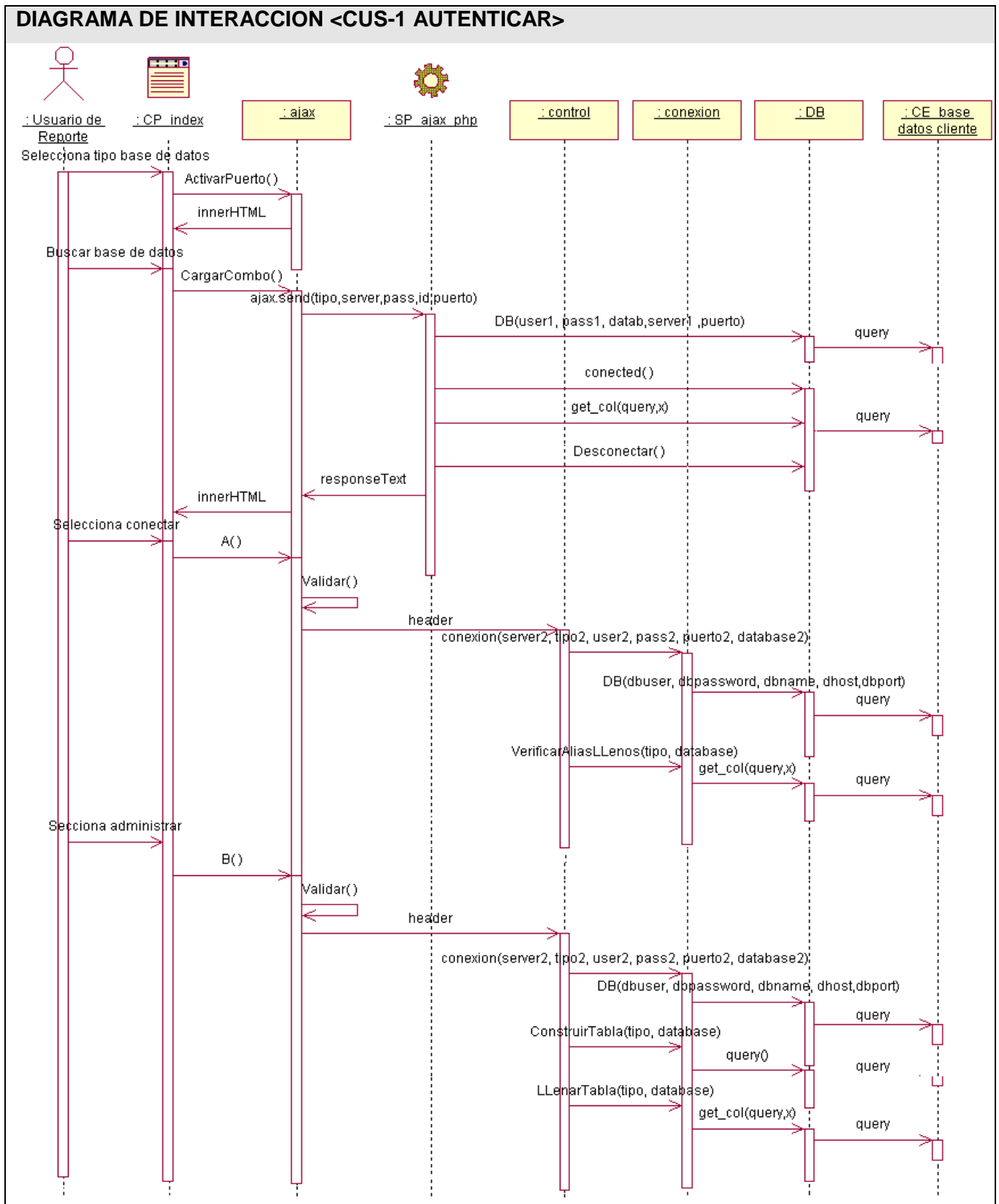


Figura 3.12: Diagrama de interacción: CUS-1 Autenticar.

El resto de los diagramas de secuencia se encuentran en el Anexo 2.

Descripción de las Clases.

Nombre: CP_index (CU-1)	
Tipo de clase: Interfaz	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	-
Descripción:	Muestra los requerimientos para conectarse a la base de datos.

Tabla 3.1: Descripción de clases: CP_index (CU-1).

Nombre: CP_admin (CU-2)	
Tipo de clase: Interfaz	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	-
Descripción:	Muestra las variables agregadas (alias de campo de la base de datos), brinda la posibilidad de explorar la base de datos, muestra información de las variables de una tabla en específico.

Tabla 3.2: Descripción de clases: CP_admin (CU-2).

Nombre: CP_admin_main (CU-2)	
Tipo de clase: Interfaz	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	-
Descripción:	Solicita introducir los datos para modificar una variable (alias) determinada.

Tabla 3.3: Descripción de clases: CP_admin_main (CU-2).

Nombre: CP_nuevo_reporte (CU-3)	
Tipo de clase: Interfaz	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	-
Descripción:	Muestra los pasos y opciones para diseñar el reporte.

Tabla 3.4: Descripción de clases: CP_nuevo_reporte (CU-3).

Nombre: CP_info (CU-4)	
Tipo de clase: Interfaz	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	-
Descripción:	Muestra un listado de reportes, brinda opciones de ejecutar o eliminar un reporte determinado.

Tabla 3.5: Descripción de clases: CP_info (CU-4).

Nombre: CP_reporte (CU-4)	
Tipo de clase: Interfaz	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	-
Descripción:	Muestra un reporte determinado, brinda opciones para exportar el mismo a otros formatos como PDF, Microsoft Word y Microsoft Excel.

Tabla 3.6: Descripción de clases: CP_reporte (CU-4).

Nombre: CP_n_busqueda (CU-5)	
Tipo de clase: Interfaz	
Atributo	Tipo

CAPITULO 3: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA HERRAMIENTA

-	-
Responsabilidad:	
Nombre:	-
Descripción:	Solicita la introducción de criterios para buscar reportes.

Tabla 3.7: Descripción de clases: CP_n_búsqueda (CU-5).

Nombre: Ajax (CU-1,2,3,4,5)	
Tipo de clase: Control (JavaScript)	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	nuevoAjax
Descripción:	Constructor del objeto Ajax.
Nombre:	ActivarPuerto
Descripción:	Función para cargar el puerto por defecto según el gestor de DB.
Nombre:	CargarCombo
Descripción:	Función para cargar la lista de base de datos disponible por servidor.
Nombre:	Validar
Descripción:	Función para validar que todos los campos estén llenos.
Nombre:	A
Descripción:	Función para redireccionar al usuario de reporte.
Nombre:	B
Descripción:	Función para redireccionar al administrador.
Nombre:	Buscar
Descripción:	Función para cargar el asistente de búsqueda.
Nombre:	Buscar1
Descripción:	Función para ejecutar la búsqueda desde el asistente de búsqueda.
Nombre:	InsertarSinonimo
Descripción:	Función para insertar el alias.
Nombre:	RunReporte
Descripción:	Función para ejecutar un reporte salvado.
Nombre:	DeleteReporte

CAPITULO 3: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA HERRAMIENTA

Descripción:	Función para borrar un reporte salvado.
Nombre:	listarReporte
Descripción:	Función para mostrar todos los reportes salvados.
Nombre:	Nuevo_Reporte
Descripción:	Función para activar el asistente para crear un nuevo reporte.
Nombre:	GuardarSelect
Descripción:	Función para adicionar las variables a mostrar.
Nombre:	Deshacer
Descripción:	Función para deshacer la ultima variable adicionada a mostrar.
Nombre:	DeshacerAll
Descripción:	Función para deshacer todas las variables adicionadas a mostrar.
Nombre:	GuardarCondicion
Descripción:	Función para adicionar una condición.
Nombre:	DeshacerCondicion
Descripción:	Función para deshacer la ultima condición adicionada.
Nombre:	GuardarNiveles
Descripción:	Función para guardar los grupos.
Nombre:	DeshacerNiveles
Descripción:	Función para deshacer los grupos.
Nombre:	AdicionarOperacion
Descripción:	Función para adicionar una operación.
Nombre:	DeshacerOperacion
Descripción:	Función para deshacer una operación.
Nombre:	GuardarEstilo
Descripción:	Función para guardar estilo.
Nombre:	Cancelar
Descripción:	Función para cancelar el reporte y cerrar el asistente.
Nombre:	GenerarReporte
Descripción:	Función para generar el reporte después de ser diseñado.

Tabla 3.8: Descripción de clases: Ajax (CU-1,2,3,4,5).

Nombre: Exportar (CU-4)	
Tipo de clase: Control (JavaScript)	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	exportar
Descripción:	Función que direcciona según el formato al que se desea exportar.

Tabla 3.9: Descripción de clases: Exportar (CU-4).

Nombre: SP_ajax.php (CU-1,2,3,4)	
Tipo de clase: Control (sever page)	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	generarConsulta
Descripción:	Función que integra todos los elementos del reporte y genera la consulta SQL.
Nombre:	modificarCSS
Descripción:	Función que modifica el ccs por la configuración del estilo guardado.
Nombre:	cancelarReporte
Descripción:	Función para destruir todos los elementos del reporte.

Tabla 3.10: Descripción de clases: SP_ajax.php (CU-1, 2, 3,4).

Nombre: SP_main (CU-2)	
Tipo de clase: Control (sever page)	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	-
Descripción:	Gestiona el proceso de actualización de las variables (alias).

Tabla 3.11: Descripción de clases: SP_main (CU-2).

Nombre: SP_loadReporte (CU-4)	
Tipo de clase: Control (sever page)	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	-
Descripción:	Gestiona el reporte.

Tabla 3.12: Descripción de clases: SP_loadReporte (CU-4).

Nombre: SP_exportar (CU-4)	
Tipo de clase: Control (sever page)	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	-
Descripción:	Gestiona la exportación del reporte según el formato especificado (PDF, Microsoft Word, Microsoft Excel).
Nombre: SP_info (CU-4,5)	
Tipo de clase: Control (sever page)	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	-
Descripción:	Gestiona el proceso de actualizar la lista de reportes (historial).

Tabla 3.13: Descripción de clases: SP_info (CU-4,5).

Nombre: SP_info (CU-4,5)	
Tipo de clase: Control (sever page)	
Atributo	Tipo
-	-
Responsabilidad:	

Nombre:	-
Descripción:	Gestiona el proceso de actualizar la lista de reportes (historial).

Tabla 3.14: Descripción de clases: SP_info (CU-4,5).

Nombre: conexion (CU-1,2,3,4)	
Tipo de clase: Control	
Atributo	Tipo
server	String
tipo	String
user	String
pass	String
database	String
puerto	String
data	db()
Responsabilidad:	
Nombre:	conexion
Descripción:	Constructor de la clase especializada.
Nombre:	ComprobarConexion
Descripción:	Función que comprueba si logro la conexión.
Nombre:	VerificarAliasLLenos
Descripción:	Función que verifica si hay alguna variable agregada.
Nombre:	ConstruirTabla
Descripción:	Función para construir la tabla report_descripcion.
Nombre:	LLenarTabla
Descripción:	Función para copia todos los atributos a la tabla report_descripcion.
Nombre:	ExisteTabla
Descripción:	Función que devuelve si existe o no una tabla dada.
Nombre:	ArregloUnico
Descripción:	Función que elimina los elementos repetidos de un arreglo dado.
Nombre:	BuscarElementoArreglo
Descripción:	Función que devuelve si existe o no un elemento dado en un arreglo dado.
Nombre:	data

CAPITULO 3: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA HERRAMIENTA

Descripción:	Función que devuelve el objeto de la clase DB.
Nombre:	VerificarAliasExistentes
Descripción:	Función que devuelve si existe o no un alias dado.
Nombre:	DameAtributo
Descripción:	Función que devuelve el atributo correspondiente a un alias dado.
Nombre:	EliminaElemento
Descripción:	Función que elimina un elemento dado de un arreglo dado.
Nombre:	VaciarArreglo
Descripción:	Función que elimina todos los elementos de un arreglo dado.
Nombre:	AdicionaElemento
Descripción:	Función que adiciona un elemento dado en un arreglo dado.
Nombre:	DameTabla
Descripción:	Función que devuelve la tabla correspondiente a un alias dado.
Nombre:	generarVertice
Descripción:	Función que genera una lista con todas las tablas como vértices.
Nombre:	generarMatrizAdyacencia
Descripción:	Función que genera una matriz de adyacencia de la lista de vértices .
Nombre:	generarOn
Descripción:	Función que devuelve una cadena con la relación de dos tablas dadas.
Nombre:	dameRelacionDirecta
Descripción:	Función que devuelve un arreglo con la relación entre dos tablas dadas.

Tabla 3.15: Descripción de clases: conexion (CU-1,2,3,4).

Nombre: BD (CU-1,2,3,4)	
Tipo de clase: Control	
Atributo	Tipo
dbh	String
dbuser	String
dbpass	String
dbhost	String
dbport	String
dbname	String

Responsabilidad:	
Nombre:	DB
Descripción:	Función que construye el objeto DB.
Nombre:	conected
Descripción:	Función que verifica si el atributo dbh se creó correctamente.
Nombre:	Desconectar
Descripción:	Función que destruye el objeto DB.
Nombre:	get_col
Descripción:	Función que devuelve una columna correspondiente a una consulta SQL.
Nombre:	query
Descripción:	Función que ejecuta una consulta SQL.
Nombre:	DameResult
Descripción:	Función que ejecuta una consulta SQL y devuelve el objeto resultante.
Nombre:	Fetch_Array
Descripción:	Función que devuelve el arreglo dado un objeto resultante.

Tabla 3.16: Descripción de clases: BD (CU-1,2,3,4).

Nombre: Reporte (CU-2,3,4,5)	
Tipo de clase: Control	
Atributo	Tipo
user	String
pass	String
dbname	String
server	String
port	String
tipobd	String
css	String
sql	String
titulo	String
fecha	String
hora	String
grupo	String

Responsabilidad:	
Nombre:	Reporte
Descripción:	Función que construye el objeto reporte.
Nombre:	Buscar
Descripción:	Función que busca un reporte dado un título una fecha y una hora.
Nombre:	Paginado
Descripción:	Función que devuelve los reportes separados por páginas.
Nombre:	Historial
Descripción:	Función que devuelve una matriz con todos los reportes salvados.
Nombre:	GetReportById
Descripción:	Función que devuelve un reporte como objeto dado un id.
Nombre:	DeleteReport
Descripción:	Función elimina un reporte dado un título una fecha y una hora.
Nombre:	GetAllReport
Descripción:	Función que devuelve todos los reportes como objetos.
Nombre:	saveReport
Descripción:	Función para salvar un reporte.

Tabla 3.17: Descripción de clases: Reporte (CU-2,3,4,5).

Nombre: grafo (CU-3)	
Tipo de clase: Control	
Atributo	Tipo
V	array
C	array
S	array
P	array
D	array
Responsabilidad:	
Nombre:	grafo
Descripción:	Función que construye el grafo
Nombre:	dijkstra
Descripción:	Función que devuelve todos los caminos mínimos dado un vértice inicial.

CAPITULO 3: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA HERRAMIENTA

Nombre:	dijkstraModificado
Descripción:	Función que devuelve todos los caminos mínimos dado un vértice i y uno final.
Nombre:	anterior
Descripción:	Función que devuelve el predecesor de un vértice.

Tabla 3.18: Descripción de clases: grafo (CU-3).

Nombre: PHPReportMaker (CU-4)	
Tipo de clase: Control	
Atributo	Tipo
sXML	String
sXSLT	String
sUser	String
sPass	String
sCon	String
sDatal	String
sSQL	String
_oParm	String
sDatabase	String
_oError	String
Responsabilidad:	
Nombre:	PHPReportMaker
Descripción:	Función que construye el objeto que construye el reporte
Nombre:	createFromTemplate
Descripción:	Función que construye el xml
Nombre:	Run
Descripción:	Función que ejecuta el reporte
Nombre:	createOutputPlugin
Descripción:	Función que transforma el xml a xhtml con xslt

Tabla 3.19: Descripción de clases: PHPReportMaker (CU-4).

Nombre: HTMLtoPDF (CU-4)
Tipo de clase: Control

Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	DomPDF
Descripción:	Función que construye el objeto PDF.
Nombre:	load_html
Descripción:	Función que carga el código HTML en un buffer para escribirlo en el pdf.
Nombre:	stream
Descripción:	Función que parsea el código HTML y lo escribe en el documento pdf.

Tabla 3.20: Descripción de clases: HTMLtoPDF (CU-4).

Nombre: Excelwriter (CU-4)	
Tipo de clase: Control	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	Exelwriter
Descripción:	Función que construye el objeto Excel
Nombre:	writeHTML
Descripción:	Función que escribe el código HTML en el documento Excel.

Tabla 3.21: Descripción de clases: Excelwriter (CU-4).

Nombre: vnd.ms-word (CU-4)	
Tipo de clase: Control	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	write
Descripción:	Función que escribe código HTML al documento Word

Tabla 3.22: Descripción de clases: vnd.ms-word (CU-4).

Nombre: CE_Report_Descripcion (CU-2,3)	
Tipo de clase: Entidad	
Atributo	Tipo
id	Integer
atributo	String
tabla	String
sinonimo	String
descripción	String
tipo	String
visible	String
autocargable	String
Responsabilidad:	
Nombre:	-
Descripción:	Almacena la información referente a los alias introducidos por el administrador

Tabla 3.23: Descripción de clases: CE_Report_Descripcion (CU-2,3).

Nombre: historial (CU-3,4,5)	
Tipo de clase: Entidad (file)	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	
Descripción:	Directorio que contiene todos los reportes salvados en forma de objetos serializados.

Tabla 3.24: Descripción de clases: historial (CU-3,4,5).

Nombre: reporte (CU-4)	
Tipo de clase: Entidad (XML file)	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	

Descripción:	Archivo xml que se construye con el reporte, XML a transformar.
--------------	---

Tabla 3.25: Descripción de clases: reporte (CU-4).

Nombre: PDF (CU-4)	
Tipo de clase: Entidad (file)	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	
Descripción:	Archivo pdf en el que se escribe y se lanza al explorador

Tabla 3.26: Descripción de clases: PDF (CU-4).

Nombre: Word (CU-4)	
Tipo de clase: Entidad (file)	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	
Descripción:	Archivo Word en el que se escribe y se lanza al explorador

Tabla 3.27: Descripción de clases: Word (CU-4).

Nombre: Excel (CU-4)	
Tipo de clase: Entidad (file)	
Atributo	Tipo
-	-
Responsabilidad:	
Nombre:	
Descripción:	Archivo Excel en el que se escribe y se lanza al explorador

Tabla 3.28: Descripción de clases: Excel (CU-4).

Diseño de la base de datos.

En el presente trabajo no existe un diseño concreto de la base de datos ya que la información que se necesita para elaborar el reporte proviene de una base de datos elaborada por el cliente. Es decir, el cliente en todos los casos, es el que aporta la base de datos para elaborar los reportes. Sólo se decidió agregar a la base de datos del cliente una tabla, para almacenar los datos de los alias de los campos del resto de las tablas, llamada “report_descripcion” la cual tiene como llave primaria “id” y los atributos: “atributo”, “tabla”, “sinónimo”, “descripción”, “tipo”, visibilidad, autocargable.

Diagrama entidad relación

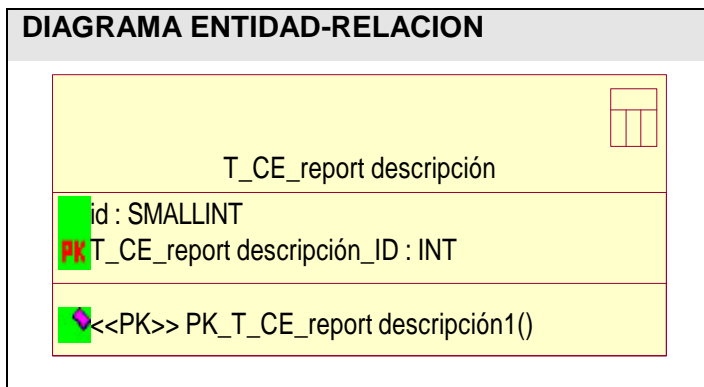


Figura 3.13 Diagrama Entidad-relación.

Método de dijkstra

Debido a que el cliente es también el proveedor de la base de datos, es necesario que la aplicación cuente con métodos que le permitan conocer el nombre de las tablas donde se encuentran los elementos solicitados por el usuario, así como conocer los enlaces entre las mismas, en caso de bases de datos de gran dimensión es necesario obtener las relaciones optimas entre las tablas para elaborar consultas más sencillas y eficientes.

Para lograr modelar las relaciones entre las tablas, se decidió considerar cada tabla como un vértice y cada enlace (atributos relacionantes) como una arista, conformándose de esta manera un grafo (ver glosario de términos) al cual se le aplica el método Dijkstra para obtener los caminos más óptimos en las consultas multitablas.

El algoritmo de Dijkstra consiste en ir explorando todos los caminos más cortos que parten del vértice origen y que llevan a todos los demás vértices; cuando se obtiene el camino más corto desde el vértice

origen, al resto de los vértices que componen el grafo, el algoritmo se detiene. El algoritmo es una especialización de la búsqueda de costo uniforme, y como tal, no funciona en grafos con aristas de costo negativo.

Descripción de las tablas.

Nombre: Report_descripcion		
Descripción: La tabla almacena la información relacionada con del alias de los campos de la base de datos.		
Atributo	Tipo	Descripción
id	int	Identificador de la tabla.
atributo	String	Nombre de un campo perteneciente a una tabla de la base de datos
tabla	String	Nombre de la tabla a la que pertenece el campo.
alias	String	Alias del campo.
descripción	String	Descripción del alias
tipo	String	Tipo de variable del atributo, entre paréntesis, la cantidad de caracteres permisibles ejemplo: varchar(20), int(5)
visible	String	Permite visualizar el atributo a incluir en el reporte.
autocargable	String	Permite que el sistema cargue los valores del atributo a la hora de establecer condiciones.

Tabla 3.29: Descripción de tablas: Report_descripcion.

Gestores de base de datos.

Para el trabajo con la aplicación se permite que el usuario pueda utilizar dos tipos de gestores MySQL y PostgreSQL. Actualmente estos gestores son los más utilizados en el área temática “Sistema de Apoyo a la Salud” y ofrecen gran seguridad y eficiencia, sobre todo en el caso de PostgreSQL.

Definiciones de diseño.

En el diseño de la aplicación, primeramente se tiene en cuenta el usuario que va a interactuar con la misma, el cual puede ser cualquier persona que necesite elaborar un reporte a partir de una base de datos determinada, o sea, la relación del cliente y la herramienta no depende de la profesión que ejerce el mismo, por lo que se propone una interfaz sencilla y fácil para ello es necesario lograr abstraer al usuario de la estructura interna de una Base de Datos relacional

Para lograr que la interfaz sea la adecuada se tienen en cuenta los siguientes aspectos:

- ✍ Lograr cohesión de la información a mostrar.
- ✍ Lograr uniformidad en los estilos de las páginas.
- ✍ Establecer un orden para la navegación y la realización de las actividades.
- ✍ Mostrar elementos que guíen al usuario en la realización de las actividades.

Concepción de la ayuda.

Debido a las características del usuario, fundamentalmente que puede ser cualquier persona, tenga bastos conocimientos informáticos o no, se elaboró una ayuda que explique detalladamente las actividades que podrá realizar cada tipo de usuario para lograr una mayor eficiencia en la elaboración de los reportes, la misma cuenta con un grupo de imágenes que le permite al usuario una mayor comprensión a la hora de realizar las operaciones.

3.3 Diagrama de despliegue.

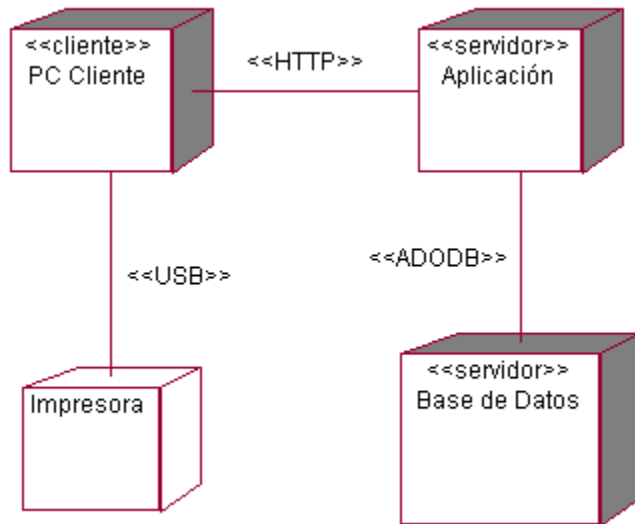


Figura 3.14 Diagrama de despliegue.

3.4 Diagramas de componentes.

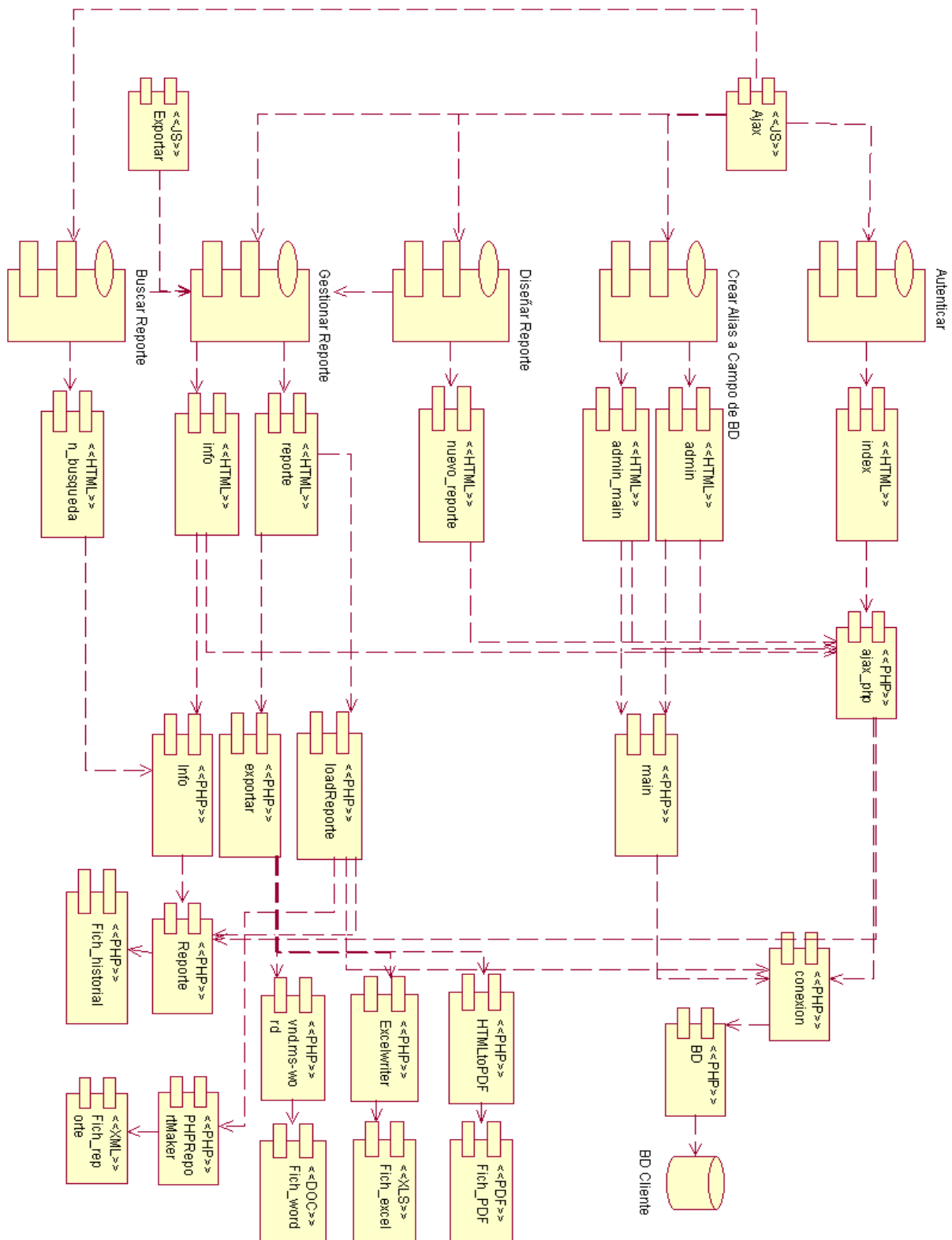


Figura 3.15: Diagrama de componentes.

Este capítulo se centró principalmente en la realización del modelo de análisis y diseño del sistema, así como en la elaboración de los diagramas de clases y de interacción de los casos de uso detallados en el capítulo anterior. Se realizó una descripción detallada de las clases así como el despliegue de la aplicación y los distintos componentes que la conforman.

CONCLUSIONES

Con el desarrollo de la aplicación se ha cumplido el objetivo propuesto ya que se ha desarrollado una herramienta en PHP para la generación dinámica de reportes, para lograrlo se han tenido en cuenta las tareas planteadas ya que:

- Se evaluaron herramientas que realizan funciones análogas en otros sistemas como el Crystal Reports, el Jasper Report y la herramienta del módulo de reportes de Akademos.
- La herramienta se desarrolló de manera general utilizando el paradigma de la programación orientada a objetos.
- Se obtuvieron los artefactos propios de las fases de análisis y diseño para el desarrollo de la herramienta como los diagramas de caso de uso y clases de diseño del sistema.
- Se diseñó la interfaz gráfica de la aplicación utilizando librerías Dojo en acorde con las exigencias del área temática.

La herramienta elaborada puede dar solución a los problemas que presenta el área temática “Sistema de Apoyo a la Salud” en la generación de los reportes que no necesiten de un modelo oficial para su diseño, así como satisfacer las necesidades de cualquier usuario que necesite crear un reporte con la información de su base de datos.

Debe significarse además, que la misma ha sido ya introducida como solución al problema de generar reportes en el proyecto: *“Sistema para el control de la colaboración médica cubana”* y se estudia su posible introducción en otros sistemas de suma importancia que son parte de los convenios de cooperación entre Cuba y la República Bolivariana de Venezuela.

RECOMENDACIONES

Se recomienda que se mantenga la herramienta actualizada con las tecnologías y lenguajes más utilizados

En el caso de realizarse futuras versiones, se debe:

- ✍ Evitar realizar modificaciones en la base de datos del cliente para ello se debe sustituir la tabla report_descripcion que actualmente se utiliza para almacenar la correspondencia de los alias con los atributos de las tablas, por otro tipo de almacenamiento ya sea a través de un XML u objetos serializados.
- ✍ Evaluar la posibilidad de agregar a las funcionalidades de la herramienta, la gestión de la seguridad en la integración a otras aplicaciones, la posibilidad de elaborar gráficos a partir de la información contenida en reporte, mejorar el diseño grafico de la interfaz ya que esta primera versión fue enfocada a las funcionalidades.

BIBLIOGRAFÍA

- AJAX, 2007. Disponible en: <http://es.wikipedia.org/wiki/AJAX>
- BRADENBAUGH, J. Aplicaciones JavaScript. S.A., A. M., 2000.
- BRUEGGE, B. D., A. Ingeniería de Software Orientado a Objetos. Prentice Hall-Pearson
- Descripción de la arquitectura en módulos del Apache, 2007. Disponible en:
<http://www.desarrolloweb.com/articulos/1112.php>
- Documentación del Servidor HTTP Apache 2.0, 2007. Disponible en:
<http://httpd.apache.org/docs/2.0/es/>
- Dojo, 2007. Disponible en: <http://dojotoolkit.org/demos>
- Dreamweaver8, 2007. Disponible en: <http://es.wikipedia.org/wiki/Dreamweaver>
- EMS soluciones, 2007. Disponible en: <http://www.sqlmanager.net/>
- FRANCO NAVARRO, J. A. UML en acción. Modelando Aplicaciones *Web*, 2005.
- GALLEGO VÁZQUEZ, J. A. Desarrollo Web con PHP y MySQL. S.A, A. M., 2003.
- GILFILLAN, I. La Biblia MySQL., 2002.
- HTML, 2007. Disponible en: http://es.wikipedia.org/wiki/C%C3%B3digo_HTML
- Jasper Report, 2007. Disponible en:
http://www.chuidiang.com/chuwiki/index.php?title=Ejemplo_basico_con_Jasper_Report
- LOPEZ RAZO, J.A .AJAX utilizando dojo, 2006.
- Java Script, 2007. Disponible en: <http://es.wikipedia.org/wiki/JavaScript>
- EGUILUZ PEREZ, J. Introducción a AJAX, 2007. Disponible en: <http://www.librosweb.es/ajax/>
- LARMAN, C. UML y Patrones. Prentice Hall Iberoamericana. 1999. p.
- Manual de referencia de MySQL 5.0, 2007 (revisión 488). Disponible en:
<http://www.calitae.com/manuales/manual-mysql-5.0-es.pdf>
- Manual de PHP, 2006, Disponible en: <http://www.php-es.com/>
- Manual XSLT Mini COMO, 2002, Disponible en: http://xml.utilitas.org/xslt_mini_como.html
- MEYER, BERTRAN, Construcción de software Orientado a Objetos. Volumen 1. La Habana 2006
- Mozilla Firefox, 2007. Disponible en: <http://es.wikipedia.org/wiki/Firefox>
- MYSQL, 2007. Disponible en: <http://mysql.conclase.net/curso/index.php>
- NuSphere PhpED Professional 5.2, 2007. Disponible en:
<http://www.mininova.org/tor/1148201>
- PHP, Disponible en: <http://www.php-es.com/>
- PostgreSQL, 2007. Disponible en: <http://es.wikipedia.org/wiki/PostgreSQL>
- Rational Suite, 2007. Disponible en: <http://www.vico.org/TallerRationalRose.pdf>

Servidor HTTP Apache. Disponible en: http://es.wikipedia.org/wiki/Apache_http_server

Sistemas Gestores de Base de Datos, 2007 (revisión 488). Disponible en:

<http://es.wikipedia.org/wiki/DBMS>

Smarty, 2007. Disponible en: <http://es.wikipedia.org/wiki/Smarty>

XML, 2007. Disponible en: <http://es.wikipedia.org/wiki/XML>

XSLT, 2007. Disponible en: <http://es.wikipedia.org/wiki/XSLT>

REFERENCIAS BIBLIOGRÁFICAS

- 1 Jasper Report, 2007. Disponible en:
http://www.chuidiang.com/chuwiki/index.php?title=Ejemplo_basico_con_Jasper_Report
- 2 HTML, 2007. Disponible en: http://es.wikipedia.org/wiki/C%C3%B3digo_HTML
- 3 Java Script, 2007. Disponible en: <http://es.wikipedia.org/wiki/JavaScript>
- 4 XML, 2007. Disponible en: <http://es.wikipedia.org/wiki/XML>
- 5 PHP, Disponible en: <http://www.php-es.com/>
- 6 Mozilla Firefox, 2007. Disponible en: <http://es.wikipedia.org/wiki/Firefox>
- 7 Servidor HTTP Apache. Disponible en: http://es.wikipedia.org/wiki/Apache_http_server
- 8 Documentación del Servidor HTTP Apache 2.0, 2007. Disponible en:
<http://httpd.apache.org/docs/2.0/es/>
- 9 Descripción de la arquitectura en módulos del Apache, 2007. Disponible en:
<http://www.desarrolloweb.com/articulos/1112.php>
- 10 Sistemas Gestores de Base de Datos, 2007 (revisión 488). Disponible en:
<http://es.wikipedia.org/wiki/DBMS>
- 11 MYSQL, 2007. Disponible en: <http://mysql.conclase.net/curso/index.php>
- 12 PostgreSQL, 2007. Disponible en: <http://es.wikipedia.org/wiki/PostgreSQL>
- 13 Smarty, 2007. Disponible en: <http://es.wikipedia.org/wiki/Smarty>
- 14 Rational Suite, 2007. Disponible en: <http://www.vico.org/TallerRationalRose.pdf>
- 15 Dreamweaver8, 2007. Disponible en: <http://es.wikipedia.org/wiki/Dreamweaver>
- 16 NuSphere PhpED Professional 5.2, 2007. Disponible en:
<http://www.mininova.org/tor/1148201>
- 17 EMS soluciones, 2007. Disponible en: <http://www.sqlmanager.net/>
- 18 Ídem = 16.
- 19 Dojo, 2007. Disponible en: <http://dojotoolkit.org/demos>
- 20 AJAX, 2007. Disponible en: <http://es.wikipedia.org/wiki/AJAX>
- 21 XSLT, 2007. Disponible en: <http://es.wikipedia.org/wiki/XSLT>

ANEXOS

Anexo 1: Modelo del negocio.

Actor del Negocio	Descripción
Usuario de Reporte	Interactúa directamente en el negocio, se beneficia con la obtención de la información contenida en el reporte, puede ser cualquier persona que necesite diseñar u obtener un reporte determinado.

Tabla A1.1 Actores del negocio.

Trabajador del Negocio	Descripción
Gestor de Reportes	Se encarga de realizar todos los procesos contenidos en el negocio, puede se cualquier persona que se encarga del manejo de reportes, por ejemplo un secretario, una bibliotecario, un estadístico.

Tabla A1.2 Trabajadores del negocio.

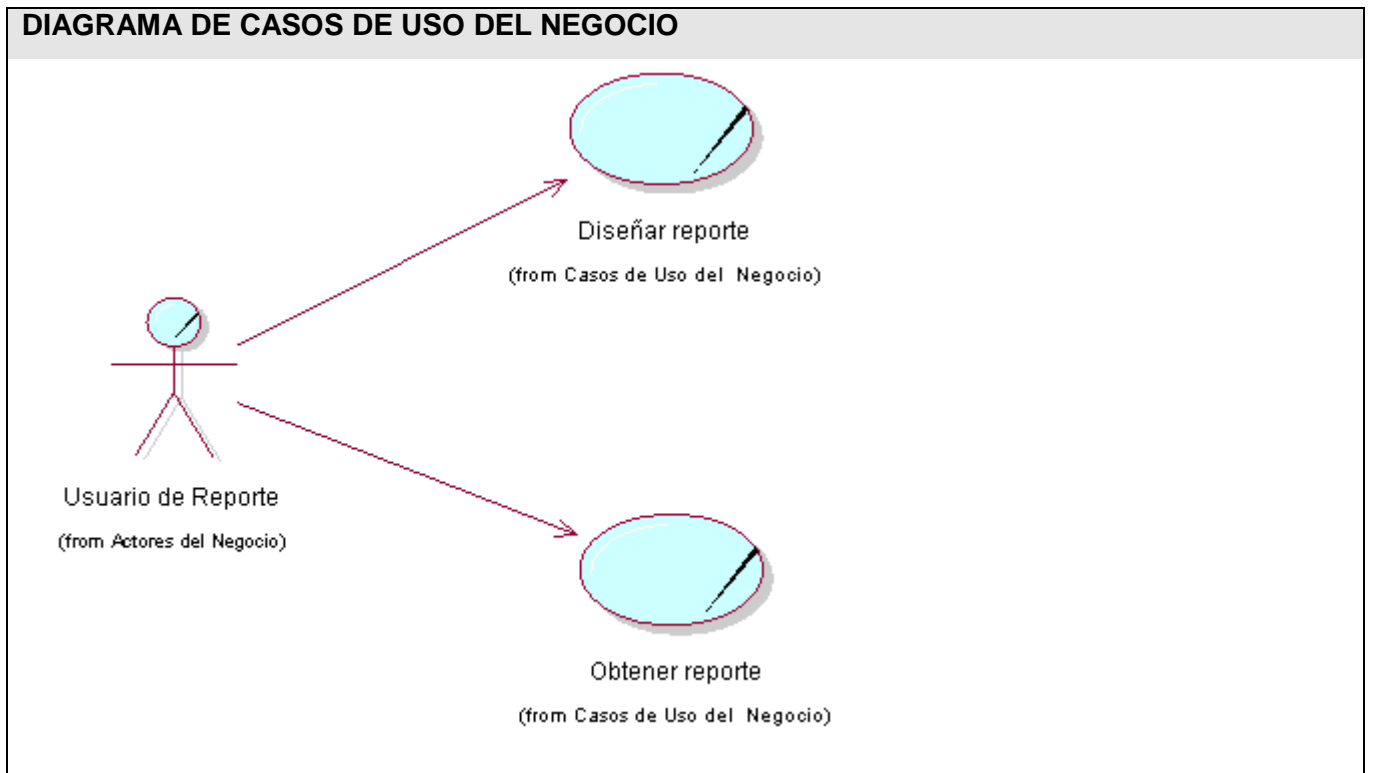


Figura A1.1 Diagrama de casos de uso del negocio.

Modelo de Casos de Uso del Negocio.

Nombre del Caso de Uso	Diseñar reporte
Actores	Usuario de Reporte (inicia)
Trabajadores	Gestor de Reportes.
Propósito	Diseñar un reporte con las especificidades deseadas.
Resumen	El caso de uso se inicia cuando el Usuario de Reporte solicita diseñar un reporte, el Gestor de Reportes recibe la solicitud y le muestra una guía para diseñar el reporte. El Usuario de Reporte diseña el reporte, el Gestor de Reportes lo almacena y el Usuario de Reporte obtiene el reporte deseado.
Curso Normal de los eventos	
Acciones del Actor	Respuesta del proceso de negocio
1. El Usuario de Reporte solicita obtener un reporte.	2. El Gestor de Reportes verifica los permisos de acceso a la base de datos. 3. Si el Usuario de Reporte tiene permiso se muestra el asistente de reporte, si no, ver Flujo Alterno 1.1
	4. El Gestor de Reportes muestra una guía de diseño para ayudar al usuario en la elaboración del reporte.
5. El Usuario de Reporte elabora el reporte.	6. El Gestor de Reportes almacena el reporte elaborado.

Tabla A1.3 Modelo de casos de uso del negocio caso de uso: "Diseñar reporte".

Nombre del Caso de Uso	Obtener reporte
Actores	Usuario de Reporte (inicia)
Trabajadores	Gestor de Reportes.
Propósito	Obtener un reporte diseñado anteriormente.
Resumen	El caso de uso se inicia cuando el usuario solicita Obtener un reporte diseñado, el Gestor de Reportes busca el reporte en la lista de reportes, si el reporte se encuentra, el Gestor de Reportes entrega el reporte al Usuario de Reporte.
Curso Normal de los eventos	

Acciones del Actor	Respuesta del proceso de negocio
1. El Usuario de Reporte solicita obtener un reporte determinado.	2. El Gestor de Reportes busca el reporte en la lista de reportes, si el reporte no se encuentra ver flujo alternativo 1.1, de encontrarse, el gestor de reportes entrega el reporte.
3. El Usuario de Reporte Obtiene el reporte.	
Flujo Alternativo 1.1 "Reporte no encontrado"	
	1. El Gestor de Reportes informa al Usuario de Reporte que no existe el reporte solicitado.
2. El Usuario de Reporte sale de la aplicación.	

Tabla A1.4 Modelo de casos de uso del negocio caso de uso: "Obtener reporte".

Diagramas de actividad.

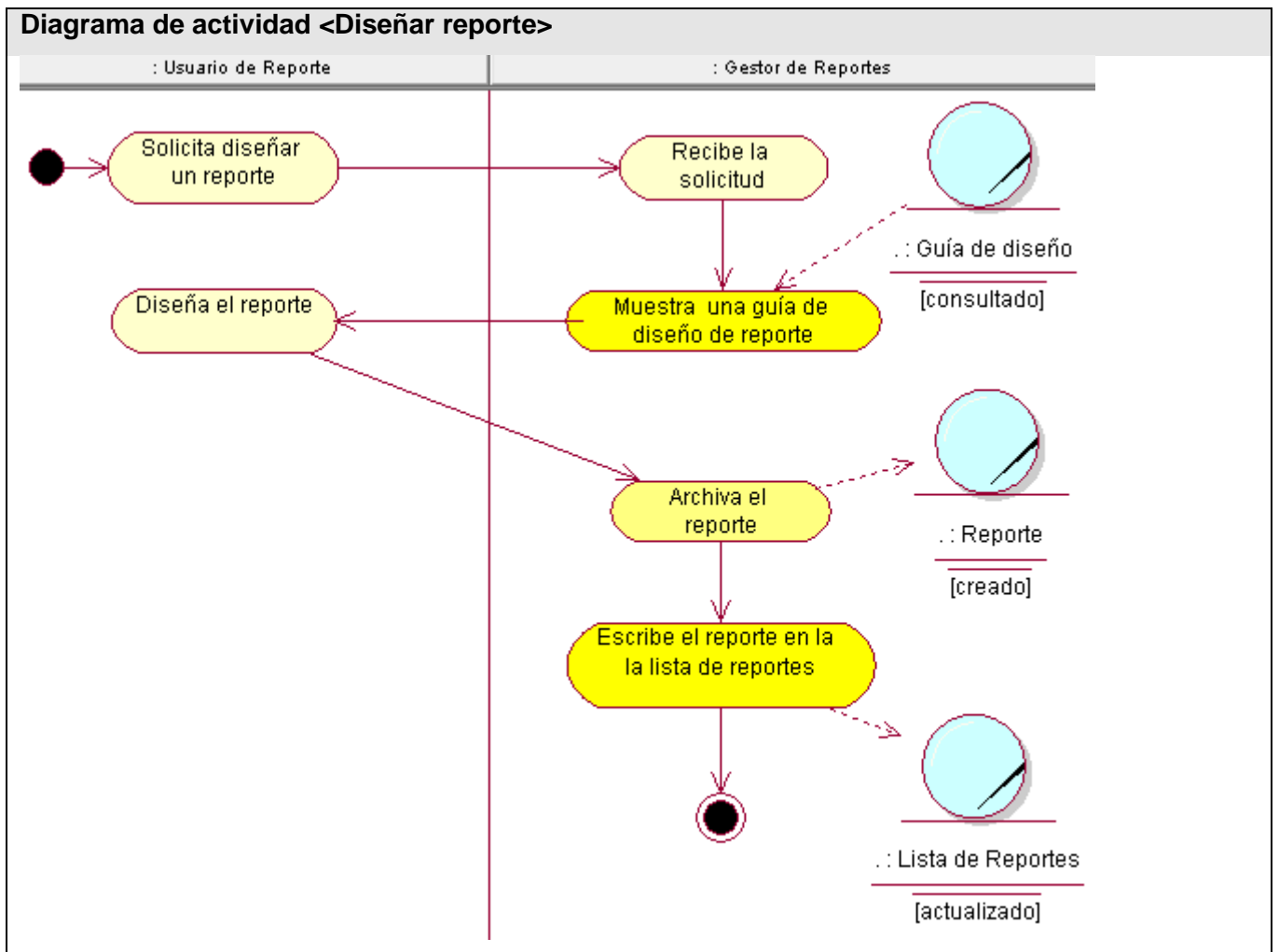


Figura A1.2 Diagrama de actividades: "Diseñar reporte".

Diagrama de actividad <Obtener reporte>

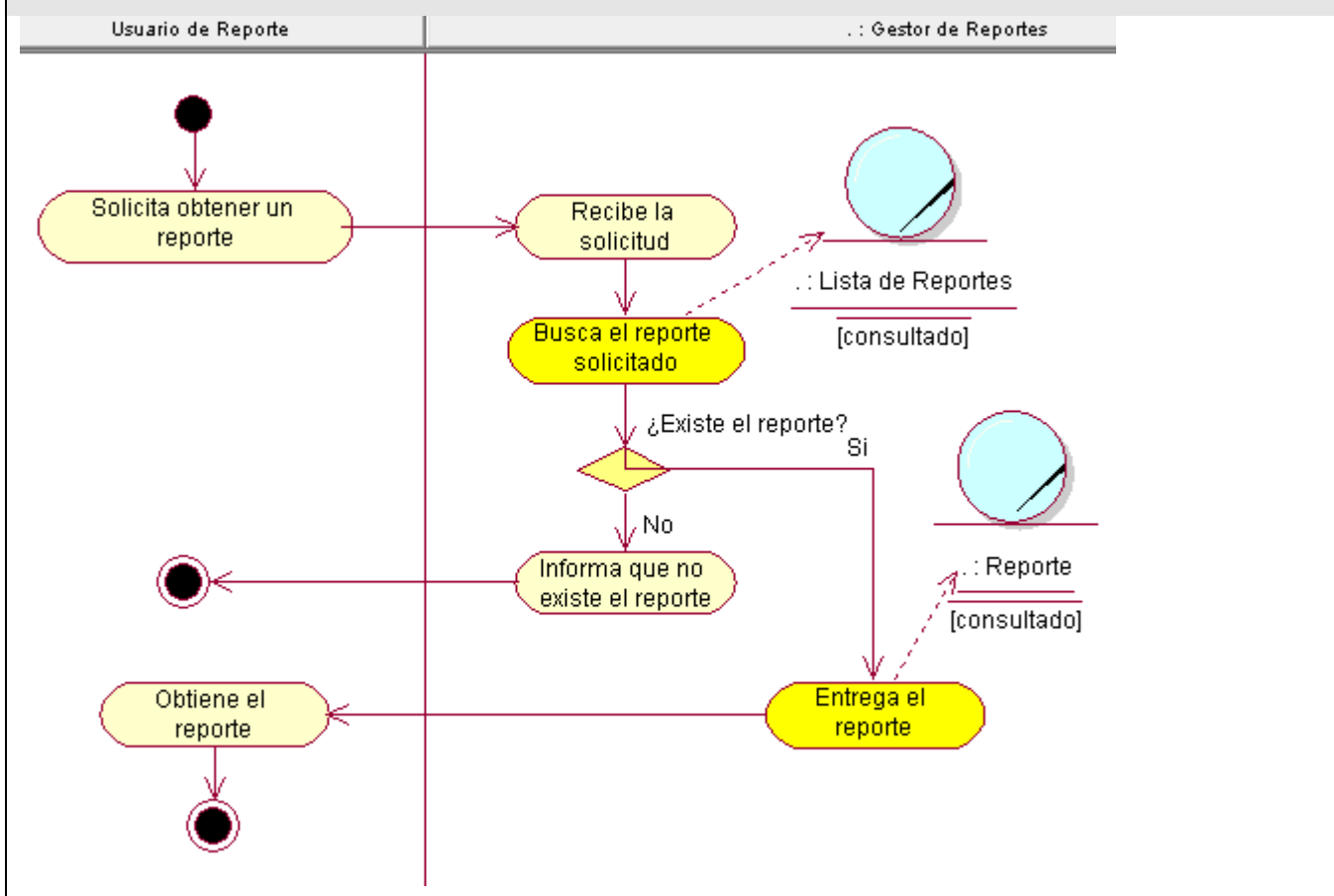


Figura A1.3 Diagrama de actividades: "Obtener reporte".

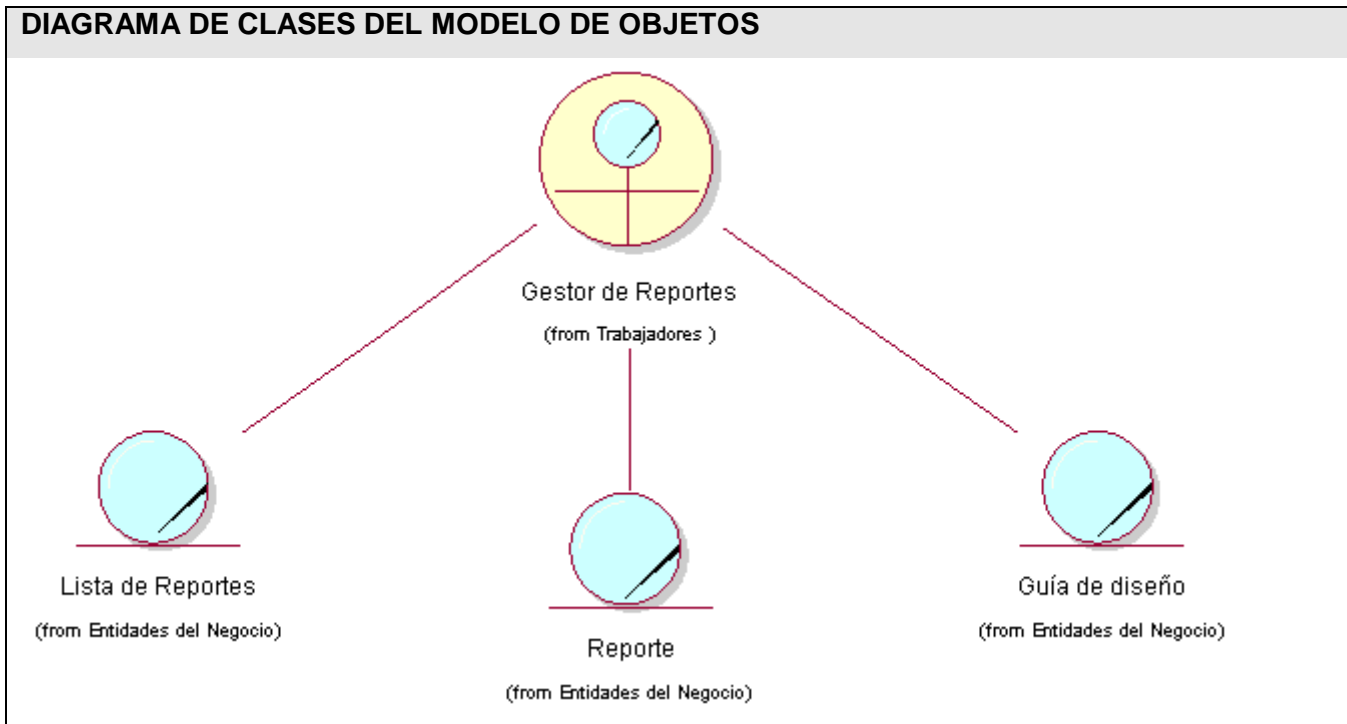


Figura A1.2 Diagrama de clases del modelo de objetos.

Anexo 2: Modelo del sistema.

Actores del sistema

Actor del Sistema	Descripción
Usuario de Reporte	Se beneficia con el diseño y obtención del reporte, puede ser cualquier persona que necesite obtener o diseñar reportes a partir de una base de datos.
Administrador de Reporte	Se beneficia con el diseño y obtención del reporte, debe ser una persona que posea un conocimiento básico de los gestores de base de datos.

Tabla A2.1 Descripción de los actores del sistema.

DIAGRAMA DE CASOS DE USO DEL SISTEMA

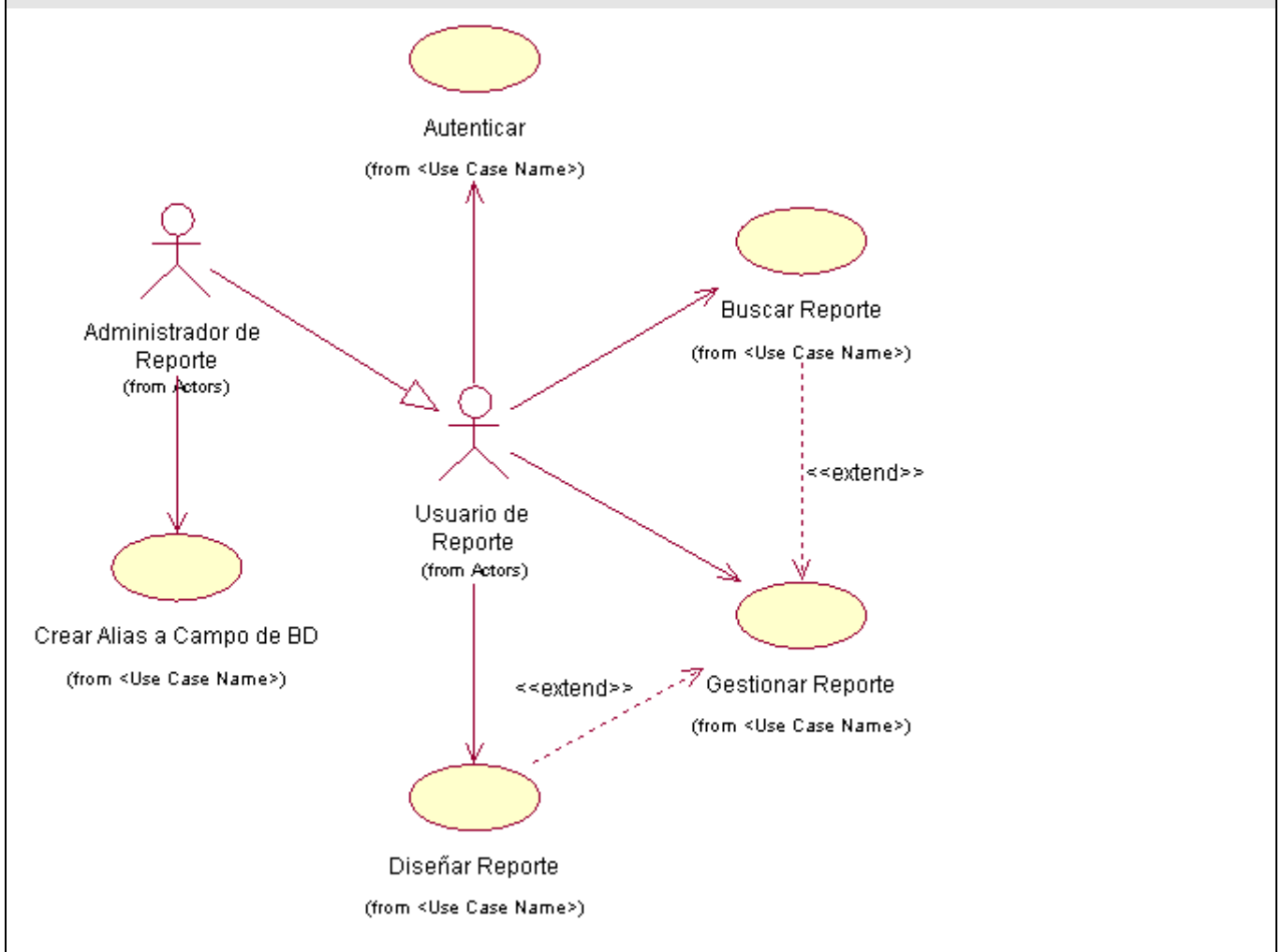


Figura A2.1 Diagrama de casos de uso del sistema.

Descripción de los casos de uso arquitectónicamente significativos

(CUS-1):	Autenticar
Actores:	Usuario de Reporte (inicia), Administrador de Reporte (inicia).
Propósito:	Entrar a la aplicación.
Resumen:	El caso de uso se encarga de confirmar la valides de los usuarios así como de direccionar los mismos a distintas páginas según su roll.
Referencias:	RF1, RF1.1
Precondiciones:	El usuario debe tener los permisos del gestor de base de datos.
Curso normal de eventos	
Acción del Actor	Respuesta del Sistema

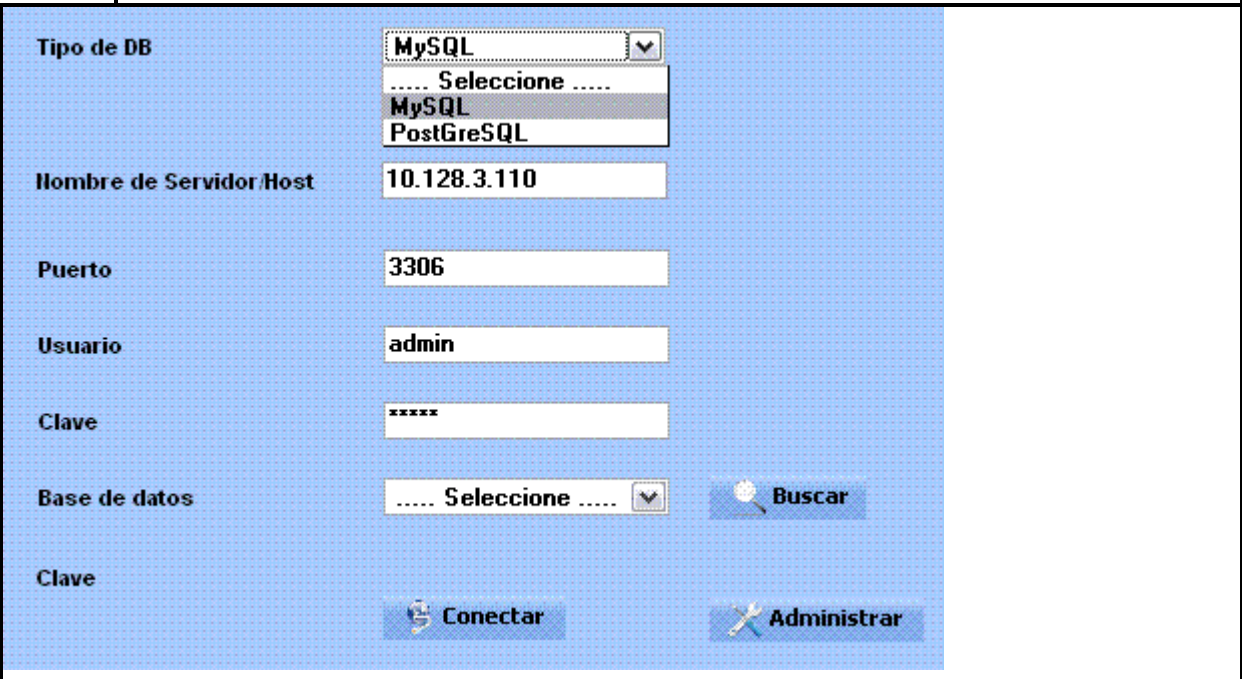
1. El usuario solicita entrar a la aplicación.	2. El sistema muestra las opciones para seleccionar el gestor de base de datos y solicita los permisos de acceso.
3. El usuario se autentica y pulsa el botón conectar (Usuario de Reporte) ó administrar (Administrador de Reporte.)	4. El sistema verifica los permisos: - si no tiene permisos ver flujo alternativo 1. - si el usuario tiene permisos de Usuario de Reporte ver punto de extensión 1. - si el usuario tiene permisos de Administrador ver punto de extensión 2.
Flujo alternativo1	
	1 el sistema informa que no tiene permisos suficientes.
Puntos de extensión:	
1 CUS-3 Diseñar Reporte. 2 CUS-2 Crear Alias a Campo de Base de Datos. 3 CUS-4 Gestionar Reporte.	
Poscondiciones:	El usuario se autenticó.
Prioridad:	Crítico.
Interfaz:	

Tabla A2.2 Descripción de los casos de usos arquitectónicamente significativos: CUS1-Autenticar.


(CUS-2):	Crear Alias a Campo de Base de Datos.	
Actores:	Administrador de Reporte (inicia)	
Propósito:	Crear un alias, a un campo de la base de datos, que sea comprensible para el Usuario de Reporte.	
Resumen:	El caso de uso permite la creación de un alias a un campo de la base de datos con el objetivo de que sea más comprensible para el usuario.	
Referencias:	RF2	
Precondiciones:	El Administrador de Reporte debe haberse autenticado.	
Curso normal de eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario pulsa el botón "Alias".	2. El sistema muestra un explorador con las tablas de la base de datos escogida.	
3. El Administrador de Reporte selecciona una tabla.	4. El sistema muestra los nombres de los campos de la tabla seleccionada así como los datos referentes al alias del campo.	
5. El Administrador de Reporte selecciona un campo y pulsa el botón modificar.	6. El sistema solicita la introducción de los datos referentes al alias.	
6. El administrador de Reportes introduce los datos y pulsa el botón "aceptar"	7. El sistema guarda los datos referentes al alias.	
Puntos de extensión:		
Poscondiciones:	Se creó un alias a un campo de la base de datos.	
Prioridad:	Secundario.	
Interfaz:		

Tabla A2.3 Descripción de los casos de usos arquitectónicamente significativos: CUS2-Crear alias a campo de base de datos.

(CUS-3):	Diseñar Reporte.	
Actores:	Usuario de Reporte (inicia)	
Propósito:	Diseñar el reporte teniendo en cuenta los atributos necesarios, el orden de los mismos y la estructura del reporte.	
Resumen:	El caso de uso representa todo el proceso de diseño del reporte es decir: incluir los atributos necesarios, establecer criterios para incluir un atributo, establecer agrupaciones, adicionar operaciones de cálculo general (suma, promedio, etc.), configurar el estilo del reporte así como guardarlo una vez diseñado, el caso de uso se inicia cuando el Usuario de Reporte pulsa el botón “Nuevo Reporte” del menú principal.	
Referencias:	RF3, RF3.1, RF3.2, RF3.3, RF3.4, RF3.5, RF3.6.	
Precondiciones:	El usuario debe haberse autenticado.	
Curso normal de eventos		
Acción del Actor	Respuesta del Sistema	
1. El Usuario de Reporte pulsa el botón “Nuevo Reporte” del menú principal.	1. El sistema muestra un conjunto de opciones para la confección del reporte.	
3. El Usuario de Reporte elige una opción pulsando el botón ó pestaña correspondiente.	6. El sistema verifica la opción elegida por el Usuario de Reporte. Si eligió la opción de: - “General” ver sección 1. - “Condición” ver sección 2. - “Grupos” ver sección 3. - “Operaciones” ver sección 4. - “Estilos” ver sección 5. - “Guardar Reporte” ver sección 6. - “Generar Reporte” ver sección 7. - “Cancelar Reporte” ver sección 8.	
Aclaración: El usuario puede elegir cualquier otra opción descrita antes de “Generar” ó “Cancelar”, se debe elegir la opción general para tener al menos un atributo en el reporte a elaborar.		
Sección 1: “General”		
	1. El sistema muestra un conjunto de atributos a	

	incluir en el reporte así como la opción de nombrar el reporte.
2. El Usuario de Reporte nombra el reporte así como secciona un atributo del reporte y pulsa el botón "Adicionar"	3. El sistema adiciona, al diseño del reporte el atributo seleccionado, así como lo muestra junto a las opciones de "Deshacer" o de "Deshacer Todo".
4. Si lo desea, el Usuario de Reporte elige una opción mostrada.	5. El sistema verifica la opción. Si el Usuario de Reporte eligió la opción de: "Deshacer" ver flujo alternativo 1.1. "Deshacer Todo" ver flujo alternativo 1.1.
Flujo alternativo 1.1: "Deshacer"	
	1. El sistema elimina el último atributo adicionado al reporte.
Flujo alternativo 1.1: "Deshacer Todo"	
	1. El sistema elimina todos los atributos seleccionados anteriormente al reporte.
Sección 2: "Condición"	
	1. El sistema muestra los atributos seleccionados anteriormente (ver sección 1).
2. El Usuario de Reporte selecciona el atributo deseado para agregarle la condición.	3. El sistema muestra los operadores que se pueden utilizar y los valores que posee el atributo seleccionado
4. El Usuario de Reporte selecciona el tipo de operador y el valor para establecer la condición.	5. El sistema muestra los operadores de condición (y/ó).
6. El Usuario de Reporte selecciona el operador de condición deseado y pulsa el botón "Adicionar".	7. El sistema Adiciona la condición introducida por el usuario al diseño del reporte. Así como brinda la opción "Deshacer".
7. Si lo desea el Usuario de Reporte pulsa el botón "Deshacer"	8. El sistema elimina la última condición adicionada al diseño del reporte.
Sección 3: "Grupos"	
	1. El sistema muestra tres niveles de agrupación y los posibles atributos por que puede agrupar en

	cada nivel.
2. El Usuario de Reporte selecciona los valores deseados para cada nivel (es opcional darle valores a los 3 niveles) y pulsa el botón “Guardar”	3. El sistema guarda la agrupación establecida por el usuario así como brinda la opción “Deshacer”.
4. Si lo desea el Usuario de Reporte pulsa el botón “Deshacer”.	4. El sistema elimina la agrupación establecida anteriormente.
Sección 4: “Operaciones”	
	1. El sistema muestra las posibles operaciones a introducir (suma, promedio, mínimo, máximo) así como los atributos “autocargable” a seleccionar (los atributos “autocargable” deben estar configurados por el Administrador de Reporte en el caso de uso Crear Alias a Campo de BD).
2. El Usuario de Reporte selecciona la operación deseada, el atributo de la misma y pulsa el botón “Adicionar”.	4. El sistema adiciona al diseño la operación seleccionada así brinda la opción “Deshacer”.
5. Si lo desea, el usuario de Reporte pulsa la opción “Deshacer”.	6. El sistema elimina del diseño la última operación introducida por el usuario.
Sección 5: “Estilos”	
	1. El sistema muestra un conjunto de opciones para configurar el estilo de las distintas partes del reporte.
2. El Usuario de Reporte selecciona una parte determinada del reporte pulsando la pestaña correspondiente al nombre de la misma.	3. El sistema muestra las opciones de color y formato para la parte del reporte determinada por el Usuario de Reporte
4. El Usuario de Reporte selecciona el color, la fuente y el tamaño de la misma para la parte del reporte seleccionada.	5. El sistema incluye el estilo seleccionado el generarse el reporte ver sección 7.
Sección 6: “Guardar Reporte”	
	1. El sistema guarda el reporte.
Sección 7: “Generar Reporte”	
	1. Ver punto de extensión 1, sección 1.


Sección 8: "Cancelar Reporte"	
	1. El sistema cancela el diseño del reporte.
Puntos de extensión:	
1 CUS-4 Gestionar Reporte.	
Poscondiciones:	Se ha diseñado el reporte.
Prioridad:	Crítico.
Interfaz:	

Tabla A2.4 Descripción de los casos de usos arquitectónicamente significativos: CUS3-Diseñar reporte.

(CUS-4):	Gestionar Reporte.
Actores:	Usuario de Reporte (inicia)
Propósito:	Gestionar los reportes diseñados.
Resumen:	El caso de uso representa todo el proceso de gestión de los reportes diseñados o sea: mostrar un reporte, eliminarlo y exportarlo a otros formatos como Microsoft Word, Microsoft Excel y PDF, el caso de uso se inicia cuando el Usuario de Reporte pulsa la pestaña "Historial" del menú principal.
Referencias:	RF4, RF4.1, RF4.2, RF4.3.
Precondiciones:	El Usuario de Reporte debe haber guardado el reporte diseñado.
Curso normal de eventos	


Acción del Actor	Respuesta del Sistema
1. El Usuario de Reporte pulsa la pestaña “Historial” del menú principal.	2. El sistema muestra una serie de reportes así como las opciones de “Mostrar Reporte” y “Eliminar Reporte” a partir de un reporte seleccionado.
3. El Usuario de Reporte elige un reporte y selecciona una opción.	4. El sistema verifica la opción seleccionada, si el Usuario de Reporte seleccionó la opción de: - “Mostrar Reporte” ver sección 1. - “Eliminar Reporte” ver sección 2.
Sección 1: “Mostrar Reporte”	
	1. El sistema muestra el reporte seleccionado así como la opción de exportarlo a otro formato.
2. Si lo desea, el Usuario de Reporte pulsa el botón que representa el tipo de formato al cual se desea exportar el reporte.	3. El sistema verifica el formato especificado por el usuario y abre el reporte en dicho formato.
Sección 1: “Eliminar Reporte”	
	1. El sistema elimina el reporte seleccionado.
Puntos de extensión:-	
Poscondiciones:	El reporte se gestionado (mostrado, exportado ó eliminado)
Prioridad:	Crítico.
Interfaz:	

Tabla A2.5 Descripción de los casos de usos arquitectónicamente significativos: CUS4-Gestionar reporte.

(CUS-5):	Buscar Reporte.
Actores:	Usuario de Reporte (inicia).


Propósito:	Buscar los reportes que cumplan una característica específica.	
Resumen:	El caso de uso representa el proceso de búsqueda de reportes que cumplen con una característica determinada, el caso de uso se inicia cuando el Usuario de Reporte pulsa la pestaña “Buscar” del menú principal.	
Referencias:	RF5	
Precondiciones:	El Usuario de Reporte debe haber guardado el reporte diseñado.	
Curso normal de eventos		
Acción del Actor	Respuesta del Sistema	
1. El Usuario de Reporte pulsa la pestaña “Buscar” del menú principal.	2. El sistema muestra la opción de introducir un criterio de búsqueda y una palabra clave.	
3. El Usuario de Reporte selecciona un criterio de búsqueda (nombre, fecha, hora) e introduce una palabra clave.	4. El sistema verifica busca los reportes que posean parte de la palabra clave según el criterio de búsqueda seleccionado. 5. Si existen reportes con esa característica el sistema muestra los reportes (ver punto de extensión 1), si no existen, ver flujo alternativo 1.	
Flujo alternativo 1		
	1. El sistema informa que no existen reportes con esa característica.	
Puntos de extensión:		
1 CUS-5 Gestionar Reporte.		
Poscondiciones:	El reporte se ha buscado.	
Prioridad:	Secundario.	
Interfaz:		

Tabla A2.6 Descripción de los casos de usos arquitectónicamente significativos: CUS1-Buscar.

Diagramas de interacción

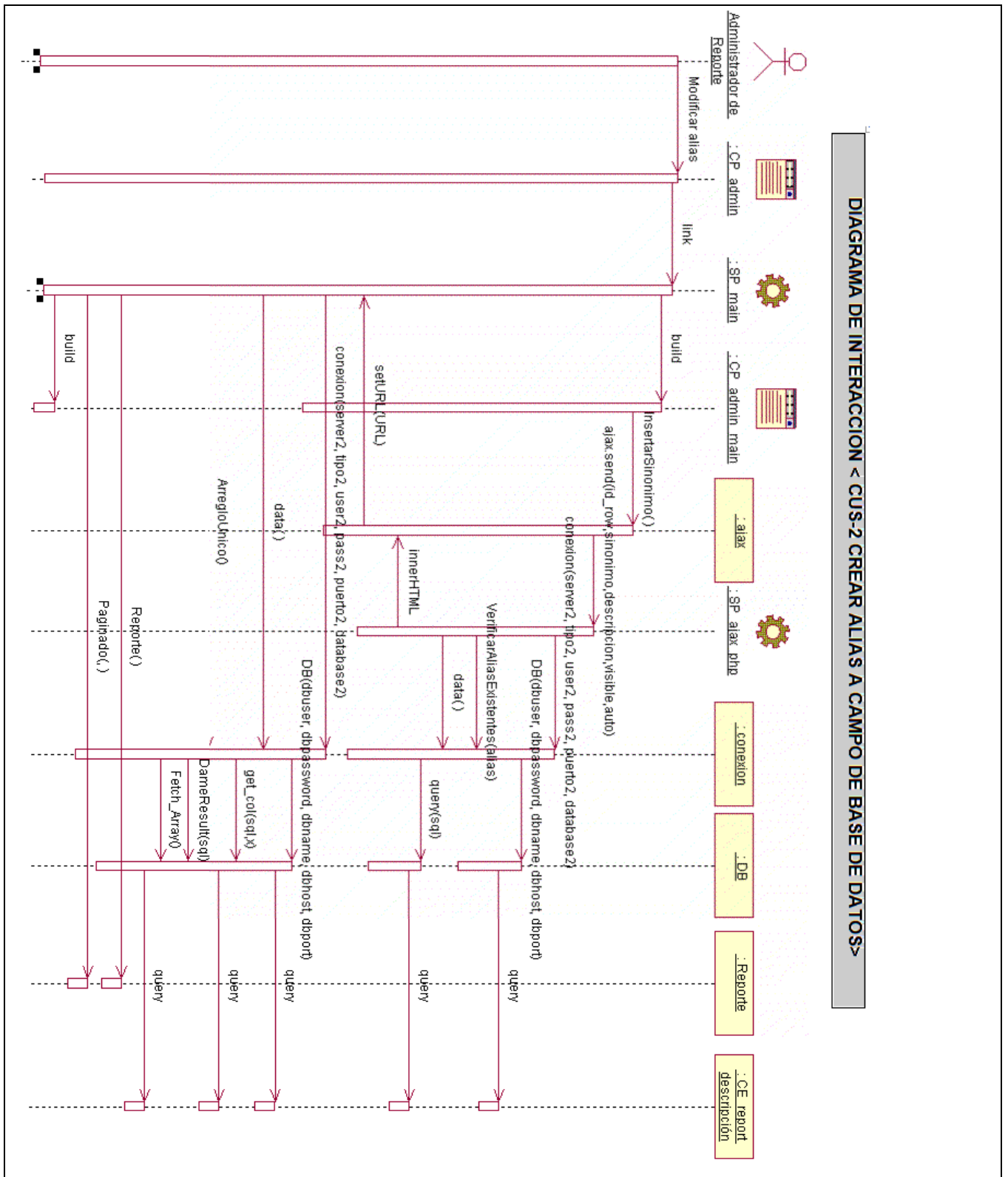


Figura A2.2: Diagrama de interacción CUS2-Crear alias a campo de base de datos.

DIAGRAMA DE INTERACCIÓN <CUS-3 DISEÑAR REPORTE>

<sección general>

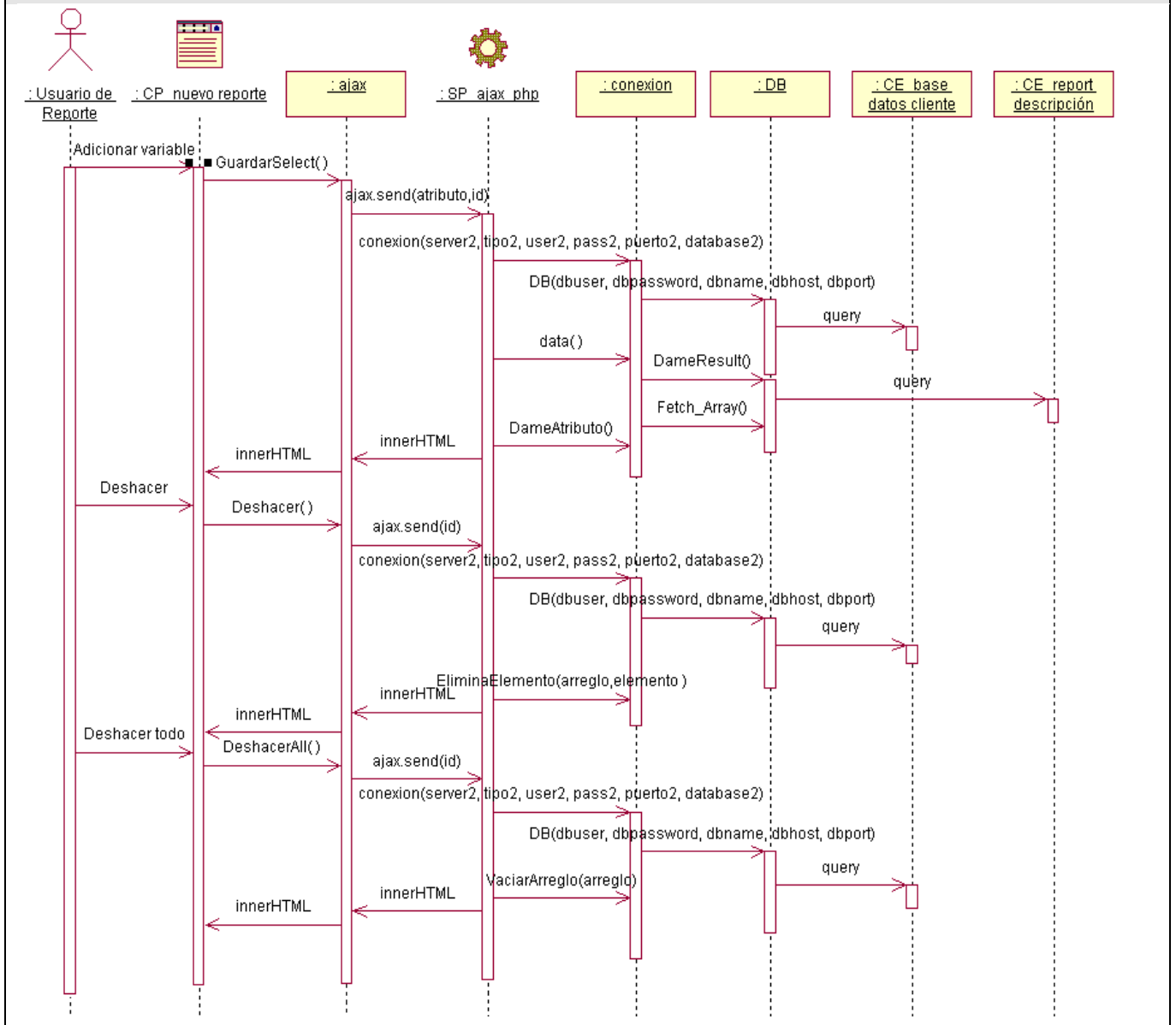


Figura A2.3: Diagrama de interacción CUS3-Diseñar reporte, sección General.

DIAGRAMA DE INTERACCION <CUS-3 DISEÑAR REPORTE>

<sección condición>

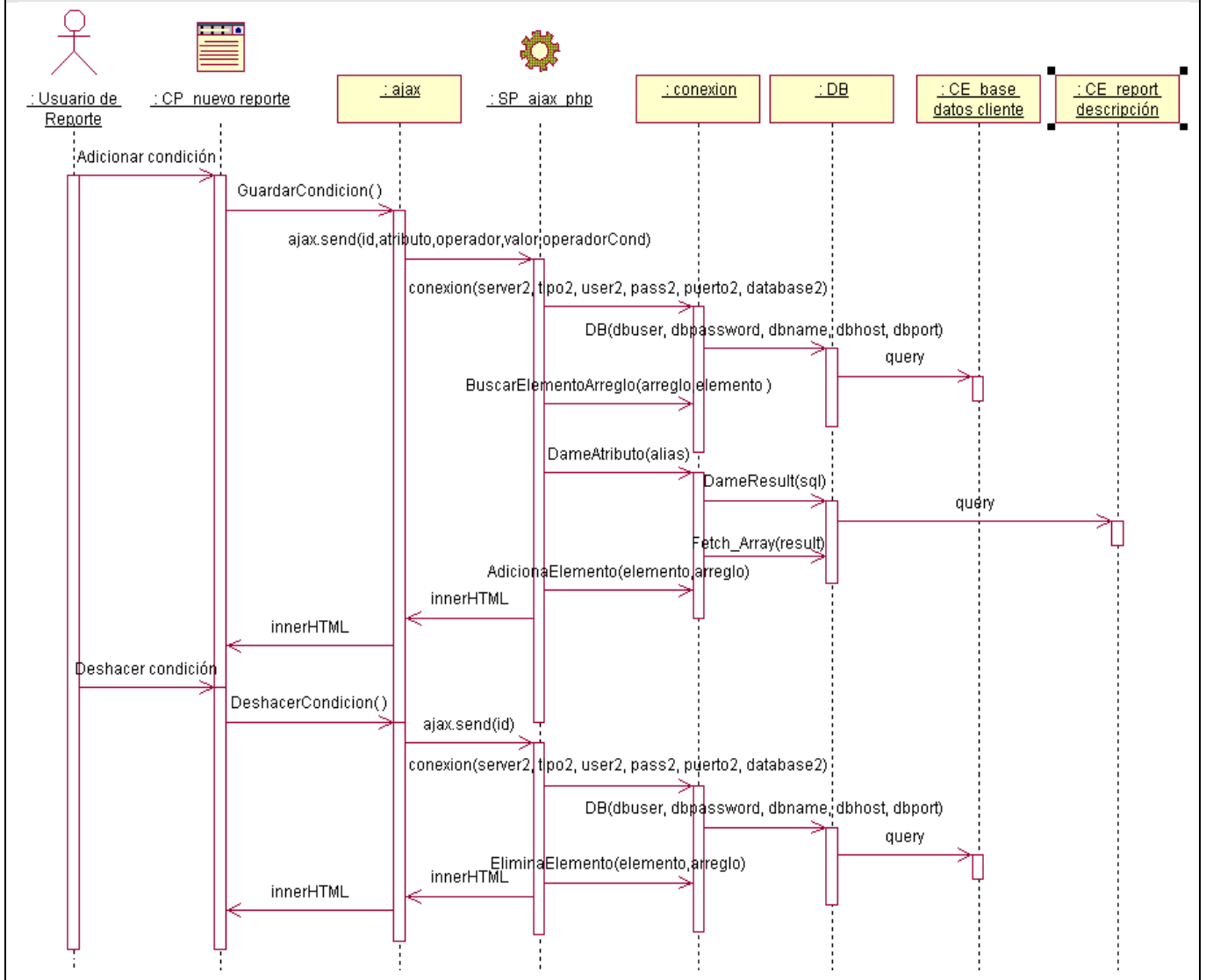


Figura A2.3: Diagrama de interacción CUS3-Diseñar reporte, sección Condición.

DIAGRAMA DE INTERACCION <CUS-3 DISEÑAR REPORTE>

<sección grupos>

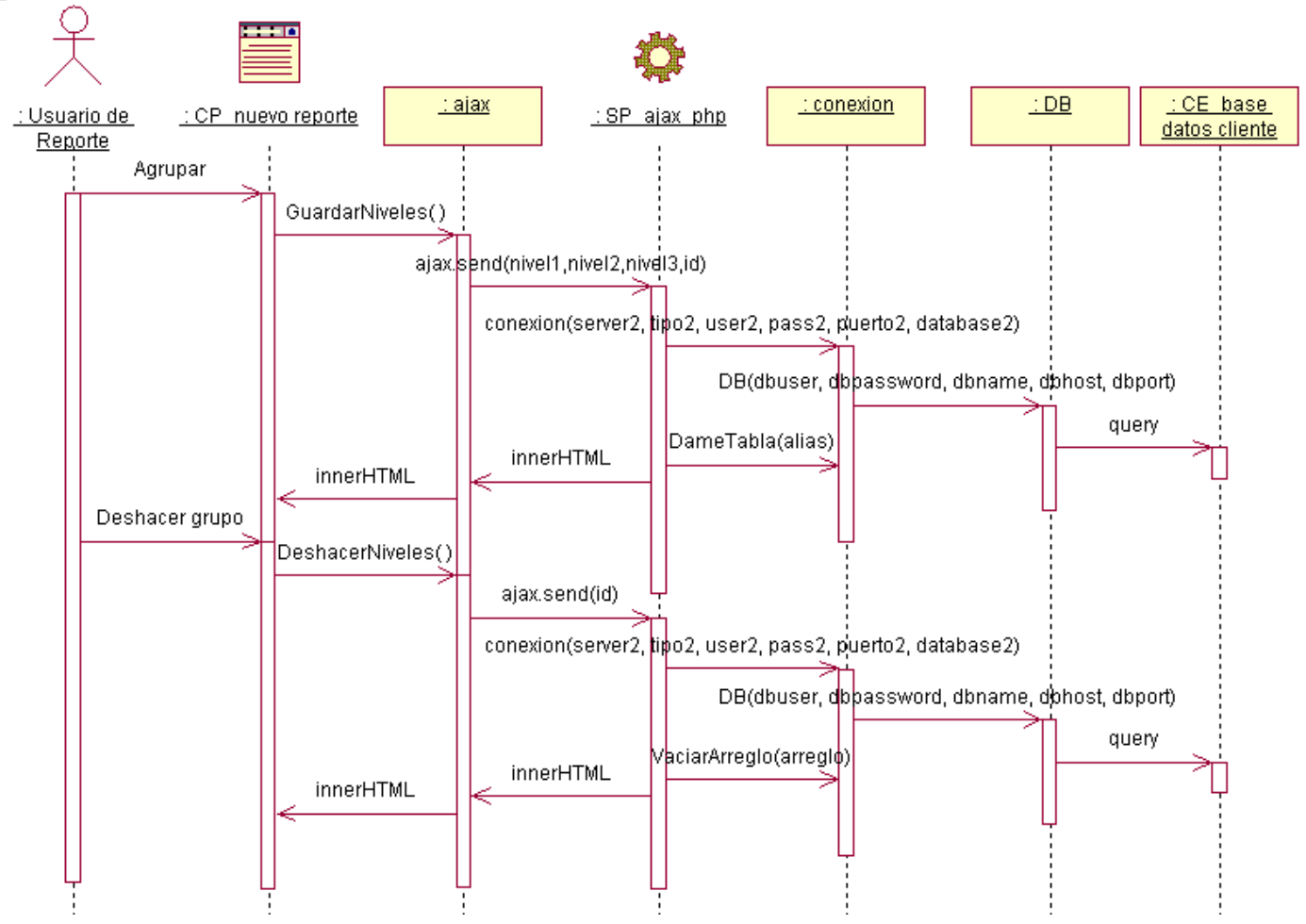


Figura A2.4: Diagrama de interacción CUS3-Diseñar reporte, sección Grupos.

DIAGRAMA DE INTERACCIÓN <CUS-3 DISEÑAR REPORTE>

<sección operaciones>

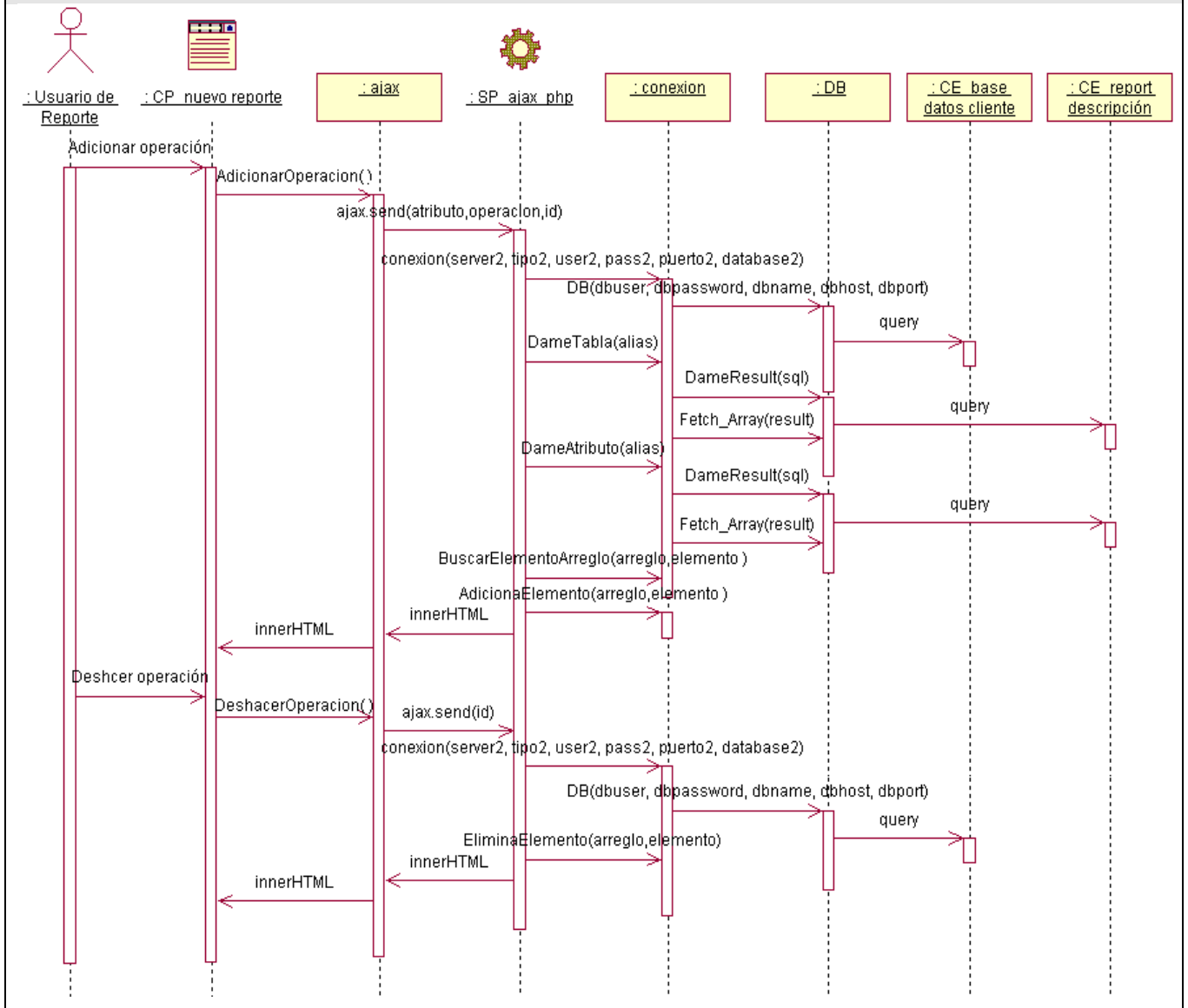


Figura A2.5: Diagrama de interacción CUS3-Diseñar reporte, sección Operaciones.

DIAGRAMA DE INTERACCION <CUS-3 DISEÑAR REPORTE>

<sección estilos>

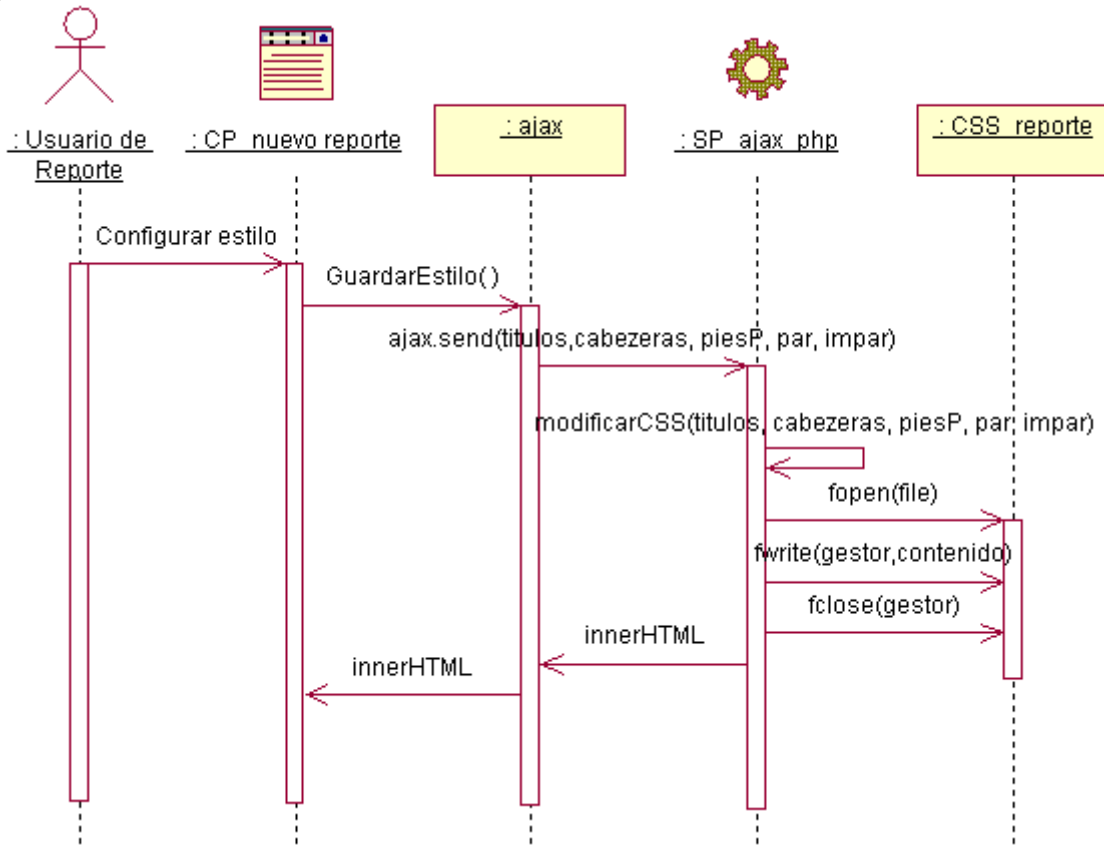


Figura A2.6: Diagrama de interacción CUS3-Diseñar reporte, sección Estilos.

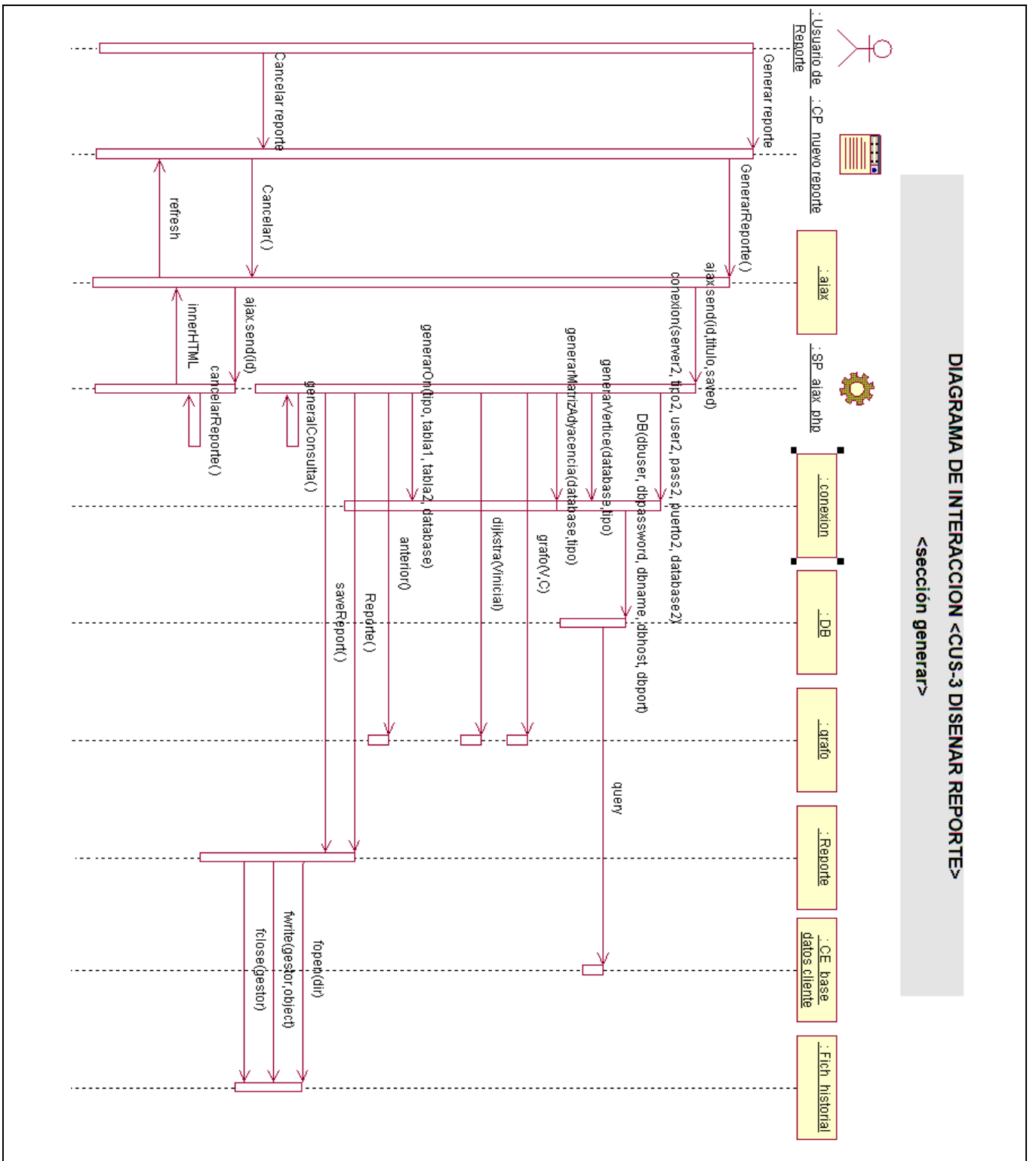


Figura A2.7: Diagrama de interacción CUS3-Diseñar reporte, sección Generar.

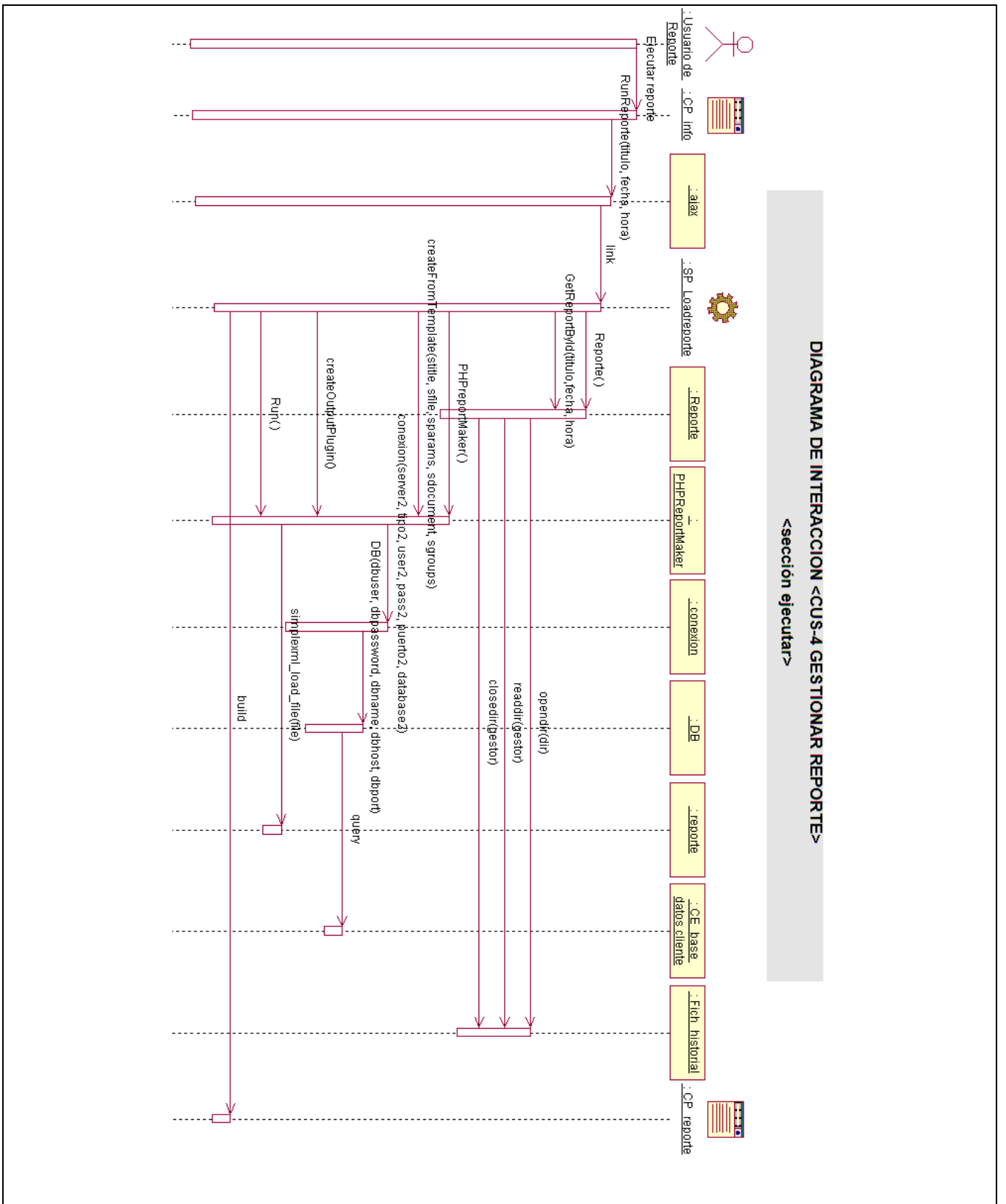


Figura A2.8: Diagrama de interacción CUS4-Gestionar reporte, sección Ejecutar.

DIAGRAMA DE INTERACCION <CUS-4 GESTIONAR REPORTE>

<sección eliminar>

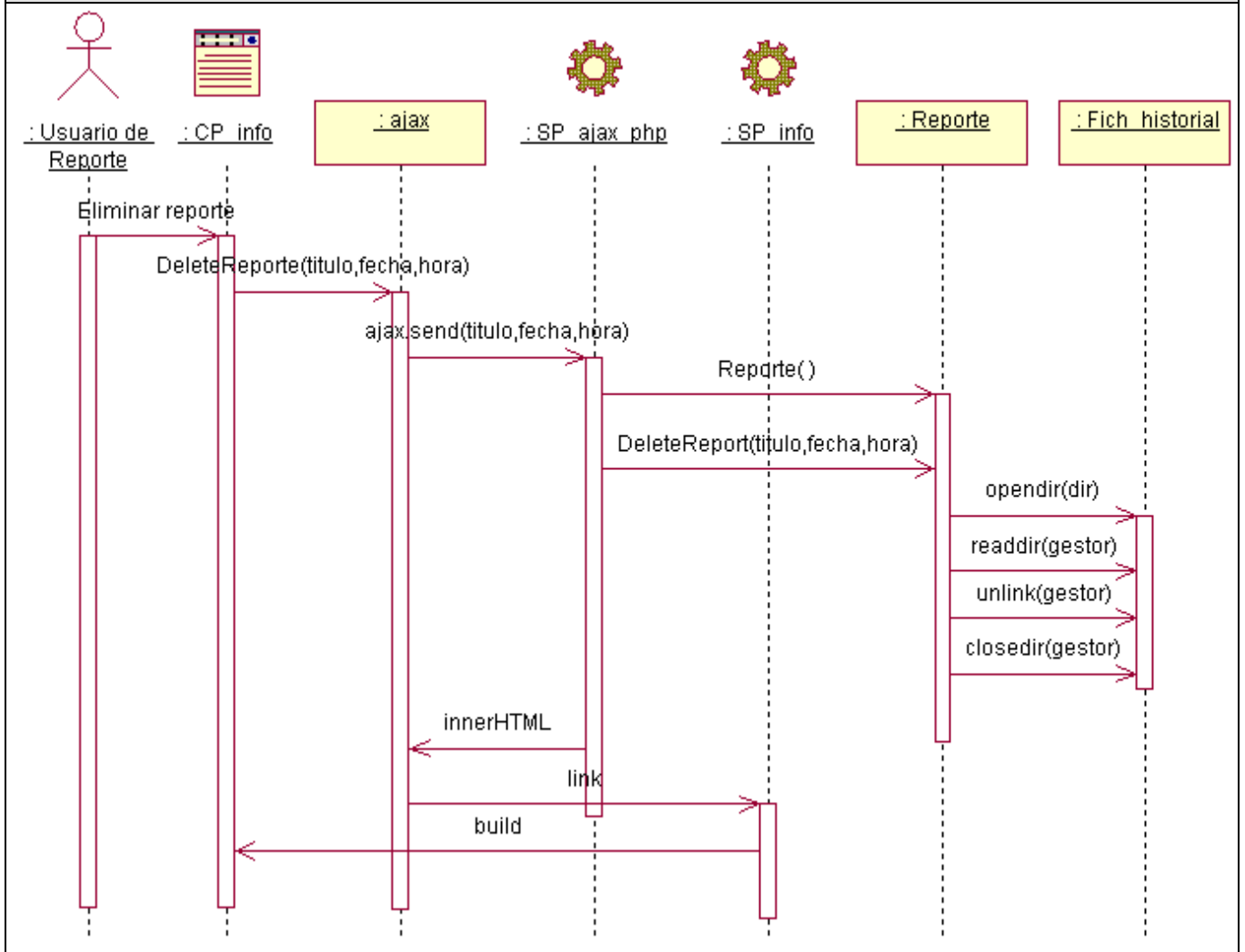


Figura A2.9: Diagrama de interacción CUS4-Gestionar reporte, sección Eliminar.

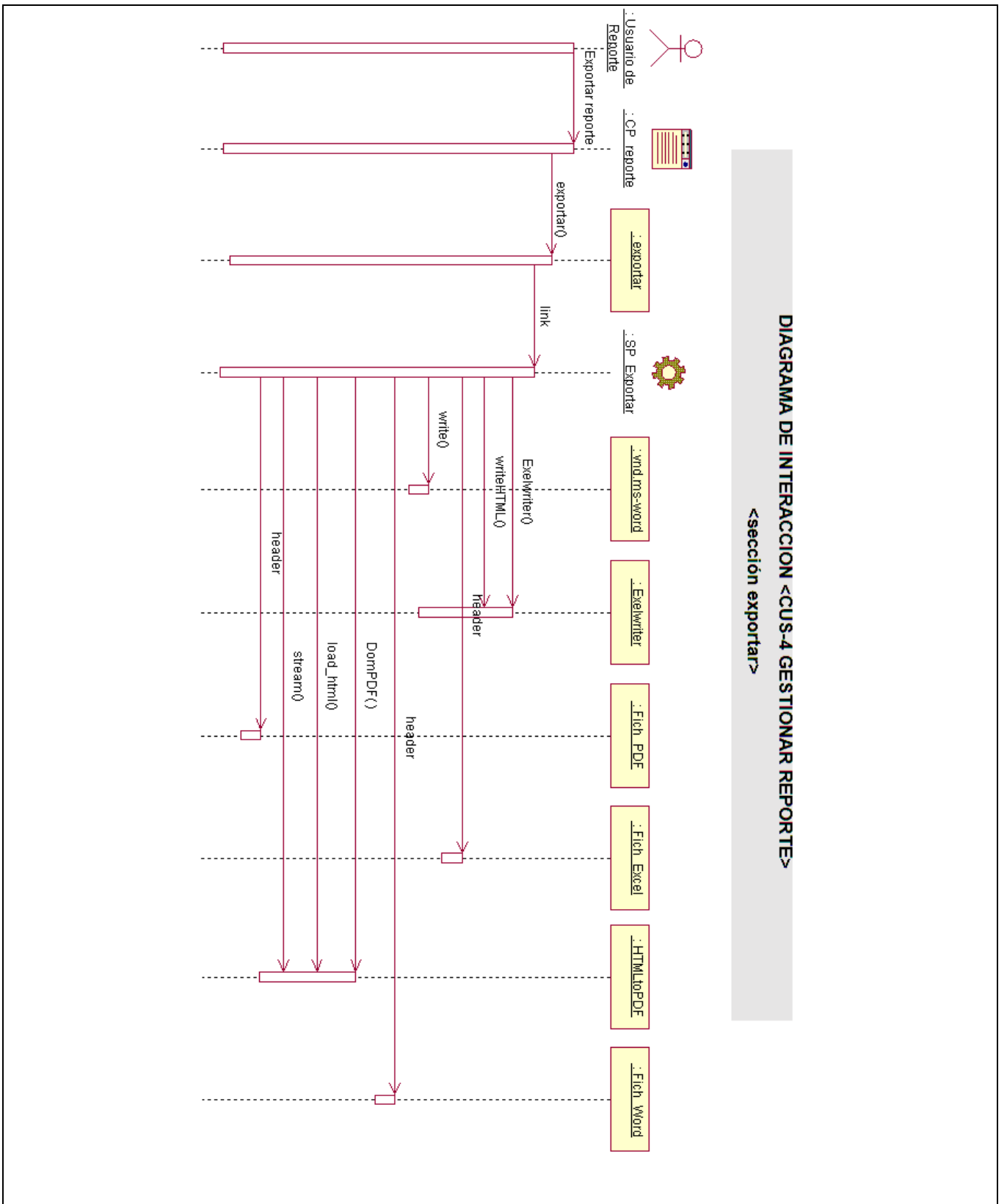


Figura A2.10: Diagrama de interacción CUS4- Gestionar reporte, sección Exportar.

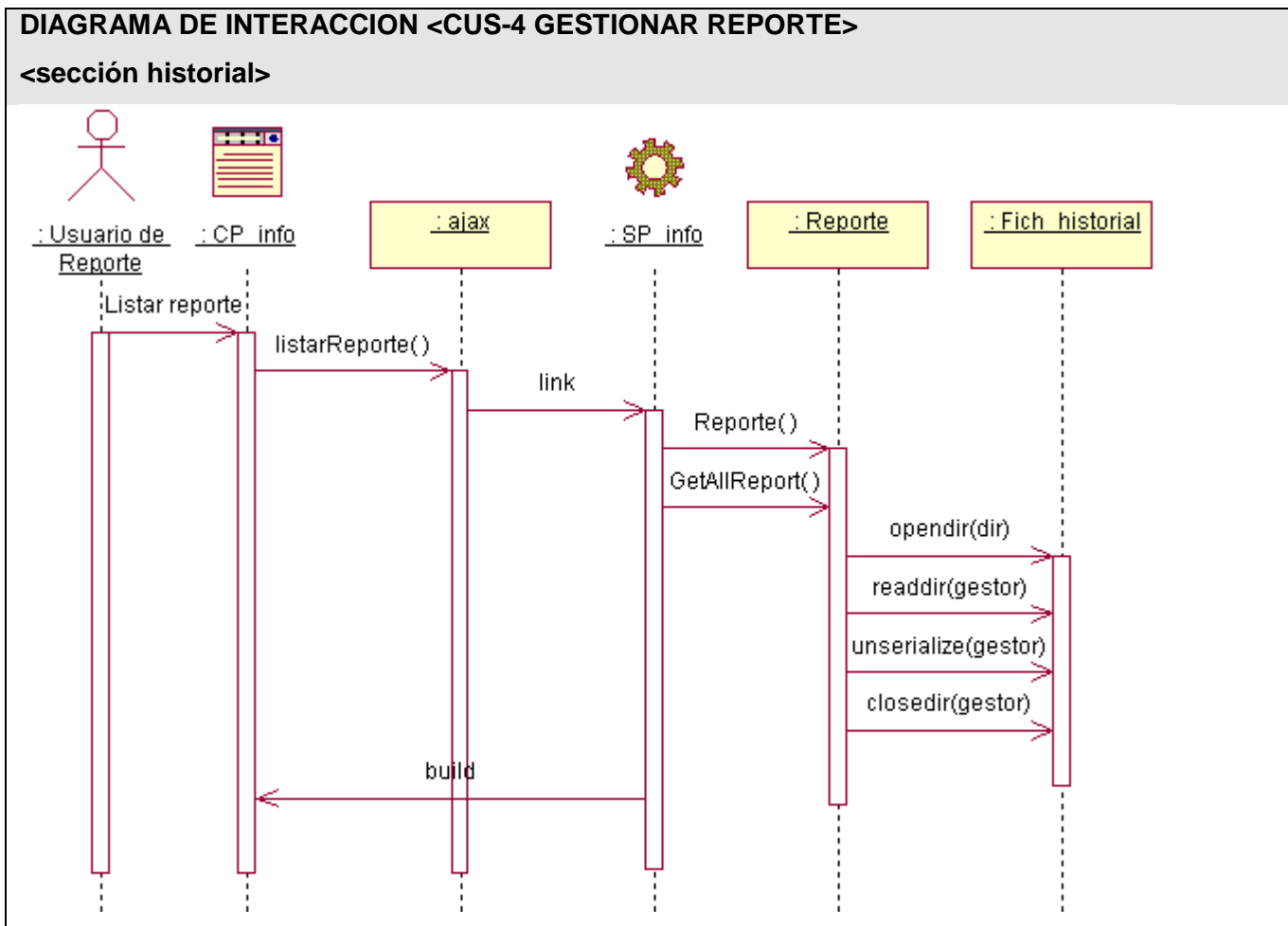


Figura A2.11: Diagrama de interacción CUS4- Gestionar reporte, sección historial.

DIAGRAMA DE INTERACCION <CUS-5 BUSCAR REPORTE>

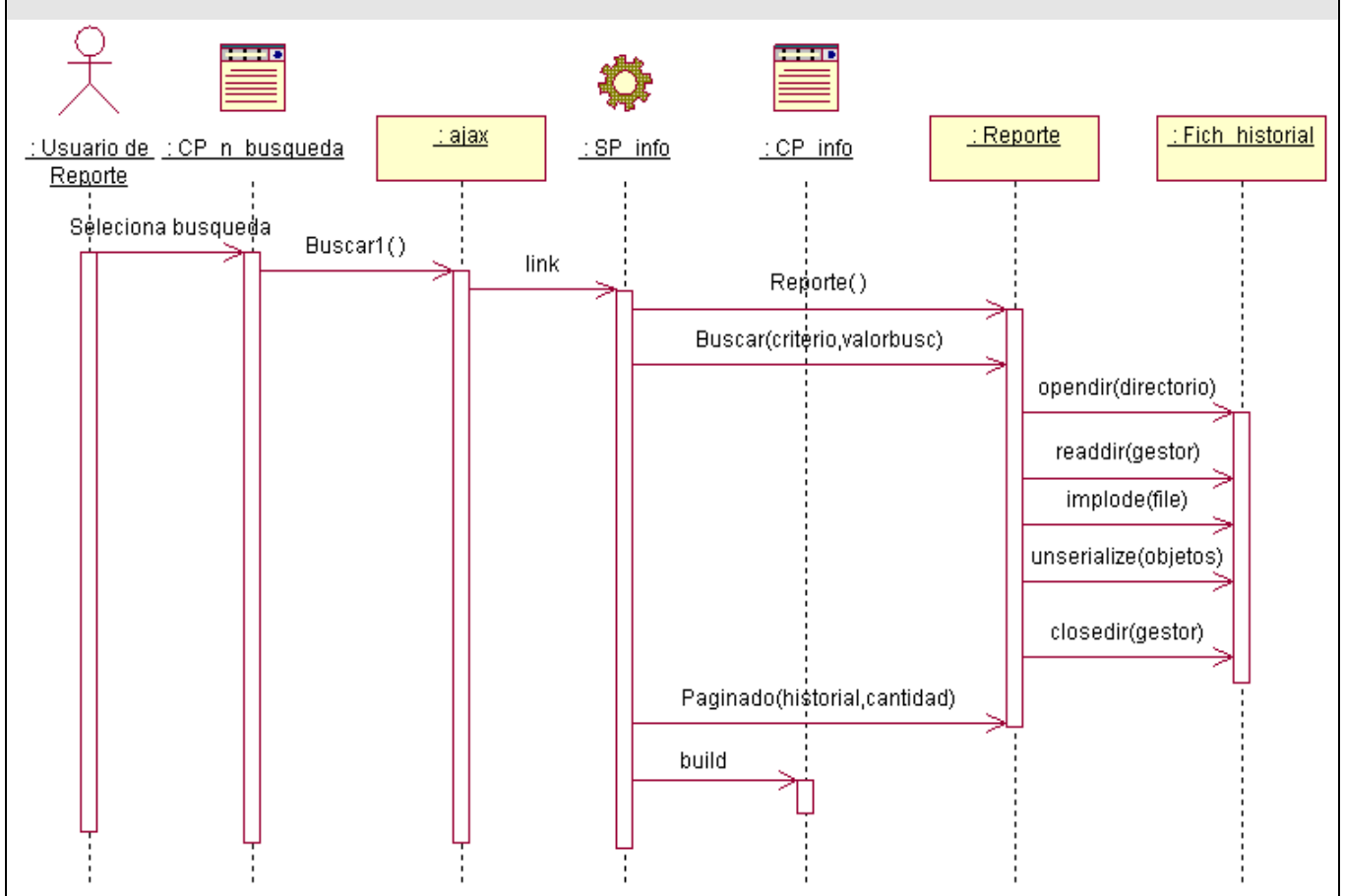


Figura A2.10: Diagrama de interacción CUS5-Buscar reporte.

Anexo 3. Aval de introducción de resultados.

Aval de introducción de resultados del producto *RPHP* dentro del Sistema para la Gestión de la Colaboración Médica


Proyecto	Reporteador
Producto	RPHP
Desarrolladores:	Lic Ernesto Sarduy Alonso, Pavel Manzanares Angulo, Roberlán Hernández Pérez
Area temática	Sistemas de apoyo a la Salud
Polo	Aplicaciones para la Salud
Jefe de área temática	Yasel Couce Sardiñas
Jefe Polo	Reinier Alonso


Importancia del resultado de este trabajo para el control de la colaboración médica cubana en el mundo.

Con el uso de la herramienta RPHP, se le ha brindado al usuario, en este caso a la Unidad Central de Cooperación Médica (UCCM), la cual es la encargada de llevar el control de la colaboración médica de nuestro país en el extranjero, la posibilidad de manejar de manera cómoda y sencilla la información que contiene sus bases de datos relacionadas con la actividad de los colaboradores. Posibilitando que se pueda de esta forma obtener información cualificada a modo de reportes, para la toma de decisiones que son muy necesarias y en muchos casos expeditas. La herramienta permite obtener reportes que luego pueden exportarse a diferentes formatos entre ellos a PDF, Excel y Word lo cual proporciona gran flexibilidad. A través de esta herramienta se puede obtener reportes muy variados y dinámicos, que informan sobre el estado de un colaborador, es decir si este está, cumpliendo misiones, de vacaciones, fallecido o a la espera para salir a cumplir una misión, agruparlos por provincias, municipios o cualquier otro criterio, especificar condiciones sobre las variables etc.

Lugar donde se ha introducido el producto.

La herramienta se ha introducido como parte de un desarrollo llevado a cabo por la Universidad de las Ciencias Informáticas, en conjunto con la Unidad Central de Cooperación Médica ((UCCM))


Lic Roberto Acosta
Lider del Proyecto Colaboración



GLOSARIO

Alias: nuevo nombre que recibe un campo de una base de datos escogida para la realización de los reportes.

API: Interfaz de Programación de Aplicaciones es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Aplicación Web: Sistema informático utilizado por los usuarios mediante el acceso a un servidor web a través de internet o de una intranet.

Bug: defecto de software, es el resultado de un fallo o deficiencia durante el proceso de creación de programas (software).

CGI: Interfaz de pasarela común (*Common Gateway Interface*, CGI), es un protocolo genérico que permite extender las capacidades de HTTP, añade funcionalidad al servidor Web, funcionalidad que podría abrir agujeros de seguridad en el servidor, ya que una aplicación en CGI mal diseñada podría permitir acceso total o parcial al servidor.

Dijkstra, algoritmo: también llamado algoritmo de caminos mínimos, es un algoritmo para la determinación del camino más corto dado un vértice origen al resto de vértices en un grafo dirigido y con pesos en cada arista.

Driver: Controlador de Dispositivo, programa informático que permite al sistema operativo interactuar con un periférico haciendo una abstracción del hardware y proporcionando una interfaz para usarlo.

Etiqueta: marca con tipo que delimita una región en los lenguajes basados en XM; Dentro de un documento de hipertexto, una instrucción. Una etiqueta de hipertexto empieza con un carácter "<" y termina con un carácter ">".

Framework: estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Grafo: conjunto de objetos llamados vértices o nodos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos de un conjunto.

HTTP: Protocolo de Transmisión Hipertexto, protocolo de comunicaciones utilizado por los programas clientes y servidores de WWW para comunicarse entre sí.

IDE: Entorno Integrado de Desarrollo, es un programa compuesto por un conjunto de herramientas para un programador, puede dedicarse en exclusiva a un solo lenguaje de programación o bien, poder utilizarse para varios.

SAS: Sistema de Apoyo a la Salud, área temática de la facultad siete de la Universidad de las Ciencias Informáticas.

Script: guión o conjunto de instrucciones que permiten la automatización de tareas creando pequeñas utilidades.

Servidor: computador en el que se ejecuta un programa que realiza alguna tarea en beneficio de otras aplicaciones llamadas clientes, tanto si se trata de un ordenador normal, un ordenador central, un miniordenador, un ordenador personal ó un sistema integrado.

SGBD: Sistema Gestor de Base de Datos, tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan; se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

SQL: Lenguaje de Consulta Estructurado, lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

TCP/IP: familia de protocolos de Internet, su nombre se debe a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP).

Trigger: disparador en una base de datos, evento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción (INSERT), actualización (UPDATE) o borrado (DELETE).

UML: Lenguaje de Modelado de Sistemas de Software, es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software, se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

URL: Localizador Uniforme de Recurso, secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

W3C: siglas de World Wide Web Consortium, consorcio internacional fundado en 1994, donde las organizaciones miembro, personal a tiempo completo y el público en general, trabajan conjuntamente para desarrollar estándares Web.