

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 1



Subsistema para
el manejo y reportes de credenciales

TRABAJO DE DIPLOMA PARA OPTAR POR EL
TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

AUTOR: **Dasiel Otero Dartayet**

TUTOR: **Lic. Joel Arencibia Ramírez**

Ciudad de La Habana, 22 de junio de 2007

“Año 49 de la Revolución”

AGRADECIMIENTOS

Agradezco a mis padres por su amor e incondicionalidad.

A mi familia.

A Mairelys, por todo.

A Pepe, alias Darien por su invaluable ayuda.

A mi tutor por su ayuda.

A todos mis amigos y compañeros, los de aquí y los de La Sierpe por ser y estar.

DEDICATORIA

A mis padres...

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Dirección de Informatización de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2007.

Dasiel Otero Dartayet

Lic. Joel Arencibia Ramírez

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: **Subsistema para el manejo y reportes de credenciales.**

Autor: Dasiel Otero Dartayet.

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero en Ciencias Informáticas; y propongo que se le otorgue al Trabajo de Diploma la calificación de ____ puntos.

____ días del mes de _____ del año 2007.

Lic. Joel Arencibia Ramírez

RESUMEN

Desde los inicios de la Universidad de las Ciencias Informáticas (UCI), se han desarrollado una serie de sistemas que manejan las credenciales de acceso a la misma. Para lograr ese fin se han desarrollado aplicaciones que se encargan del proceso de creación y modificación de estas. No obstante al intento por perfeccionar este servicio, en estos momentos el proceso de obtención de reportes sobre la acreditación de las personas que estudian o trabajan dentro de la Universidad de las Ciencias Informáticas, no satisface las necesidades del personal encargado del mismo. Lo anteriormente planteado se debe en gran medida al hecho de que han sido soluciones para dar solución a problemas inmediatos. Para darle respuesta a los problemas existentes dentro del área de acreditación de la UCI se desarrolló el presente trabajo.

El “Subsistema de manejo y reportes de credenciales” desarrolla el análisis, diseño e implementación de una aplicación Web que permita obtener reportes necesarios al área de acreditación, así como modificar los diferentes estados que pueda presentar una credencial, para ello se hace un análisis del entorno de trabajo y se muestran los requerimientos hechos por el cliente. Además se hace un análisis de costo beneficio muy importante a la hora de demostrar la viabilidad del proyecto. El trabajo está estructurado en cuatro capítulos que ayudan a la hora de comprender el proceso llevado a cabo. Como resultado del mismo se implementó el Subsistema de manejo y reportes de credenciales, que actualmente se encuentra en estado de pruebas, con una capacidad operacional que le permite responder a los requerimientos formulados por el cliente.

Palabras claves: Reportes, Sistema, Credenciales

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
INTRODUCCIÓN	5
1.1 ESTADO DEL ARTE.....	5
1.1.1 SISTEMAS DE REPORTES EN EL MUNDO.....	5
1.1.2 OBTENCIÓN DE REPORTES EN CUBA.....	6
1.1.3 PARTICULARIDADES DE LA OBTENCIÓN DE REPORTES EN LA UCI.....	6
1.2 METODOLOGÍAS.....	7
1.2.1 XP (EXTREME PROGRAMMING)	7
1.2.2 MSF (MICROSOFT SOLUTION FRAMEWORK)	8
1.2.3 RUP (RATIONAL UNIFIED PROCESS).....	8
1.3 LENGUAJES A UTILIZAR	10
1.3.1 PHP.....	10
1.3.1.1 PHP5.....	11
1.3.2 UML.....	11
1.4 SISTEMAS GESTORES DE BASE DE DATOS (SGDB)	12
1.4.1 MICROSOFT SQL SERVER 2000	12
1.5 FRAMEWORK UTILIZADO	13
1.5.1 CAKEPHP	13
1.6 HERRAMIENTAS UTILIZADAS.....	14
1.6.1 RATIONAL ROSE	14
1.6.2 MACROMEDIA DREAMWEAVER 8	14
1.6.3 ZEND DEVELOPMENT ENVIRONMENT	14
1.7 SERVIDOR DE APLICACIONES APACHE.....	15
CONCLUSIONES	15
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA	17
INTRODUCCIÓN	17
2.1 PROCESOS DE NEGOCIO.....	17
2.2 OBJETO DE AUTOMATIZACIÓN	18
2.3 INFORMACIÓN QUE SE MANEJA	19
2.4 PROPUESTA DE SISTEMA.....	19

2.5 MODELO DEL NEGOCIO	20
2.5.1 DETERMINACIÓN Y JUSTIFICACIÓN DE LOS ACTORES Y TRABAJADORES	20
2.5.2 DIAGRAMA DE CASOS DE USO DEL NEGOCIO	21
2.5.3 DESCRIPCIÓN DE CASOS DE USO DEL NEGOCIO	22
2.5.4 DIAGRAMAS DE ACTIVIDADES POR CASOS DE USO	22
2.5.5 DIAGRAMA DE CLASES DEL MODELO DE OBJETOS	25
2.6 ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE	25
2.6.1 REQUERIMIENTOS FUNCIONALES	26
2.6.2 REQUERIMIENTOS NO FUNCIONALES	27
2.7 DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA	28
2.7.1 DEFINICIÓN DE LOS ACTORES	28
2.7.2 LISTADO DE CASOS DE USO	29
2.7.3 DIAGRAMA DE CASOS DE USO DEL SISTEMA	32
2.7.4 CASOS DE USO EXPANDIDOS	33
CONCLUSIONES	41
CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA	42
INTRODUCCIÓN	42
3.1 DIAGRAMA DE CLASES DEL ANÁLISIS	42
3.2 DISEÑO	43
3.2.1 DIAGRAMAS DE CLASES DEL DISEÑO	44
3.2.2 DESCRIPCIÓN DE LAS CLASES	50
3.2.3 DISEÑO DE LA BASE DE DATOS	61
3.2.3.1 Descripción de las tablas	61
3.3 TRATAMIENTO DE ERRORES	63
3.4 SEGURIDAD	64
3.5 INTERFAZ	64
3.6 CONCEPCIÓN DE LA AYUDA	64
3.7 ANÁLISIS DE COSTO Y BENEFICIOS	64
3.7.1 BENEFICIOS	71
CONCLUSIONES	71
CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA	72
INTRODUCCIÓN	72
4.1 IMPLEMENTACIÓN	72
4.1.1 DIAGRAMA DE DESPLIEGUE	72
4.1.2 DIAGRAMA DE COMPONENTES	73

4.2 MODELO DE PRUEBA.....	74
4.2.1 DESCRIPCIÓN DE LOS CASOS DE PRUEBA DE INTEGRACIÓN.....	75
CONCLUSIONES	80
CONCLUSIONES	81
RECOMENDACIONES.....	82
REFERENCIAS BIBLIOGRÁFICAS	83
BIBLIOGRAFÍA.....	84
GLOSARIO DE TÉRMINOS	86
ANEXOS	87

INDICE DE TABLAS

Tabla 1. Definición de actores del negocio.....	20
Tabla 2. Definición de trabajadores del negocio.....	21
Tabla 3. Descripción del caso de uso del negocio Obtener Reportes.....	22
Tabla 4. Descripción del caso de uso del negocio Modificar Credencial.....	22
Tabla 4. Definición de actores del sistema.....	28
Tabla 5. Definición del CU_Obtener Reportes.....	29
Tabla 6. Definición del CU_Obtener reporte por credenciales que ha tenido una persona.....	29
Tabla 7. Definición del CU_Obtener reporte por fecha de expiración.....	30
Tabla 8. Definición del CU_Obtener reporte por área y cargo.....	30
Tabla 9. Definición del CU_Obtener reporte por estado.....	31
Tabla 10. Definición del CU_Obtener reporte por tipo de credencial.....	31
Tabla 11. Definición del CU_Modificar credencial.....	32
Tabla 12. Descripción expandida del CU del sistema Obtener Reportes.....	34
Tabla 13. Descripción expandida del CU del sistema Obtener reporte por credenciales que ha tenido una persona.....	34
Tabla 14. Descripción expandida del CU del sistema Obtener reporte por fecha de expiración.....	36
Tabla 15. Descripción expandida del CU del sistema Obtener reporte por estado.....	37
Tabla 16. Descripción expandida del CU del sistema Obtener reporte por tipo de credencial.....	38
Tabla 17. Descripción expandida del CU del sistema Modificar Credencial.....	39
Tabla 18. Descripción de la buscador_personas.....	50
Tabla 19. Descripción de la clase m_credencial.....	50
Tabla 20. Descripción de la clase credenciales_tipoestado.....	51
Tabla 21. Descripción de la clase credenciales_fechaexpiración.....	52
Tabla 22. Descripción de la clase credenciales_personas.....	52
Tabla 23. Descripción de la clase credenciales_areacargo.....	53
Tabla 24. Descripción de la clase CredencialesController.....	53
Tabla 25. Descripción de la clase ReportesController.....	54
Tabla 26. Descripción de la clase PersonasController.....	55
Tabla 27. Descripción de la clase Credencial.....	55

Tabla 28. Descripción de la clase Estadocredencial.	57
Tabla 29. Descripción de la clase Tipocredencial.	57
Tabla 30. Descripción de la clase WSAkademios.	58
Tabla 31. Descripción de la clase WSCiudadano.	59
Tabla 32. Descripción de la clase WSTrabajadores.	60
Tabla 33. Descripción de la tabla dCredencial.	62
Tabla 34. Descripción de la tabla nEstadoCredencial.	62
Tabla 35. Descripción de la tabla nTipoCredencial.	63
Tabla 36. Tipos de actores y sus pesos.	65
Tabla 37. Tipos de Casos de Uso y sus pesos.	66
Tabla 38. Factor de Complejidad Técnica.	68
Tabla 39. Factor Ambiente.	69
Tabla 40. Esfuerzo por flujo de trabajo.	70
Tabla 41. Caso de prueba de integración para Modificar credenciales.	75
Tabla 42. Caso de prueba de integración para Credenciales persona.	77
Tabla 43. Caso de prueba de integración para reporte por tipo y estado.	78
Tabla 44. Caso de prueba de integración para reporte por fecha de expiración.	79

ÍNDICE DE FIGURAS

Figura 1.1 Extreme Programming.....	7
Figura 1.2 Metodología MSF.	8
Figura 1.3 Fases e iteraciones de la Metodología RUP	10
Figura 2.1 Diagrama de Casos de Uso del negocio.	21
Figura 2.2 Diagrama de actividades CU_Obtener Reportes.	23
Figura 2.3 Diagrama de actividades CU_Modificar Credencial.	24
Figura 2.4 Modelo de objetos del negocio.	25
Figura 2.5 Diagrama de casos de uso del sistema.....	33
Figura 3.1 Diagrama de clases de análisis.	43
Figura 3.2 Diagrama de clases del diseño del CU_Modificar Credenciales.	45
Figura 3.3 Diagrama de clases del diseño CU_Reporte por área o cargo.	46
Figura 3.4 Diagrama de clases del diseño CU_Reporte por credenciales-persona.	47
Figura 3.5 Diagrama de clases del diseño CU_Reporte por estado y por tipo.	48
Figura 3.6 Diagrama de clases del diseño CU_Reporte por fecha de expiración	49
Figura 3.7 Diagrama Entidad Relación.	61
Figura 4.1 Diagrama de despliegue.....	73
Figura 4.2 Diagrama de Componentes.	74
Figura 3.8 Diagrama de secuencia del CU Modificar credencial (Sección buscar Credencial).....	87
Figura 3.9 Diagrama de secuencia del CU Modificar credencial (Sección Modificar Credencial).....	88
Figura 3.10 Diagrama de secuencia del CU Obtener reporte por área o cargo	89
Figura 3.11 Diagrama de secuencia del CU Obtener reporte por credenciales persona	90
Figura 3.12 Diagrama de secuencia del CU Obtener Reporte por estado y tipo.....	91
Figura 3.13 Diagrama de secuencia del CU Obtener Reporte por fecha de expiración.....	92

INTRODUCCIÓN

El tema de la seguridad es muy importante para cualquier institución, máxime si la misma maneja gran cantidad de medios técnicos e información. La Universidad de las Ciencias Informáticas no es la excepción, pero para lograr un nivel de seguridad adecuado, es necesario implementar una serie de acciones dentro de las cuales está el manejo de las credenciales que se necesitan para que cada persona esté debidamente identificada.

Este proceso no debe ser gestionado directamente por una persona, lo cual ha impulsado la creación de sistemas que permitan el manejo de credenciales desde el mismo surgimiento de la Universidad. Teniendo en cuenta que los sistemas anteriores han sido desarrollados para dar solución a problemas coyunturales y los cambios ocurridos desde el comienzo de la UCI, la solución que existe en este momento no responde a las actuales necesidades. Considerando lo anterior se presenta la siguiente **situación problemática**:

En nuestra Universidad no existe en este momento una herramienta idónea para procesar la gran cantidad de información que se genera del trabajo con las credenciales. En la actualidad no se conoce las personas ni el área a la que pertenecen que en determinado espacio de tiempo presentan problemas con las credenciales. Por otra parte es necesaria una herramienta para la gestión de credenciales que sea de fácil uso para el trabajo con los diferentes estados por los que pasa una credencial antes de estar en manos del titular de la misma. Se necesita además que en caso de pérdida permita actuar con inmediatez para deshabilitarla y de esa forma evitar que sea usada por otra persona.

Basta preguntarse si con la información que se cuenta en estos momentos se puede dar respuesta por ejemplo, a las siguientes preguntas:

¿Cuáles son las áreas que más incidencias tienen en el área de acreditación?

¿Cuáles credenciales vencen hoy?

Es evidente que se requiere información actualizada y confiable para conocer las tendencias de la acreditación en la UCI basados en datos actualizados de manera constante. Dando respuesta a estos problemas surge el Sistema de Gestión y Reportes de Credenciales (SGRC).

Con este sistema se pretende obtener un método rápido, eficaz y confiable de controlar las credenciales que se utilizan en la UCI. El SGRC será otra de las herramientas que se utilicen en la universidad para gestionar los procesos de manera automatizada; lo cual permitirá un mayor control, de forma más confiable, con un gasto mínimo de recursos.

En este momento existe en la UCI una aplicación de escritorio que se encarga del trabajo con las credenciales, pero esta no brinda toda la información que se necesita; además de que concentra más funcionalidades de las que debe tener; lo cual introduce riesgos para la seguridad debido a errores del operador de la misma. Esta aplicación ofrece la posibilidad de obtener algunos reportes desde la máquina en que se usa; pero impide a los directivos que necesiten información en un momento determinado, a obtenerla desde cualquier parte de la universidad.

Este sistema pretende hacer más fácil el manejo de las credenciales y que los reportes sean generados por los verdaderos interesados en este proceso, desde la Web. De esta forma, hará indudablemente más fácil su obtención, ya que actualmente esto se hace desde una aplicación de escritorio. Pero el principal aporte será que esta área, tan importante para la seguridad de la Universidad, podrá ser controlada en tiempo real y además al ser más compartimentado el trabajo con las credenciales, se podrá garantizar un mayor nivel de seguridad. Otro elemento a tener en cuenta es el manejo de los recursos que son asignados a esta área, que podrá optimizarse como resultado del análisis de la información que se obtenga.

Al analizar lo anterior se obtiene el siguiente **problema científico**:

¿Cómo facilitar la obtención de la información y la modificación de las credenciales en la Universidad de las Ciencias Informáticas?

El **objeto de estudio** del mismo es: "Manejo de las credenciales y obtención de reportes que procesen la información que se genera del trabajo con estas" y su **campo de acción** será: la Dirección de Seguridad y Protección de la UCI

Visto lo anterior se plantea la siguiente **hipótesis**:

El desarrollo de una aplicación Web facilitará el manejo de la información y la gestión de las credenciales en la Universidad de las Ciencias Informáticas

El **objetivo general** es desarrollar una aplicación Web que garantice el control de las credenciales y la obtención de reportes en la UCI.

De esto, se derivan los siguientes **objetivos específicos**:

- Estudiar los procesos del negocio actual.
- Realizar análisis y diseño del sistema propuesto.
- Implementar el resultado del análisis y el diseño.

Para darle cumplimiento a esos objetivos se ejecutaron las siguientes **tareas**:

- Realizar un estudio del entorno de trabajo.
- Identificar las necesidades del cliente.
- Declarar los requisitos que debe cumplir el sistema.
- Describir los procesos que se van a implementar en el sistema.
- Especificar los procesos que se van a implementar en el primer ciclo de desarrollo.
- Modelar conceptualmente las clases que están implicadas en el sistema.
- Desarrollar los diagramas de actividad.
- Desarrollar los diagramas que describen el diseño del sistema.
- Describir las clases del diseño.
- Implementar la aplicación

El presente trabajo está compuesto por cuatro capítulos; los cuales servirán de guía en la comprensión del proceso llevado a cabo para dar cumplimiento a los objetivos trazados.

Capítulo I Fundamentación teórica

En este capítulo se hace un estudio del estado del arte, así como un análisis del porqué de las herramientas, tecnologías y lenguajes utilizados en el desarrollo e implementación del sistema propuesto.

Capítulo II Características del sistema

Este capítulo hace un análisis del proceso de negocio existente y a partir del mismo se obtienen las actividades que serán sujetas a automatización. Por otro lado se enuncian los requerimientos funcionales y no funcionales. Se hace, además, una descripción de los casos de uso que serán implementados, así como el diagrama de casos de uso del sistema.

Capítulo III Análisis y diseño del sistema

En este capítulo se obtiene, a través de los casos de uso del sistema, el diagrama de análisis. Como resultado del análisis obtenemos el diseño de la aplicación, el cual queda expresado en el diagrama de clases del diseño, los diagramas de secuencia y la descripción de las clases. Además en este capítulo se hace el análisis de costo beneficio de la aplicación.

Capítulo IV Implementación y prueba

En este capítulo se describe la distribución del sistema propuesto a través del diagrama de despliegue. También se muestra el diagrama de componentes, el cual ayuda a la comprensión del flujo de trabajo de implementación. Además se muestra el diseño de las pruebas a realizar.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Introducción

La utilización de sistemas de reportes es la evolución lógica de cualquier sistema informático que produzca información en sus procesos de negocio. Esto está generalizado actualmente en el mundo con diferentes formas de enfocar el problema. Algunos sistemas implementan generadores de reportes, mientras otros se encargan de obtener reportes específicos que interesan a la organización a la que pertenecen, la utilización de una modalidad u otra está determinada por el interés y necesidad del cliente. Estos sistemas permiten medir las tendencias de la información almacenada. Además son herramientas que asisten en la toma de decisiones, que provienen del análisis de la información obtenida de los datos que se almacenan en las bases de datos.

Este capítulo hace un recorrido por diferentes sistemas encargados de obtención de reportes tanto a nivel internacional como en nuestro país. Además se fundamenta el uso de las herramientas y lenguajes de programación utilizados; así como la metodología de desarrollo escogida.

1.1 Estado del Arte

A continuación se mostrará una serie de sistemas que brindan reportes a sus usuarios.

1.1.1 Sistemas de reportes en el mundo.

- **SIREL (Sistema de Reportes en Línea)**

Este sistema pertenece al Centro Universitario de Ciencias Económico Administrativas de la Universidad de Guadalajara. A través del mismo se puede acceder a las actividades que se realizan dentro de la coordinación, así como para presentar reportes de actividades desglosadas por área de cada unidad. Este sistema es muy sencillo y de fácil uso. Devuelve un reporte en una pagina HTML con la posibilidad de imprimir.

- **Sistema de reportes de proyectos**

Sevilla Global como Agencia Urbana de Promoción Económica del Ayuntamiento de Sevilla, España, dispone de una amplia gama de proyectos [1]. El número de entidades que colaboran con Sevilla Global es muy amplio y diverso, centrándose principalmente en Europa. El SRP, Sistema de Reporte de Proyectos, surge como solución para realizar el seguimiento y mantenimiento de dicha cartera de proyectos, de todas las actuaciones que se realizan y de la información de contacto del personal que colabora en cada uno de los proyectos. Este sistema fue desarrollado con PHP y MySQL, presenta una interfaz amigable e intuitiva que permite que usuarios que no posean conocimientos informáticos puedan utilizarlo sin dificultad.

1.1.2 Obtención de reportes en Cuba.

En la actualidad, en Cuba se ha desarrollado software que incluye la obtención de reportes.

- **GESTACAD (Sistema para la Gestión Académica)**

Este sistema desarrollado en la Universidad de Matanzas “Camilo Cienfuegos” se encarga de la gestión académica; pero posee un módulo de reportes el cual utiliza para obtener notas por asignatura y grupo, además de reportes de los resultados académicos de un estudiante en toda su carrera.

1.1.3 Particularidades de la obtención de reportes en la UCI.

- **Akademos**

Actualmente en la UCI el sistema de gestión académica Akademos posee un módulo Generador de reportes el cual está implementado en Visual Studio.Net utilizando como lenguaje de programación C# y como gestor de bases de datos SQL2000. Estas herramientas, si bien presentan características probadas en cuanto a su eficiencia, no responden a las nuevas exigencias del país con respecto a la utilización de software libre.

Los sistemas analizados no responden a las características propias del área de acreditación de la Universidad de las Ciencias Informáticas lo que provoca que no puedan ser utilizados en la misma. Lo cual lleva a la necesidad de desarrollar uno que responda a las necesidades de esa área.

1.2 Metodologías

Para desarrollar cualquier software es necesario guiar el proceso a través de una metodología, la cual será la encargada de elaborar “el plano” sobre el cual se apoyará el equipo de desarrollo. En la actualidad existen diferentes metodologías por las que se guía el desarrollo del software entre ellas se encuentran: XP, MSF y RUP como las más utilizadas. Estas fueron estudiadas con el fin de escoger una metodología de desarrollo a utilizar. A continuación se describe cada una de ellas.

1.2.1 XP (Extreme Programming)

Esta metodología está muy extendida principalmente para los proyectos pequeños que cuentan con poco personal y poco tiempo. Se basa en ir construyendo el software y hacerle pruebas al mismo tiempo de modo que el cliente -que se convierte en parte del equipo- dé su opinión y de esa manera haya una retroalimentación; que permite construir el software de la manera que el cliente quiera. La metodología está diseñada para entregar al cliente el software que necesita, cuando lo necesita.

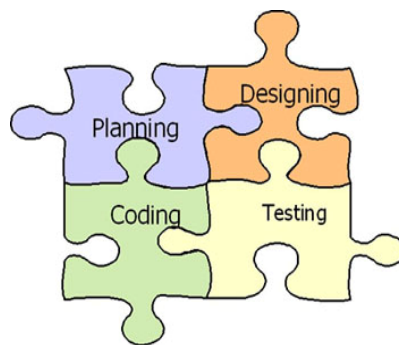


Figura 1.1 Extreme Programming.

1.2.2 MSF (Microsoft Solution Framework)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. Originalmente creado en 1994 para conseguir resolver los problemas a los que se enfrentaban las empresas en sus respectivos proyectos, se ha convertido posteriormente en un modelo práctico. MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación [2].



Figura 1.2 Metodología MSF.

1.2.3 RUP (Rational Unified Process)

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 base.
- **Construcción**, En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- **Transición**, El objetivo es llegar a obtener el *release* del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la 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 [3].

RUP está compuesto por tres elementos fundamentales

- **Actividades**, Son los procesos que se llegan a determinar en cada iteración.

- **Trabajadores**, son las personas o entes involucrados en cada proceso.
- **Artefactos**, Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Dentro de cada iteración de cada fase se llevan a cabo nueve flujos de trabajo dentro de los cuales los seis primeros son llamados de ingeniería y los demás son de apoyo.

Flujos de Trabajo de Ingeniería

- **Modelado del negocio:** Este flujo identifica los procesos de negocio, los que estarán sujetos a automatización y quiénes intervienen en los mismos.
- **Requerimientos:** Se identifican las restricciones que se imponen y lo que el sistema debe hacer.
- **Análisis y Diseño:** Describe cómo el programa será realizado y define cómo será programado.
- **Implementación:** Define cómo estarán los nodos ubicados y la ubicación de los objetos y clases en paquetes.
- **Prueba:** Se localizan los defectos del software.
- **Instalación:** Se entrega una versión operacional.

Flujos de trabajo de apoyo

- **Administración de proyecto:** Encargado de organizar el trabajo y de que se termine el proyecto en el tiempo previsto.
- **Administración de configuración y cambio:** Describe el uso y actualización concurrente de los elementos, control de versiones entre otras actividades.
- **Ambiente:** Describe los procesos y herramientas que soportarán al equipo de trabajo del proyecto.

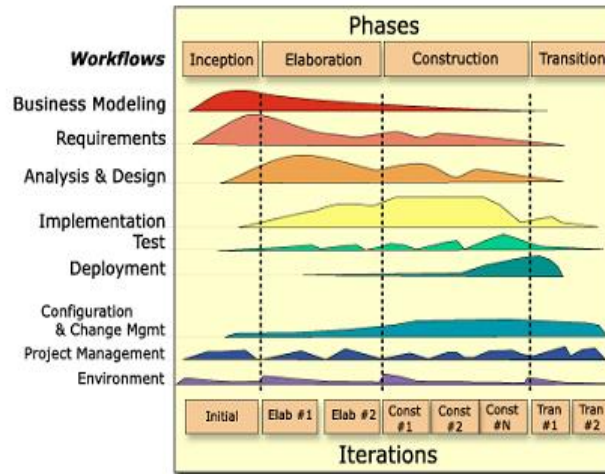


Figura 1.3 Fases e iteraciones de la Metodología RUP

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. Esta metodología es muy utilizada en el proceso de desarrollo de software por ser flexible; además, al ser iterativa, permite que se vaya construyendo el software por ciclos, por lo cual se pueden detectar errores con tiempo de antelación. Es una metodología confiable pues desde su surgimiento ha tenido una gran aceptación. Por todos estos elementos el sistema propuesto estará guiado por el Proceso Unificado de Desarrollo de Software.

1.3 Lenguajes a utilizar

1.3.1 PHP

PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor [4]. PHP puede ser utilizado en cualquiera de los principales sistemas operativos, incluyendo Linux, en variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS; o sea es multiplataforma. PHP es soportado por la mayoría de los servidores Web de hoy en día, por ejemplo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, Oreilly Website Pro Server, Caudium, Xitami, OmniHTTPd y muchos otros. De modo que con PHP se puede elegir el sistema operativo a utilizar y el servidor que se

prefiera. Otro de los aspectos a destacar es que al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que un navegador en particular lo soporte, es independiente del navegador, sin embargo, para que sus páginas funcionen, el servidor donde están alojadas debe soportar PHP. Además PHP cuenta con una gran comunidad de desarrolladores, como producto de código abierto, gozando de la ayuda de un gran grupo de programadores, lo que permite que los fallos de funcionamiento se encuentren y reparen rápidamente. Otro de los aspectos que se tuvieron en cuenta fue que PHP está dentro de los lineamientos trazados por la Dirección de Informatización en cuanto al uso de lenguajes de programación para el desarrollo de aplicaciones Web.

1.3.1.1 PHP5

En la implementación del sistema propuesto será utilizado PHP5. Esta es la versión más actual de este lenguaje. La selección se basa en que en esta versión, entre otras cosas, hace un cambio en el manejo de los objetos. Por ejemplo en PHP4 los objetos son tratados igual que otros tipos de datos básicos, como los enteros o los arreglos. O sea, cuando se realizan operaciones sobre los objetos, como asignación de variables o cuando son pasados como parámetros a funciones, todo el objeto es copiado. Por otra parte, en PHP5 todas las variables que nombran objetos son en realidad referencias. Otro de los elementos es que en PHP5 se incluyen modificadores de control de acceso para implementar el encapsulamiento, lo cual no existía en versiones anteriores. PHP tiene Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.

1.3.2 UML

El Lenguaje Unificado de Modelado (UML-Unified Modeling Language) ayuda a especificar, visualizar, y documentar modelos de sistemas de software, incluyendo su estructura y diseño. La actual versión de UML (UML 2.0) cuenta con treinta tipos de diagramas standard. UML fue originalmente creado por Rational Software, pero actualmente es atendido por el Object Management Group (OMG).

Es desde finales de la década del 90, un lenguaje de modelado orientado a objetos estándar, de acuerdo con el Object Management Group (OMG). UML es un estándar de la industria, pero no sólo de la industria

del software sino, en general, de cualquier industria que requiera la construcción de modelos como condición previa para el diseño y posterior construcción de prototipos. Además ayuda a entender la realidad de la tecnología y el funcionamiento de los procesos de negocio, reduciendo el costo y el tiempo empleado en la construcción de las piezas que constituirán el modelo. Este lenguaje reúne una serie de propiedades como:

- Es ampliamente utilizado por la industria desde su adopción por OMG.
- Modela estructuras complejas.
- Las estructuras más importantes que soporta tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos.
- Modela el comportamiento del sistema mediante casos de uso, diagramas de secuencia y de colaboración, que sirven para evaluar el estado de este.

Por todo lo anterior se decidió utilizar a UML como el lenguaje encargado de modelar el sistema que se propone.

1.4 Sistemas Gestores de Base de Datos (SGDB)

Un Sistema Gestor o Manejador de Bases de Datos (SGDB) es un conjunto de programas que permite a los usuarios crear y mantener una BD, por lo tanto, el SGDB es un software de propósito general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones. Pueden ser de propósito general o específico. Actualmente existen varios SGDB entre ellos: Oracle, MySQL, Visual Fox Pro, PostgreSQL y Microsoft SQL Server 2000, el cual será utilizado en la implementación del sistema propuesto.

1.4.1 Microsoft SQL Server 2000

Microsoft SQL Server 2000 es un sistema cliente-servidor de administración de base de datos relacional (Relational Database Management System - RDBMS) diseñado para procesamiento de transacciones en línea de alto rendimiento (online transaction processing - OLTP), almacenamiento de datos (data

warehousing) y aplicaciones de comercio electrónico (e-commerce). Proporciona seguridad, fiabilidad y escalabilidad para poner en marcha cualquier aplicación en un tiempo pequeño, destacando sus sencillas tareas de administración y su capacidad de analizar la información.

Este gestor de base de base de datos será el utilizado por sus innegables ventajas expuestas anteriormente, pero también porque la mayoría de las bases de datos con que se cuenta en estos momentos en la universidad corren sobre él. Otro aspecto que influyó en su elección es la experiencia con que se cuenta en la Universidad de las Ciencias Informáticas en el uso de este gestor, lo cual constituye sin duda una poderosa razón pues la base de datos que soportará, podría ser atendida por personas ajenas a la elaboración del software, sin necesidad de un estudio previo de un gestor desconocido; facilitando su mantenimiento.

1.5 Framework utilizado

Un Framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado [5]. Ayuda a desarrollar y unir los diferentes componentes de un proyecto. Un framework Web es una estructura definida, reusable; en el que sus componentes facilitan la creación de aplicaciones Web. Proveen una capa de abstracción sobre la arquitectura original ocultándola o adaptándola para no tener que utilizar el protocolo http de manera nativa y así acelerar los tiempos de desarrollo y mantenimiento.

1.5.1 CakePHP

CakePHP es un framework de desarrollo Web que permite la creación de aplicaciones de forma rápida. Este framework implementa el conocido patrón de diseño MVC (acrónimo de Modelo-Vista-Controlador). Dentro de sus características está que es compatible con PHP4 y PHP5, además es expandible sin la modificación de ninguno de sus archivos, pues se pueden crear componentes reutilizables. CakePHP es un framework gratis y de código abierto. Contiene un conjunto de librerías y clases que permite trabajar de una forma estructurada y rápida sin una gran pérdida de flexibilidad además, también hay que destacar su activa y colaborativa comunidad, que no se limita a su página Web [6]. Otro de los aspectos que se tuvo en cuenta es que CakePHP está avalado por la Dirección de Informatización en cuanto al uso de Framework para el desarrollo de aplicaciones Web.

1.6 Herramientas utilizadas

1.6.1 Rational Rose

Existen herramientas Case de trabajo visuales como Analise, Designe, Rational Rose, que permiten realizar el modelado del desarrollo de los proyectos. En la actualidad una de las más utilizadas en el mercado es Rational Rose y es la que se utiliza en la modelación de este proyecto. Rational Rose es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables. El navegador UML de Rational Rose nos permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Incluye un conjunto de herramientas de Ingeniería Inversa y generación de código que facilitan el producto hasta el producto final [7]. Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción.

1.6.2 Macromedia Dreamweaver 8

Dreamweaver es un editor WYSIWYG que es el acrónimo de What You See Is What You Get (en inglés, "lo que ves es lo que obtienes"). Será usado para el diseño Web de la aplicación por las facilidades que brinda. Esta herramienta brinda soporte para aplicaciones PHP y además facilita el uso de las CSS acrónimo Cascade Style Sheet (en inglés, "hoja de estilo en cascada").

1.6.3 Zend Development Environment

Este IDE, acrónimo de Integrated Development Environment (en inglés, "entorno integrado de desarrollo") es una de las herramientas más potente de programación para el lenguaje PHP. Cuenta con analizadores de código, permite completamiento de código entre otras ventajas. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión

de proyectos hasta la depuración de código. Lo más destacable es que contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir. El hecho de que esté desarrollado sobre Java da la ventaja de que puede ser utilizado en cualquier sistema operativo.

1.7 Servidor de aplicaciones Apache.

Apache es un servidor HTTP de código abierto el cual soporta tanto sistemas operativos UNIX como Windows NT. Es un servidor seguro y eficiente que cumple con los actuales standards de HTTP. Otra de sus características es que contiene mensajes de errores altamente configurables. Apache es utilizado como servidor HTTP en el 56.00 % de los sitios publicados en Internet. Fue creado en 1995 como parche a un servidor de aplicaciones llamado NCSA HTTP. Apache es un servidor altamente configurable de diseño modular. Actualmente existen muchos módulos para Apache que son adaptables a este, los cuales se clasifican en tres categorías:

- **Módulos Base:** Módulo con las funciones básicas del Apache.
- **Módulos Multiproceso:** Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones
- **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

El resto de funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

Conclusiones

En este capítulo se hizo un análisis de diferentes sistemas existentes para el manejo de reportes. Además se determinó las diferentes herramientas a utilizar en el desarrollo del proyecto. Se determinó la metodología a utilizar, así como los lenguajes que se utilizarán tanto para modelar la aplicación como para su posterior implementación.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Introducción

El presente capítulo hace un análisis del negocio en cuanto a su funcionamiento, documentos y sistemas necesarios para llevarlo a cabo. En el mismo se identifican las necesidades del cliente, se hace un estudio del entorno de trabajo. Además se hace una propuesta de sistema a partir del resultado del análisis de los requerimientos tanto funcionales como no funcionales.

2.1 Procesos de negocio

La Universidad de las Ciencias Informáticas es la primera universidad nacida al calor de la Batalla de Ideas. Tiene como objetivo fundamental la formación de los profesionales que tendrán la responsabilidad de informatizar la sociedad cubana. Con este fin, cada año entran a la Universidad poco más de 2000 estudiantes los cuales durante los cinco años de la carrera reciben una formación técnica y política que los lleven a convertirse en ese profesional preparado y comprometido que la Revolución y el país necesitan.

La calidad en la formación es fundamental dado que Cuba al ser un país bloqueado, carente de grandes recursos minerales, debe centrar su desarrollo en la producción intelectual; puesto que el capital humano es la principal riqueza con que se cuenta. Esto lleva a que la Universidad tenga una gran responsabilidad que para ser cumplida presupone la adopción de medidas especiales. Dentro de estas, entra la seguridad del centro, de las personas y los medios con que se cuenta. La acreditación es una de las principales herramientas para alcanzar este fin.

Los procesos que intervienen actualmente en la **obtención de reportes** en la UCI se comportan como sigue a continuación:

Si por necesidades propias del proceso de acreditación es necesario el conocimiento de algún dato en particular, la persona interesada debe dirigirse al área de acreditación y pedirlos al encargado de expedir las credenciales. La única forma de que un directivo interesado sepa un dato en particular es preguntando en ese lugar, lo cual hace que el proceso de obtener el dato sea lento y además demora el trabajo de los encargados de la misma. Para desactivar alguna credencial que haya expirado, el encargado del área a la que pertenecía el titular de la misma lo debe reportar, proceso el cual trae como consecuencia que una

persona que no haga estos pasos pueda usar ese documento para entrar a la universidad sin contar con autorización.

Por lo expuesto anteriormente el proceso de obtención de reportes -muy importantes para el buen funcionamiento del sistema de seguridad de la universidad- no cuenta con una herramienta que de forma eficaz brinde informaciones necesarias con un mínimo de trabajo, lo cual dificulta el trabajo de los encargados de obtenerlas y de las personas que la necesitan para cumplir con sus responsabilidades con el grado de inmediatez requerida. Al estar el módulo de obtención de reportes dentro de la actual aplicación encargada del manejo de las credenciales, es imposible que un directivo aun teniendo autorización para acceder a la información que requiere, la pueda obtener de forma inmediata. Mientras no se cuenten con datos precisos del estado en que se encuentren las credenciales en la Universidad de las Ciencias Informáticas, podrán persistir los problemas mencionados anteriormente.

2.2 Objeto de automatización

Será objeto de automatización el proceso de obtención de reportes, obteniendo información importante del área de acreditación no solo por parte de sus trabajadores, sino también de directivos que cuenten con autorización. Además, será objeto de automatización el trabajo de actualización de los estados de las credenciales.

En estos momentos se cuenta con una aplicación de escritorio que se encarga del trabajo relativo a las credenciales el cual es la fuente de información para los reportes que se obtienen. Esta aplicación cuenta con un pequeño modulo de obtención de reportes que no cumple con las expectativas del personal que la utiliza en su trabajo.

En la UCI en estos momentos los datos de las personas se encuentran en diferentes bases de datos; respondiendo, a las necesidades propias de un centro donde estudian y trabajan tantas personas. Esto hace más complejo el proceso de obtener los datos que el sistema propuesto necesita para dar respuesta a los requerimientos que el cliente planteó. Estos son:

- **Akademos:** Para la obtención referente a los estudiantes de la Universidad.

- **Trabajadores:** Para la información referente tanto a los profesores como a los trabajadores con que cuenta la Universidad.
- **Ciudadano:** En esta base de datos se encuentra la información común para cada persona de la UCI.
- **Identificación:** Esta base de datos contiene las fotos y los numero de credencial de todas las personas de la UCI.

2.3 Información que se maneja

En estos momentos los documentos que se procesan en el área de acreditación son fundamentalmente las credenciales que se expiden desde la misma. Estas credenciales cuentan con diferentes datos como son: el tipo de credencial, el/los nombre(s) y apellidos, el área y cargo, el código de barras y el número de la identificación; conocido también como “el número del solapín”. En estos momentos no se puede decir que se obtienen reportes de la aplicación existente; pues son consultas que se hacen y que ni siquiera son impresos.

2.4 Propuesta de sistema

El sistema contará con una página principal la cual dé acceso a la página de Obtención de Reportes y a la de Modificar Credencial. Si se entra dentro de la página de Obtener Reportes se mostrará un menú donde se encontrarán los diferentes reportes que brinda el sistema. Al seleccionar cada tipo de reportes se entrará a una página nueva, la cual contará con un buscador que se adecue a los requerimientos de cada reporte. En la página Modificar Credencial se tendrá un buscador que permita obtener una credencial por los diferentes atributos que caracterizan a estas. En esta misma página se mostrarán los resultados de la búsqueda mostrando la o las credenciales que responden a las características especificadas y dando la posibilidad de escogerlas y cambiarle su estado.

El sistema existente es una aplicación de escritorio que cuenta con un pequeño módulo de reportes. Como se ha analizado en este trabajo, este solo brinda un pequeño grupo de reportes que no responden a las necesidades del cliente y lo hace desde la misma aplicación encargada del procesamiento de

credenciales. Lo anteriormente expuesto hace que la misma cuente con más funcionalidades de las necesarias además la utilización del Sistema de Gestión y Reportes de Credenciales posibilitará que no se utilice la aplicación de escritorio existente para obtener reportes específicos en un momento determinado, lo cual indudablemente mejorara el proceso de obtención de reportes, al poder ser obtenidos desde cualquier lugar con las posibilidades que brinda una aplicación Web. Otro aspecto interesante será la posibilidad de modificarle el estado a una credencial con la mayor inmediatez posible.

2.5 Modelo del negocio

La utilización del SMRC permitirá obtener información confiable y oportuna sobre determinados aspectos requeridos en el área de acreditación. Además será una herramienta que permitirá de forma rápida modificar el estado en que se encuentre una credencial en un momento determinado. Lo anteriormente expresado está basado en el hecho de que el sistema será desarrollado como una aplicación Web, lo cual permite que pueda ser utilizado en cualquier momento desde cualquier lugar, dentro de la UCI, siempre que se tenga permiso de acceso a las funcionalidades que el sistema brinda. Con la implementación y puesta en funcionamiento del sistema se podrá hacer un mejor manejo de la información que se genera en la Oficina de Seguridad y Protección; pero el primer paso para lograr una aplicación que responda a los requerimientos del cliente es comprender el negocio actual dentro de la organización. Con este objetivo se lleva a cabo el modelo de negocio; el cual hace un levantamiento de cómo funcionan los procesos hasta el momento.

2.5.1 Determinación y justificación de los actores y trabajadores

Tabla 1. Definición de actores del negocio.

Actores del Negocio	Justificación
Directivo	Pide los reportes sobre las credenciales
Persona_UCI	Es el que hace la petición de cambio de estado de la credencial.

Tabla 2. Definición de trabajadores del negocio.

Trabajadores del negocio	Justificación
Especialista	Encargado de cambiar el estado de las credenciales además es el encargado de brindar los reportes a los directivos

2.5.2 Diagrama de Casos de Uso del Negocio

El diagrama de casos de uso del negocio tiene una gran importancia porque representa cómo interactúan los actores y casos de uso del negocio. Ayudando al equipo de desarrollo a comprender cómo funciona el proceso de negocio con la consiguiente ayuda a la hora de obtener los casos de uso del sistema.

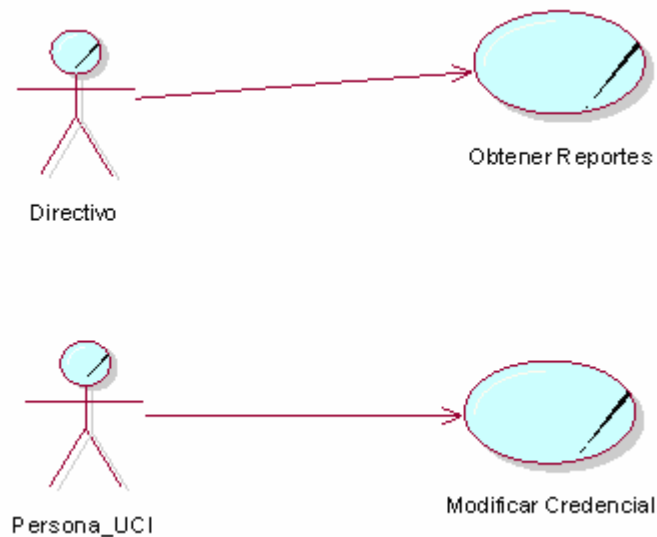


Figura 2.1 Diagrama de Casos de Uso del negocio.

2.5.3 Descripción de casos de uso del negocio

Tabla 3. Descripción del caso de uso del negocio Obtener Reportes.

CU-1	Obtener Reportes
Actor	Directivo
Trabajador	Especialista
Descripción	Comienza cuando un directivo pide un reporte determinado al especialista y da como resultado el reporte solicitado.

Tabla 4. Descripción del caso de uso del negocio Modificar Credencial.

CU-2	Modificar Credencial
Actor	Persona_UCI
Trabajador	Especialista
Descripción	Comienza cuando una Persona_UCI solicita que se le actualice o se le dé baja a su credencial según sea el caso.

2.5.4 Diagramas de Actividades por casos de uso

Los diagramas de actividades permiten ver el flujo de actividades que se lleva a cabo; permitiendo la identificación de las actividades objeto de automatización. En este caso las actividades objeto de automatización están señaladas en el diagrama.

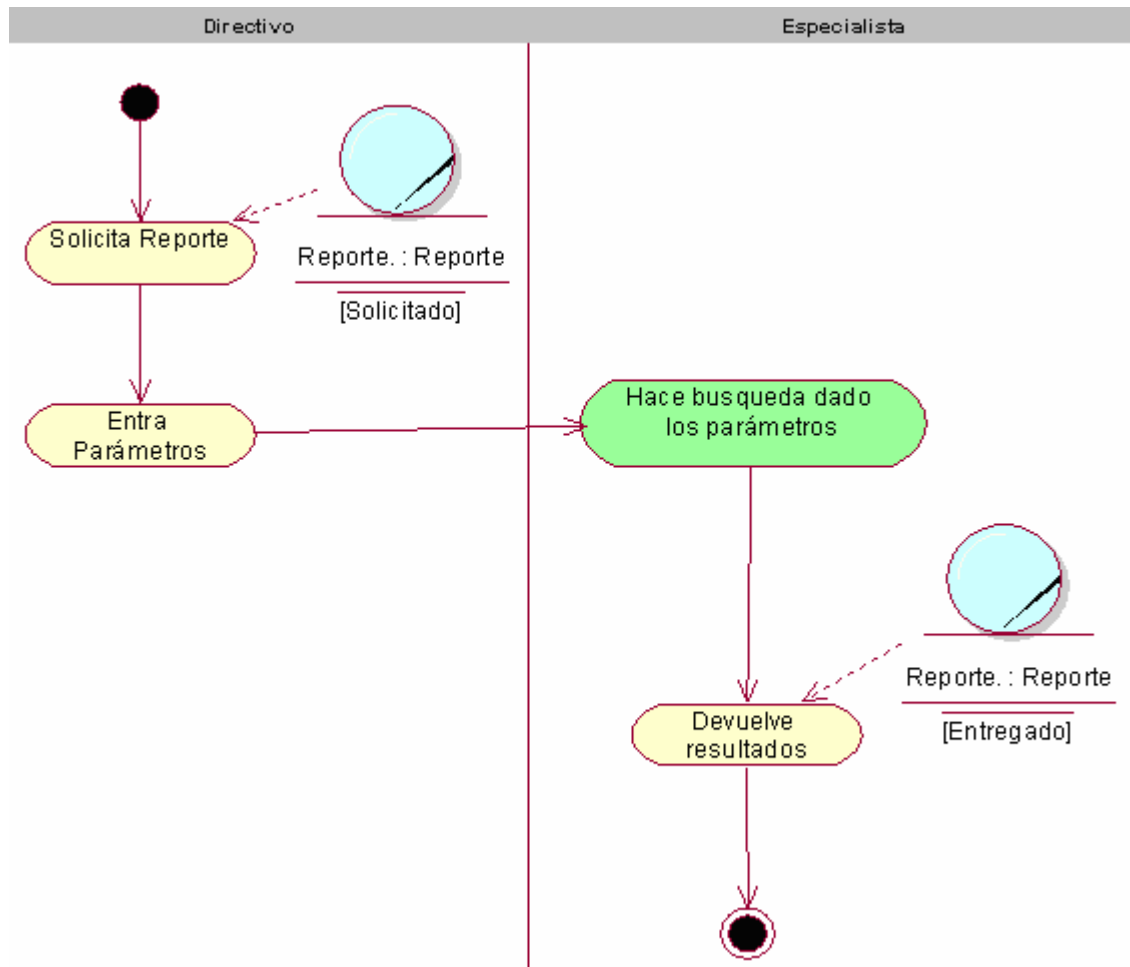


Figura 2.2 Diagrama de actividades CU_Obtener Reportes.

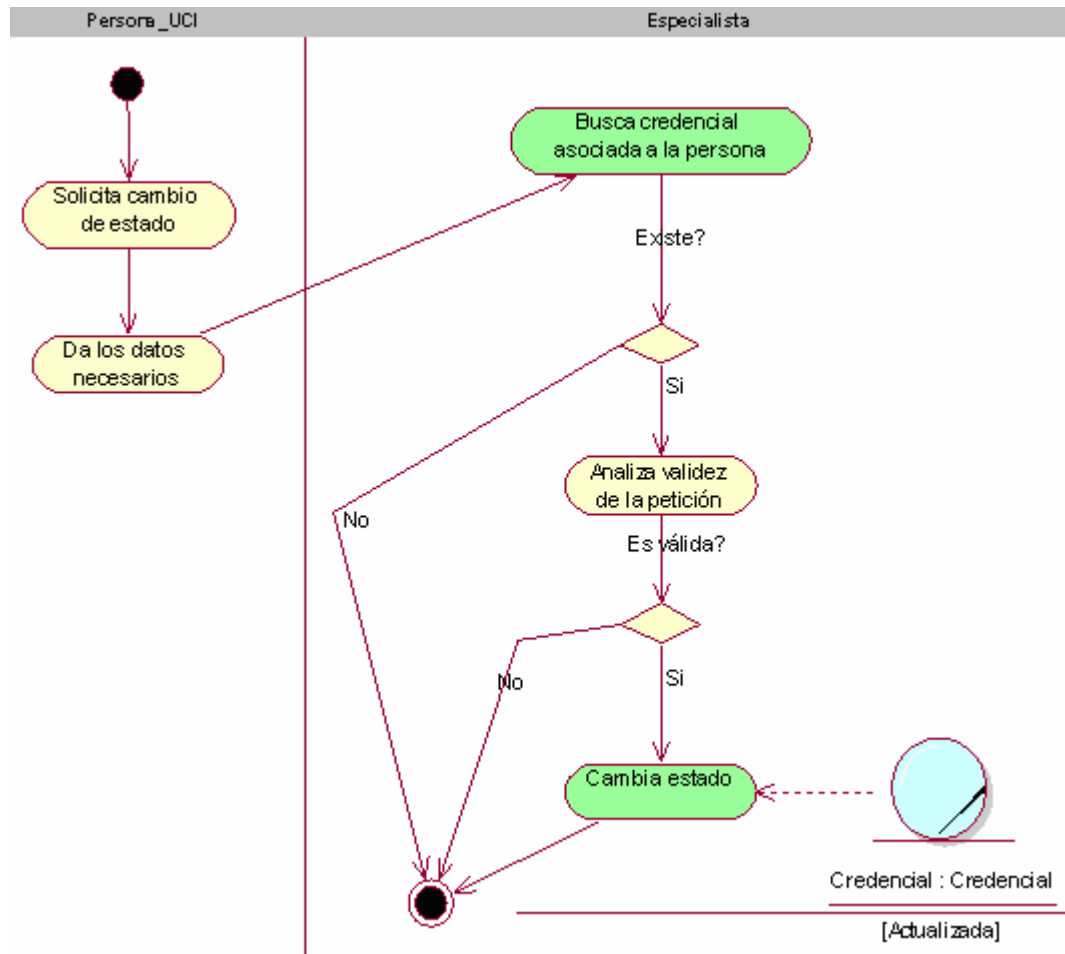


Figura 2.3 Diagrama de actividades CU_Modificar Credencial.

2.5.5 Diagrama de clases del modelo de objetos

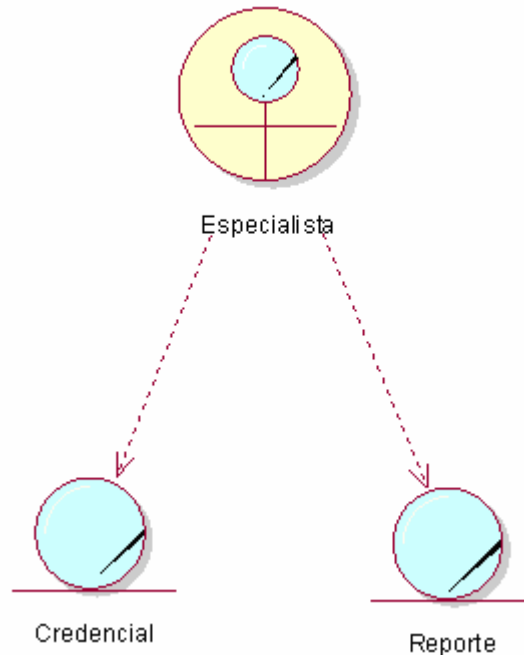


Figura 2.4 Modelo de objetos del negocio.

2.6 Especificación de los requisitos de software

El sistema propuesto requiere información de sistemas existentes en estos momentos en la Universidad de las Ciencias Informáticas. Esto hace que dependa directamente de las bases de datos de Akademos, Trabajadores, Ciudadano, e Identificación. Además de eso; se relaciona con el sistema de Credenciales existente en estos momentos, pues la información de los reportes que se generan sale del manejo de las credenciales con ese software.

Al ser RUP una metodología basada en casos de uso. La determinación de estos es de suma importancia para el desarrollo ulterior de cualquier sistema a implementar; por lo cual los requerimientos funcionales y no funcionales tienen una gran importancia a la hora de identificar los casos de uso del sistema.

2.6.1 Requerimientos Funcionales

Sitio Web

1. Mostrar reportes de:

1.1 Todas las credenciales que ha tenido una persona. Para lo cual se debe obtener para cada credencial que ha tenido:

1.1.1 Nombre(s) y Apellidos del titular.

1.1.2 Área y cargo para los trabajadores, facultad y grupo para los estudiantes.

1.1.3 Tipo credencial.

1.1.4 Estado de las credenciales

1.1.4 Fecha de activación y expiración (si posee).

1.2 Todas las personas con credenciales de un cierto tipo. Debe mostrar:

1.2.1 Nombre(s) y Apellidos del titular.

1.2.2 Área y cargo para los trabajadores, facultad y grupo para los estudiantes.

1.2.3 Número de credencial.

1.3 Todas las personas que tengan determinada fecha de expiración.

1.3.1 Nombre(s) y Apellidos del titular.

1.3.2 Área y cargo para los trabajadores, facultad y grupo para los estudiantes.

1.3.3 Número de credencial.

1.4 Todas las personas que pertenezcan a un área y/o cargo determinado.

1.4.1 Nombre(s) y Apellidos del titular.

1.4.2 Área y cargo para los trabajadores, facultad y grupo para los estudiantes.

1.4.3 Tipo de credencial

1.4.4 Número de credencial.

1.5 Todas las credenciales que estén en un estado determinado.

1.5.1 Nombre(s) y Apellidos del titular.

1.5.2 Área y cargo para los trabajadores, facultad y grupo para los estudiantes.

1.5.3 Tipo de credencial

1.5.4 Número de credencial.

2. Permitir cambiar el estado de una credencial.

2.1 Para cambiar el estado de una credencial se debe buscar por:

2.1.1 Número de credencial

2.1.2 Nombre del titular

2.1.3 Tipo de Credencial

2.6.2 Requerimientos no funcionales

Apariencia o interfaz externa

- La interfaz tendrá un diseño agradable e intuitivo.

Usabilidad

- El sistema en general se desarrolla con el objetivo de facilitar el trabajo realizado hasta el momento y mantener almacenada toda la información.
- La interfaz mostrará únicamente funcionalidades referentes a las tareas para las que está destinado el sistema.
- Se garantiza que los usuarios solo tengan acceso a la información que les corresponda.
- Facilidad de uso para todo tipo de clientes, incluyendo personas con pocos conocimientos en el uso de las computadoras.

Rendimiento

- El sistema debe garantizar la mayor eficiencia posible en cuanto al tratamiento de la información, de manera que la velocidad de procesamiento de la misma sea la máxima posible. Se debe garantizar la consistencia y disponibilidad de la información en todo momento, por lo que se requiere además un tiempo de recuperación mínimo.

Seguridad

- Las páginas del sitio solo estarán accesibles para personas que tengan usuarios autorizados en el servidor.
- La información que se obtiene no será para conocimiento de cualquier persona sino para los interesados por requerimientos de cargo y responsabilidad.

Soporte

- El sistema debe ser de fácil instalación, adaptable a numerosas plataformas y de fácil mantenimiento.

Portabilidad

- Facilidad para adaptarlos a diferentes ambientes sin necesidad de usar otros medios que los previstos. Se requiere de Windows como plataforma.

2.7 Definición de los casos de uso del sistema

Los casos de uso del sistema serán en definitiva los encargados de guiar todo el proceso de desarrollo del sistema a implementar. Es importante determinar en esta etapa los actores del sistema así como los casos de uso de los cuales se benefician.

2.7.1 Definición de los actores

Se definen los actores del sistema y se justifica el porque de la elección.

Tabla 4. Definición de actores del sistema.

Actores	Justificación
----------------	----------------------

Directivo	Pide los reportes sobre las credenciales
Especialista	Es el encargado de dar respuesta a las peticiones de cambio de estado a las credenciales hechas.

2.7.2 Listado de casos de uso

Se presenta un listado de los casos de uso del sistema con una breve información de quien es el actor que lo inicializa, el nombre del caso de uso y una breve descripción del mismo.

Tabla 5. Definición del CU_Obtener Reportes.

CU-1	Obtener Reportes
Actor	Directivo
Descripción	Este caso de uso es el que da comienzo al proceso de búsqueda, es el encargado de redireccionar al actor a la búsqueda que realmente necesita; todos los demás casos de uso de reportes son una extensión de este.
Referencia	

Tabla 6. Definición del CU_Obtener reporte por credenciales que ha tenido una persona.

CU-2	Obtener reporte por credenciales que ha tenido una persona
Actor	Directivo
Descripción	El actor busca a una persona determinada para de esa manera saber

	cuántas credenciales ha tenido.
Referencia	CU-1 Requerimiento funcional 1.1

Tabla 7. Definición del CU_Obtener reporte por fecha de expiración.

CU-3	Obtener reporte por fecha de expiración
Actor	Directivo
Descripción	El actor hace búsquedas de credenciales por su tipo y por su fecha de expiración. Es el encargado de obtener reportes de credenciales por su fecha de expiración (dirigido fundamentalmente a las credenciales de los eventuales y tercerizados).
Referencia	CU-1 Requerimiento funcional 1.3

Tabla 8. Definición del CU_Obtener reporte por área y cargo.

CU-4	Obtener reporte por área y cargo
Actor	Directivo
Descripción	El actor hace búsquedas de credenciales por el área y el cargo del titular de las credenciales. De esta forma se puede ver el estado en que se encuentra la credencial de una persona o varias personas siempre

	que tengan el mismo cargo dentro de una misma área.
Referencia	CU-1 Requerimiento funcional 1.4

Tabla 9. Definición del CU_Obtener reporte por estado.

CU-5	Obtener reporte por estado
Actor	Directivo
Descripción	El actor hace búsquedas de credenciales que se encuentren en un estado determinado para esto introduce un estado, esperando como respuesta las credenciales que responden a este estado.
Referencia	CU-1 Requerimiento funcional 1.5

Tabla 10. Definición del CU_Obtener reporte por tipo de credencial.

CU-6	Obtener reporte por tipo de credencial
Actor	Directivo
Descripción	El actor hace búsquedas de credenciales que sean de un tipo determinado.
Referencia	CU-1 Requerimiento funcional 1.2

Tabla 11. Definición del CU_Modificar credencial.

CU-7	Modificar credencial
Actor	Especialista
Descripción	El actor accede a modificar credencial y aquí es que cambia el estado de las credenciales según la solicitud de una persona o por su propio trabajo si así lo requiera.
Referencia	Requerimiento funcional 2

2.7.3 Diagrama de casos de uso del sistema

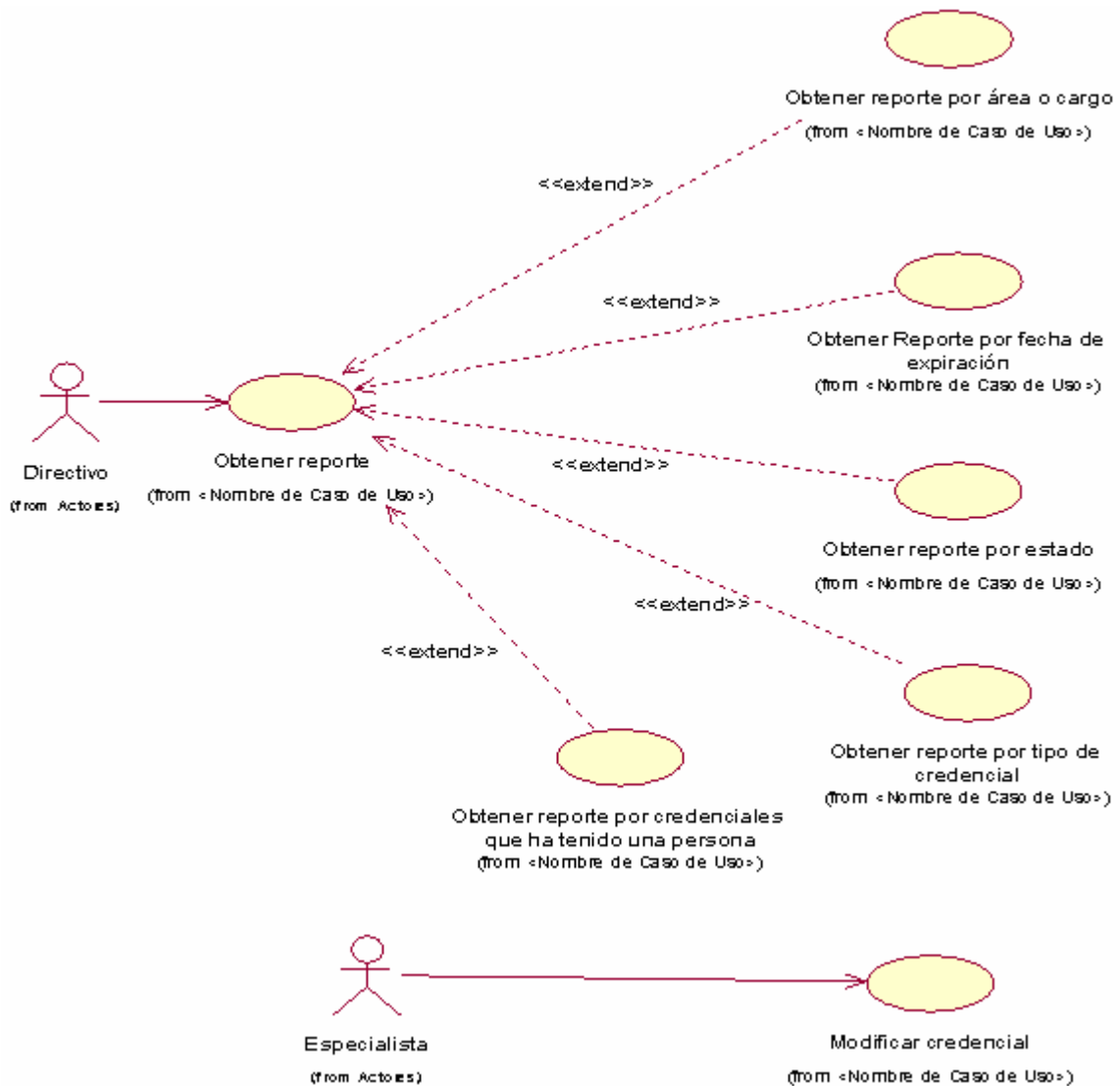


Figura 2.5 Diagrama de casos de uso del sistema.

2.7.4 Casos de uso expandidos

Se presenta una descripción detallada de los casos de uso donde se expone: nombre del caso de uso, propósito del caso de uso en cuestión, actores que lo inicializan, un pequeño resumen, las referencias, las acciones del actor y la respuesta del sistema y los flujos alternativos que puede tomar el caso de uso.

Tabla 12. Descripción expandida del CU del sistema Obtener Reportes.

Caso de uso	
CU-1	Obtener Reportes
Propósito	Obtener reportes sobre acreditación.
Actores: Directivo	
Resumen: El directivo pide informaciones que se generan del trabajo con las credenciales	
Referencias	
Acción del actor	Respuesta del sistema
1. El CU comienza cuando el actor accede al enlace Obtener Reportes.	
	2. El sistema muestra el formulario correspondiente a la búsqueda. Mostrando, tipos de reportes que se pueden obtener.
3. El actor elije el enlace al que va a entrar.	
	4. El sistema envía al usuario a la página seleccionada.

Tabla 13. Descripción expandida del CU del sistema Obtener reporte por credenciales que ha tenido una persona.

Caso de uso	
CU-2	Obtener reporte por credenciales que ha tenido una persona.

Propósito	Obtener reporte por la cantidad de credenciales que ha tenido una persona.	
Actores: Directivo.		
Resumen: El actor busca a una persona determinada para de esa manera saber cuantas credenciales ha tenido.		
Referencias	CU-1, Requerimiento funcional 1.1	
Acción del actor	Respuesta del sistema	
1. El CU comienza cuando el actor accede al enlace <u>Obtener Reportes por credenciales de una persona</u> .		
	2. El sistema muestra el formulario correspondiente a la búsqueda. Mostrando, campos donde el actor pueda introducir los datos de la persona.	
3. El actor entra los parámetros de la persona que le interesa.		
	4. El sistema muestra a la persona con la cantidad de credenciales que ha tenido. En caso que no lo encuentre ver flujo alternativo 1	
Flujo alternativo 1		
Acción del actor	Respuesta del sistema	

	Muestra al usuario que la búsqueda no ha tenido resultado dándole la opción de comenzar nuevamente el proceso.
--	--

Tabla 14. Descripción expandida del CU del sistema Obtener reporte por fecha de expiración.

Caso de uso	
CU-3	Obtener reporte por fecha de expiración.
Propósito	Hacer búsquedas de credenciales por su fecha de expiración.
Actores: Directivo.	
Resumen: Es el encargado de obtener reportes de credenciales por su fecha de expiración (dirigido fundamentalmente a las credenciales de los eventuales y tercerizados).	
Referencias	CU-1, Requerimiento funcional 1.3.
Acción del actor	Respuesta del sistema
1. El CU comienza cuando el actor accede al enlace <u>Reporte por fecha de expiración</u> .	
	2. El sistema muestra el formulario correspondiente a la búsqueda. Mostrando, el campo del parámetro que se necesita. Este es Fecha de Expiración.
3. El actor introduce la fecha d expiración.	

	4. El sistema devuelve la información que pidió el directivo. En caso que no lo encuentre ver flujo alternativo 1.
Flujo alternativo 1	
Acción del actor	Respuesta del sistema
	Muestra al usuario que la búsqueda no ha tenido resultado dándole la opción de comenzar nuevamente el proceso.

Tabla 15. Descripción expandida del CU del sistema Obtener reporte por estado.

Caso de uso	
CU-5	Obtener reporte por estado
Propósito	Hacer búsquedas de credenciales por el estado en que se encuentre
Actores: Directivo	
Resumen: El actor hace búsquedas de credenciales que se encuentren en un estado determinado para esto introduce un estado esperando como respuesta las credenciales que responden a este estado.	
Referencias	CU-1, Requerimiento funcional 1.5
Acción del actor	Respuesta del sistema

1. El CU comienza cuando el actor accede al enlace <u>Reporte por estado</u> .	
	2. El sistema muestra el formulario correspondiente a la búsqueda. Mostrando, el campo de requerido que en este caso es Estado.
3. El actor introduce el estado que le interesa.	
	4. El sistema devuelve la información que pidió el directivo. En caso que no lo encuentre ver flujo alternativo 1
Flujo alternativo	
Acción del actor	Respuesta del sistema
	Muestra al usuario que la búsqueda no ha tenido resultado dándole la opción de comenzar nuevamente el proceso.

Tabla 16. Descripción expandida del CU del sistema Obtener reporte por tipo de credencial.

Caso de uso	
CU-6	Obtener reporte por tipo de credencial.
Propósito	Hacer búsquedas de credenciales por el tipo de credencial que sea.
Actores: Directivo.	

Resumen: El actor hace búsquedas de credenciales que se encuentren en un estado determinado para esto introduce un estado esperando como respuesta las credenciales que responden a este.	
Referencias	CU-1, Requerimiento funcional 1.2.
Acción del actor	Respuesta del sistema
1. El CU comienza cuando el actor accede al enlace <u>Reporte por tipo de credencial</u> .	
	2. El sistema muestra el formulario correspondiente a la búsqueda. Mostrando, el campo de requerido que en este caso es Tipo de Credencial.
3. El actor introduce el estado que le interesa.	
	4. El sistema devuelve la información que pidió el directivo. En caso que no lo encuentre ver flujo alternativo 1.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	Muestra al usuario que la búsqueda no ha tenido resultado dándole la opción de comenzar nuevamente el proceso.

Tabla 17. Descripción expandida del CU del sistema Modificar Credencial.

Caso de uso

CU-8	Modificar Credencial.	
Propósito	Cambiarle el estado a las credenciales.	
Actores: Especialista.		
Resumen: El actor accede a Modificar Credencial y a partir de aquí es que puede trabajar con los estados de las credenciales.		
Referencias	Requerimiento funcional 2.	
Acción del actor	Respuesta del sistema	
1. El CU comienza cuando el actor accede al enlace <u>Modificar Credencial</u> .		
	2. El sistema muestra el formulario correspondiente a la búsqueda. Mostrando, los diferentes parámetros que pueden utilizarse para buscar una credencial. Estos son Nombre del titular, tipo de credencial, número de la credencial.	
3. El actor introduce cualquiera de los datos que posea.		
	4. El sistema envía la o las credenciales que respondan a los parámetros de la búsqueda mostrando el estado y un vínculo para dar posibilidades de cambiarlo. Si no lo encuentra ver Flujo Alternativo 1.	

5. Selecciona cambiar estado.	
	6. Envía una nueva página con la credencial seleccionada y un menú desplegable para que escoja el nuevo estado.
7. Selecciona nuevo estado y acepta	
	8. Envía mensaje de confirmación.
Flujo alternativo 1	
Acción del actor	Respuesta del sistema
	El sistema informa al actor que la búsqueda no tuvo resultado. Da la posibilidad de que haga una nueva búsqueda.

Conclusiones

En este capítulo se hizo un análisis del entorno de trabajo y se identificaron los requerimientos del cliente. Por otra parte se declararon los requerimientos formulados por el cliente. Como resultado de lo expresado anteriormente se logra una mayor claridad en cómo funciona el negocio y qué se hará para implementar el sistema propuesto.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA

Introducción

Este capítulo presenta el análisis y diseño del Subsistema para el Manejo y Reportes de Credenciales. El mismo mostrará “qué” y “cómo” se implementará dentro de la aplicación. El análisis dará una visión general de qué se debe hacer sin entrar en detalles, mostrando desde tres estereotipos básicos: interfaz, negocio y entidad la forma aproximada que tendrá el sistema. Mientras que el diseño dará como resultado un “mapa” de la implementación, lo cual será gracias al diagrama de clases del diseño que mostrará las clases necesarias con sus métodos y atributos. Como parte del diseño se mostrarán también los diagramas de secuencia que darán una visión dinámica de las clases colaborando indudablemente a la comprensión de la tarea a desarrollar.

3.1 Diagrama de clases del análisis

El análisis, como se ha expresado previamente, brinda una visión muy general de lo que debe hacer el sistema. El modelo del análisis brinda una vista interna del sistema que está descrita en el lenguaje del desarrollador. A continuación se expone el diagrama de clases del análisis del sistema propuesto.

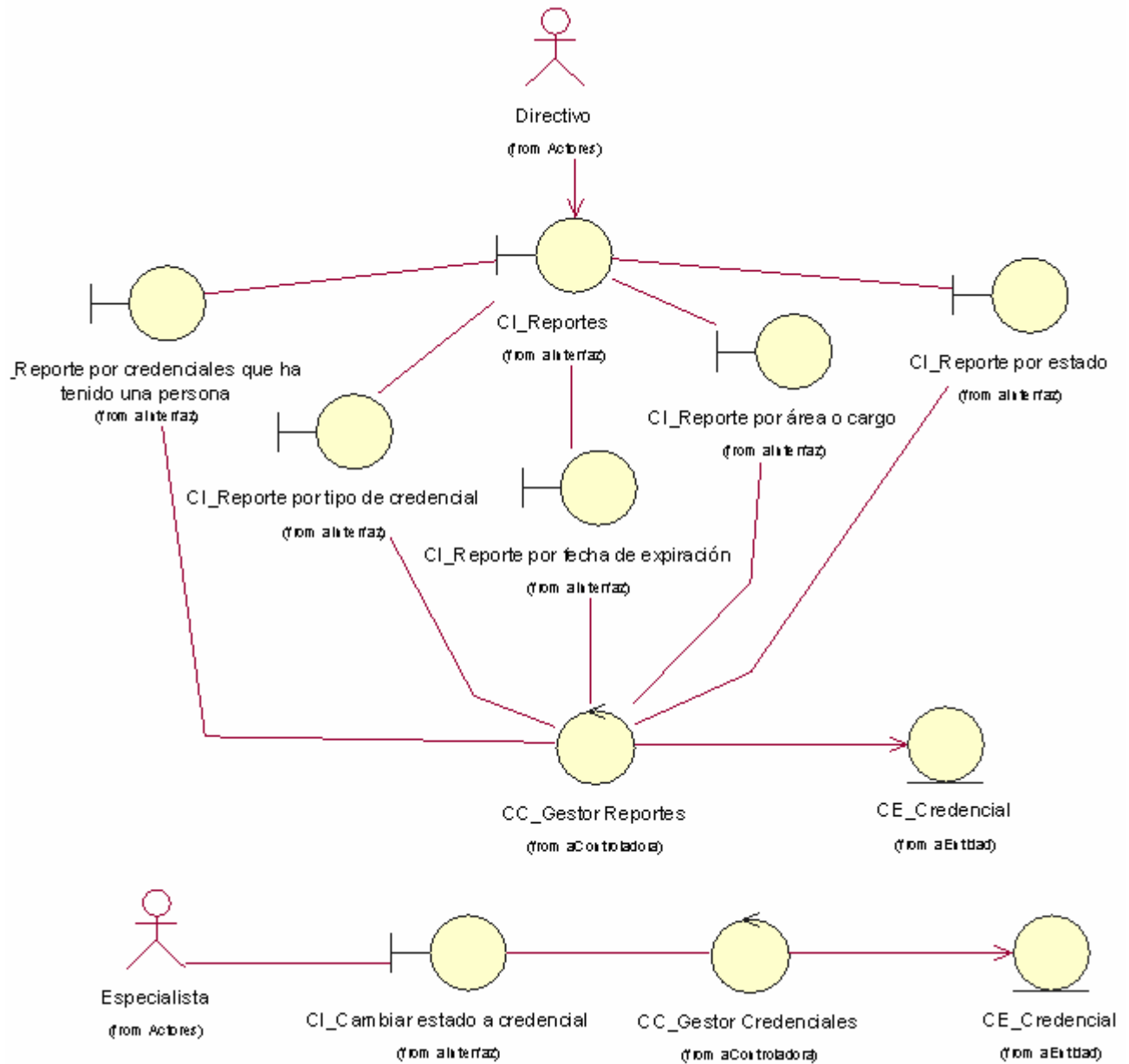


Figura 3.1 Diagrama de clases de análisis.

3.2 Diseño

Los diagramas de clase del diseño serán representados en esta sección; de los cuales, se mostrarán los diagramas de clases para cada caso de uso. Para una mejor comprensión de las clases representadas en estos diagramas serán descritas de forma textual sus métodos y atributos. Para representar el aspecto dinámico de cada uno de los diagramas de esta sección, serán utilizados los diagramas de secuencia los cuales representan la interacción entre clases. A continuación se representan los diagramas de clases del diseño para cada caso de uso.

3.2.1 Diagramas de clases del diseño

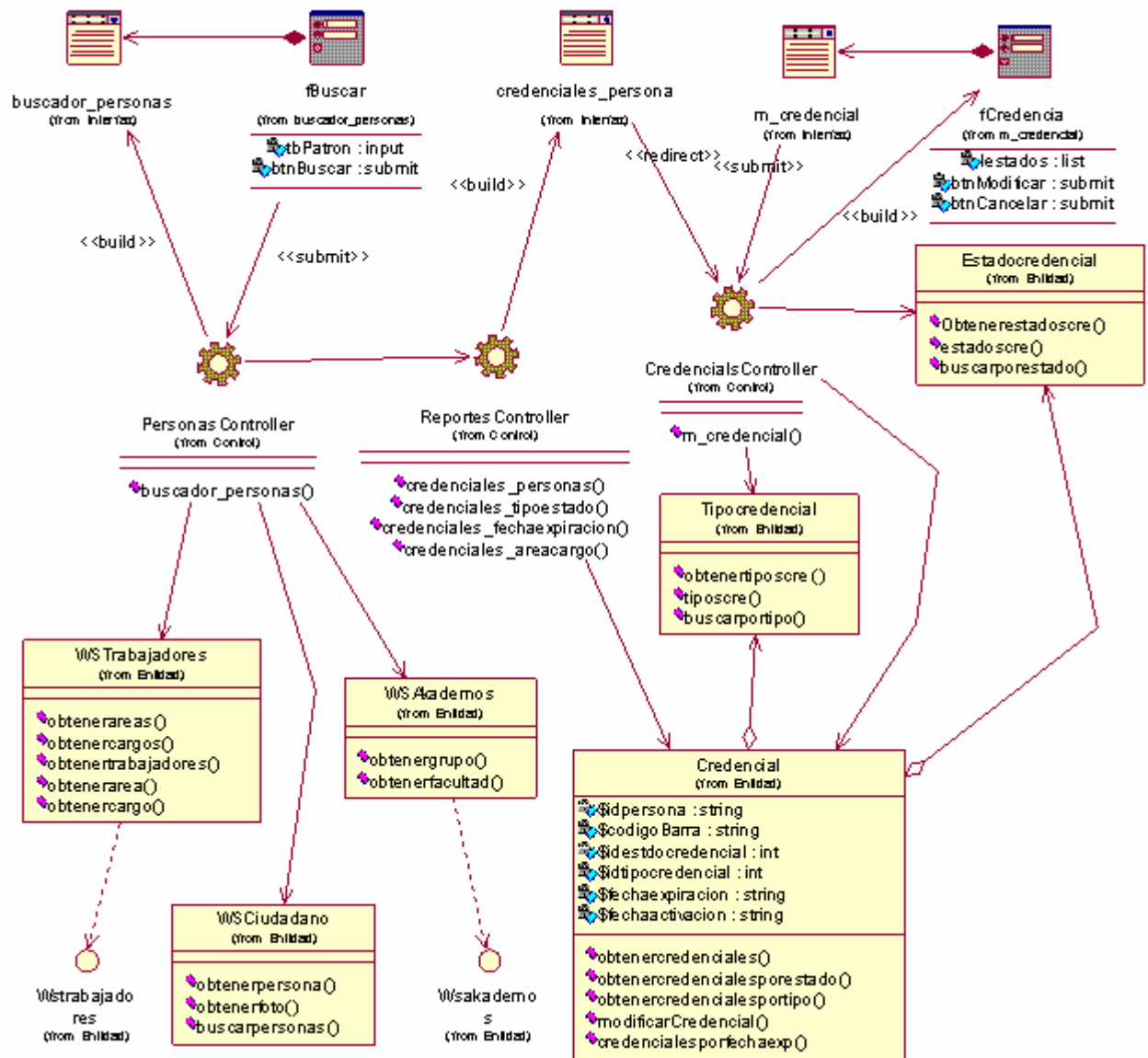


Figura 3.2 Diagrama de clases del diseño del CU_Modificar Credenciales.

Para más detalles se recomienda ver el diagrama de secuencia para la realización del caso de uso “Modificar Credenciales” (ver Anexo 1).

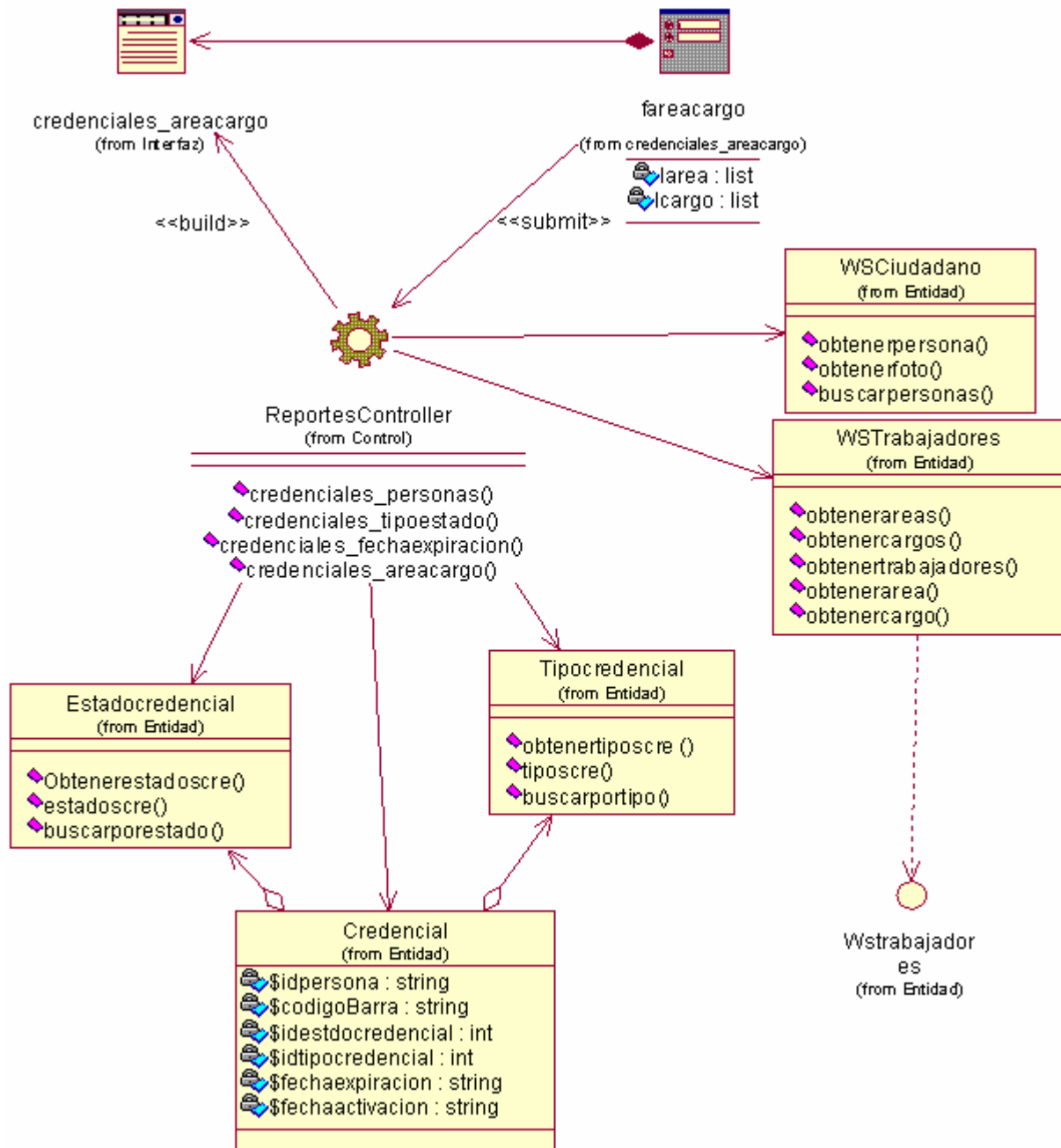


Figura 3.3 Diagrama de clases del diseño CU_Reporte por área o cargo.

Para más detalles se recomienda ver el diagrama de secuencia para la realización del caso de uso “Modificar Credenciales” (ver Anexo 1).

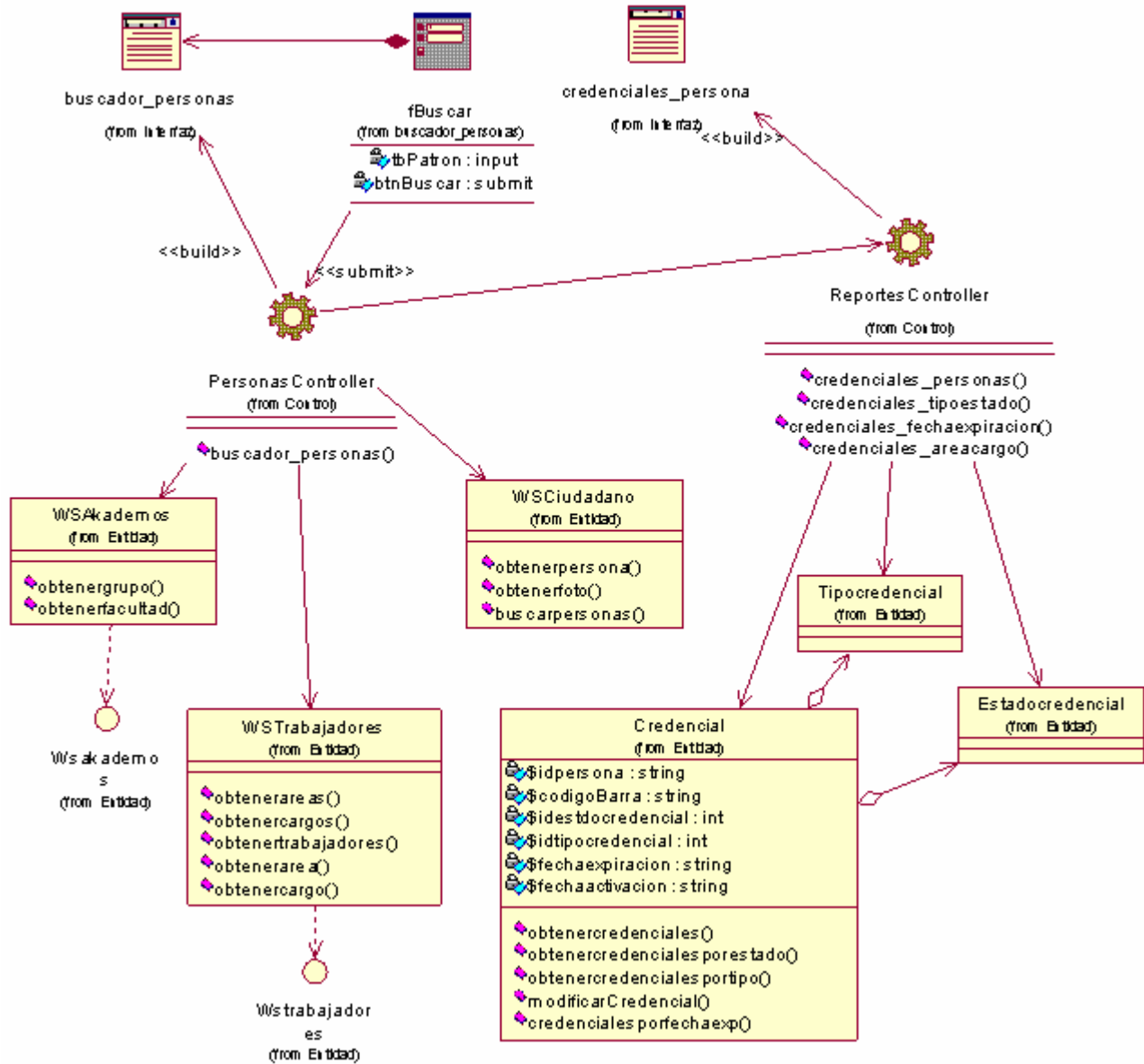


Figura 3.4 Diagrama de clases del diseño CU_Reporte por credenciales-persona.

Para más detalles se recomienda ver el diagrama de secuencia para la realización del caso de uso “Obtener reporte por credenciales que ha tenido una persona” (ver Anexo 1).

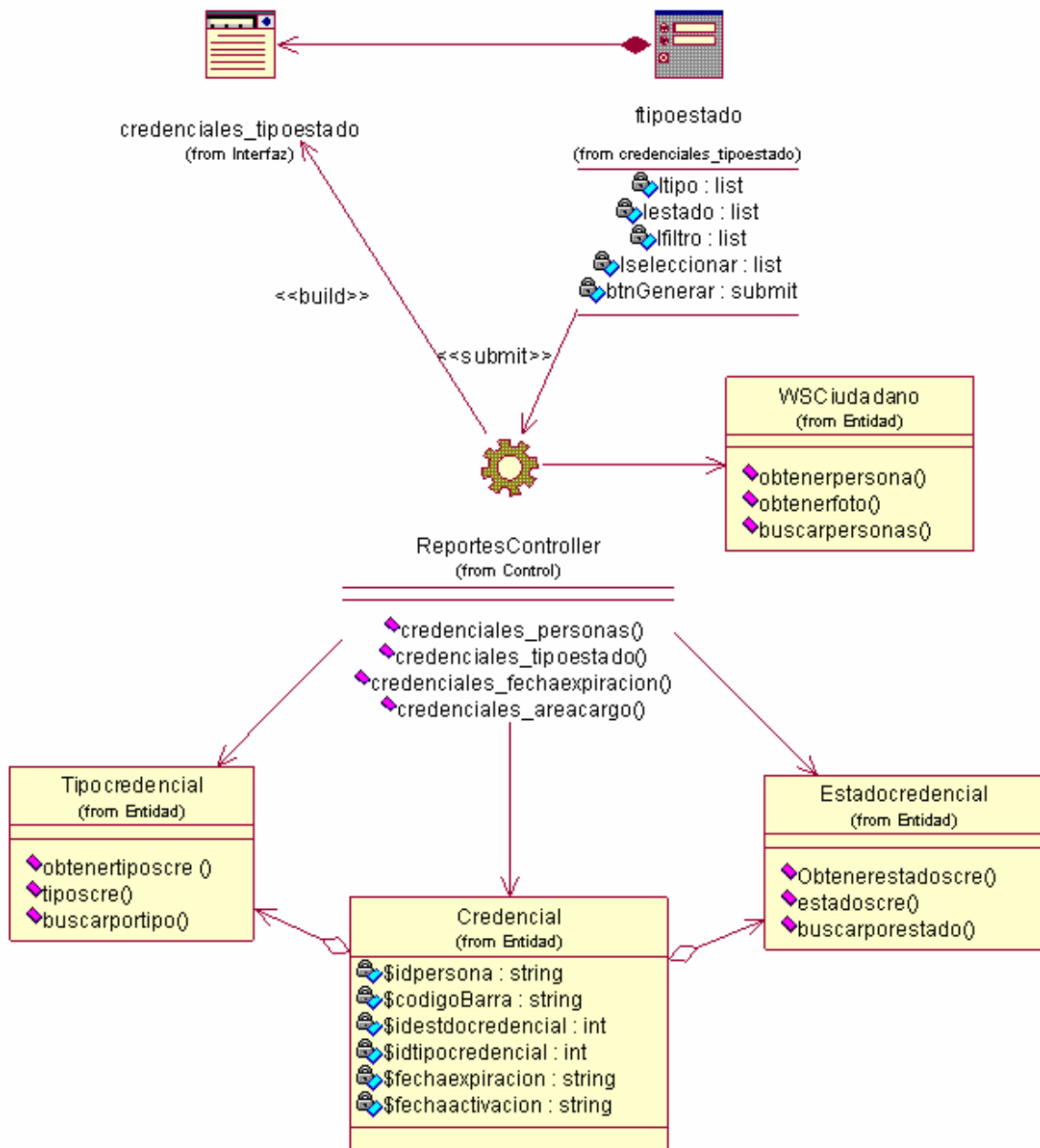


Figura 3.5 Diagrama de clases del diseño CU_Reporte por estado y por tipo.

Para más detalles se recomienda ver el diagrama de secuencia para la realización del caso de uso “Obtener reporte por estado y por tipo” (ver Anexo 1).

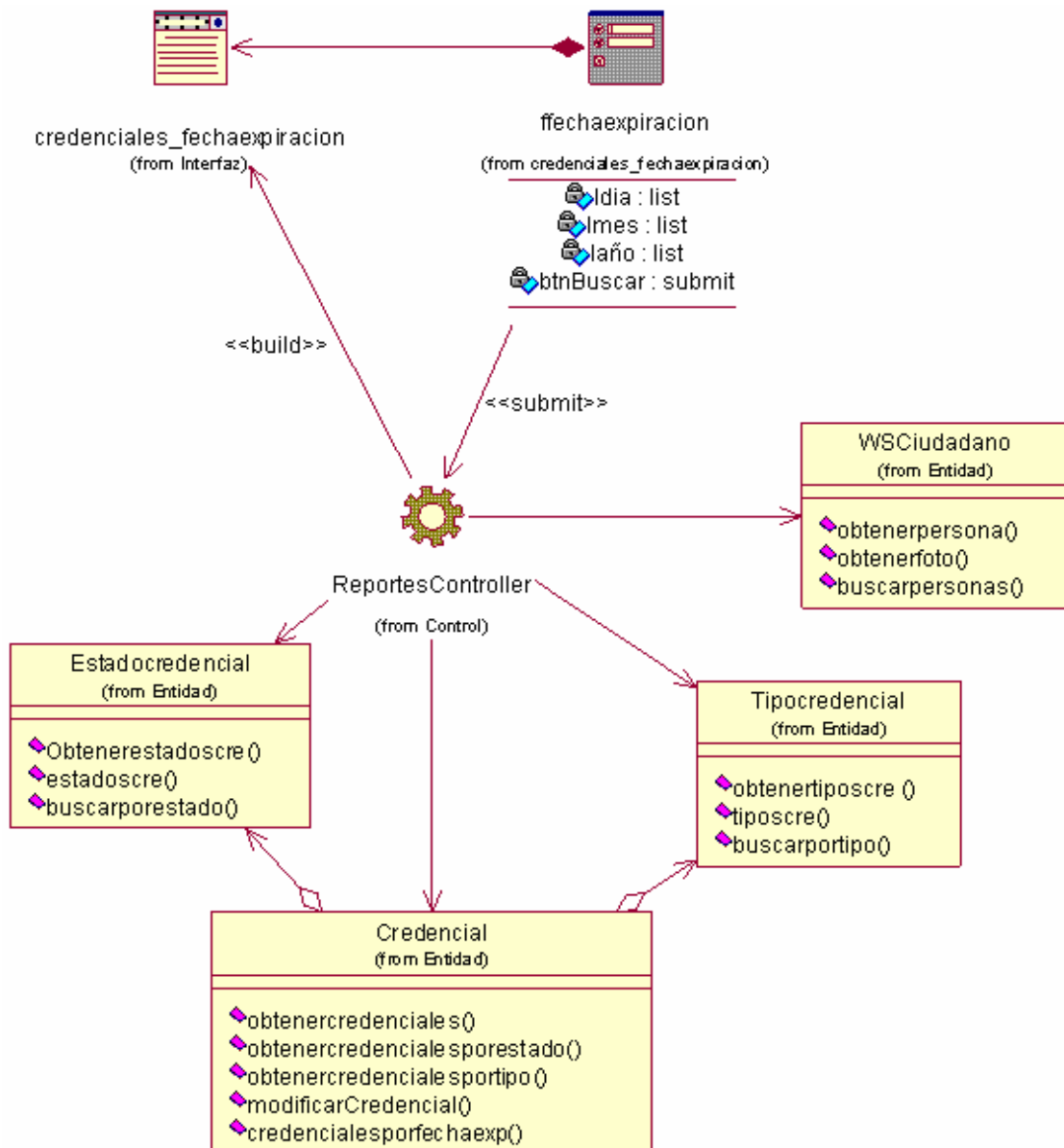


Figura 3.6 Diagrama de clases del diseño CU_Reporte por fecha de expiración

Para más detalles se recomienda ver el diagrama de secuencia para la realización del caso de uso “Obtener reporte por fecha de expiración” (ver Anexo 1).

3.2.2 Descripción de las clases

A continuación se hace una descripción de las clases del diseño las cuales fueron expuestas anteriormente. Esta sección tiene gran importancia porque describe a un desarrollador que se incorpore nuevo, las clases existentes brindándole información de los atributos y los métodos. Además esclarece a cualquier lector sobre las clases existentes y sus responsabilidades.

Tabla 18. Descripción de la buscador_personas.

Nombre: buscador_personas	
Tipo de clase Interfaz	
Atributo	Tipo
tbPatron	input
btnBuscar	submit
Para cada responsabilidad:	
Nombre:	buscar
Descripción:	Comienza el proceso de búsqueda de personas.

Tabla 19. Descripción de la clase m_credencial.

Nombre: m_credencial	
Tipo de clase Interfaz	
Atributo	Tipo
lestados	list

btnMoficar	submit
btnCancelar	submit
Para cada responsabilidad:	
Nombre:	modificar
Descripción:	Comienza el proceso de modificar las credenciales

Tabla 20. Descripción de la clase credenciales_tipoestado.

Nombre: credenciales_tipoestado	
Tipo de clase Interfaz	
Atributo	Tipo
ltipo	list
leestado	list
lfiltro	list
lseleccionar	list
btngenerar	submit
Para cada responsabilidad:	
Nombre:	buscar
Descripción:	Comienza la busqueda

Tabla 21. Descripción de la clase credenciales_fechaexpiración.

Nombre: credenciales_fechaexpiración	
Tipo de clase Interfaz	
Atributo	Tipo
Idia	list
lmes	list
laño	list
btnbuscar	submit
Para cada responsabilidad:	
Nombre:	buscar
Descripción:	Comienza la búsqueda

Tabla 22. Descripción de la clase credenciales_personas.

Nombre: credenciales_personas	
Tipo de clase Interfaz	
Para cada responsabilidad:	
Nombre:	mostrar
Descripción:	Muestra las credenciales que ha tenido una persona.

Tabla 23. Descripción de la clase credenciales_areacargo.

Nombre: credenciales_areacargo	
Tipo de clase Interfaz	
Atributo	Tipo
lcargo	list
larea	list
btngenerar	submit
Para cada responsabilidad:	
Nombre:	buscar
Descripción:	Comienza la búsqueda.

Tabla 24. Descripción de la clase CredencialesController.

Nombre: CredencialesController	
Tipo de clase Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:

m_credencial(\$idpersona,\$codigoBarra , \$idestado)	Modifica el estado de la credencial seleccionada.
---	---

Tabla 25. Descripción de la clase ReportesController.

Nombre: ReportesController	
Tipo de clase Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
credenciales_personas(\$idPersona)	Devuelve las credenciales que ha tenido cierta persona en la vista credenciales_personas. Recibe como parámetro el identificador de la persona.
Nombre:	Descripción:
credenciales_tipoestado(\$idtipo,\$idestado, \$filtro)	Devuelve las credenciales de las personas que tengan determinado tipo o determinado estado en la vista credenciales_tipoestado.
Nombre:	Descripción:
credenciales_fechaexpiracion(\$fecha)	Devuelve las credenciales con determinada fecha de expiración, recibe el parámetro \$fecha. Lo hace en la

	vista credenciales_fechaexpiracion.
Nombre:	Descripción:
credenciales_areacargo(\$area,\$cargo)	Devuelve las credenciales de las personas que tengan determinado cargo dentro de un área determinada.

Tabla 26. Descripción de la clase PersonasController.

Nombre: PersonasController	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Descripción:
buscador_personas(\$datos)	Devuelve los datos de las personas que cumplan con el patrón de búsqueda entrado y devuelve los datos en la vista buscador_personas.

Tabla 27. Descripción de la clase Credencial.

Nombre: Credencial	
Tipo de clase: Modelo	
Atributo	Tipo
ldpersona	string
codigoBarra	string

idEstadoCredencial	int
idTipoCredencial	int
FechaExp	string
Para cada responsabilidad:	
Nombre:	Descripción:
obtenercredenciales (\$idpersona)	Función que devuelve un arreglo con Tipo de Credencial, Estado de la Credencial, Fecha de Expiración, Código de barra, identificador de persona; dado el identificador de persona
Nombre:	Descripción:
Obtenercredencialesporestado (\$idestado)	Función que devuelve un arreglo con Tipo de Credencial, Estado de la Credencial, Fecha de Expiración, Código de barra, identificador de persona; dado el identificador de estado
Nombre:	Descripción:
obtenercredencialesportipo(\$idtipo)	Función que devuelve un arreglo con Tipo de Credencial, Estado de la Credencial, Fecha de Expiración, Código de barra, identificador de persona; dado el identificador de tipo de credencial.
Nombre:	Descripción:
modificarCredencial(\$idpersona, \$codigoBarra,\$estado)	Modifica la credencial que tenga determinado idpersona y código de barra.

Tabla 28. Descripción de la clase Estadocredencial.

Nombre: Estadocredencial	
Tipo de clase: Modelo	
Para cada responsabilidad:	
Nombre:	Descripción:
Obtenerestadoscre()	Devuelve un arreglo con los identificadores de estados
Nombre:	Descripción:
estadoscre()	Devuelve un arreglo con identificador de estado y el nombre de estado asociado al identificador.
Nombre:	Descripción:
buscarsestado(\$operador,\$estadocredencial)	Devuelve un arreglo con el identificador de persona, el código de barra, el estado y el tipo de la credencial además de la fecha de activación y expiración. Recibe como parámetros \$operador que es el tipo de búsqueda (igual o diferente) y \$estadocredencial que es el estado por el que se buscará.

Tabla 29. Descripción de la clase Tipocredencial.

Nombre: Tipocredencial

Tipo de clase: Modelo	
Para cada responsabilidad:	
Nombre:	Descripción:
obtenertiposcre ()	Devuelve un arreglo con los identificadores de tipos de credencial.
Nombre:	Descripción:
tiposcre()	Devuelve un arreglo con identificador de tipo y el nombre de tipo de credencial asociado al identificador.
Nombre:	Descripción:
buscarportipo(\$operador,\$tipocredencial)	Devuelve un arreglo con el identificador de persona, el código de barra, el estado y el tipo de la credencial además de la fecha de activación y expiración. Recibe como parámetros \$operador que es el tipo de búsqueda (igual o diferente) y \$tipocredencial que es el estado por el que se buscará.

Tabla 30. Descripción de la clase WSAkademoss.

Nombre: WSAkademoss
Tipo de clase: Modelo
Para cada responsabilidad:

Nombre:	Descripción:
obtenergrupo(\$aidpersona)	Llama al método ObtenerdatosEstudiante del wsAkademos devolviendo el grupo al que pertenece.
Nombre:	Descripción:
obtenerfacultad(\$aidpersona)	Llama al método ObtenerdatosEstudiante del wsAkademos devolviendo la facultad a la que pertenece.

Tabla 31. Descripción de la clase WSCiudadano.

Nombre: WSCiudadano	
Tipo de clase: Modelo	
Para cada responsabilidad:	
Nombre:	Descripción:
buscarpersonas(\$apatron)	Función que devuelve un arreglo con identificador de persona, tipo de persona, primer nombre, segundo nombre, primer apellido, segundo apellido. Recibe como parámetro \$apatron que es una cadena donde se puede entrar cualquier dato de la persona o combinación de ellos.
Nombre:	Descripción:
obtenerfoto(\$aidpersona)	Devuelve una cadena con el nombre de la foto recibiendo

	el identificador de la persona.
Nombre:	Descripción:
obtenerpersona(\$aidpersona)	Devuelve los nombres y apellidos dado el identificador de la persona.

Tabla 32. Descripción de la clase WSTrabajadores.

Nombre: WSTrabajadores	
Tipo de clase: Entidad	
Para cada responsabilidad:	
Nombre:	Descripción:
obtenerareas()	Devuelve arreglo con identificador de área y nombre de área. Llama al método obtenerAreas() del web services de trabajadores.
Nombre:	Descripción:
obtenercargos()	Devuelve arreglo con identificador de cargo y nombre de cargo. Llama al método obtenerCargos() del web services de trabajadores.
Nombre:	Descripción:
obtenertrabajadores(\$aidpersona)	Devuelve un arreglo con los identificadores de personas que correspondan a un area y/o cargo determinado. Llama al método obtenerTrabajadores() del web services

	de trabajadores.
--	------------------

3.2.3 Diseño de la Base de Datos

El almacenamiento de la información es uno de los aspectos más importantes para cualquier aplicación; por lo cual a continuación se modela base de datos que será utilizada en el sistema propuesto.

Diagrama Entidad Relación de la BD.

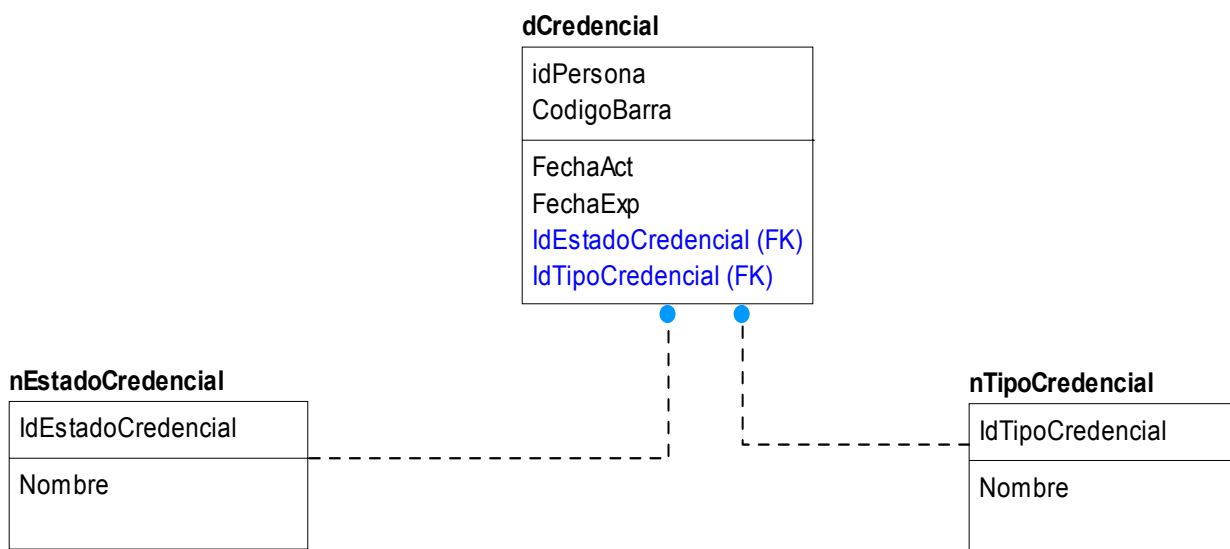


Figura 3.7 Diagrama Entidad Relación.

3.2.3.1 Descripción de las tablas

A continuación se describen cada una de las tablas mostradas en el diagrama entidad relación de la base de datos.

Tabla 33. Descripción de la tabla dCredencial.

Nombre: dCredencial		
Descripción: Esta tabla contiene los datos de las credenciales de todas las personas de la UCI		
Atributo	Tipo	Descripción
idPersona	nvarchar	Llave que contiene el identificador de cada persona
CodigoBarras	nvarchar	Llave que contiene el numero correspondiente al código de barras de la credencial
idTipoCredencial	int	Llave foránea que contiene el identificador del tipo de la credencial
idEstadoCredencial	int	Llave foránea que contiene el identificador del estado que presenta la credencial.
FechaAct	datetime	Este campo contiene la fecha en la credencial fue activada
FechaExp	datetime	Este campo contiene la fecha de expiración de la credencial

Tabla 34. Descripción de la tabla nEstadoCredencial.

Nombre: nEstadoCredencial		
Descripción: Esta tabla contiene los estados que puede presentar una credencial		
Atributo	Tipo	Descripción

IdEstadoCredencial	int	Llave que contiene el identificador de cada estado
Nombre	nvarchar	Nombre que tiene cada estado

Tabla 35. Descripción de la tabla nTipoCredencial.

Nombre: nTipoCredencial		
Descripción: Esta tabla contiene los estados que puede presentar una credencial		
Atributo	Tipo	Descripción
IdTipoCredencial	int	Llave que contiene el identificador de cada tipo de credencial
Nombre	nvarchar	Nombre que recibe cada tipo de credencial.

3.3 Tratamiento de errores

El tratamiento de errores es un factor a tener en cuenta a la hora de desarrollar cualquier aplicación. El sistema propuesto debe ser capaz del manejo de los errores que se presenten; de forma tal que afecten en la menor medida posible al usuario. Para evitar la entrada de errores por parte del usuario se reduce, en la medida de lo posible, la entrada manual de los datos. Otro aspecto importante será las validaciones del lado del cliente de manera tal que no lleguen al servidor peticiones incorrectas.

3.4 Seguridad

El tema de crear aplicaciones Web seguras es un tanto complejo, ya que requiere realizar un estudio para comprender los puntos vulnerables de la seguridad. Para la implementación del sistema propuesto se tuvo en cuenta que la información que se maneja es para el conocimiento de las personas interesadas dentro del área de acreditación. Por lo cual desde el primer momento se requerirá acceso a cualquier persona que intente hacer uso del sistema verificándose en primer lugar que sea usuario del dominio UCI.

3.5 Interfaz

La interfaz es la “cara” del sistema, lo cual tiene una gran importancia a la hora de hacer un diseño de la misma. Para lograr un entendimiento por parte del usuario se creó una interfaz sencilla de manera que el trabajo sea posible de forma intuitiva, sin grandes conocimientos de cómo funciona el sistema. Otro aspecto que se tuvo en cuenta es la organización de la información de forma tal que las informaciones que se generen tengan el mismo orden dentro de todas las páginas. La uniformidad es otro aspecto que se manejó pues cada elemento estará regido por el mismo estilo de tamaño, forma y color dentro del sitio.

3.6 Concepción de la ayuda

Dada la facilidad de uso que presenta el sistema propuesto, se implementará una ayuda que se encuentre en el contexto de la acción a realizar por parte del usuario. Es decir dentro de cada acción se incluirá un pequeño texto que indique qué se va a hacer en la misma.

3.7 Análisis de costo y beneficios

Dado el uso extendido y su probada eficacia, la metodología RUP fue la escogida para este trabajo. Esta metodología esta basada en Casos de Uso lo cual representa una ventaja a la hora de hacer un levantamiento correcto de los requerimientos del sistema. Los requerimientos quedan recogidos en términos de actores y casos de uso. Los actores representan, como se ha visto a lo largo del presente trabajo, tanto a un usuario humano como a otros sistemas informáticos que interactúan con el sistema en

cuestión. Los Casos de Uso son funcionalidades que le brindan valor a un actor determinado. Al ser las relaciones entre actores y casos de uso una forma de obtener los requerimientos pueden dar una medida de la cantidad de trabajo necesario para llevar a cabo el proyecto y por tanto su costo. La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente refinado por muchos otros autores. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. Para el cálculo del esfuerzo requerido de este proyecto será utilizado el método anteriormente expuesto.

Puntos de Casos de Uso

El primer paso es el cálculo de puntos de casos de uso sin ajustar. Lo cual se hace a partir de la siguiente ecuación:

$$\mathbf{UUCP = UAW + UUCW}$$

Donde:

UUCP: Puntos de casos de uso sin ajustar.

UAW: Factor de peso del los actores sin ajustar.

UUCW: Factor de peso de los casos de uso sin ajustar.

Cálculo de **UAW**

Tabla 36. Tipos de actores y sus pesos.

Actores del sistema:	Tipo de actor	Factor de Peso
Especialista	Complejo	3
Directivo	Complejo	3

UAW= Sumatoria de la multiplicación de la cantidad de actores de un tipo específico con su factor de peso.

Cantidad de actores de tipo complejo: 2

$$\mathbf{UAW} = 2 \times 3$$

$$\mathbf{UAW} = 6.$$

Cálculo de UUCW

Éste se calcula por los casos de uso del sistema y su complejidad. La complejidad está dada por la cantidad de secuencias atómicas que tenga el caso de uso.

Tabla 37. Tipos de Casos de Uso y sus pesos.

Casos de uso	Tipo de Caso de Uso	Factor de Peso
Obtener reporte por credenciales que ha tenido una persona	Simple	5
Obtener reporte por fecha de expiración	Simple	5
Obtener reporte por área y cargo	Simple	5
Obtener reporte por estado	Simple	5
Obtener reporte por tipo de credencial	Simple	5
Modificar credencial	Simple	5

UUCW = Sumatoria de la multiplicación de la cantidad de Casos de Uso de un tipo específico con su factor de peso.

Cantidad de casos de uso simples: 6

$$\mathbf{UUCW = 6 \times 5}$$

$$\mathbf{UUCW=30}$$

Al ser $\mathbf{UAW=6}$ y $\mathbf{UUCW=30}$ y $\mathbf{UUCP = UAW + UUCW}$ sustituyendo tenemos:

$$\mathbf{UUCP = UAW + UUCW}$$

$$\mathbf{UUCP = 6 + 30}$$

$$\mathbf{UUCP = 36}$$

Después del haber calculado los puntos de Casos de Uso sin ajustar se calculan los puntos de Casos de Uso ajustados.

$$\mathbf{UCP = UUCP \times TCF \times EF}$$

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Cálculo del Factor de complejidad técnica (**TCF**).

Se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante

Tabla 38. Factor de Complejidad Técnica.

Factor	Descripción	Peso	Valor asignado
T1	Sistema distribuido	2	1
T2	Objetivos de performance o tiempo de respuesta	1	1
T3	Eficiencia del usuario final	1	1
T4	Procesamiento interno complejo	1	1
T5	El código debe ser reutilizable	1	0
T6	Facilidad de instalación	0.5	2
T7	Facilidad de uso	0.5	2
T8	Portabilidad	2	2
T9	Facilidad de cambio	1	3
T10	Concurrencia	1	1
T11	Incluye objetivos especiales de seguridad	1	1
T12	Provee acceso directo a terceras partes	1	1
T13	Se requieren facilidades especiales de entrenamiento	1	0

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$\text{TCF} = 0.6 + 0.01 \times \Sigma (\text{Peso}_i \times \text{Valor asignado}_i)$$

$$\text{TCF} = 0.6 + 0.01 \times \Sigma (2 \times 1 + 1 \times 1 + 1 \times 1 + 1 \times 1 + 1 \times 0 + 0.5 \times 2 + 0.5 \times 2 + 2 \times 2 + 1 \times 4 + 1 \times 1 + 1 \times 1 + 1 \times 1 + 1 \times 0)$$

$$TCF=0.6 + 0.01 \times 17$$

$$TCF=0.6+0.17$$

$$TCF=0.78$$

Cálculo del Factor Ambiente (EF)

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos actores son los que se contemplan en el cálculo del Factor de ambiente.

Tabla 39. Factor Ambiente.

Factor	Descripción	Peso	Valor asignado
E1	Familiaridad con el modelo de proyecto utilizado	1.5	1
E2	Experiencia en la aplicación	0.5	1
E3	Experiencia en trabajo orientado a objetos	1	5
E4	Capacidad del analista líder	0.5	2
E5	Motivación	1	5
E6	Estabilidad de los requerimientos	2	5
E7	Personal a tiempo completo	-1	0
E8	Dificultad del lenguaje de programación	-1	0

El Factor de ambiente se calcula mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 \times \Sigma (\text{Peso} \times \text{Valor asignado})$$

$$EF = 1.4 - 0.03 \times \Sigma (1,5x3+0.5x3+1x5+0.5x3+1x5+2x5+(-1x0)+(-1x0))$$

$$EF = 1.4 - 0.03 \times 27.5$$

$$EF = 0.575$$

Al ser **UUCP = 36**, **TCF=0.77**, **EF = 0.575** y **UCP = UUCP x TCF x EF** entonces:

$$UCP=44 \times 0.78 \times 0.575$$

$$UCP=15,939$$

El esfuerzo en horas-hombre viene dado por:

$$E = UCP \times CF$$

Donde:

E= Esfuerzo

UCP= Puntos de Casos de Uso ajustados (calculado anteriormente).

CF= Factor de conversión (para este tipo de proyecto 20 horas-hombre/Punto de Casos de Uso)

Al tener **UCP= UCP=15,939**, **CF=20 horas-hombre/Punto de Casos de Uso** y **E = UCP x CF**

$$E=15,939 \times 20 \text{ horas-hombre/Punto de Casos de Uso}$$

$$E=318,78 \text{ horas-hombre}$$

Tabla 40. Esfuerzo por flujo de trabajo.

Actividad	Porcentaje	Esfuerzo
Análisis	10%	79,7
Diseño	20%	159,4

Implementación	40%	318,78
Prueba	15%	119.55
Sobrecarga	15%	119.55
Total	100%	797

El proyecto requerirá de 797 horas-hombre a desarrollar por una persona en un tiempo de 133 días; tomando que se trabajará en el proyecto como promedio 6 horas diarias. Por lo cual se puede decir que se terminará el proyecto en aproximadamente 4 meses ajustándose a los requerimientos del cliente.

3.7.1 Beneficios

El presente trabajo, como parte de las aplicaciones que se desarrollan con el objetivo de ser usadas dentro de la Universidad de las Ciencias Informáticas, no reporta un beneficio monetario directo. Sin embargo; desde el punto de vista de los beneficios intangibles que reporta, se puede decir que con el subsistema de manejo y reporte de credenciales, se humaniza el proceso de control dentro del área de acreditación. Además que permite un manejo mas inmediato de las informaciones que se generan en esa área.

Conclusiones

Con el desarrollo de este capítulo se logró una comprensión del problema que permitió la implementación del sistema propuesto. Se modeló el diagrama de clases del análisis que brindó una visión de lo que debe hacer el sistema. Con el diagrama de clases del diseño se logró determinar las clases que serán implementadas así como sus atributos y sus responsabilidades. Además se realizó un análisis de costo beneficio el cual demostró la factibilidad de llevar a cabo el proyecto.

CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA

Introducción

En este capítulo se expondrán el diagrama de despliegue y el diagrama de implementación. Estos diagramas son muy importantes dentro del flujo de trabajo de implementación porque representan la distribución del sistema propuesto. Se describirán además los componentes que son necesarios para que la aplicación propuesta funcione. El objetivo principal de esta disciplina es convertir los elementos del diseño en elementos de implementación, dichos elementos son códigos fuentes, ejecutables, binarios, entre otros.

4.1 Implementación

4.1.1 Diagrama de despliegue

El diagrama de despliegue representa la distribución física de la aplicación. La estructura de nodos que se utiliza en este, está diseñado para permitir que el usuario tenga comodidad en el trabajo además de permitir que halla una seguridad confiable. Se utiliza un servidor Web que tendrá la aplicación en sí; en otro nodo se encontrará la Base de Datos del sistema y por ultimo están las estaciones terminales que son las computadoras de la UCI desde donde se podrá acceder al sistema para utilizar los servicios que presta.

La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución, como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo:

- Dispositivos
- Procesadores
- Memoria

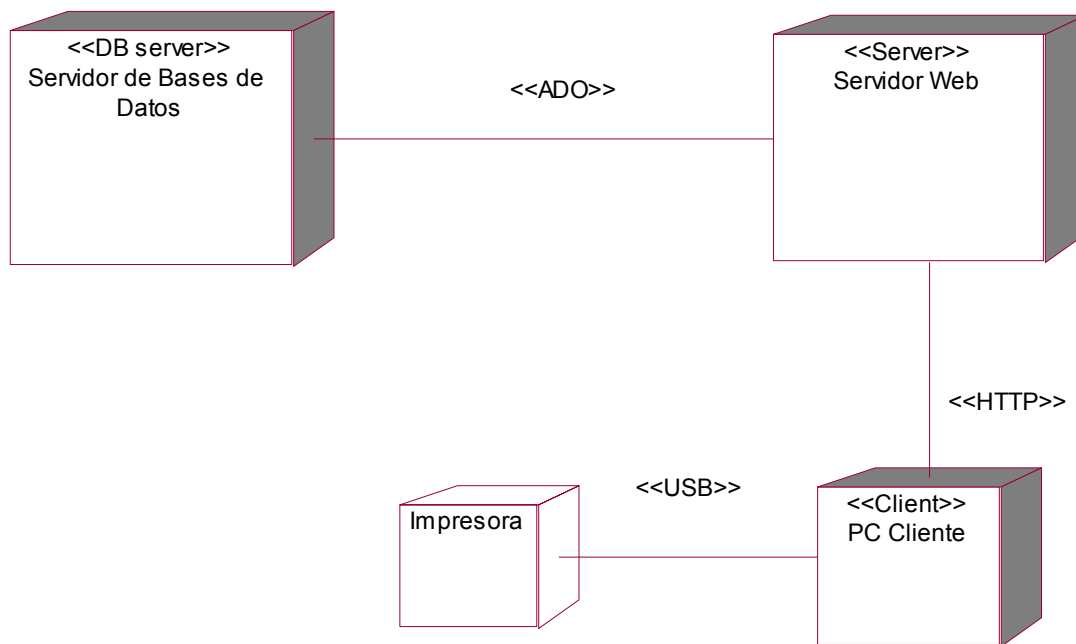


Figura 4.1 Diagrama de despliegue.

4.1.2 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, bibliotecas cargadas dinámicamente, etc. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

El diagrama de componente forma parte de la vista física de un sistema, la cual modela la estructura de implementación de la aplicación por sí misma, su organización en componentes y su despliegue en nodos de ejecución. Esta vista proporciona la oportunidad de establecer correspondencias entre las clases y los componentes de implementación y nodos. La vista de implementación se representa con los diagramas de componentes.

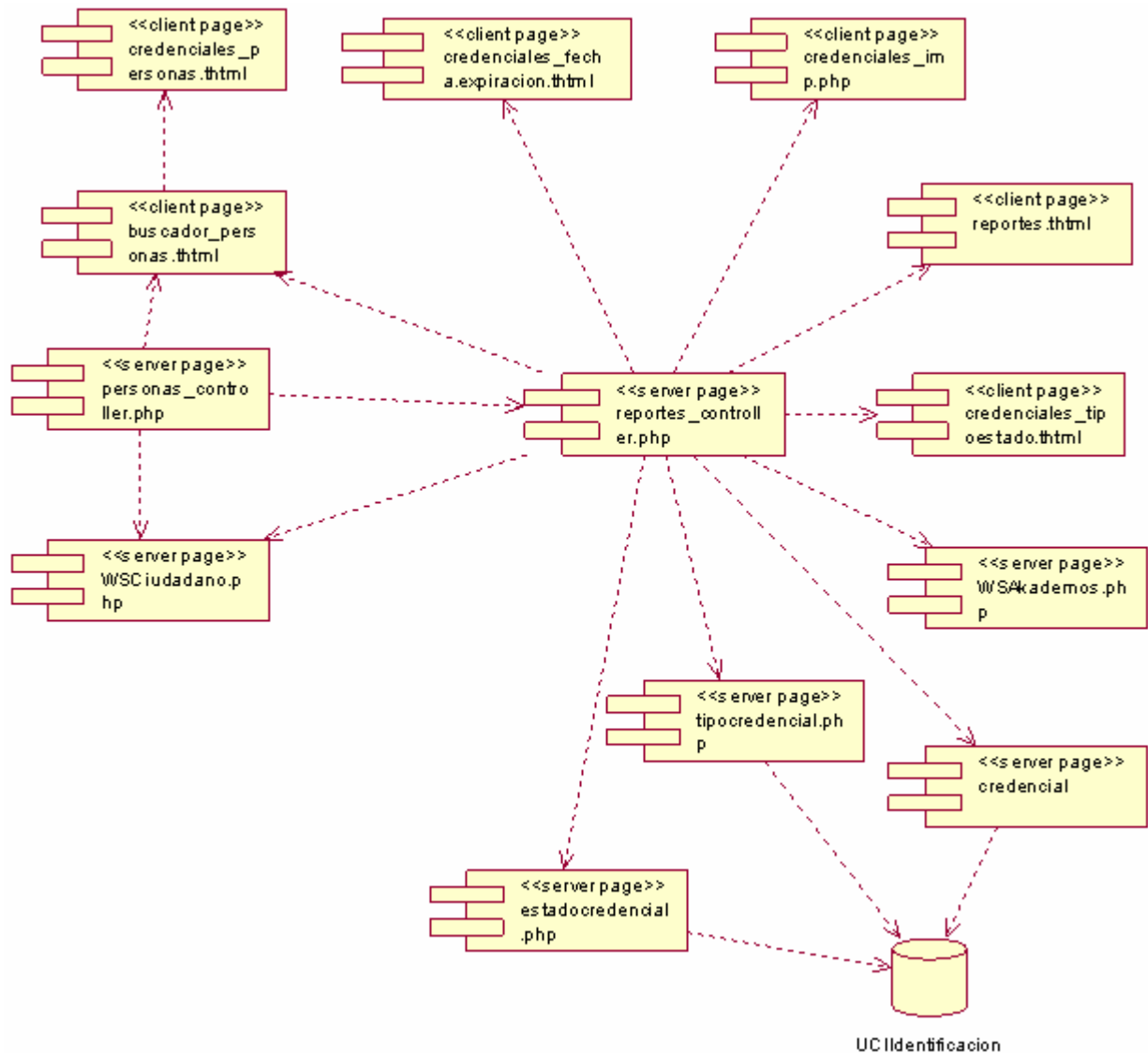


Figura 4.2 Diagrama de Componentes.

4.2 Modelo de Prueba

Las Pruebas de software se integran en las diferentes fases del Ciclo del software dentro de la Ingeniería de software. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores tiene.

Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas. Únicamente un proceso de verificación formal puede probar que no existen defectos.

Las pruebas de integración, prueban cada uno de los distintos componentes que componen la solución como un todo coherente. Esta prueba se lleva a cabo en el laboratorio de integración o de pruebas y está al cargo del propio grupo de trabajo de pruebas.

La calidad de un software es algo subjetivo que depende del contexto y del objeto que se pretenda conseguir. Para determinar dicho nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto a las especificaciones iniciales del sistema.

Las pruebas de software, testing, beta testing es un proceso usado para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador o videojuego. En el presente trabajo se hará un diseño de los casos de prueba de integración. En las siguientes pruebas se garantiza que en los casos que se espera una respuesta de la base de datos los datos sean insertados previamente.

4.2.1 Descripción de los casos de prueba de integración

Tabla 41. Caso de prueba de integración para Modificar credenciales.

Nombre del caso de uso: Modificar credenciales.			
Entrada	Resultados Esperados	Resultados de la prueba	Observaciones
1. Se introduce una cadena que contenga números y letras, separados por espacio. 84 ma bo	Persona que tenga dentro de su carné de identidad los números de la cadena y que su nombre o apellidos tengan la cadena de letras entrada, contenida dentro de ellos.	Mairelys Boeras Velazquez	Satisfactorio

	Mairelys Boeras Velazquez		
2. Se introduce un número 83121113	Devuelve las personas que tengan dentro de su carné de identidad la cadena de números entrada. Dasiel Otero Dartayet	Dasiel Otero Dartayet	Satisfactorio
3. Se introduce un Carné de Identidad válido pero que no encuentre en la base de datos. 53101013543	Mensaje de que no se obtuvo resultado por la búsqueda. La búsqueda no produjo resultado	No se encontraron resultados	Satisfactorio
4. Números y letras juntos sin espacio. 234hdhd	Mensaje de que no se obtuvo resultado por la búsqueda. La búsqueda no produjo resultado	No se encontraron resultados	Satisfactorio
5.Otros símbolos “. \$	Mensaje de que no se obtuvo resultado por la búsqueda. La búsqueda no produjo resultado	No se encontraron resultados	Satisfactorio

6. Modificar estado de la credencial. Dasiel Otero Dartayet Nuevo estado: Pendiente	Modifica y envía hacia la pagina de credenciales que ha tenido una persona mostrando la credencial como pendiente	Muestra la credencial con el estado en Pendiente	Satisfactorio.
---	---	--	----------------

Tabla 42. Caso de prueba de integración para Credenciales persona.

Nombre del caso de uso: Credenciales persona.			
Entrada	Resultados Esperados	Resultados de la prueba	Observaciones
1. Se introduce una cadena que contenga números y letras, separados por espacio. 84 ma bo	Persona que tenga dentro de su carné de identidad los números de la cadena y que su nombre o apellidos tengan la cadena de letras entrada, contenida dentro de ellos. Mairelys Boeras Velazquez	Mairelys Boeras Velazquez	Satisfactorio
2. Se introduce un número	Devuelve las personas que tengan dentro de su carné de identidad la	Dasiel Otero Dartayet	Satisfactorio

83121113	cadena de números entrada. Dasiel Otero Dartayet		
3. Se introduce un Carné de Identidad válido pero que no encuentre en la base de datos. 53101013543	Mensaje de que no se obtuvo resultado por la búsqueda. La búsqueda no produjo resultado	La búsqueda no produjo resultado	Satisfactorio
4. Números y letras juntos sin espacio. 234hdhd	Mensaje de que no se obtuvo resultado por la búsqueda. La búsqueda no produjo resultado	La búsqueda no produjo resultado	Satisfactorio
5.Otros símbolos “.\$	Mensaje de que no se obtuvo resultado por la búsqueda. La búsqueda no produjo resultado	La búsqueda no produjo resultado	Satisfactorio

Tabla 43. Caso de prueba de integración para reporte por tipo y estado.

Nombre del caso de uso: Reporte por tipo y estado.			
Entrada	Resultados Esperados	Resultados de la prueba	Observaciones

1. Se selecciona en el menú desplegable la opción tipo.	Mostrar un menú desplegable con los tipos de credencial.	Menú de tipos de credenciales	Satisfactorio
2. Se selecciona en el menú desplegable la opción estado.	Mostrar un menú desplegable con los estados de credencial.	Menú de estados de credencial	Satisfactorio
3. Tipo de credenciales Profesores	Todas las credenciales que cumplan con la condición sino mostrar que la credencial no ha sido encontrada.	Muestra los tipos de credencial de profesor	Satisfactorio
4. Estado de credenciales Pendiente	Todas las credenciales que cumplan con la condición sino mostrar que la credencial no ha sido encontrada.	Muestra credenciales con estado pendiente.	Satisfactorio

Tabla 44. Caso de prueba de integración para reporte por fecha de expiración.

Nombre del caso de uso: Reporte credenciales por fecha de expiración.			
Entrada	Resultados Esperados	Resultados de la prueba	Observaciones

<p>1. Se selecciona la fecha a entrar. Se da la opción buscar</p>	<p>Muestra las credenciales que expiren en la fecha seleccionada.</p>	<p>Muestra credenciales de esa fecha</p>	<p>Satisfactorio</p>
---	---	--	----------------------

Conclusiones

En este capítulo se representaron los componentes a ser implementados así como el diagrama de despliegue. Se logró obtener un mapa de la implementación del sistema propuesto; el cual sirvió de guía para lograr un software que responda a los requerimientos del cliente. Además se hizo el diseño de los casos de prueba de integración.

CONCLUSIONES

Con la realización del presente trabajo:

- Se desarrolló una aplicación capaz de modificar las credenciales y obtener reportes necesarios al área de acreditación.
- Se realizó un estudio del entorno de trabajo, que permitió comprender los procesos que se llevan a cabo en la actualidad.
- Se describieron los procesos, lo cual brindo una mayor claridad a la hora de implementar el sistema.
- Se modelaron conceptualmente las clases que están implicadas en el sistema.
- Se desarrollaron los diagramas de actividades, lo cual ayudo a comprender los procesos de negocio.
- Se describieron los casos de prueba de integración.
- Se integraron los conocimientos adquiridos en el transcurso de la carrera en las disciplinas de Ingeniería de Software y Programación.

RECOMENDACIONES

- Continuar el estudio de los procesos que se llevan a cabo con el objetivo de agregarle nuevas funcionalidades al sistema.
- Migrar las bases de datos que se utilizan a software libre, para cumplir con los lineamientos de la Universidad y el país.
- Asignación de mas recursos y personal para que este sistema evolucione en un futuro a la obtención de reportes de forma dinámica.

REFERENCIAS BIBLIOGRÁFICAS

- [1] REDK.NET. SRP - Sistema de Reporte de Proyectos. 2007. Disponible en: http://www.redk.net/index.php?option=com_content&task=view&id=27&Itemid=24
- [2] MSDN. Microsoft Solutions Framework. 2007. Disponible en: <http://www.microsoft.com/spanish/MSDN/estudiantes/ingsoft/planificacion/msf.msp>
- [3] REYNOX. Metodología de Desarrollo de Software (MDS). 2007. Disponible en: <http://www.reynox.com.ar/sistemas/metodologia.php#link2>
- [4] ACHOUR, MEHDI. Manual de PHP. 25 de febrero de 2006. Disponible en: <http://www.php.net/manual/es/introduction.php>
- [5] Wikipedia La enciclopedia libre. Framework. Disponible en: <http://es.wikipedia.org/wiki/Framework>
- [6] Página oficial de CakePHP. Disponible en: <http://www.cakephp.org>
- [7] Apexnet. Comparación de Herramientas de modelado UML Enterprise Architect y Rational Rose, 2005/06/15. Disponible en: <http://www.apexnet.com.ar/index.php/news/main/38/event=view>

BIBLIOGRAFÍA

1. ACHOUR, MEHDI. Manual de PHP. 25 de febrero de 2006. Disponible en: <http://www.php.net/manual/es/>
2. Arquitectura para los Sistemas que Conforman la Intranet Universitaria. Mayo 2007. Disponible en: <http://uddi/arquitectura.2007.5.9.pdf>
3. CASTEJÓN GARRIDO, JUAN SALVADOR. *Arquitectura y diseño de sistemas Web modernos*. 27 de diciembre de 2004. Disponible en: http://www.cii-murcia.es/informas/ene05/articulos/Arquitectura_y_diseno_de_sistemas_web_modernos.pdf
4. Extreme Programming: A gentle introduction, What is Extreme Programming?, 17 de febrero de 2006. Disponible en: <http://www.extremeprogramming.org/what.html>
5. FRANCO NAVARRO, J. UML en acción. Modelando Aplicaciones Web. Instituto Superior Politécnico José Antonio Echeverría (ISPJAE). [Consultado el 20 de Marzo de 2007].
6. HERNÁNDEZ GONZÁLEZ, ANAISA. Un Método para el Diseño de la Base de Datos a partir del Modelo Orientado a Objetos. Instituto Superior Politécnico José Antonio Echeverría (ISPJAE). [Consultado el: 12 de Diciembre de 2007].
7. MANAGEMENT GROUP. Introduction to OMG's Unified Modeling Language™ (UML®). 3 de abril de 2007. Disponible en: http://www.omg.org/gettingstarted/what_is_uml.htm
8. May 2007 Web Server Survey. 1 de mayo de 2007. Disponible en: http://news.netcraft.com/archives/web_server_survey.html
9. PERALTA, MARIO. ESTIMACIÓN DEL ESFUERZO BASADA EN CASOS DE USO. Disponible en: <http://www.itba.edu.ar/capis/rtis/rtis-6-1/estimacion-del-esfuerzo-basada-en-casos-de-usos.pdf>
10. REDK.NET. SRP - Sistema de Reporte de Proyectos. 2007. Disponible en: http://www.redk.net/index.php?option=com_content&task=view&id=27&Itemid=24
11. RIORDAN, REBECCA M. Aprenda Programación en Microsoft® SQLServer™ 2000 Ya. Disponible en: <http://bibliodoc.uci.cu/pdf/reg01713.pdf>

12. ROJAS PAJUELO, ROCÍO. Modelo de Objeto del Negocio. [Consultado el: 5 de Diciembre de 2006] Disponible en: <http://www.congreso.gob.pe/biblio/alerta/Alerta-7-06.pdf>
13. RUMBAUGH, JAMES. El lenguaje unificado de modelado. Manual de referencia. Disponible en: <http://bibliodoc.uci.cu/pdf/reg03050.pdf>

GLOSARIO DE TÉRMINOS

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

HTML: (*HyperText Markup Language*, lenguaje de marcas hipertextuales), lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

HTTP: (*HyperText Transfer Protocol*, protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la Web. El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceso a una página y la respuesta con el contenido.

JavaScript: Es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

MVC: Modelo Vista Controlador es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

OMG: (*Object Management Group*) es una corporación independiente y sin ánimo de lucro. Tiene el propósito de crear una arquitectura estándar para objetos distribuidos en redes (componentes).

UML: (*Unified Modeling Language*, Lenguaje Unificado de Modelado). Es un lenguaje de modelado de sistemas de software.

ANEXOS

Anexo1 Diagramas de secuencia.

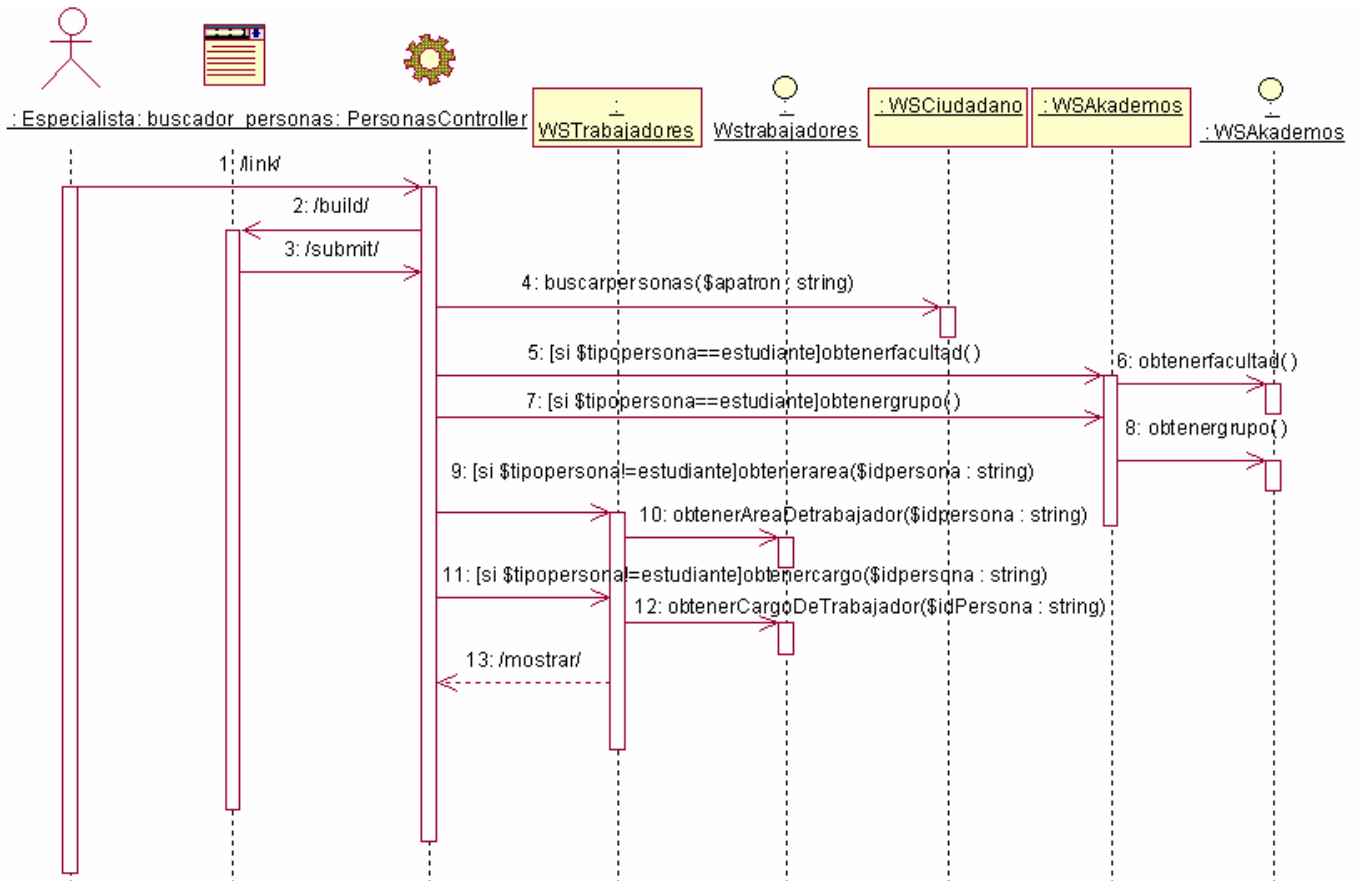


Figura 3.8 Diagrama de secuencia del CU Modificar credencial (Sección buscar Credencial).

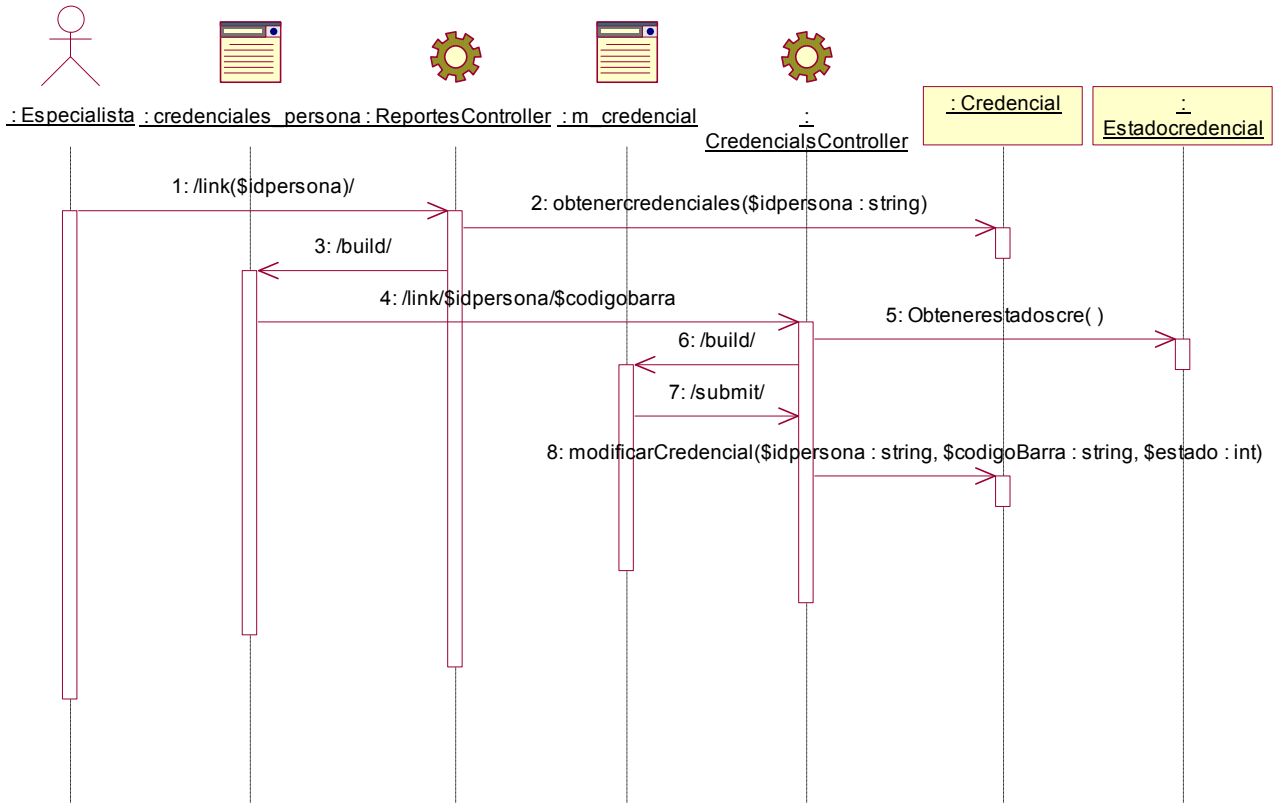


Figura 3.9 Diagrama de secuencia del CU Modificar credencial (Sección Modificar Credencial).

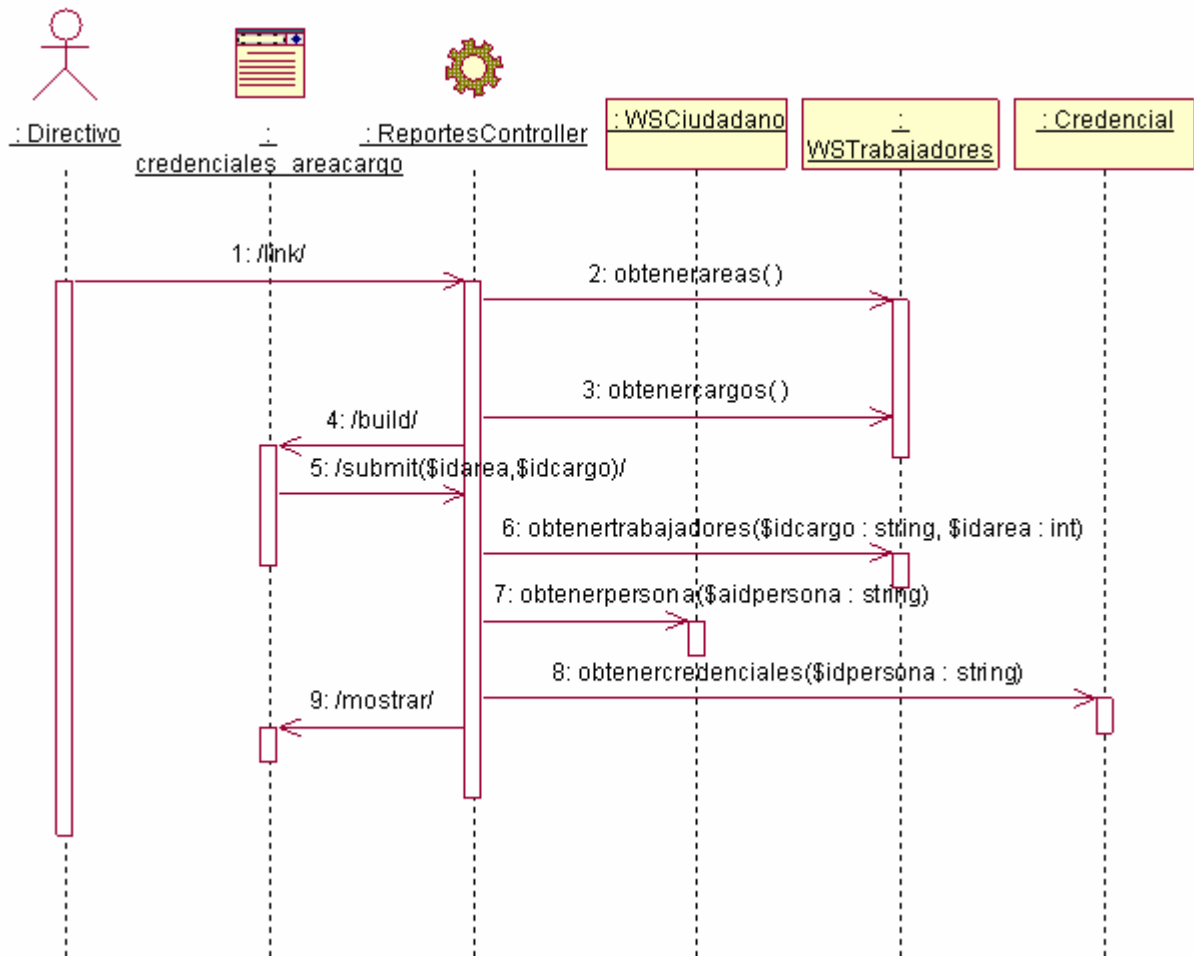


Figura 3.10 Diagrama de secuencia del CU Obtener reporte por área o cargo

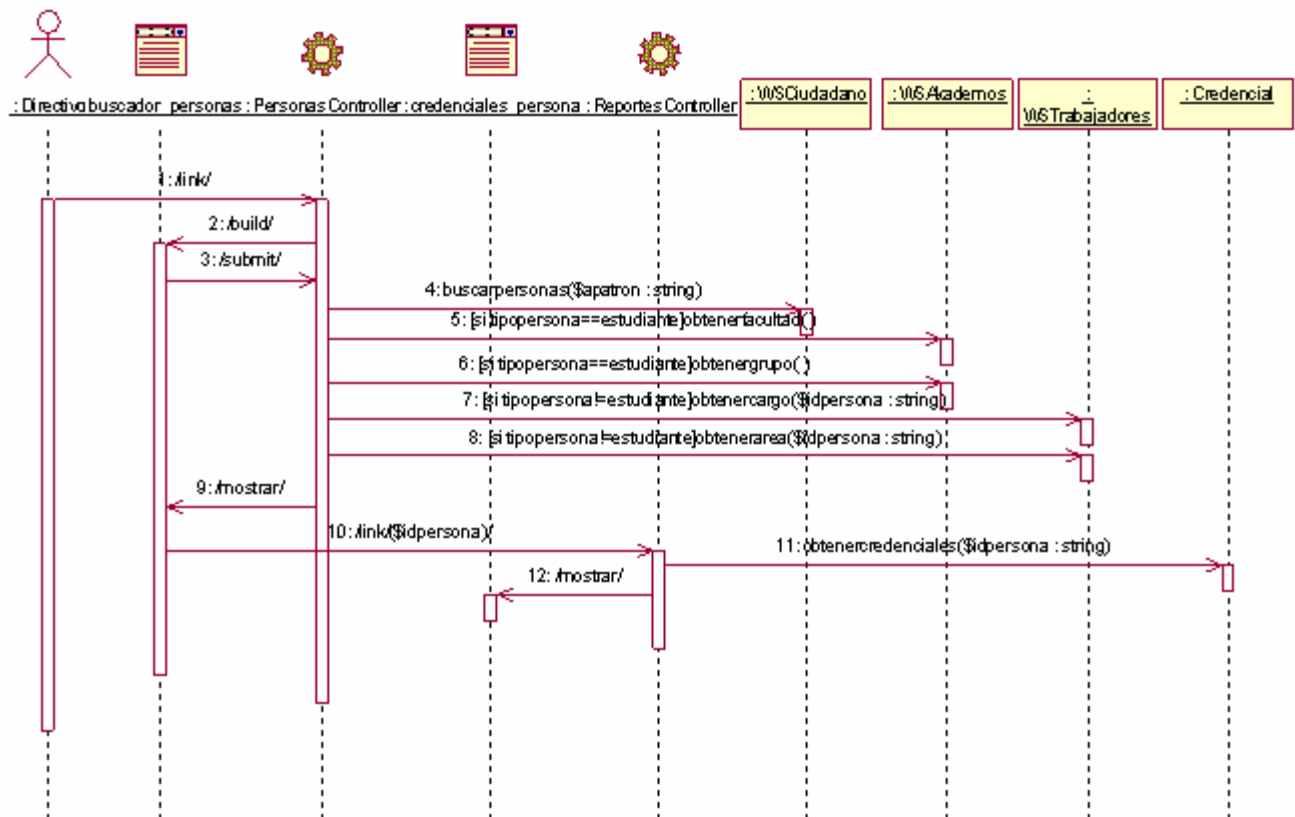


Figura 3.11 Diagrama de secuencia del CU Obtener reporte por credenciales persona

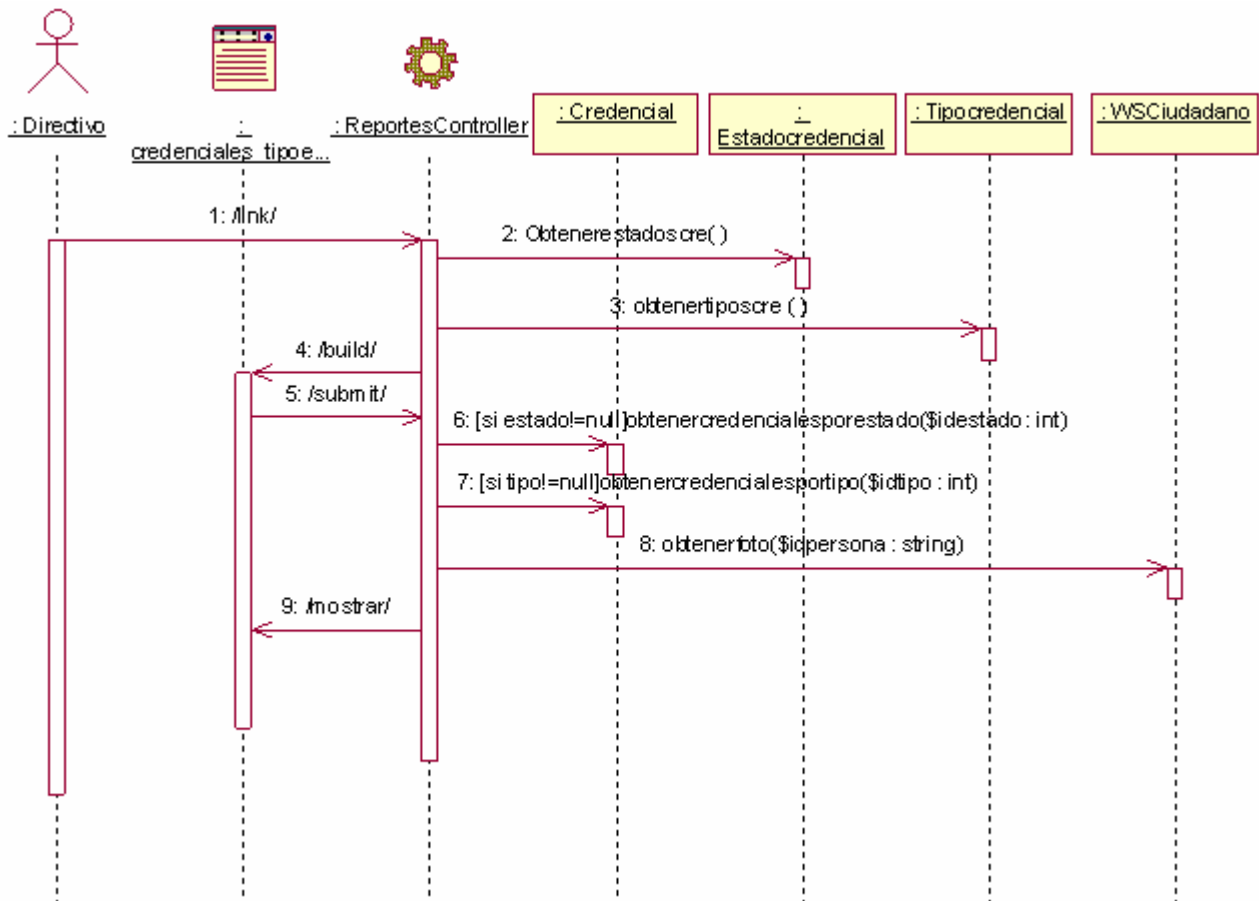


Figura 3.12 Diagrama de secuencia del CU Obtener Reporte por estado y tipo.

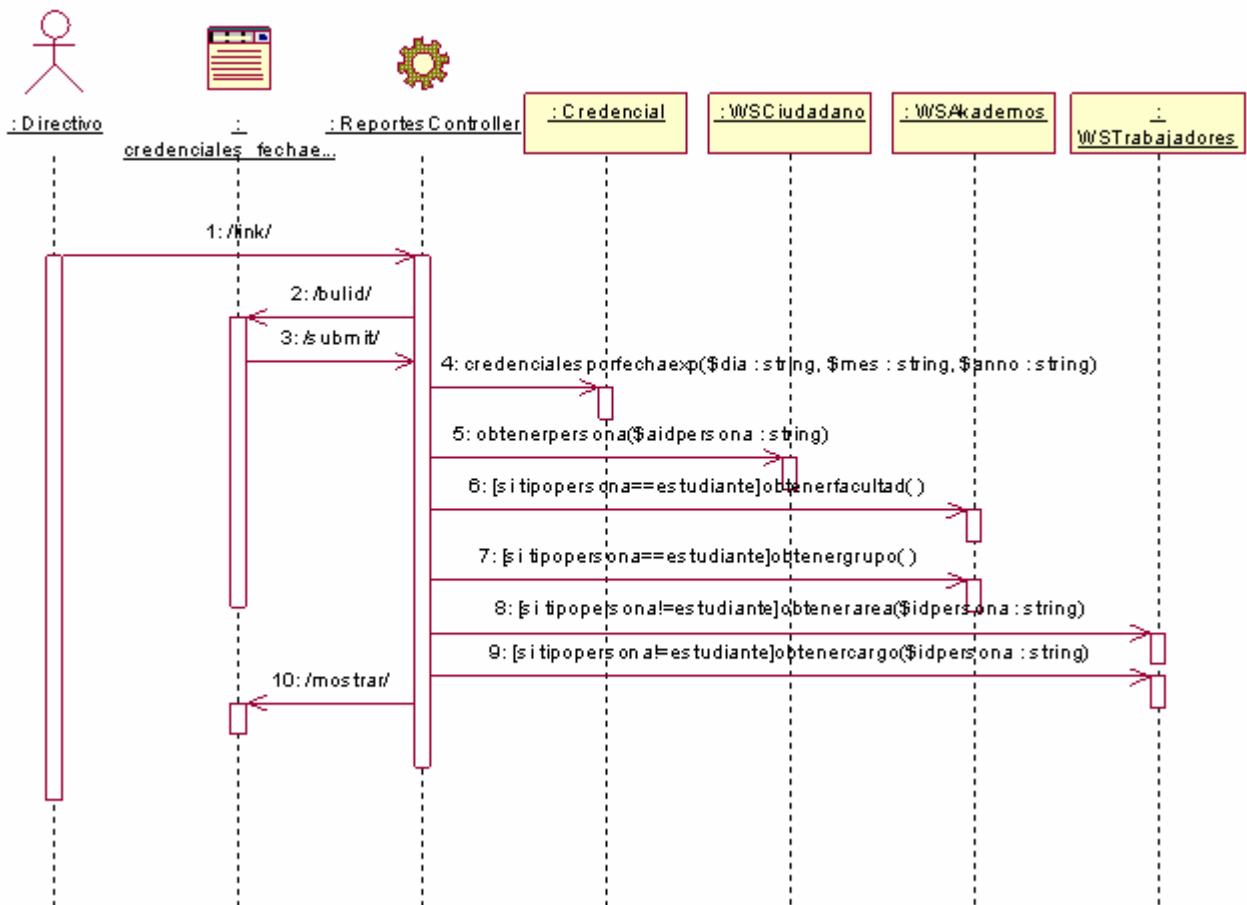


Figura 3.13 Diagrama de secuencia del CU Obtener Reporte por fecha de expiración.