

METODOLOGÍA SCRUM APLICADA A TRAVÉS DE UN SOFTWARE DE
CONSULTA DE LAS LISTAS OFAC Y ONU DE TOPBLS EN LAS PYME
COLOMBIANAS

SERGIO IVÁN BUSTOS VELÁSQUEZ

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA

Escuela de Ciencias Básicas, Tecnología e Ingeniería

Ingeniería de Sistemas

Bogotá, D.C., - Colombia

2014

METODOLOGÍA SCRUM APLICADA A TRAVÉS DE UN SOFTWARE DE
CONSULTA DE LAS LISTAS OFAC Y ONU DE TOPBLS EN LAS PYME
COLOMBIANAS

SERGIO IVÁN BUSTOS VELÁSQUEZ

Proyecto de Grado

Director: Carlos Eduardo Guevara Escobar

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA

Escuela de Ciencias Básicas, Tecnología e Ingeniería

Ingeniería de Sistemas

Bogotá, D.C., - Colombia

2014

Nota de aceptación

Presidente del Jurado

Jurado

Jurado

Bogotá, D.C., 04 de Septiembre de 2014

CONTENIDO

INTRODUCCIÓN.....	15
1 PLANTEAMIENTO DEL PROBLEMA.....	17
1.1 PROBLEMA PRINCIPAL.....	17
1.1.1 Delimitación del problema.....	17
2 ANÁLISIS DE VARIABLES	18
2.1 HERRAMIENTAS DE DESARROLLO	18
2.2 BASES DE DATOS.....	19
2.3 LENGUAJES DE PROGRAMACIÓN	19
3 OBJETIVOS.....	21
3.1 OBJETIVO GENERAL	21
3.2 OBJETIVOS ESPECÍFICOS	21
4 JUSTIFICACIÓN	22
4.1 JUSTIFICACIÓN SOCIAL	23
4.2 JUSTIFICACIÓN TECNOLÓGICA.....	24
4.3 JUSTIFICACIÓN LEGAL	25
5 HIPÓTESIS	27
6 MARCO TEÓRICO	28
6.1 ALCANCES.....	28
6.2 LÍMITES.....	28
6.3 ANTECEDENTES	29
6.3.1 TopBIs.....	29
6.3.1.1 Mae_Lista_Un (Tabla).....	30
6.3.1.2 Mae_Lista_Ofac (Tabla).....	31
6.3.2 DetectART.....	32
6.3.3 Limitaciones de las herramientas actuales.....	33
6.4 GUÍA DE INVESTIGACIÓN O DESARROLLO	33
6.5 METODOLOGÍA EMPLEADA	34
6.5.1 Metodología SCRUM.....	34

6.5.2	UML	34
6.6	HERRAMIENTAS.....	35
6.6.1	Microsoft Visual Studio 2012 Express Edition	35
6.6.2	Team Foundation Server 2012 Express Edition.....	36
6.6.3	Sql Server 2008 R2 Express Edition.....	37
7	MARCO CONCEPTUAL	38
7.1	INGENIERÍA DE SOFTWARE.....	38
7.1.1	Análisis	39
7.1.2	Especificación de los requerimientos (ERS)	42
7.1.3	Diseño y Arquitectura.....	43
7.1.4	Programación	45
7.1.5	Prueba (Testing)	45
7.1.6	Documentación	46
7.1.7	Mantenimiento.....	47
7.2	METODOLOGÍA DE DESARROLLO ÁGIL (SCRUM)	49
7.2.1	Selección de la Metodología.....	49
7.2.2	Casos de Éxito de SCRUM	51
7.2.3	Bases de la teoría	52
7.2.4	Equipo SCRUM (SCRUM TEAM)	53
7.2.5	Eventos	55
7.2.6	Artefactos	57
7.3	UML	59
7.3.1	Diagrama de Casos de Uso	59
7.3.2	Diagramas de Secuencia	62
7.3.3	Diagrama de Clases.....	62
7.3.4	Diagramas de Colaboración	64
7.3.5	Diagramas de Estado	65
7.3.6	Diagramas de Actividad.....	66
7.3.7	Diagramas de Relación de Entidad.....	67
7.4	ISO/IEC 9126.....	68
7.4.1	Funcionalidad	69
7.4.2	Fiabilidad	70

7.4.3	Usabilidad	71
7.4.4	Eficiencia	72
7.4.5	Mantenibilidad	73
7.4.6	Portabilidad	73
8.	DESARROLLO DEL PROYECTO	75
8.1	TIPO DE INVESTIGACIÓN.....	75
8.2	LÍNEA DE INVESTIGACIÓN.....	76
8.3	PRESUPUESTO.....	76
8.4	CRONOGRAMA	77
8.4.1	Línea de Tiempo	78
8.5	PLANEACIÓN DEL PROCESO SCRUM	78
8.5.1	Definición del Equipo.....	78
8.5.2	Definición de los Objetivos del Producto (Product Backlog)	79
8.5.3	Definición de los objetivos de las iteraciones (Sprint Backlog)	80
8.5.4	Planeación de las iteraciones (Sprints)	84
8.5.5	Definición de Terminado	85
8.6	REUNIÓN DE PLANIFICACIÓN DEL SPRINT 1 – 2014/04/01	85
8.6.1	Asistentes.....	85
8.6.2	¿Qué se completará en este Sprint? (Sprint Backlog).....	85
8.6.3	¿Cómo se conseguirá completar el trabajo seleccionado?.....	86
8.6.4	Acta de la Reunión	86
8.7	ANÁLISIS DEL NEGOCIO	87
8.7.1	Diagrama de Casos de Uso	87
8.7.2	Diagrama de clases para el análisis.....	87
8.7.3	Diagrama de Secuencias del Sistema	88
8.8	ANÁLISIS DE REQUERIMIENTOS	91
8.8.1	Requerimientos funcionales.....	91
8.8.2	Requerimientos no funcionales	94
8.9	DISEÑO LÓGICO	95
8.9.1	Diagramas de Secuencias de Transacciones	95
8.9.2	Diagramas de Colaboración	98
8.9.3	Diagrama de Clases para el Diseño.....	100

8.9.4	Diagramas de Actividades	103
8.9.5	Diagramas de Estado	106
8.10	REUNIÓN DE PLANIFICACIÓN DEL SPRINT 2 – 2014/04/17	107
8.10.1	Asistentes.....	107
8.10.2	¿Qué se completará en este Sprint? (Sprint Backlog).....	107
8.10.3	¿Cómo se conseguirá completar el trabajo seleccionado?	108
8.10.4	Acta de la Reunión	108
8.11	DISEÑO FÍSICO.....	109
8.11.1	Diagrama Entidad – Relación	109
8.11.2	Tablas	110
8.11.3	Vistas	113
8.11.4	Procedimientos Almacenados.....	113
8.12	INTERFAZ DE USUARIO – SPRINT 1.....	114
8.12.1	Menú de navegación.....	114
8.12.2	Pie de página	115
8.12.3	Formulario de Consultas	116
8.12.4	Formulario de Historial de Consultas	117
8.12.5	Formulario de Inicio de sesión	118
8.13	REUNIÓN DE PLANIFICACIÓN DEL SPRINT 3 – 2014/05/03	118
8.13.1	Asistentes.....	119
8.13.2	¿Qué se completará en este Sprint? (Sprint Backlog).....	119
8.13.3	¿Cómo se conseguirá completar el trabajo seleccionado?	119
8.13.4	Acta de la Reunión	120
8.14	INTERFAZ DE USUARIO – Sprint 3	120
8.14.1	Formulario de Administración de Usuarios (Consulta).....	121
8.14.2	Formulario de Administración de Usuarios (Mantenimiento).....	122
8.14.3	Formulario creación de licencias	122
8.14.4	Método de Generación de las Licencias	123
8.14.5	Algoritmo Rijndael.....	123
8.15	REUNIÓN DE PLANIFICACIÓN DEL SPRINT 4 – 2014/05/21	124
8.15.1	Asistentes.....	124
8.15.2	¿Qué se completará en este Sprint? (Sprint Backlog).....	125

8.15.3	¿Cómo se conseguirá completar el trabajo seleccionado?	125
8.15.4	Acta de la Reunión	126
9.	PRUEBAS (TESTING).....	127
9.1	Configuración De Las Pruebas (Testing)	127
9.2	Aprobación De Las Pruebas.....	132
10.	IMPLEMENTACIÓN	133
10.1	Diagrama de Componentes	133
10.2	Diagrama de Despliegue	135
10.3	Manual Técnico	136
10.4	Manual de Usuario.....	136
11.	ANEXOS	137
	CONCLUSIONES.....	141
	BIBLIOGRAFÍA.....	142

LISTA DE GRÁFICOS

Gráfico 1. Estructura de la tabla mae_lista_un en SqlServer	30
Gráfico 2. Estructura de la tabla mae_lista_ofac en Sql Server	31
Gráfico 3. Ciclo de vida del desarrollo de Software según Pressman	39
Gráfico 4. Ciclo del proceso Scrum.....	56
Gráfico 5. Actor UML.....	60
Gráfico 6. Casos de Uso UML	60
Gráfico 7. Relación de Asociación UML.....	60
Gráfico 8. Relación de Dependencia UML.....	61
Gráfico 9. Relación de Generalización: extensión y uso.	61
Gráfico 10. Diagrama de Secuencia.....	62
Gráfico 11. Diagrama de Clases: Clase	63
Gráfico 12. Diagrama de Clases: Relación de Generalización	63
Gráfico 13. Diagrama de Clases: Relación de Acumulación	63
Gráfico 14. Diagrama de Clases: Relación de Composición	64
Gráfico 15. Diagrama de Colaboración	65
Gráfico 16. Diagrama de Estados	66
Gráfico 17. Diagrama de Actividad.....	67
Gráfico 18. Diagrama Entidad-Relación	68
Gráfico 19. Línea de tiempo	78
Gráfico 20. Product Backlog	79
Gráfico 21. Sprint 1 Backlog	80
Gráfico 22. Sprint 2 Backlog	81
Gráfico 23. Sprint 3 Backlog	81
Gráfico 24. Sprint 4 Backlog	82
Gráfico 25. Acta de la reunión Sprint 1	86
Gráfico 26. Diagrama de Casos de Uso	87
Gráfico 27. Diagrama de clases para el análisis	87
Gráfico 28. Diagrama de Secuencia: Iniciar sesión.....	88
Gráfico 29. Diagrama de Secuencia: Consultar Individuo	89
Gráfico 30. Diagrama de Secuencia: Consultar Individuo (Cliente Web).....	89
Gráfico 31. Diagrama de Secuencia: Imprimir	90
Gráfico 32. Diagrama de Secuencia: Consultar Historial.....	90
Gráfico 33. Diagrama de Secuencias de Transacciones: Iniciar Sesión	95
Gráfico 34. Diagrama de Secuencias de Transacciones: Consultar Individuo.....	96
Gráfico 35. Diagrama de Secuencias de Transacciones: Consultar Individuo (Cliente Web).....	96
Gráfico 36. Diagrama de Secuencias de Transacciones: Imprimir	97
Gráfico 37. Diagrama de Secuencias de Transacciones: Consultar Historial	97

Gráfico 38. Diagrama de Colaboración: Iniciar Sesión	98
Gráfico 39. Diagrama de Colaboración: Consultar Individuo	98
Gráfico 40. Diagrama de Colaboración: Consultar Individuo (Cliente Web).....	99
Gráfico 41. Diagrama de Colaboración: Imprimir	99
Gráfico 42. Diagrama de Colaboración: Consultar Historial	100
Gráfico 43. Diagrama de Clases para el Diseño.....	100
Gráfico 44. Diagrama de Actividades: Iniciar Sesión.....	103
Gráfico 45. Diagrama de Actividades: Consultar Individuo.....	104
Gráfico 46. Diagrama de Actividades: Consultar Individuo (Cliente Web)	104
Gráfico 47. Diagrama de Actividades: Imprimir	105
Gráfico 48. Diagrama de Actividades: Consultar Historial	105
Gráfico 49. Diagrama de Estado: Iniciar Sesión	106
Gráfico 50 . Diagrama de Estado: Consultar Individuo - Usuario y Web.....	106
Gráfico 51. Acta de la reunión Sprint 2	108
Gráfico 52. Diagrama Entidad - Relación.....	109
Gráfico 53. Menú de Navegación.....	114
Gráfico 54. Pie de Página.....	115
Gráfico 55. Formulario de Consultas.....	116
Gráfico 56. Formulario de Historial de Consultas	117
Gráfico 57. Formulario de Inicio de Sesión	118
Gráfico 58. Acta de la reunión Sprint 3	120
Gráfico 59. Administración de Usuarios - Consulta	121
Gráfico 60. Administración de Usuarios - Mantenimiento	122
Gráfico 61. Generador de Licencias	122
Gráfico 62. Transformación de la Iteración del algoritmo Rijndael.	124
Gráfico 63. Acta de la reunión Sprint 4	126
Gráfico 64. Resultados de la ejecución de las pruebas.	132
Gráfico 65. Diagrama de Componentes	133
Gráfico 66. Diagrama de Despliegue	135

LISTA DE TABLAS

Tabla 1. Comparación entre las Metodologías	50
Tabla 2. Presupuesto	76
Tabla 3. Cronograma	77
Tabla 4. Product Backlog	79
Tabla 5. Lista total de objetivos	83
Tabla 6. Requerimientos Funcionales	91
Tabla 7. Requerimientos no funcionales.....	94
Tabla 8. Descripción de la tabla SegUsuarios.....	101
Tabla 9. Descripción de la tabla LogRegistro.....	101
Tabla 10. Descripción de la tabla LogConsulta.....	102
Tabla 11. Descripción de la tabla ResConsulta.....	102
Tabla 12. Descripción de la tabla LogEventos	103
Tabla 13. Descripción de la tabla SegUsuarios.....	110
Tabla 14. Descripción de la tabla LogRegistro.....	110
Tabla 15. Descripción de la tabla LogConsulta.....	111
Tabla 16. Descripción de la tabla ResConsulta.....	112
Tabla 17. Descripción de la tabla LogEventos	112
Tabla 18. Descripción de la vista VTopBlsData.....	113
Tabla 19. Descripción del Procedimiento Almacenado SP_CONSULTALISTAS	113
Tabla 20. Características de las pruebas	128

LISTA DE ANEXOS

Anexo 1. Acta Sprint 1	137
Anexo 2. Acta Sprint 2	138
Anexo 3. Acta Sprint 3	139
Anexo 4. Acta Sprint 4	140

RESUMEN

Este proyecto pretende mostrar cómo se realiza el desarrollo de una aplicación Web, utilizando la metodología SCRUM, la aplicación web creada tendrá como propósito apoyar una aplicación existente (TOPBLS), facilitando el acceso a los datos que ésta proporciona, de esta manera se expondrán las diferentes fases de la metodología y la forma como se resuelven cada uno de los incidentes a lo largo del proceso.

La aplicación web estará dividida en 3 módulos:

1. Aplicación Web de consultas
2. Servicio Web para otras aplicaciones
3. Aplicación móvil para prueba de concepto de servicio Web

Todo el proceso de desarrollo se realizará aplicando la metodología SCRUM.

Palabras clave: Aplicación, Web, Scrum, Metodología, Servicio, Desarrollo, Proyecto, UML, Ingeniería de Software.

ABSTRACT

This Project attempts to show how is done the development of a Web Application, using SCRUM methodology, the Web Application will have as a purpose to support an existing application (TOPBLS), facilitating the access to the data that it provides, thus the different phases of the methodology will be exposed, and the way how will be resolved each one of the incidents along the process.

The Web Application will be splitted in 3 modules:

1. Search Web Application
2. Web Services for another Applications
3. Mobile application for proof of concept of Web service

The entire development process will be conducted using the SCRUM methodology.

Keywords: Application, Web, Scrum, Methodology, Web Service, Development, Project, UML, Software Engineering.

INTRODUCCIÓN

El lavado de dinero, también conocido como blanqueo de dinero es una actividad ilegal que consiste en hacer que dineros obtenidos por medio de operaciones ilegales, como tráfico de drogas, contrabando, entre otras, se registren o ingresen al sistema financiero como si se tratase de actividades legales, para esto existen múltiples técnicas y operaciones.

El financiamiento al terrorismo y el lavado de dinero han estado vinculados desde hace varios años, y es por eso, que entidades internacionales como el Departamento del Tesoro Americano (*U.S. Department of the Treasury*), el Tesoro del Reino Unido (*HM Treasury*), y la Organización de las Naciones Unidas (*ONU*), entre otras, han publicado una variedad de listas sancionatorias, en donde se encuentran, personas y organizaciones que han sido identificadas como realizadores de estas actividades ilícitas

Este proyecto tiene como finalidad la construcción de un Aplicativo Web, éste se realizará utilizando la metodología SCRUM, y su propósito será consultar la información que la herramienta TOPBLS de Sidif Ltda. Provee de las listas SDN(OFAC) y ONU, y servirá como mecanismo de conocimiento del cliente, para las entidades vigiladas por la Superintendencia Financiera de Colombia.

Ya que el Lavado de Dinero es una de las principales fuentes de financiamiento del terrorismo, la mayor parte de los países han acordado realizar un seguimiento a las Organizaciones o Personas, para comprobar si utilizan esta modalidad para legalizar dinero o financiar el terrorismo, para esto se crean unas listas que especifican la información básica

de las organizaciones o personas vinculadas. Estas listas son de conocimiento público y se pueden acceder a ellas por cualquiera y desde distintos medios.

Estados Unidos y la ONU, son dos de las entidades que han puesto un mayor esfuerzo en el reconocimiento del Lavado de Activos y Financiación del Terrorismo (LAFT), por esta razón han publicado dos listas, sobre las cuales se va a desarrollar la aplicación de la que trata este proyecto, estas listas son:

- OFAC Specially Designated Nationals List
- Al-Qaida Sanctions List

En el caso Colombia, se permite que las entidades financieras tenga aplicaciones que apoyen el proceso de conocimiento del cliente por lo tanto el proyecto se llevará acabo para que las empresas tengan este conocimiento.

El análisis de estas listas trajo la necesidad de crear un nuevo formato y estructura para los datos, de manera que puedan ser entendidos por una persona sin conocimientos avanzados sobre el tema, y a su vez permitir la integración con otras plataformas de software, a través de un Servicio Web y una Aplicación Web. Para asegurar el correcto desarrollo de las aplicaciones todo el proceso se llevará a cabo utilizando la metodología SCRUM.

1 PLANTEAMIENTO DEL PROBLEMA

1.1 PROBLEMA PRINCIPAL

¿Existe una herramienta al alcance de las PYME que permita consultar individuos y entidades en las listas OFAC y ONU, y se pueda utilizar como mecanismo de control y conocimiento de las actividades vinculadas al lavado de activos y financiación del terrorismo, para dar conformidad a las instrucciones dadas por la Superintendencia Financiera de Colombia?

En la actualidad, muchas empresas, desconocen si sus clientes pertenecen al grupo de empresas o personas dedicadas a operaciones de lavado o financiación del terrorismo, es por ello que se hace necesaria el desarrollo de una aplicación que permita a las empresas vigiladas por la SFC, tener conocimiento de si su cliente se encuentra o no en estas listas.

1.1.1 Delimitación del problema

La aplicación que se va a desarrollar deberá ejecutarse en entorno Web, usará los datos existentes de la aplicación TOPBLS de Sidif Ltda. La información será válida únicamente para empresas vigiladas por la SFC, además no será una herramienta para la prevención del lavado de activos y financiamiento al terrorismo, sino como una estrategia de control que sea usada dentro el marco de las medidas implementadas por las empresas para tal fin.

2 ANÁLISIS DE VARIABLES

Para el desarrollo de este proyecto las variables a analizar son las Herramientas de desarrollo, Bases de Datos y Lenguajes de programación, basado en las condiciones actuales de la empresa SIDIF LTDA, que ha proporcionado las condiciones necesarias para llevarlo a cabo.

2.1 HERRAMIENTAS DE DESARROLLO

Las herramientas de desarrollo disponibles son:

Microsoft Visual Studio 2010: Contiene todas las herramientas necesarias para llevar a cabo el Proyecto, presenta un esquema de presentación y organización claro y funcional, permite administrar proyectos, archivos, recursos y admite la integración con sistemas de control de versiones.¹

Microsoft Visual Studio 2012: Contiene todas las herramientas necesarias para llevar a cabo el Proyecto, presenta un esquema de presentación y organización novedoso, integrado totalmente con Visual Studio Team Foundation 2012, presenta gran cantidad de ayudas de uso, es confiable y estable.²

¹ MICROSOFT CORPORATION, Microsoft Visual Studio Home, 2012. <http://www.visualstudio.com> Disponible en <http://www.visualstudio.com/es-es/visual-studio-homepage-vs.aspx>

² Ibíd.

2.2 BASES DE DATOS

Microsoft Sql Server 2008 R2: Es un potente gestor de bases de datos utilizado tanto en ambientes educativos como corporativos.³

2.3 LENGUAJES DE PROGRAMACIÓN

Visual Basic .NET: Es un lenguaje de alto nivel, multi-paradigma, permite programar orientado a objetos y estructurado, está diseñado para ser fácil de aprender y potente al mismo tiempo, utiliza una sintaxis similar a la de su predecesor Visual Basic 6, pero también apegado a las reglas de .NET Framework.⁴

C# .NET: Es un lenguaje de alto nivel, multi-paradigma, permite programar orientado a objetos y estructurado, sin embargo se especializa en la programación orientada a objetos, está diseñado para ser simple y potente usa tipos seguros (type-safe), y usa una sintaxis similar a la de C.⁵

ASP .NET: No es un lenguaje de programación en sí, pero da la impresión que lo es, ya que se integra en el código Html y Javascript, para permitir controles que automaticen y agilicen el proceso de escritura del código Web.⁶

³ Ibíd., Disponible en <http://www.microsoft.com/en-us/server-cloud/Products/sql-server-editions/sql-server-express.aspx>

⁴ Ibíd., Disponible en <http://msdn.microsoft.com/en-us/vstudio/hh388573>

⁵ Ibíd., Disponible en <http://msdn.microsoft.com/en-us/vstudio/hh341490>

⁶ Ibíd., Disponible en [http://msdn.microsoft.com/en-us/library/vstudio/4w3ex9c2\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/4w3ex9c2(v=vs.100).aspx)

HTML: Es el lenguaje estándar de internet, no existe un Proyecto basado en web que no use este lenguaje.⁷

Javascript: Es el lenguaje más común para programar las interacciones del lado del cliente.⁸

Css: No es un lenguaje de programación, pero se incluye como si lo fuera, debido a que se usa en la mayoría de los proyectos Web, permite definir un estil y apariencia unificada para los sitios Web.⁹

Xml: Es un lenguaje basado en etiquetas, normalmente se usa para el intercambio de información vía Web, es común en el uso de Servicios Web.¹⁰

⁷ WORLD WIDE WEB CONSORTIUM, Markup, <http://www.w3.org> Disponible en <http://www.w3.org/MarkUp/>

⁸ MICROSOFT CORPORATION, Op. Cit., Disponible en <http://msdn.microsoft.com/en-US/vstudio/jj673959>

⁹ WORLD WIDE WEB CONSORTIUM, Op. Cit., Disponible en <http://www.w3.org/Style/CSS/>

¹⁰ Ibíd, Disponible en <http://www.w3.org/XML/>

3 OBJETIVOS

3.1 OBJETIVO GENERAL

- Diseñar y desarrollar un aplicativo Web con la metodología SCRUM, que use los datos de la aplicación TOPBLS de Sidif Ltda, para consultar el listado SDN (OFAC) y ONU.

3.2 OBJETIVOS ESPECÍFICOS

- Permitir la consulta de las listas ONU y OFAC.
- Seguir el proceso de desarrollo de productos rápidos de SCRUM.
- Crear e implementar un algoritmo de búsqueda para consultas individuales.
- Crear un WebService que permita que otras aplicaciones obtengan la información y usar el algoritmo creado en el Servicio Web.
- Crear un generador de licencias, cada licencia debe autorizar el uso del software, la licencia debe ser por instalación.
- Todas las aplicaciones y WebServices construidos deben validar la licencia de uso.
- Desarrollar un sitio web para dispositivos móviles (Teléfonos móviles, Tablets, IPads), que sirva de prueba de concepto del Servicio Web, y que permita realizar consultas a las listas OFAC y ONU.

4 JUSTIFICACIÓN

Las empresas vigiladas por la SuperIntendencia Financiera de Colombia, no cuentan con una herramienta que les permita realizar consultas en las listas OFAC y ONU, e integrar estas consultas a su plataforma de negocio, generando un riesgo posible de lavado de activos, pues este delito según la Unidad de Información y Análisis Financiero – UIAF - consiste en “dar apariencia de legalidad a los recursos generados de sus actividades ilícitas”¹¹ en Colombia este delito busca ser evitado por medio de controles tales como la circular externa 022 de 2007 de la Superintendencia Financiera de Colombia, que indica los mecanismos que deben adoptar las entidades para su prevención.

Por esta razón resulta importante el desarrollo de una aplicación que permita conocer información acerca de si el individuo o entidad con la que se hace negocios se encuentra reportada dentro de las listas de sanción al Lavado de Activos y Financiamiento del Terrorismo.

¹¹ UNIDAD DE INFORMACIÓN Y ANÁLISIS FINANCIERO, Lavado de Activos, 2014, www.uiaf.gov.co Disponible en <http://www.uiaf.gov.co/?idcategoria=6965>

4.1 JUSTIFICACIÓN SOCIAL

El conocimiento del cliente es una responsabilidad de todas entidades vigiladas por la Superintendencia Financiera de Colombia, por ello, el desarrollo de esta aplicación, permitirá a estas entidades conocer si un individuo o entidad ha sido mencionado en las listas OFAC y ONU, disminuyendo la posibilidad de participar indirectamente, en una operación de Lavado de Dinero, lo que tiene consecuencias penales graves.

El impacto que tienen los dineros generados a partir de actividades ilegales, sobre las finanzas de un país, repercute directamente sobre el bienestar social de sus ciudadanos, el PIB (Producto Interno Bruto) de Colombia en 2013 fue 378.148 Millones de Dólares¹² y se estima que el Lavado de Dinero durante ese mismo año llegó a los 9.354 Millones de Dólares¹³.

Fecha	PIB Mill. USD\$	Lavado Mill. USD\$ (Aprox)
2013	378.148\$	9.354\$
2012	369.789\$	11.100\$
2011	336.346\$	8.314\$
2010	287.018\$	8.574\$

¹² DATOS MACRO, Colombia PIB, 2014, [www.datosmacro.com](http://www.datosmacro.com/pib/colombia) Disponible en <http://www.datosmacro.com/pib/colombia>

¹³ UNIDAD DE INFORMACIÓN Y ANÁLISIS FINANCIERO, Lavado de Activos, Op. Cit, 2013.

El Lavado de Dinero, también conocido como Blanqueo de Dinero, es el nombre que se le ha dado a las operaciones o actividades que se realizan para ocultar la procedencia ilícita de estos dineros, convirtiéndolos en legales, a través de “negocios fachada”, movimientos con bajos montos y muchas otras actividades más.

Al cerrarle las puertas al Lavado de Dinero y Financiación del Terrorismo (LAFT), el país se verá beneficiado y con él toda la sociedad, ya que el dinero circulante será legal y pasará por el ciclo normal tributario, lo que deja grandes beneficios económicos al país, que se pueden utilizar en proyectos de carácter social, infraestructura y otros.

4.2 JUSTIFICACIÓN TECNOLÓGICA

Hoy en día la tecnología se encuentra presente en la educación, comunicación y en otros aspectos de la vida cotidiana, tales como el proceso de aprendizaje, inventarios empresariales y caseros, telefonía, diversión, etc., pero no lo hace con un nivel de profundización muy fuerte, según el Ministerio de Tecnologías de la Información y Comunicaciones, sólo el 7% de las PYME se encontraban conectadas a internet en 2013, para este año la cifra supera el 20%, con los programas promovidos por MINTIC, se espera que esta cifra crezca de manera

acelerada, pues el 96.4% del sector productivo del país, está conformado por Micro, Medianas y pequeñas¹⁴.

La aplicación proveerá herramientas tecnológicas, como un Web Service y una Web de consulta, lo que permitirá a las PYME realizar su trabajo integrando esta información a su plataforma de negocio y adoptar medidas que le permitan confiar en que el individuo o entidad con la que va a realizar negocios no se encuentra en una de las listas restrictivas.

La metodología SCRUM, bajo la cual será desarrollada la solución de software, tiene como propósito asegurar el cumplimiento de una secuencia, que garantice el éxito del desarrollo, para el usuario esto se ve reflejado en soluciones de Software de alta calidad y óptimo desempeño.

4.3 JUSTIFICACIÓN LEGAL

El Lavado de Dinero es un delito contemplado en la Ley 599 de 2000 e impone multas de hasta 50.000 salarios mínimos legales mensuales vigentes y hasta 15 años de prisión (art. 323), en esta misma ley se determina que la omisión de control incurrirá en prisión hasta de 6 años y una multa hasta de 10.000 salarios mínimos legales mensuales vigentes.

¹⁴ MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y TELECOMUNICACIONES, Vive Digital, 2014, <http://www.mintic.gov.co> Disponible en <http://www.mintic.gov.co/portal/vivedigital/612/w3-propertyvalue-7235.html>

En la circular externa 022 de 2007 de la Superintendencia Financiera de Colombia, se encuentran las instrucciones relativas a la administración del riesgo de lavado de activos y de la financiación del terrorismo, y se define como:

“Se entiende por LA/FT, la posibilidad de pérdida o daño que puede sufrir una entidad vigilada por la SFC, por su propensión a ser utilizada directa o a través de sus operaciones como instrumento para el lavado de activos o la canalización de recursos para la Financiación del Terrorismo.”¹⁵ En el cual se definen dos fases:

I. Prevención del Riesgo: Que tiene como fin prevenir que se introduzca al sistema financiero recursos provenientes del Lavado de Activos y Financiación del Terrorismo (LA/FT)¹⁶.

2. Control: Busca detectar operaciones que se pretendan realizar o se hayan realizado¹⁷. Dentro de las etapas de SARLAFT que deben implementar las entidades (Identificación del riesgo, Medición o Evaluación, Control y Monitoreo), el proyecto actual pretende servir como apoyo dentro de la etapa de Control, ya que los datos que se integran por medio de TOPBLS hacen parte de la Identificación del Riesgo, y las consultas realizadas por medio de la aplicación construida en este proyecto, permiten determinar si el individuo o entidad que se vincula o se realizan operaciones, presenta riesgo operativo.

¹⁵ SUPERINTENDENCIA FINANCIERA DE COLOMBIA, Circular externa 022, 2007, Cap 11, p. 74

¹⁶ Ibíd., Circular externa 022, 2007, p. 1

¹⁷ Ibíd., Circular externa 022, 2007, p. 1

5 HIPÓTESIS

¿Es posible utilizar SCRUM en un proyecto de corta duración y baja complejidad, como crear una aplicación y servicio web que permita consultar individuos y entidades en las listas OFAC y ONU aplicando los principios de la Ingeniería de Software?

6 MARCO TEÓRICO

6.1 ALCANCES

La aplicación proveerá un método de consulta efectiva sobre las listas cautelares OFAC y ONU, la información obtenida y el Servicio Web están diseñados para formar parte de la metodología de Prevención de Lavado de Activos y Financiamiento del Terrorismo descrita en la circular externa 022 de 2007 de la Superintendencia Financiera de Colombia, por sí sola la herramienta consiste simplemente en un mecanismo de conocimiento del cliente.

Este proyecto se basará en la metodología SCRUM para el cumplimiento de sus objetivos relacionados al desarrollo de software y, aunque se explicará la estructura, bases, actores y procedimientos de la metodología, no se pretenderá profundizar en las razones o la forma como fue diseñada, tampoco se cuestionará su estructura.

La aplicación de la metodología forma parte integral del desarrollo del proyecto, las interacciones que propone, las reuniones, roles, procedimientos y documentación se llevarán a cabo de la forma como lo indica la metodología.

6.2 LÍMITES

- La información estará enfocada en cumplir las disposiciones de la Superintendencia Financiera de Colombia.

- La aplicación de consulta estará disponible únicamente para usuarios registrados.
- El registro de los usuarios de la aplicación no implica ninguna forma de pago o remuneración, puesto que uno de los objetivos de la aplicación es que esté al alcance de las PYME.

6.3 ANTECEDENTES

El Lavado de Activos es una práctica que se intenta prevenir desde 1993 cuando se expidió la Ley 67 de 1993 y del Decreto 671 de 1995, para dar cumplimiento a los acuerdos de la Convención de las Naciones Unidas de Viena (1988), desde entonces diferentes compañías han creado múltiples productos para ayudar a combatir este problema.

6.3.1 TopBls

TopBls es una aplicación de entorno Windows, cuyo propósito es descargar varias listas restrictivas, vinculantes o de conocimiento del cliente, desde fuentes públicas y privadas, interpreta estas listas y las dispone en un formato propio, que le permite integrarlas al negocio de sus clientes.

No se detallará el propósito, diseño o funcionamiento de la herramienta, sino la estructura de los datos, que tienen relevancia para la construcción de la aplicación actual.

6.3.1.1 Mae_Lista_Un (Tabla)

Esta tabla contiene los datos obtenidos desde la fuente original de las Naciones Unidas, en la misma estructura que trae el archivo original.

Gráfico 1. Estructura de la tabla mae_lista_un en SqlServer

mae_lista_un		
Nombre de columna	Tipo comprimido	Acepta valores NULL
un_code	varchar(50)	Sí
un_name	varchar(400)	Sí
un_title	varchar(400)	Sí
un_dob	varchar(400)	Sí
un_pob	varchar(400)	Sí
un_goodaka	varchar(400)	Sí
un_lowaka	varchar(400)	Sí
un_nationality	varchar(400)	Sí
un_passport	varchar(400)	Sí
un_nationalid	varchar(400)	Sí
un_addres	varchar(400)	Sí
un_listedon	varchar(400)	Sí
un_otherinfo	varchar(400)	Sí

un_code: Identificador del registro.

un_name: Nombre como aparece en el registro original, en su idioma original.

un_title: Título profesional, honorario o religioso.

un_dob: Fecha de nacimiento, incluyendo fechas alternativas de nacimiento.

un_pob: Lugar de nacimiento, incluyendo lugares alternativos de nacimiento.

un_goodaka: Alias o nombres alternativos, que son suficientes para realizar una identificación positiva.

un_lowaka: Alias o nombres alternativos, que probablemente no sean suficientes para realizar una identificación positiva.

un_nationality: Nacionalidad actual, o previa del ciudadano.

un_passport: Pasaporte o documento alternativo de viaje.

un_nationalId: Identificaciones como cédula de ciudadanía, licencia de conducción, número de seguridad social, tarjetas de crédito, etc.

un_address: Dirección de correspondencia actual o pasada, temporal o permanente.

un_listedon: Fecha en el que el comité de sanciones de Al-Qaida ubicó al individuo en la lista de sanciones de Al-Qaida y otras modificaciones realizadas al registro del individuo.

un_OtherInfo: Información adicional, importante para la identificación del individuo.

6.3.1.2 Mae_Lista_Ofac (Tabla)

Esta tabla contiene los registros de la lista OFAC – SDN *SPECIALLY DESIGNATED NATIONALS AND BLOCKED PERSONS* (Nacionales especialmente designados y personas bloqueadas), también conocida como “Lista Clinton”.

Gráfico 2. Estructura de la tabla mae_lista_ofac en Sql Server

mae_lista_ofac		
Nombre de columna	Tipo comprimido	Acepta valores NULL
ent_num	int	No
ent_alt_code	int	No
SDN_Name	varchar(200)	Sí
SDN_Type	varchar(12)	Sí
Program	varchar(20)	Sí
Title	varchar(180)	Sí
Call_Sign	varchar(8)	Sí
Vess_type	varchar(25)	Sí
Tonnage	varchar(14)	Sí
GRT	varchar(8)	Sí
Vess_flag	varchar(20)	Sí
Vess_owner	varchar(150)	Sí
Remarks	varchar(600)	Sí
ent_new	tinyint	No
ent_update	datetime	Sí
ent_note	varchar(30)	Sí

ent_num: Identificador único del registro.

ent_alt_code: Identificador único de identidad alternativa.

SDN_Name: Nombre del SDN .

SDN_Type: Tipo del SDN.

Program: Nombre del programa de sanciones.

Title: Título del Individuo.

Call_Sign: Signo de llamada del Buque.

Vess_type: Tipo del Buque.

Tonnage: Tonelaje

GRT: Tonelaje de registro bruto.

Vess_flag: Bandera

Vess_owner: Propietario

Remarks: Información adicional de importancia.

ent_new: ¿Es Nuevo el registro?

ent_update: Fecha de actualización.

ent_note: Notas.

6.3.2 DetectART

DetectART es una aplicación que posee dos versiones, una para Windows y otra Web, su propósito principal es el monitoreo de las transacciones y movimientos de las personas, y detectar posibles operaciones relacionadas con el Lavado de Dinero, contiene un módulo que permite realizar consultas de clientes en diferentes listas de control y conocimiento del cliente

(KYC por sus siglas en inglés), en el cuál es posible generar reportes de las coincidencias, y marcar clientes para seguimiento.

6.3.3 Limitaciones de las herramientas actuales

Las herramientas actuales permiten la descarga e interpretación de múltiples listas de conocimiento del cliente, pero dado el propósito de su negocio permiten que estas listas lleguen a bases de datos con un formato no abierto o que no permita que otros productos puedan tener acceso a esas listas.

6.4 GUÍA DE INVESTIGACIÓN O DESARROLLO

El desarrollo del proyecto debería llevarse a cabo utilizando las siguientes técnicas y metodologías:

- Para el control de Código fuente usar una herramienta para control de versiones (Team Foundation Server).
- Para el control de avance de SCRUM usar una herramienta para el seguimiento de proyectos (Team Foundation Server).
- Para la ejecución del proyecto usar una metodología probada y de uso común (SCRUM).
- Al finalizar el proyecto se publicará en un servidor de manera temporal, para que pueda ejecutarse durante un periodo de limitado.

6.5 METODOLOGÍA EMPLEADA

Para el desarrollo del proyecto se ha elegido la metodología SCRUM que permite desarrollar proyectos de manera rápida y organizada, a partir de la especificación de roles, reuniones y un cronograma, se ha seleccionado esta metodología debido a que este desarrollo se realizará en las instalaciones de Sidif Ltda. Y esta empresa desea implementar esta metodología en todos los proyectos siguientes.

6.5.1 Metodología SCRUM

Scrum es una metodología que consisten en aplicar una serie de buenas prácticas, como separación de roles, reuniones de equipo y revisiones, dentro de un proceso iterativo grupal, en el que se hacen entregas parciales, que se han priorizado con el objetivo de entregar resultados rápidamente.

6.5.2 UML

UML es un Lenguaje de Modelado que se utiliza para describir proceso, interacciones y acciones, y el conjunto de sus diagramas permiten describir un sistema de software preferiblemente programado basado en objetos.

6.6 HERRAMIENTAS

Para el desarrollo del proyecto se necesitan varias herramientas que ayudarán a la creación o el control de alguna de sus fases o componentes, todas las herramientas que se usarán son versiones Express que permiten un uso académico y comercial limitado sin realizar ningún tipo de pago.

6.6.1 Microsoft Visual Studio 2012 Express Edition

Visual Studio 2012 Express Edition ha evolucionado más allá de un simple Entorno de Desarrollo Integrado (IDE), para convertirse en un conjunto de herramientas que permiten llevar a cabo la tarea de desarrollo de software, que cuando se integra con otras herramientas de la familia Visual Studio como Team Foundation Server, permiten controlar todos los aspectos que el desarrollo de software implica, y su licenciamiento Express permite que se desarrollen aplicaciones de carácter educativo y comercial sin necesidad de pagar las licencias profesionales de alto costo.

Los lenguajes usados para el desarrollo de las aplicaciones son:

- Visual C#: Es un lenguaje de programación orientado a objetos, enfocado y usado especialmente para el desarrollo de Backends para Windows y Web, y FrontEnds para Windows.

- Asp.Net: Es un lenguaje creado para el diseño y desarrollo de capas de Interfaz de Usuario (UI por sus siglas en inglés), que se integra en el código Html, permitiendo la interacción de los lenguajes de cliente como Html y JavaScript con lenguajes de Servidor como C#.
- Html: Es el lenguaje natural para el desarrollo de sitios web, controla la presentación de los objetos en el explorador del cliente.
- Javascript: Es un lenguaje de programación de cliente, basado en Scripts, está orientado a objetos y controla el comportamiento de los objetos en el explorador del cliente.
- Css: Si bien no es un lenguaje en el sentido estricto, define el conjunto de reglas de apariencia aplicable a todo el sitio web.
- SQL: Lenguaje de consultas a bases de datos, permite introducir, eliminar, actualizar y consultar información dentro de una base de datos.
- XML: Es un lenguaje de marcado, que permite la interacción entre múltiples plataformas.

6.6.2 Team Foundation Server 2012 Express Edition

Team Foundation Server 2012 Express Edition, es un conjunto de herramientas que permiten administrar y controlar el código fuente desarrollado en un IDE como Visual Studio 2012, también permite que el código que se desarrolla se haga en una planificación como la de SCRUM, al ser una versión Express no requiere de pago de licencias para el uso educativo o comercial.

6.6.3 Sql Server 2008 R2 Express Edition

Es un motor de base de datos relacional, que permite almacenar datos, de manera integral y distribuida, que tiene un gran rendimiento y soporte, al ser una versión Express no requiere de pago de licencias para el uso educativo o comercial y sus limitaciones en tamaño de los datos (10gb), permiten manejar un volumen suficiente para el proyecto que se desarrolla.

7 MARCO CONCEPTUAL

7.1 INGENIERÍA DE SOFTWARE

“La Ingeniería del software es una disciplina o área de la Informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo.”¹⁸

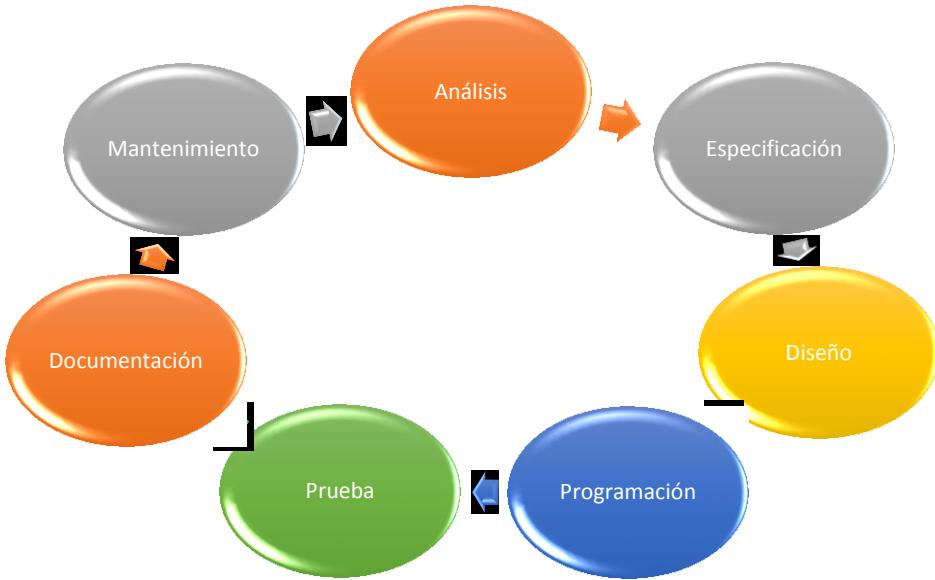
“La aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software; es decir, la aplicación de ingeniería al software.”¹⁹

Esta definición nos permite abarcar la mayoría de los proyectos de desarrollo de software de la actualidad e incluye las siguientes etapas:

¹⁸ PRESSMAN, Roger. Ingeniería del Software - Un Enfoque Práctico, 2005, Cap. 1, p. 1

¹⁹ IEEE STANDARDS ASSOCIATION. 610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology. 2002.

Gráfico 3. Ciclo de vida del desarrollo de Software según Pressman²⁰



Se dice que el proceso de desarrollo de software visto desde la perspectiva de la ingeniería de software tiene 5 etapas: Análisis, Especificación, Diseño, Programación y Prueba sin embargo existen dos adicionales, que aunque no se cuentan formalmente deben ser incluidas sin excepción en la totalidad de los proyectos de software: Documentación y Prueba, que se expresan a continuación.

7.1.1 Análisis

Es la primera etapa del ciclo de vida del software (gráfico 3), en esta etapa se debe realizar una exploración y extracción de los requerimientos, durante esta tarea el ingeniero deberá

²⁰ PRESSMAN, Roger. Op. Cit.

poseer o desarrollar habilidades para la comunicación con distintos perfiles de personas, ya que su misión consiste en entender las necesidades de cada uno de ellos, que usualmente son expresadas en sus propios términos, orientar estas necesidades con un enfoque de puntos de vista y unirlos con los de los demás para lograr una visión unificada de las necesidades, convertirlas en requerimiento y volver a iniciar el proceso, hasta que se considere que los aspectos más relevantes se han cubierto y que no se ha dejado ninguno de importancia moderada por fuera.

El analista puede encontrarse con distintos problemas, los más frecuentes son:

- El personal entrevistado no sabe lo que quiere o no sabe expresarlo
- Las necesidades están expresadas en términos que sólo resultan entendibles para la persona entrevistada.
- Los requerimientos de un rol pueden entrar en conflicto con los de otro rol o área.
- Los requerimientos pueden no ser una necesidad actual, pero provenir de una decisión administrativa.
- Pueden existir factores políticos en la especificación de los requerimientos.
- La información entregada en la etapa de análisis no fue suficiente.
- Pueden surgir nuevas necesidades con el paso del tiempo.

Para superar estos problemas el analista debe poseer o adquirir la habilidad para entender, traducir y clasificarlos, para esto existe un proceso iterativo que consiste en:

- **Comprensión del dominio:** Consiste en entender el ambiente en el cual se desarrolla el problema, límites, alcances, flujo de la información, contenido y estructura.²¹
- **Recolección de los requerimientos:** Se realiza mediante la ejecución de técnicas como entrevistas, formularios, observación, etc.²²
- **Clasificación:** Los requerimientos deben ser clasificados según distintas categorías como la relevancia para el negocio, el control, las finanzas, etc.²³
- **Solución de conflictos:** Cuando un requerimiento entra en conflicto con otro deberá realizarse un proceso de acuerdo entre las partes implicadas, para dar solución al conflicto.²⁴
- **Priorización:** Una vez determinados los requerimientos éstos deben ser ordenados por prioridad, esta prioridad determinará el enfoque del desarrollo y las entregas.²⁵

²¹ UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA, Facultad De Ingeniería, <http://yaqui.mxl.uabc.mx/>
Disponible en <http://yaqui.mxl.uabc.mx/~molguin/as/IngReq.htm>

²² Ibíd.

²³ Ibíd.

²⁴ Ibíd.

²⁵ Ibíd.

- **Validación de los requerimientos:** terminadas todas las etapas, deberá verificarse que los requerimientos sean suficientes y no existan faltantes, si se encuentran, el proceso deberá iniciarse nuevamente para los demás requerimientos.²⁶

7.1.2 Especificación de los requerimientos (ERS)

Es la segunda etapa del ciclo de vida del software (gráfico 3), Según la IEEE (Institute of Electrical and Electronics Engineers) las características que debe tener un requerimiento son²⁷:

- **Corrección:** Cada requerimiento debe solucionar una necesidad específica, debe estar delimitado y debe ser viable.²⁸
- **No ambiguo:** El requerimiento debe tener una sola interpretación, cada característica debe ser definida con términos que se encuentren en el diccionario de datos, y especificará el uso de herramientas de representación.²⁹
- **Completo:** El texto que describe el requerimiento debe ser suficiente para su comprensión, proporcionar los enlaces, contexto o explicaciones necesarias.³⁰

²⁶ Ibíd.

²⁷ IEEE STANDARDS ASSOCIATION, Op. Cit., IEEE Std. 830-1998. p. 6

²⁸ Ibíd.

²⁹ Ibíd.

³⁰ Ibíd.

- **Verifiable:** Los requerimientos deben proporcionar un mecanismo de verificación una vez que se ha terminado la aplicación, puede ser una secuencia de pasos, un informe, etc.³¹
- **Consistencia:** No debe haber conflictos entre los requerimientos, así como un requerimiento no puede estar expresado en formas distintas.³²
- **Clasificación:** Cada requerimiento debe pertenecer a una clase y tener una prioridad asignada.³³
- **Modificable:** El requerimiento debe expresarse de manera clara y ordenada, separado de otros requerimientos, de manera que su modificación tenga el menor impacto posible.³⁴
- **Explorable:** Cada requerimiento deberá contener una identificación única, hacer referencia a etapas anteriores y todos los que se derivan de este.³⁵
- **Utilizable:** Una persona que no ha tenido contacto con los requerimientos, luego del estudio de estos, debe poder realizar tareas de mantenimiento o corrección sobre un requerimiento en particular.³⁶

7.1.3 Diseño y Arquitectura

Es la tercera etapa del ciclo de vida del software (gráfico 3), esta etapa se refiere a la estructuración de los componentes del sistema, sus interacciones y cómo éstos resuelven los

³¹ Ibíd. p. 7

³² Ibíd.

³³ Ibíd.

³⁴ Ibíd.

³⁵ Ibíd.

³⁶ Ibíd.

requerimientos especificados anteriormente, no intentará entrar en detalle pero si debe crear unas guías claras que se usarán para el desarrollo del sistema.

Objetivos de la etapa de diseño:

- Identificar y proporcionar una arquitectura adecuada para la construcción del software.
- Identificar los subsistemas.
- Establecer las relaciones entre los subsistemas y componentes.
- Determinar las interfaces de comunicación.
- Especificar la funcionalidad de los componentes.

Etapas del proceso de diseño:

- **Diseño de la arquitectura:** Determina los patrones de diseño que se deben utilizar para satisfacer los requerimientos.³⁷
- **Diseño de datos:** Determina las estructuras de datos requeridas, y las operaciones que se realizarán con cada una de ellas.³⁸
- **Diseño de componentes:** Se diseñan los algoritmos, y la secuencia de los procedimientos, usando las estructuras diseñadas anteriormente.³⁹

^{37 37} PRESSMAN, Roger. Ingeniería del Software - Un Enfoque Práctico, 2006, Cap. 13-2

³⁸ Ibíd.

³⁹ Ibíd.

- **Diseño de la interfaz:** Establece las interacciones entre las personas y el sistema, hará uso de técnicas para simplificación de las tareas, permisos y ergonomía.⁴⁰

7.1.4 Programación

Es la cuarta etapa del ciclo de vida del software (gráfico 3), es la etapa en donde se construye el software a partir de las especificaciones de requerimientos, se hace de acuerdo al diseño planteado anteriormente y constituye una de las fases más tangibles del proceso, durante este periodo, los profesionales expertos en programación deberán hacer uso de distintas herramientas, técnicas y conocimientos para convertir los diseños y estructuras en un sistema que solucione cada uno de los requerimientos especificados.

7.1.5 Prueba (Testing)

Es la quinta etapa del ciclo de vida del software (gráfico 3), Consiste en realizar una serie de ejercicios para corroborar que el software satisface no sólo los requerimientos especificados, sino que cumple con estándares de calidad, tolerancia a errores, legibilidad entre otros.

Existen varios tipos de pruebas entre los cuales se encuentran:

⁴⁰ Ibíd.

- **Pruebas unitarias:** Se realizan pruebas manuales o automatizadas que permitan determinar que una fracción de software cumple con su objetivo, estas pruebas deben realizarse para la cantidad de fragmentos de código que sea posible.
- **Pruebas de integración:** Es una prueba básica en la cual se demuestra que todos los componentes del sistema pueden trabajar juntos sin ninguna falla evidente.
- **Pruebas de sistema:** Su objetivo es verificar que el sistema puede ejecutarse y funcionar con los requisitos del ambiente, como software y hardware.
- **Pruebas de rendimiento:** Permite analizar que el sistema efectúa todas las operaciones con un uso de los recursos apropiado y no desproporcionado, dentro de unos niveles subjetivos y objetivos.
- **Pruebas de aceptación** Es la última prueba, en donde el usuario evalúa cada una de las características del software, y determina si cumplen o no con lo esperado.

7.1.6 Documentación

Es la sexta etapa del ciclo de vida del software (gráfico 3), la documentación de un sistema es una fase transversal a todo el proceso, se inicia desde la primera fase, y termina hasta la finalización de las pruebas, sin embargo, también debe realizarse en la etapa de mantenimiento en caso que se presente.

Los documentos mínimos que un sistema debe entregar son:

- **Documento de especificaciones (ERS):** Consiste en describir todas las funcionalidades que debe contener el sistema, sobre el contenido de este documento deben estar de acuerdo todas las partes implicadas, ej. Contratista y cliente
- **Documentos de Diseño (Manual técnico):** Puede referirse a un documento genérico que explique la forma como el software se va a crear, también puede incluir una serie de diagramas, comúnmente especificados en notación UML.
- **Documentación del código fuente:** En la fase de programación, el desarrollador deberá dejar líneas de comentarios que expliquen cada una de las funciones que realiza una porción de código, y si el problema lo requiere por su complejidad, también deberá documentar línea por línea, para dejar claro el propósito para nuevos desarrolladores, o para él mismo en el futuro, esto facilitará la depuración y corrección de errores.
- **Manual de usuario (Ayuda):** Este documento suele ser bastante extenso, ya que tiene como propósito enseñar el uso del sistema, y cada una de las opciones que éste presenta, deberá ser claro y escrito en lenguaje natural.

7.1.7 Mantenimiento

Es la séptima etapa del ciclo de vida del software (gráfico 3), Es una etapa, aunque se considera opcional, suele estar presente en todos los proyectos de software, consiste en

realizar modificaciones al software, ya sea por correcciones de errores, perfeccionamiento o adaptaciones.

Los tipos de mantenimiento más comúnmente usados son:

- **Mantenimiento correctivo:** Una vez que se ha identificado un problema con el software, que puede ser una falla, o de rendimiento, interpretación, etc. Debe realizarse un procedimiento para solucionar este inconveniente, esto implica retomar parte de las etapas anteriores y realizar los cambios pertinentes.
- **Mantenimiento preventivo:** No se realizan cambios sobre la funcionalidad interna del software, se puede realizar para mejorar la calidad del código, o para mejorar la reusabilidad.
- **Mantenimiento adaptativo:** Se realiza cuando existen cambios en el entorno que impiden su funcionamiento, como especificaciones de hardware, cambios en el tipo de base de datos, distribución de procesos, etc.
- **Mantenimiento perfectivo:** Tiene como propósito mejorar la aplicación, ya sea internamente, por rendimiento o utilización de recursos, o externamente, en implementación de nuevas funciones.

7.2 METODOLOGÍA DE DESARROLLO ÁGIL (SCRUM)

SCRUM es una metodología de desarrollo de proyectos ágil, que permite entregar resultados rápidamente y aumentar la productividad sin descuidar la calidad, por medio de periodos cortos e iterativos.

Se caracteriza por ser una metodología ligera, fácil de aprender pero compleja de dominar⁴¹, para su funcionamiento utiliza distintos roles que se pueden distribuir en un equipo, que los interrelaciona por medio de eventos, reglas y artefactos.

7.2.1 Selección de la Metodología

Se ha seleccionado SCRUM entre otras metodologías ágiles, basado en varios factores de carácter técnico y productivo, las metodologías analizadas son AGILE, SCRUM y XP (Extreme Programming), la siguiente tabla comparativa nos muestra que de las tres metodologías analizadas Extreme Programming es la que más riesgo y complejidad representa, y Scrum, es un Marco de Trabajo que simplifica la metodología Agile y requiere menos roles.

⁴¹ SCHWABER, Ken y SUTHERLAND Jeff, La Guía de Scrum. 2013, p. 3, www.scrum.org Disponible en <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide-ES.pdf>

Tabla 1. Comparación entre las Metodologías

AGILE	SCRUM	XP
Enfocado en la Administración	Enfocado en la Administración	Enfocado en los desarrolladores
Permite cambios en el producto	Permite cambios en el producto	Permite cambios en el producto
Produce Software de Calidad	Produce Software de Calidad	Produce Software de Calidad aceptable
Reduce el riesgo.	Reduce el riesgo.	El riesgo depende de la administración del proyecto
Reduce el costo.	Reduce el costo.	Reduce el costo.
Fácil de entender	Fácil de entender	Moderadamente sencilla de entender
Difícil de Implementar	Difícil de Implementar	Difícil de Implementar
Entregas cortas	Entregas cortas	Entregas cortas
Requiere múltiples Roles	Requiere pocos Roles	Requiere múltiples Roles
Es una Metodología completa	Simplifica Agile	Es una Metodología completa

También existen otros factores para la elección de Scrum:

Laboral: Ya que el proyecto se desarrolla con el apoyo de la empresa Sidif Ltda. Se tuvo en cuenta que esta empresa está en proceso de implementación, lo que permite aportar al conocimiento no sólo del estudiante sino de la compañía.

Oportunidad: Teniendo en cuenta el ámbito laboral explicado anteriormente, se presenta la oportunidad de tener acceso a la literatura relacionada, incrementando la calidad de la información con referencias seleccionadas y material de ejemplo.

7.2.2 Casos de Éxito de SCRUM

SPOTIFY: Spotify es un servicio de reproducción de música vía Streaming, para su creación la empresa tuvo que enfrentar varios problemas relacionados con el crecimiento de la compañía, hasta que completó un conjunto de 30 equipos de desarrollo en 3 ciudades diferentes⁴², en este proceso la segmentación de los equipos y los roles provistos por Scrum los condujeron a una solución compleja, que los ha posicionado como uno de los reproductores de música vía streaming favoritos de los usuarios en internet.

AT Sistemas: AT Sistemas es una empresa Española creada a partir de la idea de abrir una Fábrica de Software, en este caso ya que la empresa no existía fue necesario implementar todos los aspectos necesarios, y utilizaron Scrum no sólo para el desarrollo de aplicaciones, sino para la estructuración de la misma compañía, enfrentaron problemas en la resolución de Bugs, negocios cambiantes y mantenimiento correctivos entre otros⁴³.

Actualmente existe una lista que contiene las principales compañías que usan SCRUM ⁴⁴, entre las cuales se encuentran gigantes del Software como Adoble Systems, Google, IBM y más de 200 compañías.

⁴² KNIBERG, Henrik e IVARSSON, Anders , Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds, 2012, disponible en <https://dl.dropboxusercontent.com/u/1018963/Articles/SpotifyScaling.pdf>

⁴³ FERNANDEZ, Antonio y AMODEO, Enrique. Ser Ágil en España: Un caso real con equipos de trabajo en remoto. 2010, disponible en <http://es.slideshare.net/eamodeorubio/ser-gil-en-espana-un-caso-real-con-equipos-de-trabajo-en-remoto>

⁴⁴ Scrum Community, Firms Using Scrum, 2014, disponible en <https://docs.google.com/spreadsheets/ccc?key=0AgfBeuoRfUzNdDIMNG82SIhmUVRhOEk0REtrdmthNWc#gi> d=5

7.2.3 Bases de la teoría

SCRUM se fundamenta en tres aspectos principales:

- **Transparencia:** Todos los miembros del equipo manejan un lenguaje común, y los estados del proceso significan lo mismo para todos, también todos los aspectos relevantes están disponibles para todos los que han sido designados como responsables⁴⁵.
- **Inspección:** Los artefactos deben ser inspeccionados periódicamente con el objetivo de detectar posibles variaciones en el proceso, estas inspecciones deben tener una frecuencia que permita que no se desvíe mucho el trayecto al objetivo, pero que tampoco interfiera con el trabajo⁴⁶.
- **Adaptación:** Cuando se detecta una desviación que afecta el resultado del proyecto, éste deberá ajustarse hasta que se retorne el rumbo que garantice que se alcanzará el objetivo.⁴⁷

⁴⁵ SCHWABER, Ken y SUTHERLAND Jeff, La Guía de Scrum. 2013, Op. CIT

⁴⁶ Ibíd.

⁴⁷ Ibíd.

7.2.4 Equipo SCRUM (*SCRUM TEAM*)

El equipo de trabajo necesario para llevar un proyecto acabo utilizando la metodología SCRUM se divide en los siguientes roles:

- **Dueño del producto (*Product Owner*):** Es el responsable del proyecto, su misión es la de comunicarse con el equipo y explicar los pasos a seguir, ordena las tareas del producto (*BackLog*) y las asigna a los individuos maximizando la productividad, también se asegura que cada miembro tenga acceso al plan de trabajo global y que cada individuo conozca lo que tiene que hacer.⁴⁸

El dueño del producto es una persona designada y su autoridad no debe ser cuestionada, si existe un grupo o comité que deba tomar decisiones, éstas serán comunicadas al equipo exclusivamente por el dueño del producto.

- **Equipo de desarrollo (*Development Team*):** Son los encargados de desarrollar el producto, se encargan de cumplir con los objetivos hasta tener una versión entregable, no posee sub equipos o divisiones y todos se complementan para cumplir los objetivos.

⁴⁸ Ibíd. p. 5

La responsabilidad del cumplimiento de los objetivos es del equipo completo, no existen responsabilidades individuales y deben organizarse internamente para elaborar cada uno de los elementos de la lista de tareas.

El equipo de desarrollo no deberá contar con menos de tres o más de nueve integrantes, superar cualquiera de estos dos límites compromete la productividad y genera limitaciones respecto a la coordinación de las tareas, comunicación interna y dirección.⁴⁹

- **Maestro SCRUM (*SCRUM Master*):** Es el encargado de verificar que se cumplan las reglas propuestas por la metodología, debe apoyar al equipo de desarrollo eliminando obstáculos en el avance, eliminar las dudas respecto al uso de SCRUM y ayudar al equipo a autogestionarse y autoorganizarse, también debe servir al dueño del producto ayudándole a encontrar las técnicas apropiadas para la creación y distribución de las tareas, entender y optimizar las estrategias actuales para maximizar el valor del equipo.⁵⁰

Se encargará de promover SCRUM y evitar que los miembros que tengan problemas con la adopción de la metodología generen una sensación incómoda o de rechazo a esta, deberá motivarlos y ayudarlos a entenderla.

⁴⁹ Ibid. p. 6

⁵⁰ Ibid.

7.2.5 Eventos

Los eventos hacen referencia a un periodo de tiempo que se emplea en el desarrollo de una actividad, los eventos deben tener una duración máxima y no debe modificarse su duración de ninguna manera.

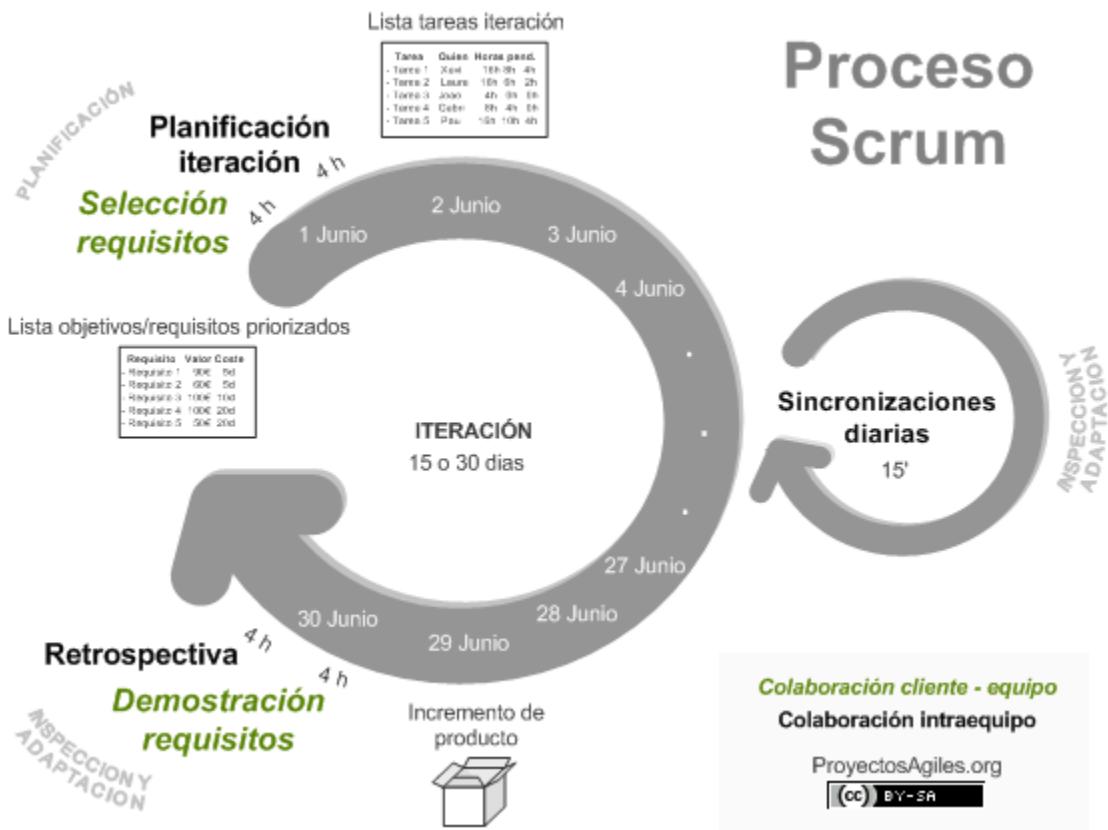
SPRINT: Es un periodo de tiempo aproximadamente entre dos a cuatro semanas en el cual se crea una versión del producto, funcional y entregable, que cumpla con una sección de los requerimientos, la duración de los Sprints debe ser preferiblemente la misma durante todo el desarrollo y deben ser consecutivos, tan pronto termina uno, debe iniciar el otro.⁵¹

Cada Sprint debe planearse, deben establecerse unos objetivos, un plan suficientemente flexible para aceptar cambios y qué producto se debe obtener al finalizarlo, los cambios introducidos dentro de un Sprint no deben afectar el objetivo final, para eso antes de iniciar un Sprint los objetivos deben clarificarse y en dado caso negociarse entre el equipo de desarrollo y el dueño del producto.

Un Sprint puede llegar a ser cancelado si el objetivo del Sprint cambia lo suficiente de manera que lo haga obsoleto o innecesario o si las condiciones comerciales cambian, debe analizarse bien la cancelación de un Sprint ya que esto resulta traumático para el equipo de desarrollo, y el dueño del producto, ya que se debe realizar una nueva planeación, ver los objetivos cumplidos y generar un nuevo plan, e iniciar un siguiente Sprint.

⁵¹ Ibíd. p. 8

Gráfico 4. Ciclo del proceso Scrum



Tomado de <http://www.proyectosagiles.org/que-es-scrum>

- **Reunión de planificación (Sprint Planning Meeting):** En esta reunión se planea el trabajo a desarrollar durante el Sprint, para un Sprint de un mes esta reunión puede tener una duración de hasta 8 horas, para periodos más cortos, pueden realizarse reuniones más cortas.
⁵²En esta reunión se decidirá qué características se van a entregar y qué es necesario para llevar a cabo ese trabajo.

⁵² Ibíd.

- **Scrum Diario (Daily Scrum):** Es una reunión de 15 minutos en la cual se define un plan para ese día, se exponen los problemas que se podrían presentar y se plantea una solución que permita cumplir el objetivo de estas 24 horas.
- **Revisión del Sprint (Sprint Review):** Una vez terminado el Sprint se realiza una revisión, para determinar qué ítems de la Lista de Tareas se han completado y cuáles no, se analizan los ítems que podrían realizarse en el siguiente Sprint y sirve como reunión de avance del proyecto.
- **Retrospectiva del Sprint (Sprint Retrospective):** Esta reunión de corta duración, alrededor de 3 horas para un Sprint de un mes y menos tiempo según la duración del Sprint, sirve para analizar qué cosas se pueden mejorar respecto al proceso, al equipo y demás aspectos que puedan mejorar la realización del siguiente Sprint.

7.2.6 Artefactos

SCRUM provee varios artefactos que sirven para proporcionar transparencia, entendimiento y guía sobre el funcionamiento del desarrollo en todas sus fases, los elementos más importantes son:

- **Lista del producto (Product BackLog):** Contiene una lista de todos los requerimientos que se conocen y entienden desde el principio, por esta razón está

evolucionando constantemente y se completa con el tiempo, puede contener no sólo características del producto, sino también mejoras, cambios, y según lo determine el Dueño del Producto puede tener un orden, un valor y una estimación.⁵³

- **Lista de pendientes del producto (Sprint BackLog):** Es el subconjunto de elementos de la Lista del Producto que se han seleccionado para un Sprint, estos elementos serán Terminados al finalizar el Sprint y se revisarán en el Scrum Diario.⁵⁴
- **Incremento:** Es la suma de todas las características de la Lista del Producto que están Terminadas, es decir, que están en condiciones de ser utilizadas, la determinación de liberar o no el incremento es del Dueño del Producto.⁵⁵
- **Definición de Terminado:** Todo el equipo debe manejar una misma definición de lo que significa “Terminar” un elemento de la Lista del Producto o un Incremento, de manera que al finalizar un Sprint el Incremento contenga una serie de ítems de la Lista del Producto “Terminados” y listos para implementar en producción.⁵⁶

⁵³ Ibíd. p. 9

⁵⁴ Ibíd.

⁵⁵ Ibíd.

⁵⁶ Ibíd.

7.3 UML

UML - Lenguaje Unificado de Modelado por sus siglas en inglés (Unified Modeling Language) proporciona una serie de diagramas que permiten describir todas las partes de un sistema, la interacción con los usuarios y otros sistemas⁵⁷.

Dentro de la metodología SCRUM no se expresa que deba usarse UML, sin embargo como lenguaje de modelamiento moderno, es ampliamente usado dentro de los proyectos de desarrollo de software orientado a objetos, a continuación se mencionan los diagramas básicos de UML usados en este proyecto.

7.3.1 Diagrama de Casos de Uso

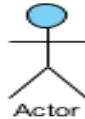
Representa la forma como un Actor (Persona o sistema), interactúa con una función específica del sistema, se divide en 3 elementos:

- **Actor:** Hace referencia a un rol que interactúa con el sistema, aunque no representa una persona en particular, se presenta como una figura muy simple con forma de persona.⁵⁸

⁵⁷ OBJECT MANAGEMENT GROUP. Unified Modeling Language. 2014. uml.org Disponible en <http://uml.org>

⁵⁸ HENSGEN, Paul. Manual de Umbrello UML Modeler, 2014, p. 10

Gráfico 5. Actor UML



- **Caso de Uso:** Se refiere a la tarea específica que se realiza por la petición de un Actor, o por la interacción con otro Caso de Uso.⁵⁹

Gráfico 6. Casos de Uso UML



- **Relaciones:** Indica el tipo de interacción que hay entre el Actor y el Caso de Uso o entre Casos de Uso, puede ser de los siguientes tipos:
 - a) **Asociación:** Indica que un Actor o Caso de Uso invoca a otro Caso de Uso, se representa con una línea recta⁶⁰.

Gráfico 7. Relación de Asociación UML



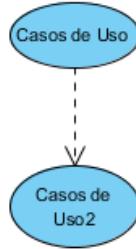
- b) **Dependencia:** Indica que una clase depende de otra⁶¹.

⁵⁹ Ibíd. p. 9

⁶⁰ Ibíd. p. 12

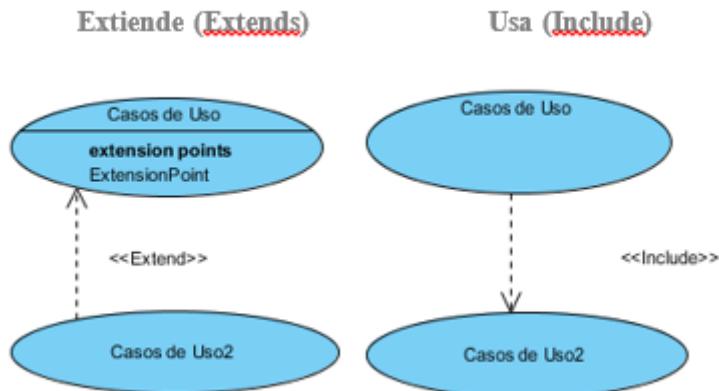
⁶¹ Ibíd.

Gráfico 8. Relación de Dependencia UML



- c) **Generalización:** Indica que un caso de uso “Usa” o “Extiende” a otro caso de uso, se dice que se “Usa” cuando es similar a otro Caso de Uso, y que “Extiende” cuando posee características similares en más de un Caso de Uso⁶²

Gráfico 9. Relación de Generalización: extensión y uso.

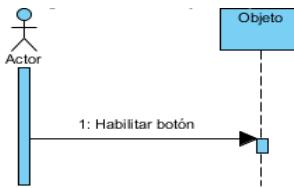


⁶² Ibíd.

7.3.2 Diagramas de Secuencia

Los Diagramas de Secuencia se encargan de describir la forma como los objetos se comunican entre sí, por medio de mensajes, pone énfasis en el orden en el que suceden las operaciones y en el momento en que sucede el envío de mensajes, estos pueden ser sincrónicos o asíncronos.

Gráfico 10. Diagrama de Secuencia



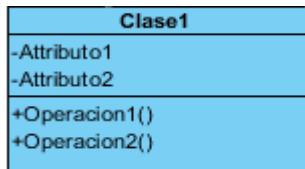
7.3.3 Diagrama de Clases

Los diagramas de clases muestran relaciones que existen entre las clases, definen sus métodos y atributos, pero no definen cómo se invocan comunican entre ellas, está conformado por:

- **Clases:** Una clase es la definición de la estructura de un objeto, una clase tiene nombre, atributos y operaciones, y se representan dentro de un rectángulo, de la siguiente forma.⁶³

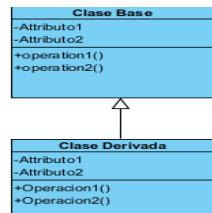
⁶³ Ibíd. p. 10

Gráfico 11. Diagrama de Clases: Clase



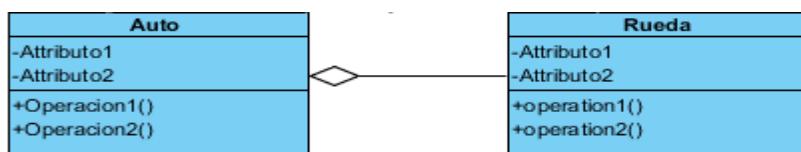
- **Generalización:** Representa el concepto de herencia de la Programación Orientada a Objetos, en donde una clase derivada “hereda” o recibe todos los atributos y operaciones de la clase base.⁶⁴

Gráfico 12. Diagrama de Clases: Relación de Generalización



- **Asociación:** Representa la relación entre dos clases, indica si la comunicación es en un sentido o en los dos, es decir, si una clase recibe mensajes de otra o si ambas pueden recibir mensajes entre sí, éstas asociaciones pueden ser de acumulación o de composición.⁶⁵
- a) **Acumulación:** Indica que una de las dos clases relacionadas actúa como completa, y se compone de varias clases de la otra, pero cada una tiene un estado Diferente

Gráfico 13. Diagrama de Clases: Relación de Acumulación

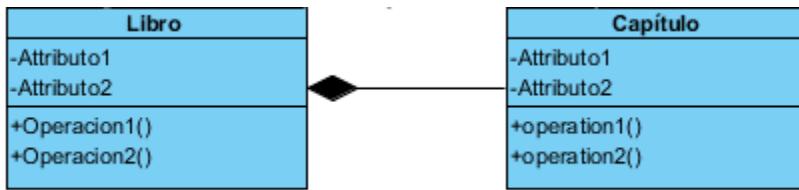


⁶⁴ Ibíd. p. 12

⁶⁵ Ibíd.

b) Composición: Muestran las relaciones entre dos clases, de la misma forma que la acumulación, pero con la diferencia que ninguna de las dos clases puede subsistir por sí sola, ya que están fuertemente acopladas⁶⁶.

Gráfico 14. Diagrama de Clases: Relación de Composición



c) Otros componentes: Se refiere a otros elementos que pueden ser graficados ya que se trata de estructuras usadas en los lenguajes de programación, los más comunes son:

- Interfaces
- Tipos de Datos
- Enumeraciones
- Paquetes

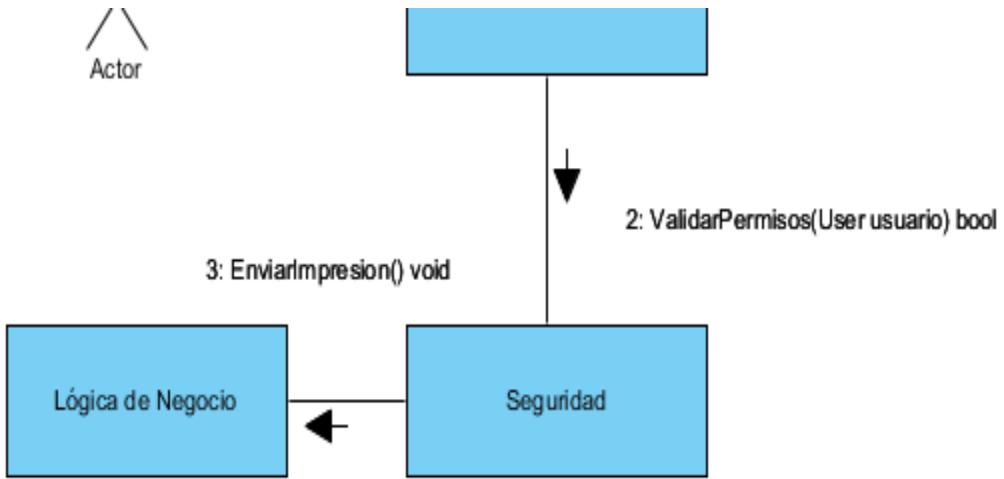
7.3.4 Diagramas de Colaboración

Similar a los Diagramas de Secuencia, se diferencian en que al mostrar la relación y el envío de mensajes entre dos objetos muestran la relación en el tiempo en que se dan estos mensajes.⁶⁷

⁶⁶ Ibíd. p. 13

⁶⁷ Ibíd. p. 15

Gráfico 15. Diagrama de Colaboración

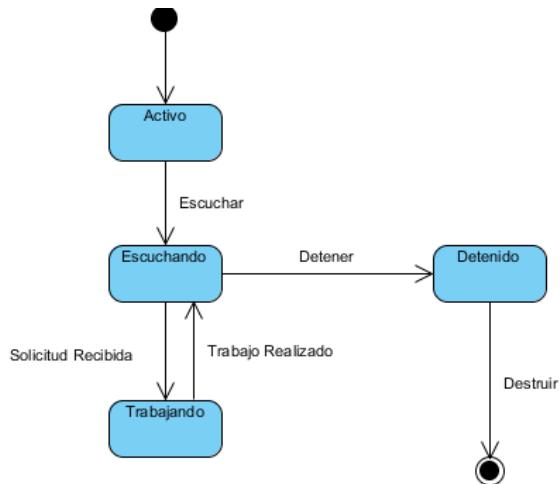


7.3.5 Diagramas de Estado

Los Diagramas de Estado representan los posibles estados de un objeto a través del tiempo, este conjunto de estados debe ser limitado y especificado, hay dos tipos de eventos especiales, Inicio y Fin, ya que ningún evento puede regresar al objeto a su estado y ningún evento puede sacarlo de su estado final, no todos los cambios en los atributos deben estar expresados en el diagrama de estados, sólo aquellos que al cambiar modifican fundamentalmente el comportamiento del objeto.⁶⁸

⁶⁸ Ibíd. p. 16

Gráfico 16. Diagrama de Estados

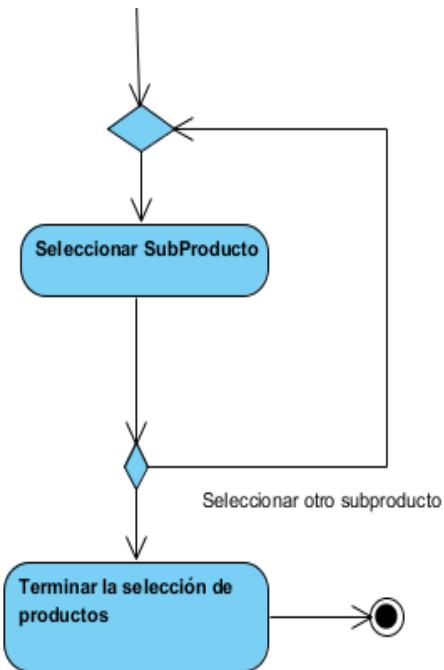


7.3.6 Diagramas de Actividad

Son similares a los diagramas de flujo, su intención es mostrar un proceso como un conjunto de actividades en una secuencia, estas secuencias las pueden llevar a cabo usuarios, o componentes de software, las acciones pueden atravesar puntos de decisión y retornar a un punto previo, formando un ciclo o continuar hasta finalizar su propósito.⁶⁹

⁶⁹ Ibíd. p. 17

Gráfico 17. Diagrama de Actividad

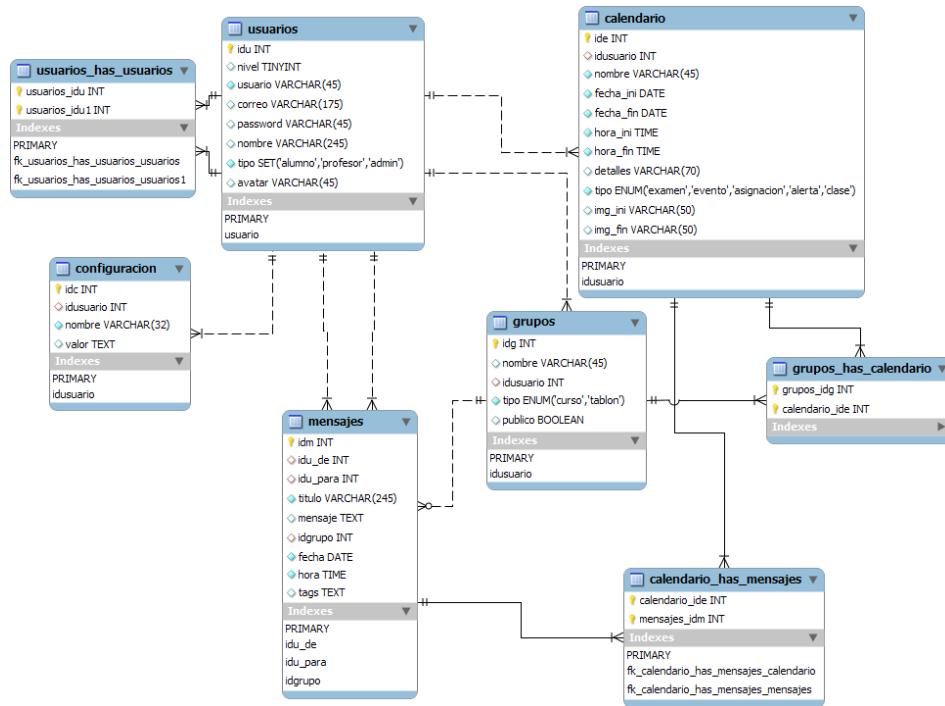


7.3.7 Diagramas de Relación de Entidad

Los diagramas de Relación de Entidad proporcionan una visión de la estructura de los elementos de la base de datos, como clases, vistas, procedimientos almacenados entre otros, que puedan ser representados en forma de entidades, se parecen a los diagramas de clases, pero muestran las relaciones entre las entidades.⁷⁰

⁷⁰ Ibíd. p. 18

Gráfico 18. Diagrama Entidad-Relación



Tomado de http://ora-flashes.blogspot.com/2012/01/modelos-de-bbdd-ii-modelo-entidad_16.html

7.4 ISO/IEC 9126

ISO 9126 es el estándar más utilizado para la evaluación de la calidad del software, establece la definición de calidad como “La totalidad de las características de una entidad que influyen en su aptitud para satisfacer las necesidades establecidas e implícitas”⁷¹, este estándar divide la calidad del software en 6 características

⁷¹ ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models 2011.

7.4.1 Funcionalidad

“La capacidad del producto software para proporcionar funciones declaradas e implícitas cuando se usa bajo condiciones especificadas”⁷² dentro de esta categoría se puede encontrar subcategorías más detalladas como:

- **Adecuación:** “La capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuarios especificados”⁷³
- **Exactitud:** “La capacidad del producto software para proporcionar los resultados o efectos correctos o acordados con el grado necesario de precisión”⁷⁴
- **Interoperabilidad:** “La capacidad del producto software para interactuar con uno o más sistemas especificados”⁷⁵
- **Seguridad de Acceso:** “La capacidad del producto software para proteger información y datos de manera que las personas o sistemas no autorizados no puedan leerlos o modificarlos, al tiempo que no se deniega el acceso a las personas o sistemas autorizados”⁷⁶

⁷² Ibíd.

⁷³ Ibíd.

⁷⁴ Ibíd.

⁷⁵ Ibíd.

⁷⁶ Ibíd.

- **Cumplimiento funcional:** “La capacidad del producto software para adherirse a normas, convenciones o legislación y prescripciones similares relacionadas con la funcionalidad”⁷⁷

7.4.2 Fiabilidad

“La capacidad del producto software para mantener un nivel especificado de prestaciones cuando se usa bajo condiciones especificadas”⁷⁸ dentro de esta categoría se puede encontrar subcategorías más detalladas como:

- **Madurez:** “La capacidad del producto software para evitar fallas como resultado de fallas en el software”⁷⁹
- **Tolerancia a Fallos:** “La capacidad del producto software para mantener un nivel especificado de prestaciones en caso de fallos software o de infringir sus interfaces especificadas”⁸⁰
- **Capacidad de Recuperación:** “La capacidad del producto software para reestablecer un nivel de prestaciones especificado y de recuperar los datos directamente afectados en caso de fallo”⁸¹

⁷⁷ Ibíd.

⁷⁸ Ibíd.

⁷⁹ Ibíd.

⁸⁰ Ibíd.

⁸¹ Ibíd.

- **Cumplimiento de la Fiabilidad:** “La capacidad del producto software para adherirse a normas, convenciones o legislación relacionadas con la fiabilidad”⁸²

7.4.3 Usabilidad

“La capacidad del producto software para ser entendido, aprendido, usado y ser atractivo para el usuario, cuando se usa bajo condiciones especificadas”⁸³ dentro de esta categoría se puede encontrar subcategorías más detalladas como:

- **Capacidad para ser entendido:** “La capacidad del producto software que permite al usuario entender si el software es adecuado y como puede ser usado para unas tareas o condiciones de uso particular”⁸⁴
- **Capacidad para ser aprendido:** “La capacidad del producto software que permite al usuario aprender sobre su aplicación”⁸⁵
- **Capacidad para ser operado:** “La capacidad del producto software que permite al usuario operarlo y controlarlo”⁸⁶
- **Capacidad de atracción:** “La capacidad del producto software para ser atractivo al usuario”⁸⁷

⁸² Ibíd.

⁸³ Ibíd.

⁸⁴ Ibíd.

⁸⁵ Ibíd.

⁸⁶ Ibíd.

⁸⁷ Ibíd.

- **Cumplimiento de la Usabilidad:** “La capacidad del producto software para adherirse a normas, convenciones, guías de estilo o legislación relacionadas con la usabilidad”⁸⁸

7.4.4 Eficiencia

“La capacidad del producto software para proporcionar prestaciones apropiadas, relativas a la cantidad de recursos usados, bajo condiciones determinadas”⁸⁹ dentro de esta categoría se puede encontrar subcategorías más detalladas como:

- **Comportamiento Temporal:** “La capacidad del producto software para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados bajo condiciones determinadas”⁹⁰
- **Utilización de Recursos:** “La capacidad del producto software para usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función bajo condiciones determinadas”⁹¹
- **Cumplimiento de la Eficiencia:** “La capacidad del producto software para adherirse a normas o convenciones relacionadas con la eficiencia”⁹²

⁸⁸ Ibíd.

⁸⁹ Ibíd.

⁹⁰ Ibíd.

⁹¹ Ibíd.

⁹² Ibíd.

7.4.5 Mantenibilidad

“La capacidad del producto software para evitar efectos inesperados debidos a modificaciones del software” dentro de esta categoría se puede encontrar subcategorías más detalladas como:

- **Capacidad para ser probado:** “La capacidad del producto software que permite que el software modificado sea validado”⁹³
- **Cumplimiento de la Mantenibilidad:** “La capacidad del producto software para adherirse a normas o convenciones relacionadas con la mantenibilidad”⁹⁴

7.4.6 Portabilidad

“La capacidad del producto software para ser transferido de un entorno a otro”⁹⁵ dentro de esta categoría se puede encontrar subcategorías más detalladas como:

- **Adaptabilidad:** “La capacidad del producto software para ser adaptado a diferentes entornos especificados, sin aplicar acciones o mecanismos distintos de aquellos proporcionados para ese propósito por el propio software considerado”⁹⁶
- **Instalabilidad:** “La capacidad del producto software para instalado en un entorno especificado”⁹⁷

⁹³ Ibíd.

⁹⁴ Ibíd.

⁹⁵ Ibíd.

⁹⁶ Ibíd.

⁹⁷ Ibíd.

- **Coexistencia:** “La capacidad del producto software para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes”⁹⁸
- **Capacidad para Reemplazar:** “La capacidad del producto software para ser usado en lugar de otro producto software, para el mismo propósito, en el mismo entorno”⁹⁹
- **Cumplimiento de la Portabilidad:** “La capacidad del producto software para adherirse a normas o convenciones relacionadas con la portabilidad”¹⁰⁰

⁹⁸ Ibíd.

⁹⁹ Ibíd.

¹⁰⁰ Ibíd.

8. DESARROLLO DEL PROYECTO

En este capítulo se llevará a cabo el contenido del proyecto, se mostrará desde el tipo de investigación, hasta el ciclo de desarrollo, se llevará a cabo el proceso de Análisis, diseño, programación, documentación y pruebas al sistema que se va a construir, este proceso se realizará siguiendo los lineamientos de la metodología de desarrollo de procesos ágiles SCRUM, que permite por medio de una serie de reuniones mayormente cortas realizar entregas funcionales llamadas incrementos.

Al inicio del proceso se definen todos los elementos que harán parte de la lista del producto (Product Backlog), durante cada Sprint (Iteración) se realizará una reunión en donde se decidirá cuáles de esos elementos se van a desarrollar durante el resto de la iteración (Sprint Backlog), para esta tarea se utilizarán herramientas de desarrollo como Visual Studio 2012 Express, Sql Server 2008 R2 Express y Team Foundation Server 2012 Express.

8.1 TIPO DE INVESTIGACIÓN

Investigación Descriptiva: “El objetivo de la investigación descriptiva consiste en llegar a conocer las situaciones, costumbres y actitudes predominantes a través de la descripción exacta de las actividades, objetos, procesos y personas.”¹⁰¹.

¹⁰¹ VAN DALEN, Deobold y MEYER, William. Manual de técnica de la investigación educacional, Paidos 1981

8.2 LÍNEA DE INVESTIGACIÓN

La línea de investigación usada en este proyecto y que se apega a las normas de desarrollo de proyectos de grado de Ingeniería de Sistemas es la “Línea de Ingeniería de Software”¹⁰².

Se creará un producto de software desarrollado bajo la metodología SCRUM, mencionada con anterioridad, con el propósito de “*Desarrollar experiencias de orden formativo y disciplinar en el campo de la investigación, con base a la construcción de software de forma sistemática y estructurada de acuerdo a los principios propios de la ingeniería de software.*”.

8.3 PRESUPUESTO

Tabla 2. Presupuesto

	TAREAS DEL PROYECTO	HORAS MANO OBRA	COSTO MANO OBRA (\$)	COSTO MATERIAL (\$)	COSTO VIAJES (\$)	OTROS COSTOS (\$)	TOTAL POR TAREA
DISEÑO DEL PROYECTO	Desarrollar especificaciones funcionales	56,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Desarrollar arquitectura del sistema	8,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Desarrollar especificaciones de diseño preliminares	8,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Desarrollar especificaciones de diseño detalladas	24,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Subtotal	96,0	\$ -	\$ -	\$ -	\$ -	\$ -
DESARROLLO DEL PROYECTO	Desarrollar componentes	192,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Obtener software	0,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Obtener hardware	0,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Subtotal	192,0	\$ -	\$ -	\$ -	\$ -	\$ -

¹⁰² UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA. Guía Para La Presentación Del Trabajo De Grado. p. 6

ENTREGA DEL PROYECTO	Instalar el sistema	3,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Entrenar a los clientes	1,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Realizar prueba de aceptación	8,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Realizar revisión posterior al proyecto	12,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Proporcionar soporte técnico bajo garantía	4,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Subtotal	28,0	\$ -	\$ -	\$ -	\$ -	\$ -
ADMINISTRACIÓN DEL PROYECTO	Reuniones/informes de progreso con el cliente	32,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Reuniones/informes de estado interno	8,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Interfaz con proveedor externo	0,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Interfaz con otros departamentos internos	0,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Administración de la configuración	0,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Control de calidad	40,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Administración global del proyecto	10,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Subtotal	90,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Subtotales	406,0	\$ -	\$ -	\$ -	\$ -	\$ -
	Total (programado)	406,0	\$ -	\$ -	\$ -	\$ -	\$ -

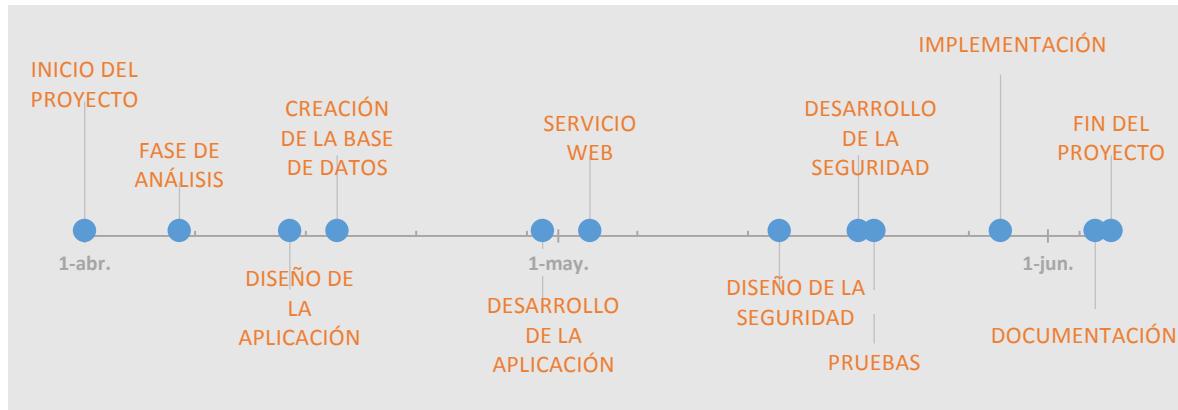
8.4 CRONOGRAMA

Tabla 3. Cronograma

DETALLES DEL PROYECTO		
FECHA INICIO	FECHA FIN	HITO
01/04/2014	01/04/2014	Inicio del proyecto
01/04/2014	07/04/2014	Fase de Análisis
08/04/2014	14/04/2014	Diseño de la Aplicación
15/04/2014	17/04/2014	Creación de la Base de datos
18/04/2014	30/04/2014	Desarrollo de la Aplicación
01/05/2014	03/05/2014	Servicio Web
04/05/2014	15/05/2014	Diseño de la Seguridad
16/05/2014	20/05/2014	Desarrollo de la Seguridad
21/05/2014	21/05/2014	Pruebas
22/05/2014	29/05/2014	Implementación
30/05/2014	04/06/2014	Documentación
05/06/2014	05/06/2014	Fin del proyecto

8.4.1 Línea de Tiempo

Gráfico 19. Línea de tiempo



8.5 PLANEACIÓN DEL PROCESO SCRUM

La metodología SCRUM requiere una planeación bien definida, deberá establecerse un equipo de trabajo, unos objetivos y estos objetivos serán divididos en ciclos (iteraciones) de máximo 20 días, realizando reuniones cortas de 10 minutos cada día, para verificar el avance, analizar los problemas y determinar rápidamente un plan de acción para solucionarlos.

8.5.1 Definición del Equipo

El equipo SCRUM estará conformado por:

- **Product Owner:** Es el dueño del producto, determina los objetivos del producto, este rol será desempeñado por Ernesto González, gerente de la compañía SIDIF LTDA.

- **SCRUM Master:** Se asegura que las etapas de SCRUM se lleven a cabo, Alfonso Quintero se encargará de esta labor.
- **Development Team:** El equipo de desarrollo se encarga de realizar todo el proceso de creación del producto, este rol será desempeñado por Sergio Bustos, estudiante de la UNAD y desarrollador de la compañía.

8.5.2 Definición de los Objetivos del Producto (Product Backlog)

Gráfico 20. Product Backlog

The screenshot shows the Visual Studio Online interface for a project titled 'ListasCautelares'. The 'WORK' tab is selected. Under 'Backlogs', 'Backlog items' is chosen. A modal window titled 'Backlog items' is open, showing a table of backlog items. The table has columns: Order, Work Item Type, Title, State, Effort, Iteration Path, and Assigned To. Six items are listed, all of type 'Product Backlog Item' and assigned to 'Sergio Bustos'. The items are: 1. Creación de la Base de Datos, 2. Creación de la aplicación, 3. Creación del Servicio Web, 4. Gestionar Licencias, 5. Gestionar Seguridad, and 6. Registro de actividades.

Order	Work Item Type	Title	State	Effort	Iteration Path	Assigned To
1	Product Backlog Item	Creación de la Base de Datos	New	2	ListasCautelares	Sergio Bustos
2	Product Backlog Item	Creación de la aplicación	New	3	ListasCautelares	Sergio Bustos
3	Product Backlog Item	Creación del Servicio Web	New	1	ListasCautelares	Sergio Bustos
4	Product Backlog Item	Gestionar Licencias	New	2	ListasCautelares	Sergio Bustos
5	Product Backlog Item	Gestionar Seguridad	New	2	ListasCautelares	Sergio Bustos
6	Product Backlog Item	Registro de actividades	New	1	ListasCautelares	Sergio Bustos

Los objetivos del producto (Product BackLog) son:

Tabla 4. Product Backlog

Orden	Título	Esfuerzo (1-3)	Descripción
1	Creación de la Base de Datos	2	Crear la base de datos, las tablas, vistas y procedimientos almacenados necesarios
2	Creación de la aplicación	3	Crear la aplicación de consultas
3	Creación del Servicio Web	1	Crear el servicio Web

4	Gestionar Licencias	2	Crear el gestor de licencias, y aplicar la validación a la aplicación y al servicio Web
5	Gestionar Seguridad	2	Crear el gestor de seguridad, la autorización y el acceso.
6	Registro de actividades	1	Crear el registro de actividades del sistema y consultas.

8.5.3 Definición de los objetivos de las iteraciones (Sprint Backlog)

Sprint 1: En el primer Sprint, se realiza el análisis y diseño de la aplicación.

Gráfico 21. Sprint 1 Backlog

Title	State	Assigned To
Crear diagramas de análisis	New	Sergio Bustos
Crear requerimientos	New	Sergio Bustos
Crear diagramas de diseño	New	Sergio Bustos

Sprint 2: En el segundo Sprint, se atienden los ítems del producto:

- Creación de la Base de Datos
- Creación de la aplicación
- Creación del Servicio Web

Gráfico 22. Sprint 2 Backlog

Title	State
Crear base de datos con tablas de TOPBIS con datos	New
Crear tablas nuevas	New
Crear vista de unificación	New
Crear procedimiento almacenado de consulta	New
Crear clases de entidades	New
Crear clases de DAL de consulta	New
Crear clases de BL de consulta	New
Crear clases utilitarias	New
Crear formularios Web de consulta	New
Crear Consulta Web que realice consultas	New

Sprint 3: En el tercer Sprint, se atienden los ítems del producto:

- Gestión Licencias
- Gestión Seguridad
- Registro de actividades

Gráfico 23. Sprint 3 Backlog

Title	State	Assigned To
Crear aplicación de generación de licencias	New	Sergio Bustos
Crear clases para validación de licencia	New	Sergio Bustos
Agregar validación de licencia a la aplicación	New	Sergio Bustos
Agregar validación de licencia al servicio web	New	Sergio Bustos
Crear clases DAL de Autenticación	New	Sergio Bustos
Crear clases BL de Autenticación	New	Sergio Bustos
Crear clases DAL de Autorización	New	Sergio Bustos
Crear clases BL de Autorización	New	Sergio Bustos
Crear formulario de Inicio de Sesión	New	Sergio Bustos

Crear clases DAL para registro de actividades	New	Sergio Bustos
Crear clases BL para registro de actividades	New	Sergio Bustos
Agregar registro de actividades a las clases que lo requerieran	New	Sergio Bustos
Crear clases DAL para registro de consulta	New	Sergio Bustos
Crear clases BL para registro de consulta	New	Sergio Bustos
Agregar registro de consulta a las clases que lo requerieran	New	Sergio Bustos
Crear mantenimiento de usuarios	New	Sergio Bustos
Crear mantenimiento de permisos de usuarios	New	Sergio Bustos
Crear consulta de registro de actividades	New	Sergio Bustos
Crear consulta de registro de consultas	New	Sergio Bustos

Sprint 4: En el cuarto Sprint, se prepara para la entrega, pruebas, implementación y documentación.

Gráfico 24. Sprint 4 Backlog

The screenshot shows the Visual Studio Online interface for the 'ListasCautelares' team. The top navigation bar includes 'HOME', 'CODE', 'WORK', 'BUILD', and 'TEST'. The 'WORK' tab is selected. On the left, a sidebar shows 'Backlogs' and 'Queries'. Under 'Backlogs', there are sections for 'Current' (Sprint 1), 'Future' (Sprint 2, Sprint 3), and 'Sprint 4' (which is currently selected). The main area displays the 'Backlog' for 'ListasCautelares Team Sprint 4'. The backlog items are listed in a table with columns for 'Title', 'State', and 'Assigned To'. All items are marked as 'New' and assigned to 'Sergio Bustos'. The items listed are: 'Crear un plan de pruebas', 'Crear y ejecutar pruebas unitarias', 'Crear diagramas de implementación', 'Crear instaladores', 'Implementar en servidor Web', 'Complementar documentación de código', 'Crear Manual Técnico', and 'Crear Manual de Usuario'.

Title	State	Assigned To
Crear un plan de pruebas	New	Sergio Bustos
Crear y ejecutar pruebas unitarias	New	Sergio Bustos
Crear diagramas de implementación	New	Sergio Bustos
Crear instaladores	New	Sergio Bustos
Implementar en servidor Web	New	Sergio Bustos
Complementar documentación de código	New	Sergio Bustos
Crear Manual Técnico	New	Sergio Bustos
Crear Manual de Usuario	New	Sergio Bustos

Lista total de objetivos

Tabla 5. Lista total de objetivos

orden	Sprint	T /h	Item	Historia	Asignado a
1	1	32	Crear diagramas de análisis	Antes de iniciar el proceso deben crearse los diagramas de análisis: Diagrama de casos de uso, Diagramas de secuencias de sistema, Diagramas de clases para el análisis	Sergio Bustos
2	1	24	Crear requerimientos	Crear una lista de los requerimientos de la aplicación, enumerados, priorizados y con descripción.	Sergio Bustos
3	1	40	Crear diagramas de diseño	Se deben crear los diagramas del diseño de la aplicación: Diagramas de interacción, Diagramas de secuencia, Diagramas de colaboración, Diagrama de clases para el diseño. Diagrama de actividades. Diagrama de estados	Sergio Bustos
4	2	2	Crear base de datos con tablas de TOPBIS con datos	Debido a que la aplicación se ejecutará apartir de los datos resultantes de otra aplicación, se creará una base de datos con las tablas de dicha aplicación, y se agregarán los datos.	Sergio Bustos
5	2	2	Crear tablas nuevas	Se crean las tablas propias de este sistema	Sergio Bustos
6	2	4	Crear vista de unificación	Las tablas de TOPBLS se unifican en una vista	Sergio Bustos
7	2	8	Crear procedimiento almacenado de consulta	Se debe crear un Procedimiento Almacenado que realice consultas sobre la vista	Sergio Bustos
8	2	1	Crear clases de entidades	Se deben crear las clases que representan entidades en el sistema	Sergio Bustos
9	2	2	Crear clases de DAL de consulta	Se crean las clases de acceso a datos del programa de consulta	Sergio Bustos
10	2	1	Crear clases de BL de consulta	Se crean las clases de lógica y validación del programa de consulta	Sergio Bustos
11	2	1	Crear clases utilitarias	Se crean las clases que son transversales a la aplicación y sirven de apoyo, como conversiones, encriptado, etc.	Sergio Bustos
12	2	60	Crear formularios Web de consulta	Se crea el programa de consulta.	Sergio Bustos
13	2	15	Crear Servicio Web que realice consultas	Se crea un servicio web que realice consultas por medio de las clases de lógica y validación de la consulta	Sergio Bustos
14	3	8	Crear aplicación de generación de licencias	Se debe crear una aplicación que genere las licencias de uso	Sergio Bustos
15	3	4	Crear clases para validación de licencia	Se crean las clases necesarias para validar la licencia	Sergio Bustos
16	3	4	Agregar validación de licencia a la aplicación	Se agregan las llamadas para la validación de la licencia en la aplicación Web	Sergio Bustos
17	3	4	Agregar validación de licencia al servicio web	Se agregan las llamadas para la validación de la licencia en el servicio Web	Sergio Bustos
18	3	2	Crear clases DAL de Autenticación	Se crean las clases DAL que permiten obtener los datos de autenticación de un usuario	Sergio Bustos
19	3	2	Crear clases BL de Autenticación	Se crean las clases BL que permiten ajustar la información para enviar a DAL.	Sergio Bustos
20	3	2	Crear clases DAL de Autorización	Se crean las clases DAL que permiten obtener los datos de autorización y permisos de un usuario	Sergio Bustos
21	3	2	Crear clases BL de Autorización	Se crean las clases BL que permiten ajustar la información para enviar a DAL., y reciben los datos y los entregan en un formato unificado.	Sergio Bustos
22	3	2	Crear formulario de Inicio de Sesión	Se crea un formulario web que permita autenticar a un usuario	Sergio Bustos
23	3	2	Crear clases DAL para registro de actividades	Se crean las clases DAL que permiten registrar actividades	Sergio Bustos
24	3	2	Crear clases BL para registro de actividades	Se crean las clases BL que ajustan la información que se va a registrar	Sergio Bustos
25	3	5	Agregar registro de actividades a las clases que lo requerieran	Se agrega un registro de actividades a cada de las clases que realizan actividades que deban ser registradas	Sergio Bustos

26	3	2	Crear clases DAL para registro de consulta	Se crean las clases DAL que permiten registrar las consultas realizadas	Sergio Bustos
27	3	2	Crear clases BL para registro de consulta	Se crean las clases BL que ajustan la información que se va a registrar	Sergio Bustos
28	3	5	Agregar registro de consulta a las clases que lo requerieran	Se agrega un registro de las consultas realizadas en la pantalla de consultas y el servicio web	Sergio Bustos
29	3	16	Crear mantenimiento de usuarios	Se crea la pantalla que administra los usuarios, con BL y DAL	Sergio Bustos
30	3	16	Crear mantenimiento de permisos de usuarios	Se crea la pantalla que administra los permisos, con BL y DAL	Sergio Bustos
31	3	8	Crear consulta de registro de actividades	Se crea la pantalla que permite ver el registro de actividades	Sergio Bustos
32	3	8	Crear consulta de registro de consultas	Se crea la pantalla que permite ver el registro de las consultas realizadas	Sergio Bustos
33	4	2	Crear un plan de pruebas	Se crea un plan de pruebas que permita validar la calidad del software desarrollado.	Sergio Bustos
34	4	32	Crear y ejecutar pruebas unitarias	Se ejecuta el plan de pruebas	Sergio Bustos
35	4	24	Crear diagramas de implementación	Se crean los diagramas que explican cómo será la implementación del sistema	Sergio Bustos
36	4	8	Crear instaladores	Se crean todos los instaladores	Sergio Bustos
37	4	2	Implementar en servidor Web	Se realiza una instalación en un servidor web de la compañía.	Sergio Bustos
38	4	2	Complementar documentación de código	Se revisa el código y se complementa la documentación de las clases y métodos.	Sergio Bustos
39	4	12	Crear Manual Técnico	Se crea un manual técnico con la información del diseño	Sergio Bustos
40	4	12	Crear Manual de Usuario	Se crea un manual usuario con la información del uso del sistema.	Sergio Bustos

8.5.4 Planeación de las iteraciones (Sprints)

Las Iteraciones (Sprints) tendrán una duración de 12 días laborales de 8 horas diarias, y se han distribuido de la siguiente manera:

Sprint 1: Desde el 1º de Abril hasta el 16 de Abril de 2014

Sprint 2: Desde el 17 de Abril hasta el 2 de Mayo de 2014

Sprint 3: Desde el 3 de Mayo hasta el 20 de Mayo de 2014

Sprint 4: Desde el 21 de Mayo hasta el 05 de Junio de 2014

8.5.5 Definición de Terminado

Todo el equipo Scrum dará por terminado un elemento de la lista de pendientes del Sprint (Sprint Backlog), cuando realice las actividades que se ha planeado que realice y esté listo para ir a pruebas, las modificaciones se realizarán unas vez terminadas las pruebas, y un incremento se dará por terminado cuando las pruebas sean aprobadas correctamente y en su totalidad.

8.6 REUNIÓN DE PLANIFICACIÓN DEL SPRINT 1 – 2014/04/01

El objetivo de la reunión es identificar los objetivos que deberán cumplirse en la iteración y establecer un procedimiento para llevarlos a cabo.

8.6.1 Asistentes

- Ernesto González – Product Owner
- Sergio Bustos – Equipo de Desarrollo
- Alfonso Quintero – Scrum Master

8.6.2 ¿Qué se completará en este Sprint? (Sprint Backlog)

En el primer Sprint, se realiza el análisis y diseño de la aplicación, esto es:

- Diagramas de Casos de Uso
- Diagramas de Clases para el análisis
- Diagramas de Secuencias del Sistema para el análisis
- Definición de los requerimientos funcionales

- Definición de los requerimientos no funcionales
 - Diagramas de Secuencias para el Diseño
 - Diagramas de Clases para el Diseño
 - Diagramas de Actividades
 - Diagramas de Estado

8.6.3 ¿Cómo se conseguirá completar el trabajo seleccionado?

En el Sprint Diario, el equipo estimará las tareas que pueden ser completadas durante un día, y no dejará tareas pendientes de un día para otro, si necesita iniciar una tarea poco antes de finalizar el día, deberá posponer su inicio hasta el día siguiente, si una tarea dura más de un día, entonces deberá dividirse en tareas más cortas.

El Product Owner ha ofrecido literatura relacionada a la creación de los diagramas que se usará como base, así como la literatura provista por la UNAD.

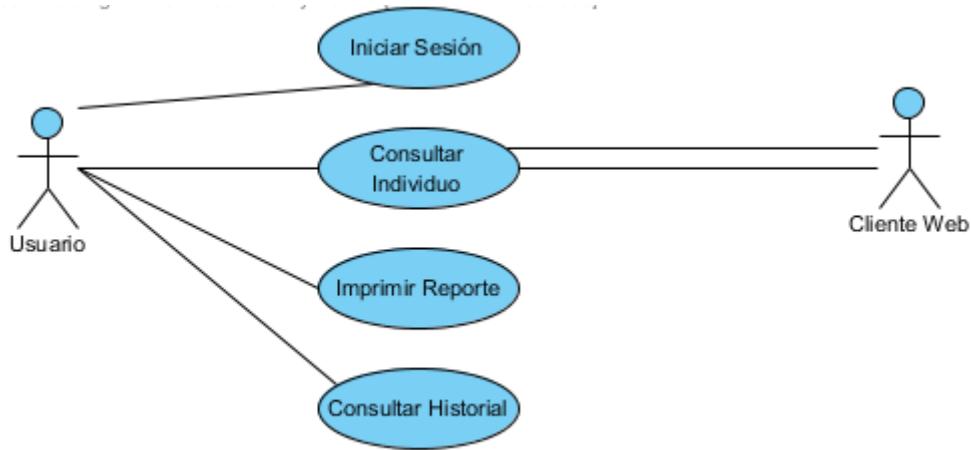
8.6.4 Acta de la Reunión

Gráfico 25. Acta de la reunión Sprint 1

8.7 ANÁLISIS DEL NEGOCIO

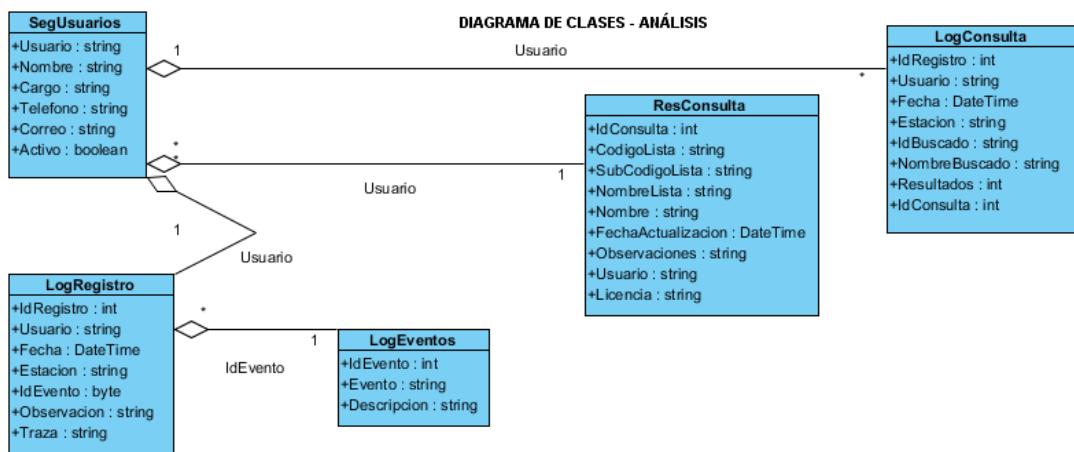
8.7.1 Diagrama de Casos de Uso

Gráfico 26. Diagrama de Casos de Uso



8.7.2 Diagrama de clases para el análisis

Gráfico 27. Diagrama de clases para el análisis



Las entidades requeridas durante la etapa de análisis son:

SegUsuarios: Representa la estructura de un registro de usuario.

LogRegistro: Representa la estructura de un registro de auditoría de actividades.

LogConsulta: Representa la estructura de un registro del histórico de consultas realizadas.

ResConsulta: Representa la estructura de un registro de la respuesta de una consulta.

LogEventos: Representa la estructura de los tipos de eventos admitidos en LogRegistro.

8.7.3 Diagrama de Secuencias del Sistema

En los diagramas de Secuencia del Sistema en la fase de análisis, se presentan las interacciones descritas en los casos de uso, sin el detalle interno, es decir; el sistema se presenta al actor como una caja negra.

Gráfico 28. Diagrama de Secuencia: Iniciar sesión

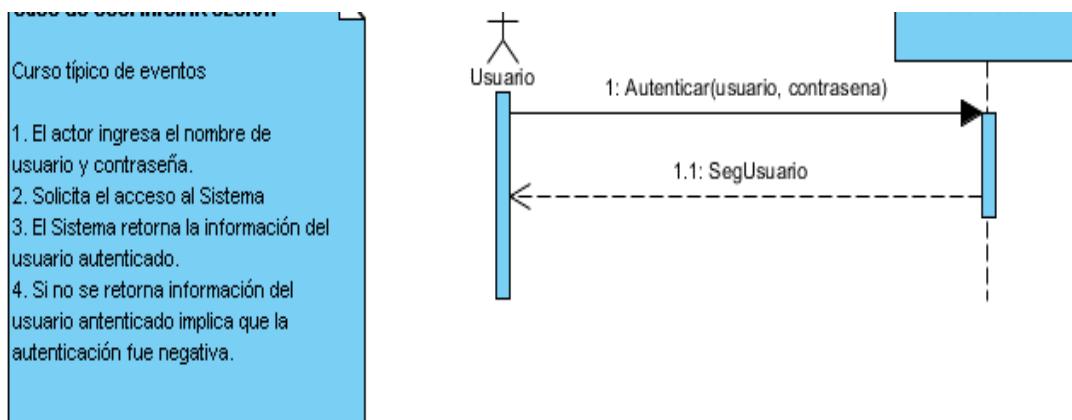


Gráfico 29. Diagrama de Secuencia: Consultar Individuo

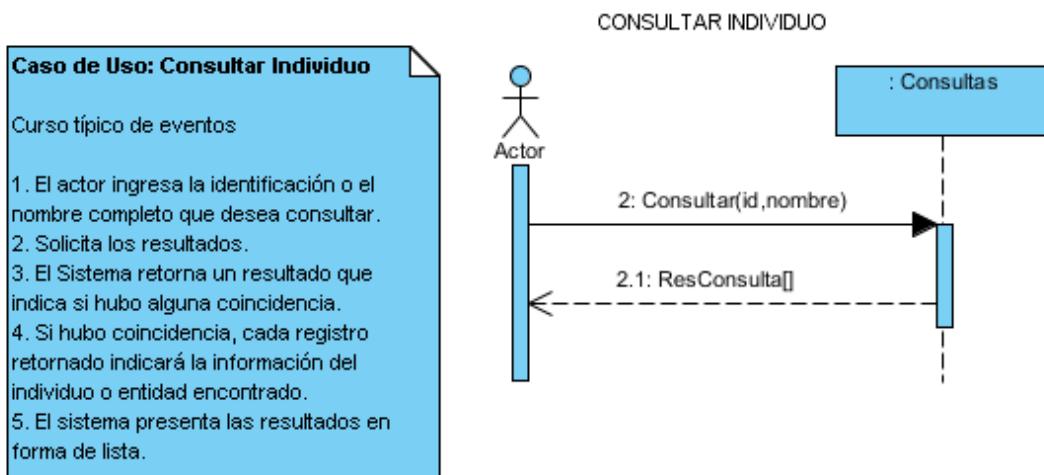


Gráfico 30. Diagrama de Secuencia: Consultar Individuo (Cliente Web)

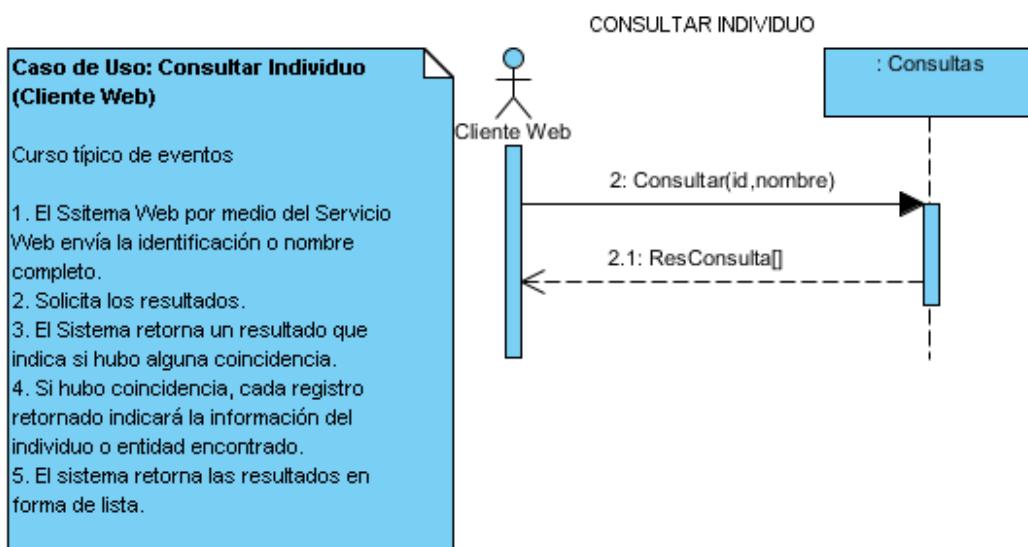


Gráfico 31. Diagrama de Secuencia: Imprimir

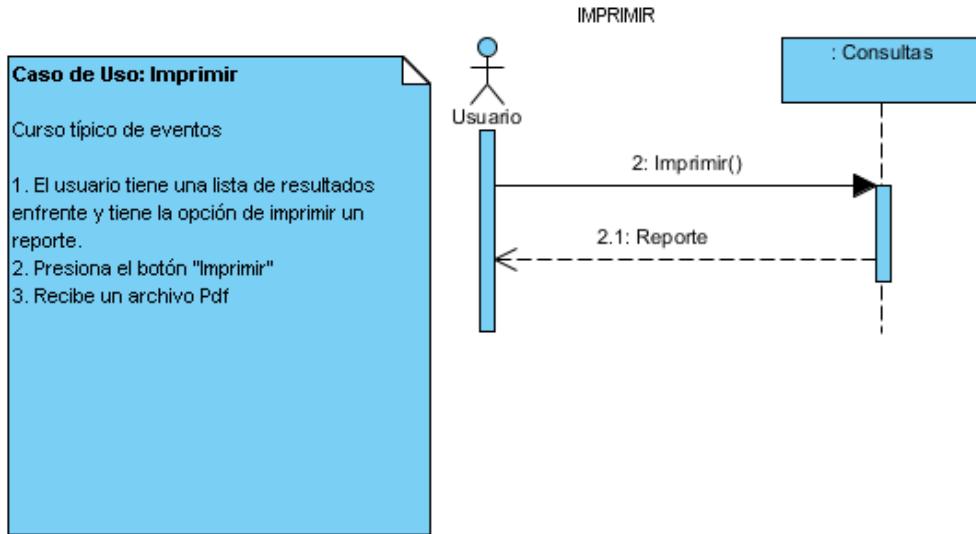
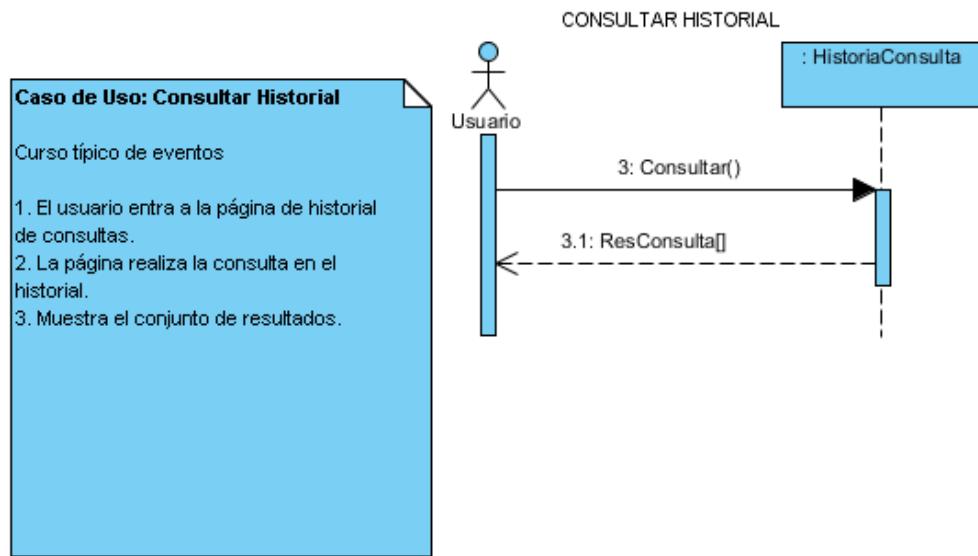


Gráfico 32. Diagrama de Secuencia: Consultar Historial



8.8 ANÁLISIS DE REQUERIMIENTOS

“El análisis de requisitos es una tarea de ingeniería del software que cubre el hueco entre la definición del software a nivel sistema y el diseño de software.” [PRESSMAN 2002]

8.8.1 Requerimientos funcionales

Tabla 6. Requerimientos Funcionales

Identificación	RF01
Nombre	Conexión a TopBls
Características	La aplicación deberá conectarse a la base de datos de TopBLS
Descripción	La aplicación deberá conectarse a la base de datos de TopBLS por medio de Ado.Net, la base de datos deberá ser SqlServer 2005 o superior
Prioridad	Alta

Identificación	RF02
Nombre	Vista personalizada
Características	Se creará una vista que unifique las tablas implicadas en el proceso
Descripción	La aplicación realizará las consultas sobre una vista que deberá ser creada, para unificar en un solo formato los datos entregados por las tablas mae_Lista_un y Mae_Lista_Ofac

Prioridad	Alta
-----------	------

Identificación	RF03
Nombre	Algoritmo de Búsqueda
Características	Se creará un algoritmo que permita optimizar el proceso de Búsqueda
Descripción	La aplicación realizará las consultas utilizando un algoritmo que permita buscar de manera sencilla un individuo a partir de un nombre completo o identificación
Prioridad	Alta

Identificación	RF04
Nombre	Resultados de Búsqueda positivos
Características	Reporte de resultados encontrados
Descripción	La aplicación deberá mostrar un reporte con la información encontrada, por cada uno de los individuos, no se expedirá un reporte de todas las coincidencias, sino uno por uno
Prioridad	Alta

Identificación	RF05
Nombre	Resultados de Búsqueda negativos
Características	Reporte de resultados vacío

Descripción	La aplicación deberá mostrar un reporte en donde mencione que la cadena buscada no fue encontrada y mencionará los tipos de listas y cantidad de registros que contiene la base de datos a una fecha determinada.
Prioridad	Alta

Identificación	RF06
Nombre	WebService de Consulta
Características	WebService de Consulta para aplicaciones externas
Descripción	Las aplicaciones externas podrán consultar en las listas a través de un WebService que implemente la búsqueda de la aplicación
Prioridad	Alta

Identificación	RF07
Nombre	Respuesta del WebService de Consulta
Características	Respuesta WebService de Consulta para aplicaciones externas
Descripción	El WebService deberá retornar una estructura compleja que indique si se encontraron resultados o no, la cantidad de registros en la base de datos, información sobre la licencia y una lista de las coincidencias encontradas.
Prioridad	Alta

Identificación	RF08
Nombre	Validación de la Licencia
Características	La aplicación deberá funcionar únicamente con una licencia válida
Descripción	Tanto el WebService como la aplicación deberán validar primero si la licencia es válida, si lo es, permitir la consulta, de lo contrario retornar una respuesta indicando que la licencia no es válida
Prioridad	Alta

Identificación	RF09
Nombre	Generación de las Licencias
Características	Un ejecutable independiente tendrá la capacidad de generar las licencias
Descripción	Se creará un ejecutable independiente que tenga la capacidad de generar las licencias para la aplicación de Búsquedas y WebService
Prioridad	Alta

8.8.2 Requerimientos no funcionales

Tabla 7. Requerimientos no funcionales

Identificación	RNF01
Nombre	Aplicación Web Móvil para prueba de WebServices
Características	Una aplicación Web, optimizada para móviles para prueba de WebServices

Descripción	Los WebServices serán probados a través de una aplicación web optimizada para móviles.
Prioridad	Media

8.9 DISEÑO LÓGICO

La actividad de diseño se refiere al establecimiento de las estructuras de datos, la arquitectura en general del software, representaciones de interfaz y algoritmos. El proceso de diseño traduce requisitos en una representación de software

8.9.1 Diagramas de Secuencias de Transacciones

Gráfico 33. Diagrama de Secuencias de Transacciones: Iniciar Sesión

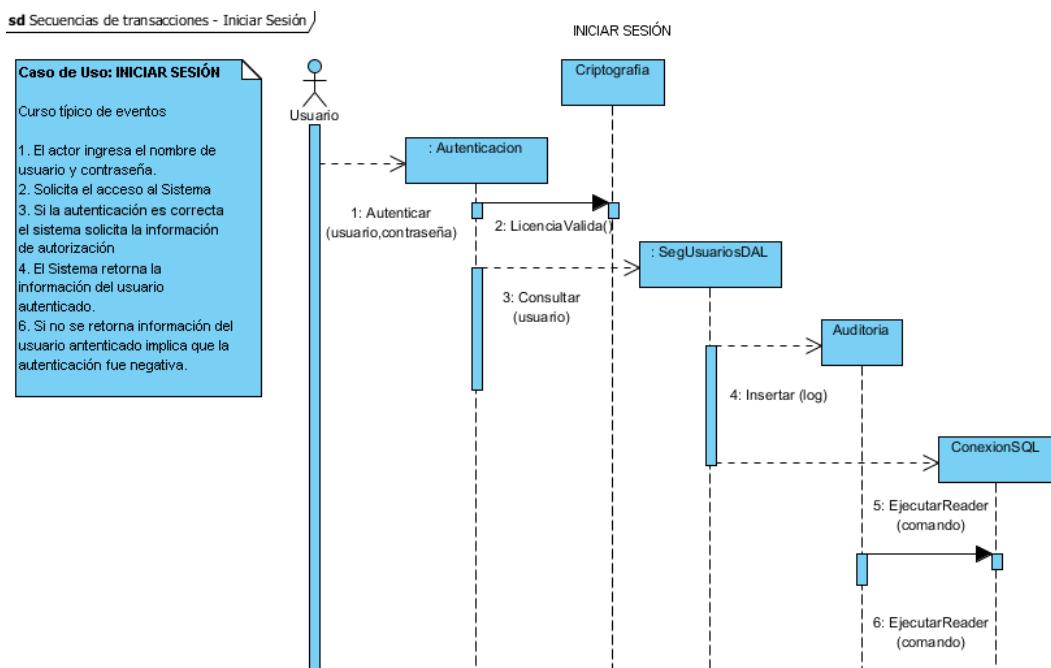


Gráfico 34. Diagrama de Secuencias de Transacciones: Consultar Individuo

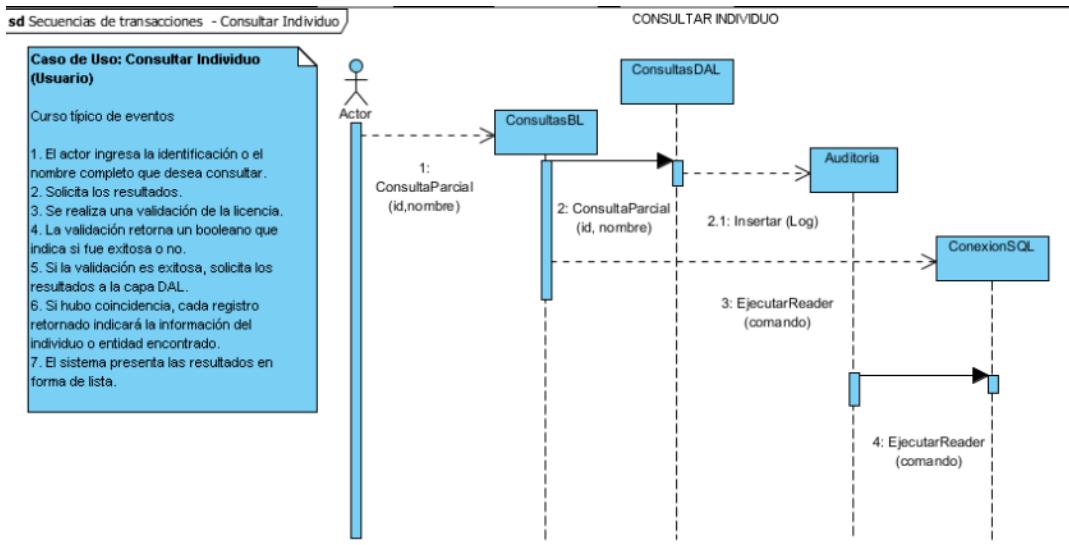


Gráfico 35. Diagrama de Secuencias de Transacciones: Consultar Individuo (Cliente Web)

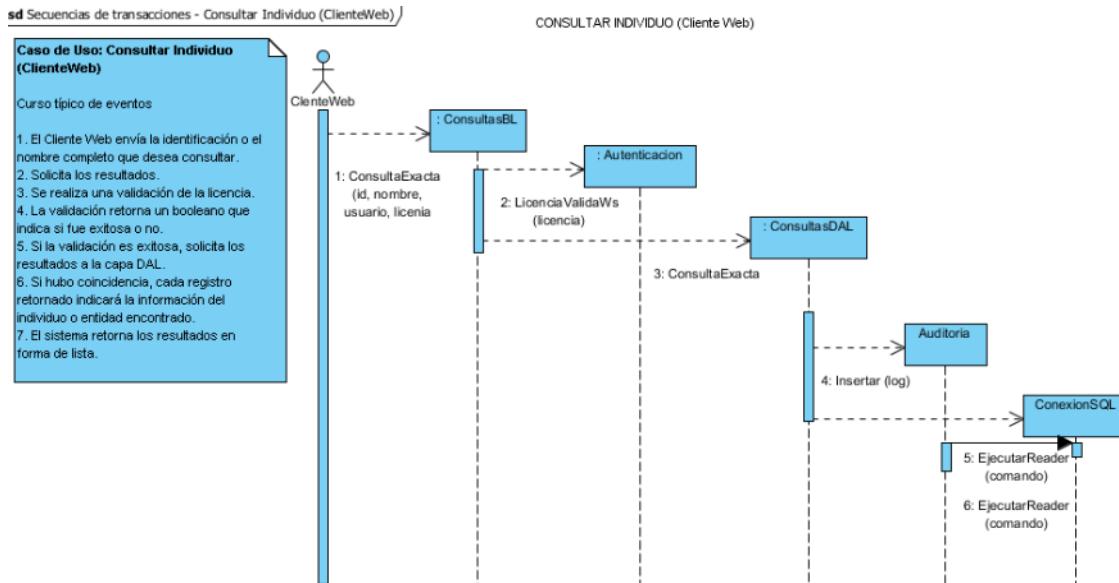


Gráfico 36. Diagrama de Secuencias de Transacciones: Imprimir

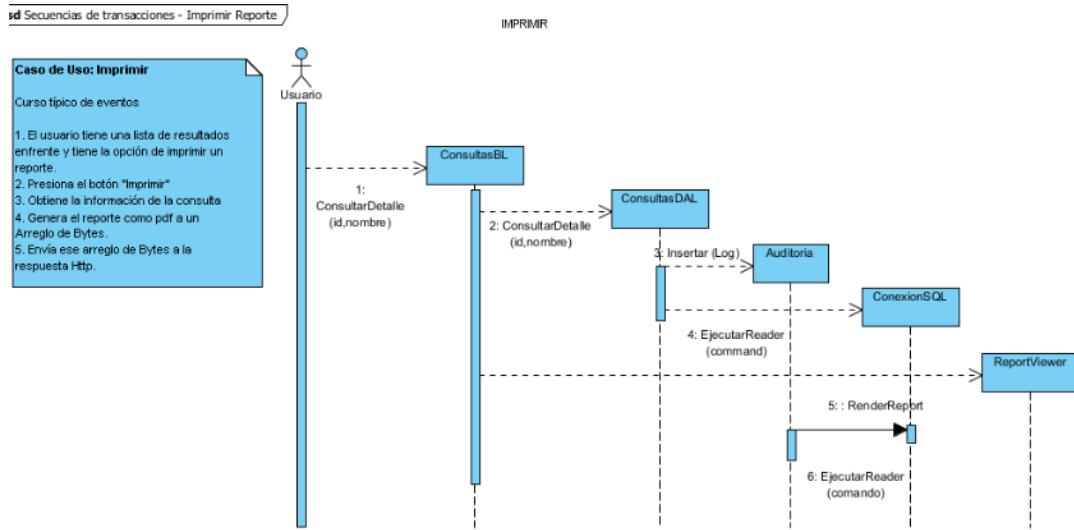
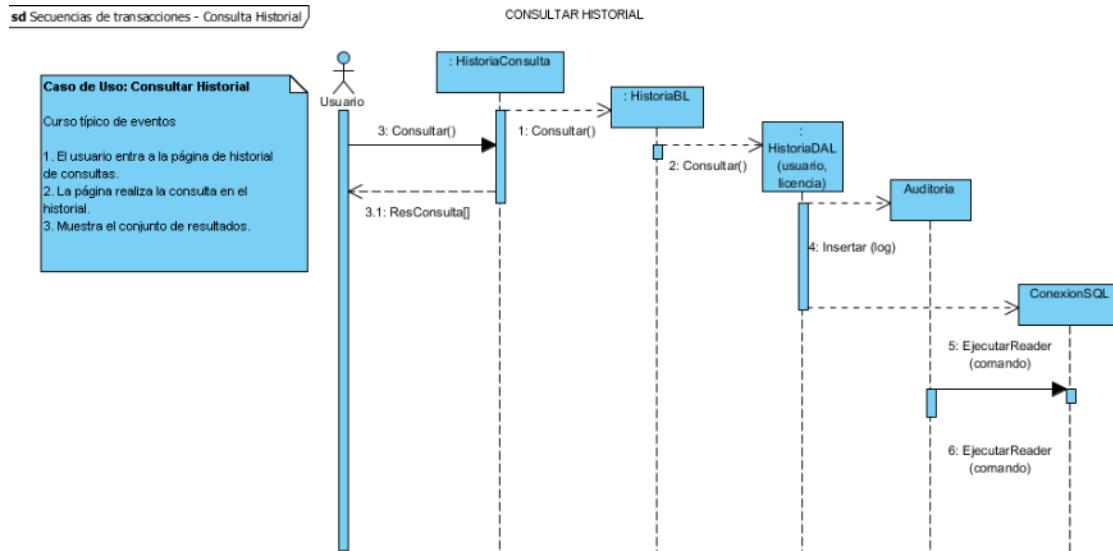


Gráfico 37. Diagrama de Secuencias de Transacciones: Consultar Historial



8.9.2 Diagramas de Colaboración

Gráfico 38. Diagrama de Colaboración: Iniciar Sesión

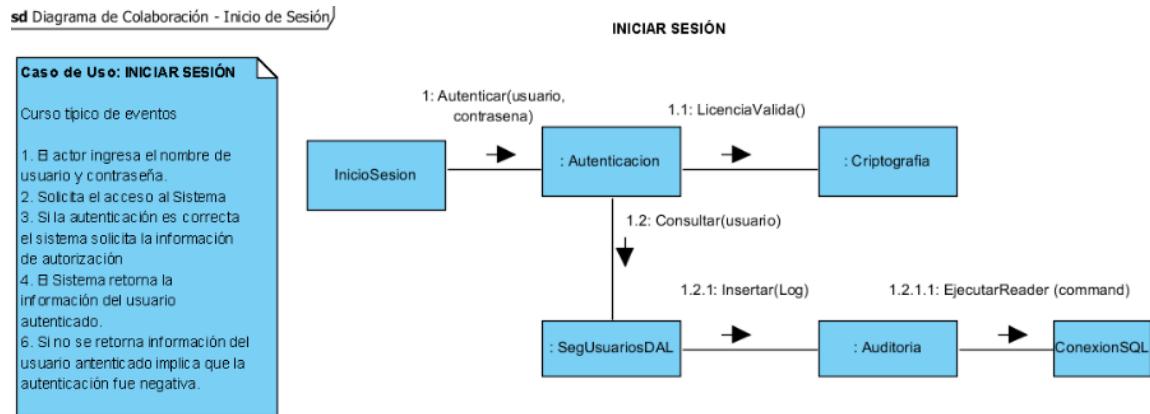


Gráfico 39. Diagrama de Colaboración: Consultar Individuo

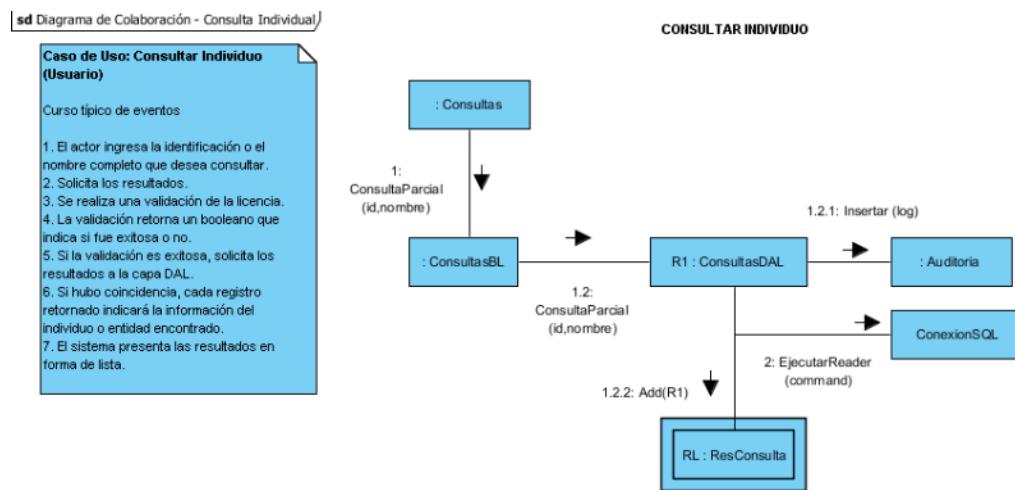


Gráfico 40. Diagrama de Colaboración: Consultar Individuo (Cliente Web)

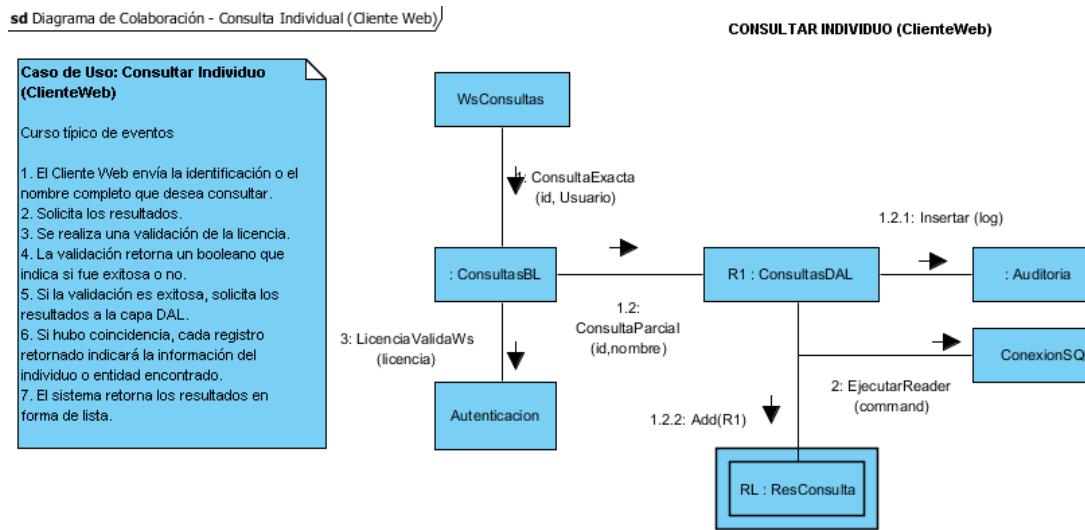


Gráfico 41. Diagrama de Colaboración: Imprimir

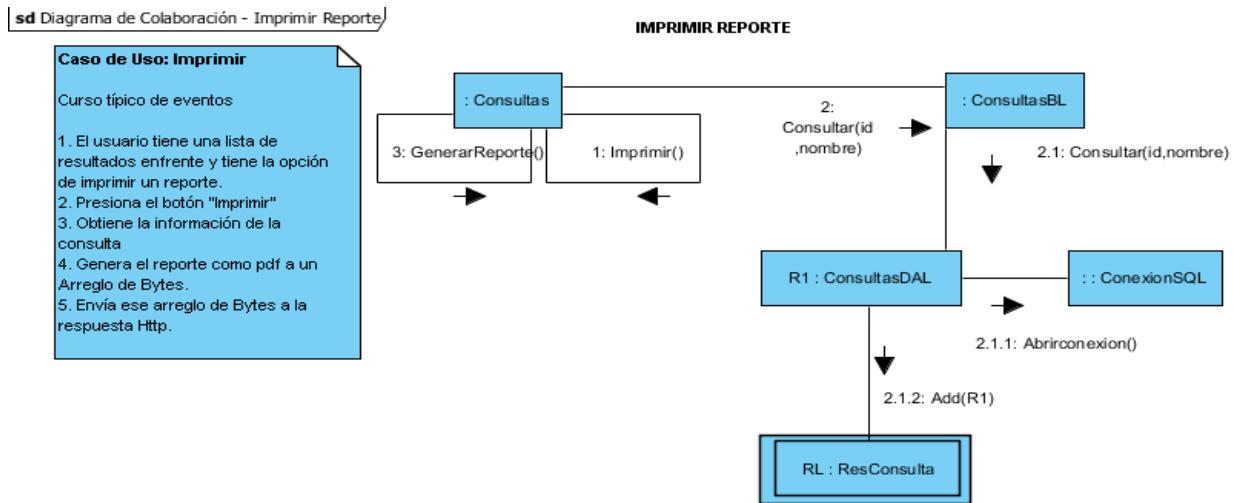
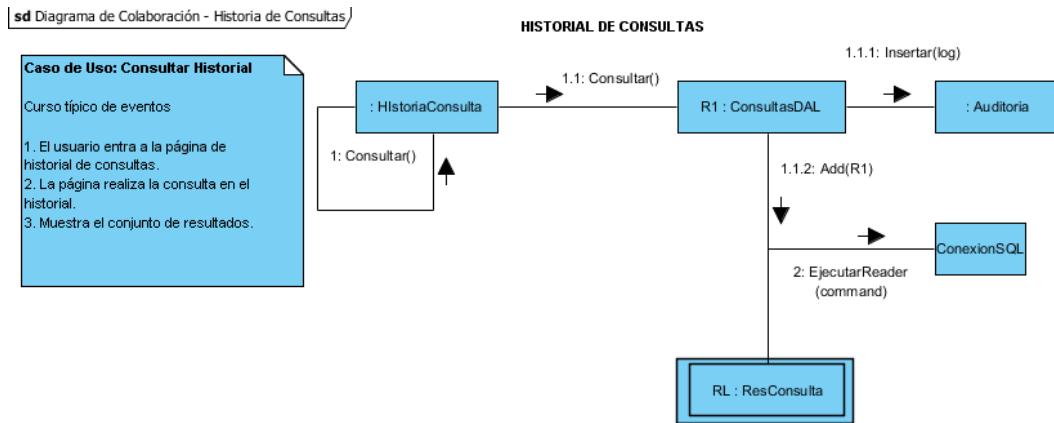
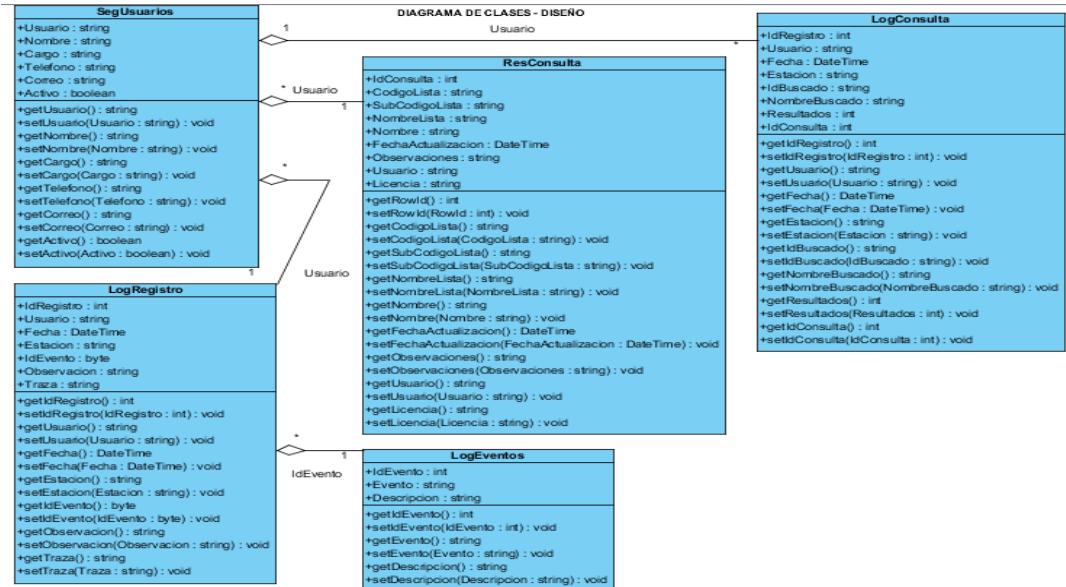


Gráfico 42. Diagrama de Colaboración: Consultar Historial



8.9.3 Diagrama de Clases para el Diseño

Gráfico 43. Diagrama de Clases para el Diseño



Las entidades requeridas durante la etapa de diseño son:

SegUsuarios: Representa la estructura de un registro de usuario.

Tabla 8. Descripción de la tabla SegUsuarios

Campo	Tipo de Datos	Descripción
<i>Usuario</i>	String	Identifica de manera única a un usuario, es la llave de la tabla.
<i>Nombre</i>	String	Nombre completo para el usuario
<i>Cargo</i>	String	Cargo que ocupa el usuario
<i>Telefono</i>	String	Teléfono de contacto
<i>Correo</i>	String	Correo electrónico de contacto
<i>Activo</i>	Boolean	Determina si el usuario está activo o no

LogRegistro: Representa la estructura de un registro del histórico de auditoría de actividades.

Tabla 9. Descripción de la tabla LogRegistro

Campo	Tipo de Datos	Descripción
<i>IdRegistro</i>	Int	Identifica de manera única el registro
<i>Usuario</i>	String	Usuario que realizó la operación, viene de SegUsuarios
<i>Fecha</i>	DateTime	Fecha/Hora en que sucedió el evento
<i>Estacion</i>	String	Información del usuario, PC o Dominio si se encuentra disponible
<i>IdEvento</i>	Byte	Identificador del evento
<i>Observacion</i>	String	Información descriptiva del evento.
<i>Traza</i>	String	Cuando el evento es un error se guarda la información de la traza del error.

LogConsulta: Representa la estructura de un registro del histórico de consultas realizadas.

Tabla 10. Descripción de la tabla LogConsulta

Campo	Tipo de Datos	Descripción
<i>IdRegistro</i>	Int	Identifica de manera única el registro
<i>Usuario</i>	String	Usuario que realizó la operación, viene de SegUsuarios
<i>Fecha</i>	DateTime	Fecha/Hora en que sucedió el evento
<i>Estacion</i>	String	Información del usuario, PC o Dominio si se encuentra disponible
<i>IdBuscado</i>	Byte	Documento de identificación que fue buscado en las listas.
<i>NombreBuscado</i>	String	Nombre que fue buscado en las listas
<i>Resultados</i>	String	Cantidad de resultados encontrados.
<i>IdConsulta</i>	Int	Identificador de la consulta, viene de ResConsulta, pero no hay restricciones.

ResConsulta: Representa la estructura de un registro de la respuesta de una consulta realizadas.

Tabla 11. Descripción de la tabla ResConsulta

Campo	Tipo de Datos	Descripción
<i>IdConsulta</i>	Int	Identifica de manera única la consulta
<i>CodigoLista</i>	String	Código que aparece en la lista en la que se encontró la coincidencia
<i>SubCodigoLista</i>	String	Código complementario que aparece en la lista en la que se encontró la coincidencia.
<i>NombreLista</i>	String	Nombre de la lista en la que se encontró
<i>Nombre</i>	String	Nombre de la coincidencia

<i>FechaActualizacion</i>	DateTime	Fecha en la que fue actualizado el registro por la fuente original.
<i>Observaciones</i>	String	Información complementaria de la coincidencia
<i>Usuario</i>	String	Usuario que realiza la consulta, viene de SegUsuarios
<i>Licencia</i>	String	Licencia con la que se ejecutó la consulta.

LogEventos: Representa la estructura de los tipos de eventos admitidos en LogRegistro.

Tabla 12. Descripción de la tabla LogEventos

<i>Campo</i>	<i>Tipo de Datos</i>	<i>Descripción</i>
<i>IdEvento</i>	Int	Identifica de manera única el evento
<i>Evento</i>	String	Nombre corto del evento
<i>Descripcion</i>	String	Información descriptiva del evento.

8.9.4 Diagramas de Actividades

Gráfico 44. Diagrama de Actividades: Iniciar Sesión

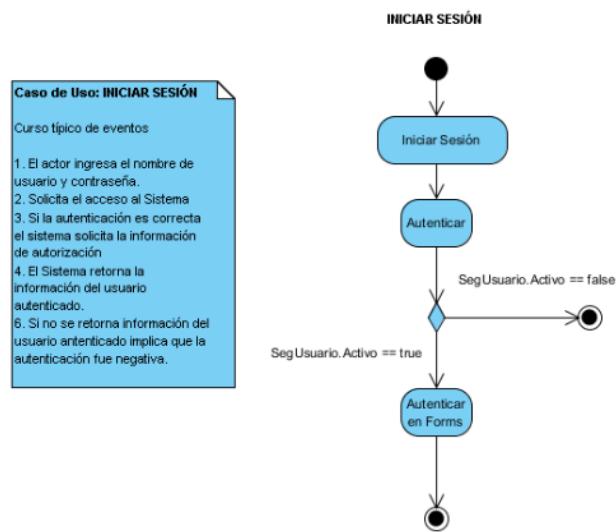


Gráfico 45. Diagrama de Actividades: Consultar Individuo

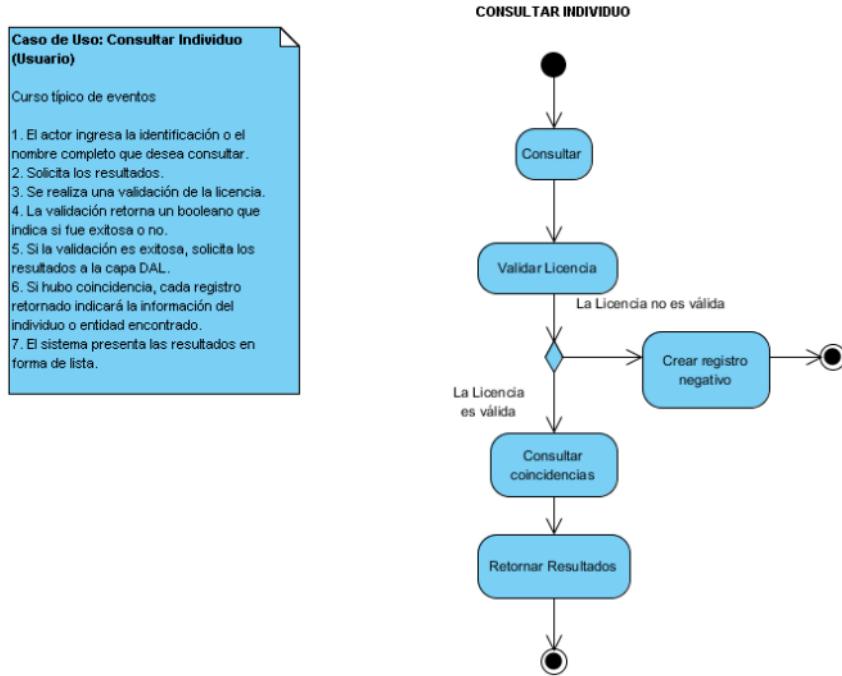


Gráfico 46. Diagrama de Actividades: Consultar Individuo (Cliente Web)

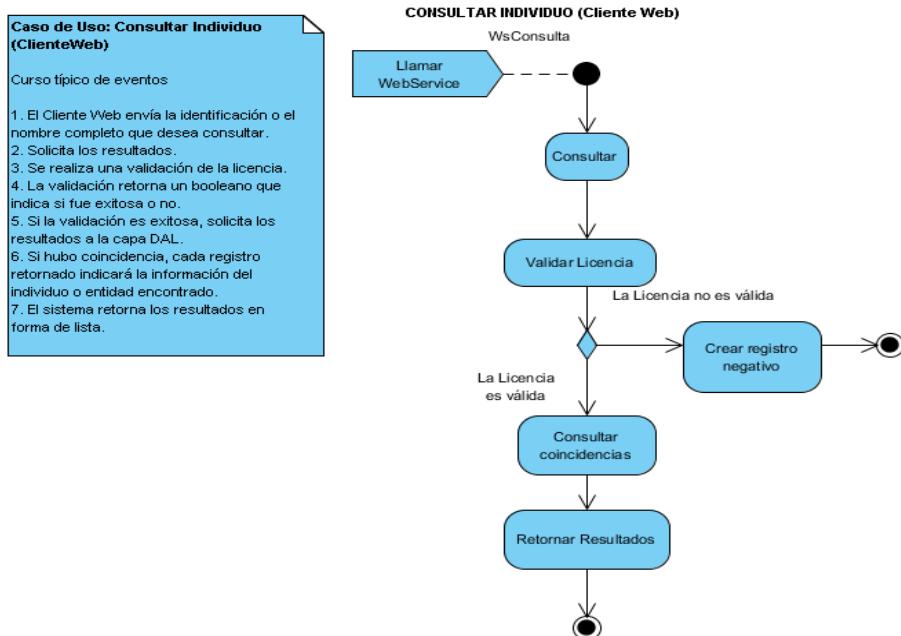


Gráfico 47. Diagrama de Actividades: Imprimir

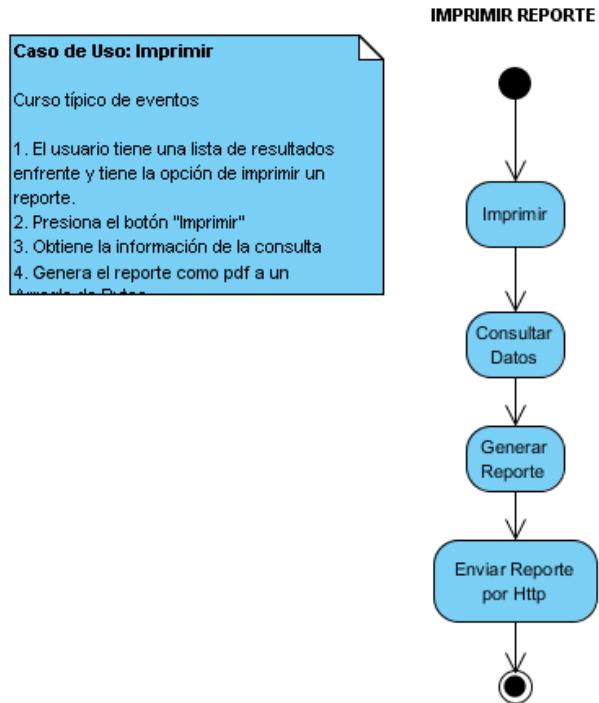
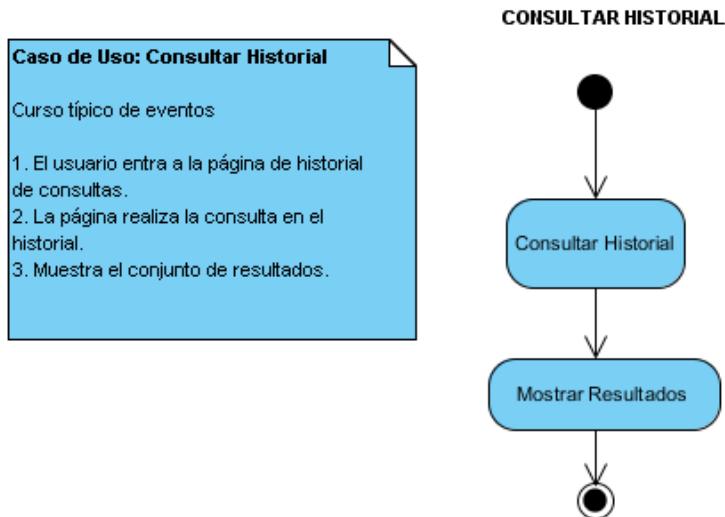


Gráfico 48. Diagrama de Actividades: Consultar Historial



8.9.5 Diagramas de Estado

Gráfico 49. Diagrama de Estado: Iniciar Sesión

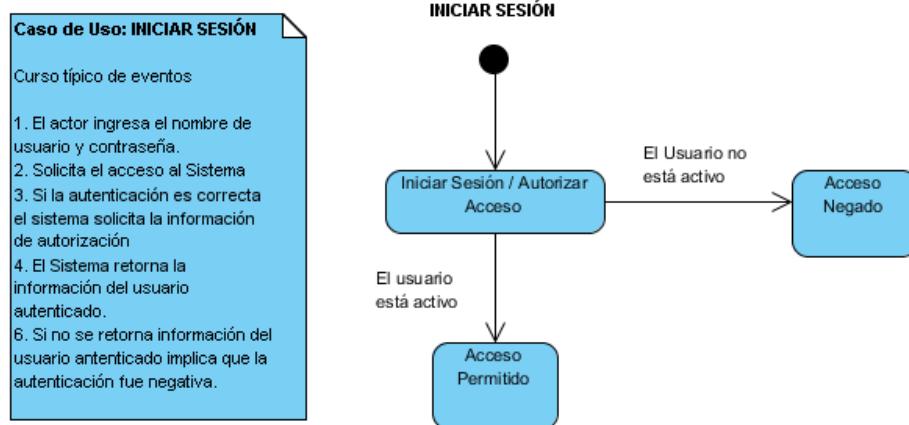
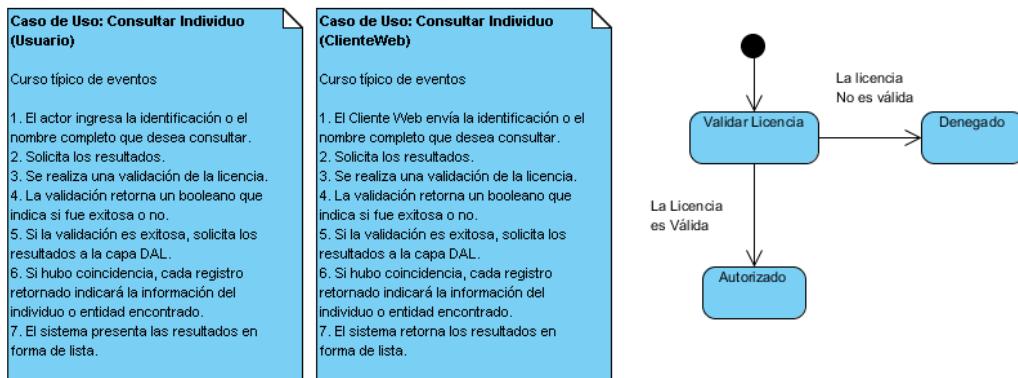


Gráfico 50 . Diagrama de Estado: Consultar Individuo - Usuario y Web



8.10 REUNIÓN DE PLANIFICACIÓN DEL SPRINT 2 – 2014/04/17

El objetivo de la reunión es identificar los objetivos que deberán cumplirse en la iteración y establecer un procedimiento para llevarlos a cabo.

8.10.1 Asistentes

- Ernesto González – Product Owner
- Sergio Bustos – Equipo de Desarrollo
- Alfonso Quintero – Scrum Master

8.10.2 ¿Qué se completará en este Sprint? (Sprint Backlog)

En el Segundo Sprint, se realizará el diseño físico y construcción de la aplicación y el servicio Web, el detalle de estas actividades es:

- Diseño físico de los objetos del sistema (Diagrama Entidad-Relación)
- Creación de la base de datos
- Importación de las tablas de TopBls
- Creación de las Tablas nuevas
- Creación de la vista de integración
- Creación del procedimiento almacenado de consulta
- Creación de las capas de Datos, Lógica de Negocio y Presentación
- Pruebas de consulta.
- Creación del reporte de coincidencias
- Creación del Servicio Web.

8.10.3 ¿Cómo se conseguirá completar el trabajo seleccionado?

En el Sprint Diario, el equipo estimará las tareas que pueden ser completadas durante un día, y no dejará tareas pendientes de un día para otro, si necesita iniciar una tarea poco antes de finalizar el día, deberá posponer su inicio hasta el día siguiente, si una tarea dura más de un día, entonces deberá dividirse en tareas más cortas, el Product Owner ha ofrecido literatura relacionada a la creación de los diagramas que se usará como base, así como la literatura provista por la UNAD.

8.10.4 Acta de la Reunión

Gráfico 51. Acta de la reunión Sprint 2

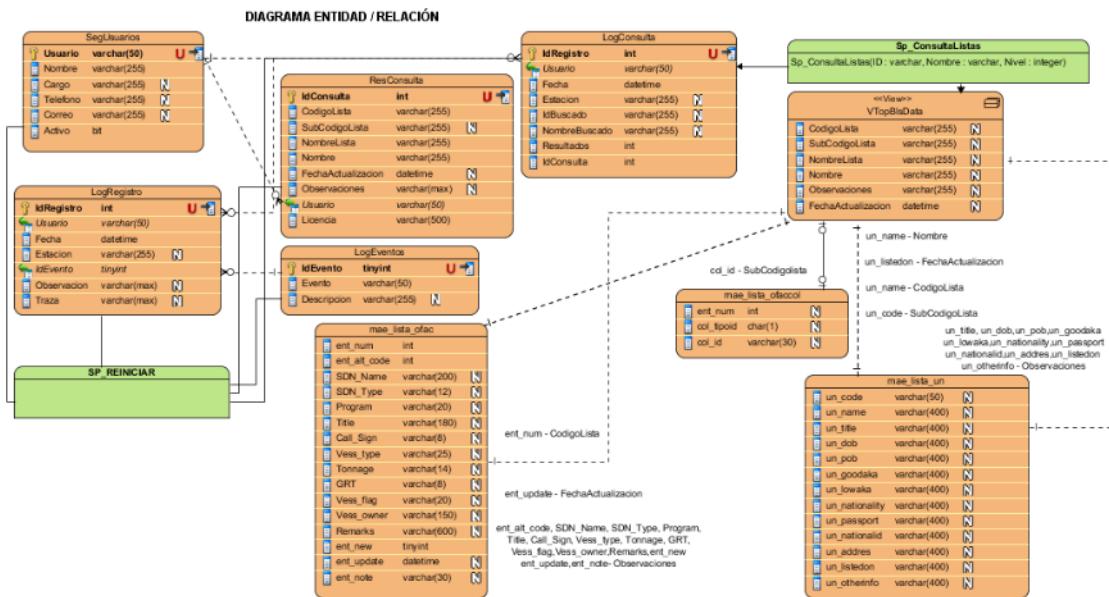


8.11 DISEÑO FÍSICO

“En el diseño físico se especifican las características de los componentes del sistema requeridos para poner en práctica el diseño lógico. Se emplean diagramas con mayor detalle técnico y con la premisa de que se deben brindar los elementos que permitan llevar a la práctica la solución definida.”¹⁰³

8.11.1 Diagrama Entidad – Relación

Gráfico 52. Diagrama Entidad - Relación



Los Objetos de la base de datos son:

¹⁰³ GALINDEZ, Vilmary. Ingeniería de Software – Diseño físico. 2013 disponible en <http://vilmarygalindez.blogspot.com/2011/02/diseno-fisico.html>

8.11.2 Tablas

SegUsuarios: Representa la estructura de un registro de usuario.

Tabla 13. Descripción de la tabla SegUsuarios

<i>Campo</i>	<i>Tipo de Datos</i>	<i>Descripción</i>
<i>Usuario</i>	Varchar	Identifica de manera única a un usuario, es la llave de la tabla.
<i>Nombre</i>	Varchar	Nombre completo para el usuario
<i>Cargo</i>	Varchar	Cargo que ocupa el usuario
<i>Telefono</i>	Varchar	Teléfono de contacto
<i>Correo</i>	Varchar	Correo electrónico de contacto
<i>Activo</i>	Boolean	Determina si el usuario está activo o no

LogRegistro: Representa la estructura de un registro del histórico de auditoría de actividades.

Tabla 14. Descripción de la tabla LogRegistro

<i>Campo</i>	<i>Tipo de Datos</i>	<i>Descripción</i>
<i>IdRegistro</i>	Int	Identifica de manera única el registro
<i>Usuario</i>	Varchar	Usuario que realizó la operación, viene de SegUsuarios
<i>Fecha</i>	DateTime	Fecha/Hora en que sucedió el evento

<i>Estacion</i>	Varchar	Información del usuario, PC o Dominio si se encuentra disponible
<i>IdEvento</i>	Smallint	Identificador del evento
<i>Observacion</i>	Varchar	Información descriptiva del evento.
<i>Traza</i>	Varchar	Cuando el evento es un error se guarda la información de la traza del error.

LogConsulta: Representa la estructura de un registro del histórico de consultas realizadas.

Tabla 15. Descripción de la tabla LogConsulta

<i>Campo</i>	<i>Tipo de Datos</i>	<i>Descripción</i>
<i>IdRegistro</i>	Int	Identifica de manera única el registro
<i>Usuario</i>	Varchar	Usuario que realizó la operación, viene de SegUsuarios
<i>Fecha</i>	DateTime	Fecha/Hora en que sucedió el evento
<i>Estacion</i>	Varchar	Información del usuario, PC o Dominio si se encuentra disponible
<i>IdBuscado</i>	Smallint	Documento de identificación que fue buscado en las listas.
<i>NombreBuscado</i>	Varchar	Nombre que fue buscado en las listas
<i>Resultados</i>	Varchar	Cantidad de resultados encontrados.
<i>IdConsulta</i>	Int	Identificador de la consulta, viene de ResConsulta, pero no hay restricciones.

ResConsulta: Representa la estructura de un registro de la respuesta de una consulta realizadas.

Tabla 16. Descripción de la tabla ResConsulta

Campo	Tipo de Datos	Descripción
<i>IdConsulta</i>	Int	Identifica de manera única la consulta
<i>CodigoLista</i>	VARCHAR	Código que aparece en la lista en la que se encontró la coincidencia
<i>SubCodigoLista</i>	VARCHAR	Código complementario que aparece en la lista en la que se encontró la coincidencia.
<i>NombreLista</i>	VARCHAR	Nombre de la lista en la que se encontró
<i>Nombre</i>	VARCHAR	Nombre de la coincidencia
<i>FechaActualizacion</i>	DateTime	Fecha en la que fue actualizado el registro por la fuente original.
<i>Observaciones</i>	VARCHAR	Información complementaria de la coincidencia
<i>Usuario</i>	VARCHAR	Usuario que realiza la consulta, viene de SegUsuarios
<i>Licencia</i>	VARCHAR	Licencia con la que se ejecutó la consulta.

LogEventos: Representa la estructura de los tipos de eventos admitidos en LogRegistro.

Tabla 17. Descripción de la tabla LogEventos

Campo	Tipo de Datos	Descripción
<i>IdEvento</i>	Int	Identifica de manera única el evento
<i>Evento</i>	VARCHAR	Nombre corto del evento
<i>Descripcion</i>	VARCHAR	Información descriptiva del evento.

8.11.3 Vistas

VTopBlsData: Unifica las tablas de TopBlsData en un formato compatible con LogConsulta

Tabla 18. Descripción de la vista VTopBlsData

Campo	Tipo de Datos	Descripción
<i>CodigoLista</i>	Varchar	Código que aparece en la lista en la que se encontró la coincidencia
<i>SubCodigoLista</i>	Varchar	Código complementario que aparece en la lista en la que se encontró la coincidencia.
<i>NombreLista</i>	Varchar	Nombre de la lista en la que se encontró
<i>Nombre</i>	Varchar	Nombre de la coincidencia
<i>FechaActualizacion</i>	DateTime	Fecha en la que fue actualizado el registro por la fuente original.
<i>Observaciones</i>	Varchar	Información complementaria de la coincidencia

8.11.4 Procedimientos Almacenados

SP_CONSULTALISTAS: Realiza el proceso de consulta sobre vtopbldata, retorna los resultados de la consulta y genera un registro en el log de consulta.

Tabla 19. Descripción del Procedimiento Almacenado SP_CONSULTALISTAS

Parámetro	Tipo de Datos	Descripción
<i>@ID</i>	Varchar	Identificación que se va a buscar
<i>@Nombre</i>	Varchar	Nombre que se va a buscar

<i>@Nivel</i>	Int	Nivel de coincidencia. Valores admitidos : <ul style="list-style-type: none">• 0 – Coincidencias parciales• 100 – Coincidencias exactas
<i>@Usuario</i>	Varchar	Usuario que realiza la consulta
<i>@Licencia</i>	Varchar	Licencia con la que se ejecutó la consulta.

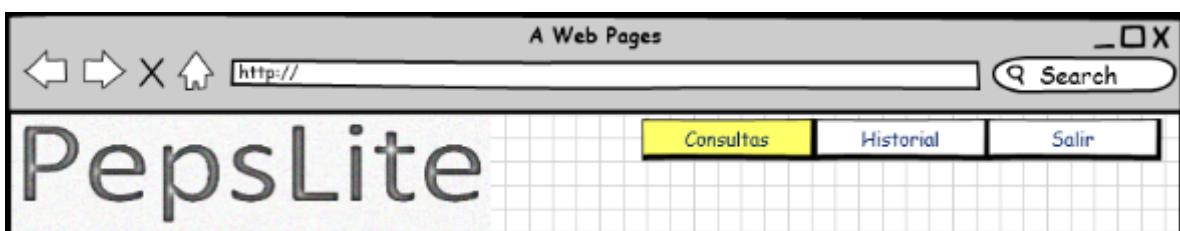
8.12 INTERFAZ DE USUARIO – SPRINT 1

La interfaz de usuario es el medio de comunicación entre el sistema informático y el usuario humano, contiene el esquema de las pantallas o formularios, links, botones, captura de datos etc.

8.12.1 Menú de navegación

Contiene elementos en la parte superior y que se mostrarán y comportarán de la misma manera en todas las páginas de la aplicación.

Gráfico 53. Menú de Navegación



Logotipo: Imagen del producto, no realiza ninguna acción.

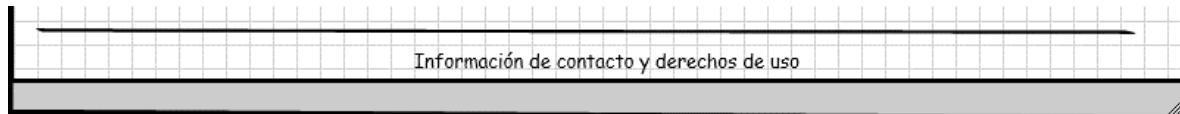
Menú superior: Es el menú de navegación, contiene 3 botones.

- **Consultas:** Lleva a la pantalla de consultas sobre listas.
- **Historial:** Redirecciona a la pantalla de historial de consultas.
- **Salir:** Cierra la sesión y redirecciona a la pantalla de inicio de sesión.

8.12.2 Pie de página

Contiene elementos en la parte inferior que se mostrarán y comportarán de la misma manera en todas las páginas de la aplicación.

Gráfico 54. Pie de Página



Información de contacto y derechos de uso: Contiene información como correo de contacto del administrador, derechos de uso de la aplicación e información del propietario de la licencia.

8.12.3 Formulario de Consultas

Gráfico 55. Formulario de Consultas

The screenshot shows a web browser window titled 'A Web Pages' with the URL 'http://'. The title bar says 'PepsLite'. The menu bar includes 'Consultas', 'Historial', and 'Salir'. Below the title, there are input fields for 'Identificación' and 'Nombre', and a 'Consultar' button. A section titled 'Resultados' displays a table with four rows of data. The columns are labeled '#', 'Lista', 'Identificación', 'Nombre', 'Observaciones', and 'Reporte'. Each row has a checked checkbox in the 'Reporte' column. The data in the table is as follows:

#	Lista	Identificación	Nombre	Observaciones	Reporte
1	OFAC	A A TRADING FZC	Programa: SDNTK	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
1	OFAC	A.T.F. INTERNAT	Programa: SDNTK	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
1	NACIONES UNID	QDEB3.02.	ZAKARYA EGGAB	Programa: SDNTK	<input checked="" type="checkbox"/>
1	NACIONES UNID	QI15-123.03.	YASSIN SYWAL	Programa: SDNTK	<input checked="" type="checkbox"/>

At the bottom of the page, there is a footer with the text 'Información de contacto y derechos de uso'.

Campos

- **Identificación:** Identificación que se va a buscar en las listas, puede estar vacío
- **Nombre:** Nombre que se va a buscar en las listas, puede estar vacío.
- **Consultar:** Realiza la consulta según los valores ingresados, al menos un campo debe ser diligenciado para realizar la consulta

Tabla de Resultados

Contiene la lista de resultados de la consulta, las columnas retornadas son:

- #: Enumera los resultados para su identificación en pantalla.
- **Lista:** Informa en qué lista fue encontrada la coincidencia.
- **Identificación:** Identificación de la coincidencia.
- **Nombre:** Nombre de la coincidencia.
- **Observaciones:** Información adicional del registro encontrado.
- **Reporte:** Abre una ventana con el reporte que muestra la información completa del registro.

8.12.4 Formulario de Historial de Consultas

Gráfico 56. Formulario de Historial de Consultas

#	Fecha	Usuario	Id Buscado	Nombre Buscado	Resultados	Reporte
1	2014/04/01	sbustos		A A TRADING	1	<input checked="" type="checkbox"/>
2	2014/04/01	sbustos		A.T.E. INTERN	4	<input checked="" type="checkbox"/>
3	2014/04/02	egonzalez	QI.C.03.02.		1	<input checked="" type="checkbox"/>
4	2014/04/03	sbustos		YASSIN SYWA	3	<input checked="" type="checkbox"/>

Información de contacto, fecha y dirección de uso

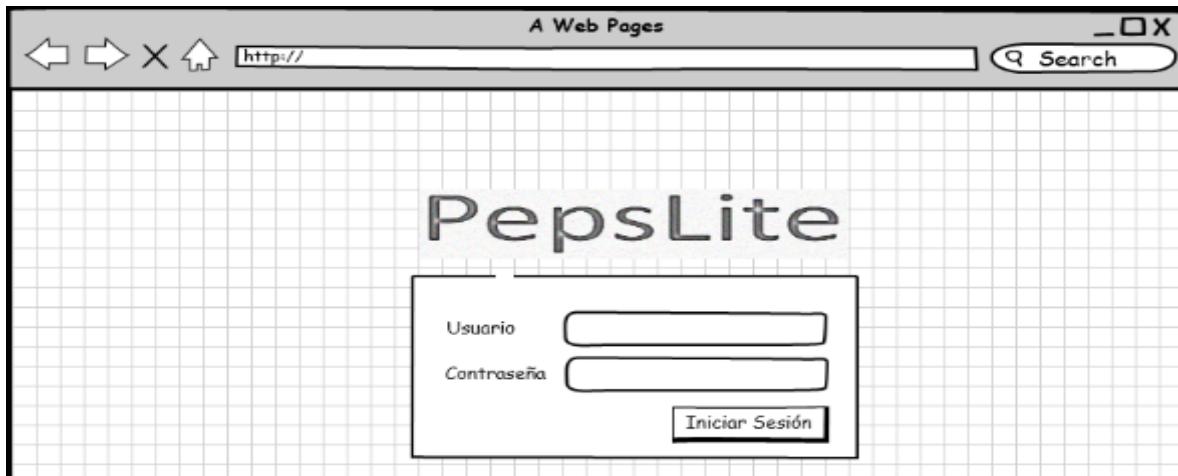
Tabla de Resultados

Contiene la lista de resultados de la consulta, las columnas retornadas son:

- #: Enumera los resultados para su identificación en pantalla.
- **Fecha:** Fecha en que se realizó la consulta.
- **Usuario:** Usuario que realizó la consulta.
- **Id Buscado:** Identificación que fue buscada.
- **Nombre Buscado:** Nombre que fue buscado.
- **Resultados:** Cantidad de registros encontrados.
- **Reporte:** Abre una ventana con el reporte que muestra la información completa de la búsqueda.

8.12.5 Formulario de Inicio de sesión

Gráfico 57. Formulario de Inicio de Sesión



Campos

- **Usuario:** Nombre de usuario usado para el inicio de sesión
- **Contraseña:** Frase secreta.
- **Iniciar Sesión:** Verifica si la información de inicio de sesión es correcta, si es así, permite el acceso, de lo contrario lo impide.

8.13 REUNIÓN DE PLANIFICACIÓN DEL SPRINT 3 – 2014/05/03

El objetivo de la reunión es identificar los objetivos que deberán cumplirse en la iteración y establecer un procedimiento para llevarlos a cabo.

8.13.1 Asistentes

- Ernesto González – Product Owner
- Sergio Bustos – Equipo de Desarrollo
- Alfonso Quintero – Scrum Master

8.13.2 ¿Qué se completará en este Sprint? (Sprint Backlog)

En el Tercer Sprint, se realizará la implementación de la validación de la licencia y registro de actividades, el detalle de estas actividades es:

- Creación del algoritmo de encriptación y des-encriptación para la generación de las licencias.
- Creación de la aplicación de generación de licencias
- Creación de las capas de Datos, Lógica de Negocio y Presentación, para la administración de los usuarios.
- Implementación de la validación de usuarios y licencias.
- Creación de las capas de Datos, Lógica de Negocio e inclusión dentro de la capa de presentación del registro de actividades.

8.13.3 ¿Cómo se conseguirá completar el trabajo seleccionado?

En el Sprint Diario, el equipo estimará las tareas que pueden ser completadas durante un día, y no dejará tareas pendientes de un día para otro, si necesita iniciar una tarea poco antes de finalizar el día, deberá posponer su inicio hasta el día siguiente, si una tarea dura más de un día, entonces deberá dividirse en tareas más cortas. El Product Owner ha ofrecido literatura relacionada a la creación de los diagramas que se usará como base, así como la literatura provista por la UNAD.

8.13.4 Acta de la Reunión

Gráfico 58. Acta de la reunión Sprint 3



8.14 INTERFAZ DE USUARIO – Sprint 3

La interfaz de usuario es el medio de comunicación entre el sistema informático y el usuario humano, contiene el esquema de las pantallas o formularios, links, botones, captura de datos en esta etapa del proyecto se incluyen las dos últimas pantallas.

8.14.1 Formulario de Administración de Usuarios (Consulta)

Gráfico 59. Administración de Usuarios - Consulta

Usuario	Nombre	Cargo	Teléfono	Correo	Activo	Admin	Agregar
sa	Administrador del Sistema	Administrador	3173897654	admin@pepslite.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Editor
sbustos	Sergio Bustos	Desarrollador		sbustos@pepslite.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Editor
egonzalez	Ernesto González	Product Owner		admin@pepslite.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Editor
harenas	Hernando Arenas	Scrum Master		admin@pepslite.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Editor

Tabla de Resultados

Contiene la lista de resultados de la consulta, las columnas retornadas son:

- **Usuario:** Nombre de identificación del usuario.
- **Nombre:** Nombre completo o descripción del usuario.
- **Cargo:** Cargo que ocupa el responsable del usuario
- **Teléfono:** Teléfono de contacto.
- **Correo:** Correo electrónico de contacto.
- **Activo:** Estado de actividad del usuario, Activo puede ingresar al sistema, de lo contrario el acceso es denegado.
- **Admin:** Rol del usuario. Cuando está marcado el usuario puede crear y editar otros usuarios.

8.14.2 Formulario de Administración de Usuarios (Mantenimiento)

Gráfico 60. Administración de Usuarios - Mantenimiento

The screenshot shows a web browser window titled 'A Web Pages'. The address bar displays 'http://'. The main content area features the 'PepsLite' logo at the top left. A navigation menu at the top right includes 'Consultas', 'Historial', a yellow-highlighted 'Usuarios' button, and 'Salir'. Below the menu, a large text input field contains the word 'PepsLite'. A central form box is titled 'Mantenimiento de Usuarios'. It contains fields for 'Usuario', 'Contraseña', 'Nombre', 'Cargo', 'Teléfono', 'Correo', 'Activo' (with a checked checkbox), and 'Admin' (with an unchecked checkbox). A 'Guardar' button is located at the bottom of the form. At the very bottom of the page, there is a link labeled 'Información de contacto y derechos de uso'.

8.14.3 Formulario creación de licencias

Gráfico 61. Generador de Licencias

The screenshot shows a window titled 'Wi'. The main area contains two text input fields: 'Autorizado a:' and 'Código de Licencia'. Below these fields are two buttons: 'Generar' on the left and 'Salir' on the right. The entire application has a simple, monochromatic design.

8.14.4 Método de Generación de las Licencias

El programa de generación de licencias utiliza el algoritmo Rijndael¹⁰⁴ con un cifrado de 256 Bits, al cual se le ha indicado la opción de llenar con 0 los bytes restantes si llegara a necesitarse, el procedimiento de generación de la licencia, consiste simplemente en encriptar un texto, se recomienda el nombre de la entidad o individuo autorizado y si la desencripción de este texto es correcta se interpreta que la licencia es válida.

8.14.5 Algoritmo Rijndael

Rijndael es el algoritmo de cifrado de bloque elegido por el Instituto Nacional de Ciencia y Tecnología de los Estados Unidos de América, (NIST) como el estándar de encripción avanzada (AES), Rijndael es un cifrado de bloques iterada. Por lo tanto, el cifrado o descifrado de un bloque de datos se lleva a cabo por la iteración (una ronda) de una transformación específica (una función de clasificación).¹⁰⁵

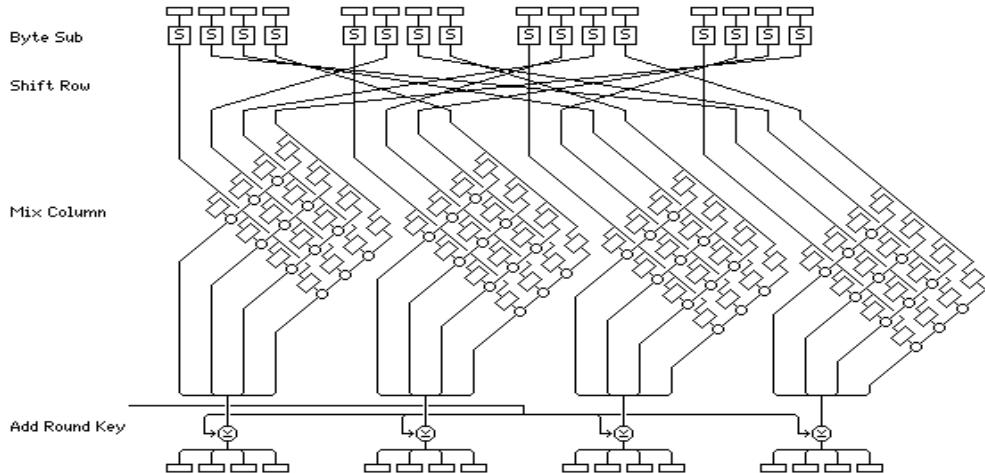
Rijndael también define un método para generar una serie de subclaves de la llave original. Las subclaves generadas se utilizan como entrada con la función de la iteración.¹⁰⁶

¹⁰⁴ DAEMEN, Joan y RIJMEN, Vincent, AES Proposal: Rijndael. 2003, Belgium. Disponible en <http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf>

¹⁰⁵ McGrill, School of Computer Science, 2000. Disponible en http://www.cs.mcgill.ca/~kaleigh/computers/crypto_rijndael.html

¹⁰⁶ Ibíd.

Gráfico 62. Transformación de la Iteración del algoritmo Rijndael.



Tomado de http://www.cs.mcgill.ca/~kaleigh/computers/crypto_rijndael.html

8.15 REUNIÓN DE PLANIFICACIÓN DEL SPRINT 4 – 2014/05/21

El objetivo de la reunión es identificar los objetivos que deberán cumplirse en la iteración y establecer un procedimiento para llevarlos a cabo.

8.15.1 Asistentes

- Ernesto González – Product Owner
- Sergio Bustos – Equipo de Desarrollo
- Alfonso Quintero – Scrum Master

8.15.2 ¿Qué se completará en este Sprint? (Sprint Backlog)

En el Cuarto Sprint, se realizará el plan de pruebas, implementación, documentación final, el detalle de estas actividades es:

- Creación y ejecución de un proyecto de pruebas en Visual Studio 2012
- Creación de los diagramas de implementación.
- Implementación en Servidor Web temporal
- Revisión del código y complementación de los comentarios de código.
- Creación del Manual Técnico.
- Creación del Manual de Usuario

8.15.3 ¿Cómo se conseguirá completar el trabajo seleccionado?

En el Sprint Diario, el equipo estimará las tareas que pueden ser completadas durante un día, y no dejará tareas pendientes de un día para otro, si necesita iniciar una tarea poco antes de finalizar el día, deberá posponer su inicio hasta el día siguiente, si una tarea dura más de un día, entonces deberá dividirse en tareas más cortas. El Product Owner ha ofrecido literatura relacionada a la creación de los diagramas que se usará como base, así como la literatura provista por la UNAD.

8.15.4 Acta de la Reunión

Gráfico 63. Acta de la reunión Sprint 4



9. PRUEBAS (TESTING)

La quinta fase del ciclo de vida de vida del software¹⁰⁷, se trata de las pruebas, esta fase pretende incrementar el nivel de calidad y reducir la cantidad de fallos que se puedan encontrar en las aplicaciones, una vez que se haya hecho entrega del producto final.

9.1 Configuración De Las Pruebas (Testing)

Para la realización de las pruebas de este proyecto se creó un Proyecto de Pruebas de Microsoft Visual Studio 2012, el esquema general de las pruebas posee las siguientes características:

- Un archivo por cada clase que se va a probar.
- Un método de prueba por cada método que se va a probar.
- El nombre del archivo de prueba será el mismo que el archivo de la clase a probar, seguido de la palabra “Test” ej. “ClaseTest.cs”
- El nombre del método de prueba será el mismo que el archivo del método a probar seguido de la palabra “Test” ej. “MetodoTest.cs”

¹⁰⁷ PRESSMAN, Roger. Op. Cit

- Si un método tiene varias sobrecargas, el método de prueba terminará en el número siguiente al anterior iniciando por el número 1, si no tiene sobrecargas el método de prueba no llevará ningún número.

Tabla 20. Características de las pruebas

<i>Clase de Prueba</i>	<i>Método de prueba</i>	<i>Características y resultados de la prueba</i>
<i>AuditoriaBLTest.cs</i>	InsertarTest()	Enviar un objeto de tipo LogRegistroE, la prueba será exitosa si el resultado es “true”
<i>AuditoriaDALTest.cs</i>	InsertarTest()	Enviar un objeto de tipo LogRegistroE, la prueba será exitosa si el resultado es “true”
<i>AutenticacionTest.cs</i>	AutenticarTest()	Enviar un usuario registrado (sa) con su contraseña, la prueba será exitosa si el método retorna un valor no null y de tipo SegUsuariosE
<i>ConexionSqlTest.cs</i>	ConexionSqlConstructorTest()	Crear un objeto de tipo ConexionSql, la prueba será exitosa si retorna un objeto de tipo ConexionSql
	AbrirConexionTest()	Se ejecuta el método AbrirConexion(), la prueba será exitosa si retorna un objeto de tipo ConexionSql
	EjecutarCommandoTest()	Se ejecuta un comando sql que actualiza un registro en la base de datos, la prueba será exitosa si retorna 1.
	EjecutarReaderTest()	Se ejecuta un comando sql de lectura de registros, la prueba será exitosa si retorna un objeto diferente de null
<i>ConsultasBLTest.cs</i>	ConsultaExactaTest()	Se consulta un individuo existente en las listas, con un usuario y licencia válidos, la prueba será exitosa si retorna un objeto diferente no null y de tipo ResConsultaE

	ConsultaParcialTest()	Se consulta un individuo existente en las listas, con un usuario y licencia válidos, la prueba será exitosa si retorna un objeto diferente no null y de tipo IList<ResConsultaE> con al menos un registro.
	ConsultarDetalleTest()	Se consulta un individuo existente en las listas, con un usuario y licencia válidos, la prueba será exitosa si retorna un objeto diferente no null y de tipo IList<ResConsultaE> con al menos un registro.
<i>ConsultasDALTest.cs</i>	ConsultasDALConstructorTest()	Crea una instancia de la clase, la prueba será exitosa si retorna un objeto de tipo ConsultasDAL
	ConsultaExactaTest()	Se consulta un individuo existente en las listas, con un usuario y licencia válidos, la prueba será exitosa si retorna un objeto diferente no null y de tipo ResConsultaE
<i>CriptografiaTest.cs</i>	ConsultaParcialTest()	Se consulta un individuo existente en las listas, con un usuario y licencia válidos, la prueba será exitosa si retorna un objeto diferente no null y de tipo IList<ResConsultaE> con al menos un registro.
	ConsultarDetalleTest()	Se consulta un individuo existente en las listas, con un usuario y licencia válidos, la prueba será exitosa si retorna un objeto diferente no null y de tipo IList<ResConsultaE> con al menos un registro.
	DesEncriptarTest()	Se envía una cadena encriptada válida, la prueba será exitosa si el valor desencriptado es igual a la cadena de referencia
	GeneradorLicenciasEncriptarTest()	Se envía una cadena encriptada válida, la prueba será exitosa si el valor desencriptado es igual a la cadena de referencia

	EncriptarTest()	Se envía una cadena sin encriptar, la prueba será exitosa si el valor encriptado es igual a la cadena encriptada de referencia
<i>HistoriaBLTest.cs</i>	ConsultarTest()	Se envía un IdConsulta existente en la base de datos, la prueba será exitosa si retorna un objeto no null de tipo IList<LogConsultaE> y contiene al menos un registro.
	ConsultarTest1()	Se consultan los registros de la base de datos, la prueba será exitosa si retorna un objeto no null de tipo IList<LogConsultaE> y contiene al menos un registro.
<i>HistoriaDALTest.cs</i>	HistoriaDALConstructorTest()	Se crea una instancia de la clase, la prueba será exitosa si retorna un objeto de tipo HistoriaDAL
	ConsultarTest()	Se envía un IdConsulta existente en la base de datos, la prueba será exitosa si retorna un objeto no null de tipo IList<LogConsultaE> y contiene al menos un registro.
	ConsultarTest1()	Se consultan los registros de la base de datos, la prueba será exitosa si retorna un objeto no null de tipo IList<LogConsultaE> y contiene al menos un registro.
<i>SegUsuariosBLTest.cs</i>	ActualizarTest()	Se envía un objeto de tipo SegUsuariosE, la prueba será exitosa si retorna true.
	ConsultarTest()	Se llama el método, la prueba será exitosa si retorna un objeto no null de tipo SegUsuariosE
	ConsultarTest1()	Se envía al método un usuario válido, la prueba será exitosa si retorna un objeto no null de tipo SegUsuariosE
	InsertarTest()	Se envía un objeto de tipo SegUsuariosE, la prueba será exitosa si retorna true.

<i>SegUsuariosDALTest.cs</i>	SegUsuariosDALConstructorTest()	Se crea una instancia de la clase, la prueba será exitosa si retorna un objeto de tipo SegUsuariosDAL
	ActualizarTest()	Se envía un objeto de tipo SegUsuariosE, la prueba será exitosa si retorna true.
	ConsultarTest()	Se llama el método, la prueba será exitosa si retorna un objeto no null de tipo SegUsuariosE
	ConsultarTest1()	Se envía al método un usuario válido, la prueba será exitosa si retorna un objeto no null de tipo SegUsuariosE
	InsertarTest()	Se envía un objeto de tipo SegUsuariosE, la prueba será exitosa si retorna true.
<i>WsConsultaListasTest.cs</i>	WsConsultaListasConstructorTest()	Se crea una instancia de la clase, la prueba será exitosa si retorna un objeto de tipo WsConsultaListas
	ConsultarTest()	Se consulta un individuo existente en las listas, con un usuario y licencia válidos, la prueba será exitosa si retorna un objeto diferente no null y de tipo ResConsultaE

Se ha habilitado de manera temporal una dirección Web para la prueba del Servicio Web, esta dirección puede ser consultada desde dispositivos móviles.

<http://190.26.67.236/PepsLiteMobileTest/login.aspx>

9.2 Aprobación De Las Pruebas

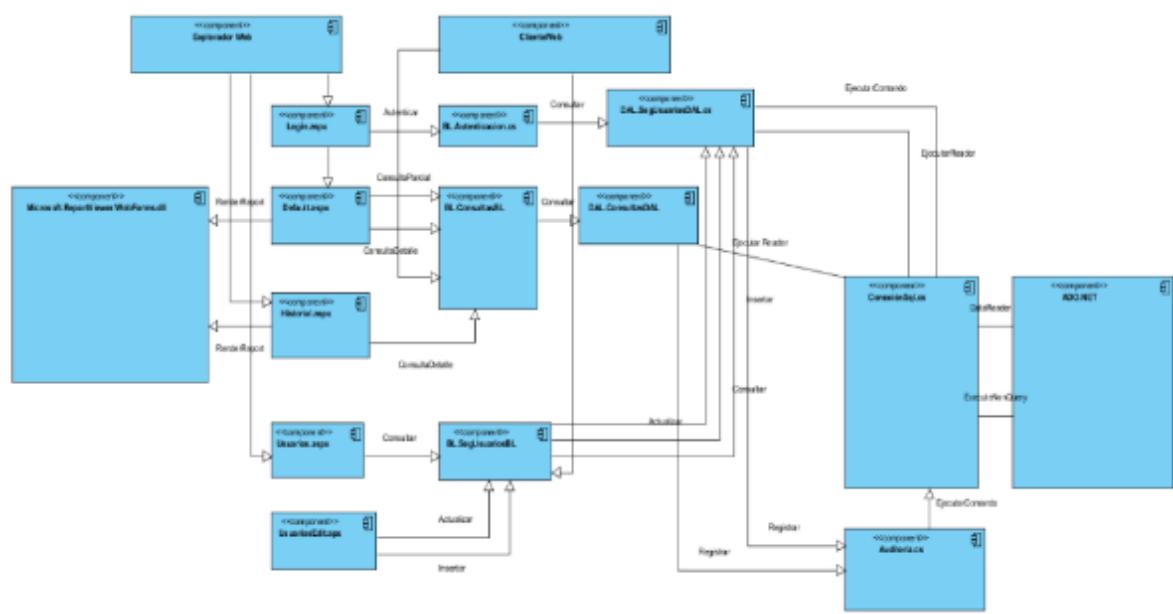
Gráfico 64. Resultados de la ejecución de las pruebas.

Explorador de pruebas	
Buscar	
Vídeo de transmisión por secuencias: Mejora de la calidad con pruebas unitarias y emulaciones	
Ejecutar todas	Ejecutar...
Lista de reproducción: Todas las pruebas ▾	
■ AuditoriaBLTest (1)	
✓ InsertarTest	121 ms
■ AuditoriaDALTest (1)	
✓ InsertarTest	12 ms
■ AutenticacionTest (1)	
✓ AutenticarTest	9 ms
■ ConexionSqlTest (4)	
✓ AbrirConexionTest	28 ms
✓ ConexionSqlConstructorTest	< 1 ms
✓ EjecutarCommandoTest	1 ms
✓ EjecutarReaderTest	< 1 ms
■ ConsultasBLTest (3)	
✓ ConsultaExactaTest	188 ms
✓ ConsultaParcialTest	232 ms
✓ ConsultarDetalleTest	3 ms
■ ConsultasDALTest (4)	
✓ ConsultaExactaTest	188 ms
✓ ConsultaParcialTest	259 ms
✓ ConsultarDetalleTest	2 ms
✓ ConsultasDALConstructorTest	< 1 ms
■ CriptografiaTest (3)	
✓ DesEncriptarTest	< 1 ms
✓ EncriptarTest	< 1 ms
✓ GeneradorLicenciasEncriptarTest	39 ms
■ HistoriaBLTest (2)	
✓ ConsultarTest	6 ms
✓ ConsultarTest1	8 ms
■ HistoriaDALTest (3)	
✓ ConsultarTest	2 ms
✓ ConsultarTest1	2 ms
✓ HistoriaDALConstructorTest	< 1 ms
■ SegUsuariosBLTest (4)	
✓ ActualizarTest	6 ms
✓ ConsultarTest	3 ms
✓ ConsultarTest1	6 ms
✓ InsertarTest	52 ms
■ SegUsuariosDALTest (5)	
✓ InsertarTest	209 ms
✓ ActualizarTest	2 ms
✓ ConsultarTest	8 ms
✓ ConsultarTest1	2 ms
✓ SegUsuariosDALConstructorTest	< 1 ms
■ WsConsultaListasTest (2)	
✓ ConsultarTest	215 ms

10. IMPLEMENTACIÓN

10.1 Diagrama de Componentes

Gráfico 65. Diagrama de Componentes



Explorador Web: Programa que permite navegar a través de sitios y aplicaciones Web a través del protocolo Http.

Cliente Web: Programa que permite realizar llamados a Servicios Web por medio del protocolo Http y SOAP.

Microsoft.ReportViewer.WebForms.dll: Ensamblado binario que contiene las clases y métodos necesarios para crear reportes en formato Pdf.

Login.aspx: Página de inicio de sesión, controla el acceso al Sistema

Default.aspx: Página de entrada a la aplicación, permite realizar las consultas individuales y exportar el reporte de coincidencias.

Historial.aspx: Página de consulta de la historia de las consultas, también permite exportar el reporte de las consultas realizadas y coincidencias encontradas.

Usuarios.aspx: Página de consulta de los usuarios del Sistema.

UsuariosEdit.aspx: Página de mantenimiento de los usuarios del Sistema, permite crear o modificar usuarios.

Autenticacion.cs: Archivo de código fuente que realiza las validaciones de seguridad de acceso al Sistema.

ConsultasBL.cs: Archivo de código fuente que permite preparar la consulta y los resultados de las coincidencias.

SegUsuariosBL.cs: Archivo de código fuente que permite preparar la consulta y mantenimiento de los usuarios del Sistema.

SegConsultasDAL.cs: Archivo de código fuente que permite generar las instrucciones SQL, para realizar las consultas y obtener los resultados.

SegUsuariosDAL.cs: Archivo de código fuente que permite generar las instrucciones SQL, para realizar las consultas y mantenimiento de los usuarios del Sistema.

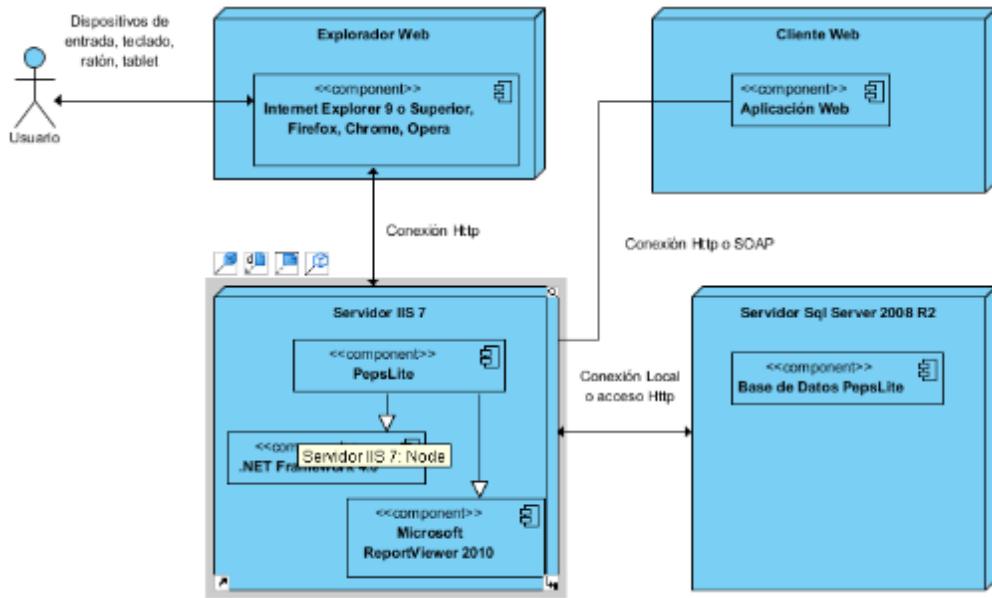
ConexionSql.cs: Archivo de código fuente que gestiona la conexión con la Base de Datos, ejecuta comandos y retorna los resultados.

Auditoria.cs: Archivo de código fuente que permite generar las instrucciones SQL, para registrar los eventos del Sistema.

ADO.NET: Plataforma perteneciente a .NET Framework, que permite vincular a las aplicaciones con las bases de datos.

10.2 Diagrama de Despliegue

Gráfico 66. Diagrama de Despliegue



Usuario: Persona que realiza la interacción con el sistema.

Explorador Web: Aplicación que permite navegar a través de sitios y aplicaciones Web.

Servidor IIS 7: Servidor de aplicaciones Microsoft Internet Information Services 7.0, habilitado para ejecutar aplicaciones basadas en .NET Framework 4.0, con Microsoft Report Viewer 2010 Instalado.

Cliente Web: Aplicación que permite consumir el Servicio Web expuesto en el Servidor IIS 7, por medio de los protocolos SOAP y Http.

Servidor Sql Server 2008R2: Servidor de Bases de Datos Sql Server 2008 R2, con la base de datos PepsLite o TopBLS con los objetos de PepsLite.

10.3 Manual Técnico

Con este documento se entrega el Manual Técnico, el cual explica la estructura de las aplicaciones, los requisitos, la guía de instalación, desinstalación y especificaciones del Servicio Web, Base de datos y Auditoría.

El archivo se entrega en el paquete de instalación, bajo el nombre “Manual Técnico PepsLite.pdf”

10.4 Manual de Usuario

De la misma forma que el Manual de Técnico, se hace entrega de un Manual de Usuario, que contiene las indicaciones para el uso de las aplicaciones y el Servicio Web.

El archivo se entrega en el paquete de instalación, bajo el nombre “Manual de Usuario PepsLite.pdf”

11. ANEXOS

Anexo 1. Acta Sprint 1

Bogotá, 01 de Abril de 2014



Ref: Acta de Reunión

Asistentes:

Ernesto González R.
Alfonso Quintero.
Sergio I. Bustos

En la reunión se llega al acuerdo que los siguientes elementos deberán ser terminados entre el 1 de Abril y el 15 de Abril de 2014, cada día se hará una pequeña reunión de 10 minutos para revisar los problemas en el avance.

Diagramas de Casos de Uso

Diagramas de Clases para el análisis

Diagramas de Secuencias del Sistema para el análisis

Definición de los requerimientos funcionales

Definición de los requerimientos no funcionales

Diagramas de Secuencias para el Diseño

Diagramas de Clases para el Diseño

Diagramas de Actividades

Diagramas de Estado

Teléfonos: (571) 214 0329 - 214 0491
Dirección: Cra. 15A N° 120 - 42 Of. 401
Bogotá - Colombia
e-mail: info@sidifltda.com
www.sidifltda.com

Anexo 2. Acta Sprint 2

Bogotá, 17 de Abril de 2014



Ref: Acta de Reunión

Asistentes:

Ernesto González R.
Alfonso Quintero.
Sergio I. Bustos

En esta reunión se llega al acuerdo, que los siguientes elementos deberán ser terminados entre el 17 de Abril y e 2 de Mayo de 2014, cada día se realizará una pequeña reunión de 10 minutos para revisar los problemas en en el avance.

- Diseño físico de los objetos del sistema (Diagrama Entidad-Relación)
- Creación de la base de datos
- Importación de las tablas de TopBIs
- Creación de las Tablas nuevas
- Creación de la vista de integración
- Creación del procedimiento almacenado de consulta
- Creación de las capas de Datos, Lógica de Negocio y Presentación
- Pruebas de consulta.
- Creación del reporte de coincidencias

A handwritten signature in black ink, appearing to read "Ernesto González R.". It is written in a cursive script.

Teléfonos: (571) 214 0329 - 214 0491
Dirección: Cra. 15A N° 120 - 42 Of. 401
Bogotá - Colombia
e-mail: info@sidifltda.com
www.sidifltda.com

A handwritten signature in black ink, appearing to read "Alfonso Quintero". It is written in a cursive script.

A handwritten signature in black ink, appearing to read "Sergio I. Bustos". It is written in a cursive script.

Anexo 3. Acta Sprint 3

Bogotá, 03 de Mayo de 2014



Ref: Acta de Reunión

Asistentes:

Ernesto González R.
Alfonso Quintero.
Sergio I. Bustos

En esta reunión se llega al acuerdo, que las siguientes tareas deberán llevarse a cabo entre el 3 de Mayo y e 20 de Mayo de 2014, cada día se realizará una pequeña reunión de 10 minutos para revisar los problemas en su avance.

- Creación del algoritmo de encriptación y des-encriptación para la generación de las licencias.
- Creación de la aplicación de generación de licencias
- Creación de las capas de Datos, Lógica de Negocio y Presentación, para la administración de los usuarios.
- Implementación de la validación de usuarios y licencias.
- Creación de las capas de Datos, Lógica de Negocio e inclusión dentro de la capa de presentación del registro de actividades.
- Creación de las capas de Datos, Lógica de Negocio y Presentación, para la consulta del registro de actividades.

A handwritten signature in black ink, appearing to read "Ernesto González R.". It is written in a cursive, fluid style.

A handwritten signature in black ink, appearing to read "Alfonso Quintero.". It is written in a cursive, fluid style.

A handwritten signature in black ink, appearing to read "Sergio I. Bustos". It is written in a cursive, fluid style.

Teléfonos: (571) 214 0329 - 214 0491
Dirección: Cra. 15A N° 120 - 42 Of. 401
Bogotá - Colombia
e-mail: info@sidifltda.com
www.sidifltda.com

Anexo 4. Acta Sprint 4

Bogotá, 21 de Mayo de 2014



Ref. Acta de Reunión

Asistentes:

Ernesto González R.
Alfonso Quintero.
Sergio I. Bustos

En la reunión se llega al acuerdo que los elementos siguientes deberán ser terminados entre el 21 de Mayo y el 5 de Junio de 2014, cada día se realizará una pequeña reunión de 10 minutos para revisar los problemas en el avance.

- Creación y ejecución de un proyecto de pruebas en Visual Studio 2012
- Creación de la aplicación de generación de licencias
- Creación de los diagramas de implementación.
- Implementación en Servidor Web temporal
- Revisión del código y complementación de los comentarios de código.
- Creación del Manual Técnico.
- Creación del Manual de Usuario

A handwritten signature in black ink, appearing to read "Ernesto González R.". It is written in a cursive script.

A handwritten signature in black ink, appearing to read "Alfonso Quintero". It is written in a cursive script.

A handwritten signature in black ink, appearing to read "Sergio I. Bustos". It is written in a cursive script.

Teléfonos: (571) 214 0329 - 214 0491
Dirección: Cra. 15A N° 120 - 42 Of. 401
Bogotá - Colombia
e-mail: info@sidifltda.com
www.sidifltda.com

CONCLUSIONES

Con la metodología SCRUM sí es posible realizar un proyecto de corta duración y baja complejidad, pues permite realizar entregas rápidas y programadas incluyendo la etapa de análisis. La metodología hace que el proceso de diseño y desarrollo sea muy flexible ya que existe comunicación continua con el dueño del producto (Product Owner) y esto permite renovar periódicamente las expectativas del cliente, de manera que no sean muy bajas o muy altas y sí, acordes al producto que se está desarrollando, adicionalmente la necesidad del equipo de desarrollo de autogestionarse y autoorganizarse, permite que en algunos casos, miembros del equipo opten por realizar modificaciones al diseño o al software no programadas, quebrando el esquema planteado por Scrum

El desarrollo de una aplicación para la consulta de individuos y entidad en las listas ONU y OFAC, permite a las empresas tener un mayor conocimiento del cliente, y la integración del servicio web hace que esta tarea se pueda automatizar e integrar con sus plataformas de negocio, facilitando el cumplimiento de las disposiciones y recomendaciones de los organismos de control.

BIBLIOGRAFÍA

BONILLA HERNÁNDEZ. Análisis de Sistemas [En Línea] (2010). [Consultado 2 Marzo 2014] Disponible en <<http://bonillahernandez.blogspot.com.es/2010/10/analisis-de-sistemas-actividad-3.html>>

BOOCH, Grady, RUMBAUGH, James y JACOBSON Ivar. The Unified Modeling Language User Guide. Un enfoque práctico. Segunda edición. Addison-Wesley Professional. 2005.

BRIGHT HUB PM. Understanding Scrum - Sample Product Backlog [En Línea] (2011). [Consultado 10 Marzo 2014] Disponible en <<http://www.brighthubpm.com/certification/65553-understanding-scrum-sample-product-backlog/>>

CASTILLO, Carlos. Notas de Consultoría. Diseño físico y lógico de una solución tecnológica. [En Línea] (2012) [Consultado 1 Abril 2014] Disponible en <<http://ingcarloscastillo.blogspot.com/2012/09/diseno-fisico-y-logico-de-una-solucion.html>>

INGENIERIA DE SOFTWARE TDEA. Ciclo De Vida Orientado A Objetos. [En Línea] [Consultado 23 Marzo 2014] Disponible en <<http://ingenieriaedesoftwaretdea.weebly.com/ciclo-de-vida-orientado-a-objetos.html>>

INSTITUTO TECNOLÓGICO DE SALINA CRUZ. DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA. Programación orientada a Objetos y modelado [En Línea] (2012). [Consultado 11 Marzo 2014] Disponible en <<http://noesanchez13.files.wordpress.com/2012/11/elementos-primordiales-en-el-modelo-orientado-a-objetos.pdf>>

ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models 2011.

ISSUU. Las pruebas de software durante el ciclo de vida de desarrollo [En Línea] (2009). [Consultado 12 Marzo 2014] Disponible en <http://issuu.com/arlo_critico/docs/pruebadesoftwareciclodvida/>

LA GÜEB DE JOAQUÍN. Apuntes Tácticos – UML. [En Línea] (2003) [Consultado 29 Marzo 2014] Disponible en <<http://jms32.eresmas.net/taticos/UML/UML08/UML0801.html>>

MOUNTAIN GOAT SOFTWARE. Topics in Scrum- Product Backlog Example [En Línea] (2006). [Consultado 9 Marzo 2014] Disponible en <<http://www.mountaingoatsoftware.com/agile/scrum/product-backlog/example>>

PRESSMAN, Roger S. Ingeniería de Software. Un enfoque práctico. Quinta edición. McGraw-Hill. Madrid. 2002.

PROYECTOS DE GUERRILLA. Las cinco etapas de ingeniería del software [En Línea] (2013). [Consultado 15 Marzo 2014] Disponible en <<http://proyectosguerrilla.com/blog/2013/02/las-cinco-etapas-en-la-ingenieria-del-software/>>

PROYECTOSAGILES.ORG. Lista de objetivos / requisitos priorizada (Product Backlog) [En Línea] (2014). [Consultado 2 Marzo 2014] Disponible en <<http://www.proyectosagiles.org/lista-requisitos-priorizada-product-backlog>>

SCHWABER, Ken y SUTHERLAND, Jeff. La Guía de Scrum. La Guía Definitiva de Scrum: Las Reglas del Juego [En Línea] (2013). [Consultado 1 Marzo 2014] Disponible en <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum_Guide%202011%20-%20ES.pdf>

SISTEMAS ISEC ROLANDO. Tecnología y Mecánica. Análisis Y Diseño UML [En Línea] (2012). [Consultado 7 Marzo 2014] Disponible en <<http://sistemas-isec-rolando.blogspot.mx>>

SOMMERVILLE, Ian y SAWYER, Pete. Requirements Engineering: A Good Practice Guide. John Wiley & Sons, Inc. New York, NY, 1997.

UML-BASED WEB ENGINEERING. Research Unit of Programming and Software Engineering. [En Línea] (2013) [Consultado 22 Marzo 2014] Disponible en <<http://uwe.pst.ifi.lmu.de/>>

UNIVERSIDAD DE LAS AMÉRICAS PUEBLA. Ingeniería de Software. Cap. 2. [En Línea] [Consultado 30 Marzo 2014] Disponible en <http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/fuentes_k_jf/capitulo2.pdf>

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA. Ingeniería de Software. [En Línea] (2006). [Consultado 18 Marzo 2014] Disponible en <http://dataoteca.unad.edu.co/contenidos/301404/301404_ContenidoEnLinea/leccin_6_definic_de_ingeniera_de_software.html>

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA. Lenguaje Unificado de Modelado UML. [En Línea] (2006). [Consultado 5 Marzo 2014] Disponible en <<http://dataoteca.unad.edu.co/contenidos/200609/exeuml/>>

UNIVERSIDAD NACIONAL DE COLOMBIA. Análisis y diseño de Algoritmos. [En Línea] [Consultado 21 Marzo 2014] Disponible en <<http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060024/Lecciones/Capitulo%20I/problemas.htm>>

UNIVERSITY OF CANTABRIA. Computers and Real-Time Group. Diagramas de actividad y diagramas de estados [En Línea] [Consultado 25 Marzo 2014] Disponible en <http://www.ctr.unican.es/asignaturas/procodis_3_II/Doc/stateDiagram.pdf>