



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

TESIS

**“APLICACIÓN WEB Y MÓVIL DE CONSULTA DE
PROYECTOS DEL SIRF (SISTEMA INSTITUCIONAL DE
REGISTRO FINANCIERO) PARA EL IINGEN UNAM”**

**QUE PARA OBTENER EL TÍTULO DE
INGENIERA EN COMPUTACIÓN**

PRESENTA

LETICIA MENDOZA ZAMORES

DIRECTOR DE TESIS:

ING. JULIO ALFONSO DE LEÓN RAZO



CIUDAD UNIVERSITARIA, ENERO DEL 2014

AGRADECIMIENTOS

Agradezco a mi papá Benito Mendoza, la persona que más admiro en el mundo, su apoyo y comprensión en mis estudios. Gracias a los valores que me ha inculcado he tomado siempre el mejor camino, el que me ha llevado al éxito.

A mi tía Catalina Mendoza quien ha sido más que una madre para mí, agradezco su apoyo incondicional, sus sabios consejos y su cariño. Ella me ha dado fortaleza para lograr cada uno de mis objetivos y este trabajo es uno de ellos.

A mi tío Irineo Mendoza le doy gracias por ser un segundo padre para mí, por sus enseñanzas y ánimos para salir adelante.

A mis hermanas Diana y Donají, y a mi hermano Daniel, quienes han sido mi motivación, les agradezco por compartir sus alegrías conmigo, por dejarme ser su ejemplo a seguir, por sus lecciones de vida y por todo su apoyo.

A mis maestros, gracias por los conocimientos y el aprendizaje que me transmitieron durante mi formación.

Agradezco a mi asesor el Ing. Julio De León por haberme guiado durante la realización de este proyecto, por sus enseñanzas, su tiempo y por todo el apoyo para llegar a la culminación del mismo.

Al Ing. Marco Ambríz le agradezco el haberme abierto las puertas de la Coordinación de Sistemas de Cómputo del IINGEN, donde conocí a grandes personas que me brindaron su amistad y apoyo.

Al Instituto de Ingeniería, que me dio la oportunidad de participar en este proyecto, brindándome las herramientas necesarias para su realización.

A la Universidad Nacional Autónoma de México, y en especial a la Facultad de Ingeniería: mi Alma Máter, por dejarme ser parte de la comunidad universitaria y por haber sido mi segundo hogar durante mi formación profesional.

A mis amigos y a todas aquellas personas que me brindaron su apoyo y comprensión durante el desarrollo de este proyecto.

La realización de este proyecto fue de gran importancia para mi desarrollo profesional, ya que puse en práctica los conocimientos teóricos y prácticos que obtuve durante la carrera y además, obtuve nuevos conocimientos derivados de la investigación y de algunos cursos sobre el desarrollo de aplicaciones móviles. Es una gran satisfacción haber aportado al Instituto de Ingeniería mi trabajo en este proyecto, lo cual fue posible gracias a las herramientas y el conocimiento que me brindó esta institución.

También me llena de satisfacción corresponder de esta manera al conocimiento que me brindó a lo largo de mi formación la máxima casa de estudios de nuestro país, de la cual estoy totalmente agradecida.

ÍNDICE

Capítulo 1 Introducción	5
Capítulo 2 Antecedentes	9
2.1 El Instituto de Ingeniería	11
2.2 Sistema Institucional de Registro Financiero SIRF	12
2.3 Situación Actual.....	13
2.3.1 Usuarios.....	13
2.3.2 Arquitectura del SIRF	13
2.3.3 Módulos que componen al SIRF	15
2.3.4 Ventajas y Desventajas del SIRF	16
2.4 Bibliografía	16
Capítulo 3 Tópicos Generales	17
3.1 Plataformas de Desarrollo de Software.....	19
3.1.1 Móvil	19
3.1.2 Web.....	22
3.1.3 Consola.....	23
3.1.4 Escritorio	24
3.2 Ciclo de Vida del Software	25
3.3 Metodologías de desarrollo de Software	27
3.3.1 RUP.....	28
3.3.2 Ágil.....	29
3.3.2.1 SCRUM.....	31
3.4 Calidad del Software.....	33
3.5 Pruebas de Software	35
3.6 Arquitecturas.....	37
3.6.1 Arquitectura Cliente – Servidor	37
3.6.2 Arquitectura orientada a servicios	39
3.7 Seguridad.....	40
3.8 Bibliografía	43
Capítulo 4 Análisis de la aplicación	45
4.1 Descripción del problema.....	47

4.2	Propuesta de solución	47
4.2.1	Análisis de recursos disponibles	48
4.2.1.1	Recursos humanos	49
4.2.1.2	Hardware.....	49
4.2.1.3	Software	50
4.2.2	Requerimientos	50
4.2.3	Elección de plataformas	53
4.2.4	Planeación y estimación de tiempos y costos.....	54
4.3	Bibliografía	55
Capítulo 5 Diseño de la aplicación.....		57
5.1	Arquitectura Cliente- Servidor.....	59
5.2	Arquitectura Orientada a Servicios (SOA).....	62
5.3	Diagramas UML	63
5.3.1	Casos de Uso	63
5.3.2	Estados	70
5.4	Diseño de la interfaz de la aplicación móvil.....	71
5.4.1	Plataformas	71
5.4.1.1	Móvil Web	71
5.4.1.2	Móvil Nativa	72
5.5	Diseño de la interfaz de la aplicación Web (ASPX).....	73
5.5.1	Para pantalla de conmutador	74
5.5.2	Para pantalla de dispositivo móvil.....	75
5.6	Bibliografía	77
Capítulo 6 Desarrollo de la aplicación.....		79
6.1	Programación de WebServices	81
6.1.1	XML	82
6.1.2	JSON	85
6.2	Programación de la aplicación Móvil.....	86
6.2.1	WindowsPhone	86
6.2.2	Iphone	89
6.3	Programación de la aplicación Web	93
6.3.1	ASPX	94

6.4	Bibliografía	107
Capítulo 7 Pruebas, depuración y liberación de la aplicación		109
7.1	Documentación de las pruebas	111
7.2	Compatibilidad de dispositivos móviles.....	116
7.3	Compatibilidad de navegadores web	117
7.4	Documentos de depuración	118
7.5	Liberación	119
7.6	Bibliografía	121
Capítulo 8 Conclusiones		123
ANEXOS		127
Anexo 1		127
Anexo 2		129
GLOSARIO.....		131

Capítulo 1

Introducción

Las Tecnologías de la información y la comunicación (TIC) han evolucionado constantemente y gracias a ello actualmente es posible tener al alcance grandes cantidades de información, se han automatizado tareas que antes requerían una mayor inversión de tiempo y de recursos y podemos estar comunicados encontrándonos en cualquier parte del mundo. Tan es así, que hemos cambiado nuestra forma de vida en los últimos años con la presencia de las nuevas tecnologías, pues cada vez dependemos más de ellas para realizar distintas actividades, tanto en el sector educativo, laboral y en nuestra vida cotidiana.

Precisamente una de las necesidades de muchas personas en la actualidad es la de estar comunicadas e informadas en cualquier momento y lugar y gracias a las TIC esto es posible, pues con la aparición y evolución de los dispositivos electrónicos portátiles y el internet, podemos consultar la información que necesitemos en cualquier instante.

Las TIC actualmente abarcan una gran gama de dispositivos, entre los cuales encontramos los dispositivos móviles: smartphones (teléfonos inteligentes) y tablets (tabletas). Estos nos permiten entre otras cosas, realizar llamadas, navegar en internet, tomar fotografías, consultar el correo electrónico, y con ayuda de las aplicaciones para estos dispositivos podemos hacer una variedad de tareas como consultar el clima, consultar la ruta más corta para llegar a un lugar, reproducir música online, jugar, consultar información de fuentes RSS y un sinnúmero de acciones que podemos realizar en nuestro dispositivo personal.

Debido al auge que existe actualmente en el ámbito móvil, la presente tesis está enfocada en el desarrollo de una aplicación web y móvil, que acerque a los investigadores del Instituto de Ingeniería la información de sus proyectos a través de este tipo de dispositivos.

Los objetivos que se cubrirán durante el desarrollo del presente trabajo son:

- Realizar una aplicación web que muestre la información referente a los proyectos de investigación del IINGEN en cualquier lugar donde se disponga de una computadora u otro tipo de dispositivo con acceso a internet.
- Realizar una aplicación móvil cuya finalidad sea que la información referente a los proyectos de investigación del IINGEN pueda ser visualizada en diferentes dispositivos móviles con conexión a internet.
- Facilitar la consulta de la información de los proyectos de investigación del IINGEN a los investigadores de la institución.

Para satisfacer dichos objetivos, se ha contemplado la siguiente estructura, la cual permitirá abordar cada uno de ellos con la intención de tener una mejor comprensión de cada una de las partes que integran la aplicación o aplicaciones a desarrollar.

1. Introducción

Antecedentes

En este capítulo se habla sobre los tipos de proyectos de investigación que se realizan dentro del Instituto de Ingeniería. Se describe el sistema actual en el que los investigadores consultan la información referente a sus proyectos, explicando sus ventajas y desventajas y el por qué se pretende desarrollar una nueva aplicación.

Tópicos generales

En esta parte se presenta un marco teórico que contiene los temas que servirán como base para desarrollar cada uno de los objetivos. Por ejemplo, hablaremos de las plataformas de desarrollo de software haciendo énfasis en las plataformas móviles. Se abarcaran algunos temas de Ingeniería de Software para poder llevar a cabo el ciclo de vida del software de las aplicaciones.

Análisis de la aplicación

En este capítulo se describe la problemática que se tiene con el sistema actual y la propuesta que se plantea para dar solución, la cual está enfocada en el desarrollo de una nueva aplicación de tipo web y otra de tipo móvil. Se realiza el análisis de los requerimientos de estas nuevas aplicaciones, así como la planeación de tiempos y costos.

Diseño de la aplicación

En este capítulo se define la arquitectura a utilizar para la implementación de las aplicaciones. Se generan casos de uso derivados de los requerimientos y se realiza el diseño de la interfaz de las aplicaciones web y móvil.

Desarrollo de la aplicación

En este capítulo se explica el desarrollo de cada una de las aplicaciones, haciendo una breve descripción de las páginas que las componen y las herramientas de desarrollo utilizadas.

Pruebas, depuración y liberación de la aplicación

En este capítulo se describen los tipos de pruebas y depuraciones que se realizaron para asegurar la calidad de las aplicaciones. También se describen los pasos a seguir en la liberación de las aplicaciones móviles para poder ser descargadas.

Capítulo 2

Antecedentes

2.1 El Instituto de Ingeniería

EL Instituto de Ingeniería de la Universidad Nacional Autónoma de México (IIUNAM) es el centro de investigación más productivo del país, pues su misión ha sido contribuir al desarrollo de éste y al bienestar de la sociedad a través de la investigación en diversas áreas de la Ingeniería, la formación de recursos humanos y la vinculación con la sociedad.

La comunidad del Instituto de Ingeniería está integrada por investigadores, técnicos académicos, personal administrativo y una amplia población de becarios que trabajan en la realización de tesis de licenciatura, maestría y doctorado.

Desde su fundación en el año de 1956, el Instituto de Ingeniería realiza proyectos de investigación enfocados a problemas generales de la ingeniería, colaborando con entidades tanto públicas como privadas. Asimismo, proporciona servicios de Ingeniería a los diversos sectores de la sociedad y ha puesto énfasis en la formación de recursos humanos y en difundir los resultados de sus investigaciones contribuyendo así al desarrollo del país y el bienestar de la sociedad.

Algunos de los proyectos son financiados con recursos que la UNAM otorga, y la mayor parte, mediante contratos de investigación con empresas solicitantes. Por ello, el prestigio del Instituto de Ingeniería es ampliamente reconocido.

En la actualidad, las áreas que cubre el Instituto incluyen una rica mezcla de las disciplinas e interdisciplinas de la ingeniería moderna. Los temas de tesis que se dirigen dentro del Instituto y los artículos que se publican representan una muestra muy variada de lo mejor de la ingeniería nacional, que honra la prestigiada tradición del Instituto de Ingeniería de la UNAM.

Entre de las especialidades que se cultivan actualmente dentro del IIUNAM se encuentran la ingeniería estructural, que estudia entre otras cosas el comportamiento estructural de edificios y el comportamiento dinámico de los materiales; la ingeniería en riesgos naturales cuya finalidad es contar con evaluaciones probabilistas de riesgo ante amenazas naturales como son sismos, huracanes, inundaciones y granizo; la ingeniería hidráulica, que estudia el control de inundaciones, estructuras hidráulicas entre otras cosas; la ingeniería eléctrica electrónica que abarca las áreas de control automático, sistemas eléctricos de potencia, autómatas celulares, la ingeniería lingüística y la optoelectrónica; la ingeniería mecánica cuyas líneas de investigación se enfocan al diseño y desarrollo de todo tipo de máquinas y mecanismos; la ingeniería en computación, cuyas ramas de estudio abarcan al diseño y manejo bases de datos, las redes de datos y el desarrollo de software; la ingeniería en telecomunicaciones que abarca desde la simulación de amplificadores ultrarápidos, hasta la fabricación de satélites de comunicación educacionales para entrenar y atraer estudiantes al mundo de la ciencia y la tecnología de punta.

En el futuro, el Instituto prevé preservar su papel de árbitro nacional de la ingeniería y actor principal del desarrollo tecnológico. Al mismo tiempo, apoyará de manera más efectiva a la docencia y la formación de expertos, conjuntamente con la Facultad de Ingeniería, como siempre,

2.1 El Instituto de Ingeniería

y acrecentará su participación en programas universitarios de punta, así como la vinculación con la industria, el desarrollo de nuevas tecnologías y la colaboración con instituciones afines.

2.2 Sistema Institucional de Registro Financiero SIRF

El Instituto de Ingeniería cuenta con un área administrativa la cual se encarga, entre otras cosas, de revisar los procedimientos para el manejo contable de los recursos del Instituto así como de cada proyecto de investigación. Registra nuevos proyectos, da seguimiento a éstos y prepara informes financieros para la toma de decisiones.

Con el fin de automatizar los procesos relacionados con el área administrativa, el Instituto ha desarrollado aplicaciones que facilitan la realización de las diversas tareas de esta área. Tal es el caso del Sistema Integral de Administración Financiera (SIAF), cuyo fin es llevar una contabilidad integral dentro del Instituto y mantener al tanto a los Investigadores de la situación financiera de sus proyectos que tiene a cargo. De la misma manera, cada dependencia de la UNAM cuenta con su propio sistema de registro financiero, con lo cual, existe un alto grado de heterogeneidad en el registro interno de todas ellas. Para simplificar y hacer más eficiente la gestión financiera universitaria surgió la necesidad de implantar un sistema único para el registro financiero de las dependencias y vincular la gestión con los sistemas internos del Patronato Universitario, agilizando de este modo la disposición oportuna de la información. Con estos propósitos y para cumplir con las obligaciones que la UNAM tiene con la Secretaría de Hacienda y Crédito Público (SHCP) en materia de declaración de impuestos, el Patronato Universitario desarrolló el Sistema Institucional de Registro Financiero (SIRF). Se trata de una aplicación general para todas las dependencias universitarias que pretende sustituir a los sistemas de registro local existentes. Actualmente, algunas dependencias ya están implantando al SIRF como sistema de registro y control financiero, como es el caso del Instituto de Ingeniería que ya lo puso en operación.

El SIRF es un sistema de registro y administración que permite a las dependencias un control y manejo general, ágil y confiable de los recursos asignados. Se trata de un sistema modular que está diseñado para cubrir las necesidades de control y registro de las operaciones financieras, así como de los bienes patrimoniales. Desde un punto de vista técnico, es un sistema desarrollado en un ambiente de red, utiliza el manejador de base de datos MySQL y se utilizó el lenguaje de programación Visual Basic para el desarrollo de todos los módulos que lo conforman. Para acceder a los diferentes módulos, el SIRF cuenta con una autenticación, por lo que cada usuario posee un identificador y una contraseña únicos para este sistema.

Para hacer más ágiles y confiables los procesos, el SIRF interactúa con los sistemas institucionales del Patronato Universitario, como son:

- Sistema de Información para la Administración Universitaria, SIAUWEB.
- Sistema Integral de Control Patrimonial, SICOP

- Sistema Integral de Ingresos Extraordinarios, SIIE.
- Sistema de Pagos en las Unidades de Proceso

2.3 Situación Actual

Como se mencionó en el punto anterior, el Instituto de Ingeniería ya puso en marcha cada módulo que compone al SIRF y es éste el sistema que se utiliza actualmente en el Instituto para llevar un control financiero y contable de los recursos, así como para realizar las consultas pertinentes.

2.3.1 Usuarios

El SIRF en sus distintos módulos que lo componen, es utilizado por el personal del área administrativa para llevar a cabo las tareas afines a ésta. También hacen uso del SIRF, exclusivamente del módulo de Ingresos Extraordinarios en la parte de consulta de proyectos, los investigadores y técnicos académicos, ya que es aquí donde pueden consultar la información, consolidada e íntegra, sobre los proyectos que tienen a cargo. Esta información consiste en el estado actual del proyecto, el presupuesto, el saldo por ejercer, el informe de los gastos y el de la cobranza.

Para los investigadores es de gran importancia contar con esta información en cualquier momento y lugar donde se encuentren, ya que es necesario que estén enterados de los estados financieros que guardan sus proyectos en tiempo real. Esta necesidad se pretende cubrir con la aplicación de la cual trata esta tesis.

2.3.2 Arquitectura del SIRF

El SIRF está basado en una arquitectura Cliente-Servidor. Esto significa que cuando un usuario (cliente) ingresa al SIRF e inicia una sesión, se conecta al servidor y manda peticiones a éste. El servidor procesa dichas solicitudes y envía los resultados al cliente.

Para que los usuarios puedan ingresar al SIRF deben estar conectados a la red interna del Instituto de Ingeniería, ya sea dentro de las instalaciones del Instituto o mediante una VPN¹. El propósito de la aplicación web y la aplicación móvil es poder acceder a la información de los proyectos a través de internet, ya sea por medio de una computadora de escritorio o portátil, o también mediante

¹ VPN (Virtual Private Network) Red Privada Virtual

2.3 Situación Actual

un dispositivo móvil como un Smartphone o una Tablet, sin tener que recurrir a la implementación y utilización de una VPN.

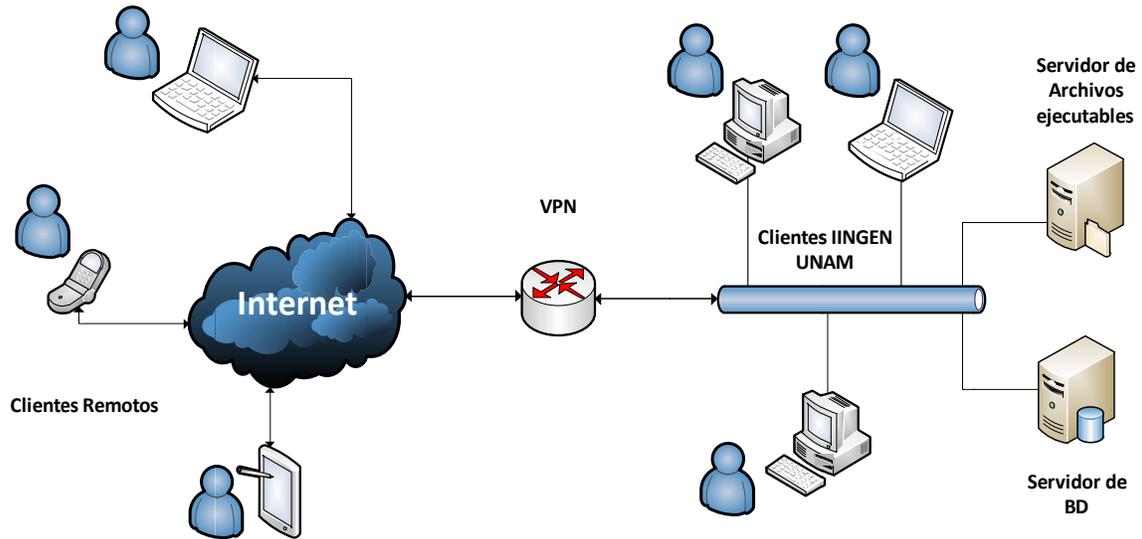


Figura 2.1 Arquitectura del SIRQ

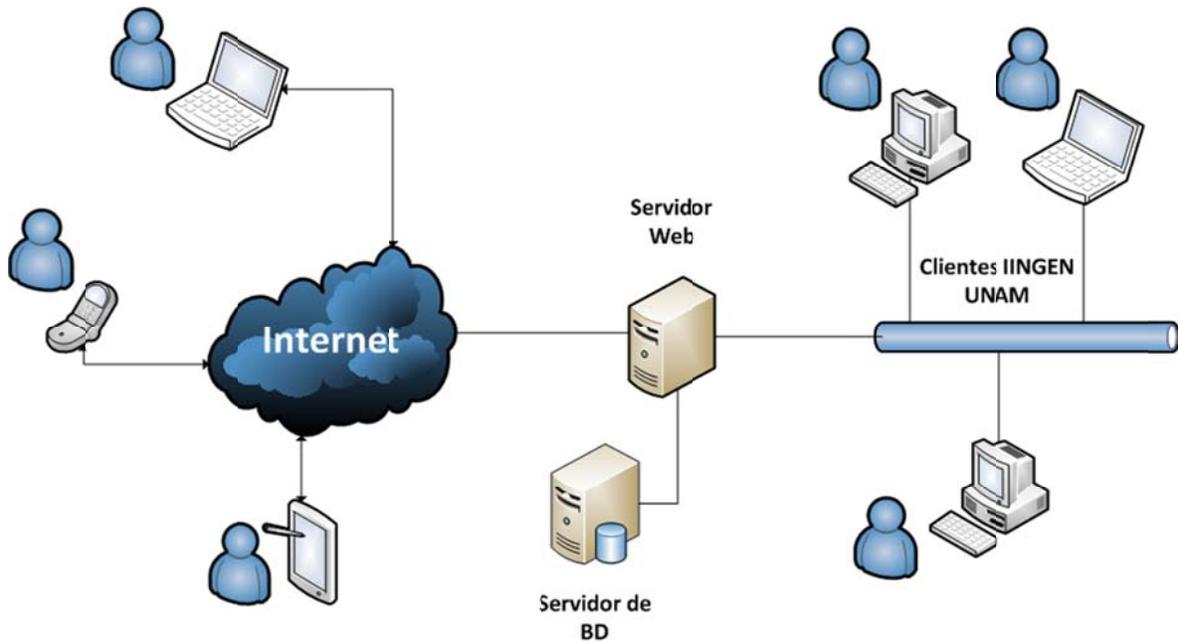


Figura 2.2 Propuesta de Arquitectura del SIRQ

2.3.3 Módulos que componen al SIRF

Como ya se mencionó antes, el SIRF es un sistema modular, cuyos módulos se describen a continuación:

<p>Administración</p> <ul style="list-style-type: none"> Este módulo se encarga de descargar la información de los catálogos iniciales los cuales se encuentran en el SIAUWEB, información sobre el presupuesto, cuentas por pagar, ingresos extraordinarios y gastos centralizados; también se encarga del control de acceso de los usuarios y de realizar algunas configuraciones esenciales como son la fecha del sistema, los tipos de cambio, entre otras.
<p>Recepción</p> <ul style="list-style-type: none"> Este módulo se utiliza en la atención a usuarios y trámites de documentos.
<p>Egresos</p> <ul style="list-style-type: none"> Es el módulo más versátil del SIRf. Se encarga de la asignación de recursos por coordinación, permite el registro e impresión de la forma administrativa, realiza la administración de documentos. También efectúa consultas detalladas de los recursos financieros asignados a las coordinaciones y a la propia dependencia y prepara la información para generar de manera automática la contabilidad del día.
<p>Contabilidad</p> <ul style="list-style-type: none"> Se encarga de la generación automática de la contabilidad, mediante la captura de información a través de los módulos de Egresos, Ingresos Extraordinarios, Caja, PAPIIT y CONACYT; también se encarga del registro de pólizas de saldos, de la generación de la balanza de comprobación y del cierre mensual y anual.
<p>Caja</p> <ul style="list-style-type: none"> Controla las chequeras asignadas al fondo fijo, registra los movimientos de la contabilidad y genera las pólizas correspondientes.
<p>Ingresos Extraordinarios</p> <ul style="list-style-type: none"> Lleva el control y la administración de convenios, contratos y apoyos, así como también de los ingresos propios. Realiza consultas diversas del estado que guardan los convenios y contratos. Estas consultas son realizadas por los investigadores y académicos para informarse del estado actual de sus proyectos desde cualquier lugar donde se encuentren. Esta parte del módulo de Ingresos Extraordinarios es a la que se enfoca esta tesis.
<p>PAPIIT-CONACYT</p> <ul style="list-style-type: none"> Controla y administra los recursos asignados a los proyectos PAPIIT (Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica). Descarga la información presupuestal y datos generales de los proyectos vía SIAUWEB, realiza el trámite de documentos y genera la contabilidad por trámite. Genera reportes financieros y consulta el estado que guardan los proyectos.
<p>Almacén</p> <ul style="list-style-type: none"> Se encarga de la administración de los bienes y realiza el control de entrada y salida de insumos.
<p>Registro Patrimonial</p> <ul style="list-style-type: none"> Imprime las etiquetas de los bienes.

Tabla 2.1 Módulos del SIRF

2.3.4 Ventajas y Desventajas del SIRF

Ventajas

- Mayor agilidad y eficiencia en los procesos administrativos del instituto.
- Consulta de saldos en tiempo real.
- Elaboración de reportes con información veraz y oportuna para la toma de decisiones.
- Conciliación en línea de los bienes institucionales y económicos mediante la implementación del código de barras en la identificación del número de inventario.
- Control preciso del ingreso de la documentación a trámite.
- Consulta en línea del estado que guarda cualquier documento dentro del sistema.
- Ahorro de tiempo en la elaboración del cierre mensual.
- Transparencia en el manejo de la información.

Desventajas

Actualmente, para consultar la información del estado actual de los proyectos es necesario estar conectados a la red del Instituto de Ingeniería. Los Investigadores algunas veces necesitan consultar la información desde otro lugar, ya sea desde otro estado de la República o desde otro País. Es por eso que la aplicación web y móvil cubrirá este requerimiento de manera que a través de internet se puedan realizarse dichas consultas.

2.4 Bibliografía

- [1] Sitio web del Instituto de Ingeniería
<http://www.iingen.unam.mx>
Noviembre 2011
- [2] Sitio web del SIAF
<http://siaf.iingen.unam.mx/introSiaf/default.htm>
Noviembre 2011
- [3] Documentación del SIRF
www.dgsa.unam.mx/sirf.pdf
Noviembre 2011

Capítulo 3

Tópicos Generales

3.1 Plataformas de Desarrollo de Software

Existen actualmente diversas formas en las que se puede desarrollar un programa o aplicación de software, dependiendo de las necesidades del cliente o de la empresa. Puede ser que el cliente requiera una aplicación de escritorio que pueda instalar y ejecutar en varias computadoras, una aplicación web que ejecute desde cualquier navegador de internet, o quizá una aplicación móvil que pueda instalar y ejecutar en su propio dispositivo móvil. Cada tipo de aplicación presenta retos continuos para los ingenieros de software y existen cada vez menos límites para poder desarrollarlas.

Actualmente, el software se puede presentar como aplicaciones móviles, aplicaciones web, aplicaciones de escritorio, aplicaciones de consola, entre otras. A continuación, para cada una de ellas explicaré los aspectos más relevantes y mencionaré algunas plataformas de desarrollo en las que podemos crear dichas aplicaciones.

3.1.1 Móvil

El concepto de movilidad en el ámbito de TIC se refiere a poder acceder a la información que necesitamos desde cualquier lugar en cualquier momento. Actualmente existe una gran variedad de dispositivos con las características necesarias para que esto sea posible, conocidos como dispositivos móviles. Éstos son aparatos de tamaño pequeño que cuentan con características tales como las que se muestran en la siguiente tabla:

1	• Capacidades limitadas de procesamiento
2	• Conexión permanente o intermitente a una red
3	• Memoria limitada
4	• Diseños específicos para una función principal y versatilidad para el desarrollo de otras funciones
5	• Tanto su posesión como su operación se asocia al uso individual de una persona, por lo que puede configurarlo a su gusto

Tabla 3.1 Características de los dispositivos móviles

3.1 Plataformas de Desarrollo de Software

Estos dispositivos son capaces de realizar diversas acciones como navegar en internet, consultar el correo electrónico, tomar fotografías y video de gran calidad, localizar lugares, ver televisión, etc. Estas tareas son realizadas por una gran variedad de aplicaciones para este tipo de dispositivos, llamadas aplicaciones móviles. Una aplicación móvil es un programa nativo de algún dispositivo móvil, es decir, una aplicación que se instala directamente en el dispositivo. Este tipo de aplicaciones están hechas para cumplir una necesidad en específico, ya que el procesamiento en un dispositivo móvil es muy restringido en comparación con el procesamiento de las computadoras de escritorio. Además, la cantidad de información que se muestra en la pantalla de un dispositivo móvil debe ser concisa.

Para desarrollar una aplicación móvil, debemos tener en cuenta una serie de consideraciones.

Antes que nada, debemos plantear la idea de lo que queremos hacer, si queremos crear algo nuevo o mejorar algo existente, así como conocer los objetivos de negocio.

Debemos decidir en qué dispositivos queremos que se ejecute nuestra aplicación, tomando en cuenta que existe una gran diversidad de dispositivos con distintas características, así como una gran variedad de plataformas o sistemas operativos móviles. Normalmente, esta decisión está condicionada por la población objetivo, pues se debe conocer cuáles son los dispositivos que predominan en ésta. No obstante, lo ideal sería desarrollar la aplicación para todas las plataformas posibles.

Actualmente, en el mercado de dispositivos móviles encontramos los siguientes sistemas operativos (Figura 3.1). Cada uno tiene ventajas e inconvenientes que dependen del uso que cada quien le dé.



Figura 3.1 Sistemas Operativos Móviles en el Mercado (Agosto 2012)

De acuerdo con una encuesta realizada en marzo del 2012 por la consultora estadounidense IDC (International Data Corporation, por sus siglas en inglés) a más de 2100 desarrolladores de todo el mundo, donde se les preguntó cuál era su plataforma móvil preferida, el 89% de los encuestados dijeron estar interesados en desarrollar aplicaciones para iPhone y el 88% para iPad, mientras que el porcentaje de interesados en desarrollar aplicaciones para Android fue del 78.6%. Por su parte, Windows Phone sigue ganando partidarios y se ha convertido en la tercer plataforma en importancia dentro del mercado móvil, mientras que BlackBerry OS sigue en caída libre perdiendo un 5.2% para quedar con tan solo el 15,5% de las preferencias de los desarrolladores durante el primer trimestre de 2012 [1].

Una vez que conocemos el mercado actual de los dispositivos móviles y hemos decidido para qué plataforma vamos a desarrollar nuestra aplicación, debemos definir el tipo de aplicación que vamos a realizar y analizar cuales servicios vamos a requerir. Por ejemplo, una aplicación que muestra información obtenida de internet, necesita forzosamente de este servicio, o una aplicación para la ubicación de lugares necesita el servicio de GPS².

Posteriormente hay que elegir la plataforma de desarrollo a utilizar. Existen plataformas de desarrollo de aplicaciones para un lenguaje nativo de un sistema operativo. A continuación se muestra una tabla con la información del lenguaje nativo para cada sistema operativo y su plataforma de desarrollo.

Sistema Operativo	Lenguaje	Plataforma de Desarrollo
Android	Java	Eclipse + Plugin + SDK
IOS	Objective C	Xcode + SDK
Windows Phone	XAML y C# o VB	Visual Studio + SDK
BlackBerry	Java	Eclipse + Plugin + SDK

Tabla 3.2 Plataformas de desarrollo de aplicaciones móviles

También existen plataformas de desarrollo para crear aplicaciones para más de un sistema operativo a la vez, con ayuda de un lenguaje intermedio. Esto facilita la tarea de programar en un lenguaje distinto para cada plataforma, lo que nos ahorra tiempo. Un ejemplo es el IDE Titanium Studio de Appcelerator, el cual está basado en el IDE Eclipse y soporta como lenguaje intermedio JavaScript, para desarrollar aplicaciones para las plataformas Android y iOS.

Durante el desarrollo de la aplicación, debemos de considerar algunas características de los dispositivos tales como:

- EL tamaño y el tipo de la pantalla (Si es touch screen o no)
- Memoria de programas y de almacenamiento
- La velocidad del microprocesador
- El control sobre el dispositivo
- Forma de navegación (GPS)

² GPS (Global Positioning System) Sistema de Posicionamiento Global

3.1 Plataformas de Desarrollo de Software

- Sensores (Por ejemplo, el acelerómetro)
- Interfaces (USB, Bluetooth, Wi-Fi, infra rojo)
- Tipo de conectividad (Red Celular GSM, GPRS)

Las aplicaciones móviles una vez concluidas, deben de pasar por un proceso de aceptación por parte de la tienda de aplicaciones correspondiente a cada plataforma, para posteriormente ser publicadas en ésta, quedando al alcance de los usuarios interesados en dicha aplicación.

3.1.2 Web

Una aplicación basada en Web (WebApp), es una aplicación que reside y se ejecuta en un servidor web y a la cual podemos acceder a través de un navegador como Internet Explorer, Mozilla Firefox, Opera, etc. Estas aplicaciones ofrecen un variado arreglo de contenido y funcionalidad a una amplia población de usuarios finales.

Algunas características que hay que tener en cuenta al desarrollar una WebApp se muestran en la siguiente tabla:

Concurrencia
•Se refiere a que un gran número de usuarios pueden tener acceso a la WebApp al mismo tiempo.
Carga Impredecible
•El número de usuarios de la WebApp puede variar constantemente. Un día puede haber 100 usuarios usando el sistema y al siguiente día puede haber más de 5000.
Desempeño
•Si un usuario de la WebApp debe esperar demasiado (para ingresar, para procesamiento en el lado del servidor, para obtener información de la Base de datos) puede decidir irse a cualquier otra parte.
Disponibilidad
•Los usuarios generalmente demandan acceso a las WebApps las 24 horas del día, los siete días de la semana, los 365 días del año.
Gobernada por los datos
•Por lo general, las WebApps se utilizan para facilitar el acceso a información que existe en bases de datos, que en algunos casos, no eran originalmente parte integral del ambiente basado en web (por ejemplo, aplicaciones financieras).

Evolución Continua

- A diferencia de las aplicaciones de escritorio, que evolucionan a lo largo de una serie de liberaciones espaciadas cronológicamente, las aplicaciones Web evolucionan constantemente.

Seguridad

- Puesto que las WebApps están disponibles mediante el acceso a internet, con la finalidad de proteger el contenido confidencial y ofrecer modos seguros de transmisión de datos, se deben implementar fuertes medidas de seguridad a lo largo de la infraestructura que sustenta una WebApp y dentro de la misma. Además, es recomendable limitar la población de usuarios finales que pueden tener acceso a la aplicación.

Estética

- Es importante para una WebApp su presentación y la disposición de sus elementos, pues, por ejemplo, cuando una aplicación se diseña para comercializar productos, parte del éxito tiene que ver con el diseño técnico.

Tabla 3.3 Características de las aplicaciones móviles

Actualmente existe un conjunto variado de lenguajes de programación y de plataformas de desarrollo con los cuales es posible crear aplicaciones web de una manera cada vez más simple.

La siguiente tabla muestra algunas plataformas de desarrollo de aplicaciones web y sus respectivos lenguajes de programación:

Lenguaje	Plataforma de Desarrollo
C#, Visual Basic	Visual Studio + .NET Framework
Java	Eclipse for Web Developers + SDK de Java
PHP	Zend Studio
PHP, Ruby on Rails	Aptana

Tabla 3.4 Plataformas de desarrollo de aplicaciones web

3.1.3 Consola

Una aplicación de consola es aquella que se ejecuta en una consola o terminal y consta solamente de texto. Para interactuar con una aplicación de consola, simplemente se hace uso de un teclado como elemento de entrada de datos y una pantalla de visualización como elemento de salida.

Ejemplos de aplicaciones de consola son los siguientes:

- Editor de textos
- Antivirus (Escáner por línea de comandos)

3.1 Plataformas de Desarrollo de Software

- Administrador de red
- Visualizador del tráfico en una red
- Lector RSS
- Cliente de IRC
- Aplicaciones de gestión de procesos

La siguiente tabla contiene algunas plataformas de desarrollo de aplicaciones de Consola y sus respectivos lenguajes de programación:

Lenguaje	Plataforma de Desarrollo
C#, Visual Basic o C++	Visual Studio
Java	Línea de comandos de Windows, la terminal de Linux o la terminal de Mac OS
Java	Eclipse

Tabla 3.5 Plataformas de desarrollo de aplicaciones de consola

3.1.4 Escritorio

Una aplicación de escritorio es un programa de software que se instala y ejecuta en una computadora de escritorio sobre un sistema operativo de interfaz visual como Windows, Mac OS o las diversas distribuciones de Linux.

Este tipo de aplicaciones dependen de las características de la computadora y del sistema operativo, ya que estos van evolucionando constantemente y van adquiriendo mejores características que hacen que las aplicaciones de escritorio se vuelvan obsoletas.

Las aplicaciones de escritorio se desarrollan para cubrir necesidades específicas de una empresa, o de un grupo de usuarios con actividades a fines, como la contabilidad, gestión de personal, etc.

Algunos ejemplos de aplicaciones de escritorio son:

- Hojas de cálculo
- Administrador de Bases de Datos
- Graficadores
- Procesadores de texto
- Software de comunicación
- Software empresarial integrado
- Editores de imágenes o video

Entre los lenguajes de programación y las plataformas de desarrollo de aplicaciones de escritorio se encuentran los que se muestran en la siguiente tabla:

Lenguaje	Plataforma de Desarrollo
C#, Visual Basic o C++	Visual Studio
Java	Línea de comandos de Windows, la terminal de Linux o la terminal de Mac OS
Java	Eclipse
Objective C	Xcode

Tabla 3.6 Plataformas de desarrollo de aplicaciones de escritorio

3.2 Ciclo de Vida del Software

El ciclo de vida del software es un proceso para la construcción de un producto de software. Este proceso ofrece control, organización y estabilidad a una actividad que puede volverse caótica si no es controlada.

El desarrollo de software por lo general involucra las siguientes etapas:

Análisis y definición de los requerimientos: En esta etapa se debe entender de forma detallada cuál es la problemática a resolver, recopilando y analizando los requisitos del cliente y así obtener las especificaciones del sistema.

Diseño del sistema: Se determina la estrategia que se va a utilizar para resolver el problema y se establece una arquitectura completa del sistema.

Diseño de programas: Su objetivo es la definición precisa de cada unidad de programa del sistema completo. Los diseñadores de software consideran los recursos de hardware y software disponibles para poder alcanzar su objetivo. Si se llega a la conclusión de que no es posible utilizar algún hardware o software, se planea utilizar una estrategia diferente.

Implementación: Partiendo del análisis y diseño de la solución, en esta etapa se procede a desarrollar la aplicación que solucione el problema mediante el uso de una herramienta computacional determinada.

Prueba unitaria: Es la prueba individual de cada subconjunto de la aplicación, bajo el mayor número de situaciones posibles a las que se pueda enfrentar, para garantizar que se implementaron de acuerdo con las especificaciones.

Prueba de Integración: Su objetivo es garantizar que los diferentes módulos del sistema trabajen en conjunto correctamente.

Prueba del sistema: Su objetivo es garantizar que el sistema cumple con las especificaciones originales.

Entrega del sistema: Se entrega el sistema completo terminado y su respectiva documentación. Ésta es de gran importancia ya que es necesaria para los usuarios del software y para desarrollos futuros.

Mantenimiento: Una vez que el programa se ha instalado y puesto en marcha para realizar la solución del problema que satisface las necesidades del usuario, es importante establecer una estructura de actualización, verificación y validación que permita a dicho programa seguir siendo útil y mantenerse actualizado. Para realizar un mantenimiento adecuado es necesario contar con una buena documentación del sistema.

Existen distintos modelos del ciclo de vida del software los cuales conducen a un equipo de trabajo a través de un conjunto de actividades que se organizan en un flujo de proceso, el cual puede ser lineal, incremental o evolutivo. Algunos modelos del ciclo de vida del software son los siguientes:

Modelo en cascada. Es el modelo más antiguo de la ingeniería de software. Sugiere un enfoque secuencial de las etapas del proceso, las cuales se representan cayendo en cascada desde una etapa a la siguiente, como se muestra en la Figura 3.2. Una etapa debe completarse para que dé inicio la siguiente. El inconveniente de este modelo es que es poco realista, ya que los proyectos generalmente no siguen este flujo lineal. Por ejemplo, los requerimientos muchas veces cambian a lo largo de todo el ciclo de vida.

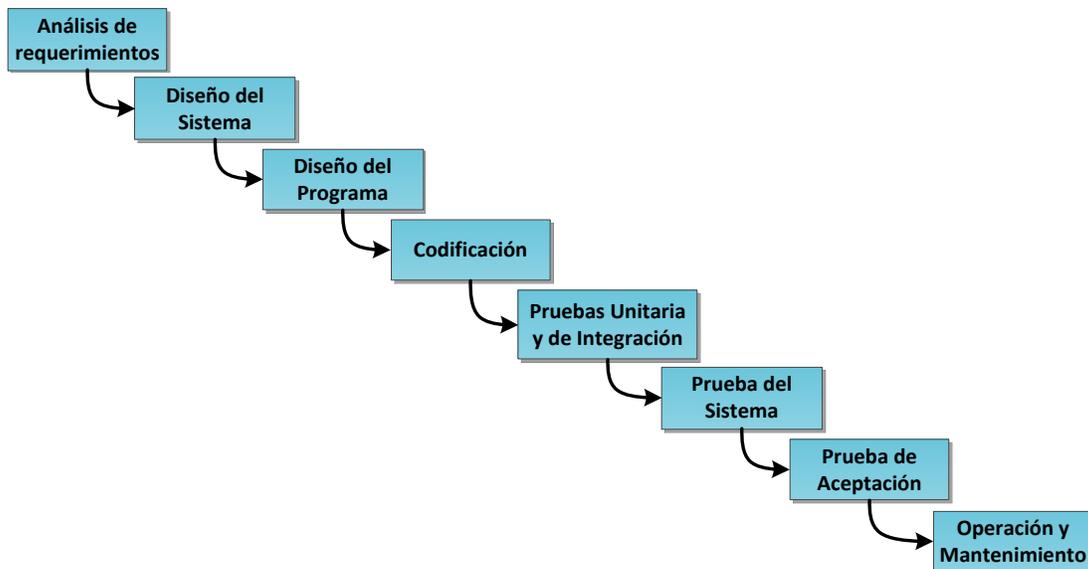


Figura 3.2 Modelo en Cascada

3.3.1 RUP

RUP (Rational Unified Process) es una metodología de realización o de evolución de software que se basa en UML³. RUP fue desarrollado por Rational Software a mediados de los 90's. Posteriormente, en el 2002, IBM adquiere Rational Software y RUP es avalado como un estándar en el ámbito internacional.

El objetivo de RUP es asegurar la producción de software de alta calidad que satisfaga la necesidad del usuario final dentro de un tiempo y presupuesto previsible. Es una metodología de desarrollo iterativo e incremental enfocada hacia los casos de uso, la arquitectura y el manejo de riesgos. El proceso de RUP debe adaptarse a las características de la organización para la que se está desarrollando el software.

RUP mejora la productividad del equipo ya que permite que cada miembro del grupo sin importar su responsabilidad específica acceda a la misma base de datos de conocimiento. Esto hace que todos compartan el mismo lenguaje, la misma visión y el mismo proceso acerca de cómo desarrollar software.

Las seis mejores prácticas de desarrollo que aplica RUP son las siguientes:

- Gestión de requerimientos
- Desarrollo de software en forma iterativa
- Uso de arquitecturas basadas en componentes
- Modelización visual del software
- Verificación de la calidad del software
- Control de cambios

El ciclo de vida RUP es una implementación del modelo en espiral. Se establecen tareas en fases e iteraciones. Las fases del proceso son las siguientes:

Inicio (Inception) <ul style="list-style-type: none">•Consiste en evaluar el proyecto, se obtiene una visión general de los requerimientos del proyecto y se decide llevar adelante o no el proyecto en función de los recursos requeridos, se determinan los principales casos de uso y se hace un primer esbozo de arquitectura.
Elaboración <ul style="list-style-type: none">•Esta fase tiene como objetivo construir la arquitectura del sistema. Una vez concluida la elaboración, se conocen definitivamente las exigencias del proyecto y su arquitectura. También se tiene una lista de riesgos.
Construcción <ul style="list-style-type: none">•Corresponde al desarrollo de software de la arquitectura de manera incremental.
Transición <ul style="list-style-type: none">•Comprende la instalación del software en los equipos del cliente.

Tabla 3.7 Fases del ciclo de vida de RUP

³ UML (Unified Modeling Language) Lenguaje Unificado de Modelado

Gráficamente, el ciclo de vida de RUP puede representarse en dos dimensiones (Figura 3.4):

- El eje horizontal representa el tiempo y muestra el aspecto dinámico del proceso, expresado en término de ciclos, fases, iteraciones y metas.
- El eje vertical representa el aspecto estático del proceso y se expresa en términos de actividades, artefactos, trabajadores y flujos de trabajo.

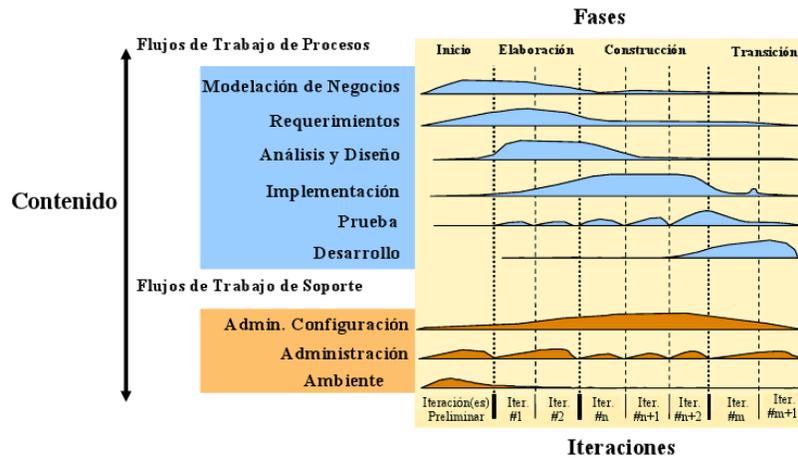


Figura 3.4 Ciclo de vida RUP [4]

3.3.2 Ágil

El término “Métodos ágiles” aplicado al desarrollo de software, surge a partir de una reunión convocada en febrero del 2001 en Salt Lake City donde participaron 17 expertos de la industria del software incluyendo algunos de los creadores de metodologías de desarrollo de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos.

Los integrantes de esta reunión resumieron en cuatro postulados el enfoque al cual dirigen el desarrollo de software, lo que ha quedado denominado como “Manifiesto Ágil” [6]:

Enfoque de los Métodos Ágiles	Enfoque de los Métodos Tradicionales
Individuos e interacciones	Procesos y herramientas
Software funcionando	Documentación extensiva
Colaboración con el cliente	Negociación contractual
Respuesta ante el cambio	Seguir un plan

Tabla 3.8 Comparativa entre el enfoque de los métodos ágiles vs métodos tradicionales

3.3 Metodologías de Desarrollo de Software

Aunque se valoran los elementos de la derecha, las metodologías ágiles se enfocan más en los de la izquierda.

El Manifiesto ágil consta de doce principios sobre los que se basan los métodos ágiles (Tabla 3.9).

Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.	Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.	Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.	Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.	El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.	El software funcionando es la medida principal de progreso.	Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.	La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.	Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.	A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Tabla 3.9 Principios del manifiesto ágil [6]

Un equipo ágil es un equipo rápido que responde de manera apropiada a los cambios. Cambios en el software que se va a construir, cambios entre los miembros del equipo, cambios debidos a las nuevas tecnologías, cambios de todo tipo que pueden incidir en el producto que se construye o en el proyecto que crea el producto.

Un proceso de software ágil debe ser adaptable, ya que resulta difícil predecir cuáles requisitos del software persistirán y cuales cambiarán. De igual forma, las prioridades del cliente generalmente cambian mientras se desarrolla el proyecto. Para llevar a cabo una adaptación incremental, un equipo ágil requiere de la retroalimentación con el cliente para que sea factible realizar las adaptaciones apropiadas.

Los incrementos del software, ya sean prototipos ejecutables o una porción incremental de desarrollo, deben entregarse en cortos periodos de tiempo para que la adaptación mantenga un buen ritmo con el cambio. Este enfoque iterativo permite al cliente evaluar el incremento del software de manera regular, proporcionar la retroalimentación necesaria al equipo de desarrollo e influir sobre las adaptaciones del proceso que se realizan para adecuar la retroalimentación.

3.3.2.1 SCRUM

Scrum es un marco de trabajo iterativo e incremental que se emplea en entornos en los cuales los requisitos son inestables y requieren rapidez y flexibilidad.

Scrum es una metodología ágil, y como tal:

- Es un modo de desarrollo de carácter adaptable más que predictivo.
- Está orientada a las personas más que a los procesos.
- Emplea la estructura de desarrollo ágil incremental basada en iteraciones y revisiones.

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos, conocidos como iteraciones. Cada iteración tiene que proporcionar un resultado completo, un incremento operativo que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

Estas iteraciones son la base del desarrollo ágil, y Scrum gestiona su evolución a través de reuniones breves diarias en las que todo el equipo revisa el trabajo realizado el día anterior y el previsto para el día siguiente.

Scrum denomina “sprint” a cada iteración de desarrollo y recomienda realizarlas con duraciones de 30 días o hasta 15 días si así se requiere. El sprint es por tanto el núcleo central que proporciona la base de desarrollo iterativo e incremental.

Al finalizar cada iteración se lleva a cabo una revisión con todas las personas implicadas en el proyecto. El desarrollo incremental implica que al final de cada iteración se dispone de una parte del producto operativa que se puede inspeccionar y evaluar.

El desarrollo Scrum va generando el diseño y la arquitectura final de forma evolutiva durante todo el proyecto.

Las actividades que se llevan a cabo en Scrum son las siguientes (Figura 3.5):

- **Planificación del Sprint:** Jornada de trabajo previa al inicio de cada sprint en la que se determina cuál va a ser el trabajo y los objetivos que se deben cubrir en esa iteración.
- **Seguimiento del Sprint** Breve revisión diaria, en la que cada miembro describe tres cuestiones:
 - El trabajo que realizó desde la reunión anterior
 - El trabajo que tiene previsto realizar hasta la próxima reunión
 - Impedimentos que se deben solventar para que pueda realizar el trabajo

3.3 Metodologías de Desarrollo de Software

- **Revisión del Sprint:** Análisis y revisión del incremento generado, así como una retrospectiva de lecciones aprendidas para mejorar la implementación de Scrum en la organización.

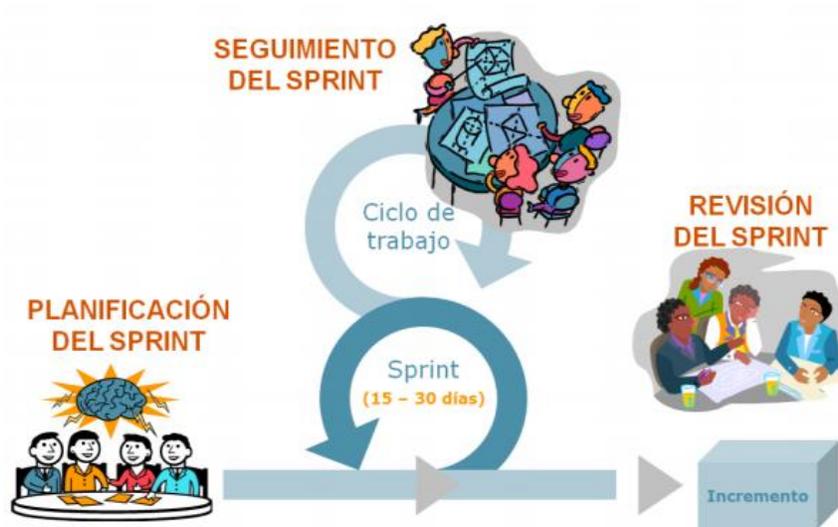


Figura 3.5 Actividades de SCRUM [5]

3.4 Calidad del Software

La calidad del software es el grado con el que un sistema o aplicación de software cumple con los requerimientos especificados y las necesidades o expectativas del cliente o usuario.

La siguiente tabla muestra algunos factores que determinan la calidad del software:

<p>Corrección</p> <ul style="list-style-type: none"> • El grado en que una aplicación realiza con exactitud las tareas expresadas en su especificación.
<p>Fiabilidad</p> <ul style="list-style-type: none"> • Es el grado de una aplicación de llevar a cabo las operaciones especificadas con la precisión requerida.
<p>Robustez</p> <ul style="list-style-type: none"> • El grado de una aplicación de reaccionar apropiadamente ante condiciones excepcionales.
<p>Eficiencia</p> <ul style="list-style-type: none"> • Es la capacidad de una aplicación para hacer buen uso de los recursos de hardware y software que necesita.
<p>Portabilidad</p> <ul style="list-style-type: none"> • Es la facilidad con que una aplicación puede ser migrada entre diferentes plataformas de hardware o software.
<p>Integridad</p> <ul style="list-style-type: none"> • Es la característica de un sistema de ser capaz de proteger sus diferentes componentes contra los procesos o elementos que no tengan derecho de acceso a los mismos.
<p>Facilidad de Uso</p> <ul style="list-style-type: none"> • Es la facilidad con la que un usuario puede interactuar con la aplicación.
<p>Facilidad de Mantenimiento</p> <ul style="list-style-type: none"> • Es el esfuerzo necesario para localizar y arreglar un error en un programa.
<p>Flexibilidad</p> <ul style="list-style-type: none"> • Es el esfuerzo necesario para localizar y arreglar un error en un programa.

Tabla 3.10 Factores de la calidad del software [7]

3.4 Calidad del Software

Durante el desarrollo de un software es necesario llevar a cabo un control de calidad. Éste involucra una serie de inspecciones, revisiones y pruebas empleadas a lo largo del proceso de software para garantizar que cada producto de trabajo satisfaga los requisitos que se le han asignado.

Las revisiones son el método más utilizado para validar la calidad de un proceso o producto. Las revisiones del software son un “filtro” para el proceso de software, ya que se aplican en varios puntos durante la ingeniería de software y sirven para descubrir errores y defectos que hay que eliminar.

Se conoce como revisión técnica formal (RTF) a la actividad de control de calidad del software cuyos objetivos son:

1. Descubrir errores en la función, lógica o implementación en cualquier representación del software.
2. Verificar que el software en revisión satisface sus requisitos.
3. Garantizar que el software se ha representado de acuerdo con los estándares predefinidos.
4. Lograr software desarrollado de una manera uniforme.
5. Hacer proyectos más manejables.

Por otro lado, la seguridad del software es una actividad de aseguramiento de la calidad que se enfoca en la identificación y evaluación de los peligros potenciales que pueden afectar negativamente al software y provocar una falla de todo el sistema. Si los peligros se pueden identificar temprano en el proceso de software, las características de diseño de software se pueden especificar de modo que eliminarán o controlarán los peligros potenciales.

Para que las organizaciones en general puedan garantizar la calidad de sus productos y servicios y que éstos satisfagan las expectativas de los clientes, se han desarrollado sistemas de garantía de calidad avalados por estándares ISO⁴. El estándar para el aseguramiento de la calidad que se aplica a cualquier sector económico e industrial es el ISO 9001:2000. Este estándar contiene 20 requisitos que deben estar presentes para un sistema eficiente de garantía de calidad. Puesto que el estándar ISO 9001:2000 es aplicable a todas las disciplinas de ingeniería, se ha desarrollado un conjunto especial de directrices ISO (ISO 90003) que ayudan a interpretar el estándar para emplearlo en el proceso de desarrollo de Software ([Anexo 1](#)).

Los requisitos que delinea este estándar abordan tópicos como responsabilidad de la gestión, sistema de calidad, revisión de contrato, control de diseño, control de documentos y datos, identificación y seguimiento de producto, control de proceso, inspección y pruebas, acciones correctivas y preventivas, control de riesgos de calidad, auditorías de calidad interna, entrenamiento, servicio y técnicas estadísticas.

⁴ ISO (International Organization for Standardization) Organización Internacional de Normalización

3.5 Pruebas de Software

Las pruebas de software son un elemento crítico para la garantía de la calidad del software. Gracias a las pruebas del software podemos descubrir errores no conocidos, cometidos al realizar su diseño y construcción. A medida que se descubren errores en cada prueba, deben diagnosticarse y corregirse empleando un proceso llamado depuración.

Una prueba exitosa es aquella en la que se descubre un nuevo error. Por eso, se deben realizar casos de prueba en los que se tenga una alta probabilidad de encontrar errores.

Son varios tipos de pruebas que se hacen antes de entregar el sistema al cliente, con la seguridad de que funcionará correctamente.

Primero se verifica cada componente del sistema, aislado de los demás componentes. Esta prueba, conocida como **prueba unitaria**, verifica que el componente funciona correctamente. Cuando el conjunto de componentes ha superado la prueba unitaria, el paso siguiente es asegurar que las interfaces entre los componentes están definidas y se manejan correctamente. La **prueba de integración** es el proceso de verificar que los componentes del sistema trabajan juntos conforme a lo descrito en las especificaciones del sistema.

Una vez asegurado que la comunicación entre componentes es correcta de acuerdo con el diseño, se prueba el sistema para asegurar que funciona correctamente. Una **prueba funcional** evalúa el sistema para comprobar que las funciones descritas por la especificación de requerimientos son correctamente ejecutadas por el sistema integrado. El resultado es un sistema en funcionamiento.

Después, mediante una **prueba de rendimiento** es posible comprobar las propiedades emergentes del sistema tales como rendimiento y fiabilidad. Cuando la prueba se lleva a cabo exitosamente se produce un sistema validado.

El siguiente paso es reunirse con el cliente para tener la certeza de que el sistema trabaja de acuerdo con sus expectativas. La **prueba de aceptación** se hace en conjunto con el cliente; el sistema se comprueba con respecto a la especificación de requerimientos del cliente.

Al completar la prueba de aceptación, el sistema aceptado se instala en el ambiente en el que será utilizado y se ejecuta una última **prueba de instalación**, para verificar que el sistema funciona como debe hacerlo, aún en otros equipos de cómputo.

Otros tipos de pruebas enfocadas a aplicaciones web y móviles son las siguientes:

Prueba de Contenido

Este tipo de pruebas sirven para descubrir errores semánticos y sintácticos en el contenido. Abarca tanto el contenido estático como la información que se obtiene de la base de datos en ese

momento. Este tipo de pruebas asegura que la información pasa correctamente de la base de datos al servidor y es mostrada correctamente al usuario.

Prueba de Interfaz

Estas pruebas ejercitan los mecanismos de interacción del usuario con la aplicación, y valida los aspectos estéticos de la interfaz del usuario. El objetivo es descubrir los errores que resultan de mecanismos con una pobre implementación de interacción, u omisiones, inconsistencias o ambigüedades que se han introducido a la interfaz de forma inadvertida.

Prueba de Navegación

Estas aplican los casos de uso derivados del diseño de la aplicación. Se prueban los mecanismos de navegación como botones, enlaces, barras de menú implementados dentro de la plantilla de la interfaz para garantizar que los errores que impiden completar un caso de uso se identifiquen y corrijan.

Prueba de Componentes

Se verifica cada componente funcional del sistema, aislado de los demás componentes

Prueba de Configuración

Intentan descubrir los errores que son específicos respecto de un usuario o ambiente de servidor particulares. Se crea una matriz de referencia cruzada que define todos los posibles sistemas operativos, navegadores, plataformas de hardware y protocolos de comunicación. Estas pruebas se encaminan a descubrir errores por cada posible configuración.

Prueba de Seguridad

Incorpora una serie de pruebas para explotar las vulnerabilidades en la Web App y su ambiente.

Prueba de Desempeño

Abarca una serie de pruebas para valorar cómo afecta el aumento del tráfico de usuarios el tiempo de respuesta y confiabilidad de la Web App. Cuáles componentes de la Web App son responsables de la degradación del desempeño y qué características de uso provocan que ocurra degradación y cómo la degradación del desempeño impacta los objetivos y requisitos globales de la Web App.

3.6 Arquitecturas

Una arquitectura de software es la organización fundamental de un sistema en términos de sus componentes, la relación entre ellos y con el ambiente, y de los principios que guían su diseño y evolución.

3.6.1 Arquitectura Cliente – Servidor

La arquitectura Cliente/Servidor es una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma.

Arquitectura Cliente/Servidor de dos capas

La arquitectura cliente/servidor genérica, conocida como arquitectura de dos capas tiene dos tipos de nodos en la red: clientes y servidores.

En este tipo de arquitectura, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y éste envía uno o varios mensajes con la respuesta (provee el servicio con sus propios recursos).

El cliente o remitente de una solicitud tiene las siguientes características:

- Es quien inicia solicitudes o peticiones.
- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

El servidor o receptor de la solicitud enviada tiene las siguientes características:

- Se encuentran en espera de que lleguen las solicitudes de los clientes.
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No es frecuente que interactúen directamente con los usuarios finales.

3.6 Arquitecturas

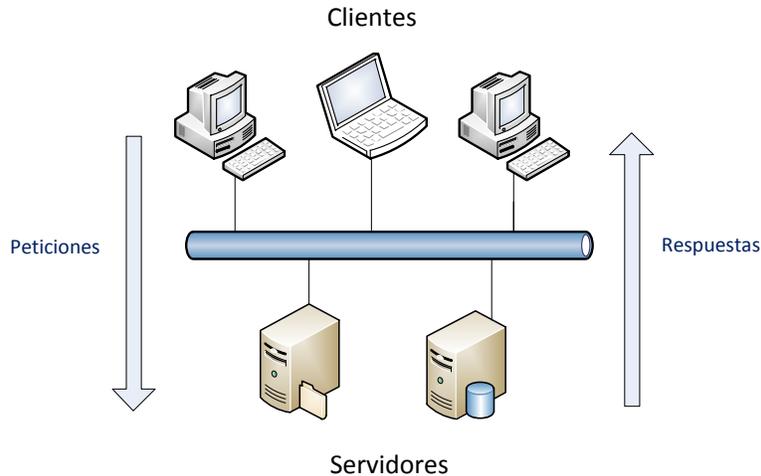


Figura 3.6 Arquitectura Cliente/Servidor de dos capas

Arquitectura Cliente/Servidor de tres capas

La arquitectura de tres niveles o tres capas se compone de tres tipos de nodos. A diferencia de la arquitectura genérica, en ésta existe un nivel intermedio. Sus componentes son:

- Un cliente, es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador Web) para la presentación.
- El servidor de aplicaciones (también denominado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
- El servidor de datos, que proporciona al servidor de aplicaciones los datos que requiere.

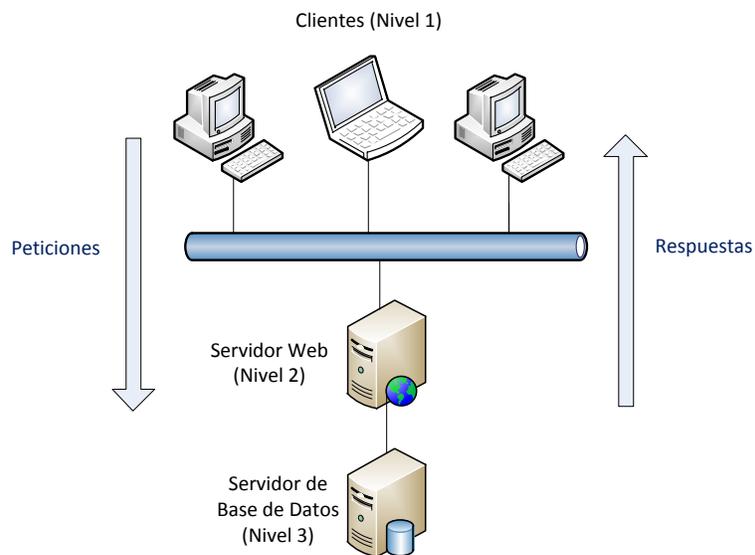


Figura 3.7 Arquitectura Cliente/Servidor de tres capas

En la arquitectura en tres capas, las aplicaciones al nivel del servidor son descentralizadas de uno a otro, es decir, cada servidor se especializa en una determinada tarea, (por ejemplo: servidor web/servidor de bases de datos). La arquitectura en tres capas permite:

- Un mayor grado de flexibilidad
- Mayor seguridad, ya que la seguridad se puede definir independientemente para cada servicio y en cada nivel
- Mejor rendimiento, ya que las tareas se comparten entre servidores

3.6.2 Arquitectura orientada a servicios

La Arquitectura Orientada a Servicios (SOA) es un marco de trabajo para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios, esto con el objetivo de hacer más flexible la manutención de las soluciones de negocio.

La estrategia de orientación a servicios permite la creación de servicios y aplicaciones compuestas que pueden existir con independencia de las tecnologías subyacentes. En lugar de que todos los datos y lógica de negocio residan en un mismo equipo de cómputo, el modelo de servicios facilita el acceso y consumo de los recursos de TI a través de la red. Puesto que los servicios están diseñados para ser independientes, autónomos y para interconectarse adecuadamente, pueden combinarse con suma facilidad en aplicaciones complejas que respondan a las necesidades de una organización. Las organizaciones que adoptan la orientación a servicios pueden crear y reutilizar servicios y aplicaciones, y adaptarlos ante los cambios evolutivos que se producen dentro y fuera de ellas, y con ello adquirir la agilidad necesaria para ganar ventaja competitiva.

Arquitectura orientada a servicios (SOA) permite la creación de sistemas altamente escalables que, en forma bien definida exponen e invocan a los servicios (comúnmente Servicios Web), lo cual facilita la interoperabilidad entre diferentes sistemas propios o de terceros.

Esta arquitectura tiene como componente central a los servicios, los cuales tienen las siguientes características:

- Los servicios implementan funciones de negocio
- Los servicios se caracterizan por un grado bajo de acoplamiento
- Los servicios son componentes aislados que definen sus interfaces independientes de los entornos técnicos de implementación.
- SOA se basa en estándares abiertos como XML⁵, SOAP⁶, WSDL⁷ entre otras.

⁵ XML (eXtensible Markup Language) Lenguaje de Marcas Extensible

⁶ SOAP (Simple Object Access Protocol) Protocolo Simple de Acceso a Objetos

⁷ WSDL (Web Services Description Language) Lenguaje de Descripción de Servicios Web

3.6 Arquitecturas

- Los servicios se pueden integrar por lo general a cualquier solución que los requiera.

En SOA existen programas que proporcionan servicios y otros que los consumen:

- **Proveedores de servicios:** son programas que solo entregan información, administran la lógica de negocio y mantienen su integridad.
- **Consumidores de servicios:** son los programas que únicamente consumen servicios. Normalmente se trata de aplicaciones que disponibilizan la información integrada para la interfaz de usuario.

Beneficios de SOA

- SOA estandariza las funciones genéricas utilizadas por muchas aplicaciones expresándolas en forma de servicios reutilizables.
- Todo sistema puede interpretarse como un conjunto de servicios.
- Cualquier futuro cambio se realiza de manera transparente, afectado solo a las funciones y unidades afectadas.

Servicios Web

Los servicios Web son la forma más habitual de implementar SOA. Se trata de aplicaciones que utilizan estándares para el transporte, codificación y protocolo de intercambio de información. Permiten la intercomunicación entre sistemas de cualquier plataforma y se utilizan en una gran variedad de escenarios de integración dentro de las organizaciones.

Los servicios web se basan en un conjunto de estándares de comunicación, como son XML para la representación de datos, SOAP (Simple Object Access Protocol) para el intercambio de datos y el lenguaje WSDL (Web Services Description Language) para describir las funcionalidades de un servicio Web.

3.7 Seguridad

“La seguridad de la información consiste en la protección de los sistemas de información respecto al acceso no autorizado o modificación de la información en el almacenamiento, proceso y tránsito y contra la denegación de servicio para los usuarios autorizados, incluyendo aquellas medidas necesarias para detectar, documentar y contrarrestar dichas amenazas” [8].

La seguridad de la información debe garantizar:

- **Confidencialidad:** Capacidad de proporcionar acceso a usuarios autorizados y negarlo a no autorizados.
- **Integridad:** Capacidad de garantizar que la información no ha sido manipulada y de que los servicios procesan correctamente la información.
- **Disponibilidad:** Capacidad de acceder a información o utilizar un servicio siempre que lo necesitemos.

Para conseguir la seguridad en el software desarrollado, los requerimientos de seguridad deben ser parte integral del ciclo de desarrollo, desde la toma de requerimientos hasta la implantación. Cuanto más seguros sean nuestros sistemas por diseño, menos medidas de seguridad necesitaremos.

Algunas medidas de seguridad que hay que tener en cuenta en el diseño de una aplicación son las siguientes:

Autenticar al autor

Las funciones deben identificar siempre que sea posible al actor que ha desencadenado algún evento.

Autorizar al actor

Las funciones deben dar al actor identificado los derechos predefinidos bien en el sistema o bien en un sistema externo, idealmente un directorio centralizado.

Minimizar los privilegios

Se deben conceder al actor sólo los privilegios mínimos que permiten utilizar la funcionalidad autorizada.

Dejar evidencia

Debe existir evidencia, idealmente en un log sólo modificable por la aplicación, de todos los usos de servicios en el sistema que superen un determinado umbral de criticidad. Debe dejarse evidencia de todos los intentos de autenticación, tanto los exitosos como los fallidos. En el log debe incluirse al menos: usuario, dirección de origen, fecha y hora, recurso al que quería acceder y derecho que se pretendía ejercer sobre el recurso.

Garantizar la calidad y secreto de las contraseñas

El usuario normalmente seguirá el camino de mínimo esfuerzo para elegir su contraseña. Debemos evaluar la calidad de ésta, y rechazar las que no la alcancen, explicando al usuario el motivo específico del rechazo. Las contraseñas deben cambiarse periódicamente. Al introducir contraseñas, no debe aparecer en pantalla nada que indique a un posible espía ni la contraseña ni su longitud. Hay que limitar el número de posibles intentos de ingresar al sistema; en caso de que se exceda este número, el sistema debe notificar al administrador.

Prevenir los errores del usuario

Los usuarios cometen errores. Por lo tanto es aconsejable solicitar su confirmación de cualquier uso del sistema que tenga consecuencias irreversibles, repercusiones económicas, y en general cualquier funcionalidad descrita como sensible en los requerimientos.

No almacenar ni transportar información secreta en claro

En el caso de contraseñas puede utilizarse en su lugar un hash, que nos permita probar su conocimiento sin necesidad de revelarla ni de almacenarla. El tiempo de permanencia en cualquier almacenamiento intermedio, sea caché, memoria o archivos temporales debe minimizarse borrándola en cuanto sea posible.

Utilizar los mejores algoritmos de cifrado disponibles

Tanto al almacenar como al transmitir la información o enviar un mensaje, debemos utilizar los mejores algoritmos existentes en el mercado, y no algoritmos inventados por nosotros mismos. Los algoritmos de cifrado son diseñados y revisados por matemáticos expertos.

Lanzar una excepción cuando se produzca un error

En caso de un error previsto, el programa debe terminar registrando en un log la razón del error. En el caso de errores imprevistos, la información guardada en el log debe permitir investigar el fallo mientras el programa esté en desarrollo.

Cifrado

En términos de cifrado, el mensaje original recibe el nombre de texto claro, el mensaje codificado se denomina texto cifrado, el proceso de convertir el texto claro a texto cifrado se denomina cifrado y el proceso de recuperar el texto claro a partir del texto cifrado se denomina descifrado.

Para que el cifrado sea eficaz, el emisor y el receptor deberían ser las únicas partes que conozcan la forma en que se ha cifrado y descifrado el mensaje.

El cifrado ayuda a garantizar tres cosas:

- **Confidencialidad.** Sólo el receptor adecuado podrá descifrar el mensaje que se le haya enviado.
- **Autenticación.** Los mensajes cifrados que reciba habrán sido generados por una fuente de confianza.
- **Integridad.** Cuando envíe o reciba un mensaje no será alterado durante su trayecto.

Algunos mecanismos criptográficos son de un solo sentido; es decir, producen texto cifrado que no se puede descifrar. Un ejemplo de un cifrado unidireccional es el *hash*. Un hash es un número de gran tamaño, generado matemáticamente a partir de un mensaje de texto claro. Como el hash no contiene información del mensaje original, éste no se puede deducir del hash. Un hash resulta útil para verificar que alguien conoce un secreto sin tener que almacenar en realidad dicho secreto.

Seguridad basada en roles

La seguridad basada en roles permite controlar el acceso de los usuarios a los recursos en base a su identidad y su pertenencia a un grupo. De acuerdo al rol que tiene cada persona, es como se integra a un grupo.

Cada rol conlleva su propio conjunto de derechos o permisos. Por ejemplo, para el Módulo de Consulta de Proyectos del SIRF que cubre esta tesis, una de las tareas asignadas al rol de director es revisar la información de todos los proyectos existentes. El rol de académico garantiza al académico el derecho a ver la información de sus propios proyectos, pero no los de ningún otro.

Todo esto se logra utilizando los mecanismos de *autenticación* y *autorización*. La autenticación verifica que el usuario es quien dice ser y la autorización verifica que el usuario tiene el permiso para llevar a cabo una determinada actividad.

3.8 Bibliografía

- [1] <http://www.bgr.com/2012/03/20/app-developers-continue-to-choose-ios-android-and-blackberry-fall-behind/>
Agosto 2012
- [2] Ingeniería de Software, Teoría y práctica. Pfleeger, Shari Lawrence. Prentice Hall. Argentina. 2002.
- [3] Ingeniería del Software. Sommerville, Ian. Pearson Educación. Séptima Edición. Madrid. 2006.
- [4] <http://bibliotecakatherinebrecht.blogspot.mx/2011/04/metodologia-rup.html>
Junio 2012
- [5] Flexibilidad con Scrum, Principios de diseño e implantación de campos de SCRUM. Palacio, Juan. 2008.
- [6] <http://agilemanifesto.org/iso/es/manifesto.html>
Agosto 2012
- [7] <http://www.slideshare.net/jcampo/calidad-de-software>
Junio 2012
- [8] INFOSEC Glossary <https://infosec.uga.edu/glossary.php>
Mayo 2012
- [9] Ingeniería del Software, Un enfoque práctico. Roger S. Pressman. Mc Graw Hill. Sexta Edición. México. 2006.

3.7 Seguridad

- [10] Client-Server Internet Environments, The Modern IT Infrastructure. Amjad Umar, Ph.D., Prentice Hall, USA, 1997.
- [11] Seguridad para Microsoft Visual Basic .NET. Ed Robinson, Mc Graw Hill, España, 2003.
- [12] Seguridad de la información. Vicente Aceituno Canal, Creaciones Copyright, España, 2004.
- [13] Business Process Management, Fundamentos y Conceptos de implementación. Hitpass, Bernhard. BPM Center. Chile. 2012.
- [14] <http://www.slideshare.net/soreeracioneygarcia/consids-basicas-para-el-desarrollo-de-aplicaciones-mviles#btnNext>
Mayo 2012
- [15] <http://www.soaagenda.com/journal/articulos/arquitectura-orientada-a-servicios/>
Julio 2012
- [16] <http://naslyuribe0507ita.blogspot.mx/2010/11/hay-distintos-tipos-de-ciclos.html>
Julio 2012
- [17] Guevara Soriano, Anahid. “Dispositivos Móviles”. En: Punto Seguridad, Seguridad en TIC, Agosto 2010.
- [18] Scrum Manager, Gestión de Proyectos. Palacio, Juan. Octubre 2009.

Capítulo 4

Análisis de la aplicación

4.1 Descripción del problema

Actualmente, los investigadores que cuentan con proyectos registrados en el SIRF hacen uso del módulo de consulta de proyectos para revisar la información referente al estado actual de sus proyectos en cuestiones financieras.

Dicho módulo consiste en una aplicación de escritorio para la plataforma Microsoft Windows desarrollada con el lenguaje de programación Visual Basic 6 y está basada en la arquitectura cliente-servidor. Cada usuario debe ejecutar la aplicación desde su equipo de cómputo para poder acceder a la información de la base de datos a través de ésta.

El módulo de consulta de proyectos del SIRF, al ser una aplicación de escritorio, presenta algunas carencias como son:

- Instalación y actualización en cada equipo de cómputo.
- Dependencia del sistema operativo y sus capacidades.
- Conexión forzosa a la red interna del instituto de ingeniería.
- Implementación adicional de una VPN para su uso externo al instituto.
- Otros requerimientos de software como conectores a bases de datos, librerías, etc.

Los usuarios del instituto han hecho algunas observaciones con respecto a su experiencia en la utilización de esta aplicación. Ellos comentan que:

- Es complicada de utilizar.
- El color de la letra es muy clara y no es visible para personas con afecciones visuales.
- En el caso del director, los subdirectores o coordinadores, comentan que no pueden ver la información de los jefes de proyecto a su cargo, necesario para tomar decisiones.
- La aplicación no se ajusta con la resolución que algunas personas utilizan.
- La mayoría de los usuarios olvidan su usuario y contraseña del SIRF, ya que son distintos a los que utilizan normalmente para las otras aplicaciones dentro del Instituto.

4.2 Propuesta de solución

Considerando las carencias y comentarios de los usuarios de la aplicación de consulta de proyectos del SIRF, surgió la idea de tomar las anteriores como áreas de oportunidad y así crear una aplicación de tipo Web del mismo módulo, así como también una aplicación para dispositivos móviles. Estas aplicaciones cubrirán tanto los requerimientos ya existentes como los nuevos requerimientos que surgieron a partir de la experiencia de utilizar la aplicación de escritorio. Se pretende que estas aplicaciones replacen al sistema actual.

4.2 Propuesta de Solución

Una aplicación Web presenta las siguientes ventajas:

- No requiere de instalación.
- No se requiere un sistema operativo específico para ejecutar la aplicación, tan solo contar con un navegador web.
- No requieren de actualización por parte del cliente.
- Puede ejecutarse desde cualquier lugar en cualquier computadora o dispositivo móvil que tenga internet y un navegador web.
- Fácil de utilizar.

Las desventajas de una aplicación Web son:

- Conexión forzosa a internet.
- Dependencia a plug-ins dependiendo del navegador que se utilice.

Una aplicación móvil presenta las siguientes ventajas:

- Consultas desde cualquier lugar en cualquier momento mientras se disponga de una conexión a internet.
- Interfaz amigable y familiar al usuario.

Sus desventajas son:

- Conexión forzosa a internet.
- Su desarrollo es costoso, ya que se requiere desarrollar una aplicación nativa para cada plataforma móvil.

4.2.1 Análisis de recursos disponibles

Antes de comenzar un proyecto, se debe estimar y planificar el esfuerzo, los recursos y el tiempo que se requerirán para la realización del mismo.

Para esto se establece un plan del proyecto. Es un documento usado como guía para la ejecución y control del proyecto. Este documento contiene la relación de recursos humanos, recursos de hardware y de software, tiempo estimado, costo estimado y un plan de manejo de riesgos.

4.2.1.1 Recursos humanos

Para desarrollar un proyecto de software se debe contar con un equipo de trabajo en el cual se repartan las distintas tareas que conllevan a la realización del proyecto. Un equipo de alto rendimiento es de un tamaño reducido que no sobrepase los 10 integrantes, debe constar de un objetivo compartido, debe haber una comunicación efectiva entre los integrantes y una confianza mutua. Cuando un proyecto es pequeño, una o dos personas pueden realizar todas las tareas de ingeniería de software. Es muy importante que cada persona esté capacitada para cubrir su rol.

El área de Ingeniería de Software de la Coordinación de Sistemas de Cómputo del Instituto de Ingeniería se encarga de desarrollar muchos de los sistemas que utiliza toda la comunidad del instituto. Dentro de esta área trabajan en conjunto ingenieros en computación en la realización de estos sistemas. Para el proyecto que concierne a esta tesis, los recursos humanos disponibles pertenecientes a esta área fueron:

Rol	Personal
Director del Proyecto	Ing. Marco Ambríz Maguey
Líder del proyecto y desarrollador	Ing. Julio Alfonso de León Razo
Desarrollador	Leticia Mendoza Zamores

Tabla 4.1 Recursos Humanos

4.2.1.2 Hardware

El área de Ingeniería de Software cuenta con la infraestructura necesaria para el desarrollo de nuevos sistemas, así como también, para dar mantenimiento a los sistemas existentes.

Se requirió de componentes de hardware que soportaran las herramientas de software necesarias para el desarrollo de la aplicación web y móvil de consulta de proyectos del SIRF.

En cuanto a los recursos de hardware, se contó con:

Recurso	Características
Un Servidor de Bases de Datos	
Un Servidor Web	
Dos Computadoras de escritorio	Sistema operativo Windows 7 Profesional, procesador Core i7 a 3.4 GHz y 16 Gb en RAM
Dos Computadoras MAC	Sistema operativo OS X Lion 10.7.4, procesador Core i7 a 2.8 GHz y 8 Gb en RAM
Dispositivos móviles	Sistema operativo Windows Phone e iOS

Tabla 4.2 Recursos de Hardware

4.2.1.3 Software

En cuanto a las herramientas de software, se hizo uso de las siguientes:

- Manejador de Bases de Datos MySQL Workbench
- Manejador de Bases de Datos Microsoft SQLServer 2008
- Herramienta de desarrollo Microsoft Visual Studio 2010 con .NET Framework
- Windows Phone SDK 7.1
- Herramienta de desarrollo Appcelerator Titanium
- SDK de iOS
- Herramienta de desarrollo XCode

4.2.2 Requerimientos

La ingeniería de requerimientos es el proceso de describir, analizar, documentar y verificar lo que el cliente quiere, cómo interactuarán los usuarios finales con el software y abarca el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio.

“Un requerimiento se define como una condición o capacidad que un usuario necesita para poder resolver un problema o lograr un objetivo”⁸.

Los requerimientos se clasifican en requerimientos del usuario y requerimientos del sistema. Los requerimientos del usuario son declaraciones en lenguaje natural y en diagrama de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar. Los requerimientos del sistema establecen con detalle las funciones o servicios y restricciones operativas del sistema.

Los requerimientos del sistema en cuestión se describen a continuación:

Aplicación Web

Se requiere de una aplicación web del módulo de consulta de proyectos del SIRF que tome como base la aplicación de escritorio existente.

Cuando un usuario ingrese a la aplicación, éste debe ser autenticado con ayuda del Active Directory del Instituto, con la finalidad de que el usuario teclee el usuario y contraseña que utiliza generalmente para acceder a su correo electrónico o a otras aplicaciones del instituto, para que así no haya confusiones con las contraseñas. El sistema debe identificar el tipo de usuario que se ha autenticado. Existen cuatro tipos de usuario, los cuales se describen a continuación:

⁸ IEEE <http://www.ieee.org/index.html>

0. **Académico (Jefe de proyecto):** Es el usuario autorizado para ver solamente la información de sus propios proyectos y poder hacer uso del apartado App Móvil.
1. **Coordinador:** Es el usuario que, además de poder ver la información de sus proyectos, está autorizado para acceder a la información de los proyectos de los investigadores y técnicos académicos, pertenecientes a su coordinación. Además debe poder hacer uso del apartado App Móvil.
2. **Subdirector:** Es el usuario que, además de poder ver la información de sus proyectos, está autorizado para acceder a la información de los proyectos de los investigadores y técnicos académicos que pertenecen a su subdirección. Además debe poder hacer uso del apartado App Móvil.
3. **Administrador:** Es la persona autorizada para poder ver la información de los proyectos de todos los investigadores y técnicos académicos, para fines administrativos de la dependencia, así como inclusive detectar errores en la información capturada.
4. **Director:** Es la persona que además de poder ver la información de sus proyectos, puede acceder a la información de todos los investigadores y técnicos académicos que cuenten con proyectos dentro del instituto. Además debe poder hacer uso del apartado App Móvil.

Una vez que el usuario ha sido autenticado e identificado, debe ser autorizado para poder acceder a los distintos apartados de los que debe constar la aplicación. Éstos son:

- **Mis proyectos:** Muestra la lista de los proyectos del usuario que ingresa a la aplicación.
- **Búsqueda por Coordinación:** Realiza una búsqueda, por coordinación, de algún usuario para consultar sus proyectos y la información de cada uno de éstos.
- **Búsqueda por Investigador:** Realiza la búsqueda de algún usuario, de entre todos los usuarios registrados en el SIRF, para consultar sus proyectos y la información de cada uno de éstos.
- **Búsqueda por Proyecto:** Realiza la búsqueda de algún proyecto, de entre todos los proyectos registrados en el SIRF, para consultar su información.
- **App Móvil:** Permite configurar la contraseña con la cual se podrá acceder a la aplicación móvil. Ésta contraseña deberá cifrarse mediante algún algoritmo de hash y debe cumplir con las siguientes características: debe formarse por al menos 8 caracteres, debe contener como mínimo una letra mayúscula, una minúscula y un número. Posteriormente se debe almacenar en la base de datos para su utilización posterior.

La siguiente tabla muestra los apartados a los que cada tipo de usuario podrá acceder:

Tipo de Usuario \ Apartado	Mis Proyectos	Búsqueda por Coordinación	Búsqueda por Investigador	Búsqueda por Proyecto	App Móvil
Administrador		x	x	x	
Académico	x				x
Coordinador	x	x			x
Subdirector	x	x			x
Director	x	x	x	x	x

Tabla 4.3 Apartados correspondientes a cada tipo de usuario

4.2 Propuesta de Solución

Para cada proyecto que se consulte, se mostrará la siguiente información:

- Datos generales del proyecto
- Información sobre las asignaciones y el presupuesto por rubros
- Desglose de la cobranza
- Desglose de los gastos
- Reporte con la información relevante del proyecto, el cual podrá ser exportado a formato .pdf o .xls para su impresión y/o edición

La aplicación debe tener un apartado de **contacto**, donde se mostrará la información de las personas a las que se les pueden hacer llegar dudas o comentarios sobre la información consultada o sobre el funcionamiento de la aplicación.

Se requiere que la aplicación web detecte si se ha ingresado a ella desde una computadora o desde un dispositivo móvil.

Si detecta que se ha ingresado a ella desde una computadora, deberá mostrar la versión completa, que es la que ya se ha descrito.

Si de lo contrario, detecta que se ha ingresado desde un dispositivo móvil (navegador web móvil), deberá mostrar una versión ligera, la cual se describe a continuación.

Aplicación Web en su versión móvil

- Esta es un fragmento de la aplicación web completa cuya funcionalidad es la misma a la de las aplicaciones móviles nativas, cuya opción es necesaria ya que de esta manera es posible abarcar más dispositivos móviles para los cuales no se ha desarrollado una aplicación nativa. **Debido a que existe una gran diversidad de dispositivos con distintas plataformas móviles, la aplicación web móvil podrá ser consultada en cualquier dispositivo que cuente con un navegador web.**

Aplicación Móvil Nativa

Se requiere de una aplicación para cada una de las plataformas siguientes: iOS (iPhone y iPad) y Windows Phone.

Esta aplicación tendrá una autenticación en la cual, el usuario será el mismo con el que se ingresa a la aplicación web, y la contraseña será la que se configura previamente en el apartado App Móvil de la aplicación web.

Una vez que se ingresa a la aplicación móvil, se deberá desplegar el listado de los proyectos del usuario que se ha autenticado. Luego, para cada proyecto, se mostrará un menú con las siguientes opciones:

- Datos generales
- Asignaciones
- Cobranza
- Gastos

La aplicación debe tener un apartado de contacto, donde se mostrará la información de las personas a las que se les pueden hacer llegar dudas o comentarios sobre el funcionamiento de la aplicación.

La aplicación de cada plataforma móvil, debe ser gratuita y estar al alcance de los investigadores y técnicos académicos del Instituto de Ingeniería para su descarga e instalación.

Es importante mencionar que una aplicación nativa puede hacer uso de los recursos del hardware del dispositivo, por lo que su funcionamiento es más eficiente que cuando se trabaja con una aplicación web para móviles.

4.2.3 Elección de plataformas

En el capítulo 3 se hizo mención de algunas de las plataformas de desarrollo de aplicaciones. Para el desarrollo de la aplicación web de Consulta de Proyectos del SIRF, el Web Service y la aplicación móvil para Windows Phone se eligió el entorno de desarrollo Visual Studio.

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Los lenguajes de programación Visual Basic, Visual C# y Visual C++ utilizan el mismo entorno de desarrollo integrado (IDE) que habilita el uso compartido de herramientas facilitando la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML.

Para el desarrollo de la aplicación móvil para iPhone y iPad se eligió el entorno de desarrollo Appcelerator Titanium en conjunto con el SDK de iOS.

Appcelerator Titanium es un framework basado en javascript para desarrollar aplicaciones móviles (iOS y Android). Contrario a otras plataformas (como phonegap), Titanium genera aplicaciones nativas, por lo que se ejecutan con el desempeño y ventajas de una aplicación nativa. Básicamente, desde el ambiente de desarrollo de Titanium se crea la interfaz gráfica y se programa el comportamiento en javascript, y en base a esto el motor de Titanium genera un proyecto nativo en Xcode (en el caso de iOS) o un proyecto nativo de Android. Ya con esto, se puede compilar utilizando las herramientas correspondientes para generar ejecutables nativos para cada plataforma. Además de las ventajas de desempeño que ofrece el que se generen aplicaciones

4.2 Propuesta de Solución

nativas, otra ventaja es que estas aplicaciones serán aceptadas en el Apple App Store sin problemas.

4.2.4 Planeación y estimación de tiempos y costos.

El costo del software y el tiempo de desarrollo son muy difíciles de estimar con precisión, pues hay muchas variables (humanas, técnicas, ambientales, políticas) que pueden afectar el costo final del software, el tiempo y el esfuerzo aplicado a desarrollarlo.

La estimación de un proyecto de software se manifiesta en varios factores:

- El tamaño del producto a construir
- El esfuerzo humano y monetario que conllevará
- El tiempo que llevará su desarrollo
- La estabilidad de los requisitos del producto
- El hardware y software que se requieren

Existen varias técnicas de estimación de proyectos de software. Una de ellas es la llamada técnica de descomposición, la cual se basa en la descomposición del problema viéndolo como un conjunto de problemas más pequeños (divide y vencerás).

Para el proyecto en cuestión, la planeación de tiempos se hizo de la siguiente manera:

Tarea	Tiempo estimado
Creación de consultas a la BD del SIRF	2 meses
Creación de procedimientos almacenados	1 mes
Creación del Web Service	10 días
Creación de la Aplicación Web	5 meses
Creación de la aplicación móvil para Windows Phone	1 mes
Creación de la aplicación móvil para iPhone y iPad	2 meses
Creación de la Base de datos en SQLServer para la autenticación y autorización de usuarios	20 días
Tiempo total:	12 meses

Tabla 4.4 Estimación de tiempos

La estimación del costo de un proyecto consiste en estimar los costos de los elementos necesarios para llevar a buen fin el mismo, tal es el caso de recursos humanos, materiales e indirectos (Agua, Luz, Renta del mobiliario, papelería, etc.).

Para el proyecto en cuestión, se consideran los costos en forma de porcentajes, debido a que los recursos fueron proporcionados y absorbidos por el IINGEN:

Recurso	[%]
Recursos humanos	40
Hardware	40
Software	15
Otros costos	5

Tabla 4.5 Estimación de costos

4.3 Bibliografía

[1] Ingeniería del Software, Un enfoque práctico. Roger S. Pressman. Mc Graw Hill. Sexta Edición. México. 2006.

[2] Gestión de recursos Humanos en Proyectos Informáticos. <http://linki.es/1pzz>
Octubre 2012

[3] [http://msdn.microsoft.com/es-es/library/fx6bk1f4\(v=vs.100\).aspx](http://msdn.microsoft.com/es-es/library/fx6bk1f4(v=vs.100).aspx)
Noviembre 2012

[4] <http://www.sg.com.mx/content/view/1089>
Noviembre 2012

Capítulo 5

Diseño de la Aplicación

5.1 Arquitectura Cliente- Servidor

La arquitectura planteada para la aplicación de consulta de proyectos del SIRF se muestra en la figura 5.1. Se trata de una arquitectura de tres capas, las cuales se describen a continuación:

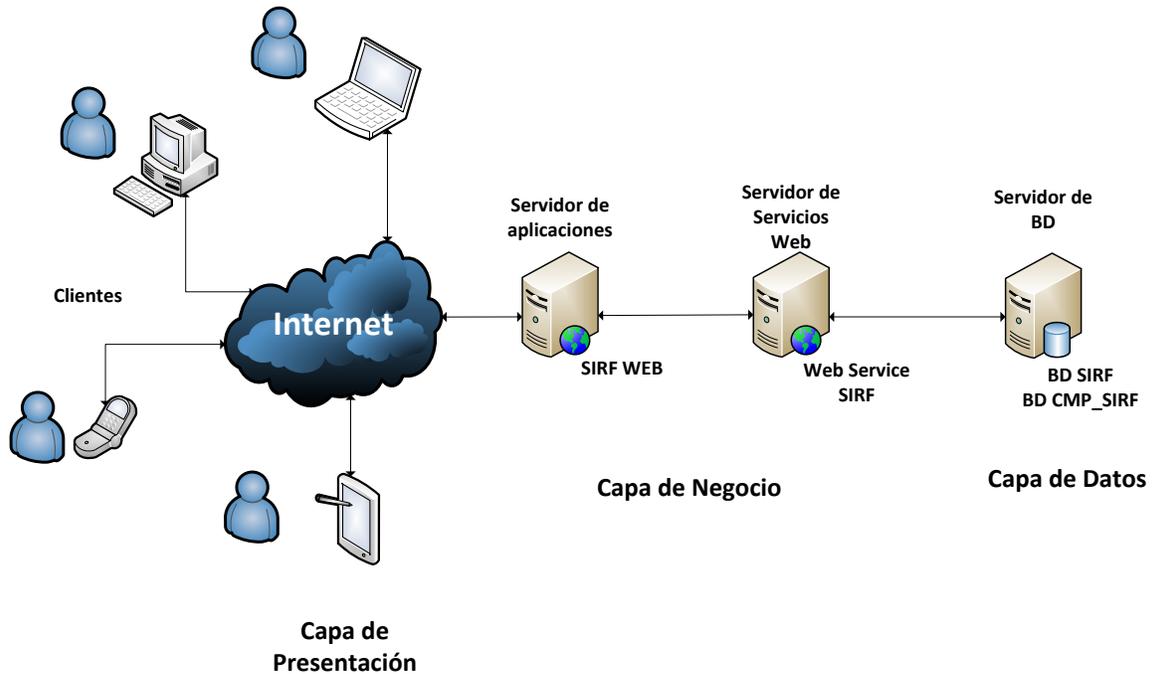


Figura 5.1 Arquitectura de la aplicación de consulta de proyectos del SIRF

Capa de Presentación

La capa de presentación la compone la interfaz de usuario. Es el medio por el cual los usuarios se comunican con el sistema. Para diseñar una interfaz de usuario, se plantean prototipos que ayuden a visualizar su funcionalidad y las interacciones con el usuario, para su posterior desarrollo. En este caso, se tomó en parte la aplicación de escritorio del SIRF como prototipo, pues los usuarios ya se encuentran familiarizados con la forma de distribución de la información. Sin embargo, se propuso mejorarla diseñando una interfaz más amigable y fácil de utilizar.

5.1 Arquitectura orientada a servicios (SOA)

Figura 5.2 Aplicación de escritorio del módulo de consulta de proyectos del SIRF

Capa de Negocio

La capa de negocio está compuesta por el Web Service y también por las funciones lógicas de la aplicación las cuales definen el comportamiento del sistema. Esta capa se comunica con la capa de presentación para recibir las solicitudes por parte del usuario, validarlas, procesarlas y con la capa de datos para obtener los datos requeridos y enviarlos a la capa de presentación para ser mostrados.

Capa de Datos

La capa de datos la compone el servidor de base de datos, pues es aquí donde se encuentran las bases de datos donde se almacena la información y se obtiene la misma.

Hay dos bases de datos que son requeridas para este sistema: La base de datos del SIRF y una base de datos con la información del personal del instituto de ingeniería.

La Base de Datos del SIRF contiene toda la información referente a los proyectos de cada investigador. Se trata de una Base de Datos de MySQL, creada por Patronato Universitario.

Para fines de la aplicación del módulo de consulta de proyectos del SIRF, se utilizaron algunas tablas de esta base de datos. El modelo relacional que contiene a estas tablas, y el cual fue extraído del modelo completo de la base de datos del SIRF es el siguiente:

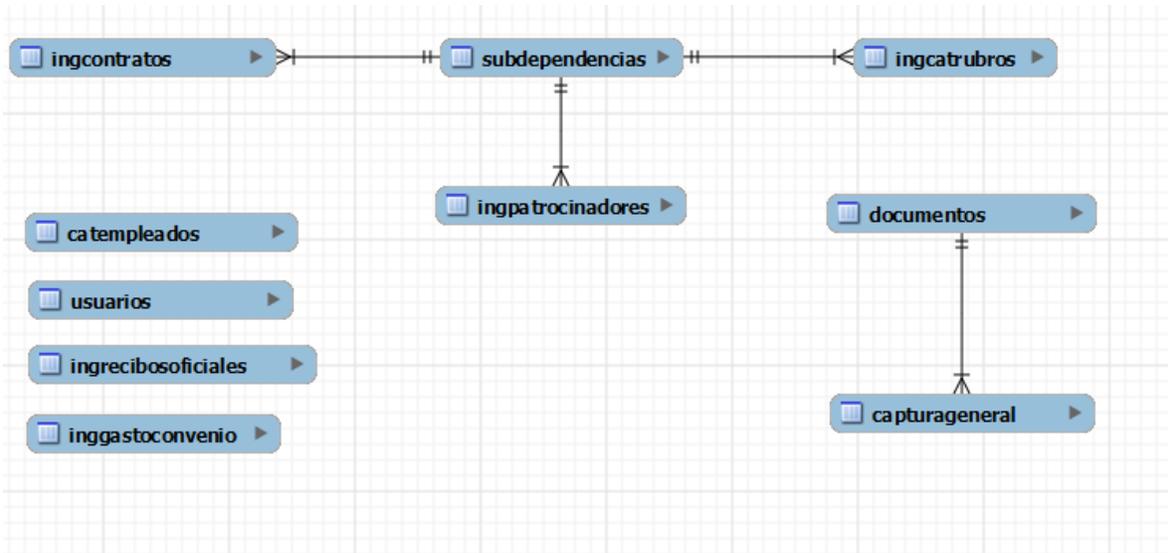


Figura 5.3 Modelo Relacional del módulo de consulta de proyectos del SIRF

La base de datos del personal contiene la información del personal del instituto, tal como el cargo que tiene, su RFC, su Login, etc. Se trata de una base de datos de Microsoft SQL Server y fue creada por la coordinación de sistemas de cómputo.

La información de esta base de datos será útil para realizar la autenticación y autorización de los usuarios. También en ella se almacenará la contraseña para acceder a la aplicación móvil.

El modelo relacional de esta base de datos es el siguiente:

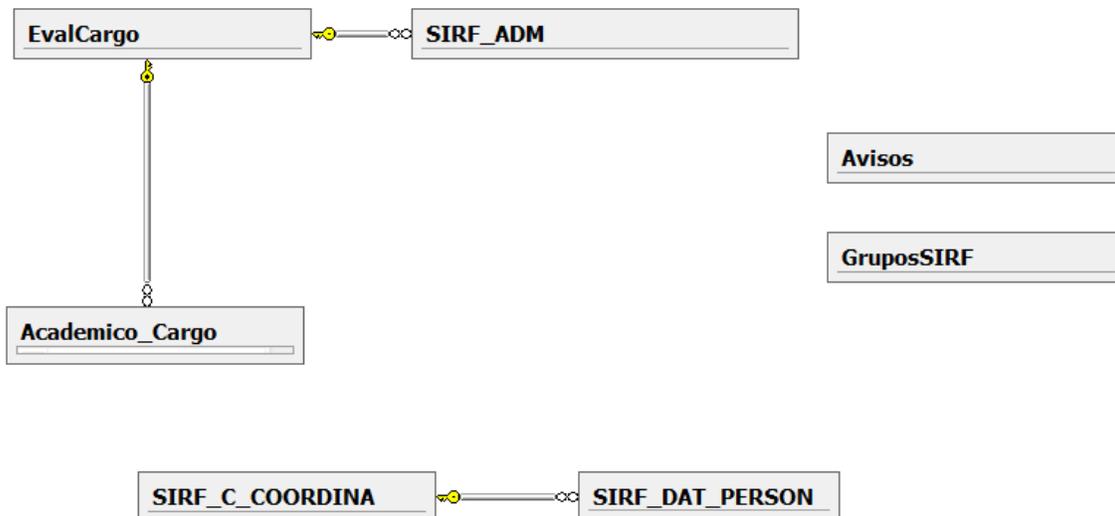


Figura 5.4 Modelo Relacional de la base de datos del personal del IINGEN

5.2 Arquitectura Orientada a Servicios (SOA)

El módulo de consulta de proyectos del SIRQ constará de una aplicación Web, diversas aplicaciones móviles, y hará uso de dos bases de datos. Es necesario recurrir a la implementación de Servicios Web (Web Services) que se encarguen de comunicarse con las Bases de Datos y de extraer la información requerida por cada una de las aplicaciones.

El Web Service servirá de intermediario entre las aplicaciones y las bases de datos. Como se explicó en el capítulo 3, los servicios Web permiten la intercomunicación entre sistemas de cualquier plataforma, pues facilitan la integración de aplicaciones independientemente del lenguaje con el que sean desarrolladas. La arquitectura SOA para la aplicación se muestra a continuación.

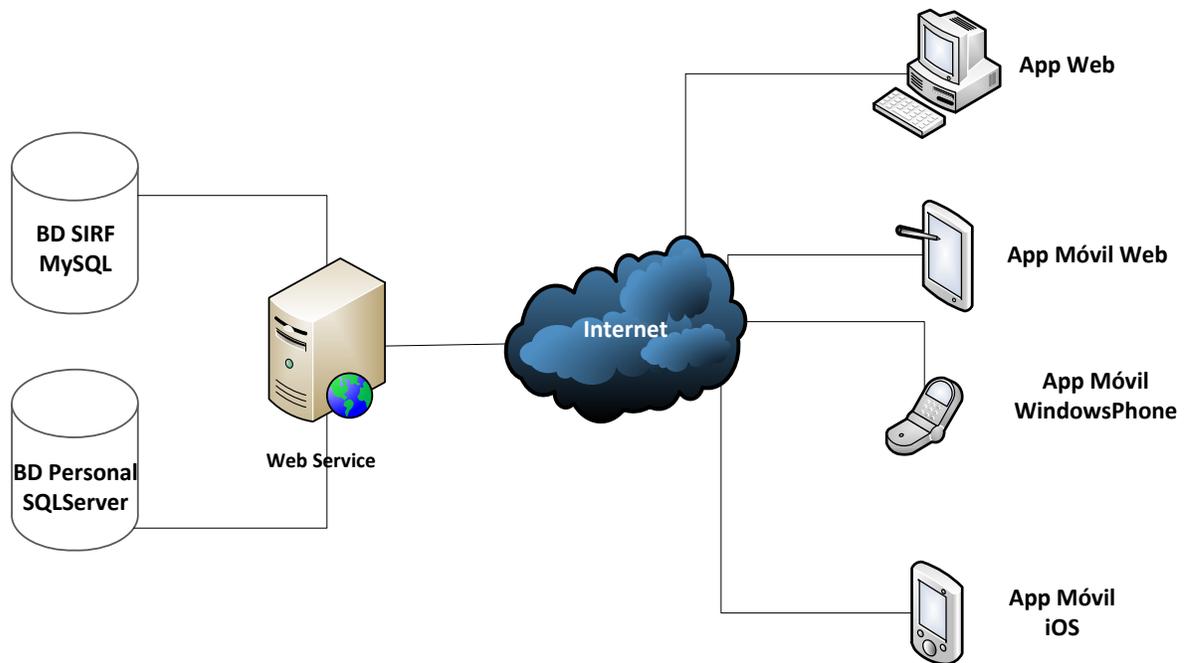


Figura 5.5 Arquitectura SOA de la aplicación de consulta de proyectos del SIRQ

5.3 Diagramas UML

UML (Unified Modeling Language) es el lenguaje de modelado universal que nos permite representar un sistema a partir de modelos.

Cuando un equipo de trabajo desarrolla un sistema y las tareas se dividen entre cada integrante, éstos deben estar comunicándose constantemente, mediante un lenguaje que todos entiendan.

UML es una herramienta para la comunicación entre los participantes en el desarrollo de un proyecto de software. Es independiente de lenguajes de programación y metodologías de desarrollo.

Utiliza una serie de diagramas o modelos, los cuales son útiles para captar y representar los requisitos de un sistema, de modo que todo el equipo de trabajo esté de acuerdo con ellos y tengan la misma visión del proyecto.

Algunos tipos de diagramas de UML son los siguientes. Cada uno nos aporta una perspectiva distinta del proyecto.

- Diagrama de clases ([Ver anexo 2](#))
- Diagrama de casos de uso
- Diagrama de estados
- Diagrama de interacción
- Diagrama de secuencia
- Diagrama de colaboraciones
- Diagrama de actividades

5.3.1 Casos de Uso

Un diagrama de casos de uso representa la forma en cómo un usuario (Actor) interactúa con el sistema. Su ventaja principal es la facilidad para interpretarlos, por lo que son de gran utilidad en la comunicación con el cliente.

Un diagrama de casos de uso se compone de:

Actores: Son entidades externas al sistema que interactúan de manera directa con éste. Los actores representan roles que interpretan personas, periféricos u otros sistemas.

Caso de Uso: Especifica una secuencia de acciones que pueden ser efectuadas por el sistema con interacción de uno o varios actores.

5.3 Diagramas UML

Relaciones: Indican que hay una interacción del actor con el sistema para llevar a cabo el caso de uso.

Para el Sistema de Consulta de proyectos del SIRF, los actores que intervienen son los distintos tipos de usuario descritos en el capítulo 4, los cuales son:

- Director: Es el director del Instituto de Ingeniería
- Subdirector: Son los subdirectores del Instituto de Ingeniería
- Coordinador: Son los coordinadores del Instituto de Ingeniería
- Académico (Jefe de Proyecto): Son los académicos que cuentan con proyectos en el SIRF.
- Administrador: Son miembros de la CSC y de la Secretaría Administrativa del instituto autorizados.

El diagrama de casos de uso de la aplicación de Consulta de Proyectos del SIRF es el siguiente:

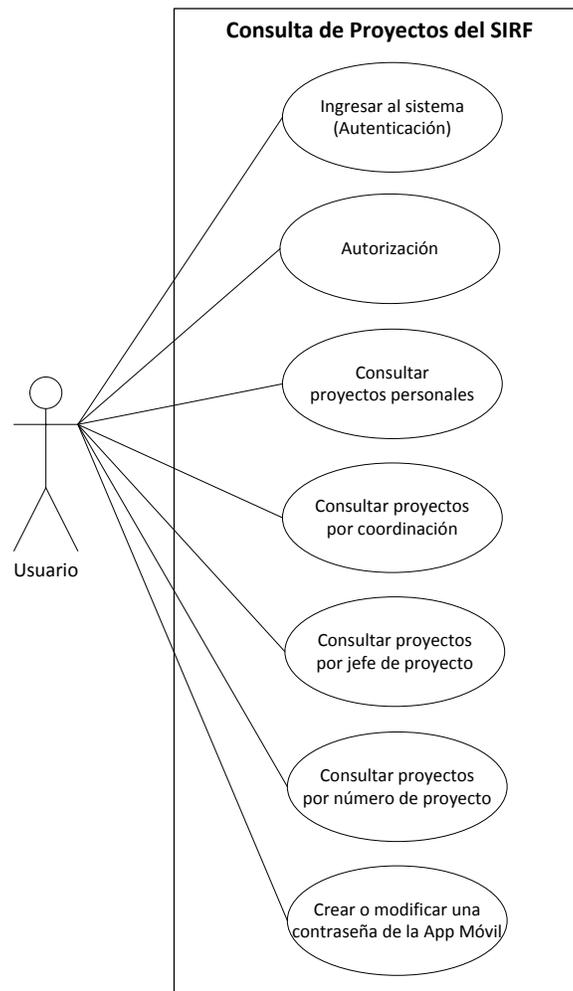


Figura 5.6 Diagrama de casos de uso de la aplicación de consulta de proyectos del SIRF

A continuación se detalla cada caso de uso:

Caso de Uso	Consultar proyectos personales
Actor(es)	Pueden ser: Director, Subdirector, Coordinador o Jefe de Proyecto
Precondición	Que el usuario haya sido autenticado y autorizado
Post-condición	Ninguna
Resumen	Consultar la información de los proyectos que tiene a cargo la persona que ingresa al sistema

Tabla 5.1 Caso de uso: Consultar Proyectos Personales

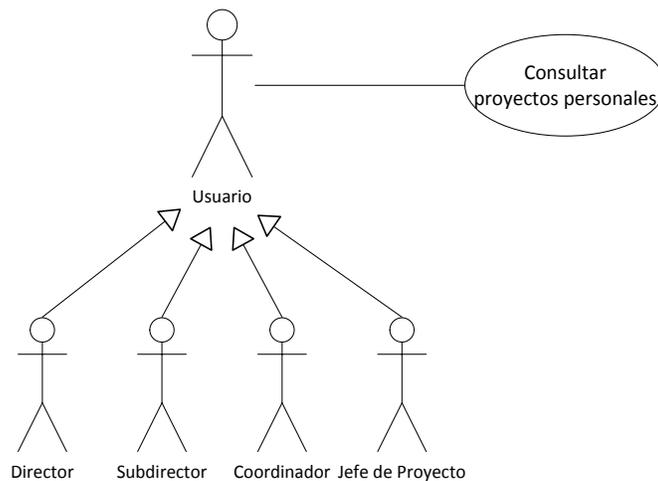


Figura 5.7 Caso de uso: Consultar Proyectos Personales

Flujo del caso de uso:

Paso	Actor	Paso	Sistema	Excepción
1	Ingresa al sistema	2	Muestra pantalla de inicio de sesión	
3	Introduce Login y Password	4	Autentica y autoriza al usuario	E1- Datos incorrectos
		5	Muestra pantalla de menú principal	
6	Selecciona "Mis Proyectos"	7	Muestra la lista de proyectos	
8	Selecciona el proyecto a consultar	9	Muestra el menú con las opciones de consulta: <ul style="list-style-type: none"> • Datos Generales • Asignaciones • Cobranza • Gastos • Reporte 	
10	Selecciona la opción a consultar	11	Muestra la información del proyecto para la opción seleccionada. Si seleccionó "Reporte", genera un reporte en PDF o .XLS con la información más relevante del proyecto.	

Caso de Uso	Consultar proyectos por coordinación
Actor(es)	Pueden ser: Director, Subdirector o Administrador
Precondición	Que el usuario haya sido autenticado y autorizado
Post-condición	Ninguna
Resumen	Consultar la información de otros proyectos buscando por coordinación.

Tabla 5.2 Caso de uso: Consultar Proyectos por coordinación

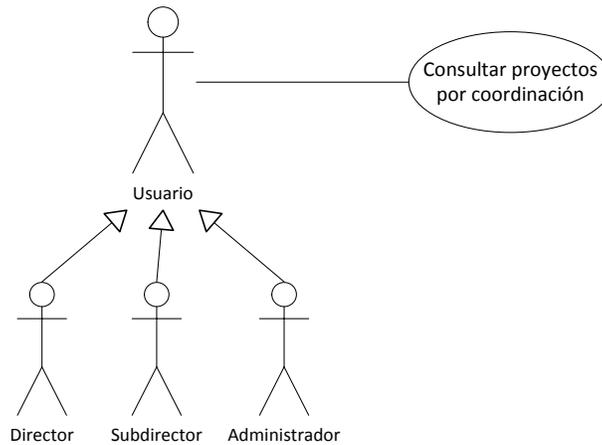


Figura 5.8 Caso de uso: Consultar Proyectos por coordinación

Flujo del caso de uso:

Paso	Actor	Paso	Sistema	Excepción
1	Ingresa al sistema	2	Muestra pantalla de inicio de sesión	
3	Introduce Login y Password	4	Autentica y autoriza al usuario	E1- Datos incorrectos
		5	Muestra pantalla de menú principal	
6	Selecciona "Búsqueda por Coordinación"	7	Muestra la lista de coordinaciones	
8	Selecciona una coordinación	9	Muestra la lista de jefes de proyecto de la coordinación seleccionada	
10	Selecciona un jefe de proyecto	11	Muestra la lista de proyectos del jefe de proyecto seleccionado	
12	Selecciona el proyecto a consultar	13	Muestra el menú con las opciones de consulta: <ul style="list-style-type: none"> • Datos Generales • Asignaciones • Cobranza • Gastos • Reporte 	
14	Selecciona la opción a consultar	15	Muestra la información del proyecto para la opción seleccionada. Si seleccionó "Reporte", genera un reporte en PDF o .XLS con la información más relevante del proyecto.	

Caso de Uso	Consultar proyectos por jefe de proyecto
Actor(es)	Pueden ser: Director, Coordinador o Administrador
Precondición	Que el usuario haya sido autenticado y autorizado
Post-condición	Ninguna
Resumen	Consultar la información de otros proyectos buscando por jefe de proyecto

Tabla 5.3 Caso de uso: Consultar Proyectos por Jefe de Proyecto

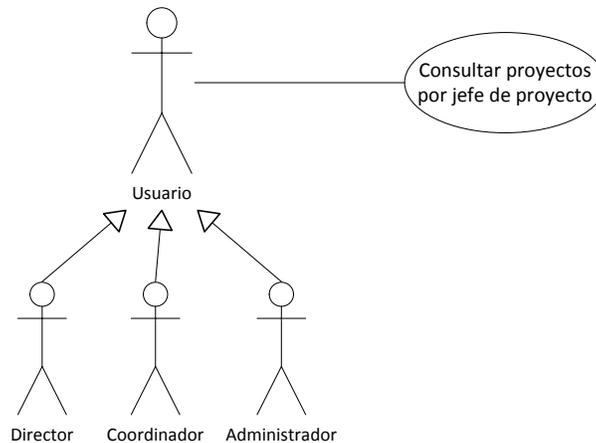


Figura 5.9 Caso de uso: Consultar Proyectos por Jefe de Proyecto

Flujo del caso de uso:

Paso	Actor	Paso	Sistema	Excepción
1	Ingresa al sistema	2	Muestra pantalla de inicio de sesión	
3	Introduce Login y Password	4	Autentica y autoriza al usuario	E1- Datos incorrectos
		5	Muestra pantalla de menú principal	
6	Selecciona "Búsqueda por Jefe de Proyecto"	7	Muestra la lista de jefes de proyecto	
8	Selecciona un jefe de proyecto	9	Muestra la lista de proyectos del jefe de proyecto seleccionado	
10	Selecciona el proyecto a consultar	11	Muestra el menú con las opciones de consulta: <ul style="list-style-type: none"> • Datos Generales • Asignaciones • Cobranza • Gastos • Reporte 	
12	Selecciona la opción a consultar	13	Muestra la información del proyecto para la opción seleccionada. Si seleccionó "Reporte", genera un reporte en PDF o .XLS con la información más relevante del proyecto.	

Caso de Uso	Consultar proyectos por número de proyecto
Actor(es)	Pueden ser: Director o Administrador
Precondición	Que el usuario haya sido autenticado y autorizado
Post-condición	Ninguna
Resumen	Consultar la información de otros proyectos buscando por número de proyecto

Tabla 5.4 Caso de uso: Consultar Proyectos por Número de Proyecto

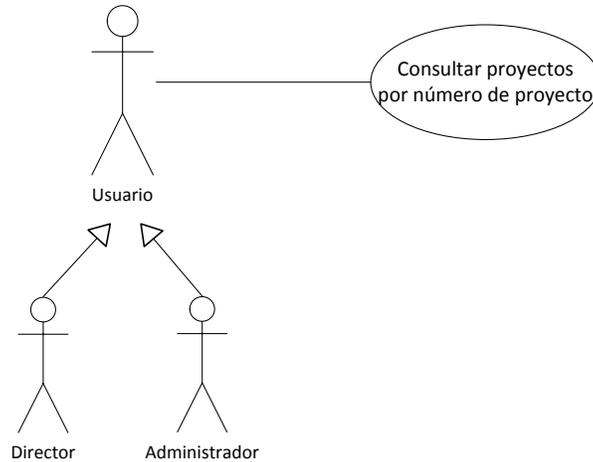


Figura 5.10 Caso de uso: Consultar Proyectos por Número de Proyecto

Flujo del caso de uso:

Paso	Actor	Paso	Sistema	Excepción
1	Ingresar al sistema	2	Muestra pantalla de inicio de sesión	
3	Introduce Login y Password	4	Autentica y autoriza al usuario	E1- Datos incorrectos
		5	Muestra pantalla de menú principal	
6	Selecciona "Búsqueda por Proyecto"	7	Muestra la lista de todos los proyectos	
8	Selecciona el proyecto a consultar	9	Muestra el menú con las opciones de consulta: <ul style="list-style-type: none"> • Datos Generales • Asignaciones • Cobranza • Gastos • Reporte 	
10	Selecciona la opción a consultar	11	Muestra la información del proyecto para la opción seleccionada. Si seleccionó "Reporte", genera un reporte en PDF o .XLS con la información más relevante del proyecto.	

Caso de Uso	Crear o modificar una contraseña para la App Móvil
Actor(es)	Pueden ser: Director, Subdirector, Coordinador o Jefe de Proyecto
Precondición	Que el usuario haya sido autenticado y autorizado
Post-condición	Que quede almacenada la nueva contraseña
Resumen	Crear o modificar una contraseña con la cual se podrá acceder a la aplicación móvil de Consulta de Proyectos del SIRF

Tabla 5.5 Caso de uso: Crear o modificar una contraseña para la App Móvil

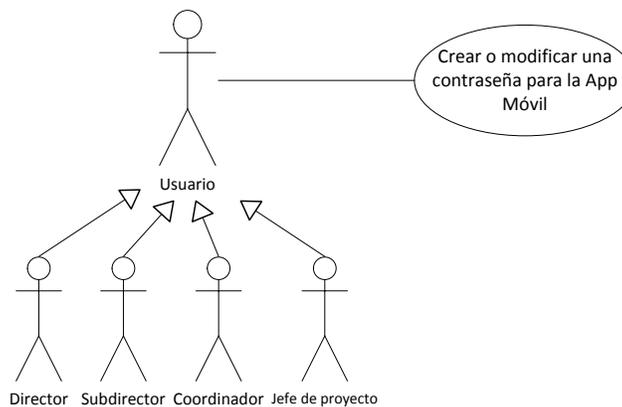


Figura 5.11 Caso de uso: Crear o modificar una contraseña para la App Móvil

Flujo del caso de uso:

Paso	Actor	Paso	Sistema	Excepción
1	Ingresa al sistema	2	Muestra pantalla de inicio de sesión	
3	Introduce Login y Password	4	Autentica y autoriza al usuario	E1- Datos incorrectos
		5	Muestra pantalla de menú principal	
6	Selecciona "App Móvil"	7	Muestra formulario para ingresar contraseña	
8	Ingresa una nueva contraseña cuya longitud sea mayor a 8 caracteres y menor a 16 y que contenga al menos una letra mayúscula, una minúscula y un número.	9	Comprueba que la contraseña cumpla con las características especificadas.	E2- La contraseña no cumple con las características especificadas.
		10	Guarda la nueva contraseña	

5.3.2 Estados

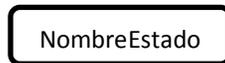
Un objeto, dependiendo del valor de sus atributos, se encontrará en un estado concreto y a lo largo del tiempo irá adquiriendo un estado distinto como respuesta a los sucesos. El diagrama de estados es una manera de representar este tipo de cambios en el sistema.

Los diagramas de estados son muy útiles para representar la interfaz de usuario, utilizando las flechas para reflejar el camino que tomará la ejecución de la aplicación.

En el diagrama de estados se tienen las siguientes constantes:

- Las diferentes situaciones en que se puede encontrar un objeto (estados).
- Qué cambios de estado son posibles (transiciones).
- Cuál es el hecho que los produce (acontecimientos).

Un estado se representa con un rectángulo con los vértices redondeados como se muestra a continuación:



Las transiciones se representan mediante flechas de punta coloreada que van del estado de salida al de llegada.

El diagrama de estados para nuestro sistema es el siguiente:

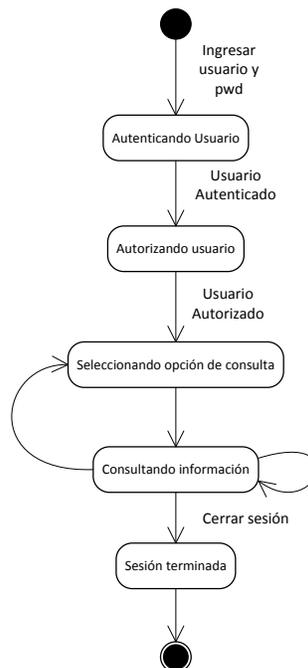


Figura 5.12 Diagrama de estados de la aplicación de Consulta de proyectos del SIRF

5.4 Diseño de la interfaz de la aplicación móvil

Una aplicación móvil debe tener un propósito claro, debe ser intuitiva y agradable de utilizar. Su contenido tiene que ser conciso y debe facilitar al usuario el acceso a la información que necesite.

Para crear una aplicación móvil se recomienda diseñarla en papel antes de utilizar las herramientas, debemos pensar en cómo navegará el usuario a través de la aplicación y cómo influenciará esto en la creación de la aplicación.

5.4.1 Plataformas

Existen dos tipos de aplicaciones móviles: aplicaciones móviles web y aplicaciones móviles nativas. Cada una tiene ventajas y desventajas de una sobre la otra. A continuación se explica para cada una, sus características, sus ventajas y desventajas.

5.4.1.1 Móvil Web

Cuando ingresamos a una página web desde nuestro dispositivo móvil, nos encontramos con que el navegador web móvil nos redirecciona a una versión móvil de esa página. Esta versión está diseñada especialmente para este tipo de dispositivos que son de tamaño reducido, pues muestran las opciones de navegación de más interés para el usuario, así como la información más concisa. Estas aplicaciones web para dispositivos móviles son accedidas desde un navegador web, en este caso, un navegador web móvil. A continuación se presentan las características, ventajas y desventajas de las aplicaciones móviles web.

Características

1. Se accede a ellas a través del navegador de internet del dispositivo.
2. En cada plataforma, la aplicación se verá casi de la misma forma.
3. Para su desarrollo se utilizan las mismas herramientas para crear cualquier aplicación web: por ejemplo HTML, CSS y JavaScript.

Ventajas

1. Compatibilidad: Se utiliza el mismo código fuente para todos los sistemas operativos móviles. Así, solo se tiene que dar mantenimiento a una sola versión.
2. Mayor número de usuarios: Se puede abarcar la mayoría de los dispositivos móviles: no sólo a los que tienen Android y iOS, sino hasta el menos común, como Bada (Samsung).

5.4 Diseño de la interfaz de la aplicación móvil

3. **Fácil acceso:** No es necesario descargar la aplicación e instalarla, únicamente se accede a la dirección URL.
4. **Siempre actualizada:** Las actualizaciones se cargan automáticamente y de inmediato, no es necesario esperar una semana o más.
5. **Desarrollo más sencillo y más rápido:** muchas veces, es mucho más fácil desarrollar y diseñar dentro de un browser utilizando HTML, CSS y JavaScript que hacer lo mismo en Java, Objective-C o JavaScript (Appcelerator, Sencha). Posicionar elementos, estilizarlos y ciertas flexibilidades son posibles en minutos con tecnologías móviles.

Desventajas

1. **No se pueden utilizar componentes nativos:** Estas aplicaciones, al ser accedidas desde el navegador, no se tiene o es poca la disponibilidad de los componentes y funcionalidades nativas del teléfono.
2. **Posibilidad de que no funcione o se presenten problemas en algunos navegadores.**
3. **No se puede utilizar offline.** Estas aplicaciones necesitan forzosamente una conexión a internet, ya que difícilmente guardan la información necesaria en el dispositivo

5.4.1.2 Móvil Nativa

Las aplicaciones móviles nativas son las que se instalan en el dispositivo y generalmente existe una para cada sistema operativo o plataforma móvil. Actualmente existe una diversa gama de dispositivos móviles con distintos sistemas operativos. El desarrollo de aplicaciones móviles se vuelve un reto, pues desarrollar una idea conlleva realizar una aplicación para cada sistema operativo móvil, si se quiere abarcar el mayor número de usuarios posibles.

A continuación se presentan las características, ventajas y desventajas de las aplicaciones móviles nativas.

Características

1. **Se descargan desde el mercado de aplicaciones móviles para el sistema operativo que tenga el dispositivo:** por ejemplo el AppStore (iOS) o el MarketPlace (WindowsPhone).
2. **Es una aplicación distinta por cada sistema operativo:** por ejemplo: una para Android, una para WindowsPhone y otra para iOS (en muchos casos una para iPhone y otra para iPad).
3. **Se desarrollan con lenguajes como Java para Android, C# para WindowsPhone, u Objective-C para iOS,** aunque hay alternativas como Appcelerator que permiten desarrollar aplicaciones multiplataforma (cross-platform) utilizando JavaScript.
4. **Se obtiene una mejor experiencia de usuario**

Ventajas

1. Puede utilizar funcionalidades nativas del dispositivo como la cámara, el acelerómetro, almacenamiento interno, etc.
2. Mejor experiencia de usuario, ya que generalmente llegan a desempeñarse mejor que las aplicaciones web y presentan una interfaz más amigable.
3. Hay un mercado para estas aplicaciones en las cuales las personas pueden encontrarlas y descargarlas, por ejemplo el AppStore o el MarketPlace. Aquí, las aplicaciones pueden ser puestas a la venta.
4. Puede utilizarse en modo OffLine, es decir, sin necesidad de estar conectado a internet, en caso que la APP almacene datos previamente consultados en OnLine.

Desventajas

1. Son más costosas de desarrollar pues se necesita invertir mucho más tiempo en el desarrollo de una versión por cada plataforma.
2. Mayor costo de mantenimiento, pues mantener varias versiones de la aplicación es más costoso.
3. Subir una aplicación al mercado es tedioso, ya que puede llevar hasta más de una semana el proceso de certificación. De igual forma, si se le agrega una nueva funcionalidad o se descubre un bug y se arregla, hay que esperar otra semana para que los responsables del mercado aprueben esos cambios.
4. Aun cuando hay desarrollo multiplataforma, no abarca todas las plataformas existentes, por ejemplo: Appcelerator permite desarrollar para Android, iPhone e iPad, pero se tendrá que desarrollar aparte la aplicación para WindowsPhone y Blackberry.

5.5 Diseño de la interfaz de la aplicación Web (ASPX)

Como se mencionó en el capítulo 4, se eligió la plataforma Visual Studio para el diseño e implementación de la aplicación Web. Con el Framework de ASP .Net es posible crear el diseño de la aplicación y la lógica de programación por separado.

El diseño se crea en páginas aspx, las cuales son soportados por un archivo de código subyacente aspx.cs el cual contiene la lógica.

5.5.1 Para pantalla de conmutador

En esta etapa se definieron las distintas pantallas con las que contará la aplicación web, las cuales se muestran en el siguiente mapa de navegación:



Figura 5.13 Mapa de Navegación de la Aplicación Web de Consulta de proyectos del SIRF



Figura 5.13-a Desglose del apartado: Información del Proyecto

También se implementó el diseño de cada una de estas pantallas.

INSTITUTO DE INGENIERÍA UNAM **Consulta de proyectos SIRF**

Contacto Usuario: Noyola Robles Adalberto
Cerrar Sesión

- Mis Proyectos**
En este apartado puede consultar la información de sus proyectos personales.
- Búsqueda por Coordinación**
En este apartado puede consultar la información de otros proyectos por coordinación.
- Búsqueda por Proyecto**
En este apartado puede consultar la información de otros proyectos por número de proyecto.
- Búsqueda por Jefe de Proyecto**
En este apartado puede consultar la información de otros proyectos por jefe de proyecto.
- App Móvil**
En este apartado puede configurar una nueva contraseña para acceder a la aplicación móvil del SIRF.

SIRF Web Versión 1.0 Instituto de Ingeniería, UNAM
La actualización de la información de los proyectos del SIRF está a cargo de Secretaría Administrativa

Figura 5.14 Diseño de la página principal (Para el caso del Director)

5.5.2 Para pantalla de dispositivo móvil

En el caso de la versión móvil de la aplicación web, solo se podrán consultar los proyectos personales. El usuario debe autenticarse para poder tener acceso a ésta.

Las distintas pantallas con las que contará la aplicación web móvil son las siguientes:

5.5 Diseño de la interfaz de la aplicación web (ASPX)



Figura 5.15 Mapa de Navegación de la Aplicación Web Móvil de Consulta de proyectos del SIRF



Figura 5.16 Diseño de la página de Inicio de Sesión de la aplicación móvil web

5.6 Bibliografía

[1] DUM: Desarrollo Unificado con Métrica. José Ignacio Peláez Sánchez. Universidad de Málaga.

[2] Ingeniería del Software. Benet Campderrich Falgueras. Editorial UOC. 2003. Barcelona.

[3] Ingeniería del Software. Sommerville, Ian. Pearson Educación. Séptima Edición. Madrid. 2006.

Capítulo 6

Desarrollo de la Aplicación

6.1 Programación de WebServices

Como se explicó en el capítulo anterior, se pensó en la implementación de un Web Service que realice la tarea de conectarse a las bases de datos y obtener o almacenar la información requerida por cada una de las aplicaciones del módulo de consulta de proyectos del SIRF.

Primero que nada, se crearon procedimientos almacenados en ambas bases de datos. Un procedimiento almacenado es una unidad de código compuesta por una o más sentencias SQL, puede recibir o regresar parámetros y se almacena en un servidor para su ejecución posterior.

Se creó un procedimiento almacenado por cada consulta que requiere el módulo de consulta de proyectos del SIRF para mostrar la información requerida. El Web Service mandará a ejecutar los procedimientos almacenados cuando las aplicaciones se lo soliciten.

El conjunto de procedimientos almacenados creados en cada base de datos se muestran a continuación:

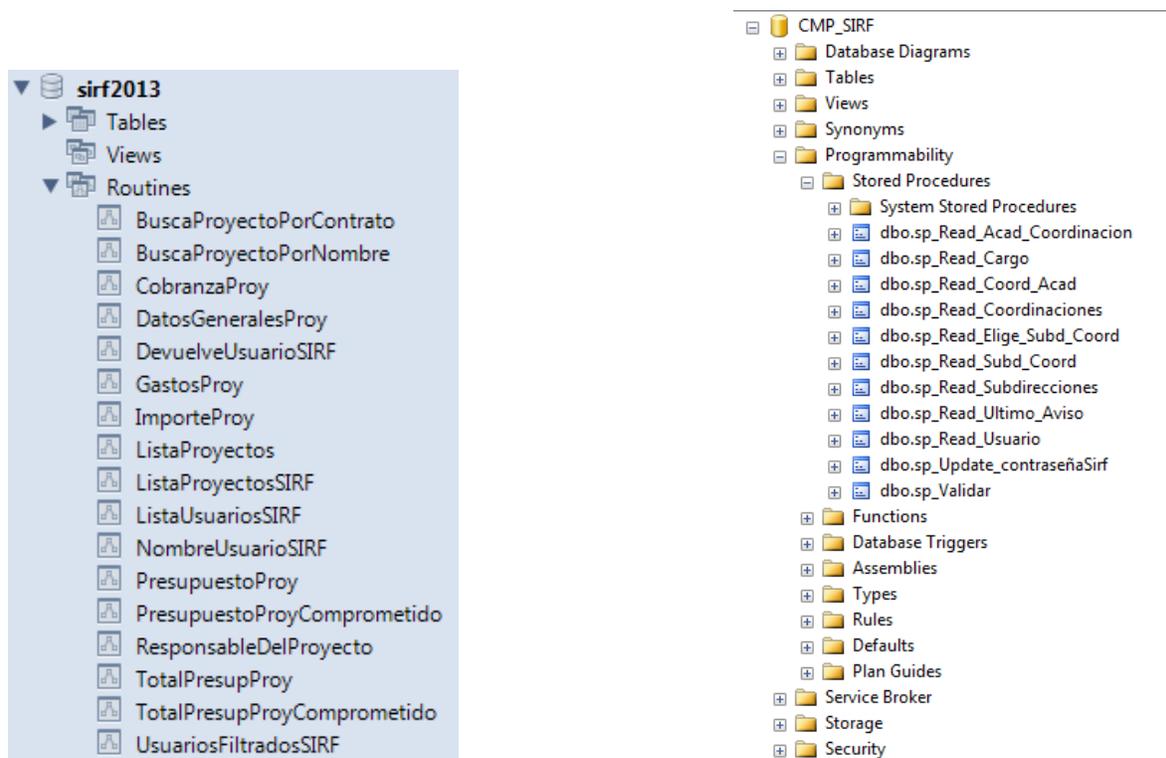


Figura 6.1 Procedimientos almacenados en ambas bases de datos

El siguiente es uno de los procedimientos almacenados que se programaron en la Base de Datos del SIRF en MySQL. Este procedimiento recibe como parámetro de entrada el ID del Usuario y regresa la lista de proyectos asociados a este usuario.

```

102
103 CREATE PROCEDURE ListaProyectos (IN UsuarioSIRF varchar(8))
104 BEGIN
105 SELECT
106 CASE `ic`.`tipo`
107     WHEN 'AN' THEN 'APOYO NAL.'
108     WHEN 'CN' THEN 'CONVENIO NAL.'
109     WHEN 'DO' THEN 'DONATIVO'
110     WHEN 'PN' THEN 'PROYECTO NAL.'
111 END AS `Tipo`,
112 `ic`.`Grupo`,
113 `ic`.`Contrato`,
114 `ic`.`Nombre` as 'NombreProy',
115 `ic`.`Coordinacion`,
116 `ic`.`Acontratacion` as 'Creacion'
117 FROM
118 `ingcontratos`.`ic`
119     INNER JOIN
120 `usuarios`.`u` ON substring(`ic`.`respinterno`, 1, 10) = substring(`u`.`rfc`, 1, 10)
121 where
122 `u`.`usuario` = UsuarioSIRF
123 ORDER BY `ic`.`Acontratacion` desc;
124 END

```

Figura 6.2 Procedimiento almacenado: ListaProyectos

6.1.1 XML

Un servicio Web XML es un componente de software que se comunica con otras aplicaciones mediante mensajes XML y envía dichos mensajes a través de protocolos estándares de internet como el Hypertext Transfer Protocol (HTTP).

Existen dos funciones fundamentales cuando se trabaja con los servicios web:

1. **Crear un servicio web:** al crear un servicio web XML, está creando una aplicación que expone su funcionalidad a los clientes de servicios web XML.
2. **Tener acceso a un servicio web (Consumir un servicio web):** al tener acceso a un servicio web, la aplicación cliente busca, hace referencia y utiliza la funcionalidad contenida dentro de un servicio web independiente. El cliente de un servicio web XML es normalmente una aplicación que puede enviar, recibir y procesar mensajes hacia el servicio web y desde él.

Se utilizó Visual Studio para la creación del Servicio Web XML, pues el entorno de desarrollo de ASP .NET nos permite crear este tipo de aplicaciones, lo que permite a estos servicios web tener acceso a numerosas características de .NET Framework, como autenticación, almacenamiento en caché y administración de estados.

Dos archivos principales son los que constituyen un servicio web XML: el archivo .asmx y el archivo asmx.cs. Un servicio Web XML no posee interfaz de usuario, por lo que el archivo .asmx únicamente contiene información del tipo de archivo y una directiva a la página de código

subyacente. El archivo `.asmx.cs`, también denominado página de código subyacente, contiene la lógica del servicio Web XML, la cual puede implementarse con el lenguaje de programación visual basic o C#. En este caso se eligió C#.

Para poder tener acceso a la base de datos del SIRF de MYSQL, es necesario obtener el conector de MySQL para .NET de la página web de MySQL para luego agregar la referencia `MySQL.Data.dll` al proyecto. Esto se hace dando clic derecho a la carpeta de “referencias” y seleccionando la opción “Agregar Referencia”.

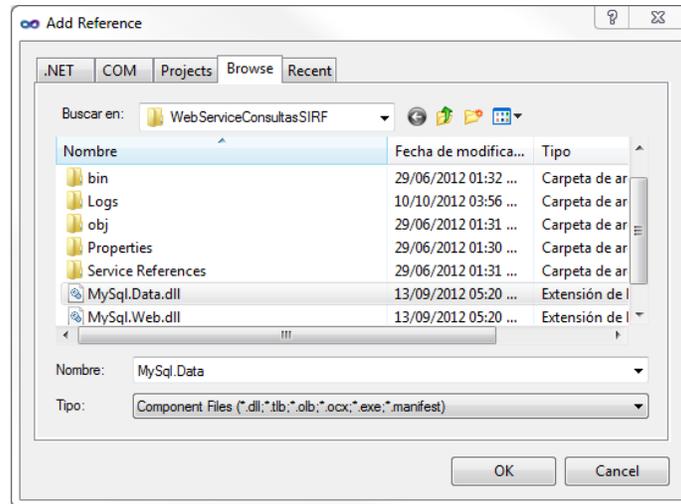


Figura 6.3 Agregar referencia `MySQL.Data.dll` al proyecto

Para conectar el webservice con la base de datos de SQLServer, se utilizará la herramienta ADO.NET incluida dentro de las librerías de .NET.

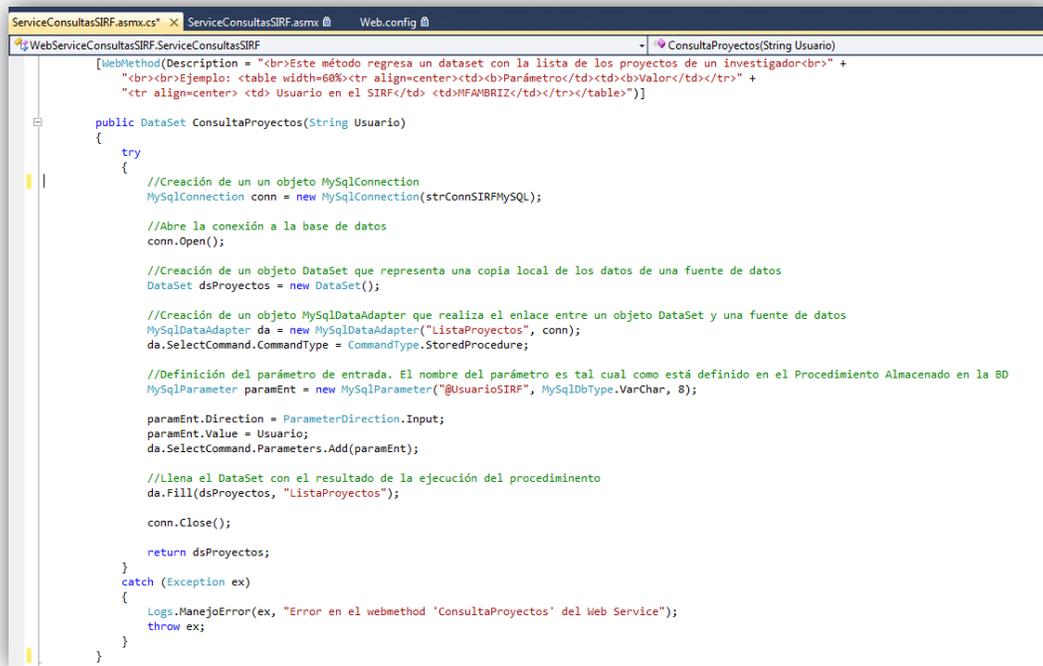
Hecho lo anterior se prosiguió con el desarrollo de los Web Methods del Web Service. Se crearon tres Web Methods por cada procedimiento almacenado que ejecuta:

El primero devuelve un objeto tipo `DataSet` el cual es consumido por la aplicación Web.

El segundo devuelve un objeto tipo `object[][]` el cual es consumido por la aplicación móvil para Windows Phone.

El tercero devuelve un objeto tipo `List<string>` el cual es consumido por la aplicación móvil para iOS.

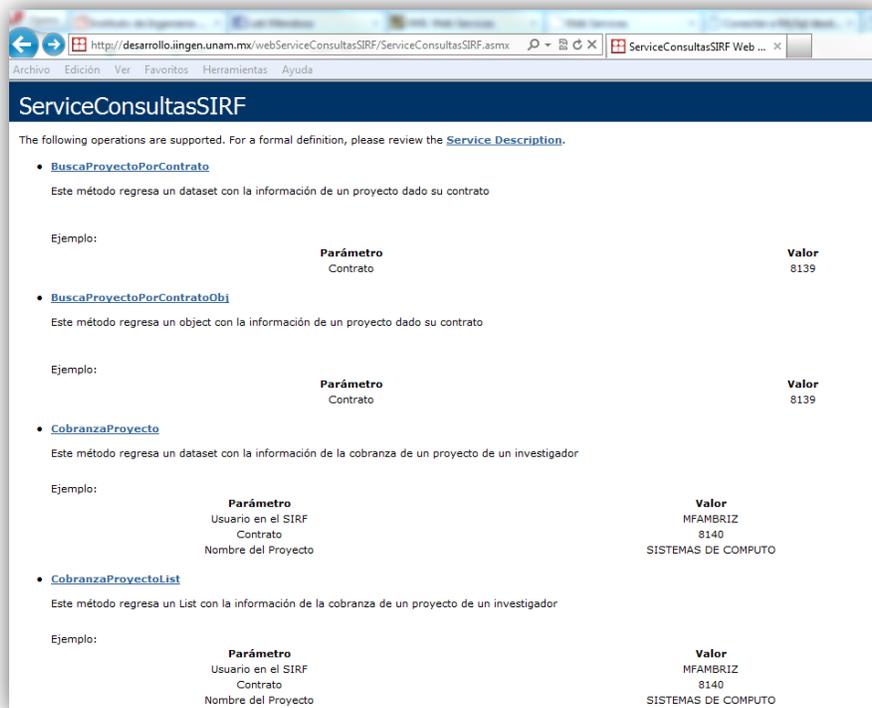
6.1 Programación de WebServices



```
ServiceConsultasSIRF.asmx.cs ServiceConsultasSIRF.asmx Web.config
WebServiceConsultasSIRF.ServiceConsultasSIRF ConsultarProyectos(String Usuario)
[WebMethod(Description = "<br>Este método regresa un dataset con la lista de los proyectos de un investigador<br>" +
"<br><br>Ejemplo: <table width=60%<tr align=center><td><b>Parámetro</td><td><b>Valor</td></tr>" +
"<tr align=center><td> Usuario en el SIRF</td> <td>MFAMBRIZ</td></tr></table>")]
public DataSet ConsultarProyectos(String Usuario)
{
    try
    {
        //Creación de un objeto MySqlConnection
        MySqlConnection conn = new MySqlConnection(strConnSIRFMySQL);
        //Abre la conexión a la base de datos
        conn.Open();
        //Creación de un objeto DataSet que representa una copia local de los datos de una fuente de datos
        DataSet dsProyectos = new DataSet();
        //Creación de un objeto MySqlDataAdapter que realiza el enlace entre un objeto DataSet y una fuente de datos
        MySqlDataAdapter da = new MySqlDataAdapter("ListaProyectos", conn);
        da.SelectCommand.CommandType = CommandType.StoredProcedure;
        //Definición del parámetro de entrada. El nombre del parámetro es tal cual como está definido en el Procedimiento Almacenado en la BD
        MySqlParameter paramEnt = new MySqlParameter("@UsuarioSIRF", SqlDbType.VarChar, 8);
        paramEnt.Direction = ParameterDirection.Input;
        paramEnt.Value = Usuario;
        da.SelectCommand.Parameters.Add(paramEnt);
        //Llena el DataSet con el resultado de la ejecución del procedimiento
        da.Fill(dsProyectos, "ListaProyectos");
        conn.Close();
        return dsProyectos;
    }
    catch (Exception ex)
    {
        Logs.ManejoError(ex, "Error en el webmethod 'ConsultarProyectos' del Web Service");
        throw ex;
    }
}
```

Figura 6.4 Bloque de código del WebService

La pantalla que se muestra al ejecutar el Web Service es la siguiente:



ServiceConsultasSIRF

The following operations are supported. For a formal definition, please review the [Service Description](#).

- BuscaProyectoPorContrato**
Este método regresa un dataset con la información de un proyecto dado su contrato
Ejemplo:

Parámetro	Valor
Contrato	8139
- BuscaProyectoPorContratoObj**
Este método regresa un object con la información de un proyecto dado su contrato
Ejemplo:

Parámetro	Valor
Contrato	8139
- CobranzaProyecto**
Este método regresa un dataset con la información de la cobranza de un proyecto de un investigador
Ejemplo:

Parámetro	Valor
Usuario en el SIRF	MFAMBRIZ
Contrato	8140
Nombre del Proyecto	SISTEMAS DE COMPUTO
- CobranzaProyectoList**
Este método regresa un List con la información de la cobranza de un proyecto de un investigador
Ejemplo:

Parámetro	Valor
Usuario en el SIRF	MFAMBRIZ
Contrato	8140
Nombre del Proyecto	SISTEMAS DE COMPUTO

Figura 6.5 Ejecución del Web Service en el Navegador Web

Al ejecutar algún método, se generará un archivo XML con la información obtenida de la ejecución de un procedimiento almacenado, como se muestra en la siguiente imagen:

```
<?xml version="1.0" encoding="UTF-8"?>
<DataSet xmlns="http://webservices.sirf.ingen.unam.mx">
  <xs:schema xmlns="" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:xs="http://www.w3.org/2001/XMLSchema" id="NewDataSet">
    <xs:element msdata:UseCurrentLocale="true" msdata:IsDataSet="true" name="NewDataSet">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded" minOccurs="0">
          <xs:element name="ListaProyectos">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Tipo" minOccurs="0" type="xs:string"/>
                <xs:element name="Grupo" minOccurs="0" type="xs:string"/>
                <xs:element name="Contrato" minOccurs="0" type="xs:string"/>
                <xs:element name="NombreProy" minOccurs="0" type="xs:string"/>
                <xs:element name="Coordinacion" minOccurs="0" type="xs:string"/>
                <xs:element name="Creacion" minOccurs="0" type="xs:int"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
  <diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
    <NewDataSet xmlns="">
      <ListaProyectos diffgr:id="ListaProyectos1" msdata:rowOrder="0">
        <Tipo>PROYECTO NAL.</Tipo>
        <Grupo>001</Grupo>
        <Contrato>OPERM1712013</Contrato>
        <NombreProy>OPERM1712013</NombreProy>
        <Coordinacion>5601</Coordinacion>
        <Creacion>2013</Creacion>
      </ListaProyectos>
      <ListaProyectos diffgr:id="ListaProyectos2" msdata:rowOrder="1">
        <Tipo>PROYECTO NAL.</Tipo>
        <Grupo>001</Grupo>
        <Contrato>AMBRIZ2012</Contrato>
        <NombreProy>ACTUALIZACION DEL HARDWARE DE LOS SERVIDORES MICRO</NombreProy>
        <Coordinacion>9309</Coordinacion>
        <Creacion>2012</Creacion>
      </ListaProyectos>
      <ListaProyectos diffgr:id="ListaProyectos3" msdata:rowOrder="2">
        <Tipo>PROYECTO NAL.</Tipo>
        <Grupo>004</Grupo>
        <Contrato>COMPUTO2012</Contrato>
        <NombreProy>ACTUALIZACION DEL HARDWARE DE LOS SERVIDORES</NombreProy>
        <Coordinacion>9318</Coordinacion>
        <Creacion>2012</Creacion>
      </ListaProyectos>
      <ListaProyectos diffgr:id="ListaProyectos4" msdata:rowOrder="3">
        <Tipo>PROYECTO NAL.</Tipo>
        <Grupo>001</Grupo>
        <Contrato>5601OPERACION</Contrato>
        <NombreProy>GASTOS DE OPERACION SISTEMAS DE COMPUTO</NombreProy>
        <Coordinacion>5601</Coordinacion>
        <Creacion>2011</Creacion>
      </ListaProyectos>
      <ListaProyectos diffgr:id="ListaProyectos5" msdata:rowOrder="4">
        <Tipo>PROYECTO NAL.</Tipo>

```

Figura 6.6 Archivo XML generado por el Web Service

6.1.2 JSON

Otra forma de enviar información es mediante el uso de JSON.

JSON (JavaScript Object Notation) es otro formato de intercambio de datos, fácil de interpretar y generar, está basado en el lenguaje de programación JavaScript y es más ligero que XML.

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales que son soportadas por cualquier lenguaje de programación.

Para el proyecto que concierne esta tesis, no se utilizó JSON. Se optó por crear webservices XML, ya que es el formato que el equipo de Ingeniería de Software ha utilizado para desarrollar webservices que sean consumidos por otras aplicaciones web .Net del Instituto de Ingeniería y lo más conveniente fue crear webservices que pudieran adaptarse a los sistemas actuales del Instituto de Ingeniería.

6.2 Programación de la aplicación Móvil

6.2.1 WindowsPhone

Windows Phone es el Sistema Operativo Móvil de Microsoft. Utiliza el lenguaje de diseño metro el cual se basa en la iconografía y tipografía de los diferentes sistemas de transporte alrededor del mundo, ya que estos son universales. Además, utiliza el color para deleitar al usuario, personalizar la experiencia y enfatizar jerarquías.

Para los desarrolladores proporciona una plataforma de desarrollo ágil y a la vez robusto para poder construir aplicaciones de una forma sencilla y rápida. Las herramientas necesarias para poder desarrollar aplicaciones de Windows Phone 7.5 son las siguientes:

- Visual Studio .NET 2010 (cualquier edición)
- SDK de WindowsPhone 7.1
- Actualización del SDK de WindowsPhone 7.11
- Expression Blend para Windows Phone

Éstas se obtienen desde el sitio web para desarrolladores de Windows Phone, en la siguiente página: <https://dev.windowsphone.com/en-us>

Los requisitos de software y hardware del SDK de Windows Phone 7.1 son los siguientes:

SDK de Windows Phone 7.1

Requisitos	
Hardware	Software
3 GB de RAM como mínimo 4 GB de espacio disponible Tarjeta de Vídeo con driver WDDM 1.1 capaz de ejecutar DirectX 10	Cualquier versión de Windows Vista (x86 ó x64) con SP2 excepto "Starter Edition" -ó- Cualquier versión de Windows 7 (x86 ó x64) excepto "Starter Edition"

Figura 6.7 Requisitos de SW y HW del SDK de Windows Phone 7.1

Windows Phone nos ofrece una manera de obtener aplicaciones bien estructuradas, separando el diseño de la lógica de la aplicación.

XAML (eXtensible Application Markup Language) es el lenguaje de marcado basado en XML, utilizado para diseñar la interfaz de usuario de las aplicaciones de Silverlight para Windows Phone 7.1. En XAML se utilizan las etiquetas para instanciar objetos y modelar datos.

La lógica del programa se desarrolla con el lenguaje C#. A cada elemento declarado en XAML le corresponde una clase en el API de Silverlight.

Elemento XAML	Clase
<Ellipse />	System.Windows.Shapes.Ellipse
<Rectangle />	System.Windows.Shapes.Rectangle
<Button />	System.Windows.Controls.Button
<ListBox />	System.Windows.Controls.ListBox
<Storyboard />	System.Windows.Media.Animation.Storyboard

Figura 6.8 Elementos XAML y sus clases correspondientes

Una vez teniendo las herramientas necesarias, se procedió con el desarrollo de la aplicación para Windows Phone. Se crearon los siguientes archivos, que conforman la aplicación:

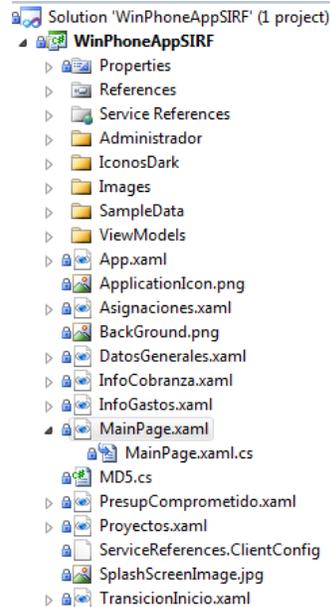


Figura 6.9 Aplicación de Windows Phone en Visual Studio

La siguiente imagen muestra cada una de las pantallas de las que consta la aplicación.



Figura 6.10 Pantallas de la aplicación de Windows Phone

6.2.2 Iphone

iOS es el sistema operativo diseñado por Apple que se encuentra instalado en los dispositivos de la misma empresa: iPod, iPhone y iPad. Para poder crear aplicaciones para estos dispositivos se requiere tener una computadora Mac que tenga la herramienta XCode con el SDK de iOS instalado, el cual se obtiene desde la Mac App Store.

Como se mencionó en el capítulo 4, se eligió la plataforma Titanium, creada por la empresa Appcelerator para desarrollar la aplicación de iOS con el lenguaje de programación JavaScript. Esta plataforma se obtiene del sitio de Appcelerator en la sección de descargas: <http://www.appcelerator.com/products/download>.

Una vez teniendo las herramientas necesarias se comenzó con el desarrollo de la aplicación, siguiendo la misma idea del diseño que las aplicaciones web móvil y móvil nativa para Windows Phone.

El proyecto en Titanium se muestra en la siguiente imagen:

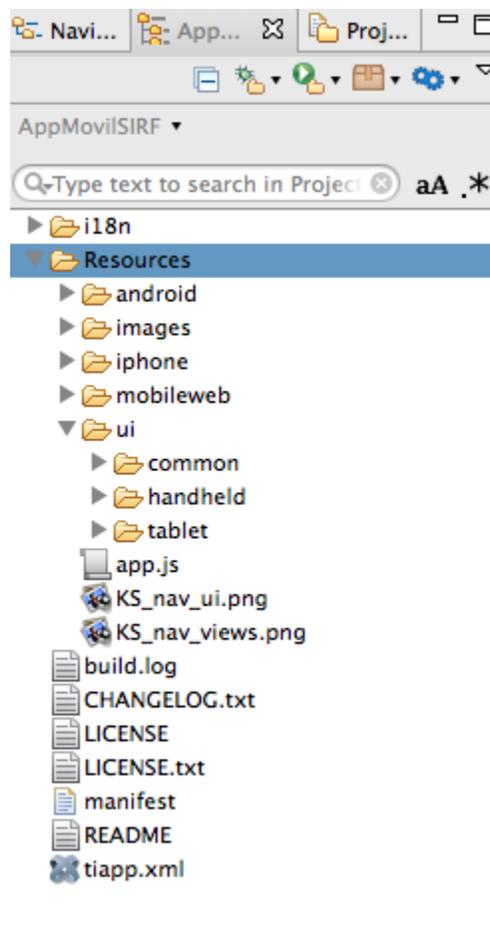


Figura 6.11 Aplicación de iOS en Appcelerator Titanium

6.2 Programación de la aplicación Móvil

La aplicación muestra una interfaz distinta cuando se trata de un iPad y de un iPhone, debido a los distintos tamaños de la pantalla.

Como podemos ver, el proyecto en Titanium contiene tres carpetas:

- **Handheld:** Aquí se encuentran las páginas que se mostrarán en un iPhone.
- **Tablet:** Aquí se encuentran las páginas que se mostrarán en un iPad.
- **Common:** Aquí se encuentran los archivos que contienen funciones que son utilizadas por ambas interfaces.

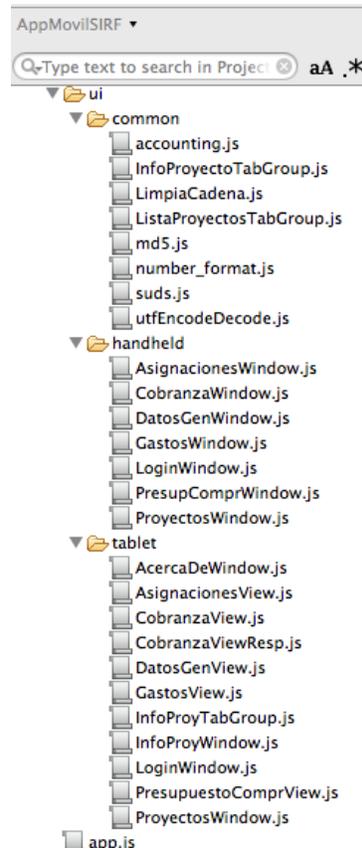


Figura 6.12 Archivos de la aplicación de iOS

Mediante el siguiente bloque de código ubicado en la página principal de la aplicación (app.js) es posible reconocer si el dispositivo es un iPad o un iPhone, direccionando a la carpeta correspondiente según sea el caso.

```
if (isTablet) {
    Win = require('ui/tablet/LoginWindow');
    new Win('Consulta de Proyectos').open();
}
else {
    Win = require('ui/handheld/LoginWindow');
    new Win('Consulta de Proyectos').open();
}
```

La funcionalidad para ambos dispositivos es la misma, pero el diseño de la interfaz es distinto, pues en el iPad se aprovechó el mayor espacio que ofrece, mostrando la información de una manera más amigable. Un ejemplo se observa en las siguientes imágenes que muestran la parte de la cobranza de un proyecto en la pantalla de iPhone y de iPad.

Recibo	Elaboración	Imp. Bruto	Imp. Neto	Ret. UNAM
8204290	2009	1,565,000.00	1,565,000.00	0.00
8293056	2009	1,565,000.00	1,565,000.00	0.00
8375239	2009	1,565,000.00	1,565,000.00	0.00
8375240	2009	1,565,000.00	1,565,000.00	0.00
8375273	2009	1,565,000.00	1,565,000.00	78,250.00

Figura 6.13 Parte de la Cobranza en pantalla de iPhone

Recibo	Imp. Neto
8204290	\$1,565,000.00
8293056	\$1,565,000.00

Figura 6.14 Parte de la Cobranza en pantalla de iPad

La siguiente imagen muestra cada una de las pantallas de iPhone de las que consta la aplicación.



Figura 6.15 Pantallas de la aplicación de iPhone

6.3 Programación de la aplicación Web

En Visual Studio se generó un proyecto solución para la aplicación Web de Consulta de Proyectos del SIRF, el cual se divide en cuatro sub proyectos, uno por cada capa de acuerdo a la arquitectura planteada:

- **DataAccess:** Contiene las clases que hacen una conexión directa con la base de datos de SQLServer y ejecutan los procedimientos almacenados en esta base de datos.
- **LogicadeNegocio:** Contiene la referencia al WebService y la clase que contiene los métodos que lo consumen.
- **Presentación:** Contiene cada una de las páginas .aspx y su correspondiente archivo de código subyacente aspx.cs.
- **Framework:** Contiene una clase para el manejo de errores en toda la aplicación.

La estructura del proyecto solución en Visual Studio se muestra en la siguiente figura:

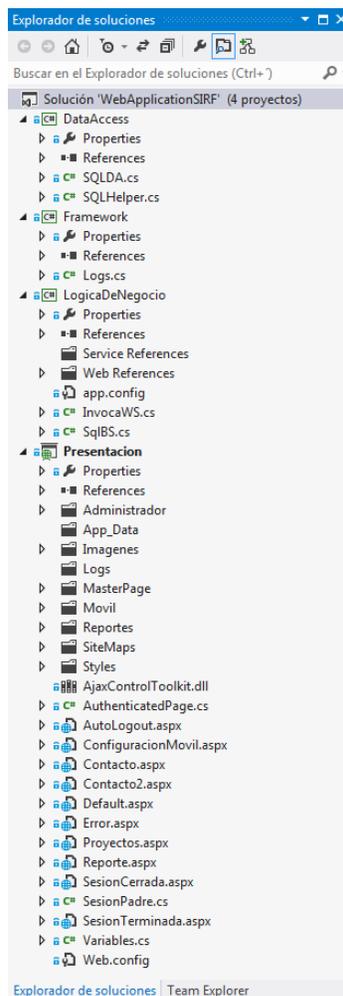


Figura 6.16 Aplicación Web en Visual Studio

6.3.1 ASPX

Las páginas aspx son las que se muestran al usuario, el cual utiliza su funcionamiento para poder obtener la información que requiere. El desarrollo de la lógica de cada página se realizó en su respectiva página de código subyacente. A continuación se explica el funcionamiento de algunas de ellas:

- **Página Principal (Defalut.aspx)**

Esta es la página principal de la aplicación. Realiza la autenticación e identificación del usuario. Si el usuario no se encuentra en la base de datos se asume que ese usuario no cuenta con permisos para ingresar al sistema. La página muestra un menú, en el cual de acuerdo al tipo de usuario identificado, se mostrará las opciones a las que puede tener acceso. Para el caso del director, se muestra la siguiente pantalla:



Figura 6.17 Página Principal de la App Web (Caso Director)

Para el caso de algún académico que es jefe de proyecto, se muestra la pantalla siguiente:



Figura 6.18 Página Principal de la App Web (Caso Jefe de Proyecto)

- **Mis Proyectos (Proyectos.aspx)**

Cuando el usuario selecciona la opción “Mis Proyectos” del menú principal, es direccionado a esta página, la cual muestra la lista de proyectos del usuario. Estos proyectos se muestran en un GridView que es llenado con la información que se obtiene de la base de datos al invocar el método “ListaProyectos” del webservice.

```
GVProys.DataSource = iws.ListaProyectos(idResponsable);
GVProys.DataBind();
```

Al seleccionar un proyecto, se almacenan los datos del proyecto seleccionado y es re direccionado a la página Reporte.aspx.

Tipo	Grupo	Núm. Contrato	Nombre	Coordinación	Creación	Acción
CONVENIO NAL.	010	MEMO BIOENERGÉTICO	MEMO BIOENERGÉTICOS	3283	2010	Consultar
CONVENIO NAL.	008	MUNICIPALESTEZIU	MUNICIPALESTEZIUTLAN	3228	2010	Consultar
CONVENIO NAL.	008	CHIAPAS 5 ESTUDI	CHIAPAS 5 ESTUDIOS	3286	2009	Consultar
PROYECTO NAL.	008	9002	BECAS POSDOCTORAL	1111	2009	Consultar
APOYO NAL.	002	CONACYT 83670	CONACYT 83670	1102	2008	Consultar

Figura 6.19 Lista de Proyectos

- **Información del Proyecto (Reporte.aspx)**

Esta página contiene toda la información del proyecto seleccionado, distribuida en las siguientes categorías:

Datos Generales

Contiene la información general del proyecto.

The screenshot shows the 'Consulta de proyectos SIRF' interface. At the top, there is a header with the UNAM logo and the text 'INSTITUTO DE INGENIERÍA UNAM'. The main title is 'Consulta de proyectos SIRF'. Below this, there is a navigation bar with 'Inicio / Mis Proyectos / Información del Proyecto' and a user profile 'Usuario: Noyola Robles Adalberto' with a 'Cerrar Sesión' link. The main content area displays project information: 'Responsable: NOYOLA ROBLES ADALBERTO' and 'Contrato: MEMO BIOENERGÉTICI Proyecto: MEMO BIOENERGÉTICOS'. A sidebar on the left lists navigation options: 'Datos Generales', 'Asignaciones', 'Cobranza', 'Trámites', and 'Reporte'. The main content is a table with the following data:

Patrocinador	IDRC010101010 CENTRO INTERNACIONAL DE INVESTIGACION PARA EL DESARROLLO (IDRC)	Copatrocinador	IDRC010101010 CENTRO INTERNACIONAL DE INVESTIGACION PARA EL DESARROLLO (IDRC)
Vigencia:	26/03/2010 - 26/03/2014	Situación:	VIGENTE
Coordinación: 3283			
Registro UNAM: 2580551525III10			
Estudio: Adaptación de las Ciudades de América Latina y el Caribe al cambio climático a través de un mejor uso de los recursos bio			
Observaciones: MEMORANDO DE CONDICIONES DE LA SUBVENCIÓN. La moneda a utilizar son dólares canadienses y el monto del convenio se inc			
Responsable 1	NORA560525SPA NOYOLA ROBLES ADALBERTO	Responsable 2	NORA560525SPA NOYOLA ROBLES ADALBERTO
		Cuenta I.E.:	282

Figura 6.20 Datos Generales del Proyecto

Asignaciones

Esta sección se compone de tres partes:

- **Importe del Proyecto:** Muestra el importe bruto y neto del proyecto
- **Presupuesto por rubros:** Muestra el presupuesto por rubros asignado al proyecto
- **Totales:** Muestra la suma del presupuesto asignado al proyecto

Importe del Proyecto

Importe con IVA M.N.	Importe con IVA M.E.	Importe sin IVA M.N.	Importe sin IVA M.E.	% I.V.A.
\$2,189,246.62	\$0.00	\$2,189,246.62	\$0.00	EX

Presupuesto por Rubros

Rubro	Autorizado	Ejercido	En Trámite	Saldo	Estatus
004 BECAS	248,650.00	0.00	0.00	248,650.00	VIGENTE
005 MATERIALES, LIBROS, HERRAMIENTAS	89,182.99	88,181.06	300.00	701.93	VIGENTE
006 EQUIPO Y MOBILIARIO	4,921.56	0.00	0.00	4,921.56	VIGENTE
007 VIÁTICOS Y PASAJES	310,517.80	306,939.34	1,400.02	106,396.45	VIGENTE
009 SERVICIOS EXTERNOS	1,000,782.31	949,918.58	44,045.01	306,229.34	VIGENTE
010 EMPRESAS SUBCONTRATADAS	19,593.03	0.00	0.00	19,593.03	VIGENTE
TOTALES	1,673,647.69	1,345,038.98	45,745.03	686,492.31	

Total Autorizado	\$1,673,647.69
Total Ejercido	\$1,345,038.98
Total en Trámite	\$45,745.03
Saldo por Ejercer	\$686,492.31

Figura 6.21 Asignaciones del Proyecto

Cobranza

Contiene la información de la cobranza del proyecto.

Recibo	Remesa	Elaboración	Importe Bruto	Importe Neto	Situación	Consultar
8128120	110625	2010	1,009,147.52	1,009,147.52	FINALIZADO	Detalle
8453289	112121	2010	1,098,157.28	1,098,157.28	FINALIZADO	Detalle

Figura 6.22 Cobranza del Proyecto

6.3 Programación de la aplicación Web

Cuando se seleccione ver el detalle de algún cobro, se mostrará un pop-up como se muestra en la siguiente imagen:

Detalle del cobro	
Recibo	8128120
Remesa	110625
Elaboración	2010
Importe Bruto	\$1,009,147.52
Importe Neto	\$1,009,147.52
Retención UNAM	\$0.00
Participación Dep.	\$0.00
Descuento	0%
Moneda	01
Situación	FINALIZADO
Fecha Cobrado	
Fecha Depositado	18/05/2011
Fecha Ingresado T	24/05/2011
Cuenta I.E.	282

Figura 6.23 Popup con el detalle del cobro

Trámites

En este apartado se muestra la información de los trámites del proyecto

 INSTITUTO DE INGENIERÍA UNAM		Consulta de proyectos SIRF						
Contacto Inicio / Mis Proyectos / Información del Proyecto			Usuario: Noyola Robles Adalberto Cerrar Sesión					
Responsable: NOYOLA ROBLES ADALBERTO Contrato: MEMO BIOENERGÉTICO Proyecto: MEMO BIOENERGÉTICOS								
Datos Generales	Clave	Referencia	Fecha	Situación	RFC Proveedor	Factura	Importe	Concepto
Asignaciones	009 SERVICIOS EXTERNOS	BA / 000006 - 3	13/02/2013 12:46:34 p.m.	PENDIENTE DE ENVIAR A FIRMA	VPR680820KU9	GFLTFO	13,507.00	
Cobranza	009 SERVICIOS EXTERNOS	BA / 000006 - 4	13/02/2013 12:46:34 p.m.	PENDIENTE DE ENVIAR A FIRMA	VPR680820KU9	GFLTFO [S]	232.00	
Trámites	009 SERVICIOS EXTERNOS	BA / 000006 - 1	13/02/2013 12:46:34 p.m.	PENDIENTE DE ENVIAR A FIRMA	VPR680820KU9	MGUKSA	13,507.00	
Reporte	009 SERVICIOS EXTERNOS	BA / 000006 - 2	13/02/2013 12:46:34 p.m.	PENDIENTE DE ENVIAR A FIRMA	VPR680820KU9	MGUKSA	232.00	
	005 MATERIALES, LIBROS, HERRAMIENTAS	RP / 000097 - 2	12/02/2013 05:57:17 p.m.	UPA	UNA2907232301	N103528	300.00	CASETAS, GASOLINA
	009 SERVICIOS EXTERNOS	RP / 000097 - 1	12/02/2013 05:57:17 p.m.	UPA	UNA2907232301	N0937	320.00	CASETAS, GASOLINA
	009 SERVICIOS EXTERNOS	BA / 000007 - 1	11/02/2013 07:43:14 p.m.	PENDIENTE DE ENVIAR A FIRMA	VPR680820KU9	DSGFWM	15,645.00	
	009 SERVICIOS EXTERNOS	BA / 000007 - 2	11/02/2013 07:43:14 p.m.	PENDIENTE DE ENVIAR A FIRMA	VPR680820KU9	DSGFWM [S]	232.00	
	009 SERVICIOS EXTERNOS	RP / 000096 - 1	31/01/2013 09:44:58 p.m.	ACEPTADO POR UPA	UNA2907232301	N118763	5,568.00	CURSO DE EVALUACIÓN DE IMPACTOS EN EL CICLO DE VIDA
	007 VIÁTICOS Y PASAJES	GS / 000006 - 1	31/01/2013 09:28:38 p.m.	ACEPTADO POR UPA	NORAS60525SPA		1,040.00	

Figura 6.24 Trámites del Proyecto

Reporte

En esta parte se creó un informe con Crystal Reports. Este informe contiene la información correspondiente a las asignaciones del proyecto.

Este reporte puede exportarse a un archivo PDF o XLS para su impresión o edición.

INSTITUTO DE INGENIERÍA UNAM **Consulta de proyectos SIRF**

Usuario: Noyola Robles Adalberto

Inicio / Mis Proyectos / Información del Proyecto

Responsable: NOYOLA ROBLES ADALBERTO

Contrato: MEMO BIOENERGÉTICO Proyecto: MEMO BIOENERGÉTICOS

Datos Generales
Asignaciones
Cobranza
Trámites
Reporte

14/03/2013

Responsable : NOYOLA ROBLES ADALBERTO

Contrato : MEMO BIOENERGÉTICO Proyecto : MEMO BIOENERGÉTICOS

Importe del Proyecto

Importe del Proyecto M.N.	Importe del Proyecto M.E.	Importe Neto M.N.	Importe Neto M.E.	%Iva
2189246.62	0.00	2189246.62	0.00	EX

Presupuesto por Rubros

Rubro	Autorizado	Ejercido	En Trámite	Saldo	Estatus
004 BECAS	248650.00	0.00	0.00	248650.00	VIGENTE
005 MATERIALES, LIBROS, HERRAMIENTAS	89182.99	88181.06	300.00	701.93	VIGENTE
006 EQUIPO Y MOBILIARIO	4921.56	0.00	0.00	4921.56	VIGENTE
007 VIÁTICOS Y PASAJES	310517.80	306939.34	1400.02	106396.45	VIGENTE
009 SERVICIOS EXTERNOS	1000782.31	949918.58	44045.01	306229.34	VIGENTE
010 EMPRESAS SUBCONTRATADAS	19593.03	0.00	0.00	19593.03	VIGENTE
TOTALES	1673647.69	1345038.98	45745.03	688492.31	

Totales	
Total Autorizado	1673647.69
Total Ejercido	1345038.98
Total en Trámite	45745.03
Saldo por Ejercer	688492.31

Figura 6.25 Reporte del Proyecto

- **App Móvil (ConfiguracionMovil.aspx)**

En esta página se configura una contraseña para el usuario logueado. Esta contraseña servirá para ingresar a las aplicaciones móviles, tanto las nativas como la App Web en su versión móvil.

La contraseña debe cumplir con las siguientes características:

- Debe tener al menos 8 caracteres y un máximo de 16.
- Debe contener al menos una letra mayúscula, una minúscula y un número.

La página verifica que la contraseña no sea una cadena vacía.

```
if (TbPwd1.Text == "")
{
    LbMsj.Text = "Ingrese una contraseña válida";
    LbMsj.Visible = true;
}
```

Luego el sistema recibe la contraseña dos veces para verificar que el usuario no se haya equivocado al ingresarla.

```
else if(TbPwd2.Text != TbPwd1.Text)
{
    LbMsj.Text = "Las contraseñas no coinciden. Vuelva a intentarlo";
    LbMsj.Visible = true;
}
```

Después comprueba que el tamaño de la contraseña sea mayor a 8 caracteres y menor a 16.

```
else if (TbPwd1.Text.Length < 8)
{
    LbMsj.Text = "La contraseña debe de tener por lo menos 8 caracteres";
    LbMsj.Visible = true;
}

else if (TbPwd1.Text.Length > 16)
{
    LbMsj.Text = "La contraseña debe tener menos de 16 caracteres";
    LbMsj.Visible = true;
}
```

Posteriormente se utiliza una expresión regular que verifica que la contraseña cumpla con las características especificadas:

```
Regex patron = new Regex("(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])");
```

Si la contraseña cumple con las características, es encriptada mediante un algoritmo y luego almacenada en la base de datos.

INSTITUTO DE INGENIERÍA UNAM

Consulta de proyectos SIRF

Usuario: Noyola Robles Adalberto

En este apartado usted puede crear una contraseña o cambiar una ya existente con la cual podrá acceder a la versión móvil del Módulo de Consulta de Proyectos del SIRF.

Configuración de la contraseña

Ingrese una nueva contraseña:

Vuelva a ingresar la misma contraseña:

Guardar

Nota:
La contraseña debe tener al menos 8 caracteres.
Debe contener al menos una letra mayúscula, una minúscula y un número.

Figura 6.26 Configuración Móvil

- **Búsqueda por Coordinación (Coordinaciones.aspx)**

Esta página muestra la lista de coordinaciones a las que puede tener acceso el usuario identificado.

De acuerdo al tipo de usuario identificado, tendrá los siguientes permisos:

- El director podrá ver todas las coordinaciones y las respectivas personas que las conforman.
- Un subdirector podrá ver las coordinaciones que pertenecen a su subdirección y las respectivas personas que las conforman.
- Un coordinador podrá ver la lista de personas que pertenecen a su coordinación.

Al dar clic en alguna persona para consultar, será direccionado a la página `Proyectos.aspx` de la misma carpeta, en la cual se encuentra la lista de proyectos de esa persona, en caso de que los tenga. Luego al seleccionar algún proyecto será direccionado a la página `ReporteAdmin.aspx` en la cual podrá consultar la información referente a ese proyecto.



Figura 6.27 Búsqueda por Coordinación

- **Búsqueda por proyecto (ProyectosSIRF.aspx)**

Esta página permite realizar una búsqueda de algún proyecto mediante el número de su contrato.

En primera instancia, muestra la lista de todos los proyectos registrados en el SIRF. Dentro del textbox el usuario puede ingresar algún contrato o parte de éste y luego al dar clic en buscar, se desplegará el proyecto o la lista de proyectos que contienen esa cadena.

Al dar clic en algún proyecto, será direccionado a la página ReporteAdmin.aspx de la misma carpeta, en la cual podrá consultar la información referente a ese proyecto.

Consulta de proyectos SIRF

Contacto Usuario: Noyola Robles Adalberto
 Inicio / Búsqueda por Número de Proyecto Cerrar Sesión

Búsqueda por Número de Proyecto

Tipo	Grupo	Núm. Contrato	Nombre	Coordinación	Creación	Consultar
PROYECTO NAL.	004	0003	PROY. EJECT. CONSTITUCION DEL LAB.	8103	2010	
PROYECTO NAL.	005	0102	VALIDACION DE ALGORITMOS DE DISEÑO	5107	2010	
PROYECTO NAL.	005	0103	DESINTOXICACIÓN SOLAR DE AGUAS 2DA ETAPA	5108	2010	
PROYECTO NAL.	004	0105	GPO DE TECNOLOGÍA SUSTENTABLE (GTS)	5109	2010	
PROYECTO NAL.	008	0106	ESTUDIO DE VIABILIDAD DE LA APERTURA	5110	2010	
PROYECTO NAL.	005	0132	DESINTOXICACION SOLAR DE AGUAS	5111	2010	
PROYECTO NAL.	004	0136	INVESTIGACION Y NICHOS DE OPORTUNIDAD	5509	2010	
PROYECTO NAL.	005	0304	EFFECTOS ROTURA DEL OLEAJE	3118	2010	
PROYECTO NAL.	008	0306	PROGRAMA INTEGRAL PARA APROXIMAR A LA SUSTENTABIL	3119	2010	
PROYECTO NAL.	006	0314	CARACTERIZACION SUELO Y SUBSUELO	3424	2010	
PROYECTO NAL.	004	0333	MANTENIMIENTO DE INSTAL. LAB. BIOPROCESOS	3130	2003	
PROYECTO NAL.	004	0351	PLANTA TRATAMIENTO EDIF. 12 Y 18	3132	2010	

[1 2 3 4 5 6 7 8 9 10 ...](#)

Figura 6.28 Búsqueda por Número de Proyecto

- **Búsqueda por Jefe de Proyecto (UsuariosSIRF.aspx)**

Esta página la cual permite realizar una búsqueda de algún investigador mediante el nombre del mismo.

En primera instancia, muestra la lista de todos los investigadores en el SIRF. Dentro del textbox el usuario puede ingresar el nombre de algún investigador. Al dar clic en el botón buscar, se desplegará el nombre o la lista de nombres que contienen la cadena ingresada.

Al dar clic en alguna persona para consultar, será direccionado a la página Proyectos.aspx de la misma carpeta, en la cual se encuentra la lista de proyectos de esa persona, en caso de que los tenga. Luego al seleccionar algún proyecto será direccionado a la página ReporteAdmin.aspx de la misma carpeta, en la cual podrá consultar la información referente a ese proyecto.

Consulta de proyectos SIRF

Contacto Usuario: Noyola Robles Adalberto
[Inicio](#) / Búsqueda por Jefe de Proyecto Cerrar Sesión

Búsqueda por Jefe de Proyecto

ingrese el nombre a buscar

Nombre del Investigador	Usuario	Consultar
AGUIRRE GONZALEZ JORGE	JAGUIRRE	
AIRE UNTIVEROS CARLOS MAXIMO	CMAIREUN	
ALCANTARA NOLASCO LEONARDO	LALCANTA	
ALCOCER MARTINEZ DE CASTRO SERGIO MANUEL	SMALCOCE	
ALMANZA SALGADO RAFAEL	RALMANZA	
ALMORA MATA DAVID	DALMORAM	
ALVAREZ ICAZA LONGORIA LUIS AGUSTIN	LAALVARE	
AMBRIZ MAGUEY MARCO FLORENTINO	MFAMBRIZ	
ANGELES CAMACHO CESAR	CANGELES	
ANTUN CALLABA JUAN PABLO	JPANTUNC	
AUVINET GUICHARD GABRIEL YVES ARMAND	GYAUVINE	
AUVINET GUICHARD GABRIEL YVES ARMAND	inves	

1 2 3 4 5 6 7 8 9 10 ...

Figura 6.29 Búsqueda por Jefe de Proyecto

Para el caso de la parte móvil de la aplicación Web, se tienen las siguientes pantallas:

- **Pantalla Principal (Default.aspx)**

Esta es la pantalla principal de la aplicación web en su versión móvil. Aquí es donde el usuario ingresa su usuario y contraseña para ser autenticado y autorizado.

Recordemos que la contraseña para las aplicaciones móviles es la que se configura dentro de la aplicación web. Ésta contraseña está almacenada de forma cifrada mediante cierto algoritmo.

Cuando un usuario ingrese a la aplicación móvil, ya sea nativa o web, la aplicación debe cifrar la contraseña y compararla con la que está almacenada en la base de datos. Este es el proceso de autenticación que debe realizar la aplicación móvil.



Figura 6.30 Página principal de la App Web Móvil

- **Proyectos (ProyectosMovil.aspx)**

Una vez que el usuario haya sido autenticado, será direccionado a la página que contiene la lista de sus proyectos que tiene a cargo. Aquí, el usuario podrá seleccionar algún proyecto para consultar su información respectiva.

Proyectos				
MURIA VILA DAVID				
Tipo	Núm. Contrato	Nombre	Creación	Acción
PROYECTO NAL.	OPERM1312013	OPERM1312013	2013	Consultar
CONVENIO NAL.	AUTURBSUR2012	AUTOPISTA URBANA SUR PRUEBAS EXPERIMENTALES	2012	Consultar
CONVENIO NAL.	CONV0402012	CONV0402012	2012	Consultar
CONVENIO NAL.	AUSUR2012	AUTOPISTA URBANA SUR ASESORIA TECNICA ASPECTOS EST	2012	Consultar
PROYECTO NAL.	R608R403	R608R403	2011	Consultar

Figura 6.31 Página de Proyectos de la App Web Móvil

- **Menú Principal (MenuReporte.aspx)**

Cuando un usuario selecciona un proyecto, el sistema lo direcciona a la página del menú principal. Aquí se encuentran las opciones de consulta que son: Datos Generales, Asignaciones, Cobranza y Gastos.



Figura 6.32 Página del menú principal de la App Web Móvil

- **Datos Generales (DatosGenerales.aspx)**

Esta página muestra los datos generales del proyecto seleccionado.



Figura 6.33 Página de Datos Generales del Proyecto de la App Web Móvil

- **Asignaciones (Asignaciones.aspx)**

Esta página muestra las asignaciones del proyecto seleccionado.

The screenshot shows a mobile application interface with a red header bar containing 'Atrás' and 'Asignaciones'. Below the header, it displays the project name and responsible person. A summary table shows the project's total budget and net amount. A detailed table lists budget items with columns for 'Rubro', 'Autorizado', 'Ejercido', 'En trámite', 'Saldo', and 'Estatus'.

Asignaciones					
Responsable: MURIA VILA DAVID					
Proyecto: AUTOPISTA URBANA SUR ASESORIA TECNICA ASPECTOS EST					
Importe del Proyecto					
Importe M.N. Importe Neto M.N. % I.V.A.					
\$4,176,000.00 \$3,600,000.00 16					
Presupuesto Por Rubros					
Rubro	Autorizado	Ejercido	En trámite	Saldo	Estatus
004 BECAS	550,000.00	0.00	0.00	550,000.00	VIGENTE
005 MATERIALES, LIBROS, HERRAMIENTAS	135,800.00	0.00	0.00	135,800.00	VIGENTE
006 EQUIPO Y MOBILIARIO	130,000.00	11,092.50	0.00	118,907.50	VIGENTE
007 VIÁTICOS Y PASAJES	400,000.00	28,243.92	0.00	371,756.08	VIGENTE

Figura 6.34 Página de Asignaciones del Proyecto de la App Web Móvil

6.4 Bibliografía

[1] [http://msdn.microsoft.com/es-mx/library/yzbxwf53\(v=vs.90\).aspx](http://msdn.microsoft.com/es-mx/library/yzbxwf53(v=vs.90).aspx)

[3] <http://sg.com.mx/content/view/1089>

Capítulo 7

Pruebas, depuración y liberación de la aplicación

7.1 Documentación de las pruebas

Como se explicó en el punto 3.5, para garantizar la calidad de un software, es necesario realizar pruebas a lo largo de su desarrollo y al término de éste. Recordemos que el objetivo de las pruebas es encontrar errores no conocidos para corregirlos y garantizar que el sistema satisfaga los requisitos especificados.

Se realizaron varios tipos de pruebas de las aplicaciones del módulo de consulta de proyectos del SIRF, para lo cual se montó un escenario con las condiciones reales en cuanto a infraestructura, de tal manera que representara una similitud a lo que se presentará cuando se liberen las aplicaciones.

Recursos	Nombre/Tipo
Servidor de base de datos	Servidor Windows Server 2008 con SQL server 2008 y MySQL Workbench instalados
Red	Red del Instituto de Ingeniería
Bases de Datos	Bases de datos de prueba (SQL server y MySQL)
Servidor Web	Windows server 2008 con IIS 7.0
Computadoras personales y configuración.	Computadoras personales con distintos sistemas operativos (Windows, Linux, Mac OS) con sus correspondientes navegadores de internet.
Dispositivos móviles	Dispositivos móviles con sistema operativo iOS y Windows Phone 7

Tabla 7.1 Infraestructura del ambiente de pruebas

A continuación, para cada tipo de prueba se muestra una matriz que detalla algunos ejemplos de pruebas realizadas.

Pruebas de Contenido

#	Nombre	Descripción	Condiciones de Ejecución	Entradas	Resultado obtenido	Resultado esperado
1	Validación del contenido textual estático de cada página	Se valida que el texto de cada página se muestre correctamente sin errores sintácticos.	Existen enlaces a cada página	Sitio disponible Usuario autenticado	El contenido se muestra correctamente	El contenido se muestra correctamente

7.1 Documentación de las pruebas

2	Validación del contenido textual dinámico de cada página	Se valida que el texto de cada página se muestre correctamente sin errores sintácticos.	Existen enlaces a cada página	Sitio disponible Usuario autenticado	Se muestra el código html de algunos caracteres como la letra "ñ" en lugar del mismo carácter	El contenido se muestra correctamente
---	--	---	-------------------------------	---	---	---------------------------------------

Tabla 7.2 Matriz de pruebas de contenido

Pruebas de Interfaz

Al interactuar con la aplicación, navegando a lo largo de todas las páginas se fueron encontrando detalles que podían mejorarse en la interfaz para lograr que los usuarios interactuaran de una forma simple con la aplicación.

#	Nombre	Descripción	Condiciones de Ejecución	Entradas	Resultado obtenido	Resultado esperado
1	Entrada de datos	Se valida que se realice correctamente la entrada de los datos de autenticación	La página de autenticación está disponible y se muestra correctamente	Sitio disponible	Se realizó la validación de los datos ingresados	Se realiza la validación de los datos ingresados
2	Botones	Se valida que todos los botones y enlaces estén disponibles y realicen la acción esperada	Existen enlaces a cada página El usuario entiende a dónde lo lleva cada botón y enlace	Sitio disponible	Se realizó la validación de cada botón	Se realizó la validación de cada botón

Tabla 7.3 Matriz de pruebas de interfaz

Por ejemplo, la siguiente imagen muestra la página principal de la aplicación web. Después de varias pruebas se llegó al diseño mostrado en la imagen, la cual es una forma muy simple de llevar al usuario a las siguientes páginas y que éstos entiendan de qué se trata cada sección.



Figura 7.1 Prueba de Interfaz

Pruebas de Navegación

#	Nombre	Descripción	Condiciones de Ejecución	Entradas	Resultado obtenido	Resultado esperado
1	Navegabilidad	Se prueba la navegabilidad a cada una de las páginas que componen a la aplicación.	Existen enlaces a cada página	Sitio disponible Usuario autenticado	Se intenta navegar por cada una de las páginas de atrás hacia adelante y de adelante hacia atrás, pero no es posible ir a una página por la que se pasó para	Se navega por cada una de las páginas de la aplicación de atrás hacia adelante y de adelante hacia atrás.

7.1 Documentación de las pruebas

					<p>llegar a la actual.</p> <p>Se implementó un sitemap que le muestre al usuario en qué página se encuentra y por cuáles páginas ha pasado. Con esto, el usuario puede dirigirse a alguna de las páginas anteriores a la que se encuentra haciendo clic en el nombre de la página a la que quiere navegar.</p>	
--	--	--	--	--	--	--

Tabla 7.4 Matriz de pruebas de navegación

Pruebas de Componentes

#	Nombre	Descripción	Condiciones de Ejecución	Entradas	Resultado obtenido	Resultado esperado
1	Conexión a la BD	Se valida que la aplicación tenga conexión a la BD	Que exista comunicación con el servidor de BD	Servidor, Base de datos, usuario, contraseña	Acceso a la BD	Acceso a la BD
2	Web Service	Se prueban los métodos del web service con sus parámetros de entrada	Que exista comunicación con el servidor donde se encuentra el Web Service	Datos de entrada de cada uno de los métodos del Web Service	La información que devuelve el Web Service es la que se esperaba.	La información que devuelve el Web Service es la que se esperaba.
3	Cifrado	Se prueba la función que realiza el cifrado de la contraseña ingresada la cual es comparada con la que se encuentra	La página de autenticación está disponible y se muestra correctamente	Nombre del usuario y contraseña sin cifrar	La función regresa true si la contraseña que ingresó, luego de ser cifrada corresponde con la que se encuentra	La función regresa true si la contraseña que ingresó, luego de ser cifrada corresponde con la que se encuentra

		almacenada en la BD para autenticar al usuario en las aplicaciones móviles.			almacenada en la BD.	almacenada en la BD.
--	--	---	--	--	----------------------	----------------------

Tabla 7.5 Matriz de pruebas de componentes

Pruebas de Configuración

Se realizaron pruebas en varias computadoras y dispositivos móviles de características distintas, analizando algunas variables como son el navegador web, la versión del navegador web, el tipo de conectividad y la resolución de la pantalla.

#	Nombre	Descripción	Condiciones de Ejecución	Entradas	Resultado obtenido	Resultado esperado
1	Navegador Internet Explorer	Se prueba la funcionalidad completa de la aplicación web	La aplicación está disponible	Datos de autenticación	Algunos bordes de las paginas no se visualizan	Se visualizan correctamente las paginas
2	Navegador Mozilla Firefox	Se prueba la funcionalidad completa de la aplicación web	La aplicación está disponible	Datos de autenticación	Se visualizan correctamente las paginas	Se visualizan correctamente las paginas
3	Navegador Google Chrome	Se prueba la funcionalidad completa de la aplicación web	La aplicación está disponible	Datos de autenticación	Algunas imágenes no se visualizan	Se visualizan correctamente las paginas
5	Dispositivo móvil con Sistema Operativo Windows Phone (Nokia Lumia 610) y conectividad Wi-fi	Se prueba la funcionalidad completa de la aplicación móvil	La aplicación se encuentra instalada en el dispositivo	Datos de autenticación	Se visualiza correctamente cada una de las ventanas que conforman a la aplicación	Se visualiza correctamente cada una de las ventanas que conforman a la aplicación
6	Dipositivo móvil con Sistema	Se prueba la funcionalidad completa de	La aplicación se	Datos de autenticación	Se visualiza correctamente cada una de	Se visualiza correctamente cada una de

7.1 Documentación de las pruebas

	Operativo iOS (Iphone 4) y conectividad Wi-fi	la aplicación móvil	encuentra instalada en el dispositivo		las ventanas que conforman a la aplicación	las ventanas que conforman a la aplicación
--	---	---------------------	---------------------------------------	--	--	--

Tabla 7.6 Matriz de pruebas de configuración

Prueba de Seguridad

La seguridad de la aplicación web depende en gran parte de su configuración en el servidor IIS el cual implementa mecanismos que incluyen la autenticación de usuarios, con lo que solo los usuarios autorizados podrán ingresar a la aplicación. Actualmente, el área de Ingeniería de Software en conjunto con el área de servidores Windows del IINGEN está trabajando para incluir el uso de certificados con el objetivo de propiciar un canal de comunicación más seguro entre el cliente y el servidor.

#	Nombre	Descripción	Condiciones de Ejecución	Entradas	Resultado obtenido	Resultado esperado
1	Autenticación del usuario	Se prueba la autenticación del usuario registrado en el Active Directory	Que el usuario esté registrado en el active directory y tenga acceso a la aplicación	Usuario y contraseña	Usuario autenticado	Usuario autenticado

Tabla 7.7 Matriz de pruebas de seguridad

7.2 Compatibilidad de dispositivos móviles

Hacer una página web compatible con dispositivos móviles amplía y mejora la experiencia de los usuarios, ya que éstos utilizan cada vez más los dispositivos móviles para navegar en internet, consultar información, realizar búsquedas de productos o servicios, entre otras cosas.

Actualmente existe una gran variedad de dispositivos y sistemas operativos móviles. La elección de cada uno depende de las necesidades del usuario o de la empresa. Si queremos tener éxito con nuestras aplicaciones móviles, éstas deben abarcar el mayor número de estos dispositivos.

Existen diversas herramientas para probar aplicaciones web móviles mediante simuladores de dispositivos móviles, sin necesidad de tener los dispositivos físicamente. Una de ellas es la herramienta Ripple. Se trata de una extensión del navegador web Google Chrome que nos permite cargar una página web en el emulador de algún dispositivo móvil. Es muy útil para verificar que en determinado tamaño de pantalla la aplicación se verá bien. Se utilizó esta herramienta para probar la aplicación web móvil y verificar que los objetos y textos se visualizaran correctamente.

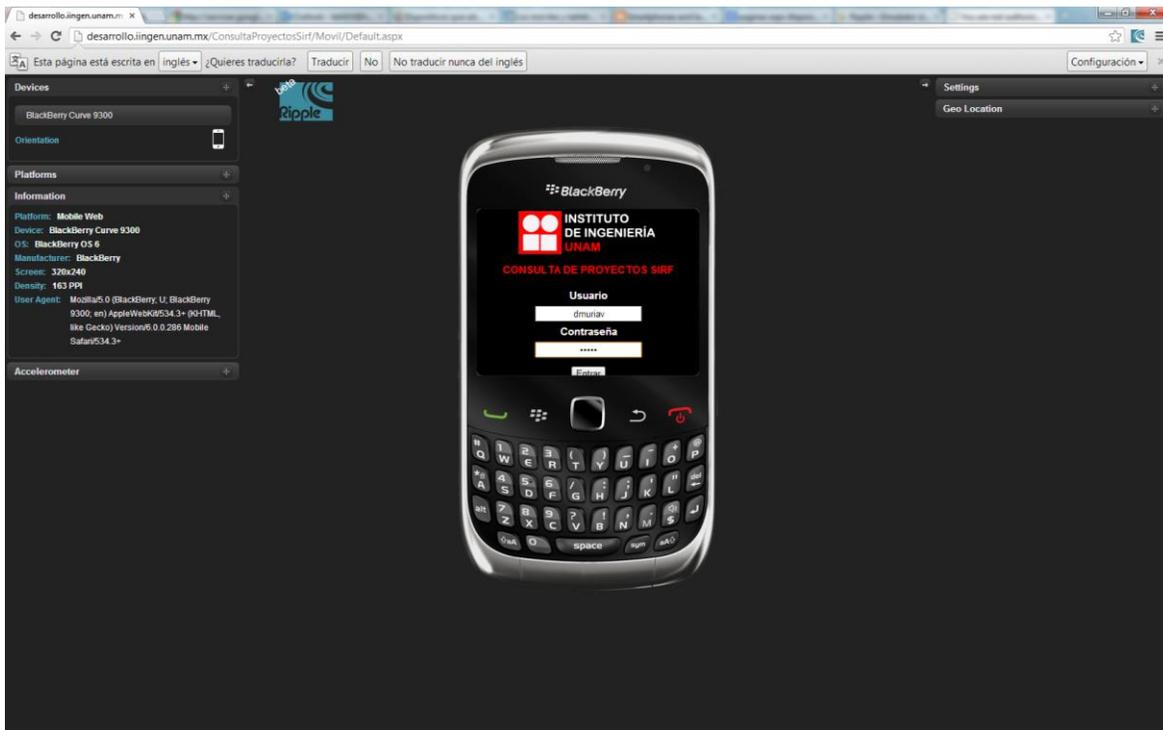


Figura 7.2 La herramienta Ripple

7.3 Compatibilidad de navegadores web

Una página web es compatible con determinado navegador web si se muestra como se definió durante su diseño, y su funcionamiento es correcto. En la actualidad existen diversos navegadores web que tienen algunas variaciones en su interpretación de código HTML, ASPX y CSS. Esto hace que algunas aplicaciones web no funcionen o no se muestren correctamente en algunos navegadores.

7.3 Compatibilidad de navegadores web

La aplicación web se probó en los navegadores web más utilizados en la actualidad, ya que los usuarios utilizan éstos para navegar en internet ya sea desde su computadora o desde su dispositivo móvil.

La siguiente tabla muestra un resumen de compatibilidad de navegadores web de la aplicación.

Navegador	Grado de compatibilidad
Internet Explorer 10.0	90 % (No muestra correctamente los bordes de las páginas)
Mozilla Firefox 11.0	100 %
Google Chrome 26.0	95 % (No muestra las imágenes en el reporte de Crystal Reports)
Opera 12.15	80 % (No muestra el reporte de Crystal Reports)
Safari 6.0.2	90 % (No respeta algunos tipos de letra)

Tabla 7.8 Compatibilidad de Navegadores Web

7.4 Documentos de depuración

La depuración consiste en encontrar la raíz de un error para comprenderlo y corregirlo. Es el último recurso de aseguramiento de la calidad, pues desde que se inicia el desarrollo se debe ser muy cuidadoso y seguir unas buenas prácticas de programación.

Los errores de programación se clasifican en tres categorías: errores de compilación, errores en tiempo de ejecución y errores lógicos.

Errores de compilación

Son errores que impiden que el programa se ejecute. Si el compilador de Visual Studio detecta código que no reconoce emite un error de compilación. La mayoría de los errores de compilación se deben a errores cometidos al escribir el código.

Errores en tiempo de ejecución

Son errores que aparecen mientras se ejecuta el programa. Aparecen normalmente cuando el programa intenta una operación que es imposible que se lleve a cabo. Un ejemplo de esto es la división por cero.

Errores lógicos

Este tipo de errores impiden que el programa haga lo que estaba previsto. Su código puede compilarse y ejecutarse sin errores, pero el resultado de una operación puede generar un resultado no esperado. Estos errores son los más difíciles de detectar y corregir. Sin embargo, los

IDE's de desarrollo como Visual Studio o Titanium Studio disponen de herramientas de depuración que nos facilitan el trabajo.

Se realizaron pruebas y depuraciones durante la programación, con ayuda de las herramientas de desarrollo Visual Studio y Titanium Studio que nos permiten depurar paso a paso las aplicaciones y encontrar si era el caso, la raíz del error. Al finalizar el desarrollo también se realizaron pruebas y depuraciones para asegurar la calidad del sistema.

En el código se identificaron los bloques donde era posible que se generara algún tipo de error conocido. Estos errores se controlaron con bloques de código dentro de los cuales se manda a escribir a un archivo de logs el error generado y en qué bloque de código se generó. Esto facilita la depuración en caso de que se genere algún error desde alguna computadora cliente. Con solo revisar el archivo de logs se podrá identificar el error y en dónde se generó, lo que facilita la depuración.

7.5 Liberación

Una vez que se realizaron las pruebas y depuraciones pertinentes, el sistema está listo para ser liberado. Liberar el sistema consiste en lo siguiente:

- La aplicación web y web móvil pasará al servidor de producción.
- Las aplicaciones móviles serán publicadas en la tienda de aplicaciones correspondiente.

El primero paso es la creación de la base de datos **CMP_SIRF** en el servidor de producción con SQL Server 2008. Recordemos que esta base de datos fue creada por el área de Sistemas de Información en un servidor de pruebas. De esta base de datos se obtuvo el script CreateSIRF.sql el cual contiene los comandos para la creación de las tablas, los procedimientos almacenados y la asignación de permisos del usuario. El script genera la cuenta de usuario, que es el que utiliza el web service para conectarse con esta base de datos.

Lo siguiente es la publicación del webservice en el servidor de producción para que pueda ser consumido por las aplicaciones web y móviles.

Para poner en marcha la aplicación web se requiere considerar lo siguiente:

Requerimientos técnicos

- Tener el IIS instalado en el servidor. Es el principal requisito pues gracias al IIS nuestro servidor actúa como un servidor web, el cual se encarga de proporcionar las herramientas y funciones necesarias para administrar aplicaciones web.
- La aplicación web emplea **".net Framework 4.0"**.

7.5 Liberación

- La aplicación web requiere que en el servidor se encuentren las librerías del Crystal Reports Release 2 XI.

Tipo de Acceso al sitio web

- Se requiere acceso restringido a este sitio web. Se requiere autenticación de Windows para entrar a cada zona del sitio Web, dependiendo el rol.

Tipo de Acceso al sitio web Móvil

- Se requiere acceso público

Publicación de la aplicación móvil para WindowsPhone

Windows Phone Dev Center es el sitio web para los desarrolladores de Windows Phone en el cual se encuentran las herramientas necesarias para construir, publicar y administrar aplicaciones para Windows Phone alrededor del mundo. Para publicar aplicaciones es necesario registrarse en este sitio y adquirir una membresía. Una vez adquirida la membresía, se envía la aplicación (archivo XAP) para ser revisada y posteriormente distribuida.

Una forma de distribución de aplicaciones Windows Phone es la distribución selectiva. Éste es el proceso de liberación de la aplicación a través del Windows Phone Dev Center mientras se mantiene oculto en la tienda de Windows Phone ([Windows Phone Store](#)). Puede enviar un enlace de su aplicación a un conjunto específico de usuarios a través de una página web. Esta es la forma en la que se distribuirá la aplicación de consulta de proyectos del SIRF una vez aceptada.

Publicación de la aplicación móvil para iOS

El proceso de publicación de aplicaciones en el APP Store se describe a continuación. El primer paso es registrarse como desarrollador de Apple y entrar al programa iOS Developer Program, debiendo pagar la cuota anual de desarrollador.

Una vez aceptada nuestra cuenta, debemos entrar al portal de iTunes Connect teniendo listo lo siguiente:

- Nombre de la App
- Descripción de la App
- Icono 512x512px

- Capturas de pantalla: al menos una para iPhone y otra para iPad en los siguientes formatos 320x460 (sin barra de estado); 320x480; 640x920 (retina, sin barra de estado); 640x960 (retina); para orientación paisaje 480x300 (sin barra de estado); 480x320; 960x600 (retina, sin barra de estado); o 960x640 (retina).

Dentro de iTunes Connect hay que dirigirnos al apartado Manage your applications, Add new app y rellenaremos los datos de la App. Una vez completados los datos hay que dar clic en Ready to Upload Binary. Los archivos binarios se suben desde el Xcode. Por último se envía la aplicación para su revisión, y una vez aceptada, ya estará disponible en el App Store.

7.6 Bibliografía

[1] Ingeniería del Software, Un enfoque práctico. Roger S. Pressman. Mc Graw Hill. Sexta Edición. México. 2006.

[2] <http://es.scribd.com/doc/27519905/8/Caracteristicas-de-IIS>
Abril 2012

[3] <http://msdn.microsoft.com/es-mx/library/s9ek7a19%28v=vs.80%29.aspx>
Abril, 2012

[4] <http://technet.microsoft.com/es-es/bb530393.aspx>
Abril, 2012

[5] <http://www.imaginanet.com/blog/como-subir-apps-a-apple-store.html>
Abril, 2012

[6] <http://sostic.farvista.net/?q=node/27>
Mayo, 2012

Capítulo 8

Conclusiones

A lo largo de este trabajo titulado “Aplicación Web y Móvil de consulta de proyectos del SIRF”, se desarrollaron cuatro aplicaciones: una aplicación web, una aplicación web móvil, una aplicación móvil para la plataforma Windows Phone y una aplicación móvil para la plataforma iOS. Cada una significó un reto, pues se utilizaron plataformas de desarrollo y lenguajes de programación distintos para su desarrollo.

Estas aplicaciones cumplen con los objetivos planteados al inicio de este trabajo, pues tienen la finalidad de acercar la información referente a sus proyectos a los investigadores del IINGEN para que pueda ser consultada de una manera fácil, práctica y amigable. Esta información puede consultarse desde cualquier lugar y en cualquier momento mediante alguna computadora que cuente con un navegador web haciendo uso de la aplicación web, o a mediante algún dispositivo móvil a través de las aplicaciones móviles, siempre y cuando se tenga acceso a internet.

Gracias a estas aplicaciones, no se necesita forzosamente un equipo de cómputo con determinadas características ni estar necesariamente conectados a la red interna del Instituto de Ingeniería como se venía haciendo con la aplicación de escritorio, pues la información ahora es consultada a través de internet mediante la aplicación web. Por otro lado, las aplicaciones móviles nativas sí dependen de algunas características de los dispositivos, como por ejemplo del sistema operativo; en este caso se abarcaron dos: iOS y Windows Phone. Sin embargo, para contrarrestar esto, se creó la aplicación web móvil la cual puede ser utilizada en dispositivos móviles con cualquier Sistema Operativo, a través de un navegador web móvil.

Los dispositivos móviles en la actualidad son una gran área de oportunidad en todos los ámbitos. En este caso, además de las aplicaciones desarrolladas para las plataformas móviles propuestas, se espera que se desarrollen a futuro para otras plataformas móviles, como por ejemplo Android.

Una vez cubiertos los objetivos, el presente trabajo queda abierto para todos aquellos ingenieros que deseen profundizar más en el tema, aportando nuevas ideas y trazando nuevas metas según lo permita el avance de la tecnología. También queda abierto para que pueda ser tomado como base para el desarrollo de nuevas funcionalidades en las aplicaciones.

Se espera también que este trabajo despierte el interés por desarrollar más aplicaciones móviles dentro del Instituto de Ingeniería para beneficio de su comunidad, pues existe una diversidad de áreas de investigación que pueden auxiliarse de este tipo de herramientas.

ANEXOS

Anexo 1

Estándar ISO 9000-3

ISO 9000-3 es la norma para la gestión de calidad de los productos de software y servicios relacionados, derivada del estándar ISO 9001. Su objetivo es proveer las especificaciones de cómo aplicar el estándar ISO 9001 para el desarrollo, aplicación y mantenimiento del software.

Las ideas básicas de este estándar son:

- El control de calidad debe ser aplicado a todas las fases de la producción de software, incluido el mantenimiento y tareas posteriores a su implantación.
- Debe existir una estricta colaboración entre la organización que adquiere el software y el proveedor del mismo.
- El proveedor del software debe definir su sistema de calidad y asegurar que toda la organización ponga en práctica este sistema.

Algunas cláusulas de esta norma que representa los requisitos y directrices de gestión de calidad del software se muestran en la siguiente tabla:

Administración de la Responsabilidad <ul style="list-style-type: none">•La dirección de la empresa debe definir y documentar su política y sus objetivos con respecto a la calidad.•Las responsabilidades, autoridades y relaciones entre todo personal, cuyo trabajo afecte la calidad del producto, deben ser definidas.
Sistema de Calidad <ul style="list-style-type: none">•La empresa debe establecer y mantener un sistema de calidad documentado.
Auditorías Internas de la Calidad <ul style="list-style-type: none">•La empresa debe llevar un sistema de auditorías internas de calidad y sus resultados deben ser documentados.
Revisión de Contratos <ul style="list-style-type: none">•La empresa debe establecer y mantener procedimientos para la revisión de los contratos y para la coordinación de estas actividades.

Control de Documentos y Datos

- La empresa debe establecer y mantener procedimientos para controlar todos los documentos y datos

Planificación del diseño y el desarrollo

- Esta cláusula exige la definición de un proceso disciplinado o metodología.

Inspección y Pruebas

- La empresa debe asegurar que los productos no se utilicen o procesen hasta que sean inspeccionados y verificados.

Control del producto no conforme

- Los productos que no cumplan los requerimientos especificados, no deben ser instalados, usados o puestos en producción.

Capacitación

- La empresa debe establecer y mantener procedimientos para identificar las necesidades de capacitación y proveer entrenamiento a todo el personal, que además debe ser calificado con base a su educación, entrenamiento o experiencia.

Referencia Bibliográfica:

<http://www.praxiom.com/iso-90003.htm>

Agosto 2012

<http://tinyurl.com/a9gyl6e>

Agosto 2012

Diagrama de Clases de la Aplicación Web de Consulta de Proyectos del SIRF

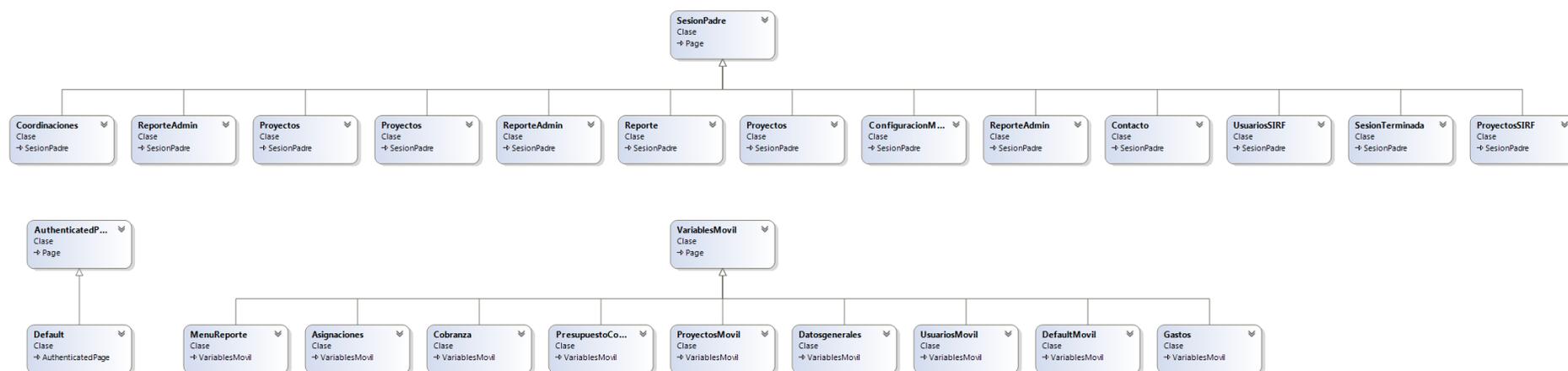


Diagrama de Clases de la Aplicación Móvil de Consulta de Proyectos del SIRF

GLOSARIO

Antivirus: Herramienta de software cuyo objetivo es detectar y eliminar virus informáticos.

Aplicación: Programa informático que permite a un usuario utilizar una computadora con un fin específico.

Base de Datos: Conjunto de información organizada con características en común.

Consola: También llamada terminal, es un programa al que se le ingresan órdenes y un intérprete de comandos analiza la secuencia de caracteres ingresada y, si la sintaxis de la orden es correcta, la ejecuta, recurriendo para ello a las funciones que ofrece el sistema operativo o el programa que representa.

Depurar: Proceso de identificación y eliminación de errores en un programa informático.

Framework: Es un marco de aplicación o conjunto de bibliotecas orientadas a la reutilización a muy gran escala de componentes software para el desarrollo rápido de aplicaciones.

Hash: Una función hash o función de resumen es un algoritmo que recibe un mensaje de entrada y regresa una cadena alfanumérica de longitud fija llamada código hash, que sirve como identificador único del mensaje de entrada. No se puede obtener el mensaje original a partir del código hash.

IDE: Es un entorno de programación que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

IIS (Internet Information Services): Son un conjunto de servicios para Microsoft Windows que admiten la creación, configuración y administración de sitios Web, además de otras funciones de Internet.

IIUNAM: Instituto de Ingeniería de la Universidad Nacional Autónoma de México

ISO International Organization for Standardization (Organización Internacional de Normalización): es una federación de alcance mundial integrada por cuerpos de estandarización nacionales de 130 países, encargada de promover el desarrollo de normas internacionales de fabricación (tanto de productos como de servicios), comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica.

Modelo Relacional: Es un modelo de datos donde los datos se representan en forma de tabla.

.NET: Es un framework de Microsoft que permite un rápido desarrollo de aplicaciones.

Plataforma de desarrollo de software: Es el conjunto de herramientas que nos permiten desarrollar aplicaciones. Consta de un IDE, una API, una documentación y diversas herramientas que permiten el desarrollo de aplicaciones.

PDA (Personal Digital Assistant): es una computadora de mano con la que se pueden realizar muchas de las funciones que hace una computadora de escritorio (ver películas, crear documentos, juegos, correo electrónico, navegar por Internet, reproducir archivos de audio, etc.) pero con la ventaja de ser portátil.

Plug-in: es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande. Un plug-in es un programa que incrementa o aumenta las funcionalidades de un programa principal. Por lo general es producido por una compañía diferente a la que produjo el primer programa.

Prototipo: Es una versión provisional del sistema, que solo tiene lo imprescindible para que el usuario final pueda comprobar si se han entendido bien sus requisitos.

RSS: Son las siglas de Really Simple Syndication, un formato XML para compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos. El formato permite distribuir contenidos sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS (agregador).

Servidor Web: Es un programa que está diseñado para transferir hipertextos, páginas web o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. El programa implementa el protocolo HTTP (HyperText Transfer Protocol) que pertenece a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.

SIAF: Sistema Integral de Administración Financiera

SIAUWEB: Sistema de Información para la Administración Universitaria

SIRF: Sistema Institucional de Registro Financiero

Sitemap: Los sitemaps son archivos que contienen un listado en formato XML de todas las páginas que componen un sitio web.

SmartPhone: Un teléfono inteligente es un teléfono móvil que soporta

SOAP (Protocolo Simple de Acceso a Objetos): es un protocolo de comunicación basado en XML que permite una comunicación de mensajes entre aplicaciones vía HTTP, normalmente utilizado en WebServices.

Tablet: es una computadora portátil de mayor tamaño que un teléfono inteligente o una PDA, integrado en una pantalla táctil (sencilla o multitáctil) con la que se interactúa primariamente con

los dedos o una pluma stylus (pasiva o activa), sin necesidad de teclado físico ni ratón. Estos últimos se ven reemplazados por un teclado virtual y, en determinados modelos, por una mini-trackball integrada en uno de los bordes de la pantalla.

TIC: Las Tecnologías de la Información y la Comunicación, también conocidas como TIC, son el conjunto de tecnologías desarrolladas para gestionar información, enviarla de un lugar a otro, almacenarla y procesarla.

Transact-SQL: Es un lenguaje universal que se emplea en cualquier sistema gestor de bases de datos relacional. Tiene un estándar definido, a partir del cual cada sistema gestor ha desarrollado su versión propia.

UML: Es el modelo estándar para la construcción de software orientado a objetos.

VPN: Virtual Private Network (Red Privada Virtual): Es una red privada construida dentro de una infraestructura de red pública, como por ejemplo Internet. Las empresas pueden usar una red VPN para conectar de manera segura oficinas y usuarios remotos por medio de un acceso a Internet económico suministrado por un tercero, en lugar de a través de enlaces WAN dedicados o enlaces de acceso telefónico de larga distancia.

XML Extensible Markup Language (Lenguaje de Marcas Extensible): Es un formato universal para documentos y datos estructurados en Internet; este estándar permite el intercambio de información estructurada entre diferentes plataformas.

WSDL Web Services Description Language (Lenguaje de Descripción de Servicios Web): Lenguaje basado en XML para describir servicios web. Permite describir la interfaz pública de los servicios web; eso significa que detalla los protocolos y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo.