



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE INGENIERÍA

**“DESARROLLO DEL SISTEMA DE LIQUIDACIÓN  
DE VALORES DE UNA CASA DE BOLSA”**

**INFORME DE TRABAJO PROFESIONAL**

QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO EN COMPUTACIÓN

PRESENTA:

SERGIO DAVID JARDÓN ÁVALOS

DIRECTOR DE TRABAJO PROFESIONAL:

M.C. ALEJANDRO VELÁZQUEZ MENA

CIUDAD UNIVERSITARIA, ENERO DE 2015.





## ÍNDICE

<b>1</b>	<b>INTRODUCCIÓN</b> .....	<b>3</b>
1.1	Objetivo General.....	5
1.2	Objetivos Específicos .....	5
<b>2</b>	<b>PROYECTOS</b> .....	<b>9</b>
2.1	Organigrama.....	9
2.2	Proyecto N.1 .....	14
2.3	Proyecto N.2 .....	15
2.4	Proyecto N.3 .....	16
2.5	Proyecto N.4 .....	17
2.6	Proyecto N.5 .....	19
2.7	Proyecto N.6 .....	20
2.8	Proyecto N.7 .....	22
2.9	Proyecto N.8 .....	24
2.10	Proyecto N.9 .....	25
2.11	Proyecto N.10 .....	26
<b>3</b>	<b>METODOLOGÍA</b> .....	<b>31</b>
3.1	Descripción del Proyecto .....	37
3.2	Desempeño Actual y el Deseado .....	37
3.3	Definición del Problema .....	38
3.4	Requerimientos .....	38
3.5	Evaluación de Soluciones .....	40
3.5.1	Interfaz de usuario .....	40
3.5.2	Características de la solución .....	41
3.5.3	JQuery.....	42
3.5.4	Tecnologías para el servidor de aplicaciones. ....	43
3.6	Plan de Trabajo.....	47
3.7	Planeación de liberaciones .....	48
3.7.1	Scrum Team.....	49
3.7.2	Supuestos .....	50
3.7.3	Riesgos.....	51
3.8	Sprint planning .....	52
3.9	Arquitectura del Sistema .....	52
3.9.1	Beneficios .....	53
3.9.2	Diagramas.....	54
3.10	Ejecución y Desarrollo del proyecto .....	54
3.10.1	Desarrollo del Sprint.....	55
3.10.2	Nomenclatura.....	55
3.10.3	Componentes del servidor web.....	57
3.10.4	Componentes del servidor de base de datos y reportes. ....	57
3.10.5	Integración continua .....	57
3.10.6	Esquema de Seguridad.....	58
3.11	Pruebas y Demostración del producto .....	59
3.11.1	Pruebas Unitarias .....	59
3.11.2	Pruebas de Integración.....	60
3.11.3	Pruebas con usuarios de producción.....	60
3.11.4	Pruebas de Carga/Stress.....	60

3.11.5 Pruebas de Liberación .....	60
<b>3.12 Cierre de Sprint .....</b>	<b>61</b>
<b>4 LIBERACIÓN A PRODUCCIÓN.....</b>	<b>65</b>
<b>4.1 Resultados .....</b>	<b>65</b>
<b>CONCLUSIONES.....</b>	<b>69</b>
<b>BIBLIOGRAFÍA Y REFERENCIAS .....</b>	<b>73</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>77</b>
<b>ANEXO A. MINUTAS.....</b>	<b>85</b>
<b>ANEXO B. MATRIZ DE APROBACIONES .....</b>	<b>86</b>
<b>ANEXO C. ESTIMACIÓN POR PLANNING POKER.....</b>	<b>87</b>

## ÍNDICE DE TABLAS

TABLA 1. INFORMACIÓN DEL ÁREA.....	5
TABLA 2. PARTICIPANTES DEL PROYECTO SLV .....	49
TABLA 3. NOMENCLATURA DE PAQUETES. ....	55
TABLA 4. NOMENCLATURA DE OBJETOS. ....	56
TABLA 5. NOMENCLATURA DE ARCHIVOS DE CONFIGURACIÓN. ....	56
TABLA 6. NOMENCLATURA DE COMPONENTES WEB.....	57
TABLA 7. NOMENCLATURA DE OBJETOS DE BASE DE DATOS. ....	57
TABLA 8. REGISTRO DE PRUEBAS UNITARIAS. ....	60
TABLA 9. REGISTRO DE PRUEBAS DE INTEGRACIÓN. ....	60
TABLA 10. REGISTRO DE PRUEBAS CON USUARIO. ....	60
TABLA 11. REGISTRO DE PRUEBAS DE CARGA. ....	60
TABLA 12. REGISTRO DE PRUEBAS DE LIBERACIÓN. ....	61

## ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN I. FLUJO DE OPERACIÓN EN EL MERCADO BURSÁTIL MEXICANO.....	4
ILUSTRACIÓN II. ORGANIGRAMA. ....	10
ILUSTRACIÓN III. ORGANIGRAMA PARA LA INICIATIVA.....	12
ILUSTRACIÓN IV. ÁREAS INVOLUCRADAS .....	13
ILUSTRACIÓN V. FASES DEL SDLC.....	31
ILUSTRACIÓN VI. PROCESO SCRUM .....	34
ILUSTRACIÓN VII. PROCESO SCRUM. ....	35
ILUSTRACIÓN VIII. GRÁFICA BURNDOWN. ....	36
ILUSTRACIÓN IX. APLICACIONES RIA.....	40
ILUSTRACIÓN X. JQUERY GOOGLE TRENDS.....	41
ILUSTRACIÓN XI. UTILIZACIÓN DE JQUERY A NIVEL MUNDIAL. ....	43
ILUSTRACIÓN XII. ARQUITECTURA STRUTS 2. ....	44
ILUSTRACIÓN XIII. SPRING FRAMEWORK.....	45
ILUSTRACIÓN XIV. METODOLOGÍA ÁGIL DEL PROYECTO SLV.....	48
ILUSTRACIÓN XV. RELEASE PLANNING .....	49
ILUSTRACIÓN XVI. DIAGRAMA DE ARQUITECTURA PROPUESTA. ....	54
ILUSTRACIÓN XVII. PROCESO DE INTEGRACIÓN CONTINUA.....	58
ILUSTRACIÓN XVIII. PROCESO DE PRUEBAS. ....	59





## CAPÍTULO 1. INTRODUCCION

---





# 1 Introducción.

El crecimiento tecnológico en diversas áreas como telecomunicaciones, computación y desarrollo de software ha tenido un gran impacto en casi todos los sectores de la sociedad moderna. Es ya muy difícil imaginar nuestra vida cotidiana sin el acceso casi instantáneo a las fuentes de información desde nuestros dispositivos móviles, computadoras personales, tableta o computadoras portátiles. Dentro de esta vertiginosa evolución de las tecnologías de la información, no se encuentran ajenas las instituciones que participan en los mercados bursátiles como son las casas de bolsa, bancos, sociedades de inversión, bancos centrales, depósitos de valores, bolsa de valores, etc. Las telecomunicaciones son el vehículo de información y datos que mueven los capitales y unifican al mundo financiero, los programas de algoritmos de trading procesan millones de datos por segundo y son capaces de marcar tendencias dentro de los mercados financieros.

El sistema Financiero en México es un conjunto de instituciones que se encargan de captar los recursos económicos de personas físicas o morales para ponerlo a disposición de otras empresas o instituciones gubernamentales que lo requieren para invertirlo. Los mercados de valores son una parte imprescindible del juego de los sistemas financieros. El financiamiento mediante el crédito bancario ha perdido importancia ante el financiamiento bursátil. El mercado bursátil, al igual que la actividad bancaria, es un servicio de interés público y su prestación constituye un derecho de los participantes que el estado regula. El mercado de valores se define como el conjunto de leyes, reglamentos, instituciones, intermediarios y participantes en general tendientes a poner en contacto la oferta y la demanda de títulos de crédito.

La Bolsa Mexicana de Valores es el lugar físico donde se efectúan y registran las operaciones que hacen las casas de bolsa. Los inversionistas compran y venden acciones e instrumentos de deuda a través de intermediarios bursátiles, llamados casas de bolsa que a su vez operan en el mercado de valores. El público inversionista del mercado de valores canaliza sus órdenes de compra o venta de acciones a través de un promotor de una casa de bolsa. Estos promotores son especialistas registrados que han recibido capacitación y han sido autorizados por la Comisión Nacional Bancaria y de Valores, por sus siglas CNBV. Las órdenes de compra o venta son entonces transmitidas de la oficina de la casa de bolsa al mercado bursátil a través del sofisticado Sistema Electrónico de Negociación, Transacción, Registro y Asignación (BMV-SENTRA Capitales) donde esperarán encontrar una oferta igual pero en el sentido contrario y así perfeccionar la operación.

La Ley del Mercado de Valores establece que la compensación y liquidación de valores es un servicio público que solo puede ofrecer el banco central, las contrapartes centrales y los depositarios centrales de valores. Los depositarios de valores podrán ofrecer sus servicios a entidades financieras nacionales o extranjeras o a depositarios centrales de valores extranjeros sin actuar como contraparte en tales operaciones. En México, el único depositario de valores es el Instituto de Depósito de Valores, por sus siglas, INDEVAL, dueño y operador del sistema de liquidación de valores DALI.

Como parte de las funciones que desempeña INDEVAL dentro del Mercado de Valores en México, se encuentran los siguientes:

- Depósito y custodia: Consiste en la guarda física de los valores y/o su registro electrónico en la institución autorizada para este fin, la cual asume la responsabilidad por los valores en depósito.
- Administración de Valores: Ejercicios de derechos en efectivo, en especie o mixtos.
- Transferencia: Es el cambio de propiedad de valores. El cambio de propiedad se realiza por el procedimiento de giro o transferencia de cuenta a cuenta mediante los registros que tiene INDEVAL de cada participante.
- Compensación: Mecanismo mediante el cual se determinan los importes y volúmenes

netos a intercambiar en dinero y valores entre las contrapartes de una operación.

- **Liquidación:** Proceso a través del cual las contrapartes cumplen con las obligaciones derivadas de una operación, es decir, un mecanismo para que los valores sean traspasados de las cuentas de valores de un nuevo propietario, y así mismo, el efectivo sea acreditado a las cuentas de la contraparte correspondiente.

En la Ilustración I se muestra el flujo general de una operación dentro del mercado de Valores en México.

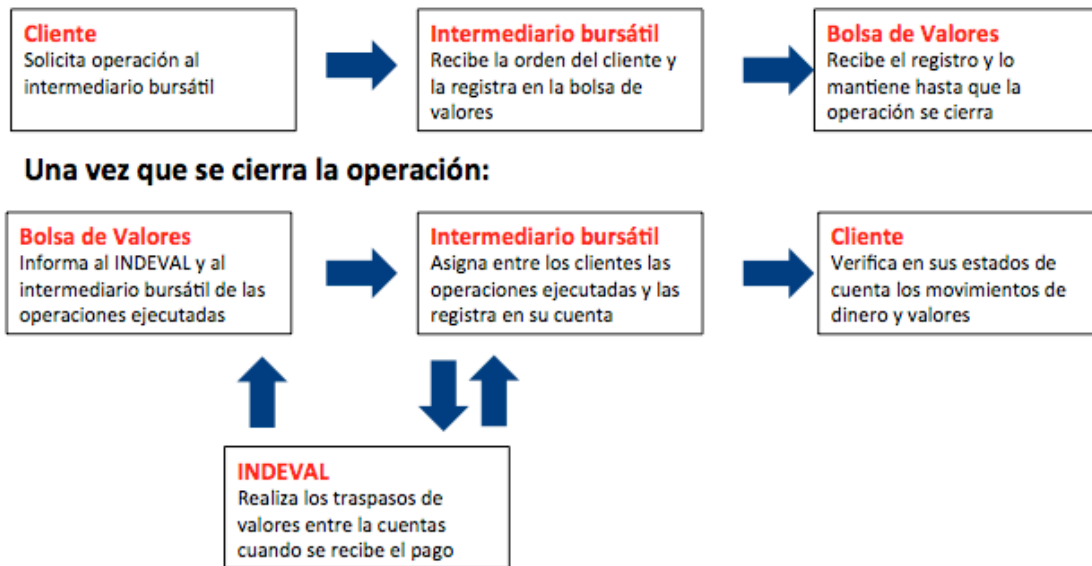


ILUSTRACIÓN I. FLUJO DE OPERACIÓN EN EL MERCADO BURSÁTIL MEXICANO.

Dentro de este informe profesional, se redacta la implementación de un sistema que sirve para la liquidación y traspaso de títulos para una Casa de Bolsa para las áreas de Back Office de Valores, encargada de asegurar el cumplimiento de las operaciones realizadas y pactadas entre los intermediarios participantes en el mercado bursátil mexicano. Para simplificar su referencia, llamaremos a este sistema como SLV.

Previo al desarrollo de esta iniciativa, la liquidación de títulos con el depositario central de valores, INDEVAL, se llevaba a cabo de manera manual a través de la exportación de archivos separados por coma y complementando información de diversos reportes o de forma manual. Esto conllevaba a diversos errores operativos en la captura de la información, identificación tardía de errores, que a su vez implicaban multas de las entidades reguladoras por falta de controles automáticas así como por incumplimientos operativos.

Cabe mencionar que parte central del proyecto, era la automatización de la operatividad debido a los altos volúmenes transaccionales que incrementaban de forma anual en los sistemas de la casa de bolsa. Además de la automatización, también se requerían controles adecuados para monitorear y garantizar la correcta operación del día.

A continuación, en la tabla 1 se muestran algunas cifras relacionadas con el área de liquidación de valores.

T a b l a 1 . I n f o r m a c i ó n d e l á r e a

Cantidad de usuarios	Usuarios concurrentes	Transacciones mensuales	Crecimiento esperado	Información histórica
100	30	50,000	10%	10 años en línea

Debido a la expectativa de crecimiento en cuanto a volumen operativo y de información, también se consideraron otros aspectos como performance, calidad del servicio, información en tiempo real, alarmas operativas, alertas tecnológicas, entre otras mejoras que se diseñaron como solución del sistema.

### **1.1 Objetivo General.**

El objetivo de esta iniciativa, fue implementar un sistema dentro de la Casa de Bolsa capaz de realizar la liquidación de valores con el depositario central de valores en México, INDEVAL, de forma automática.

Con esta implementación se esperaba disminuir los costos operativos al reducir el personal que da seguimiento a la liquidación de forma manual, así como los riesgos y multas de las autoridades centrales por incumplimientos o quebrantos. De igual manera, como parte de la alineación de las estrategias de tecnología con los lineamientos globales de la Casa de Bolsa, se buscaba desarrollar una plataforma que cumpliera con los estándares tecnológicos utilizando componentes y frameworks que permitieran el desarrollo de un sistema con un nivel de desempeño óptimo para el volumen de operaciones, una interfaz de usuario amigable y cumplir con los estándares de seguridad de la información.

Dentro de esta iniciativa participé como Scrum Master de Enero del 2013 a Noviembre del 2013.

### **1.2 Objetivos Específicos.**

A continuación se detallan los objetivos específicos que busca el desarrollo del sistema SLV:

- Automatizar la liquidación de las operaciones de traspasos de títulos de la Casa de Bolsa.
- Utilizar un protocolo de comunicación estándar para el intercambio de información bursátil con el depósito central de valores. Esta mensajería se conoce como SWIFT.
- Implementar los controles y alertas necesarias para la correcta liquidación de las operaciones de traspaso de valores.
- Implementar un Front End para los usuarios de Back Office de Valores para el seguimiento y monitoreo de operaciones en tiempo real.
- Garantizar la entrega y recepción de mensajes bursátiles con INDEVAL

- Poder generar reportes de cierre de día y consultar información histórica de hasta 10 años.
- Generar gráficas y estadísticas con la información clasificada por área de negocio, volumen de operaciones y monto de operaciones.
- Utilizar tecnología que permita tener un rendimiento óptimo para el manejo de volumen operativo.
- Definir accesos y privilegios de usuarios de acuerdo a los estándares establecidos por el área de seguridad de la información y los requerimientos del usuario.
- Establecer los estándares para metodologías de desarrollo ágiles dentro de la Casa de Bolsa.

Los objetivos fueron definidos por el usuario de negocio responsable de la iniciativa, y los cuales fueron revisados por el área de tecnología y sistemas. De acuerdo al proceso de administración de proyectos se registró el caso de negocio y los requerimientos de alto nivel en el documento de plan de proyecto para formalizar el alta de la iniciativa en la Oficina de Proyectos, con las aprobaciones de todos los interesados.

En los capítulos contenidos en este informe profesional se presenta la siguiente información:

#### Capítulo 1

En este capítulo se presenta el objetivo general y específico de la iniciativa en la cual participe en el año 2013. Este proyecto fue elegido como piloto para metodologías ágiles dentro del Grupo Financiero. De igual forma, se presenta el organigrama de la empresa y el organigrama de la atención de la iniciativa.

#### Capítulo 2

En este capítulo se presenta un resumen de los proyectos más significativos a los cuales me he enfrentado desde mi ingreso al campo laboral en el año 2005, adicional a éstos, en mi experiencia profesional he trabajado en proyectos de mantenimiento de los sistemas y atención de la operación del día de las áreas de la Casa de Bolsa.

#### Capítulo 3

En este capítulo se detalla los pasos que se siguieron desde la necesidad de crear el proyecto, para poder dar solución a un problema existente, hasta el cierre del mismo. Se detallan todos los pasos que se siguieron bajo la metodología en cascada del desarrollo de sistemas para su ejecución y éxito.

#### Capítulo 4

En este capítulo se detallan los resultados que generó este proyecto, las actividades que se realizaron en su post-implementación, las conclusiones de mi participación en el proyecto y la bibliografía de referencia.

#### Anexo

Se despliega el formato de las minutas que se utilizó durante la vida del proyecto; así como la matriz de aprobaciones que se utilizó para cada cambio de fase del proyecto.

## CAPÍTULO 2. PROYECTOS

---



## 2 **Proyectos.**

### 2.1 **Organigrama.**

La estructura organizacional del departamento de Tecnología y Sistemas de la Casa de Bolsa cuenta con diversas áreas para poder dar soporte y mantenimiento a las necesidades de negocio de la misma. Cada una de las áreas tiene asignada ciertos roles y tareas dentro de la organización de acuerdo a su especialidad.

Las principales funciones de las distintas áreas de Tecnología y Sistemas se describen a continuación.

- Oficina de proyectos. Encargada de establecer los lineamientos para los procesos de proyectos así como de los reportes de estatus semanales y mensuales a la Dirección de Sistemas.
- Operaciones. Encargada del mantenimiento y monitoreo de los Servidores de la Casa de Bolsa. Dentro de sus funciones se encuentran a cargo los servidores de base de datos, Web, Windows, etc.
- Calidad. Encargada de realizar las pruebas descritas en los documentos de aceptación y pruebas del usuario. Aseguramiento de la calidad de los componentes de software. También se aseguran del cumplimiento de los procesos de Software descritos en los lineamientos de Tecnología.
- Arquitectura y Diseño. Encargados de asegurar que los componentes y desarrollos de Sistemas dentro de la Casa de Bolsa, cumplan con los lineamientos establecidos por Tecnología y Sistemas. También aprueban los diseños y soluciones propuestas por el área de Desarrollo de Sistemas.
- Change Management. Realizar la planeación y organización de los cambios en los servidores o aplicaciones de la Casa de Bolsa.
- Internal Compliance. Asegurar los cumplimientos con las políticas internas para el desarrollo de software así como estándares establecidos por la regulación del mercado.
- Seguridad de la Información. Encargada de administrar los permisos y roles de los usuarios dentro de las aplicaciones de la Casa de Bolsa.
- Software Management. Encargada de definir los procesos de software para cumplir con los estándares establecidos por Tecnología y Sistemas globales. También se encarga de establecer la documentación para cumplir con cada una de las fases de los procesos. Define las métricas para evaluar a los miembros del área de Tecnología y Sistemas conocidas como Key Performance Indicators (KPI's).
- Centro de Desarrollo de Sistemas: Encargada de realizar la implementación, mantenimiento y soporte de los sistemas internos y de terceros de la Casa de Bolsa. Dividido en dos subdirecciones especializadas en atender a los distintos negocios de la Casa de Bolsa.



Como integrante de la Casa de Bolsa, me encontraba laborando dentro del área de Tecnología y Sistemas, bajo la subdirección del Centro de Desarrollo 1, el cual se encarga de dar soporte a diversos usuarios de las áreas de Back Office de Valores y Tesorería, Sociedades de Inversión, Recursos Humanos, contabilidad, Contratos, Control y Proceso, Compliance entre otros. Durante mi estancia en la Casa de Bolsa, tuve distintos roles desde Ing. Sistemas Junior y Senior con un perfil de desarrollador especialista, hasta líder de proyecto y gerente de sistemas con tres equipos de trabajo bajo mi responsabilidad, con un total de 12 personas a mi cargo.

En la Ilustración II se muestra el organigrama dentro del cual desempeñé mis funciones en el área de Tecnología y Sistemas.

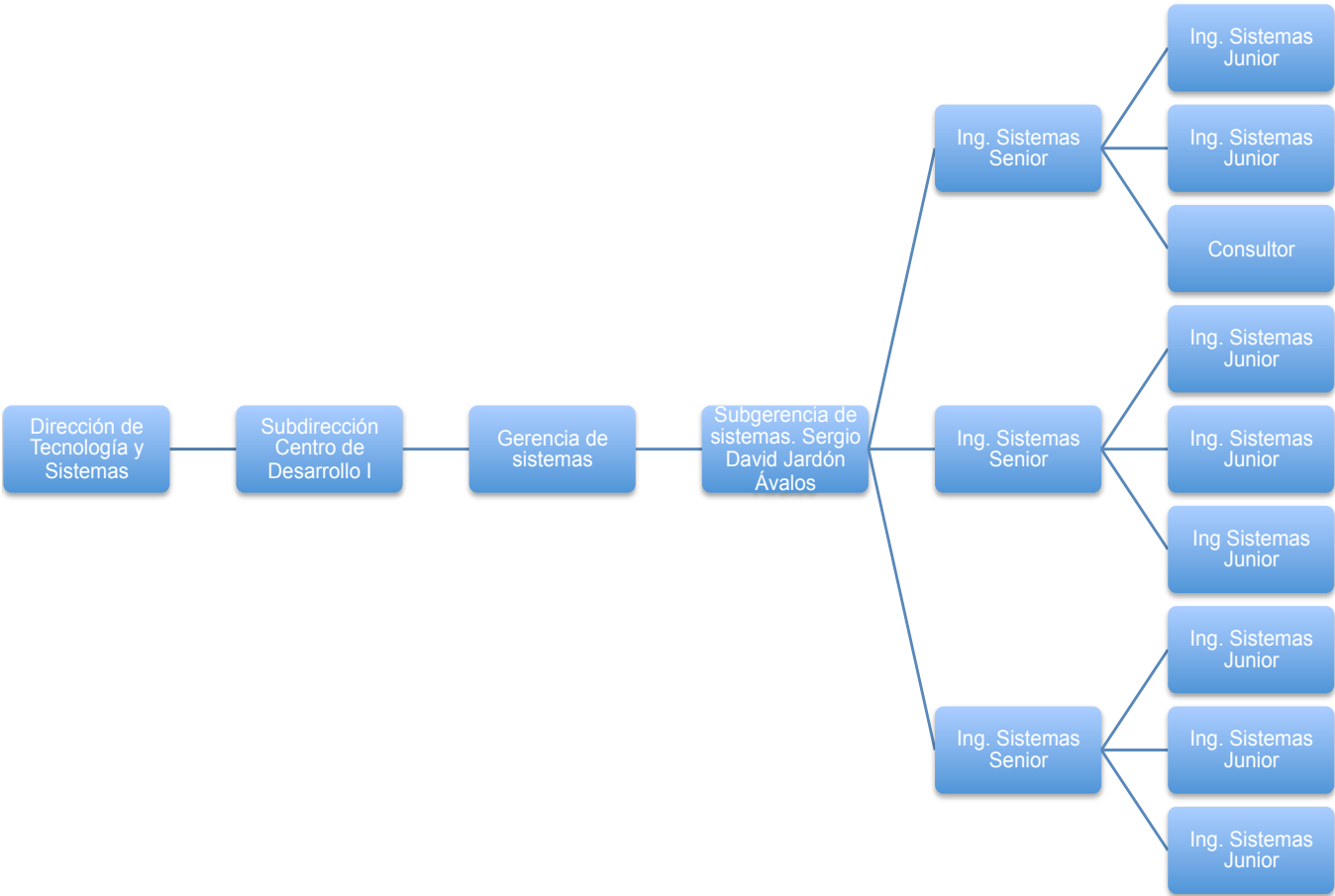


ILUSTRACIÓN II. ORGANIGRAMA.

Como parte de mis responsabilidades dentro de la Casa de Bolsa se encontraban las siguientes funciones descritas a continuación:

- Elaboración del plan de proyecto, estableciendo el alcance inicial, identificar a los principales usuarios interesados, conformación del equipo de trabajo.
- Elaboración de estimados de costos, tiempo y recursos.
- Definición de requerimientos del sistema y criterios de aceptación con los usuarios.
- Elaboración del plan de comunicaciones y estrategia de comunicación.
- Desarrollo de plan de pruebas con el área de Calidad.
- Definición de Tecnologías e interfaces requeridas con el visto bueno del área de Arquitectura.
- Seguimiento y Control de las actividades de Desarrollo
- Gestión de Riesgos y problemas.
- Control del proyecto, modificaciones al alcance, tiempo o recursos de acuerdo a las necesidades del proyecto.
- Coordinación con las áreas de Calidad, Arquitectura, Change Management para cumplir con las líneas base del proyecto.
- Seguimiento a las pruebas integrales y funcionales con el usuario final.
- Obtener visto bueno del usuario para la implementación en producción.
- Presentaciones y juntas en el comité de Sistemas para reportes de avance.
- Juntas de seguimiento diarias y semanales.
- Cierre del proyecto y documentación de lecciones aprendidas y mejores prácticas.
- Evaluación del proyecto de acuerdo al cumplimiento establecido en el plan de trabajo final.

Debido a las expectativas de las áreas de negocio con el proyecto, las necesidades de cumplir con un tiempo para que el producto saliera al mercado, o time to market en inglés, y requerimientos que se encontraban dimensionados en un nivel muy alto de definición, fue necesario adoptar metodologías de desarrollo de sistemas ágiles. Se determinó utilizar Scrum como metodología ágil para la administración y desarrollo del proyecto. Como resultado, el proyecto se dividió en 11 iteraciones o sprints como se le conoce en la metodología, de 4 semanas cada uno. De acuerdo a la metodología, fue necesario conformar un equipo de trabajo multidisciplinario: 1 scrum master o líder de proyecto, 2 Ingenieros en Sistemas Junior, 1 Arquitecto de Sistemas, 1 Ingeniero de pruebas, 1 administrador de cambios, 1 representante de negocio o usuario final. Durante mi participación en el proyecto me desempeñé como Scrum Master debido a mi experiencia y conocimientos dentro de la Casa de Bolsa. En el desarrollo del mismo proyecto, fue necesario involucrar a otros equipos de trabajo debido al impacto que tenía, en particular con las áreas de Préstamo de Valores, Sociedades de Inversión, Arbitraje, Internacional, Valores, Custodia Externa y Asset Management.

En la Ilustración III se muestra el organigrama bajo el cual se desarrolló la iniciativa del sistema



ILUSTRACIÓN III. ORGANIGRAMA PARA LA INICIATIVA

Para poder llevar a cabo el proyecto, fue necesario contar con la participación de diversas áreas de Tecnología y Sistemas, esto para garantizar una visión integral de la solución implementada, tomando en cuenta los distintos ángulos en el desarrollo de software, y realizando un balance entre las necesidades del negocio y las necesidades del área de Tecnología y Sistemas.

De igual manera, el alcance definido en los requerimientos del usuario, tenían un impacto o interacción con otras áreas de negocio dentro de la Casa de Bolsa.

En la Ilustración IV se muestran las áreas involucradas dentro del proyecto.



ILUSTRACIÓN IV. ÁREAS INVOLUCRADAS

## 2.2 Proyecto N.1

**Nombre:** Cifrado de información de cuentas de clientes de banca privada en una Casa de Bolsa.

**Rol:** Ingeniero en Sistemas Junior.

**Recursos:** Dos Ingenieros en Sistemas Junior.

**Año – Duración:** Sep 2005 – Ene 2006.

### Objetivo

Rol de Ing. Sistemas Junior en la implementación de un componente para migrar la información de cuentas de clientes de banca privada para una Casa de Bolsa. El objetivo del proyecto era cifrar la información demográfica de los clientes de banca privada como son nombres completos, domicilios, datos de contacto, en un algoritmo que cumpliera con los requerimientos mínimos de seguridad para la institución, y que dicha información pudiera ser consultada a través de los diversos sistemas de Fronts para negocio, de acuerdo al perfil de seguridad de los usuarios.

### Características

1. Lenguaje de programación: Visual Basic 6.0, Sistema Web desarrollado en J2EE, JSP's.
2. Modelo: Cliente Servidor / Modelo, Vista, Controlador.
3. Base de Datos: Microsoft SQL Server.
4. Servidor de Aplicaciones: Windows Server 2003, Weblogic Server, Sun Solaris Iplanet.
5. Sistema Operativo: UNIX, Windows.

### Principales Actividades

1. Análisis y Diseño del sistema.

Realizar la documentación para las fases de análisis y diseño funcional de la solución propuesta de acuerdo a los estándares de calidad. Establecer con el usuario los puntos funcionales, criterios de aceptación y alcance de la solución de acuerdo a los requerimientos del negocio.

2. Diseño técnico del sistema.

Realizar el diseño técnico y arquitectónico de la solución propuesta. Asegurar la generación del documento para la fase de diseño del proyecto de acuerdo al alcance establecido en el plan de proyecto. Proponer los componentes y frameworks para implementar la solución. Obtener la aprobación del arquitecto de acuerdo a los estándares internos. Elaborar los diagramas de componentes y de entidad relación. Obtener la aprobación del administrador de la base de datos.

3. Implementación y desarrollo del sistema.

4. Desarrollo de los componentes Web en Java para el Front de usuarios de banca patrimonial. Implementación de la solución de acuerdo a lo establecido en el documento de diseño, pruebas unitarias y de regresión. Desarrollo de los sistemas Visual Basic para los usuarios del área de Contratos. Implementación de la solución de acuerdo a lo establecido en el documento de diseño, pruebas unitarias y de regresión.

5. Pruebas y Calidad.

Ejecución y seguimiento de las pruebas técnicas y funcionales en el área de calidad. Asegurar con el usuario final que la solución implementada cumpliera los requisitos establecidos en el alcance del proyecto.

6. Cierre del proyecto.

Realizar la documentación de cierre del proyecto y lecciones aprendidas.

### 2.3 Proyecto N.2

**Nombre:** Desarrollo de Sistema de Transacciones para una Casa de Bolsa.

**Rol:** Ingeniero en Sistemas Junior.

**Recursos:** Dos ingenieros en sistemas Junior.

**Año - Duración:** Ene 2006 – Jun 2006.

#### Objetivo

Rol de Ing. Sistemas Junior en la implementación de un sistema encargado de dar mantenimiento a los catálogos de transacciones de una Casa de Bolsa. El objetivo del sistema era crear un Front-End para el área de Control y Procesos con diversos perfiles, mantenimientos, bitácoras, seguimiento de auditoría, reportes, búsquedas para el mantenimiento de los catálogos que definen el motor transaccional de la Casa de Bolsa.

#### Características

1. Lenguaje de programación: Sistema Web desarrollado en J2EE, JSP's,
2. Frameworks y tecnologías utilizadas: Struts, Java.
3. Modelo: Modelo, Vista, Controlador.
4. Servidor de Aplicaciones: Weblogic Server, Sun Solaris Iplanet.
5. Base de datos: Microsoft SQL Server.
6. Sistema Operativo: UNIX.

#### Principales Actividades

1. Análisis y diseño del sistema.

Realizar la documentación para las fases de análisis y diseño funcional de la solución propuesta de acuerdo a los estándares de calidad. Establecer con el usuario los puntos funcionales, criterios de aceptación y alcance de la solución de acuerdo a los requerimientos del negocio.

2. Diseño técnico del sistema.

Realizar el diseño técnico y arquitectónico de la solución propuesta. Asegurar la generación del documento para la fase de diseño del proyecto de acuerdo al alcance establecido en el plan de proyecto. Proponer los componentes y marcos de trabajo o frameworks para implementar la solución. Obtener la aprobación del arquitecto de acuerdo a los estándares internos. Elaborar los diagramas de componentes y de entidad relación. Obtener la aprobación del administrador de la base de datos.

3. Implementación y desarrollo del sistema.

Desarrollo de los componentes de acuerdo a lo establecido en el documento de diseño.

4. Pruebas y Calidad.

Ejecución y seguimiento de las pruebas técnicas y funcionales en el área de calidad. Asegurar con el usuario final que la solución implementada cumpliera los requisitos establecidos en el alcance del proyecto.

5. Cierre del proyecto.

Realizar la documentación de cierre del proyecto y lecciones aprendidas.

## **2.4 Proyecto N.3**

**Nombre:** Implementación de sistema de liquidación de títulos para una Casa de Bolsa.

**Rol:** Líder de Proyecto.

**Recursos:** Dos ingenieros en sistemas Junior.

**Año - Duración:** Jul 2006 – Dic 2006.

### **Objetivo**

Rol de Ing. Sistemas Sr en la implementación de un sistema encargado de realizar la liquidación de títulos para las cuentas de la Casa de Bolsa con el depósito Central de Valores de México. El objetivo del sistema, era calcular la cantidad de títulos para realizar los traspasos con las contrapartes o intermediarios, realizar la liquidación con el Depósito Central de Valores (INDEVAL) mediante el estándar financiero 15022 (SWIFT), dar seguimiento a las liquidaciones mediante el manejo de mensajes de estatus recibidos a través del INDEVAL, realizar un Front para los usuarios del Back Office de Valores.

### **Características**

1. Lenguaje de programación: Sistema Web desarrollado en J2EE, JSP's,
2. Frameworks y tecnologías: Struts, iText, AJAX, JMS.
3. Modelo: Modelo, Vista, Controlador.
4. Servidor de Aplicaciones: Weblogic Server, Sun Solaris Iplanet.
5. Base de datos: Microsoft SQL Server.
6. Sistema Operativo: UNIX.

### **Principales Actividades**

1. Planeación e Inicio del proyecto.

Realizar el plan de proyecto de acuerdo a los requerimientos del usuario. Identificar a los principales interesados en el proyecto y usuarios afectados. Realizar documentación de estimación de costos, tiempo y recursos iniciales. Obtener aprobación del patrocinador del proyecto.

## 2. Análisis y Diseño del Sistema.

Realizar la documentación para las fases de análisis y diseño funcional de la solución propuesta de acuerdo a los estándares de calidad. Establecer con el usuario los puntos funcionales, criterios de aceptación y alcance de la solución de acuerdo a los requerimientos del negocio.

## 3. Diseño técnico de la solución.

Realizar el diseño técnico y arquitectónico de la solución propuesta. Asegurar la generación del documento para la fase de diseño del proyecto de acuerdo al alcance establecido en el plan de proyecto. Proponer los componentes y marcos de trabajo o frameworks para implementar la solución. Obtener la aprobación del arquitecto de acuerdo a los estándares internos. Elaborar los diagramas de componentes y de entidad relación. Obtener la aprobación del administrador de la base de datos.

## 4. Desarrollo e implementación del sistema.

Desarrollo de los componentes. Implementación de la solución de acuerdo a lo establecido en el documento de diseño, pruebas unitarias y de regresión. Ejecución y seguimiento de las pruebas técnicas y funcionales en el área de calidad.

## 5. Pruebas y Calidad.

Dar seguimiento a la corrección de defectos técnicos y funcionales. Asegurar con el usuario final que la solución implementada cumpliera los requisitos establecidos en el alcance del proyecto.

## 6. Cierre del proyecto.

Realizar la documentación de cierre del proyecto y lecciones aprendidas.

### **2.5 Proyecto N.4**

**Nombre:** Desarrollo de módulo de apertura y de información de cliente para un portal de una Casa de Bolsa y Banco.

**Rol:** Líder de Proyecto.

**Recursos:** Tres ingenieros en sistemas Junior.

**Año - Duración:** Ene 2009 – Dic 2009.

#### **Objetivo**

Rol de Líder de proyecto en la implementación del módulo de apertura de cuentas de clientes para un portal financiero de una Casa de Bolsa y Banco. El objetivo del módulo, era crear un Front-End Web disponible desde Internet, para que los clientes pudieran realizar la solicitud de apertura de un contrato de intermediación bursátil a través de dicho sistema. Dentro del mismo, el cliente era capaz de anexar toda la información y documentación necesaria para completar el proceso de apertura sin necesidad de presentarse a una sucursal. Dentro del módulo, el cliente podía dar seguimiento a su solicitud de inicio hasta que se completara el proceso con la firma de su contrato de inversión.



## Características

1. Lenguaje de programación: Sistema Web desarrollado en J2EE, JSP's,
2. Frameworks tecnologías: Struts, iText, Flex.
3. Modelo: Modelo, Vista, Controlador.
4. Servidor de Aplicaciones: Weblogic Server, Sun Solaris Iplanet.
5. Base de datos: Microsoft SQL Server.
6. Sistema Operativo: UNIX.

## Principales Actividades

1. Planeación e inicio.

Elaboración de plan de proyecto e identificación de principales usuarios interesados. Definición de roles y responsabilidades. Elaborar alcance, tiempo y recursos iniciales. Establecer plan de comunicaciones y seguimiento.

2. Análisis del sistema.

Coordinar juntas de definición y alcance inicial con usuarios. Establecer criterios de aceptación. Obtener aprobación de análisis y alcance del proyecto.

3. Diseño técnico.

Coordinar juntas de definición y diseño técnico con el área de arquitectura. Obtener aprobación de solución propuesta. Evaluar productos del mercado y soluciones internas.

4. Plan de pruebas.

Coordinar juntas de definición y diseño de plan de pruebas funcionales y técnicas con área de calidad. Elaboración de plan de pruebas.

5. Coordinación de las actividades de desarrollo.

Coordinar a los integrantes del equipo de desarrollo para asegurar la ejecución de las tareas asignadas a cada miembro del equipo y cumplir con lo establecido en el plan de proyecto de acuerdo al alcance y tiempos. Asegurar que se desarrolle de acuerdo a los estándares establecidos.

6. Coordinación del equipo de trabajo.

Asegurar el cumplimiento de las tareas asignadas a cada recurso del proyecto así como dar seguimiento al progreso del proyecto de acuerdo a lo establecido en tiempo, recursos y alcance. Juntas de seguimiento con los líderes asignados a los distintos módulos del proyecto.

7. Seguimiento de pruebas.

Coordinar y participar en las pruebas integrales con el usuario para garantizar la promoción de un producto de acuerdo a lo establecido en los requerimientos y los criterios de aceptación.

8. Cierre del proyecto.

Elaborar documento de cierre del proyecto. Establecer los criterios de éxito del proyecto

con base en las líneas base de tiempo, recursos y costo. Documentación de lecciones aprendidas y junta de cierre.

## **2.6 Proyecto N.5**

**Nombre:** Desarrollo del sistema de administración de valores para una Casa de Bolsa.

**Rol:** Líder de Proyecto.

**Recursos:** Tres ingenieros en sistemas Junior, dos consultores externos.

**Año - Duración:** Dic 2009 – Mar 2012.

### **Objetivo**

Rol de líder de proyecto en la implementación de un sistema de administración de Valores para una Casa de Bolsa. El objetivo del sistema era el mantenimiento de las emisoras del mercado, con sus características como valor nominal, nombre de la emisora, serie, tipo de valor, ISIN, fechas de corte de cupón, fechas de vencimiento, así como de la creación, mantenimiento y pago de los ejercicios de derechos patrimoniales y corporativos decretados por los participantes en el mercado bursátil.

Este proyecto fue desarrollado por un equipo externo para el motor encargado del cálculo y definición de los ejercicios de derechos, y de un equipo interno para el desarrollo de interfaces e integración del sistema dentro del entorno de la Casa de Bolsa

El proyecto fue dividido en tres fases: la primer fase constaba del mantenimiento de emisoras, la segunda fase Derechos Corporativos y generador de derechos y la tercer fase derechos patrimoniales y liquidación de derechos.

### **Características**

1. Lenguaje de programación: Sistema Web desarrollado en J2EE, JSP's
2. Frameworks: Struts, Spring, Ibatis, Enterprise Service Bus, JMS, Flex.
3. Modelo: Modelo, Vista, Controlador.
4. Servidor de Aplicaciones: Weblogic Server, Sun Solaris Iplanet.
5. Base de datos: Microsoft SQL Server
6. Sistema Operativo: LINUX

### **Principales Actividades**

1. Planeación e inicio.

Elaboración de plan de proyecto e identificación de usuarios interesados. Definición de roles y responsabilidades. Elaborar documentación de alcance, tiempo y recursos iniciales. Establecer plan de comunicaciones y seguimiento. Elaboración de contrato de confidencialidad y plan de pagos a proveedor.

2. Análisis del sistema.

Coordinar juntas de definición y alcance inicial con usuarios. Establecer criterios de aceptación del proyecto. Obtener aprobación de análisis y alcance del proyecto. Identificar principales riesgos y problemas en el proyecto.

3. Diseño técnico del sistema.

Coordinar juntas de definición y diseño técnico con el área de arquitectura. Obtener aprobación de solución propuesta. Evaluar productos del mercado y soluciones internas.

4. Plan de pruebas y Calidad.

Coordinar junta de definición y diseño de plan de pruebas funcionales y técnicas con área de calidad. Elaboración de plan de pruebas. Establecer criterios de calidad de volumen, tiempo de respuesta, usabilidad en el sistema de acuerdo a los requerimientos del usuario.

5. Coordinación de las actividades de desarrollo.

Coordinar a los integrantes del equipo de desarrollo para asegurar la ejecución de las tareas asignadas a cada miembro del equipo y cumplir con lo establecido en el plan de proyecto de acuerdo al alcance y tiempos. Asegurar que se desarrolle de acuerdo a los estándares establecidos en el área de Tecnología y Sistemas.

6. Coordinación del equipo de trabajo.

Asegurar el cumplimiento de las tareas asignadas a cada recurso del proyecto así como dar seguimiento al progreso del proyecto de acuerdo a lo establecido en tiempo, recursos y alcance. Coordinar juntas de seguimiento con los proveedores para seguimiento de entregables. Actualizar las líneas base del proyecto para costos, tiempo y recursos.

7. Seguimiento de pruebas.

Coordinar y participar en las pruebas integrales con el usuario para garantizar la promoción de un producto de acuerdo a lo establecido en los requerimientos y los criterios de aceptación. Asegurar que los defectos encontrados fueran atendidos y cerrados de acuerdo a la criticidad dentro del proyecto.

8. Cierre del proyecto.

Elaborar documento de cierre del proyecto. Establecer los criterios de éxito del proyecto con base en las líneas base de tiempo, recursos y costo. Documentación de lecciones aprendidas y junta de cierre.

## **2.7 Proyecto N.6**

**Nombre:** Migración de sistema de pagos a terceros para una Casa de Bolsa.

**Rol:** Líder de Proyecto.

**Recursos:** Dos ingenieros en sistemas Junior.

**Año - Duración:** Abr 2012 – Jun 2012.

### **Objetivo**

Rol de líder de proyecto en la implementación de un sistema para realizar los pagos a terceros de los clientes de la Casa de Bolsa. El objetivo del proyecto era cumplir con las regulaciones establecidas por el Banco Central de México (BANXICO) para realizar el pago a cuentas de terceros entre los intermediarios en menos de 30 segundos una vez autorizadas las operaciones por las áreas de control internas.

El proyecto fue desarrollado por un equipo externo para migrar la plataforma de pagos y un equipo interno de desarrollo de interfaces. De acuerdo a los estándares internos, la migración de la plataforma de pagos también se migró de infraestructura tecnológica.

### **Características**

1. Lenguaje de programación: Sistema Web desarrollado en J2EE, JSP's.
2. Frameworks: Struts, Spring, Ibatis, EJB's.
3. Modelo: Modelo, Vista, Controlador.
4. Servidor de Aplicaciones: Weblogic Server, Sun Solaris Iplanet.
5. Base de datos: Microsoft SQL Server.
6. Sistema Operativo: UNIX.

### **Principales Actividades**

1. Planeación e inicio.

Elaboración de plan de proyecto e identificación de usuarios interesados. Definición de roles y responsabilidades. Elaborar documentación de alcance, tiempo y recursos iniciales. Establecer plan de comunicaciones y seguimiento. Establecer criterios de éxito del proyecto. Elaboración de contrato de confidencialidad y plan de pagos a proveedor. Identificar principales riesgos y problemas del proyecto.

2. Análisis del sistema.

Coordinar juntas de definición y alcance inicial con usuarios. Establecer criterios de aceptación. Obtener aprobación de análisis y alcance del proyecto.

3. Diseño técnico.

Coordinar juntas de definición y diseño técnico con el área de arquitectura. Obtener aprobación de solución propuesta. Evaluar productos del mercado y soluciones internas.

4. Plan de pruebas.

Coordinar junta de definición y diseño de plan de pruebas funcionales y técnicas con área de calidad. Elaboración de plan de pruebas. Establecer criterios de calidad de volumen, tiempo de respuesta, usabilidad en el sistema de acuerdo a los requerimientos del usuario.

5. Coordinación de las actividades de desarrollo.

Coordinar a los integrantes del equipo de desarrollo para asegurar la ejecución de las tareas asignadas a cada miembro del equipo y cumplir con lo establecido en el plan de proyecto de acuerdo al alcance y tiempos. Asegurar que se desarrolle de acuerdo a los estándares establecidos de acuerdo a los lineamientos de Tecnología y Sistemas.

6. Coordinación del equipo de trabajo.

Asegurar el cumplimiento de las tareas asignadas a cada recurso del proyecto así como dar seguimiento al progreso del proyecto de acuerdo a lo establecido en tiempo, recursos y alcance. Coordinar juntas de seguimiento con los proveedores para seguimiento de entregables.

7. Seguimiento de pruebas.

Coordinar y participar en las pruebas integrales con el usuario para garantizar la promoción de un producto de acuerdo a lo establecido en los requerimientos y los criterios de aceptación. Asegurar que los defectos encontrados fueran atendidos y cerrados de acuerdo a la criticidad dentro del proyecto.

8. Cierre del proyecto.

Elaborar documento de cierre del proyecto. Establecer los criterios de éxito del proyecto con base en las líneas base de tiempo, recursos y costo. Documentación de lecciones aprendidas y junta de cierre.

## **2.8 Proyecto N.7**

**Nombre:** Sistema de control de la posición propia de una Casa de Bolsa.

**Rol:** Líder de Proyecto.

**Recursos:** Dos ingenieros en sistemas Junior.

**Año - Duración:** Jun 2012 – Dic 2012.

### **Objetivo**

Rol de líder de proyecto en la implementación de un sistema para realizar el cálculo de la posición interna de la Casa de Bolsa. El objetivo del sistema era calcular la posición de las distintas áreas de negocio de la Casa de Bolsa como son: Corretaje, Deuda, Préstamo de Valores, Internacional, Arbitraje, Tesorería entre otras. El sistema realizaba un cálculo de la posición en línea de las cuentas de las áreas, y mediante un Front-End para el usuario y mediante parámetros configurables, mostraba posibles quebrantos y posiciones en corto de los contratos. El sistema de igual manera generaba de manera automática los traspasos de efectivo para cubrir dichos cortos entre las áreas internas.

### **Características**

1. Lenguaje de programación: Sistema Web desarrollado en J2EE, JSP's.
2. Frameworks: Struts, Spring, Ibatis, JQuery.
3. Modelo: Modelo, Vista, Controlador.
4. Servidor de Aplicaciones: Weblogic Server, Sun Solaris Iplanet.
5. Base de datos: Microsoft SQL Server.
6. Sistema Operativo: UNIX

### **Principales Actividades**

1. Planeación e inicio.

Elaboración de plan de proyecto e identificación de usuarios interesados. Definición de roles y responsabilidades. Elaborar documentación de alcance, tiempo y recursos iniciales. Establecer plan de comunicaciones y seguimiento. Establecer criterios de éxito del proyecto.

2. Análisis del sistema.

Coordinar juntas de definición y alcance inicial con usuarios. Establecer criterios de aceptación de los requerimientos. Obtener aprobación de análisis y alcance del proyecto.

3. Diseño técnico.

Juntas de definición y diseño técnico con el área de arquitectura. Obtener aprobación de solución propuesta. Evaluar productos del mercado y soluciones internas.

4. Plan de pruebas.

Junta de definición y diseño de plan de pruebas funcionales y técnicas con área de calidad. Elaboración de plan de pruebas. Establecer criterios de calidad de volumen, tiempo de respuesta, usabilidad en el sistema de acuerdo a los requerimientos del usuario.

5. Coordinación de las actividades de desarrollo.

Coordinar a los integrantes del equipo de desarrollo para asegurar la ejecución de las tareas asignadas a cada miembro del equipo y cumplir con lo establecido en el plan de proyecto de acuerdo al alcance y tiempos. Asegurar que se desarrolle de acuerdo a los estándares establecidos.

6. Coordinación del equipo de trabajo.

Asegurar el cumplimiento de las tareas asignadas a cada recurso del proyecto así como dar seguimiento al progreso del proyecto de acuerdo a lo establecido en tiempo, recursos y alcance. Juntas de seguimiento con los proveedores para seguimiento de entregables.

7. Seguimiento de pruebas.

Coordinar y participar en las pruebas integrales con el usuario para garantizar la promoción de un producto de acuerdo a lo establecido en los requerimientos y los criterios de aceptación. Asegurar que los defectos encontrados fueran atendidos y cerrados de acuerdo a la criticidad dentro del proyecto.

8. Cierre del proyecto.

Elaborar documento de cierre del proyecto. Establecer los criterios de éxito del proyecto con base en las líneas base de tiempo, recursos y costo. Documentación de lecciones aprendidas y junta de cierre.

## 2.9 Proyecto N.8

**Nombre:** Sistema de control de la liquidación de Valores.

**Rol:** Líder de Proyecto.

**Recursos:** 3 ingenieros en sistemas Junior, 1 Ingenieros en Sistemas Senior.

**Año - Duración:** Ene 2013 – Nov 2013.

### Objetivo

Rol de líder de proyecto en la implementación de un sistema para controlar la liquidación de títulos. El objetivo del sistema era realizar la liquidación de títulos de todas las áreas que conforman a la Casa de Bolsa como son Sociedades de Inversión de Capitales, Internacional, Arbitraje, Mercado de Dinero, Traspasos entre cuentas propias, Custodia. Debido al volumen de información, era necesario automatizar dichos procesos con ciertos controles internos para realizar la liquidación con del Depositario Central de Valores, además de contar con un Front End con estadísticas, reportes, información en tiempo real y mensajes de alerta para que el usuario pudiera dar seguimiento a cualquier eventualidad que se presentara.

### Características

1. Lenguaje de programación: Sistema Web desarrollado en J2EE, JSP's, JQuery
2. Frameworks: Struts, Spring, Mybatis, Jasper Reports, Sonic Enterprise Service Bus.
3. Modelo: Modelo, Vista, Controlador.
4. Servidor de Aplicaciones: Weblogic Server, Sun Solaris Iplanet.
5. Base de datos: Microsoft SQL Server.
6. Sistema Operativo: UNIX.

### Principales Actividades

1. Planeación e inicio.

Elaboración de plan de proyecto e identificación de usuarios interesados. Definición de roles y responsabilidades. Elaborar documentación de alcance, tiempo y recursos iniciales. Establecer plan de comunicaciones y seguimiento. Establecer criterios de éxito del proyecto.

2. Análisis del sistema.

Coordinar juntas de definición y alcance inicial con usuarios. Establecer criterios de aceptación de los requerimientos. Obtener aprobación de análisis y alcance del proyecto.

3. Diseño técnico.

Juntas de definición y diseño técnico con el área de arquitectura. Obtener aprobación de solución propuesta. Evaluar productos del mercado y soluciones internas.

4. Plan de pruebas.

Junta de definición y diseño de plan de pruebas funcionales y técnicas con área de calidad. Elaboración de plan de pruebas. Establecer criterios de calidad de volumen, tiempo de respuesta, usabilidad en el sistema de acuerdo a los requerimientos del usuario.

5. Coordinación de las actividades de desarrollo.

Coordinar a los integrantes del equipo de desarrollo para asegurar la ejecución de las tareas asignadas a cada miembro del equipo y cumplir con lo establecido en el plan de proyecto de acuerdo al alcance y tiempos. Asegurar que se desarrolle de acuerdo a los estándares establecidos.

6. Coordinación del equipo de trabajo.

Asegurar el cumplimiento de las tareas asignadas a cada recurso del proyecto así como dar seguimiento al progreso del proyecto de acuerdo a lo establecido en tiempo, recursos y alcance. Juntas de seguimiento con los proveedores para seguimiento de entregables.

7. Seguimiento de pruebas.

Coordinar y participar en las pruebas integrales con el usuario para garantizar la promoción de un producto de acuerdo a lo establecido en los requerimientos y los criterios de aceptación. Asegurar que los defectos encontrados fueran atendidos y cerrados de acuerdo a la criticidad dentro del proyecto.

8. Cierre del proyecto.

Elaborar documento de cierre del proyecto. Establecer los criterios de éxito del proyecto con base en las líneas base de tiempo, recursos y costo. Documentación de lecciones aprendidas y junta de cierre.

## **2.10 Proyecto N.9**

**Nombre:** Implementación de plataforma Global para Banca Patrimonial en una Casa de Bolsa

**Rol:** Líder de Proyecto.

**Recursos:** 2 líderes de proyecto.

**Año - Duración:** Nov. 2013 – Dic 2014.

### **Objetivo**

Realizar el análisis para la implementación de la plataforma global para Banca Patrimonial e inversiones de Mercado de Dinero, Mercado de Capitales, Sociedades de Inversión, Caución Bursátil y Derivados, Tesorería y Back Office de Contratos y Liquidaciones. Coordinar juntas con las áreas de la Casa de Bolsa y de Banca Patrimonial para realizar el GAP análisis de los requerimientos necesarios para adoptar la plataforma global. Adaptar la plataforma global al mercado mexicano para cumplir con las regulaciones locales establecidas por el Banco Central y la Secretaría de Hacienda. Promover las mejores prácticas del negocio de Inversiones Patrimoniales en México para que sean adoptadas en las regiones globales. Identificar los cambios en los sistemas locales.



## Principales Actividades

### 1. Análisis de la plataforma global.

Conocer de primera mano las características de la plataforma global. Entrenamiento de sus características de negocio y técnicas. Elaboración de listado de funciones e interfaces. Entrenamiento en la metodología utilizada. Revisión de listado de GAPS de las regiones de Brasil y EUA.

### 2. Análisis de negocio.

Coordinar juntas con las áreas de negocio de Mercado de Dinero, Mercado de Capitales, Sociedades de Inversión, Tesorería, Contratos y Liquidación de Valores y Efectivo para realizar sesiones de análisis con el fin de determinar las necesidades locales dentro de la plataforma global. Generar documentación de GAPS o diferencias entre las plataformas locales y globales, historias de usuario, requerimientos funcionales y no funcionales para las sesiones de análisis detallado o deep dive.

### 3. Estimación inicial.

Realizar estimación inicial de requerimientos funcionales y no funcionales. Identificar negocios afectados y aplicaciones afectadas. Coordinar junta de trabajo con líderes técnicos para revisión de impacto.

### 4. Diseño de solución técnica.

Realizar diagramas de solución propuesta para adoptar la plataforma global en los negocios de Banca Patrimonial para los sistemas de la Casa de Bolsa y del Banco.

### 5. Reportes de seguimiento.

Coordinar junta mensual de seguimiento con la Dirección de Tecnología y Sistemas. Junta semanal con el resto de los negocios afectados. Elaborar reporte de estatus y avance.

## 2.11 Proyecto N.10

**Nombre:** Implementación de la política de prácticas de Venta.

**Rol:** Líder de Proyecto

**Recursos:** 6 ingenieros en sistemas Junior, 2 Ingenieros en Sistemas Senior

**Año - Duración:** Nov 2013 – Jun 2013

### Objetivo

Rol de líder de proyecto en la implementación de la política de prácticas de venta establecida por la Comisión Nacional Bancaria y de Valores, CNBV por sus siglas. El objetivo es clasificar las operaciones de compra / venta de títulos de Mercado de Dinero, Mercado de Capitales, Sociedades de Inversión, Ejercicios de Derecho y Traspaso de Valores dentro de alguno de los servicios de inversión que ofrece la Casa de Bolsa, asesorado, ejecutado o mixto. Esto con la finalidad de poder reportar a las autoridades el origen de las inversiones de los clientes de la Casa de Bolsa.

## Características

1. Lenguaje de programación: Sistema Web desarrollado en J2EE, JSP's, JQuery
2. Frameworks: Struts, Spring, Mybatis, Jasper Reports, Sonic Enterprise Service Bus.
3. Modelo: Modelo, Vista, Controlador.
4. Servidor de Aplicaciones: Weblogic Server, Sun Solaris Iplanet.
5. Base de datos: Microsoft SQL Server.
6. Sistema Operativo: UNIX.

## Principales Actividades

1. Planeación e inicio.

Elaboración de plan de proyecto e identificación de usuarios interesados. Definición de roles y responsabilidades. Elaborar documentación de alcance, tiempo y recursos iniciales. Establecer plan de comunicaciones y seguimiento. Establecer criterios de éxito del proyecto.

2. Análisis del sistema.

Coordinar juntas de definición y alcance inicial con usuarios. Establecer criterios de aceptación de los requerimientos. Obtener aprobación de análisis y alcance del proyecto.

3. Diseño técnico.

Juntas de definición y diseño técnico con el área de arquitectura. Obtener aprobación de solución propuesta. Evaluar productos del mercado y soluciones internas.

4. Plan de pruebas.

Junta de definición y diseño de plan de pruebas funcionales y técnicas con área de calidad. Elaboración de plan de pruebas. Establecer criterios de calidad de volumen, tiempo de respuesta, usabilidad en el sistema de acuerdo a los requerimientos del usuario.

5. Coordinación de las actividades de desarrollo.

Coordinar a los integrantes del equipo de desarrollo para asegurar la ejecución de las tareas asignadas a cada miembro del equipo y cumplir con lo establecido en el plan de proyecto de acuerdo al alcance y tiempos. Asegurar que se desarrolle de acuerdo a los estándares establecidos.

6. Coordinación del equipo de trabajo.

Asegurar el cumplimiento de las tareas asignadas a cada recurso del proyecto así como dar seguimiento al progreso del proyecto de acuerdo a lo establecido en tiempo, recursos y alcance. Juntas de seguimiento con los proveedores para seguimiento de entregables.

7. Seguimiento de pruebas.

Coordinar y participar en las pruebas integrales con el usuario para garantizar la promoción de un producto de acuerdo a lo establecido en los requerimientos y los criterios de aceptación. Asegurar que los defectos encontrados fueran atendidos y cerrados de acuerdo a la criticidad dentro del proyecto.

8. Cierre del proyecto.

Elaborar documento de cierre del proyecto. Establecer los criterios de éxito del proyecto con base en las líneas base de tiempo, recursos y costo. Documentación de lecciones aprendidas y junta de cierre.

## CAPÍTULO 3. METODOLOGÍA

---



### 3 Metodología

El ciclo de vida de desarrollo de sistemas, o SDLC por sus siglas en inglés, es el marco de trabajo que define las actividades y pasos dentro del proceso de desarrollo de sistemas. El SDLC define de manera detallada los procesos y documentos para desarrollar, mantenimiento y reemplazo de software. También establece la mejora continua del proceso a través de lecciones aprendidas y en la calidad del software desarrollado.

Dentro de la Casa de Bolsa, se establecieron diversos enfoques para el desarrollo de software. Estos modelos se establecieron por el área de Software Management para cumplir con las necesidades de negocio y de acuerdo a las características de los proyectos se definen de la siguiente manera.

- Modelo Cascada o Waterfall.

Es el modelo más conocido y utilizado dentro de la metodología de desarrollo de Software. El modelo de cascada se llama así debido a que emplea un enfoque de “arriba hacia abajo”. Es una secuencia de etapas en las que la salida de cada etapa se convierte en la entrada para el siguiente. Estas etapas pueden ser caracterizados y se dividen en diferentes formas, incluyendo las siguientes fases: identificación del problema, análisis de requerimientos o necesidades de negocio, diseño de la solución, construcción o desarrollo, pruebas y aseguramiento de calidad, implementación y cierre.

La Ilustración V muestra la metodología cascada y la relación entre cada una de sus fases.



ILUSTRACIÓN V. FASES DEL SDLC

Las fases de la metodología en cascada se describe brevemente a continuación.

### 1. Identificar Problemas

En esta primera fase del ciclo de vida del desarrollo de sistemas, el objetivo es el de identificar problemas, oportunidades y objetivos. Esta etapa es crítica para el éxito del resto del proyecto, ya que de determina la factibilidad de resolver alguna necesidad de negocio a través del desarrollo de un producto de software o la adquisición de algún producto existente en el mercado que sea capaz de cumplir sus necesidades.

### 2. Definición de Requerimientos

En esta fase se definen los requerimientos de negocio, se identifican a los interesados o afectados por el proyecto, y se define el líder de proyecto para la iniciativa. Se realizan reuniones con los gerentes, las partes interesadas y los usuarios con el fin de determinar los requisitos y respondiendo a las siguientes preguntas.

- ¿Quién va a utilizar el sistema ?
- ¿Cómo van a utilizar el sistema ?
- ¿Qué datos deben ser introducidos en el sistema?
- ¿Qué datos debe ser emitida por el sistema?
- ¿Cuál es el volumen de datos que va a manejar el sistema?

Estas son preguntas generales que le sean respondidas durante una fase de recopilación de requisitos . Después de reunir estos requisitos se analizan requisito para su validez, y también se estudia la posibilidad de incorporar los requisitos del sistema para el desarrollo. Por último , se crea un documento de Especificación de Requerimientos que sirve al propósito de la directriz para la próxima fase del modelo.

### 3. Análisis de requerimientos.

Esta fase está enfocada completamente en la recopilación de información del sistema a desarrollar. En esta fase se consigue toda la información necesaria para el desarrollo del sistema. Una vez reunida la información referente al proyecto, esta información es analizada y plasmada en el documento de Especificación de Requerimientos de Software. Dentro de las especificaciones del sistema también se definen los criterios de aceptación de los requerimientos. Generalmente esta fase del SDLC requiere varias sesiones de definición entre los equipos de Tecnología y Sistemas y los usuarios del sistema. Como salida de esta fase, se actualizan los planes de trabajo con una estimación de costo, tiempo y recursos nivel 1 (+/- 20%) y se obtiene la aprobación del usuario con el alcance definido para el proyecto.

### 4. Diseño de la solución.

En esta fase del diseño del sistema y el software se prepara a partir de las especificaciones de requisitos que fueron estudiados en la fase de análisis. Dentro de esta fase, se detallan los requisitos de hardware y software del sistema para definir la arquitectura del mismo. Como parte de las actividades de esta fase, se realizan reuniones con el equipo de Arquitectura para establecer las tecnologías utilizadas para la implementación de la solución, ya que deben de cumplir con los estándares establecidos por Tecnología y Sistemas. Además se genera el documento de diseño arquitectónico y de interfaces entre los sistemas. También se pueden generar prototipos o pruebas de concepto para evaluar posibles alternativas de implementación. Por último, se detallan requerimientos no funcionales o técnicos para las especificaciones de procesos batch y de usabilidad de las pantallas requeridas para la solución

del Sistema. Como salida de esta fase se actualizan los planes de trabajo con una estimación de costos, tiempo y recursos en un nivel 2 (+/- 10%) y se obtiene la aprobación de la gerencia de Sistemas y del área de Arquitectura.

#### 5. Construcción o Desarrollo de la solución

En esta fase del SDLC se utilizan los documentos elaborados en la fase de análisis y diseño para realizar la codificación del sistema. En esta fase se desarrollan las interfaces, pantallas, procesos, bases de datos y programas para poder cumplir con los requerimientos del usuarios. El desarrollo debe apegarse a los estándares establecidos por el área de Tecnología y Sistemas. Dichos estándares definen la nomenclatura de funciones y clases, organización de paquetes y programas, frameworks o tecnologías permitidas, colores e imágenes permitidos. En esta fase se definen las pruebas técnicas para cada uno de los componentes, y se ejecutan las pruebas unitarias en un ambiente exclusivo del área de Desarrollo conocido como SIT por sus siglas en inglés, System Integration Testing. Una vez terminadas las pruebas unitarias, técnicas y funcionales, se realizan pruebas con el usuario para obtener el visto bueno y poder seguir con la siguiente fase del proyecto. También se define en esta fase la fecha de entrega al área de Calidad y la estimación final del proyecto con un nivel 3 (+/- 5%).

#### 6. Pruebas y Aseguramiento de la Calidad

En esta fase se realizan las pruebas de código por el área de Calidad en base a lo establecido como pruebas técnicas, unitarias, funcionales y criterios de aceptación. Como parte de las pruebas también se ejecutan validaciones de seguridad, volumen, usabilidad y regresión para garantizar que el producto esté listo para que el usuario pueda realizar pruebas integrales. Dichas pruebas se ejecutan en un ambiente controlado por la Gerencia de Calidad, ya que es un ambiente muy cercano al productivo. En caso de existir algún error durante las pruebas, el área de Calidad reporta los defectos encontrados dentro de la herramienta corporativa y se procede a atender y cerrar los errores. De igual manera, el área de Calidad revisa la documentación de todo el proceso para garantizar que se está cumpliendo con lo establecido en el SDLC. Una vez terminadas estas actividades, el área de Calidad documenta los pasos necesarios para instalar el producto en producción y se obtiene el visto bueno del usuario patrocinador de la iniciativa para poder proceder con la siguiente fase.

#### 7. Implementación

En esta fase del proceso, el área de Administración de Cambios o Change Management recibe el producto final por parte del equipo de Calidad con las instrucciones o pasos para instalar el software o sistema en un ambiente productivo. Esta instalación incluye todos los sistemas afectados, bases de datos, actualización a servidores o archivos de configuración necesarios. El área de Change Management se encarga de coordinar a los equipos de Operaciones para proceder con dichos cambios a través de Solicitudes de Cambio a Producción, Solicitud de cambio o Request for Change o RFC por sus siglas en inglés. Una vez finalizada la implementación en producción, el usuario patrocinador del proyecto, recibe una notificación para proceder con la certificación del software misma que debe aprobar para proceder con la siguiente fase del proyecto.

#### 8. Mantenimiento

Después de instalar un sistema, se le debe dar mantenimiento, es decir, los programas de cómputo tienen que ser modificados y actualizados cuando lo requieran. El mantenimiento se realiza por dos razones. La primera es la corrección de errores del software. No importa cuán exhaustivamente se pruebe el sistema, los errores se cuelan en los programas de cómputo. Los errores en el software comercial para PC se documentan como "anomalías conocidas", y se corrigen en el lanzamiento de nuevas versiones del software o en revisiones intermedias. En el software hecho a la medida, los errores se deben corregir en el momento que se



detectan. La otra razón para el mantenimiento del sistema es la mejora de las capacidades del software en respuesta a las cambiantes necesidades de una organización. En síntesis, el mantenimiento es un proceso continuo durante el ciclo de vida de un sistema de información. De cualquier manera que el sistema requiera algún mantenimiento o modificación, éstos se atienden dentro de alguna iniciativa que cumpla con el proceso SDLC.

El otro modelo del SDLC utilizado dentro la Casa de Bolsa es la metodología ágil. Las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno. En la Casa de Bolsa, a inicios del 2013 se eligieron varios proyectos como pilotos para ejecutar el desarrollo de Software bajo la metodología Scrum. Los procesos definidos para esta metodología cumplen con los estándares establecidos por el área de Software Management

- Modelo ágil o Scrum.

La metodología ágil adoptada por la Casa de Bolsa elegida para el desarrollo de Software es Scrum. El concepto de Scrum tiene su origen en un estudio de 1986 sobre los nuevos procesos de desarrollo utilizados en productos exitosos en Japón y los Estados Unidos. Los equipos que desarrollaron estos productos partían de requisitos muy generales, así como novedosos, y debían salir al mercado en mucho menos del tiempo del que se tardó en lanzar productos anteriores. Estos equipos seguían patrones de ejecución de proyecto muy similares. En 1993 se realizó el primer Scrum para desarrollo de software y en 1995 el proceso fue formalizado. En 2001 un grupo de personas muy relevantes en lo que empezaba a ser el desarrollo ágil escribieron los valores fundamentales de los procesos ágiles bajo un manifiesto conocido como el “Manifiesto Ágil”.

La Ilustración VI muestra el proceso ágil o Scrum para la metodología de desarrollo de Sistemas.

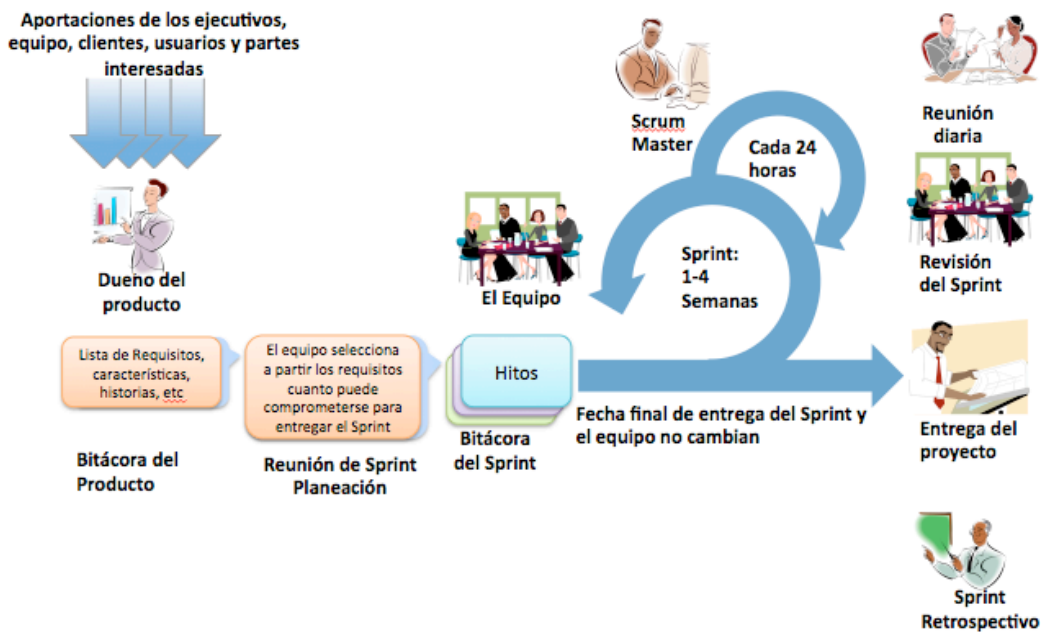


ILUSTRACIÓN VI. PROCESO SCRUM

Las fases de la metodología ágil se describe brevemente a continuación.

### Junta de product Backlog

El product backlog es una lista priorizada de requerimientos generada por el product owner o usuario patrocinador del proyecto. La junta de product backlog es una junta donde se reúne el scrum master, el scrum team y el product owner para determinar la complejidad de cada uno de los requerimientos del product backlog, la velocidad del equipo para atender los requerimientos en el marco de un sprint, el número total de sprints y el plan de liberaciones.

### Junta de Inicio de Sprint

En esta junta se toman los requerimientos que van a ser atendidos en el sprint, y se realiza el análisis y diseño arquitectónico de cada uno de ellos. También para cada requerimiento se enlistan una serie de actividades para completarlo y un estimado en horas para cada actividad. También se determina la fecha de la demostración frente al usuario y la hora y lugar donde se llevará a cabo la junta de seguimiento diaria.

### Junta Diaria

Es una junta de seguimiento del sprint que se ejecuta de forma diaria. Se realiza con el scrum team, el scrum master y de ser necesario el product owner. Cada miembro del scrum team responde a las siguientes preguntas.

- ¿Qué hice el día de ayer?
- ¿Qué voy a hacer el día de hoy?
- ¿Existe algún riesgo o impedimento para mis actividades?

La idea de la junta es que el Scrum master actualice el tablero de las actividades clasificándolas en pendientes, en proceso y finalizadas. No existe una asignación específica de tareas, más bien la metodología fomenta la auto organización del equipo, y cada miembro una vez que termine una tarea debe tomar alguna otra tarea que esté pendiente.

La Ilustración VII muestra un ejemplo de un tablero Scrum que se utiliza para el seguimiento de actividades diarias y control del Sprint.

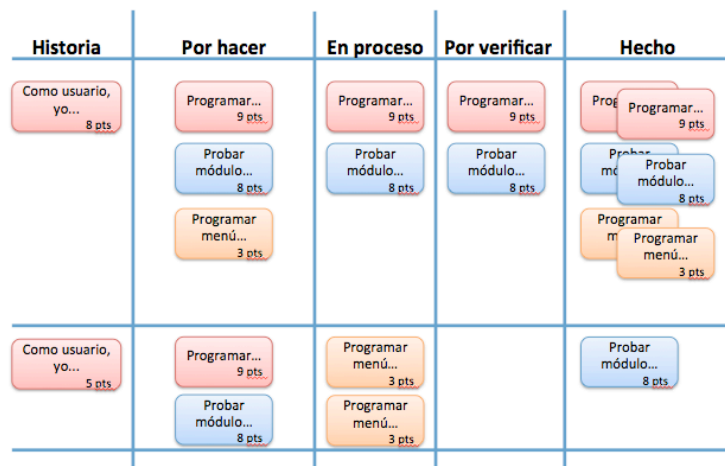


ILUSTRACIÓN VII. PROCESO SCRUM.

La ilustración VIII muestra la gráfica de burndown de las actividades durante el sprint. El objetivo es mostrar de manera ilustrativa el desempeño del equipo durante el sprint e identificar posibles desviaciones para tomar decisiones.

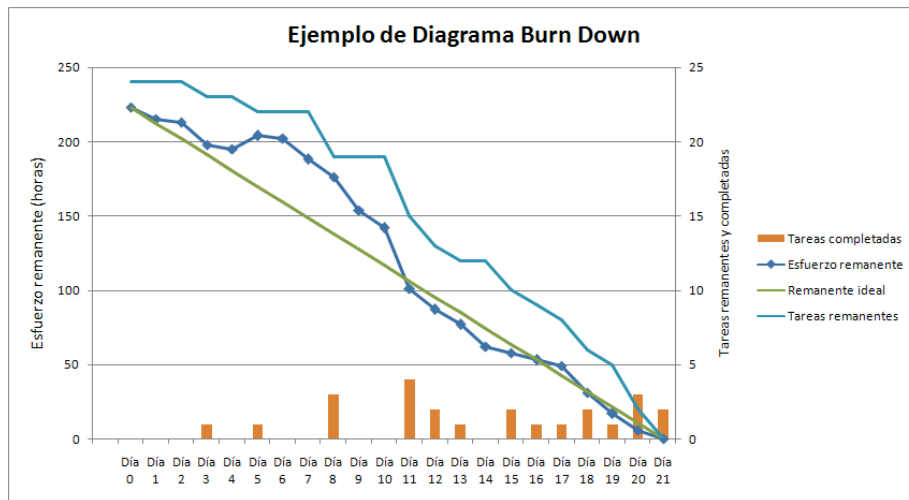


ILUSTRACIÓN VIII. GRÁFICA BURNDOWN.

#### Demostración del producto

Esta junta se realiza al final del sprint una vez que se finalizaron todos los requerimientos. En esta reunión se presenta el producto instalado en un ambiente de pruebas o UAT con los logros que realizó el equipo durante el sprint. El product owner determina si el producto demostrado es candidato para liberarse a producción o requiere que se ajuste en el siguiente sprint.

#### Junta de retrospectiva

Un sprint se cierra con la junta de retrospectiva, donde el Scrum Master y el Scrum team, responden a las siguientes preguntas para generar la documentación de las lecciones aprendidas.

- ¿Qué hicimos bien como equipo?
- ¿Qué podemos mejorar como equipo?
- ¿Qué se hizo mal?
- ¿Qué podemos intentar más?

#### Clasificación de proyectos

Dentro de la Casa de Bolsa los proyectos pueden clasificarse de acuerdo al esfuerzo de recursos requerido. Las clasificaciones de los proyectos son las siguientes.

- Enhancement o mejora.

Un enhancement o mejora, es un cambio de producto o actualización que aumenta las capacidades del software o hardware. Las mejoras permiten escalabilidad de software y hardware de rendimiento del producto. Este tipo de actividad requiere un esfuerzo menor a 80 hrs de trabajo. Por su naturaleza este tipo de trabajo sólo puede ser implementado por metodología en cascada.

- Actividad.

Al igual que una mejora, una actividad es un cambio de producto o actualización que aumenta las capacidades del software o hardware. La diferencia es que este tipo de actividad requiere un esfuerzo mayor a 80 hrs de trabajo y menor a 600 hrs de trabajo. Por su naturaleza este tipo de trabajo sólo puede ser implementado por metodología en cascada.

- Proyecto Menor

Un proyecto menor es un tipo de trabajo de software para la implementación de nuevos productos o sistemas. Este tipo de trabajo requiere un esfuerzo mayor a 600 hrs y menor a 1500 hrs. Por sus características, este tipo de trabajo puede ser implementado bajo metodología en cascado o ágil.

- Proyecto Mayor

Un proyecto mayor es un tipo de trabajo de software para la implementación de nuevos productos o sistemas. Este tipo de trabajo requiere un esfuerzo mayor a 1500 hrs. Por sus características, este tipo de trabajo puede ser implementado bajo metodología en cascado o ágil.

El proyecto descrito en este informe profesional por sus características de esfuerzo requerido, era un proyecto mayor con un esfuerzo mayor a 1500 horas. Debido a los requerimientos del usuario y de negocio, se adoptó una metodología ágil para cumplir con las expectativas de negocio.

### **3.1 Descripción del Proyecto**

A continuación se describe el proyecto considerando sus antecedentes, el problema, requerimientos, evaluación de soluciones y los pasos a seguir dentro del proceso de administración de proyectos.

### **3.2 Desempeño Actual y el Deseado**

Previo al desarrollo de la implementación del sistema de liquidación de títulos, el proceso para realizar esta actividad dentro de la Casa de Bolsa se realizaba por un proceso manual a través de dos aplicaciones que formaban parte del Back Office de Valores. El usuario ingresaba a una de las aplicaciones para obtener un reporte con las operaciones del día, y posteriormente importaba dichas operaciones en el portal del Depositario Central de Valores para ingresar las operaciones para su liquidación. Dicho proceso implicaba riesgos operativos en el mismo, como la posibilidad de errores en la importación o exportación de archivos, duplicar información, modificar datos, así como de recursos dedicado a dichas actividades. Además, el uso del sistema del Depositario Central de Valores conllevaba tarifas de acuerdo al número de operaciones importadas en el mismo.

Por lo anterior, fue necesario plantear un sistema para mejorar la operación del área, considerando la disminución de costos, el volumen operativo, los riesgos transaccionales, así como la

automatización de la mayor parte del proceso. De igual forma, se estableció como objetivo la migración tecnológica de varios de los componentes del sistema, utilizando tecnología de punta.

### **3.3 Definición del Problema**

Parte de los problemas que se tienen se encuentran los siguientes.

- Falta de interfaces entre los sistemas de negocio y el área de Back Office de Valores.
- Falta de interfaz de usuario para monitoreo de operaciones en línea.
- Falta de protocolo estándar de comunicación con el Depositario Central de Valores.
- Falta de validación en el sistema para evitar duplicidad de operaciones.
- Falta de controles para evitar modificación de información.
- Falta de un diseño arquitectónico estandarizado en la tecnología.
- Costos elevados en el proceso de liquidación de Valores.
- Proceso manual operativamente lento.
- Falta de información histórica para auditorías de negocio o tecnológicas.
- Atención prolongada para liberación de nuevas implementaciones.
- Riesgos operativos.
- Falta de repositorio central de código.
- Falta de desarrolladores con experiencia en negocio.

### **3.4 Requerimientos**

Para la recopilación de requerimientos de acuerdo a la metodología ágil establecida dentro de la Casa de Bolsa, se utilizó la técnica de Junta de Product Backlog. Dentro de esta junta participaron integrantes de diversas áreas de Tecnología y Sistemas y de negocio. Los participantes de esta junta se enlistan a continuación.

- Scrum Master.

Su función o rol es la de velar por que todos los participantes del proyecto sigan los valores y principios ágiles, las reglas y proceso ágil, fomentar la colaboración dentro del equipo y con el usuario. Dentro de la junta de Product Backlog es el moderador entre el equipo Scrum y el usuario, asegurándose que se comprendan los requerimientos o historias de usuario, que se prioricen, que se determine la velocidad del equipo y el número de Sprints requeridos para atender el proyecto.

- Scrum Team.

Grupo de personas que de manera conjunta desarrollan el producto del proyecto. Tienen un objetivo común, comparten la responsabilidad del trabajo que realizan en cada iteración y en el proyecto. Este equipo de trabajo se encarga de determinar una complejidad de cada requerimiento o historia de usuario de acuerdo a sus propios criterios. En caso de que una historia de usuario resulte muy compleja, el equipo puede proponer partirla en requerimientos menos complejos. El equipo está conformado por 1 arquitecto del Sistema, 1 tester, 2 desarrolladores de Sistemas, 1 administrador de proyectos, 1 representante de Operaciones y Change Management, 1 usuario de negocio y 1 líder de proyecto.

- Product Owner.

Es el representante de todas las personas interesadas en los resultados del proyecto y actuar como interlocutor único ante el equipo, con autoridad para tomar decisiones. Es el propietario de la planificación del proyecto. Define la lista de requerimientos y genera las historias de usuario, también define la prioridad de cada uno de los requerimientos. En la junta de producto backlog, se encarga de explicar al Scrum Team los requerimientos para que el equipo pueda estimar la complejidad de cada uno de ellos. También da su visto bueno al documento de product backlog como salida de esta junta.

Los requerimientos de alto nivel que resultaron de la junta de Product Backlog se presentan a continuación.

- Automatizar liquidación de operaciones de NAFIN, Recompra, Posición Propia, Sociedades de Inversión, Internacional, Traspaso de Valores, Mercado de Dinero, Custodia Externa, Préstamo de Valores.
- Enviar por mensajería y un protocolo estándar información de traspasos de títulos con INDEVAL.
- Recibir por mensajería y un protocolo estándar confirmación de operaciones de INDEVAL.
- Crear los mantenimientos del sistema para configurar los parámetros de la liquidación de las contrapartes y las operaciones del sistema.
- Implementar un módulo del sistema para configurar el envío automático de las operaciones de las diferentes áreas de negocio de la Casa de Bolsa.
- Realizar un dashboard con un semáforo de estatus de operaciones para consultar la liquidación en tiempo real con INDEVAL.
- Guardar información histórica para consultar hasta 10 años de información en línea.
- Realizar una pantalla para consultar estadísticas agrupados por área de negocio, por número de operaciones, cantidad de títulos, monto total a liquidar, cancelaciones.
- Crear los reportes de cierre de día, cancelaciones, operaciones programadas, conciliaciones del área.
- Agregar notificaciones y alertas para eventos del área.
- Guardar bitácora de cancelaciones y envío de operaciones.
- Crear los perfiles de seguridad para los distintos roles de los usuarios en el sistema.
- Realizar la migración tecnológica de los componentes del sistema para cumplir con los estándares de Tecnología y Sistemas.

### 3.5 Evaluación de Soluciones

#### 3.5.1 Interfaz de usuario

Como parte de los requerimientos de negocio, fue necesario evaluar alternativas para la interfaz de usuario. De manera tradicional, las aplicaciones Web desarrolladas funcionaban mediante peticiones HTTP desde la máquina del usuario hacia el servidor, provocando un gran tráfico en la Intranet, así como sobrecargas y tiempos de respuesta que debido a las características del negocio resultaban inaceptables.

Las aplicaciones RIA o Rich Internet Applications nacieron como una combinación de las características de las aplicaciones Web y de Escritorio. Estas soluciones ofrecen una interacción que mejora la experiencia del usuario y así mejorar la productividad. Las interfaces resultan más intuitivas y flexibles para que esté al alcance del cliente.

La ilustración IX muestra las características de las aplicaciones RIA.



ILUSTRACIÓN IX. APLICACIONES RIA.

Dentro de la gama de productos evaluados se encontraban diversas tecnologías. En la Casa de Bolsa anteriormente se habían utilizado tecnologías propietarias como Adobe Flex, sin embargo, debido a las nuevas tendencias marcadas por la llegada de HTML 5 y JQuery, y las ventajas de desarrollar en Software Libre, se decidió utilizar JQuery como tecnología para el desarrollo del Front End. Esto respaldado por la información del mercado.

La figura X muestra la evolución del interés generado alrededor de tecnologías RIA y de JQuery comparado con algunas otras tecnologías como Flex, Google Web Toolkit o ExtJS.



ILUSTRACIÓN X. JQUERY GOOGLE TRENDS.

Otro factor fundamental en la elección fue el anuncio de Adobe de detener el soporte a Flex Framework a partir del año 2014, costos de las licencias, costo de mantenimiento encontrando personal especializado en esta tecnología y no poder modificar los componentes base para la personalización de widgets.

### 3.5.2 Características de la solución

Estas son una lista de características de la solución para el Front End.

- Experiencia rica del usuario.
- Controles estilo escritorio como copiar, pegar, cambiar columnas, ordenar información.
- Controles estilo Drag & Drop.



- Respuesta inmediata para el usuario.
- Capacidad Offline de aplicaciones.
- Interfaz flexible con la posibilidad de modificar apariencia, contenido y servicios disponibles.
- Desarrollo rápido de aplicaciones.
- Menor cantidad de peticiones al servidor de aplicaciones.
- Separación de la vista y estilos y de la lógica de los controles.
- No requiere instalación de ningún plug in adicional.
- Soportado por la mayoría de los navegadores modernos.
- Versatilidad ya que la ejecución es en tiempo real.

Al utilizar tecnologías como JQuery, se redujo la curva de aprendizaje al ser un lenguaje de programación abierto, y con similitudes con Javascript. Además, se tomó ventaja de la integración de dicho lenguaje con otros frameworks como Struts y Spring y de un protocolo de comunicación estándar para aplicaciones escritas en JQuery como lo es JSON.

### **3.5.3 JQuery**

JQuery es una librería JavaScript de código libre, que funciona en múltiples navegadores, y que es compatible con CSS3. Su objetivo principal es hacer la programación “scripting” mucho más fácil y rápida del lado del cliente. Con jQuery se pueden producir páginas dinámicas así como animaciones parecidas a Flash en relativamente corto tiempo.

La siguiente es una lista de características de JQuery

- Independiente del código HTML y CSS, permitiendo un mantenimiento más sencillo del código y aumentar la productividad del equipo de desarrollo.
- Independencia del navegador sobre el que se visualiza.
- Existencia de plug ins en el mercado para distintas necesidades. Gráficos, controles, tablas, alertas, etc.
- Curva de aprendizaje corta.
- Grado de penetración en el mercado muy alto
- Protocolo de comunicación estándar con JSON

De acuerdo a los análisis de las tendencias del mercado, para el año 2013 el grado de penetración y usabilidad de esta tecnología se encontraba en pleno auge para la mayor parte de los desarrollos de aplicaciones Web como se muestra en la ilustración XI.

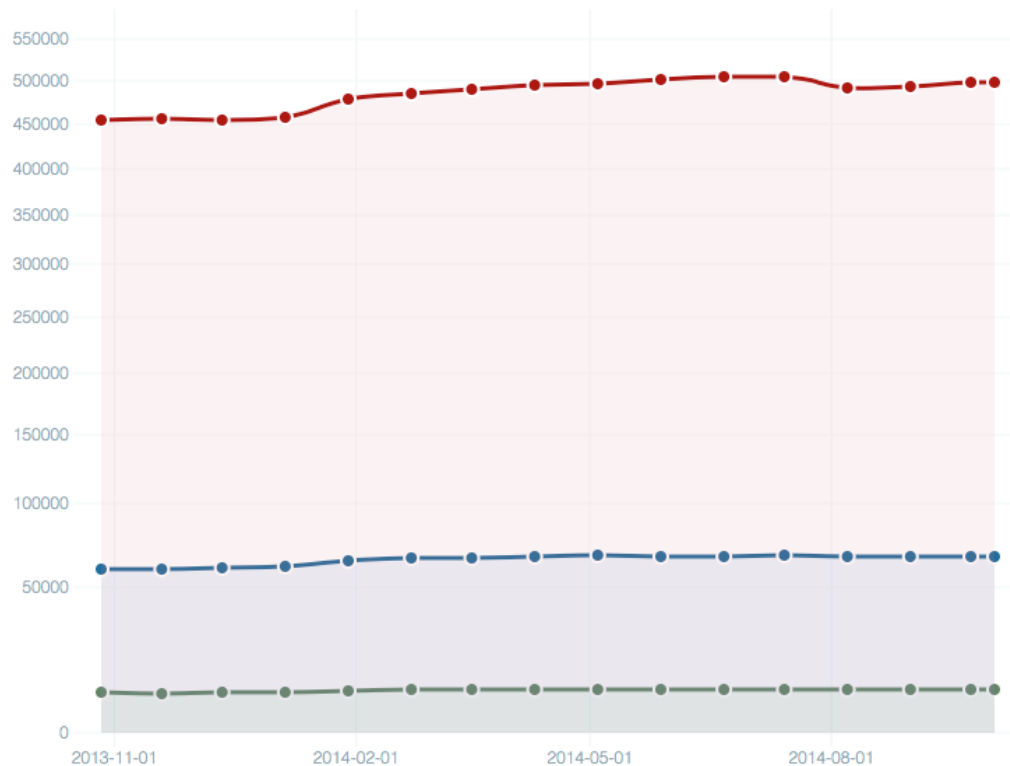


ILUSTRACIÓN XI. UTILIZACIÓN DE JQUERY A NIVEL MUNDIAL.

### 3.5.4 Tecnologías para el servidor de aplicaciones.

Como parte de la solución propuesta, se evaluaron diversas alternativas para el servidor de aplicaciones. Debido a la gran experiencia y conocimiento sobre tecnologías desarrolladas en Java, y el gran soporte de los lineamientos del área de Arquitectura y Tecnología y Sistemas, se decidió utilizar los siguientes Frameworks de desarrollo de aplicaciones Java.

- Struts

Struts es un framework de aplicación web de código abierto desarrollado por Apache. Struts está basado en el patrón MVC. Se utiliza para construir aplicaciones Web basadas en Servlets y JSP que pueden ejecutarse en cualquier contenedor de servlets incluyendo los servidores de aplicaciones J2EE. Mediante Struts se hace uso de patrones de diseño J2EE que son útiles en la construcción de aplicaciones J2EE. Y puesto que Struts sigue los patrones incluidos en el MVC, este framework es relativamente simple de usar y aprender. Struts proporciona al desarrollador un conjunto de etiquetas JSP personalizadas que facilitan la integración del framework con las páginas JSP.

Las principales características de Struts son las siguientes.

- Permite modelar las aplicaciones basadas en JSP y Servlets mediante el uso del patrón de diseño MVC
- Configuración de controladores a través de archivos XML.
- Incorpora características extras como internacionalización, validación, integración con Jasper Reports, Spring framework y JSON.
- Transporte automático de los datos introducidos en el cliente (JSP) hasta el controlador (Action) mediante formularios (ActionForm).
- Transporte automático de los datos enviados por el controlador (Action) a la parte de presentación (JSP) mediante formularios (ActionForm).
- La separación de los componentes en capas (MVC) simplifica notablemente el desarrollo y su mantenimiento.

La figura XII muestra la arquitectura general de una aplicación basada en Struts.

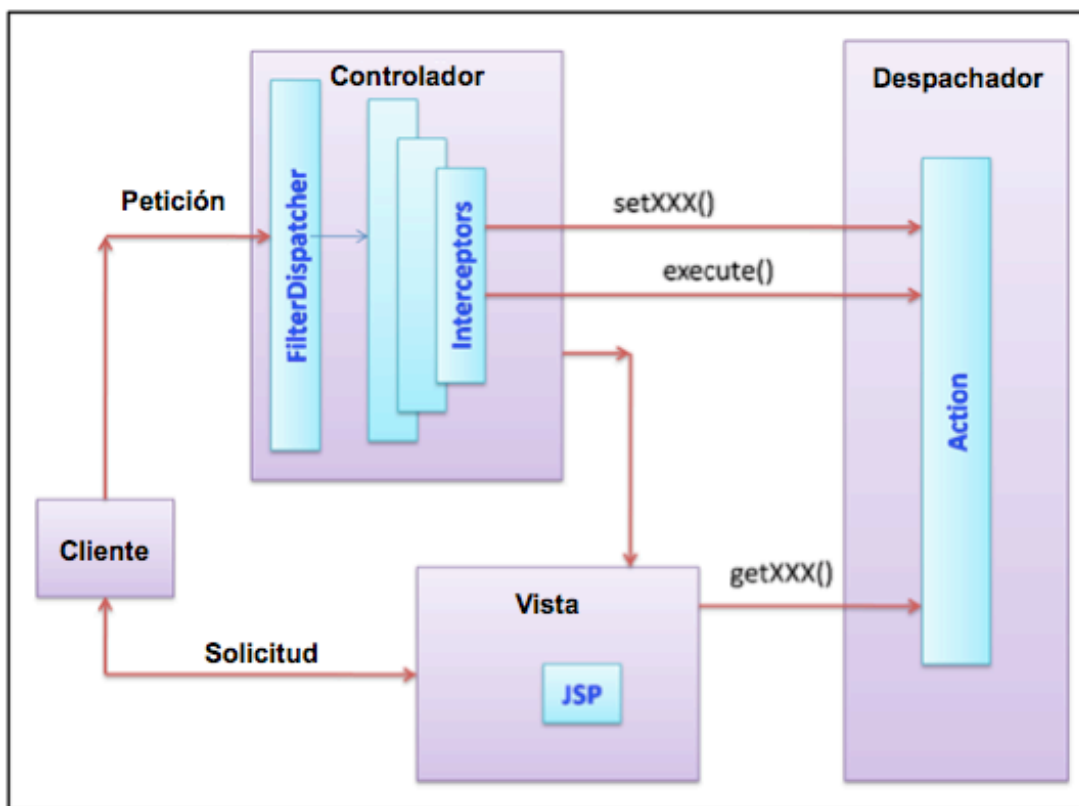


ILUSTRACIÓN XII. ARQUITECTURA STRUTS 2.

- Spring

Spring es un framework de código abierto que proporciona un marco de trabajo para el desarrollo de aplicaciones J2EE. El framework está basado en el uso de archivos planos JavaBeans para la lógica de aplicación y archivos XML para la configuración. Spring es un framework liviano y no intrusivo, es decir, los objetos que se programan no tienen dependencias en clases específicas de Spring. Sus características principales son inyección de dependencias y programación orientada a aspectos. El objetivo es lograr un bajo acoplamiento entre los objetos de nuestra aplicación. Con este patrón de diseño, los objetos no crean o buscan sus dependencias (objetos con los cuales colabora) sino que éstas son dadas al objeto. El contenedor (la entidad que coordina cada objeto en el sistema) es el encargado de realizar este trabajo al momento de instanciar el objeto. Se invierte la responsabilidad en cuanto a la manera en que un objeto obtiene la referencia a otro objeto.

De esta manera, los objetos conocen sus dependencias por su interfaz. Así la dependencia puede ser intercambiada por distintas implementaciones a través del contenedor. En resumen, programaremos orientado a interfaces e inyectaremos las implementaciones a través del contenedor.

Las principales características de Spring son las siguientes.

- No es intrusivo con otras tecnologías para integrarse fácilmente.
- Configuración mediante archivos XML.
- Programación mediante la API y mediante un estándar JSR.
- Uso de Interfaces.
- Patrón de diseño Singleton, Inversión de Control, Modelo Vista Controlador.

La ilustración XIII muestra la integración del framework de spring con struts.

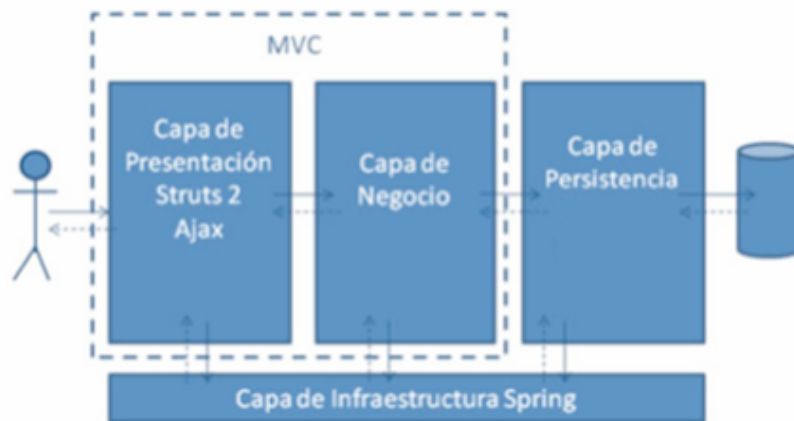


ILUSTRACIÓN XIII. SPRING FRAMEWORK.

Mybatis

Mybatis es una herramienta de persistencia de Java que se encarga de mapear sentencias SQL y procedimientos almacenados con objetos a partir de archivos XML o anotaciones. MyBatis es software libre y se ha desarrollado bajo la licencia de Apache.

Dentro de sus principales características se encuentra las siguientes.

- Permite utilizar todas las funcionalidades de la base de datos como procedimientos almacenados, vistas, consultas de cualquier complejidad o funcionalidades específicas del proveedor.
- Es una herramienta indicada para bases de datos legadas.
- Simplifica la programación frente al uso directo de JDBC
- Proporciona un motor de mapeo de resultados SQL.
- Soporta la composición de sentencias SQL.
- Soporta integración con Spring Framework .
- Soporta el uso de caché declarativa.

#### Integración con INDEVAL

Para la comunicación con INDEVAL, se decidió utilizar mensajería vía JMS a través de la Red Financiera en la cual se encuentran las instituciones participantes en el mercado mexicano. Para la comunicación de mensajes financieros, se utilizó un API propietaria del INDEVAL capaz de generar mensajes en formato SWIFT, por sus siglas en inglés, Society for Worldwide Interbank Financial Telecommunication, el cuál define las reglas y características para los diferentes mensajes entre instituciones financieras.

Para garantizar el correcto envío de información entre la Casa de Bolsa y el INDEVAL, se determinó utilizar un servidor de mensajería dentro de un Enterprise Service Bus. El principal objetivo de este componente dentro del sistema, era el de recibir los mensajes desde el aplicativo, transformarlos en mensajes SWIFT estándar y enviarlos vía JMS a INDEVAL. Esto con el fin de garantizar cualquier caída en el sistema, fuera capaz de recuperar los mensajes pendientes y reenviarlos al INDEVAL.

#### Otros componentes del sistema

A continuación se lista otra serie de componentes utilizados en el sistema como parte de la solución diseñada.

- Log4j  
Utilizado para el logeo de información dentro del sistema. Se utilizó de tal manera que podía establecerse el nivel de logeo de acuerdo a las necesidades, por ejemplo, en caso de algún error, subir el nivel de logeo a DEBUG para determinar todas las entradas y salidas de los diferentes componentes del sistema.
- Itext y JasperReports

Utilizados para desarrollar los componentes de reportes del sistema. Mediante estas tecnologías se generan templates para los reportes los cuales pueden tener una salida en formato XLS, CSV, Word o PDF de manera más sencilla.

- Quartz Scheduler

Se utilizó para que el usuario fuera capaz de calendarizar procesos de forma automática, estableciendo parámetros como día, hora, intervalos de ejecución. Esto con la finalidad de evitar tareas repetitivas y automatizar los procesos operativos de inicio y cierre de día. También se programaron alertas en caso de falla y notificaciones a buzones de negocio.

- Maven

Se utilizó maven para automatizar tareas de generación de código y compilación así como para gestión de dependencias. Debido a que el sistema se desarrolló con un enfoque incremental, contar con una herramienta que automatizara estas actividades era crítico para el éxito del mismo.

- Team Foundation Server

Se utilizó TFS por sus siglas en inglés, como plataforma de colaboración y administración del proyecto ágil. En dicha plataforma se llevó a cabo el seguimiento de las actividades de las distintas áreas, actualización de la bitácora de requerimientos, entrega de código, control de versiones y generación de builds mediante la integración con script de Maven.

### **3.6 Plan de Trabajo**

El primer paso dentro de la metodología ágil establece la generación del Product Backlog y realizar la junta de Product Backlog.

En esta junta se reúne el Scrum Team, Scrum Master y Product Owner para analizar cada uno de los requerimientos. También el Scrum Team, como parte de sus responsabilidades, se encarga de determinar la complejidad de cada uno de los requerimientos con una metodología de poker planning, donde cada integrante de acuerdo a su experiencia, conocimientos y especialidad, asigna una complejidad. El objetivo es que el Scrum team se ponga de acuerdo en esta complejidad para cada uno de los requerimientos. La puntuación de complejidad se le conoce en la metodología ágil como Story Points. El siguiente paso del proceso, es determinar cuántos Story Points puede terminar el Scrum Team en un periodo de un sprint, que son 4 semanas o 20 días de trabajo continuo. Una vez determinado el número de story points por sprint, o conocido en el lenguaje ágil como velocidad del equipo, se determina el número de sprints en el cual se terminara de implementar todos los requerimientos.

La figura XIV ilustra el proceso ágil dentro de la Casa de Bolsa

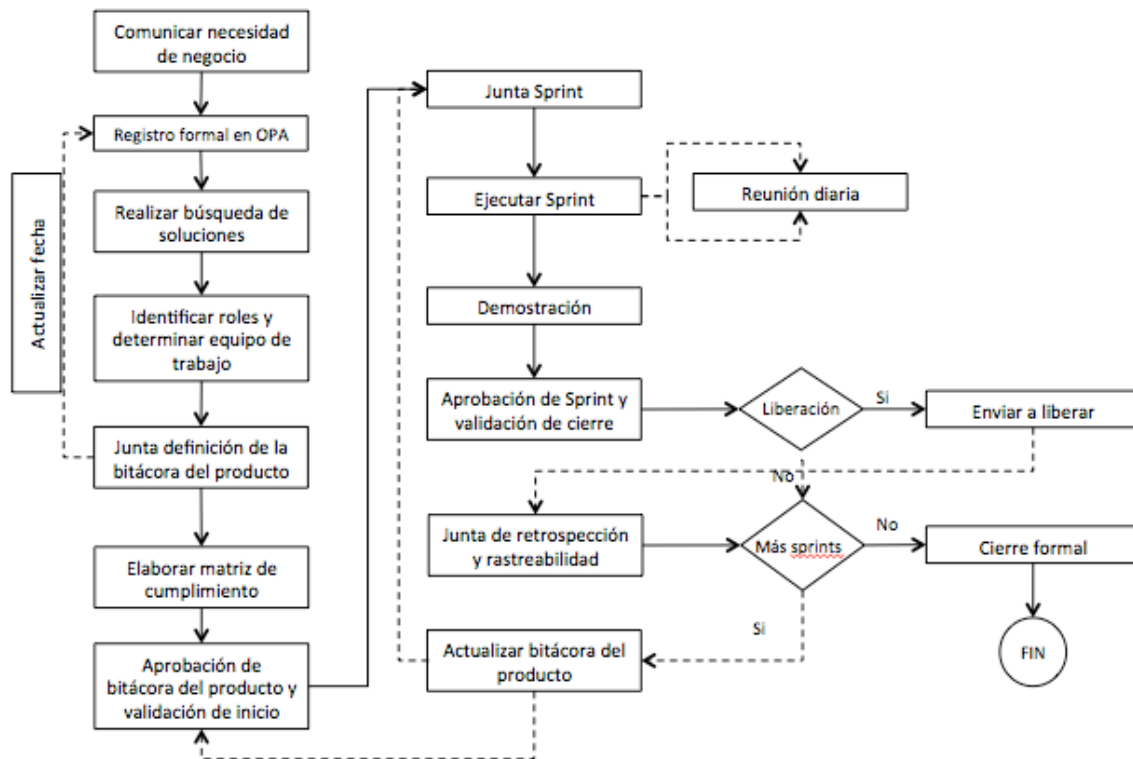


ILUSTRACIÓN XIV. METODOLOGÍA ÁGIL DEL PROYECTO SLV

Una vez determinado el número de sprints totales en el proyecto, se determina cuáles sprints implican una liberación o release a producción. Este plan se plasma en el documento de release planning. De acuerdo a la metodología ágil de la Casa de Bolsa, dentro de un sprint, se realizan de forma iterativa en periodos de 20 días, el análisis, diseño, construcción, pruebas y liberación de los requerimientos atendidos en el sprint.

Una vez finalizada la junta de product backlog, se genera la documentación de planeación con la siguiente información en el documento de plan de trabajo.

- Estimación de costos nivel en base a los recursos y fechas de cada sprint.
- Número total de sprints con su fecha inicial y final
- Fecha de cierre de proyecto.
- Plan de liberaciones.
- Riesgos del proyecto.
- Matriz de roles y responsabilidades.

Este plan de trabajo es aprobado por el product owner y las gerencias de negocio y de Tecnología para dar formalizado el inicio del proyecto.

### 3.7 Planeación de liberaciones





Participante	Tipo		Nivel de poder	Nivel de interés
	Interno	Externo		
Project Sponsor	X		Alto	Alto
Product Owner	X		Alto	Alto
Gerencia de Desarrollo	X		Alto	Alto
Scrum Master	X		Medio	Alto
Ing. en sistemas senior	X		Medio	Alto
Ing. Sistemas Junior		X	Bajo	Alto
Tester de Calidad	X		Bajo	Alto
Arquitecto de Sistemas	X		Alto	Bajo
Oficina de Proyectos	X		Medio	Medio
Ing. del área de seguridad.	X		Alto	Bajo
Gerencia de Calidad.	X		Medio	Medio
Ing. del área de liberación	X		Bajo	Bajo

### 3.7.2 Supuestos

Los siguientes son los supuestos que se consideraron en el desarrollo de este sistema de liquidación de valores para los usuarios de back office de valores en la casa de bolsa, que involucran las áreas de desarrollo, calidad y liberación.

- El Product Owner proveerá definiciones completas de sus requerimientos en una lista priorizada.
- El Product Owner determinará los criterios de aceptación de los requerimientos.
- Disponibilidad del Product Owner para asistir a las juntas de Product Backlog, Inicio de Sprint, Daily Sprint, Demostración, Restrospectiva y Cierre del proyecto.
- Compromiso del Product Owner y del equipo de trabajo para defender los intereses del proyecto.
- Patrocinio por parte del área de negocio para los recursos humanos y técnicos requeridos en el proyecto.
- Asignación de gente de Tecnología y Sistemas con conocimiento de las tecnologías utilizadas.
- No se considera dentro del alcance ningún sistema que no esté nombrado en el documento de bitácora de requerimientos.
- No se aceptarán cambios durante la ejecución del sprint.

- Se podrán agregar requerimientos en la junta de inicio de sprint siempre y cuando el equipo sea capaz de atenderlos.
- El Product Owner entregará scripts para la ejecución de pruebas unitarias, de integración y de funcionalidad.
- El Product Owner actualizará el documento del manual de usuario del sistema.
- El área de desarrollo será quien se encargue de desarrollar y coordinar la implementación del producto.
- El área de calidad esta considerada para la realización de pruebas, validaciones y puntos de calidad.
- El área de operación está considerada para apoyar en la implementación y manejo de base de datos, instalación y configuración de los servidores.
- El área de Change Management es responsable del proceso de liberación y puesta en marcha del aplicativo.
- El Scrum Master se desempeñará como facilitador del proyecto para resolver cualquier riesgo o problema que se presente durante la ejecución del proyecto.
- El Scrum Team y Product Owner pueden determinar la cancelación de un sprint o iteración en caso de que no sea factible alcanzar a terminar los requerimientos o exista un problema que impida lograr los objetivos.
- El Scrum Team y Product Owner se comprometen a reunirse para las juntas de Product Backlog.

### **3.7.3 Riesgos**

Es necesario especificar los riesgos en cada una de las fases del proyecto, cuando se inició el desarrollo de este proyecto, se especificaron los siguientes riesgos:

- Contar con definiciones incompletas y/o incorrectas de los requerimientos del usuario.
- No cumplir con los requerimientos técnicos iniciales.
- No tener disponibles recursos para el proyecto.
- Contar con desviaciones en el proyecto por falta de compromiso en captura de actividades
- Contar con cambio de prioridades para los recursos asignados al proyecto.
- Contar con cambio de prioridades para el desarrollo del proyecto que genere sobrecostos y/o retrasos.
- Pobre coordinación entre los tiempos del área de desarrollo, calidad y reléase.
- Retraso en cierre de actividades por falta de captura en el área de calidad.
- Contar con retrasos en tiempos por falta de pruebas en el área de calidad.
- Contar con retrasos en liberación por tiempos del área de reléase.
- Incumplimiento con las fechas compromiso.

### **3.8 Sprint planning**

La junta de sprint planning se ejecuta una vez aprobada la bitácora de requerimientos. En esta junta se vuelven a reunir el Scrum Team, Scrum Master y product owner con el fin de desglosar las actividades necesarias para completar los requerimientos que se van a atender durante el sprint. Durante esta junta se determina la arquitectura o modelos arquitectónicos para cada uno de los requerimientos, las especificaciones técnicas del sistema, las especificaciones funcionales del sistema, funcionalidad de las interfaces de usuario, entradas y salidas, criterios para las pruebas como datos o información necesaria, sistemas impactados y pruebas funcionales.

Esta información se integra en un documento llamado Sprint Backlog, el cual contiene la siguiente información.

- Requerimientos que forman parte del sprint.
- Actividades de cada requerimiento.
- Número de horas para completar las actividades.
- Modelo arquitectónico.
- Especificaciones para las pruebas.
- Especificaciones de interfaz de usuario.
- Requerimientos de volumen y stress.
- Dependencias con otros sistemas.

Una vez finalizada la junta de Sprint Backlog, se actualiza la documentación del plan de proyecto con las fechas del sprint, estimado de costos y riesgos. También se genera el plan de comunicaciones estableciendo la hora y lugar para el seguimiento diario. Por la metodología utilizada, los principales canales de comunicación son el equipo son el Scrum Master, que se encarga de ser un facilitador para que el Scrum Team pueda ejecutar sus actividades, y el Product Owner, que de igual forma debe facilitar al Scrum Team con las definiciones y toma de decisiones del lado de negocio.

Este documento es aprobado de nueva cuenta por el product owner y las gerencias de negocio y Tecnología y Sistemas.

### **3.9 Arquitectura del Sistema**

La solución arquitectónica se desarrollará bajo una plataforma WEB, dicha solución tendrá las siguientes características:

- La interfaz del usuario se realizará con JQuery
- La lógica de negocio se implementará con Java EE
- Se utilizará el patrón de diseño Modelo – Vista – Controlador
- Se implementó de acuerdo a los estándares del grupo financiero un modelo de tres capas con las siguientes características.
  - Servidor web con balanceador de cargas.
  - Servidor de aplicaciones con configuración de grupo de servidores (cluster).
  - Motor de base de datos

- Enterprise Service Bus para la mensajería con INDEVAL con configuración en un grupo de Servidores
- Esquema de contingencia para toda la infraestructura.

Se utilizaron los siguientes marcos de trabajo para la solución arquitectónica, los cuales nos permiten utilizar mejores prácticas en el desarrollo de software; así como a minimizar las líneas de código y posibles errores:

- Struts
- Spring
- Ibatis
- Log4j
- iText
- JasperReports
- Enterprise Service Bus

### **3.9.1 Beneficios**

Con esta arquitectura se busca mejorar sustancialmente el rendimiento de la plataforma, ya que adicionalmente se considera:

- Desacoplamiento entre las capas de nuestro modelo.
- Inyección de dependencias
- Autenticación de Usuarios en capa de negocio.
- Validaciones de input y output de información.
- Desconcentración de labores mediante procesos maker / checker.
- Protocolos de comunicación seguros en cada una de las capas de nuestro modelo.
- Protección contra pérdida de información y calidad del servicio.
- Manejo de errores y bitácoras.
- Concurrencia y transacciones.
- Reutilización de componentes.
- Escalabilidad y redundancia.
- Alto Rendimiento.
- Alto soporte.
- Alta disponibilidad.
- Utilización de Patrones de Diseño.
- Utilización de marcos de trabajo de última generación.
- Automatización de pruebas con Junit.
- Builds automáticos y manejo de dependencias.
- Entrega incremental.

### 3.9.2 Diagramas

Esta propuesta se revisó con el área de arquitectura de la casa de bolsa y se presentó el diagrama de la ilustración XVI.

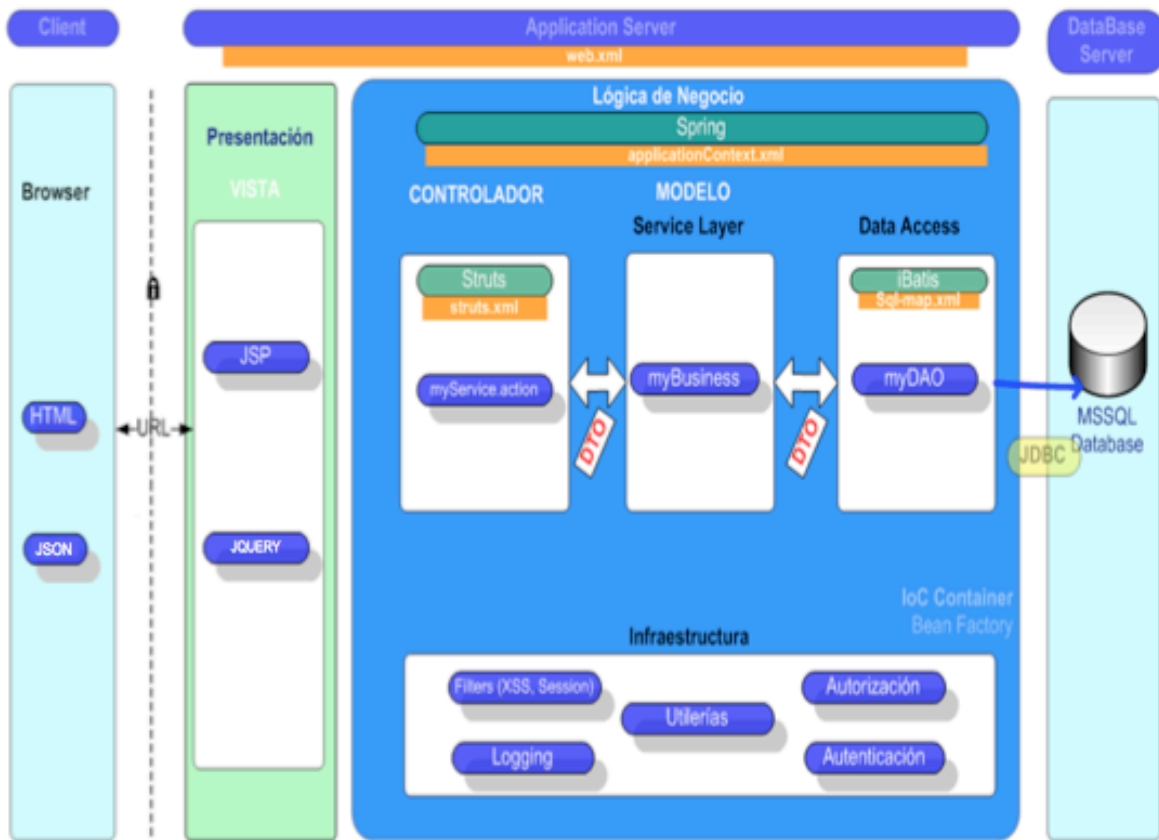


ILUSTRACIÓN XVI. DIAGRAMA DE ARQUITECTURA PROPUESTA.

Como parte de las funciones que cubre el área de arquitectura, se construyen en conjunto los diagramas de despliegue, interfaces, topología y componentes.

### 3.10 Ejecución y Desarrollo del proyecto

En esta sección se describe el proceso de desarrollo de software durante el proyecto así como los estándares utilizados durante el desarrollo del mismo.

### 3.10.1 Desarrollo del Sprint

Una vez finalizada la junta de sprint y aprobado el documento de Sprint Backlog, el Scrum Team comienza con sus actividades diarias para cumplir con el objetivo de atender los requerimientos durante un periodo de 20 días.

El sprint comienza con la junta diaria donde los miembros del Scrum Team, el Scrum Master y de ser necesario el Product Owner contestan a las preguntas de ¿Qué hicieron el día anterior?, ¿Qué van a hacer el día de hoy? Y si existe algún obstáculo o riesgo que evite completar alguna actividad. Una vez terminada esta ronda, cada miembro del equipo actualiza el tablero de control donde se encuentran todas las actividades del sprint colocando las actividades en terminadas, en proceso o pendientes. En base a esta información, el Scrum Master actualiza la gráfica de actividades o burndown chart. Conforme avanzan los días para completar el sprint, el burndown chart es una herramienta bastante útil para identificar posibles retrasos, riesgos o actividades que se están retrasando y tomar decisiones o acciones para corregir el rumbo del sprint antes de finalizar el periodo de 20 días.

### 3.10.2 Nomenclatura

Como buena práctica en el proyecto, se determinó establecer ciertos estándares para los componentes desarrollados por el equipo. Al ser un desarrollo en múltiples servidores (application server, enterprise service bus, servidor para tareas calendarizadas) se determinó la siguiente nomenclatura de objetos Java y organización de los paquetes.

Tabla 3. Nomenclatura de paquetes.

Campo	Descripción	Ejemplo
Título de la pantalla	Nombre completo de la pantalla	Consulta de traspasos de valores
Nombre del módulo	Debe ser de manera abreviada, bajo el diccionario de datos aprobado por el comité de sistemas de la Casa de Bolsa	traspasoVal
Nombre del paquete	Sistema_modulo_versión.jar	slv_traspasoVal_1.0.0.jar
Paquete de archivos de configuración y properties	Se toma como base el paquete estándar de sistemas	config.xml
Paquete del módulo	mx.com.compania.utils.modulo	mx.com.compania.utils.traspasoVal
Paquete para clases de acceso a base de datos	mx.com.compania.utils.modulo.dao	mx.com.compania.utils.traspasoVal.dao
Paquete para clases para lógica de negocio	mx.com.compania.utils.modulo.business	mx.com.compania.utils.traspasoVal.business

<b>Paquete para clases de tipo de objeto de transferencia de datos</b>	mx.com.compania.utils.modulo.modelo	mx.com.compania.utils.traspasoval.modelo
--	-------------------------------------	--

La definición de la nomenclatura para los objetos es la siguiente, la cual se muestra en la Tabla 4.

Tabla 4. Nomenclatura de objetos.

<b>Campo</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>Clase Controladora</b>	SivClsNombreAction.java	SivClsTraspasoValAction.java
<b>Interfaz de la clase de negocio</b>	SivClsNombreBusinessService.java	SivClsTraspasoValBusinessService.java
<b>Implementación de la clase de negocio</b>	SivClsNombreBusinessImpl.java	SivClsTraspasoValBusinessImpl.java
<b>Interfaz de la clase de acceso a la base de datos</b>	SivClsTraspasoValDAO.java	SivClsTraspasoValDAO.java
<b>Implementación de la clase de acceso a la base de datos</b>	SivClsNombreDAOImpl.java	SivClsTraspasoValDAOImpl.java
<b>Clase de pruebas unitarias</b>	SivClsNombreUnitTest.java	SivClsTraspasoValUnitTest.java
<b>JSP</b>	slvmodulo.jsp	slvtraspasoval.jsp
<b>Archivos de propiedades</b>	slvmodulo.properties	slvtraspasoval.properties

Para los archivos de configuración de los marcos de trabajo, la nomenclatura es la que se muestra en la Tabla 5.

Tabla 5. Nomenclatura de archivos de configuración.

<b>Marco de Trabajo</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>Struts</b>	struts-modulo.xml	struts-traspasoval.xml
<b>Spring</b>	modulo-applicationcontext.xml	traspasoval-applicationcontext.xml
<b>Ibatis</b>	modulomap.xml	traspasovalmap.xml
<b>Log4j</b>	Log4j.properties, ya que solo se tiene uno por todo el proyecto.	Log4j.properties

### 3.10.3 Componentes del servidor web

En el caso de los componentes estáticos, se estableció la siguiente nomenclatura como se muestra en la tabla No 6.

Tabla 6. Nomenclatura de componentes web.

Componente	Descripción	Ejemplo
Imágenes	slvnombre.jpg	slvlogin.jpg
HTML	slvmodulo.html	slvtraspasoval.html
JS	slvmodulo.js	slvtraspasoval.js
CSS	slvmodulo.css	slvtraspasoval.css

### 3.10.4 Componentes del servidor de base de datos y reportes.

Para los componentes del servidor de base de datos y servidor de reportes, solo se colocará como prefijo la abreviación del sistema y una breve descripción abreviada del componente, por ejemplo: siidetposicion.rpt o para un servicio de base de datos sii\_r\_consdetposicion, en donde el segundo campo se refiere si estamos hablando de una consulta (c), inserción (i), actualización(a) o un proceso (p).

Tabla 7. Nomenclatura de objetos de base de datos.

Tipo de Objeto	Descripción	Ejemplo
Stored procedure	BaseDatos_sp_descripcion	Val_spu_traspasoVal
Vista	BaseDatos_vw_descripcion	Val_vw_nombreInstitucion
Trigger	BaseDatos_tx_tabla	Val_ti_traspasoVal

### 3.10.5 Integración continua

En un proyecto de estas características es natural generar una cantidad considerable de componentes de software. Para evitar colisiones en cuanto a versiones o errores de dependencias o compilados, se determinó hacer uso de herramientas como Team Foundation System y Maven para realizar la integración continua y builds del proyecto de forma automatizada. Esto garantizaba las entregas de código al área de calidad así como la generación de paquetes para instalar en el proceso de release.



La figura XVII ilustra el proceso de integración en el proceso del Sprint.

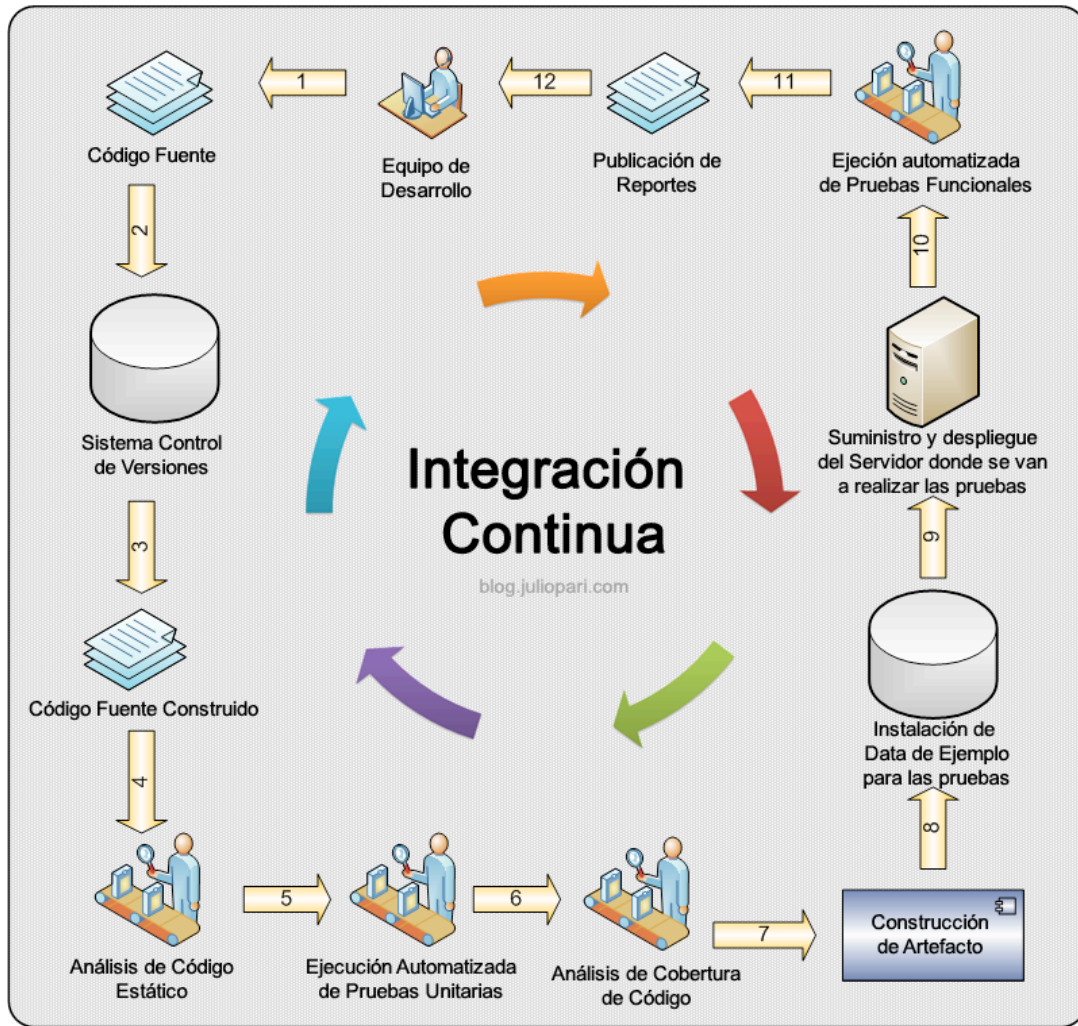


ILUSTRACIÓN XVII. PROCESO DE INTEGRACIÓN CONTINUA.

### 3.10.6 Esquema de Seguridad

El esquema de seguridad se desarrolló bajo los estándares del grupo financiero, del cual pertenece la casa de bolsa, para este informe solo se mencionará que se manejaron filtros de crossite, filtros de sesión, validadores de url, validadores de navegación, encapsulamiento de objetos, utilización del componente estándar de seguridad con el nivel de cifrado requerido para cualquier desarrollo de acuerdo a los estándares del grupo financiero.

Además se implementaron y desarrollaron todos los controles para el cierre de posibles vulnerabilidades, solicitados por el grupo financiero para dar cumplimiento a las políticas internas de seguridad.

### 3.11 Pruebas y Demostración del producto

Como parte del alcance de la junta de inicio de sprint, se define las características de las pruebas para probar cierto requerimiento en particular así como el script de pruebas funcionales y los criterios de aceptación. Además, el tester del equipo debe ejecutar las pruebas de seguridad, regresión y stress.

Para cada uno de los requerimientos se realizaron los siguientes tipos de pruebas:

- Pruebas unitarias
- Pruebas de integración
- Pruebas con usuarios de producción
- Pruebas de carga y Stress
- Pruebas de liberación

A diferencia de una metodología en cascada, en la metodología ágil la entrega es continua, una vez que se termina un requerimiento y se prueba en desarrollo se entrega al tester para que comience las pruebas.

La figura XVIII ilustra el proceso de pruebas en la metodología.



ILUSTRACIÓN XVIII. PROCESO DE PRUEBAS.

#### 3.11.1 Pruebas Unitarias

Son aquellas pruebas individuales de los componentes que conforman la solución, a continuación se muestra la Tabla 8, con un ejemplo:

Tabla 8. Registro de pruebas unitarias.

Fecha	Acción	Responsable	Otros participantes
MM-YYYY	Se detalla los pasos a seguir para poder probar el requerimiento.	Líder Técnico	Usuarios de Negocio, Desarrolladores y Gerente de Sistemas

### 3.11.2 Pruebas de Integración

Son aquellas pruebas de los componentes individuales en conjunto, a continuación se muestra la Tabla 9, con un ejemplo:

Tabla 9. Registro de pruebas de integración.

Fecha	Acción	Responsable	Otros participantes
MM-YYYY	Probar los diferentes escenarios en el que se encontrará el usuario al momento de interactuar con el sistema en el ambiente de QA	Líder de Proyecto	Usuarios de Negocio, Desarrolladores y Gerente de Sistemas

### 3.11.3 Pruebas con usuarios de producción

Estas son las pruebas que involucran una interacción con los usuarios de producción para probar todos los componentes a liberar, a continuación se muestra la Tabla 10, con un ejemplo:

Tabla 10. Registro de pruebas con usuario.

Fecha	Acción	Responsable	Otros participantes
MM-YYYY	Validar los requerimientos y el script de pruebas técnicas	Usuarios de Negocio	Desarrolladores, Líder de Proyecto y Gerente de Sistemas

### 3.11.4 Pruebas de Carga/Stress

Las pruebas al ambiente sometiéndolo a una capacidad superior a la que se anticipó que manejaría en un ambiente de producción, e.g.: enviar diez veces la carga de información prevista a través de la red, a continuación se muestra la Tabla 11, con un ejemplo:

Tabla 11. Registro de pruebas de carga.

Fecha	Acción	Responsable	Otros participantes
MM-YYYY	Aumentar gradualmente el número de transacciones recibidas para verificar el comportamiento con una mayor cantidad de datos	Área de Calidad	Desarrolladores, Líder de Proyecto y Gerente de Sistemas

### 3.11.5 Pruebas de Liberación

Pruebas completas del usuario sobre los entregables, a continuación se muestra la tabla 12, con un ejemplo:

Tabla 12. Registro de pruebas de liberación.

Fecha	Acción	Responsable	Otros participantes
MM-YYYY	Revisar todo el funcionamiento del sistema, conforme a los requerimientos que se entregan.	Área de Calidad y usuarios de negocio.	Desarrolladores, Líder de Proyecto y Gerente de Sistemas

### 3.12 Cierre de Sprint

Para cada uno de los sprints o iteraciones ejecutadas en el proyecto, se realiza al final una junta con el Scrum Team, Scrum Master y Product Owner con el objetivo de mejorar de manera la productividad del equipo y la calidad del producto que está desarrollando. El Scrum Team analiza cómo ha sido su manera de trabajar durante el Sprint, responde a la pregunta de por qué está consiguiendo o no los objetivos a que se comprometió al inicio del sprint . De manera general responde a las siguientes preguntas.

- Qué cosas han funcionado bien.
- Cuales hay que mejorar.
- Qué cosas quiere probar hacer en la siguiente iteración.
- Qué ha aprendido.
- Cuales son los problemas que podrían impedirle progresar adecuadamente.

El Scrum Master se encargará de ir eliminando los obstáculos identificados que el propio equipo no pueda resolver por sí mismo. También se actualiza el plan de trabajo con los riesgos u oportunidades que se han identificado, los problemas que se han presentado y la matriz de impacto de cada uno de ellos.

De igual manera al finalizar esta junta se determina si el proyecto continua con su siguiente sprint o iteración. En caso de que el product owner determine seguir y o modificar el alcance del proyecto, estas modificaciones al alcance se documentan en el plan de trabajo y se integran a la bitácora de requerimientos para que se integren en un nuevo sprint de acuerdo a las prioridades establecidas por el negocio.



## CAPÍTULO 4. LIBERACIÓN A PRODUCCIÓN

---



## **4 Liberación a producción.**

Para el proceso de liberación o instalación en producción, al finalizar el sprint el área de calidad genera una solicitud de cambio con las especificaciones del paquete a instalar y las instrucciones para instalarlo. Dicha documentación contiene la siguiente información.

- Aplicación.
- Versión.
- Permisos requeridos.
- Creación de carpetas.
- Modificación de configuración de servidores.
- Procedimiento para generar instalación.
- Procedimiento de regresión en caso de error.

El propósito es que el área de Administración de Cambios o Change Management tenga un mejor control y conocimiento del contenido del paquete que le manda el área de calidad para que realicen la liberación en la fecha y hora que se especificó y acordó con el usuario. Además de coordinar a los equipos de operaciones que se requiera para la instalación del producto en producción.

### **4.1 Resultados**

A continuación se enlistan algunos de los resultados y beneficios que se obtuvo con la implementación del sistema.

- Disminución de costos operativos.

Ahorro estimado en utilización del portal del INDEVAL de aproximadamente 1,000,000.00 MXN anuales

- Disminución de errores de usuario.

Se eliminaron procesos manuales en la generación de movimientos y exportación.

- Automatización de operaciones.

Se automatizaron las operaciones de las áreas de Préstamo de Valores, Sociedades de Inversión, Posición propia, Traspaso de Valores, Recompra, Custodia Externa, Mercado de Dinero, Capitales y Valores.



- Seguimiento en tiempo real.

Posibilidad de consultar las operaciones desde que son capturadas por las áreas de negocio, hasta que se completa el ciclo de liquidación en INDEVAL.

- Reportes y Estadísticas.

Consultar la información en un tablero con gráficas, estadísticas, alertas y reportes en PDF y excel.

- Consultas históricas.

Información histórica de hasta 10 años disponible para auditorías de negocio o regulatorias.

- Escalabilidad y Volumen.

Sistema diseñado para soportar un volumen de crecimiento de 10% de operaciones durante 10 años y más de 100,000 operaciones diarias.

- Bitácoras.

Disponibilidad de bitácoras en el sistema para seguimiento de operaciones.

- Esquema de contingencia.

Sistema diseñado para contingencia operativa y garantizando la entrega de mensajes al INDEVAL.

- Interfaz de usuario amigable.

Diseño de interfaz gráfica de usuario con componentes personalizables, datagrids, gráficas, exportación a excel, pdf, copiar registros, ordenamiento para facilitar el uso del sistema.

- Iteraciones continuas.

Entregas o releases contínuos para que el usuario pudiera utilizar el sistema y ajustar y priorizar sus requerimientos.

- Integración Tecnología y Negocio.

Participación de todas las áreas de Tecnología y Sistemas de manera continua y comunicación directa con el usuario de negocio.

## CONCLUSIONES

---



## **Conclusiones**

Comencé mi trayectoria profesional en septiembre del 2005 como Ingeniero en Sistemas Junior en la Casa de Bolsa en el departamento del Centro de Desarrollo de Tecnología y Sistemas. Desde un comienzo, participé en proyectos de implementación y desarrollo de sistemas en diversas tecnologías.

Como parte de mi desarrollo dentro de la institución, crecí profesionalmente como Ingeniero en Sistemas Senior, Líder de proyecto, Subgerente y Gerente de Sistemas. Ofrecí mi apoyo, soporte y conocimientos a diferentes áreas dentro de la Casa de Bolsa como Clientes y Contratos, Recursos Humanos, Internal Compliance, Liquidación de Valores y Efectivo, Banca Patrimonial, Sociedades de Inversión y Global Common Team. Participé durante estos años en más de 30 proyectos medianos y grandes con diferentes características.

Este informe profesional describe uno de los proyectos más significativos ya que representó el primer proyecto administrado con metodologías ágiles dentro de la Casa de Bolsa y el Grupo Financiero. El resultado del mismo era determinante para la adopción de alternativas a la administración tradicional de la metodología. Tuve la fortuna de capacitarme como Certified Scrum Master para la gestión y administración correcta de la iniciativa.

La duración de la iniciativa fue de casi un año finalizando en noviembre del 2013, logrando establecer las bases para la adopción de Scrum dentro de la Casa de Bolsa y cumplir con los objetivos y requerimientos solicitados por el área de negocio. Las prácticas y tecnologías utilizadas sentaron como base estándares en la metodología de Software Management y mejores prácticas en el centro de desarrollo para el desarrollo de sistemas web, interfaces de usuario, integración continua, herramientas de control de versiones, mensajes financieros, pruebas unitarias y automatización de pruebas.

Además, la experiencia adquirida de los conocimientos de negocio, comunicación y negociación con áreas de sistemas y usuarios, planeación y organización de equipos de trabajo, representaron una oportunidad de crecimiento dentro del sector financiero para participar en proyectos globales.

Es un orgullo para mí estar desarrollando mi carrera profesional con muchos éxitos y aprendizaje, pero sobre todo, el haber podido aplicar los conocimientos adquiridos como estudiante de la Facultad de Ingeniería. Desde los principios básicos de programación estructurada, hasta el diseño de soluciones empresariales, ingeniería de software, administración de proyectos, diseño de bases de datos, lógica de programación, programación orientada a objetos, entre otros, sirvieron como los cimientos para mi desarrollo personal y profesional.

Además, estos conocimientos se complementaron con mi participación como becario de la División de Ingeniería Eléctrica dentro del Programa de Tecnología en Cómputo, dándome la oportunidad de participar en diversos proyectos para apoyar al programa y complementar mi aprendizaje.



## BIBLIOGRAFÍA Y REFERENCIAS

---



## ***Bibliografía y Referencias***

1. Scrum alliance. [www.scrumalliance.org](http://www.scrumalliance.org)
2. Spring framework. <http://projects.spring.io/spring-framework/>
3. Mybatis. <http://mybatis.github.io/mybatis-3/es/>
4. Struts framework. <http://struts.apache.org/release/2.3.x/>
5. Depósito Central de Valores. <http://www.indeval.com.mx>
6. Bolsa Mexicana de Valores. <http://www.bmv.com.mx>
7. Comisión Nacional para la Protección y Defensa de Usuarios de Servicios Financieros. <http://www.condusef.gob.mx/index.php/instituciones-financieras2/132-sector-bursatil/502-icomo-funciona-el-mercado-de-valores>
8. Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison – Wesley. 2012.
9. Pro Java EE Spring Patterns: Best Practices and Design Strategies Implementing Java EE Patterns with Spring Framework. Apress – Dhrubojyoti Kayal 2008.
10. A Guide to the Project Management Body of Knowledge. Project Management Institute. 2009
11. Google Trends. (n.d.). Consultado el 16 de Octubre de 2014. Página web de tendencias de Google: <https://www.google.com/trends/explore#cat=013&q=jquery%2C%20%2Fm%2F04gkgv%2C%20%2Fm%2F0d6jq1%2C%20ext%20js%2C%20%2Fm%2F02pqhn2&cmpt=q>
12. Wikipedia consultado el 24 de enero del 2015. <http://es.wikipedia.org/wiki/AJAX>





## GLOSARIO DE TÉRMINOS

---



## **Glosario de Términos**

### **AJAX.**

Acrónimo de Asynchronous JavaScript And XML por sus siglas en inglés. Es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma, es posible realizar cambios sobre la página sin recargarla.

### **Arbitraje**

Una operación de arbitraje, consiste en adquirir un producto financiero en un mercado a un precio y revenderlo a otro mayor en un mercado diferente. Como consecuencia de ello, se obtiene una igualdad en precios en todos los mercados.

### **Asset Management**

Es una actividad que consiste en la gestión responsable del capital y las garantías de las empresas como bonos y acciones por parte de instituciones administradoras, para alcanzar los mayores beneficios a los inversores.

### **Back Office Valores**

Las funciones de Back office de Valores consiste en supervisar, verificar y liquidar las operaciones financieras realizadas por las áreas de trading.

### **Banco**

Es una empresa financiera que se encarga de captar recursos en la forma de depósitos, y prestar dinero, así como la prestación de servicios financieros.

### **Banco Central**

Es la institución que ejerce como autoridad monetaria y como tal suele ser la encargada de la emisión del dinero legal y en general de diseñar y ejecutar la política monetaria del país.

### **Bolsa de valores**

Es una organización privada que brinda las facilidades necesarias para que sus miembros, atendiendo los mandatos de sus clientes, introduzcan órdenes y realicen negociaciones de compra y venta de valores, tales como acciones de sociedades o compañías anónimas, bonos públicos y privados, certificados, títulos de participación y una amplia variedad de instrumentos de inversión.

### **Casa de Bolsa**

Las Casas de Bolsa son sociedades anónimas dedicadas a la intermediación con valores, lo que comprende el poner en contacto a oferentes y demandantes de valores, así como ofrecer y negociar valores por cuenta propia o de terceros en el mercado primario o secundario.

### **CNBV**

La Comisión Nacional Bancaria y de Valores (CNBV) es una autoridad financiera mexicana, órgano desconcentrado de la Secretaría de Hacienda y Crédito Público (SHCP) con autonomía técnica y facultades ejecutivas. Su objetivo es supervisar y regular a las entidades integrantes del sistema financiero mexicano, a fin de procurar su estabilidad y correcto funcionamiento, así como mantener y fomentar el sano y equilibrado desarrollo de dicho sistema en su conjunto, en protección de los intereses del público.

## DALI

Es un sistema de liquidación de operaciones con valores. DALÍ utiliza un algoritmo de compensación que frecuentemente compensa y liquida grupos de operaciones y liga la entrega de valores con el pago correspondiente, es decir, sigue la práctica de “entrega contra pago”.

## Deposito de valores

Es una sociedad anónima de capital variable que en general tiene por objetivo prestar el servicio de guardar, administrar, compensar, liquidar y transferir valores.

## EJB

Enterprise JavaBeans forman parte del estándar de construcción de aplicaciones empresariales J2EE. Su especificación detalla cómo los servidores de aplicaciones proveen objetos desde el lado del servidor, que son, precisamente, los EJB. Dentro de la especificación se detalla la comunicación remota de objetos, transacciones, control de la concurrencia, mensajería, servicios de nombres y de directorio, seguridad, ubicación de componentes en un servidor de aplicaciones.

## Enterprise Service Bus

Un bus de servicios de empresa (ESB) consiste en un combinado de arquitectura de software que proporciona servicios fundamentales para arquitecturas complejas a través de un sistema de mensajes (el bus) basado en las normas y que responde a eventos. Los desarrolladores normalmente implementan un ESB utilizando tecnologías de productos de infraestructura de middleware que se basan en normas reconocidas.

## Flex

Es una plataforma de desarrollo web para construir aplicaciones enriquecidas de internet, también llamadas RIA en un modelo multi-capa, las aplicaciones Flex son el nivel de presentación.

## Framework

Es un conjunto estandarizado de conceptos, prácticas y criterios, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.

## Gráfica burndown

Es una representación gráfica del trabajo por hacer en un proyecto en el tiempo. Usualmente el trabajo remanente (o backlog) se muestra en el eje vertical y el tiempo en el eje horizontal. Es decir, el diagrama representa una serie temporal del trabajo pendiente.

## Ibatis

iBATIS es un framework de código abierto basado en capas que se ocupa de la capa de Persistencia (se sitúa entre la lógica de Negocio y la capa de la Base de Datos).

## Indeval

El Instituto de Depósito de Valores, o INDEVAL por sus siglas, posee autorización de acuerdo a la Ley, para operar como Depósito Central de Valores. En consecuencia, se encuentra catalogado como proveedor para el sistema financiero, de guarda, custodia, administración, compensación y liquidación de valores.

## IOC

Inversión de control (*Inversion of Control* en inglés, IoC) es un método de programación en el que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, en los que la interacción se expresa de forma imperativa haciendo llamadas a procedimientos (procedure calls) o funciones.

## Itext

Text es una biblioteca Open Source para crear y manipular archivos PDF, RTF, y HTML en Java.

## J2EE

Java Platform, Enterprise Edition o Java EE, es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

## Jasper

JasperReports es una librería de creación de reportes que tiene la habilidad de entregar contenido enriquecido en formatos PDF, HTML, XLS, CSV y XML. Está escrito completamente en Java y puede ser usado en gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones web, para generar contenido dinámico.

## JDBC

Java Database Connectivity, por sus siglas JDBC, que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

## JMS

Servicio de mensajes Java, también conocida por sus siglas en inglés JMS, es un estándar de mensajería que permite a los componentes de aplicaciones basados en la plataforma Java crear, enviar, recibir y leer mensajes. También hace posible la comunicación confiable de manera síncrona y asíncrona.

## JQuery

Es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol de objetos, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

## JSON

JSON, acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

## JSP

JavaServer Pages (JSP) es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML, entre otros tipos de documentos.

## MVC

El modelo vista controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

## Préstamo de valores

Una operación de préstamo de valores puede efectuarse sobre instrumentos de deuda o acciones, y consiste en un acuerdo contractual entre 2 partes, en el cual el "prestamista" se compromete a ceder temporalmente la posición en un instrumento al "prestatario" a cambio de un premio económico. Los valores no dejan de ser propiedad del prestamista, por lo que al término del contrato, el prestatario deberá devolver estos o algún instrumento de la misma especie.

## Product Backlog

El product backlog es un documento de alto nivel para todo el proyecto. Contiene descripciones genéricas de todos los requisitos, funcionalidades deseables, etc. priorizadas según su retorno sobre la inversión (ROI) . Es el qué va a ser construido. Es abierto y solo puede ser modificado por el product owner. Contiene estimaciones realizadas a grandes rasgos, tanto del valor para el negocio, como del esfuerzo de desarrollo requerido.

## RIA

Las rich Internet application, o RIA (en español "aplicaciones de Internet enriquecidas"), son aplicaciones web que tienen la mayoría de las características de las aplicaciones de escritorio tradicionales. Estas aplicaciones utilizan un navegador web para ejecutarse y por medio de complementos o mediante una máquina virtual se agregan las características adicionales.

## Scrum

SCRUM es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el Scrum Master, que mantiene los procesos y trabaja de forma similar al director de proyecto, el ProductOwner, que representa a los interesados externos o internos, y el equipo que incluye a los desarrolladores.

### Scrum Master

Su rol primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El Scrum Master no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El Scrum Master se asegura de que el proceso Scrum se utiliza como es debido. El Scrum Master es el que hace que las reglas se cumplan.

### Scrum Team

El equipo tiene la responsabilidad de entregar el producto. Un pequeño equipo de 3 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación, etc).

## SDLC

Denominado como ciclo de vida del desarrollo de software, o SDLC por sus siglas en inglés, es una estructura aplicada al desarrollo de un producto de software. Hay varios modelos a seguir para el establecimiento de un proceso para el desarrollo de software, cada uno de los cuales describe un enfoque diferente para diferentes actividades que tienen lugar durante el proceso.

## Sentra

Sistema Electrónico de Negociación, SENTRA, es el sistema desarrollado y administrado por la Bolsa Mexicana de Valores (BMV), al que tienen acceso los Miembros, para la formulación de posturas y concertación de operaciones en el mercado de capitales.

## Sociedades de inversión

Estas instituciones forman carteras de valores o portafolios de inversión con los recursos que captan del público inversionista. La selección de estos valores se basa en el criterio de diversificación de riesgos. Al adquirir las acciones representativas del capital de estas sociedades, el inversionista obtiene ventajas tales como la diversificación de sus inversiones, principio fundamental para disminuir el riesgo y, la posibilidad de participar del Mercado de Valores en condiciones favorables sin importar el monto de los recursos aportados.

## Spring

Spring es un framework liviano y no intrusivo. Sus características principales son inyección de dependencias y programación orientada a aspectos.



## SQL

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

## Struts

Struts es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma Java EE (Java Enterprise Edition).

## SWIFT

Por sus siglas, Society for Worldwide Interbank Financial Telecommunication, es una organización que tiene a cargo una red internacional de comunicaciones financieras entre bancos y otras entidades financieras.

## XML

Por sus siglas en inglés de eXtensible Markup, es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

## ANEXOS

---



## **Anexo A. Minutas**

Las minutas que se manejaron, las registró el área de control de desarrollo y fueron supervisadas por el líder de proyecto, se enviaron posteriormente a los involucrados para su visto bueno.

Para la redacción de las minutas, se tomaron las plantillas definidas por el área de tecnología y sistemas y las cuales están constituidas por las siguientes secciones:

1. Acuerdos
2. Pendientes
  - Descripción de pendiente.
  - Responsable.
  - Fecha de registro.
  - Fecha objetivo.
  - Última Fecha objetivo.
  - Estatus.
  - Comentarios.
3. Riesgos
  - Descripción de riesgo
  - Fecha de registro
  - Severidad
  - Estatus
  - Líder de Proyecto
  - Responsable
  - Identificador de actividad del plan de trabajo.
  - Tipo de Riesgo
4. Problemas
  - Descripción del problema.
  - Estatus
  - Prioridad
  - Fecha de Registro
  - Líder de Proyecto
  - Responsable
  - Fecha Objetivo
  - Identificador de la actividad del plan de trabajo.

## **Anexo B. Matriz de Aprobaciones**

### *Fase de Planeación*

- *Aprobación del sponsor del proyecto.*
- *Aprobación del director del área del usuario.*
- *Aprobación del gerente de desarrollo.*

### *Product Backlog*

- *Aprobación del usuario.*
- *Aprobación del director del área del usuario.*
- *Aprobación del gerente de desarrollo.*
- *Aprobación del área de arquitectura.*
- *Aprobación del área de calidad.*
- *Aprobación del área de seguridad de la información.*
- *Notificación al área de liberación.*
- *Notificación a oficina de proyectos.*

### *Sprint Backlog*

- *Aprobación del usuario.*
- *Aprobación del gerente de desarrollo.*
- *Aprobación del área de arquitectura.*
- *Notificación a oficina de proyectos.*
- *Notificación a Internal Compliance.*

### *Cierre de Sprint*

- *Aprobación del usuario.*
- *Aprobación del gerente de desarrollo.*
- *Aprobación del área de arquitectura.*
- *Notificación a oficina de proyectos.*
- *Notificación a change management.*

### *Demostración de producto y aceptación de pruebas*

- *Aprobación del usuario.*
- *Aprobación del gerente de desarrollo.*
- *Aprobación del área de calidad.*
- *Notificación al área de liberación.*

### *Cierre del proyecto*

- *Aprobación del usuario.*
- *Aprobación del director del área del usuario.*
- *Aprobación del gerente de desarrollo.*
- *Notificación a oficina de proyectos.*

### ***Anexo C. Estimación por planning Poker.***

Es una técnica para calcular una estimación basado en un consenso, en su mayoría utilizado para estimar el esfuerzo o el tamaño relativo de las tareas de desarrollo de software. Las cartas en el mazo están numeradas. Un mazo típico contiene tarjetas mostrando la secuencia de Fibonacci incluyendo un cero: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89. Otros mazos utilizan progresiones similares como por ejemplo la siguiente secuencia: 0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40, 100.

En la reunión de la estimación a cada estimador se le da un conjunto de las tarjetas. La reunión prosigue de la siguiente manera:

Un moderador, por lo general el Scrum Master, que no jugará, preside la reunión, apoyado y asesorado por el Product Owner. El desarrollador con más conocimiento de una determinada característica proporciona una breve introducción sobre la misma. El equipo tiene la oportunidad de hacer preguntas y discutir para aclarar los supuestos y riesgos.

Cada persona coloca una tarjeta boca abajo que representa su estimación en Story Points. Todo el mundo muestra sus tarjetas en forma simultánea. A las personas con estimaciones altas y bajas se les da un tiempo para ofrecer su justificación para la estimación y la discusión continúa. Se repite el proceso de cálculo hasta que se alcance un consenso. El desarrollador más especializado tiene una gran parte del voto de consenso, aunque el moderador puede negociar el consenso. Se puede utilizar un reloj para asegurar que el debate sea estructurado, el Scrum Master podrá en cualquier punto terminar el reloj y cuando se acaba toda discusión debe cesar y otra ronda de póquer se juega con otra historia de usuario.