

Reporte Final de Estadía

Fredy Arturo Reyes Rojas

“Cambio de plataforma de sucursales a web”



Universidad Tecnológica del Centro de Veracruz

Programa educativo

Tecnologías de la Información

Reporte para obtener título de Ingeniero en Tecnologías de la
Información

Proyecto de estadía realizado en la empresa Solser Sistem S.A. de
C.V.

Nombre del proyecto

“Cambio de plataforma de sucursales a web”

Presenta

Fredy Arturo Reyes Rojas

Cuitláhuac, Ver., a 5 de Abril de 2018.



Universidad Tecnológica del Centro de Veracruz

Programa Educativo
Tecnologías de la Información

Nombre del Asesor Industrial
Ing. María Isabel Pozas Naranjo

Nombre del Asesor Académico
Lic. Sergio Velázquez Bonilla

Jefe de Carrera
Lic. César Aldaraca Juárez

Nombre del Alumno
Fredy Arturo Reyes Rojas

Contenido

AGRADECIMIENTOS	1
RESUMEN	2
CAPÍTULO 1. INTRODUCCIÓN	3
1.1 Estado del Arte	4
1.2 Planteamiento del Problema.....	5
1.3 Objetivos	5
1.4 Definición de variables.....	6
1.5 Hipótesis.....	6
1.6 Justificación del Proyecto	6
1.7 Limitaciones y Alcances.....	7
1.8 Solser Sistem S.A. de C.V.	9
1.8.1 Historia de la empresa	9
1.8.2 Misión	9
1.8.3 Visión	9
1.8.4 Objetivo	10
1.8.5 Valores	10
1.8.6 Políticas de la empresa	10
1.8.6.1 Cumplimiento a visión, misión y valores	10
1.8.6.2 Participación en iniciativas organizacionales	10
1.8.6.3 Uso de estacionamiento.....	11
1.8.6.4 Salario y viáticos.....	11
1.8.6.4.1 Pago de nómina	11
1.8.6.4.2 Reembolso de viáticos.....	12
1.8.7 Organigrama general.....	12
1.8.8 Organigrama del área CMMi	13
1.8.9 Procesos que se realizan en la empresa	13
1.8.10 Mercado de impacto de los productos o servicios brindados por la empresa	14
10.8.11 Impacto en el área de tecnologías de la información y comunicación	14
CAPÍTULO 2. METODOLOGÍA	16

2.1 Metodología SCRUM Solser	16
2.2 Roles método SCRUM.....	17
2.3 Gestión de fases del proyecto	18
2.4 Uso de metodología SCRUM	19
2.5 Gestión del modelo de gobierno del proyecto	20
CAPÍTULO 3. DESARROLLO DEL PROYECTO	21
3.1 Estructura del proyecto	21
3.1.2 Estructura de directorios	21
3.2 Construcción	23
PROPUESTA TÉCNICA.....	23
Resumen Ejecutivo	23
<i>Antecedentes.....</i>	23
<i>Objetivo.....</i>	23
<i>Alcance.....</i>	23
<i>Referencias.....</i>	25
<i>Premisas.....</i>	25
Tecnologías Propuestas y Arquitectura.....	25
<i>Tecnologías Software y Hardware:</i>	25
<i>Diagrama de la Arquitectura Propuesta.</i>	27
<i>Situación actual de las operaciones en Sucursal</i>	29
<i>Detalle generales de la solución.....</i>	29
<i>Detalle de actividades</i>	30
Riesgos y dependencias.	41
PLAN DE TRABAJO	43
<i>Objetivo del documento</i>	43
<i>Audiencia del documento.....</i>	43
Información general del proyecto	43
<i>Claves y Nombres.....</i>	43
<i>Objetivos del Proyecto.....</i>	43

Proceso definido para el Proyecto	44
<i>Proceso Estándar y Adaptaciones</i>	<i>44</i>
Recursos Involucrados en el Proyecto.....	44
Planeación y Seguimiento del proyecto	45
<i>Valoración del Proyecto</i>	<i>45</i>
<i>Documentos Técnicos Proyecto / Mantenimiento</i>	<i>46</i>
<i>Administración de Datos / Información / Conocimiento.....</i>	<i>47</i>
<i>Seguimiento</i>	<i>48</i>
<i>Resolución de Problemas</i>	<i>48</i>
<i>Otros planes que influyen al proyecto / mantenimiento.....</i>	<i>48</i>
Organización del Proyecto	49
<i>Organigrama de Equipo de trabajo.....</i>	<i>49</i>
<i>Principales Hitos y Entregables</i>	<i>49</i>
<i>Delegación de Responsabilidades</i>	<i>50</i>
<i>Evaluación de desempeño de equipo integrado</i>	<i>50</i>
ARQUITECTURA DE LA APLICACIÓN	51
<i>Sobre este Documento (Propósito).....</i>	<i>51</i>
<i>Referencias.....</i>	<i>51</i>
Descripción General.....	51
<i>Objetivos del diseño</i>	<i>51</i>
Vistas y Planos	52
<i>Vista de Casos de Uso</i>	<i>52</i>
<i>Diagrama:.....</i>	<i>53</i>
<i>Vista Lógica</i>	<i>53</i>
<i>Módulos</i>	<i>53</i>
<i>Diagrama de clases.....</i>	<i>55</i>
<i>Diagramas de Secuencia</i>	<i>55</i>
Arquitectura del Proyecto.....	56
Calidad.....	57

INVENTARIO DE PROGRAMAS	58
INVENTARIO DE CLASES	61
<i>Clase 1 [cajacontroller] 01 Apertura de inicio de día</i>	<i>61</i>
<i>Clase 2 [CierreTemporalEstacionController] 02 Cierre temporal de estación</i>	<i>64</i>
<i>Clase 3 [AperturaDeEstacionController] 03 Apertura de estación controller.....</i>	<i>66</i>
<i>Clase 4 [CapturaTarjetasController] 500 Captura de tarjetas.....</i>	<i>68</i>
<i>Clase 5 [CierreEstacionFinDiaController] 04 Cierre de estación fin de día controller</i>	<i>72</i>
<i>Clase 6 [CapturarConcentracionTombolaController] 46 Captura de concentración de Tómbola controller 75</i>	<i>75</i>
<i>Clase 7 [SolicitudDeDotacionBilleteController] 27 Solicitud de dotación billete</i>	<i>78</i>
<i>Clase 8 [SolicitudDeDotacionDeBilleteBusinessImpl] Solicitud de dotación de billete businessImpl.....</i>	<i>81</i>
<i>Clase 9 [SolicitudDeDotacionDeBilleteBusiness] Solicitud de dotación de billete Bussines.....</i>	<i>83</i>
<i>Clase 10 [EliminacionDeSobrantesDeCajaController] 20 ELIMINACION DE SOBRANTES DE CAJA CONTROLLER</i>	<i>84</i>
<i>Clase 11 [FaltantePorRoboAsaltoController] 45 Faltante por robo o asalto.....</i>	<i>87</i>
<i>Clase 12 [SobrantesDeCajaController] 19 Sobrante de caja</i>	<i>89</i>
<i>Clase 13 [DevolucionACajaEnGeneralController] 06 Devolución a caja en general controller.....</i>	<i>91</i>
<i>Clase 14 DesgloceDeMonedaNacionalModalController] COGEN01 Desglose de moneda nacional modal controller.....</i>	<i>94</i>
<i>Clase 15 [DotacionDeCajaGeneralController] 05 Dotación de caja general controller</i>	<i>99</i>
<i>Clase 16 [TotalesCifrasSucursalController] 21 Totales cifras sucursal controller</i>	<i>102</i>
<i>Clase 17 [SolicitudDeDotacionMorrallaController] 28 Solicitud de dotación de morralla</i>	<i>104</i>
<i>Clase 18 [PagoDeBilleteAutenticoController] 51 Pago de billete autentico controller</i>	<i>106</i>
<i>Clase 19 [PagoDeBilleteAutenticoController] Pago de billete autentico controller</i>	<i>109</i>
<i>Clase 20 [ReimpresionDeTicketController] 11 Reimpresión de ticket controller</i>	<i>112</i>
<i>Clase 21 [EliminacionFaltantesDeCajaController] 18 Eliminación faltantes de caja controller</i>	<i>114</i>
MATRIZ DE TRAZABILIDAD.....	117
<i>Objetivo del documento</i>	<i>117</i>
CAPÍTULO 4. RESULTADOS Y CONCLUSIONES.....	121
4.1 Resultados.....	121
4.2 Trabajos Futuros	122

<i>4.3 Recomendaciones</i>	123
ANEXOS	124
<i>Inventario de Pantallas</i>	124
<i>Modelado de datos</i>	127
BIBLIOGRAFÍA	129

Tabla de ilustraciones

<i>Ilustración 1: Organigrama general Solser Sistem</i>	12
<i>Ilustración 2: Organigrama CMMi</i>	13
<i>Ilustración 3: Iteraciones metodología SCRUM</i>	16
<i>Ilustración 4: Roles SCRUM</i>	17
<i>Ilustración 5: Fases del proyecto</i>	18
<i>Ilustración 6: Metodología SCRUM</i>	19
<i>Ilustración 7: Herramienta Jira Software</i>	20
<i>Ilustración 8: Estructura de directorios</i>	22
<i>Ilustración 9: Diagrama de arquitectura propuesta</i>	27
<i>Ilustración 10: Organización del equipo de trabajo</i>	49
<i>Ilustración 11: CU – Consulta de clientes</i>	53
<i>Ilustración 12: Vista lógica</i>	53
<i>Ilustración 13: Diagrama de clases</i>	55
<i>Ilustración 14: Diagrama de secuencias</i>	55
<i>Ilustración 15: Flujo del programa</i>	60
<i>Ilustración 16: Matriz de trazabilidad</i>	120
<i>Ilustración 17: Inventario de pantalla</i>	124
<i>Ilustración 18: Inventario de pantalla</i>	125
<i>Ilustración 19: Inventario de pantalla</i>	125
<i>Ilustración 20: Inventario de pantalla</i>	126
<i>Ilustración 21: Inventario de pantalla</i>	126
<i>Ilustración 22: Inventario de pantalla</i>	127
<i>Ilustración 23: Modelado de datos</i>	127
<i>Ilustración 24: Modelado de datos</i>	128
<i>Ilustración 25: Modelado de datos</i>	128

Índice de tablas

Tabla 1 Requerimientos de Software.....	26
Tabla 2 Componentes de Hardware	27
Tabla 3 Detalle de actividades.....	41
Tabla 4 Información general del proyecto.....	43
Tabla 5 Recursos Involucrados en el Proyecto	45
Tabla 6 Documentos Técnicos Proyecto	47
Tabla 7 Principales Hitos y Entregables	49
Tabla 8 Delegación de Responsabilidades.....	50
Tabla 9 Vista de Casos de Uso	52
Tabla 10 Plataforma.....	58
Tabla 11 Descripción General del Programa	59
Tabla 12 Inventario de clases Cajacontroller.....	63
Tabla 13 Inventario de clases CierreTemporalEstacionController	65
Tabla 14 Inventario de clases AperturaDeEstacionController	67
Tabla 15 Inventario de clases CapturaTarjetasController	69
Tabla 16 Inventario de clases CierreEstacionFinDiaController	73
Tabla 17 Inventario de clases CapturarConcentracionTombolaController	76
Tabla 18 Inventario de clases SolicitudDeDotacionBilleteController	79
Tabla 19 Inventario de clases SolicitudDeDotacionBilleteBusinessImpl	81
Tabla 20 Inventario de clases SolicitudDeDotacionDeBilleteBusiness	83
Tabla 21 Inventario de clases EliminacionDeSobrantesDeCajaController	85
Tabla 22 Inventario de clases FaltantesPorRoboAsaltoController.....	88
Tabla 23 Inventario de clases SobrantesDeCajaController	90
Tabla 24 Inventario de clases DevolucionACajaEnGeneralController.....	91
Tabla 25 Inventario de clases DesgloceDeMonedaNacionalModalController.....	96
Tabla 26 Inventario de clases DotacionDeCajaGeneralController	100
Tabla 27 Inventario de clases TotalesCifrasSucursalController	103
Tabla 28 Inventario de clases SolicitudDeDotacionMorrallaController	104
Tabla 29 Inventario de clases PagoDeBilleteAutenticoController	107
Tabla 30 Inventario de clases PagoDeBilleteAutenticoController	110
Tabla 31 Inventario de clases ReimpresionDeTicketController.....	113
Tabla 32 Inventario de clases EliminacionFaltantesDeCajaController	115

AGRADECIMIENTOS

A:

Dios, por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

Mi madre Marta Rojas, por haberme dado la vida, por su amor incondicional, por siempre creer en mí y por ser el pilar más importante en mi vida. Mamá gracias por darme una carrera para mi futuro, todo esto te lo debo a ti.

Mi padre Arturo Reyes, por los ejemplos de perseverancia y constancia que lo caracterizan, sé que este momento es tan especial para ti como lo es para mí.

Mi hermana Brenda Reyes, porque tu afecto y cariño son los detonantes de mi felicidad, de mi esfuerzo, de mis ganas de buscar lo mejor para ti, por ser mi compañía y mi apoyo.

Mis primos y amigos, Miguel, Enrique, Abel, Beatriz, Fabiola, Daniel, Sra. Luz, Eliana, Gilberto, Mar, Juan P. y todas aquellas personas con las que compartí buenos y malos momentos, por brindarme su amistad, su apoyo y trasmitirme sus buenos deseos.

Finalmente, a mis maestros, quienes se tomaron el arduo trabajo de trasmitirme sus diversos conocimientos, agradezco el tiempo dedicado para aclarar cualquier tipo de duda que surgiera y por la caridad y exactitud que con la que impartieron cada clase.

Fredy Arturo Reyes Rojas

RESUMEN

La presente tesis tiene como objetivo gestionar el proyecto “Cambio de plataforma de sucursales a web” para las sucursales de un cliente de la empresa Solser Sistem en la ciudad de Querétaro. Dicho proyecto se desarrolla de acuerdo a OLAS que son los niveles y etapas de madures del proyecto.

La metodología que se utilizó para obtener los resultados deseados, consiste en una metodología Ágil la cual se crea con el objetivo de satisfacer las necesidades de nuestros clientes y se utiliza para minimizar los riesgos durante la realización de un proyecto. Esta metodología Ágil de Solser también interviene en las necesidades de nuestros clientes en cuanto a la modernización y reingeniería de plataformas tecnológicas en software, y por consiguiente en temas de hardware en que están operando las empresas.

En el primer capítulo se describe el entorno y requerimientos del proyecto. Se plantea el estado del arte, el planteamiento del problema, los objetivos generales como específicos, la hipótesis, justificación del proyecto, limitaciones y alcances, así como datos relevantes de la empresa. El segundo capítulo refiere a la metodología que se utilizó para cumplir con los objetivos del proyecto. El tercer capítulo expone el desarrollo del proyecto en el cual se redacta el diseño, la elaboración y la integración de todos los elementos, es aquí donde se dan a conocer los distintos tipos de pruebas y técnicas de análisis necesarias para determinar una posible evaluación al final del proyecto. En el cuarto y último capítulo presentan los resultados y conclusiones que se llegó con este estudio, se responde el objetivo general y se redactan las recomendaciones futuras para la mejora del proyecto.

CAPÍTULO 1. INTRODUCCIÓN

El presente documento contiene la gestión del proyecto CPlasu Cambio de Plataforma de Sucursales, el cual es un producto de un análisis minucioso de los requisitos del sistema, según estos pueden ser satisfechos con las tecnologías y características discutidas con los clientes y usuarios.

El objetivo del documento es realizar la gestión del proyecto CPlasu, que consiste en realizar la migración de las 350 funcionalidades que actualmente existen en las sucursales del cliente a un ambiente web que permita una optimización de las funcionalidades y mantener un ambiente preparado para la integración de nuevos módulos. Permitiendo que la aplicación proporcione de manera íntegra la comunicación con los diferentes dispositivos (Periféricos) que se utilizan actualmente en las 910 sucursales para las operaciones diarias como digitalización de documentos y/o cheques, lectoras de tarjetas de crédito o débito mismo que debe interactuar con la aplicación

1.1 Estado del Arte

María Elena Ciolli (Diciembre de 2007) señala que;

La aplicación de la tecnología computacional a lo largo de ya varias décadas ha conducido a sistemas de información grandes y complejos, muchos de los cuales parecen haber alcanzado un punto límite en sus posibilidades de adecuación y evolución. Si bien este problema viene siendo advertido desde hace mucho tiempo, el crecimiento exponencial de la Web y la necesidad de mantener competitividad en un mundo dinámico y cambiante le dieron al problema una trascendencia adicional. En efecto, las organizaciones deben hacer un efectivo aprovechamiento del potencial que ofrece la Web en plazos perentorios, pero al mismo tiempo, gran parte de su conocimiento, experiencia y reglas del negocio están concentrados en estos sistemas de información que parecen inmunes a su adaptación.

Por esta razón, durante los últimos años se comprueba un creciente número de experiencias de migraciones a la Web de aplicaciones distribuidas no basadas en la Web, convirtiéndose este tema en un importante campo de investigación y práctica profesional de la Ingeniería de Sistemas.

Es así que se vienen presentando y ensayando diferentes propuestas, se desarrollaron productos y se dispone de amplia bibliografía para la migración de sistemas apoyados en plataformas mainframes tradicionales a entornos abiertos. Sin embargo, no hay tanta información referida a la realización de las pruebas de migración y mucho menos sobre las características de las validaciones de las aplicaciones en entornos distribuidos sobre la Web.

Esta comprobación sirvió de estímulo para el trabajo que se presenta, en el que se aborda el problema de migración de sistemas a la Web desde el enfoque de testing, estudiando en especial las especificaciones de requerimientos de interfaces con el usuario y funcionales.

El trabajo realizado se apoya en el conocimiento previo sobre testing de regresión, testing de caja negra y testing de interfaces gráficas con el usuario. Asimismo, dado que el ciclo de vida del testing está embebido dentro del ciclo de vida del software, el enfoque propuesto promueve la reutilización de casos de pruebas existentes, resultantes de la trazabilidad con los casos de uso planteados para el sistema distribuido, y la automatización de las pruebas unitarias, de integración y de regresión.

(Pág. 3)

1.2 Planteamiento del Problema

La gestión del proyecto acarrea una serie de riesgos que se generan en la creación y ciclo de vida del mismo. El proyecto CPlasu no tiene una visión del estado real del proyecto y por ende de su evolución.

El desarrollo de la fase OLA 4 (Ventanilla/Sucursales) no cuenta con normas de gestión de proyectos, esto genera retraso en la entrega de avances al cliente.

Actualmente las funcionalidades de Sucursales y plataforma del sistema están desarrolladas bajo ambiente Stand-alone y con tecnologías obsoletas de Visual Basic, lo que implica un mayor esfuerzo al momento de implementar una actualización de las funcionalidades o la integración de nuevos módulos para las sucursales.

1.3 Objetivos

Gestionar el proyecto CPlasu que consta de la migración de las 350 funcionalidades que actualmente existen en las sucursales a un ambiente Web que permita una optimización de las funcionalidades y mantener un ambiente preparado para la integración de nuevos módulos.

Objetivos específicos

- Generar una propuesta técnica y dar a conocer al cliente
- Elaborar plan de trabajo
- Realizar arquitectura de la aplicación
- Realizar inventario de clases
- Realizar la reestructuración de las pantallas
- Realizar las comparaciones necesarias para la ejecución de las labores del proyecto (Matriz de Trazabilidad).

1.4 Definición de variables

Problema: El Sistema actual esta desarrolladas bajo ambiente Stand-alone y con tecnologías obsoletas de Visual Basic, lo que implica un mayor esfuerzo al momento de implementar una actualización de las funcionalidades o la integración de nuevos módulos para las sucursales.

Variable independiente: (Causa) Sistema obsoleto.

Variable dependiente: (Efecto) Un mayor esfuerzo al momento de implementar nuevos módulos.

Variable interviniente: Sistema desarrollado bajo un ambiente Stand-alone y con tecnologías obsoletas de Visual Basic.

1.5 Hipótesis

La elaboración de este Software CPlasSu (Cambio de plataforma de sucursales) se basó en la siguiente hipótesis:

Es posible gestionar un proyecto de software donde, su contenido, su interfaz y sus funciones se modernicen, haciendo un cambio a la ingeniería tecnológica actual del software y por consecuencia el hardware.

Gestionando la etapa 4 dicha OLA se encuentra en el grupo de elaboración durante el ciclo de vida del proyecto.

1.6 Justificación del Proyecto

Con la finalidad de alcanzar el objetivo de esta tesis, por medio del modelo de gobierno se gestiona el proyecto CPlasSu, este mismo consta de la Planificación de los procesos que comienza con una misión, metas y objetivos que deben lograrse. Así mismo se Desarrollan planes, procedimientos, se establece una organización y asigna recursos y responsabilidades con el propósito de alcanzar los objetivos propuestos.

Por medio de la arquitectura y las tecnologías propuestas que se integran en la implementación de la solución, se brinda la facultad para hacer uso de todos los requerimientos del cliente. Permitiendo

de esta manera la integración funcional dentro de la misma gestión para las operaciones que requieran la utilización de algunos componentes para poder concluirla.

1.7 Limitaciones y Alcances

El planteamiento general de esta propuesta está fundamentado en el requerimiento de realizar una gestión de la modernización y reingeniería de la plataforma tecnológica actual de Software y por consecuencia del hardware en que están operando las sucursales y unidades de negocio del cliente, la propuesta implica la implementación de componentes de software operando en una plataforma 100% WEB centralizada. En forma enunciativa, esta propuesta en su parte fundamental propone el siguiente alcance:

- Plantea la construcción y diseño de una aplicación WEB, componentes de software y middleware fundamentada en una arquitectura y un modelo de 3 capas lógicas, como las buenas prácticas generales lo marcan.
- Plantea mantener las interfaces de comunicación e intercambio de datos y transacciones actuales con organismos o instituciones externas mediante la utilización de componentes de software que las integren, administren y provean elementos de seguridad y rastreabilidad (log's), de las operaciones que pasan por estas interfaces.
- Plantea mantener y fortalecer los canales seguros de comunicación hacia la base de datos central, manteniendo el uso del InterAct (middleware actual), en todos sus ámbitos e integrarlos en una plataforma única.
- Plantea dentro de la estrategia de modernización, la migración de las reglas de negocio que se encuentran actualmente dentro del sistema Cliente hacia el motor de Base de Datos (en lenguaje de SP's), donde residen actualmente la mayoría de éstas, y así separar completamente la capa de negocio de la de presentación.
- El diseño e implementación de una nueva Plataforma WEB Centralizada, encaminada a la modernización de las funcionalidades actuales, mejora ergonómica y funcional.
- En lo que a herramientas de desarrollo de software respecta, esta propuesta está basada en el uso de Java, ya que es uno de los lenguajes con mayor compatibilidad y portabilidad. Al desarrollar la solución para el banco en una versión específica de Java, se garantizará que ésta funcione en las versiones posteriores del lenguaje sin problemas. En otras palabras,

Java es una multiplataforma y no introduce cambios rupturistas y drásticos en las nuevas versiones.

Se consideran 5 grandes módulos que contienen 350 funcionalidades, los cuales son los siguientes:

1. Plataforma
2. Ventanilla
3. Gestión de Sucursal
4. Seguridad
5. Catálogos

Nuestra propuesta cumple con los puntos del alcance mencionado anteriormente, así mismo contempla la migración de 350 funcionalidades (que se describen en el apartado 3.4 Funcionalidad incluida) que están desarrolladas actualmente con tecnología de Visual Basic, a una plataforma web, desarrollada bajo la tecnología de Java 1.8, con un modelo de arquitectura en 3 capas que soporte la instalación y utilización de periféricos, así mismo que sea compatible con una arquitectura de hardware para altos volúmenes de transacciones.

Nuestra solución también incluye la estrategia de liberación, soporte y puesta a punto en producción.

El proyecto se desarrollará en base a la metodología ágil SCRUM como lo propone el cliente y además, complementar la documentación del proyecto con algunos artefactos de la metodología SOLSER basada en CMMI nivel 3 para desarrollo (versión 1.3).

El proyecto se desarrollará utilizando los estándares de desarrollo de SOLSER verificados por el cliente.

1.8 Solser Sistem S.A. de C.V.

1.8.1 Historia de la empresa

Solser Information Technology es una compañía certificada en CMMi NIVEL 3 y experiencia en Metodología SCRUM. Contamos con más de 14 años de experiencia generando soluciones con tecnologías innovadoras en desarrollo de software en México y Sudamérica.

Brindamos consultoría en el área de Tecnologías de la Información (TI), desarrollando e integrando software para negocios, dando mantenimiento a aplicaciones tecnológicas y proporcionando outsourcing especializado bajo los mejores estándares de calidad que existen en el mercado, ofrecemos soluciones para Entidades Financieras, Empresas Privadas, Aseguradoras, Industrias y Gobierno.

Solser Information Technology nace en el año 2002, con la visión de convertirnos en una empresa exportadora de soluciones en Tecnología de Información hecha en México hacia el mundo. Nuestro Centro de Soluciones actualmente se ubica en la ciudad de Querétaro con una producción al año de más de 100,000 hrs y con capacidad para 180,000 y contamos con oficinas comerciales en la Ciudad de México y Cancún Quintana Roo.

1.8.2 Misión

Ser de las primeras empresas en brindar soluciones que ayuden a desarrollar e incrementar los negocios de nuestros clientes, a través del potencial de nuestros profesionales, la calidad y la innovación tecnológica, desde México para el Mundo.

1.8.3 Visión

En SOLSER buscamos consolidarnos como líderes en el mercado de Soluciones Tecnológicas en México y otros países, reconocidos por nuestra experiencia y compromiso en brindar soluciones tecnológicas de alto valor para hacer crecer los negocios de nuestros clientes, basados en las mejores prácticas de calidad.

1.8.4 Objetivo

Brindar consultoría en el área de tecnologías de la información, desarrollando e integrando software para negocios, dando mantenimiento a aplicaciones tecnológicas y proporcionar outsourcing bajo los mejores estándares de calidad existentes en el mercado.

1.8.5 Valores

Los valores corporativos hacen referencia a las virtudes mínimas que deberá poseer todo aquel que permanezca a la organización, de igual forma la empresa en sí, desde todos los niveles debe velar por el cumplimiento de los mismos, en todas y cada una de las actividades diarias, dichos valores son:

- Responsabilidad y Compromiso: Cumplir con los objetivos de nuestros clientes es nuestra meta para alcanzar el éxito.
- Dinamismo e integridad: Creemos en la capacidad de lograr los más grandes retos, basados en el esfuerzo, la disciplina y la honestidad.
- Rectitud: Nos comprometemos a hacer siempre lo mejor que podemos basados en el principio de que cumplir con el clientes es lo más importante.
- Respeto: Tratamos con respeto a las personas dentro y fuera de la empresa, porque sabemos que es la base de la confianza en todo lo que hacemos.
- Actitud positiva y trabajo en equipo: En SOLSER creemos que tenemos la capacidad de pensar, hacer y lograr que todo sea posible a través del trabajo en equipo.

1.8.6 Políticas de la empresa

1.8.6.1 Cumplimiento a visión, misión y valores

Todos los recursos de Solser Sistem, tienen que participar en el cumplimiento de la Misión, la Visión y los Valores de la empresa. Cualquiera desviación al respeto se debe reportar a Jefe Inmediato, Recursos Humanos.

1.8.6.2 Participación en iniciativas organizacionales

- Es obligatorio para los recursos que participen en cualquiera iniciativa organizacional

- En el caso de que algún colaborador no pueda o no le sea posible participar en la o las iniciativas organizacionales deberá contar con Bobo del Coordinador de área, Líder de Línea o Jefe inmediato.
- Solser Sistem promoverá la capacitación y el entrenamiento de sus recursos desarrollando acciones y programas con ese fin de acuerdo a las necesidades identificadas por los Líderes de Proyecto, Líderes de Línea o Coordinadores de Área.
- El personal de Solser Sistem debe participar en los programas de capacitación o entrenamiento para los que sean seleccionados.
- El recurso, con autorización de su jefe inmediato y sujetándose a las disposiciones y limitaciones que rijan en Sol Ser Sistem puede asistir dentro del horario de trabajo, a eventos y clases de capacitación, siempre que tenga que ver sobre temas que sean de interés institucional. En caso de que sean de otro carácter deberá contar con autorización de jefe inmediato.
- Todos los recursos deben seguir la Metodología definida por Solser Sistem (QMS) al pie de la letra para llevar a cabo las actividades de sus proyectos o áreas correspondientes.

1.8.6.3 Uso de estacionamiento

- El personal de seguridad tiene prohibido el movimiento de automóviles.
- El estacionamiento es para uso exclusivo de los recursos y usuarios externos.
- Los lugares de estacionamiento son autorizados y asignados únicamente por Recursos Humanos y Capacitación
- Todo el personal de Solser Sistem así como proveedores y visitantes deberán respetar los cajones de estacionamiento, ocupando un solo cajón para su vehículo.
- Solser Sistem no asume responsabilidad alguna por daños, deterioros, accidentes o robos que puedan ocurrir a los vehículos estacionados en sus instalaciones.

Para faltas recurrentes Recursos Humanos y capacitación resolverá y determinará la sanción la cual puede llegar a derivarse en la rescisión del contrato de trabajo.

1.8.6.4 Salario y viáticos

1.8.6.4.1 Pago de nómina

- El pago de nómina se llevará cabo cada fin mes este puede ser un depósito o bien un cheque a nombre del recurso.

- Si el recurso expide factura a Solser Sistem este deberá entregar la misma al Responsable de Contabilidad y Finanzas 5 días antes finalizar el mes de lo contrario el pago no será efectuado hasta el siguiente mes.

1.8.6.4.2 Reembolso de viáticos

El recurso deberá elaborar el FCF-001 Formato Gastos le cual deberá imprimir y firmar y enviarlo a Responsable de Contabilidad y Finanzas. El reembolso se reflejará en el siguiente pago de su salario mensual, este puede ser en depósito o bien en efectivo.

1.8.7 Organigrama general

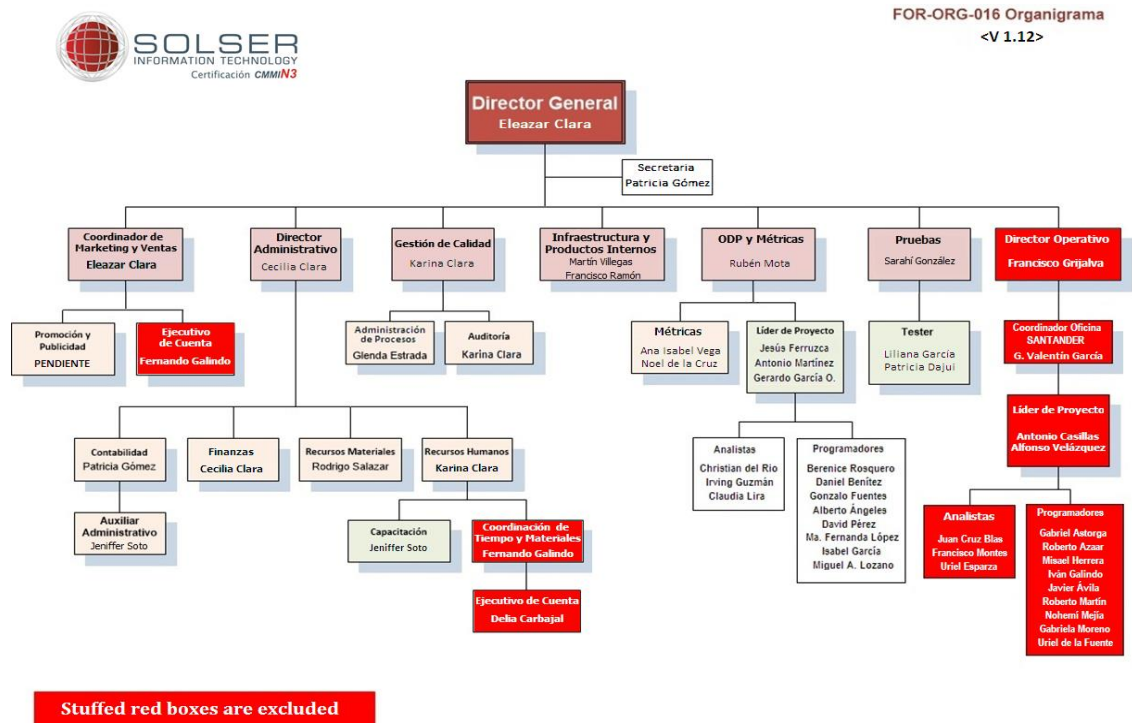


Ilustración 1: Organigrama general Solser Sistem

1.8.8 Organigrama del área CMMi

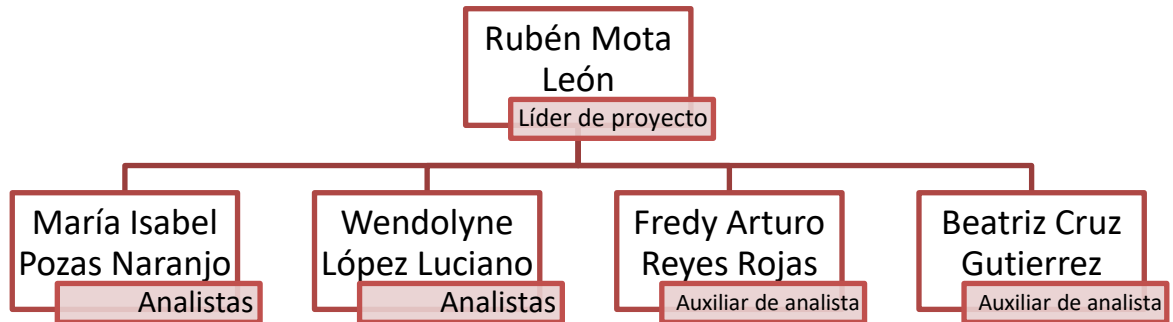


Ilustración 2: Organigrama CMMi

1.8.9 Procesos que se realizan en la empresa

Outsourcing

- De personas especializadas en IT y SAP

Productos

- Administración de contratos
- Facturas electrónicas
- Punto de venta

Consultoría

- Procesos bancarios
- Procesos de aseguradoras
- Procesos de financieros
- Procesos de ventas
- Ingeniería de Requerimientos (PMI)
- Arquitectura de aplicaciones

Infraestructura

- Soporte preventivo y correctivo
- Hosting integral de aplicaciones
- Call Center
- Help Desk

1.8.10 Mercado de impacto de los productos o servicios brindados por la empresa

Solser Sistem es una fábrica de desarrollo de software nivel internacional que utiliza los más altos estándares de calidad con lo cual se garantiza que los productos son de nivel internacional a costo nacional.

El principal Core Bussines está enfocado en el Sector Financiero, pero de manera paralela actualmente Solser Sistem ha crecido en otros sectores como el Automotriz, petrolero, Metalúrgico, sector Retail, Recursos Humanos y manufactura.

10.8.11 Impacto en el área de tecnologías de la información y comunicación

Solser Sistem siendo una empresa dedicada a la consultoría en el área de Tecnologías de la Información (TI), desarrollando e integrando software para negocios tiene un gran impacto en las TI ya que entre sus principales productos y servicios se encuentran:

Facturación electrónica

Solución completa de facturación electrónica WEB, diseñada para el procesamiento de volúmenes grandes de facturas, lo que garantiza el rápido y fácil procesamiento para cualquier tamaño de empresa.

Administración de contratos

Sistema de gestión de contratos integral, perfecto para cualquier empresa que busque una mejor organización y gestión en el proceso de formalización de un contrato. Sus características de gran

alcance, flexible, intuitivo y fácil de usar hacen de la gestión de contratos de una empresa sea una tarea sencilla.

Punto de venta

Punto de venta dinámico integrado con herramientas que permiten incrementar las ventas.
Gestión de personal y centros de distribución con configuraciones independientes.

Control de inventarios centrales y de las tiendas que le permitirá conocer de forma oportuna y detallada su Stock.

CAPÍTULO 2. METODOLOGÍA

2.1 Metodología SCRUM Solser

En SOLSER la Metodología Ágil se crea con el objetivo de satisfacer las necesidades de nuestros clientes y se utiliza para minimizar los riesgos durante la realización de un proyecto, pero de manera colaborativa a través de una respuesta ágil y rápida pero sin perder de vista la importancia y calidad del producto.

La metodología Ágil de SOLSER también interviene en las necesidades de nuestros clientes en cuanto a la modernización y reingeniería de plataformas tecnológicas en software, y por consiguiente en temas de hardware en que están operando las empresas.

Nuestro método tiene la capacidad de implementar componentes de software que puedan operar en una plataforma 100% WEB centralizada que pueda modernizar las funciones actuales, eficiencia ergonómica y funcionalidad.

Nuestra Metodología está basada en el Método SCRUM de gestión por iteraciones y está fundamentada en 4 puntos importantes:



Ilustración 3: Iteraciones metodología SCRUM

2.2 Roles método SCRUM

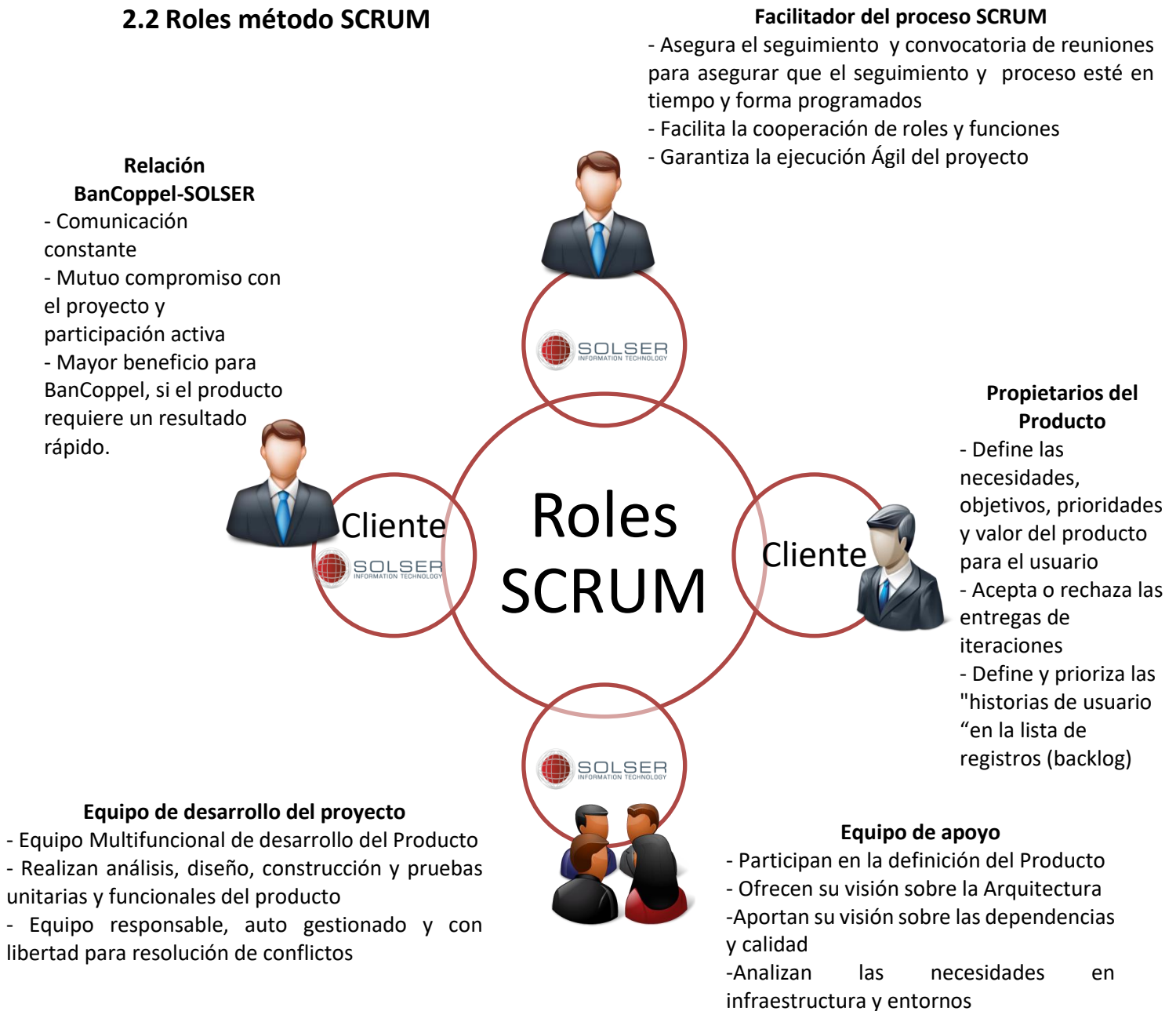


Ilustración 4: Roles SCRUM

2.3 Gestión de fases del proyecto

Nuestra Metodología está basada en el Método SCRUM de gestión por iteraciones el cual lleva un proceso finamente estructurado, contemplando el ciclo de vida del proyecto de manera segura, ágil y eficaz, favoreciendo así el incremento considerable en la productividad y calidad de los servicios que el cliente ofrece.

La metodología SOLSER está alineada perfectamente a los requerimientos de gestión del proyecto de acuerdo a nuestra certificación CMMI Nivel 3 para desarrollo, por lo tanto cubre perfectamente con las siguientes fases

- **INICIO** : Conformación del equipo de trabajo, lista de pendientes, planificación del lanzamiento
- **PLANEACIÓN Y ESTIMACIÓN**: definición de actividades, estimación de trabajo, creación de historias de usuario.
- **IMPLEMENTACIÓN** : Reunión diaria de 15 min, creación de entregables, actualización de lista de pendientes
- **REVISIÓN Y RETROSPECTIVA**: validación de sprint y retrospectiva
- **LANZAMIENTO**: Envío de entregables y retrospectiva del proyecto



Ilustración 5: Fases del proyecto

2.4 Uso de metodología SCRUM

El alcance de la propuesta abarca el ciclo de vida ágil de un proyecto.

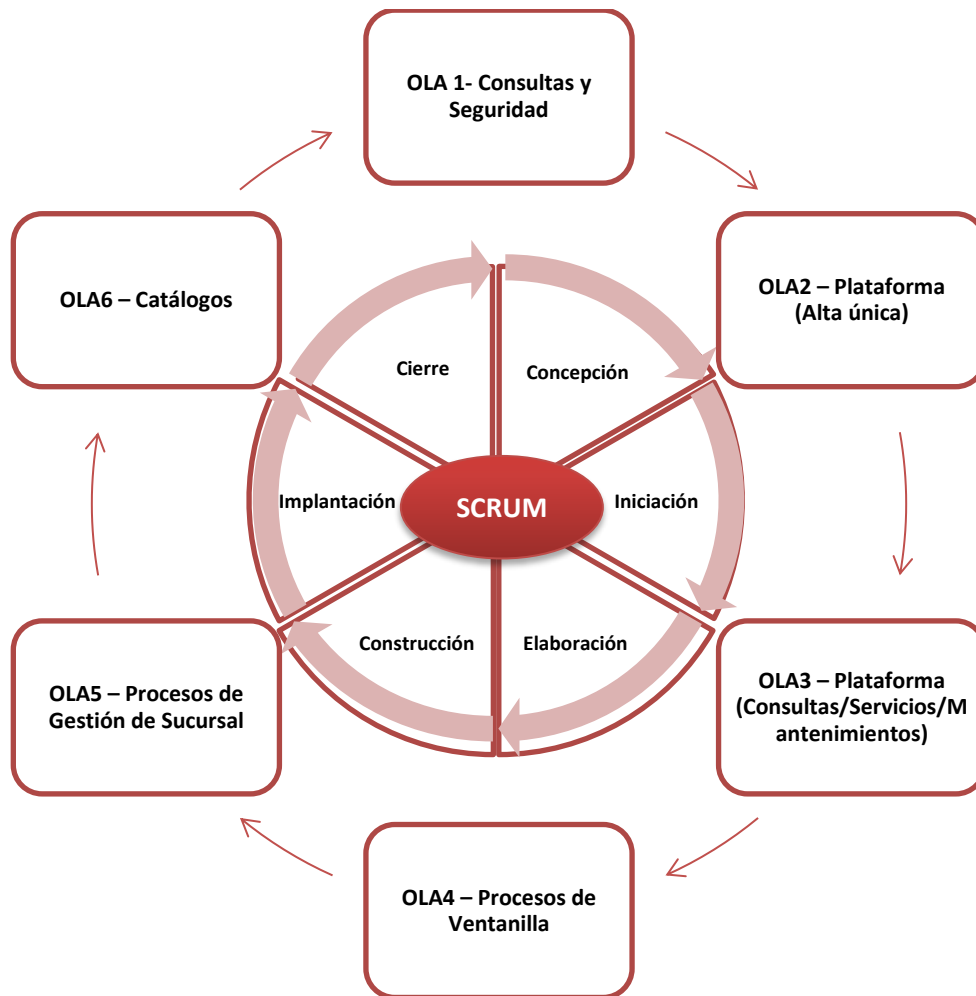


Ilustración 6: Metodología SCRUM

El contenido de este capítulo se centra en todos los pasos a seguir para el desarrollo del proyecto, llevándolo de la mano con sus alcances y objetivos. Es posible utilizar imágenes para apoyar la descripción de su metodología.

Además, describe las técnicas (cualitativas y/o cuantitativas) y metodologías empleadas para la elaboración de la tesina. Debe informar el lugar, tiempo en el que será realizado el estudio y los elementos de estudio (sujetos y/u objetos de estudio), así como la cantidad de elementos de estudio (tamaño de la muestra).

2.5 Gestión del modelo de gobierno del proyecto

Para la gestión de gobierno, la metodología ágil de SOLSER propone hacer uso de la herramienta JIRA la cual permite llevar el control de las actividades de todo el desarrollo de manera controlada y con el uso de Sprints.

El software de JIRA está diseñado para que todos los miembros de tu equipo de software puedan planificar, realizar seguimientos y publicar un magnífico software.

PROCESO:

Planificación: Crea historias de usuario e incidencias, planifica sprints y distribuye tareas entre tu equipo de software.

Realiza un seguimiento: Prioriza y analiza el trabajo de tu equipo en su contexto y con una completa visibilidad.

Lanzamiento: Entrega con confianza y seguridad, sabiendo que la información que tienes es siempre la más actualizada.

Crea informes: Mejora el rendimiento del equipo con datos visuales en tiempo real de gran utilidad.

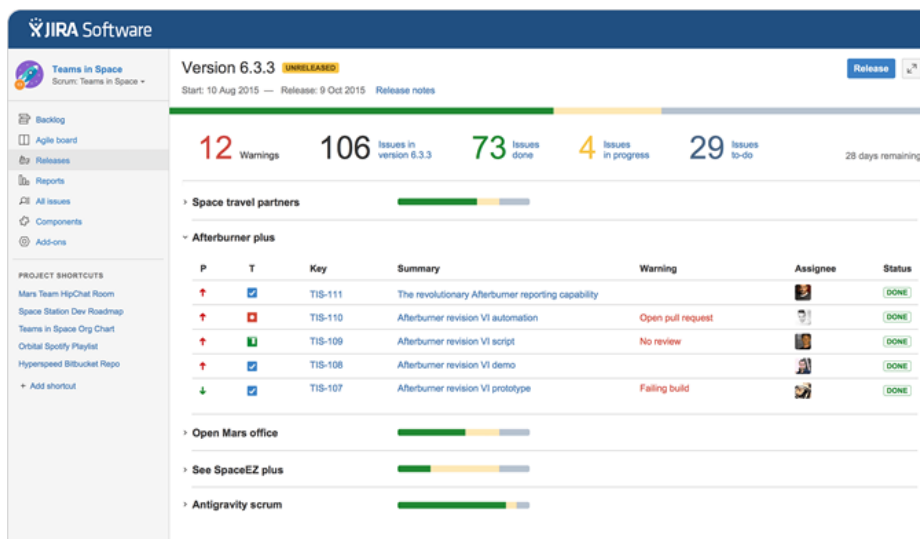


Ilustración 7: Herramienta Jira Software

CAPÍTULO 3. DESARROLLO DEL PROYECTO

En este capítulo se exponen los entregables de la gestión del proyecto. Las actividades y decisiones importantes tomadas en cuanto a cuestiones de implementación mostrando la estructura del proyecto de acuerdo a sus etapas. También se hace referencia a los problemas surgidos y cuáles han sido las soluciones.

3.1 Estructura del proyecto

En este apartado se explica cómo se ha implementado la estructura del proyecto, en la cual se gestiona el contenido del sistema (CPlaSu). Se muestra cómo se organiza la gestión de los datos de acuerdo a las etapas de la metodología.

3.1.2 Estructura de directorios

Al inicio del proyecto se pensó en SVN como el entorno de la gestión del proyecto ya que este transmite menos información a través de la red y hace que el funcionamiento sea más rápido. Pero este a su vez traía distintas complicaciones para esta gestión, entre estas el cambio de nombre de archivos o directorios, borrado de archivos, almacenamientos de copias locales y pequeños problemas de codificación de acentos.

Más adelante se consideró la posibilidad de utilizar SmartCVS y se decidió cambiar a este entorno para implementar toda la gestión del proyecto. SmartCVS brinda una visión general óptima del proyecto, ya que se puede visualizar los archivos que se han modificado y de que archivos existen nuevas versiones en el repositorio. Esto ayuda a detectar posibles conflictos antes de que ocurran.

La estructura del directorio queda de la siguiente manera:

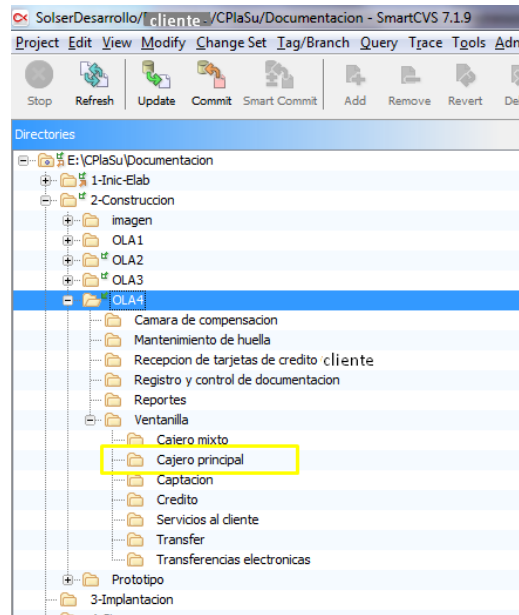


Ilustración 8: Estructura de directorios

Toda la estructura cuelga de la carpeta con el nombre del proyecto (CPlaSu). Dentro de esta carpeta se encuentra otra carpeta llamada “Documentación” la cual contiene las fases del proyecto, así como de sus Sprints y entregables de cada etapa.

3.2 Construcción

PROPUESTA TÉCNICA

Resumen Ejecutivo

A continuación, se presenta la propuesta técnica para la solución del requerimiento solicitado, el cual consiste en realizar el **Cambio de Plataforma Sucursales Web**.

Antecedentes

Actualmente las funcionalidades de Sucursales y plataforma BanCliente están desarrolladas bajo ambiente Stand-alone y con tecnologías obsoletas de Visual Basic, lo que implica un mayor esfuerzo al momento de implementar una actualización de las funcionalidades o la integración de nuevos módulos para las sucursales.

Objetivo

Realizar la migración de las 350 funcionalidades que actualmente existen en las sucursales de BanCoppel a un ambiente Web que permita una optimización de las funcionalidades y mantener un ambiente preparado para la integración de nuevos módulos. Permitiendo que la aplicación proporcione de manera íntegra la comunicación con los diferentes dispositivos (periféricos) que se utilizan actualmente en las 910 Sucursales para las operaciones diarias como digitalización de documentos y/o cheques, lectoras de tarjetas de crédito o débito mismos que deben interactuar con la aplicación.

Alcance

El planteamiento general de esta propuesta está fundamentado en el requerimiento de realizar una modernización y reingeniería de la plataforma tecnológica actual de Software y por consecuencia del hardware en que están operando las sucursales y unidades de negocio de BanCliente, la propuesta implica la implementación de componentes de software operando en una plataforma 100% WEB centralizada. En forma enunciativa, esta propuesta en su parte fundamental propone el siguiente alcance:

- Plantea la construcción y diseño de una aplicación WEB, componentes de software y middleware fundamentada en una arquitectura y un modelo de 3 capas lógicas, como las buenas prácticas generales lo marcan.
- Plantea mantener las interfaces de comunicación e intercambio de datos y transacciones actuales con organismos o instituciones externas mediante la utilización de componentes de software que las integren, administren y provean elementos de seguridad y rastreabilidad (log's), de las operaciones que pasan por estas interfaces.
- Plantea mantener y fortalecer los canales seguros de comunicación hacia la base de datos central, manteniendo el uso del InterAct (middleware actual), en todos sus ámbitos e integrarlos en una plataforma única.
- Plantea dentro de la estrategia de modernización, la migración de las reglas de negocio que se encuentran actualmente dentro del sistema Cliente hacia el motor de Base de Datos (en lenguaje de SP's), donde residen actualmente la mayoría de éstas, y así separar completamente la capa de negocio de la de presentación.
- El diseño e implementación de una nueva Plataforma WEB Centralizada, encaminada a la modernización de las funcionalidades actuales, mejora ergonómica y funcional.
- En lo que a herramientas de desarrollo de software respecta, esta propuesta está basada en el uso de Java, ya que es uno de los lenguajes con mayor compatibilidad y portabilidad. Al desarrollar la solución para el banco en una versión específica de Java, se garantizará que ésta funcione en las versiones posteriores del lenguaje sin problemas. En otras palabras, Java es una multiplataforma y no introduce cambios rupturistas y drásticos en las nuevas versiones.

Se consideran 5 grandes módulos que contienen 350 funcionalidades, los cuales son los siguientes:

1. Plataforma
2. Ventanilla
3. Gestión de Sucursal

4. Seguridad
5. Catálogos

Nuestra propuesta cumple con los puntos del alcance mencionado anteriormente, así mismo contempla la migración de 350 funcionalidades (que se describen en el apartado 3.4 Funcionalidad incluida) que están desarrolladas actualmente con tecnología de Visual Basic, a una plataforma web, desarrollada bajo la tecnología de Java 1.8, con un modelo de arquitectura en 3 capas que soporte la instalación y utilización de periféricos, así mismo que sea compatible con una arquitectura de hardware para altos volúmenes de transacciones.

El proyecto se desarrollará en base a la metodología ágil SCRUM como lo propone BanCliente y además, complementar la documentación del proyecto con algunos artefactos de la metodología SOLSER basada en CMMI nivel 3 para desarrollo (versión 1.3).

Referencias

Para llevar a cabo un desarrollo ágil de las necesidades planteadas se hará uso de la metodología de SCRUM y complementada con la metodología SOLSER CMMI nivel 3.

- Metodología SCRUM (Anexo 1 metodología ágil SOLSER)
- Metodología SOLSER CMMI nivel 3 versión 1.3

Premisas

- Que el BUS funcione adecuadamente durante la etapa de implementación a pruebas.
- Migración de 350 funcionalidades desarrolladas en Visual Basic a un ambiente web desarrollado bajo la tecnología de Java.
- Las funcionalidades tendrán mejor desempeño al estar implementadas bajo la tecnología Java

Tecnologías Propuestas y Arquitectura Tecnologías Software y Hardware:

Tipo de aplicación	WEB
Lenguaje de Programación	Java 1.8

Servidor de funcionalidades	JBoss EAP 6
	SFTP
	RedHat Linux Enterprise Linux 7.
	Base de Datos Distribuida
	JBoss Fuse Works
Tecnologías utilizadas	Java FX
	Spring
	HTML
	JavaScript
	Certificado De firma de Código Java
	JIRA
	NGINX
Navegadores soportados	Internet Explorer 7 en adelante, Chrome, FireFox y Safari.
Sistemas operativos	Windows a partir de XP y Linux
Periféricos	Soporte de periféricos utilizados por BanCoppel

Tabla 1 Requerimientos de Software

OBSERVACIÓN: Para la siguiente tecnología, es necesario adquirir una licencia que tiene un costo no incluido en esta propuesta. Esta tecnología permitirá que las aplicaciones Java que se ejecuten desde un navegador, tengan permiso de hacer uso de los periféricos:

- Certificado de Firma de Código Java

Componentes de Hardware

Sucursales	8 Equipos clientes
Dispositivos actuales	Multifuncionales -Impresoras/Escáner-, Digital persona, Pin Pad, LS-100, Check Reader, Teclado alfanumérico, Impresora de ticket, IdBox
Servidores	Clúster de Balance de cargas

Dos Clústers con ESB y servidores de aplicaciones
Servidor SFTP
Cluster de Base de Datos

Tabla 2 Componentes de Hardware

Diagrama de la Arquitectura Propuesta.

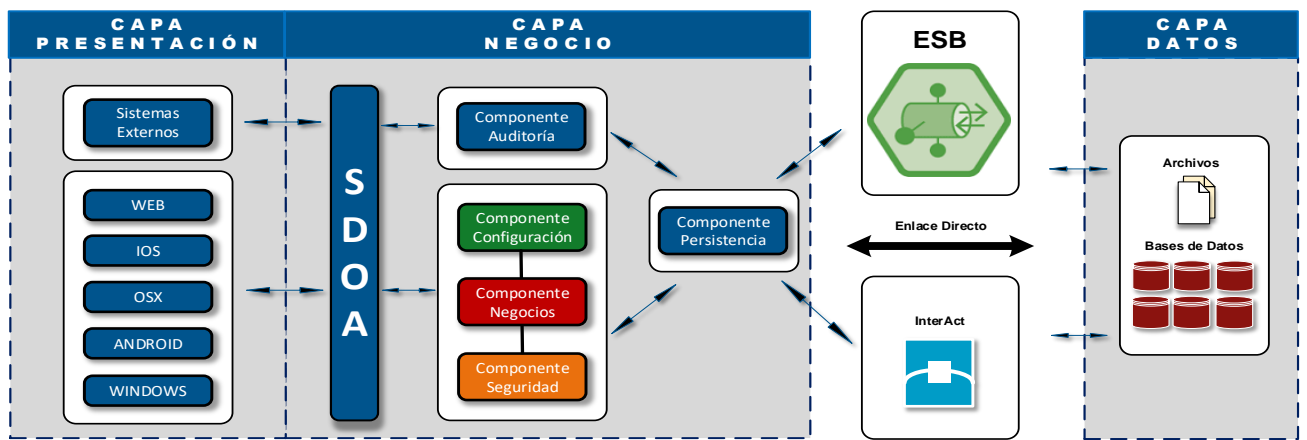


Ilustración 9: Diagrama de arquitectura propuesta

Dentro de los componentes integrados de la arquitectura se encuentran:

- **Componente de Presentación:** Es el conjunto de interfaces visuales que conforman la aplicación.
- **Componente de Negocio:** Servicios que incorporan validaciones y reglas del negocio, para éste último, sólo si la Institución lo requiere.
- **Componente Persistencia:** Componente de comunicaciones hacia el repositorio de datos (Middleware Bancliante), se encarga de la transformación de las tramas a Objetos

(Beans).

- **Componente de Configuración:** Se encarga de mantener y distribuir los parámetros de configuración entre las diferentes capas y componentes del sistema.
- **Componente de Seguridad:** Es un conjunto de herramientas para la autenticación de los usuarios dentro del sistema, validación de sesiones y cifrado de información sensible.
- **Componente de Auditoria:** Registra los eventos del sistema (operaciones financieras), se considera como una pieza independiente con la finalidad de optimizar el rendimiento del aplicativo mediante la recepción de mensajes y almacenamiento de la información de forma asíncrona.
- **Interfaces para exportar servicios:** La arquitectura del sistema permitirá incorporar interoperabilidad con sistemas de terceros, sin impactar los componentes de negocio a través de Web Services. El desarrollo de estos servicios de comunicación no se contempla dentro del alcance de esta propuesta.

El objetivo primordial de una arquitectura distribuida es mantener un alto desempeño y disponibilidad del sistema, así como agilizar el desarrollo y mantenimiento del mismo, gracias a que el código se divide en capas y módulos que atienden tareas específicas.

Una característica fundamental de la arquitectura distribuida es que permite desplegar los módulos de un sistema en múltiples servidores, lo que se traduce en un notable aumento en el rendimiento y estabilidad del sistema, así como un mayor aprovechamiento de la infraestructura disponible.

La Arquitectura de SOLSER es distribuida en multicapas y en la separación modular de componentes de desarrollo y de negocio bien identificadas y agrupados.

En la propuesta de solución presentada se garantiza el uso de la aplicación web para todos los navegadores de manera indistinta como son Chrome, Firefox, Internet Explorer 7 en adelante, manteniendo la calidad visual y funcional de la aplicación web. De la misma manera es indistinto el uso de la aplicación web en los Sistemas Operativos Windows XP y Windows 7 por lo que no se ve limitada para su uso en diferentes versiones del sistema operativo.

Situación actual de las operaciones en Sucursal

Los servidores locales de las sucursales reciben en promedio 1,254 transacciones diarias, esto implica que se ejecutan 157 transacciones por equipo aproximadamente (en una sucursal con 8 equipos), lo que resulta en 1.2 millones de transacciones aproximadamente hacia los servidores (al multiplicar por las 910 sucursales).

El mayor volumen diario de transacciones se da entre las 11:00 y las 19:00, cada transacción puede estar compuesta de una o de múltiples tramas (cada trama es de 10 Kb).

Por medio de la solución que presentamos se garantiza que la aplicación web se puede distribuir en diferentes servidores, permitiendo así un soporte de transacciones superior a las 500 operaciones por segundo.

Detalles generales de la solución

Por medio de la arquitectura y las tecnologías que se integran en la implementación de la solución, se brinda la facultad para hacer uso de todos los dispositivos periféricos en la aplicación web. Permitiendo de esta manera la integración funcional dentro de la misma aplicación para las operaciones que requieran la utilización de algún dispositivo periférico para poder concluirla.

Se contempla la creación de una misma aplicación web que contenga las funcionalidades de las Sucursales y la utilización de los dispositivos periféricos sin necesidad de mantener por separado estas dos acciones del sistema. Brindando la ventaja de mantener centralizado el sistema para mantenimientos y actualizaciones de ambas acciones.

Detalle de actividades

Nombre de tarea
MIGRACIÓN DE FUNCIONALIDADES DE SUCURSALES A WEB
CONCEPCION
Reunión con cliente para levantamiento de requerimientos
Reunión en instalaciones de BanCliente para solución de dudas
Lectura y entendimiento de la documentación proporcionada por el cliente.
Reunión instalaciones del cliente para resolver dudas del requerimiento
Reunión instalaciones del cliente para resolver dudas del requerimiento
Diseñar propuesta del Diagrama de la solución
Crear cronograma de actividades
Elaborar Estimación
Primer análisis de la estimación
Convocar a reunión para revisión de estimación
Llevar acabo reunión y elaborar minuta
Obtener VoBo de la estimación
Elaborar Propuestas
Elaborar Propuesta Técnica
Reunión para revisión de avances en la creación de las propuestas.
Crear minuta de reunión.
Ajustes al diagrama de Arquitectura
Ajustes a Propuesta Técnica
Elaborar Propuesta Económica

Revisión de Propuesta Económica
Crear Minuta de reunión.
Ajustes a Propuesta Económica
Envío de propuesta técnica y económica a jefe inmediato
Obtener VoBo de las propuestas
Enviar propuesta Técnica y Económica al Cliente
Establecer Contratos de Proyecto
INICIACIÓN
Elaboración del plan de trabajo
Establecimiento del equipo de trabajo
Configuración del ambiente de trabajo
Crear matriz de trazabilidad
ELABORACIÓN
Establecer ambiente de desarrollo para el equipo de trabajo
Establecer equipo con Sistema Operativo Windows XP
Establecer equipo con Sistema Operativo Windows 7
Instalar Java 1.8 en cada equipo de trabajo
Instalar Java Scene Builder
Instalar los drivers correspondientes a los periféricos que se usaran
Re-evaluación de las Aplicaciones de BanCliente
Verificación de componentes reutilizables para este proyecto
Identificar y diseñar los componentes del sistema
Definición del Inventario de Clases
Inventario de Pantallas
Ajustar Modelo de Datos

Secuencia de Integración
Revisar y actualizar la matriz de trazabilidad
Prueba de Concepto
Construcción de la Prueba de Concepto
Ajuste del diseño de la construcción de la aplicación
CONSTRUCCIÓN
OLA 1 – Consultas y Seguridad.
Entregable 1
Analizar el código actual de los store procedure
Extracción de reglas de negocio de las capas visuales
Creación de Casos de uso
Definición del plan de pruebas
Diseño de los Casos de Prueba
Sprint 1 - Consultas Generales
Pruebas de calidad SOLSER Sprint 1
Sprint 2
Acceso OFI
Alta Única(Principal)
Módulos Generales
Pruebas de calidad SOLSER Sprint 2
Sprint 3 - Seguridad y Administración
Pruebas de calidad SOLSER Sprint 3
Actualización de Matriz de trazabilidad
Preparación para entrega
Generar Componentes EARs

Empaquetar componentes de entrega
Enviar correo al cliente con detalle de entrega.
Soporte SOLSER para Testing integral
Soporte SOLSER para Piloto de OLA
Soporte SOLSER para Optimización
Soporte SOLSER para Despliegue de OLA
Soporte SOLSER para Estabilización
Pruebas validación BanCoppel
Obtener VoBo por parte del cliente y carta de aceptación firmada
OLA 2 - Alta Única Clientes
Entregable 2
Mesa Funcional en el levantamiento de los requerimientos
Analizar el código actual de los store procedure
Extracción de reglas de negocio de las capas visuales
Creación de Casos de uso
Definición del plan de pruebas
Diseño de los Casos de Prueba
Sprint 4 - Alta única (Alta Clientes)
Pruebas de calidad SOLSER Sprint 4
Sprint 5 - Alta única (Alta Prospectos)
Pruebas de calidad SOLSER Sprint 5
Sprint 6 - Alta Única (Alta Clientes)
Pruebas de calidad SOLSER Sprint 6
Actualización de Matriz de trazabilidad
Preparación para entrega

Generar Componentes EARs
Empaquetar componentes de entrega
Enviar correo al cliente con detalle de entrega.
Soporte SOLSER para Testing integral
Soporte SOLSER para Piloto de OLA
Soporte SOLSER para Optimización
Soporte SOLSER para Despliegue de OLA
Soporte SOLSER para Estabilización
Pruebas validación BanCliente
Obtener VoBo por parte del cliente y carta de aceptación firmada
Entregable 3
Analizar el código actual de los store procedure
Extracción de reglas de negocio de las capas visuales
Creación de Casos de uso
Definición del plan de pruebas
Diseño de los Casos de Prueba
Sprint 7 - Alta Única (Apertura de Cuentas)
Pruebas de calidad SOLSER Sprint 7
Sprint 8 - Alta Única (Apertura de Productos)
Pruebas de calidad SOLSER Sprint 8
Sprint 9 - Alta Única (Servicios Adicionales)
Pruebas de calidad SOLSER Sprint 9
Sprint 10 - Alta Única (Otras Funcionalidades)
Pruebas de calidad SOLSER Sprint 10
Actualización de Matriz de trazabilidad

Preparación para entrega
Generar Componentes EARs
Empaquetar componentes de entrega
Enviar correo al cliente con detalle de entrega.
Soporte SOLSER para Testing integral
Soporte SOLSER para Piloto de OLA
Soporte SOLSER para Optimización
Soporte SOLSER para Despliegue de OLA
Soporte SOLSER para Estabilización
Pruebas validación BanCliente
Obtener VoBo por parte del cliente y carta de aceptación firmada
OLA 3 – Consultas, Mantenimientos y Servicios
Entregable 4
Analizar el código actual de los store procedure
Extracción de reglas de negocio de las capas visuales
Creación de Casos de uso
Definición del plan de pruebas
Diseño de los Casos de Prueba
Sprint 11 - Consultas de Alta única
Pruebas de calidad SOLSER Sprint 11
Sprint 12 - Mantenimientos
Pruebas de calidad SOLSER Sprint 12
Sprint 13 - Líneas de Crédito
Pruebas de calidad SOLSER Sprint 13
Sprint 14 - Servicios 1

Pruebas de calidad SOLSER Sprint 14
Sprint 15 - Servicios 2
Pruebas de calidad SOLSER Sprint 15
Sprint 16 - Servicios 3
Pruebas de calidad SOLSER Sprint 16
Actualización de Matriz de trazabilidad
Preparación para entrega
Generar Componentes EARs
Empaquetar componentes de entrega
Enviar correo al cliente con detalle de entrega.
Soporte SOLSER para Testing integral
Soporte SOLSER para Piloto de OLA
Soporte SOLSER para Optimización
Soporte SOLSER para Despliegue de OLA
Soporte SOLSER para Estabilización
Pruebas validación BanCliente
Obtener VoBo por parte del cliente y carta de aceptación firmada
OLA 4 – Procesos de Ventanilla
Entregable 5
Analizar el código actual de los store procedure
Extracción de reglas de negocio de las capas visuales
Creación de Casos de uso
Definición del plan de pruebas
Diseño de los Casos de Prueba
Sprint 17 - Caja General

Pruebas de calidad SOLSER Sprint 17
Sprint 18 - Caja Mixta
Pruebas de calidad SOLSER Sprint 18
Sprint 19 - Caja Principal
Pruebas de calidad SOLSER Sprint 19
Actualización de Matriz de trazabilidad
Preparación para entrega
Generar Componentes EARs
Empaquetar componentes de entrega
Enviar correo al cliente con detalle de entrega.
Soporte SOLSER para Testing integral
Soporte SOLSER para Piloto de OLA
Soporte SOLSER para Optimización
Soporte SOLSER para Despliegue de OLA
Soporte SOLSER para Estabilización
Pruebas validación BanCliente
Obtener VoBo por parte del cliente y carta de aceptación firmada
Entregable 6
Analizar el código actual de los store procedure
Extracción de reglas de negocio de las capas visuales
Creación de Casos de uso
Definición del plan de pruebas
Diseño de los Casos de Prueba
Sprint 20 - Crédito
Pruebas de calidad SOLSER Sprint 20

Sprint 21 - Captación
Pruebas de calidad SOLSER Sprint 21
Sprint 22 - Servicios al Cliente 1
Pruebas de calidad SOLSER Sprint 22
Sprint 23 - Servicios al Cliente 2
Pruebas de calidad SOLSER Sprint 23
Sprint 24 - Transferencias
Pruebas de calidad SOLSER Sprint 24
Sprint 25 - Consultas y Mantenimientos
Pruebas de calidad SOLSER Sprint 25
Actualización de Matriz de trazabilidad
Preparación para entrega
Generar Componentes EARs
Empaquetar componentes de entrega
Enviar correo al cliente con detalle de entrega.
Soporte SOLSER para Testing integral
Soporte SOLSER para Piloto de OLA
Soporte SOLSER para Optimización
Soporte SOLSER para Despliegue de OLA
Soporte SOLSER para Estabilización
Pruebas validación BanCliente
Obtener VoBo por parte del cliente y carta de aceptación firmada
OLA 5 – Procesos de Gestión de Sucursal
Entregable 7
Analizar el código actual de los store procedure

Extracción de reglas de negocio de las capas visuales
Creación de Casos de uso
Definición del plan de pruebas
Diseño de los Casos de Prueba
Sprint 26 - Procesos Especiales
Pruebas de calidad SOLSER Sprint 26
Sprint 27 - Procesos de Fin de Día
Pruebas de calidad SOLSER Sprint 27
Sprint 28 - Procesos de Fin de Día (Reportes 1)
Pruebas de calidad SOLSER Sprint 28
Sprint 29 - Procesos de Fin de Día (Reportes 2)
Pruebas de calidad SOLSER Sprint 29
Actualización de Matriz de trazabilidad
Preparación para entrega
Generar Componentes EARs
Empaquetar componentes de entrega
Enviar correo al cliente con detalle de entrega.
Soporte SOLSER para Testing integral
Soporte SOLSER para Piloto de OLA
Soporte SOLSER para Optimización
Soporte SOLSER para Despliegue de OLA
Soporte SOLSER para Estabilización
Pruebas validación BanCliente
Obtener VoBo por parte del cliente y carta de aceptación firmada
OLA 6 – Catálogos

Entregable 8
Analizar el código actual de los store procedure
Extracción de reglas de negocio de las capas visuales
Creación de Casos de uso
Definición del plan de pruebas
Diseño de los Casos de Prueba
Sprint 30 - Mantenimientos 1
Pruebas de calidad SOLSER Sprint 30
Sprint 31 - Mantenimientos 2
Pruebas de calidad SOLSER Sprint 31
Actualización de Matriz de trazabilidad
Preparación para entrega
Generar Componentes EARs
Empaquetar componentes de entrega
Enviar correo al cliente con detalle de entrega.
Soporte SOLSER para Testing integral
Soporte SOLSER para Piloto de OLA
Soporte SOLSER para Optimización
Soporte SOLSER para Despliegue de OLA
Soporte SOLSER para Estabilización
Pruebas validación BanCliente
Obtener VoBo por parte del cliente y carta de aceptación firmada
IMPLANTACIÓN
Entrega de manuales de usuario e instalación
Generar Componentes EARs Finales

Generar Componentes de Base de Datos
Empaquetar proyecto Completo para entrega al cliente
CIERRE
Elaborar Presentación de Cierre de Proyecto
MONITOREO
Riesgos
Generar/Actualizar Plan de Respuesta a Riesgos
Revisar/Actualizar status de riesgos en el reporte
Actualizar BD Riesgos
Compartir el plan de respuesta al riesgo
Proyecto
Revisar reporte de actividades de equipo vs cronograma
Convocatoria a reunión a equipo
Reunión con equipo y Elaboración de minuta
Elaboración de reporte de seguimiento
Análisis del estado del proyecto junto con jefe inmediato
Elaboración de reporte de seguimiento para cliente
Actualizar Plan de Trabajo
Tener reportes y cronograma actualizado

Tabla 3 Detalle de actividades

Riesgos y dependencias.

Riesgos Solser

- Que no se cuente con el ambiente requerido en Solser
- Que la Infraestructura no esté lista.

Mitigación de estos riesgos

- En Solser ya se tiene una estrategia de mitigación de estos riesgos.

Riesgos BanCliente

- Que no se tenga listo el ESB cuando se requiera.
- Que no se cuente con los dispositivos periféricos a tiempo.
- Que no se provean los insumos, cuando se requieran.
- Que no funcione el Interact o tenga interrupciones.
- No contar con los SP's
- Si existen cambios en los SP's, que no se reflejen en el tiempo requerido.
- Falta de información en el Gap Análisis (no contar con los programas fuentes o que estén incompletos)
- Que el personal clave de BanCliente, no esté disponible durante la vida del proyecto.
- Que no se cumplan los tiempos programados para la realización de las pruebas en cada etapa, por el personal de BanCliente.

PLAN DE TRABAJO

Objetivo del documento

El propósito de este documento es presentar a los involucrados un plan de trabajo con las tareas asignadas, para llevar a cabo el cumplimiento del requerimiento de Cambio de Plataforma de Sucursales a Web (CPlaSu), así como las políticas que se deben seguir.

Audiencia del documento

Gerentes, líderes de proyecto y equipo de trabajo.

Información general del proyecto

Claves y Nombres

Clave Proyecto/OT	S057-PRYE20150929-022
Nombre Corto Proyecto	CPlaSu
Nombre Largo Proyecto	Cambio de Plataforma de Sucursales a Web
Nombre Líder de Proyecto	Rubén Mota León
Fecha Elaboración	15/Enero/2018

Tabla 4 Información general del proyecto

Objetivos del Proyecto

Realizar la migración de las 350 funcionalidades que actualmente existen en las sucursales del cliente a un ambiente Web que permita una optimización de las funcionalidades y mantener un ambiente preparado para la integración de nuevos módulos. Permitiendo que la aplicación proporcione de manera íntegra la comunicación con los diferentes dispositivos

(periféricos) que se utilizan actualmente en las 910 Sucursales para las operaciones diarias como digitalización de documentos y/o cheques, lectoras de tarjetas de crédito o débito mismos que deben interactuar con la aplicación.

Proceso definido para el Proyecto

Proceso Estándar y Adaptaciones

La metodología que se va utilizar es la de SCRUM, implementando documentos que se generen con el formato que solicite el cliente.

Recursos Involucrados en el Proyecto

Rol	Nombre	Forma de Involucramiento / Responsabilidad	Área / Departamento	Teléfono/ Ext.	Email
Coordinador de Proyecto	Rubén Mota León	Coordinar en conjunto con el cliente los requerimientos a realizar.	Desarrollo		
Líder de Proyecto	Luz Ma. Elías Sebastián	Estimación de tiempos, reuniones con el cliente,	Gestión		
Líder Técnico	Jesús Ferruzca Luna	Desarrollo de requerimientos, Aclarar dudas con equipo de trabajo, especificar reglas de negocio.	Desarrollo		
Analista	José Paz	Análisis de sistema y Desarrollo de requerimientos	Análisis/Desarrollo		
Analista	Hernando Guerrero	Análisis de sistema y Desarrollo de requerimientos	Análisis/Desarrollo		

Rol	Nombre	Forma de Involucramiento / Responsabilidad	Área / Departamento	Teléfono/ Ext.	Email
Analista	Edgar Rosas	Análisis de sistema y Desarrollo de requerimientos	Análisis/Desarrollo		
Analista	Iván Ortega	Análisis de sistema y Desarrollo de requerimientos	Análisis/Desarrollo		
Desarrollador	Juan Mata	Desarrollo de requerimientos	Desarrollo		
Desarrollador	Marco Hernández	Desarrollo de requerimientos	Desarrollo		
Desarrollador	Alejandro Díaz	Desarrollo de requerimientos	Desarrollo		
Desarrollador	Eduardo Ortiz	Desarrollo de requerimientos	Desarrollo		
Desarrollador	Raúl Monroy	Desarrollo de requerimientos	Desarrollo		

Tabla 5 Recursos Involucrados en el Proyecto

Planeación y Seguimiento del proyecto

Valoración del Proyecto

Especificación Requerimientos (FDS-003Especificacion_Req.doc)

FMD-002Propuesta_Tecnica_Aplicaciones_20151029

/home/cvs/SolserDesarrollo/BanCliente/CPlaSu/Documentacion/1- Inic-Elab/

Documentos Técnicos Proyecto / Mantenimiento

Los documentos técnicos a generar durante el desarrollo del proyecto y la ruta donde se podrá localizar cada uno de ellos, se describe a continuación.

Fase	Nombre de Documento	Ruta
Inic-Elab	FOS-001PlanTrabajo	/home/cvs/SolserDesarrollo/BanCliente/CPlaSu/Documentacion/1- Inic-Elab/
	FDS-007Cronograma	/home/cvs/SolserDesarrollo/BanCliente/CPlaSu/Documentacion/1- Inic-Elab/
	FDS-006CasosUso	/home/cvs/SolserDesarrollo/BanCliente/CPlaSu/Documentacion/1- Inic-Elab/
	FDS-004Matriz_Trazabilidad	/home/cvs/SolserDesarrollo/BanCliente/CPlaSu/Documentacion/1- Inic-Elab/
	FDS-015 Arquitectura de la Aplicación.	/home/cvs/SolserDesarrollo/BanCliente/CPlaSu/Documentacion/1- Inic-Elab/
	F-DS-023Inventario de Programas.	/home/cvs/SolserDesarrollo/BanCliente/CPlaSu/Documentacion/1- Inic-Elab/
	FDS-026Modelo de Datos	/home/cvs/SolserDesarrollo/BanCliente/ CPlaSu /Documentacion /2- Construccion
	FDS-027Inventario de clases.	/home/cvs/SolserDesarrollo/BanCliente/CPlaSu/Documentacion/1- Inic-Elab/

Fase	Nombre de Documento	Ruta
Construcción	FDS-010Reporte Revisión x Pares.	/home/cvs/SolserDesarrollo/BanCliente/CPlaSu/Documentacion /2-Construccion
	FPS-018 Casos de Prueba.	/home/cvs/SolserDesarrollo/BanCliente/ CPlaSu /Documentación /2-Construccion
	MAD-001Manual de Usuario.	/home/cvs/SolserDesarrollo/BanCliente/ CPlaSu /Documentacion /2-Construccion
	MAD-002Manual Técnico.	/home/cvs/SolserDesarrollo/BanCliente/ CPlaSu /Documentacion /2-Construccion
Implantación	FDS-033Reporte Incidentes.	Herramienta COS
Cierre	FOR-009Presentación de Cierre	/home/cvs/SolserDesarrollo/BanCliente/CPlaSu/Documentacion /2-Construccion

Tabla 6 Documentos Técnicos Proyecto

Administración de Datos / Información / Conocimiento

El área encargada de proporcionar los servicios y los recursos (no humanos) para establecer el ambiente de desarrollo de proyecto es el encargado de Administración de la Configuración.

Se proporcionará un repositorio donde todos los documentos que se generen durante el ciclo de vida del proyecto serán almacenados. La ruta del repositorio es:

/home/cvs/SolserDesarrollo/BanCliente/CPlaSu/Documentacion/

Seguimiento

Para dar seguimiento al proyecto los desarrolladores, analistas e involucrados con el proyecto deberán informar los avances realizados. El líder de proyecto deberá comunicar a ODP los avances llevados hasta ese momento, así como cualquier incidencia que pueda ocurrir durante el ciclo de vida del proyecto. Al término de cada fase del proyecto se realizará una reunión donde participará el equipo de trabajo y en la cual se revisarán dichos informes, además de las actividades que se plasman en el cronograma de trabajo y se dará monitoreo y seguimiento de los riesgos detectados, así como nuevos riesgos que puedan surgir.

Resolución de Problemas

Todos los problemas e imprevistos con el equipo de trabajo que puedan presentarse, tales como impuntualidad, incidentes, conducta de los miembros del equipo, entre otros, se ven reflejados en documento **POL-RHC-001 Estándares de Ambiente de Trabajo**, en cual, se definen todas las acciones que se deberán de tomar en cuenta y la escalabilidad para resolverlos.

Otros planes que influyen al proyecto / mantenimiento

- Plan de calidad.
- Plan de Auditorías.
- Plan de Pruebas.
- Plan de Administración de riesgos.

Organización del Proyecto

Organigrama de Equipo de trabajo

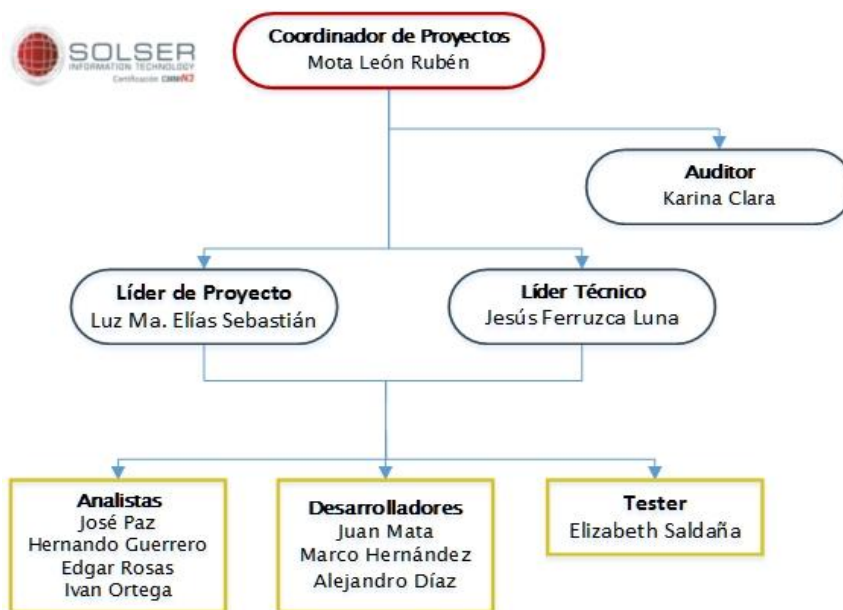


Ilustración 10: Organización del equipo de trabajo

Principales Hitos y Entregables

Hito / Entregable	Fecha Compromiso
OLA 1 Plataforma y Seguridad	[08/2018]
OLA 2 Plataforma (Alta Única)	[12/2018]
OLA 3 Plataforma (Consulta/Servicios/mantenimiento)	[04/2018]
OLA 4 Gestión de Sucursales /Ventanilla Caja/Ventanilla	[08/2018]
OLA 5 Gestión de Sucursal	[11/2018]
OLA 6 Catálogos	[03/2018]
OLA 7 Varios (46) posibles funcionalidades nuevas	[06/2018]

Tabla 7 Principales Hitos y Entregables

Delegación de Responsabilidades

Responsable	Suplente
Rubén Mota León	Luz Ma. Elías Sebastián
Luz Ma. Elías Sebastián	Jesús Ferruzca Luna
Jesús Ferruzca Luna	José Gabriel Paz Sánchez

Tabla 8 Delegación de Responsabilidades

Evaluación de desempeño de equipo integrado

La evaluación del desempeño del equipo de trabajo asignado al proyecto: los desarrolladores, analistas e involucrados con el proyecto enviarán al líder de proyecto los avances realizados, así como los informes de actividades realizadas, el líder de proyecto informará a Oficina de Proyectos los avances llevados a cabo en el Proyecto, así como cualquier incidencia que pueda repercutir en retrasos en la entrega del requerimiento solicitado.

ARQUITECTURA DE LA APLICACIÓN

Sobre este Documento (Propósito)

El presente documento contiene el diseño elaborado para el proyecto CPlaSUC **Cambio de Plataforma de Sucursales**, el cual es producto de un análisis minucioso de los requisitos del sistema, según estos pueden ser satisfechos con las tecnologías y características discutidas con los clientes y usuarios. El cual está organizado en:

- Características generales del diseño;
- Requisitos atendidos por el diseño
- Modelos y las diferentes Vistas que lo detallan.

Los modelos son utilizados tanto para el análisis de requisitos, como para el diseño de la solución; estos son presentados por medio de Vistas o Diagramas generalmente utilizando notaciones gráficas como UML.

Referencias

Para la realización de este documento nos apoyamos de los siguientes documentos:

- Plataforma Sucursales RFP.PDF

Descripción General

Objetivos del diseño

Actualizar la arquitectura tecnológica para tener un crecimiento tanto vertical sobre la infraestructura, como horizontal en la arquitectura tecnológica, mejorando el rendimiento y cumpliendo con los requerimientos del cliente.

Vistas y Planos

Los sistemas intensivos en software se encuentran formados por un conjunto de componentes, que no son más que los elementos listos para ser producidos por el proyecto, dichos componentes se distribuyen sobre los distintos equipos según lo que se detalla en la *Vista de Despliegue*.

La Vista de Casos de Uso presenta la o las funcionalidades que formaran el sistema

Por otro lado, la *Vista Lógica* presenta al sistema como un todo, indicando en términos propios de la tecnología utilizada las partes que lo forman y relaciones entre ellas.

Vista de Casos de Uso

ID CU	Nombre CU	Sinopsis
1	CU-Consulta de Clientes	Consultar, Cancelar y Salir.

Tabla 9 Vista de Casos de Uso

Diagrama:

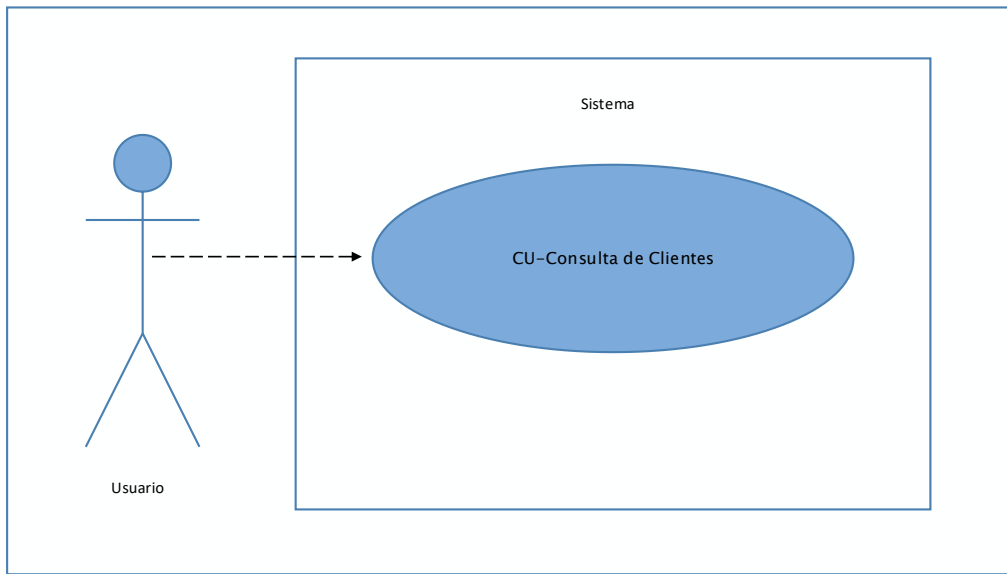


Ilustración 11: CU – Consulta de clientes

Vista Lógica

Módulos

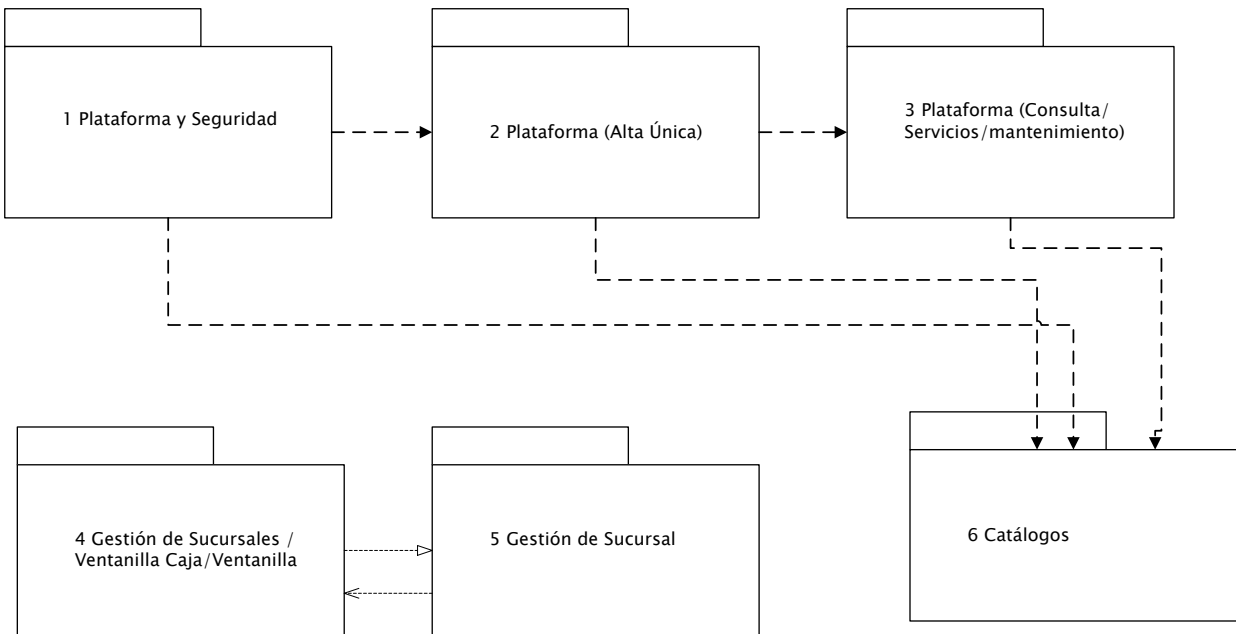
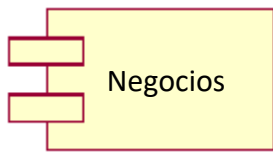
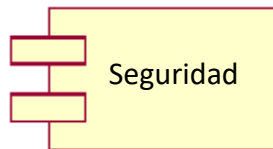


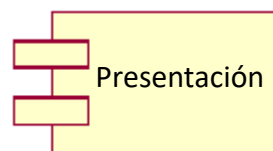
Ilustración 12: Vista lógica



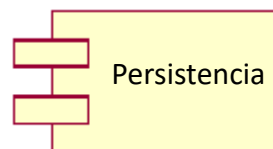
El módulo de negocio está contenido en el componente EARCPlaSucNegocio en el cual se encuentran los servicios que incorporan las validaciones y reglas del negocio y son programas codificados en Java



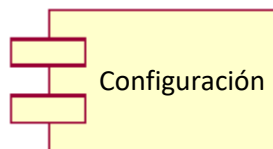
El módulo de seguridad está contenido en el componente EARCPlaSucSeguridad en el cual se encuentran los Servicio para la Autenticación de usuario y el Control sobre las operaciones permitidas bajo el cifrado de información sensible de datos.



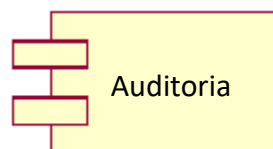
El módulo de presentación está contenido en el componente EARCPlaSucWeb en el cual se encuentran las interfaces visuales que conforman la aplicación, y está formada por elementos que integran tecnologías como JSP, JSTL, HTML y JavaScript



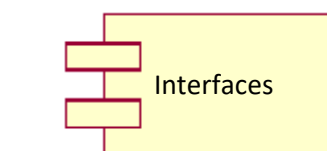
El módulo de persistencia está contenido en el componente EARCPlaSucPersistencia el cual realiza las comunicaciones hacia Repositorio de Datos por medio de Middleware InterAct o ESB.



El módulo de Configuración está contenido en el componente EARCPlaSucConfig el cual mantiene y distribuye los parámetros de configuración entre capas y componentes del sistema.



El módulo de Auditoria está contenido en el componente EARCPlaSucAuditoria el cual registra los eventos del sistema (operaciones financieras), almacena información de forma asíncrona generando bitácoras de autorías.



El módulo de Interfaces está contenido en el componente EARCPlaSuInterfaces este módulo del sistema permitirá incorporar interoperabilidad con sistemas de terceros, sin impactar los componentes de negocio a través de Web Services. El desarrollo de estos servicios de comunicación no se contempla dentro del alcance de esta propuesta.

Diagrama de clases

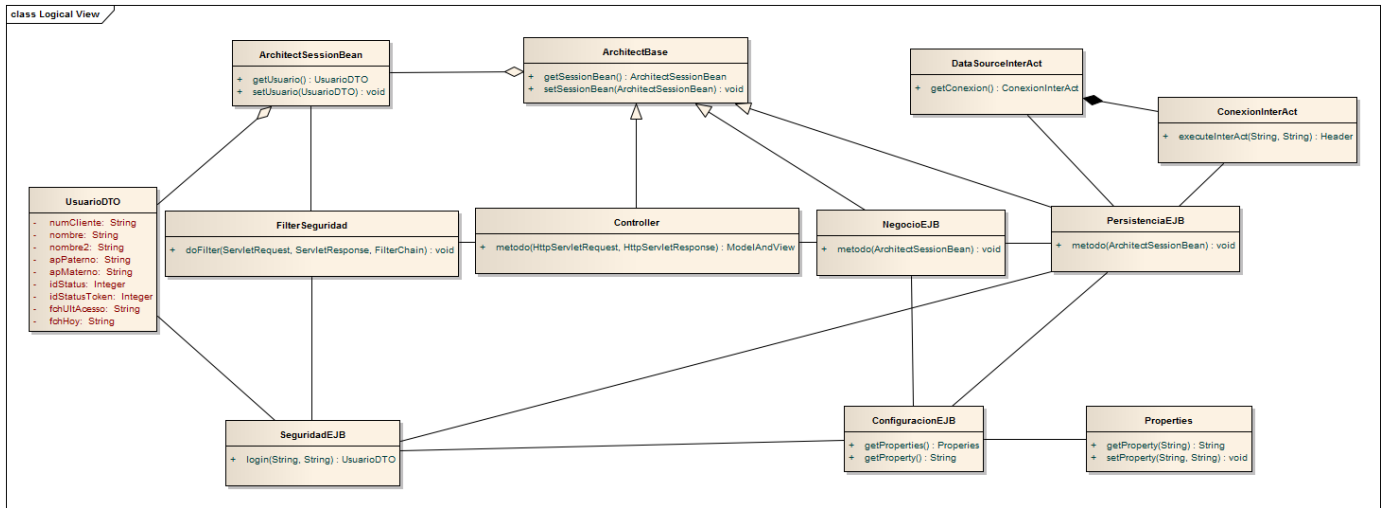


Ilustración 13: Diagrama de clases

Diagramas de Secuencia

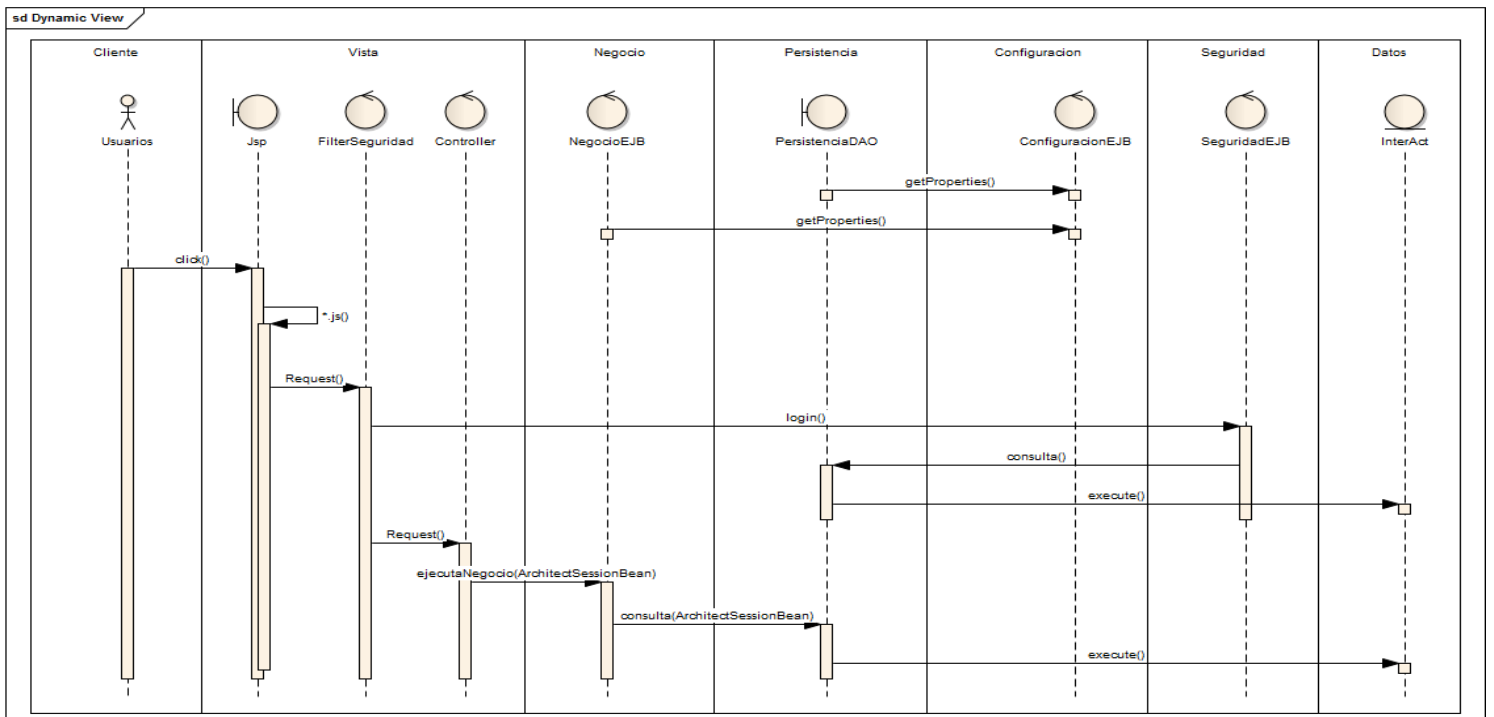


Ilustración 14: Diagrama de secuencias

Arquitectura del Proyecto

La Arquitectura de la aplicación consiste en 4 capas (Presentación, Negocio, Persistencia, y Datos) la cual utiliza como soporte el Spring Framework (Spring o MVC).

Presentación:

Es el conjunto de interfaces visuales que conforman la aplicación, y está formada por elementos que integran tecnologías como JSP, JSTL, HTML y JavaScript.

Negocio:

Servicios que incorporan las validaciones y reglas del negocio y son programas codificados en Java.

Persistencia:

Componente de comunicaciones hacia el repositorio de datos (Middleware BanCliente), se encarga de la transformación de las tramas a Objetos (Beans), y son programas codificados en Java que InterActúan con el API de BanCliente.

Datos:

Middleware de BanCliente. Es el medio por el cual se van a ejecutar las consultas SQL o los Store Procedure.

Calidad

Alta disponibilidad: En un ambiente distribuido, el sistema alcanza niveles de servicio del 99% de tiempo online.

Seguridad: Control sobre las operaciones permitidas dentro del sistema, generación de bitácoras de auditoría.

Escalabilidad: Capacidad para incrementar los recursos de Hardware para soportar mayor carga de trabajo.

Fiabilidad: Capacidad para mantener la consistencia de la información mediante el uso de transacciones en cada operación.

Flexibilidad: El diseño permite incorporar cambios del negocio sin impactar a los componentes base.

INVENTARIO DE PROGRAMAS

Plataforma:	Multiplataforma y Multinavegador
--------------------	----------------------------------

Tabla 10 Plataforma

Descripción General del Programa

ID	001
Nombre de Programa	Cambio de Plataforma Sucursales Web
Funcionalidad	Proporciona el resultado de Consultas Generales, Altas Únicas, Mantenimientos y Servicios.
Lenguaje	Java 1.8
Tecnologías que involucra	Java FX, Spring, HTML, JavaScript, Certificado De firma de Código Java, JIRA, NGINX.
Tipo de programa	Clase de java
Flujo del Programa	<p>FDS-006CasosUso_Consulta de Clientes.</p> <p>FDS-006CasosUso_ConsultaEdoCta_ConsultaEdoMov.</p> <p>FDS-006CasosUso_Consulta de Tasa de Inversiones.</p> <p>FDS-006CasosUso_Consulta de Cheques Devueltos.</p> <p>FDS-006CasosUso_Mayoria de Edad Cuenta Efectiva Niños.</p> <p>FDS-006CasosUso_Consulta de Cálculo de Intereses TDC.</p> <p>FDS-006CasosUso_Consulta de Chequeras y Cheques.</p> <p>FDS-006CasosUso_Consulta de Consulta de Pagaré.</p> <p>FDS-006CasosUso_Consulta de Cuentas Con Chequeras Canceladas.</p> <p>FDS-006CasosUso_Consulta de Vencimiento de Pagarés e Inversiones a 7 días.</p> <p>FDS-006CasosUso_Consulta de Saldos Retenidos.</p> <p>FDS-006CasosUso_Constancias_ISR.</p> <p>FDS-006Inversiones_sin_cuenta_EJE_con_mas_de_tres_años.</p>

	<p>FDS-006CasosUso_Consulta_Tabla_Amortizacion.</p> <p>FDS-006CasosUso_Actualizacion_de_CURP.</p> <p>FDS-006CasoUso_Consulta_Poliza_Club_Proteccion.</p> <p>FDS-006CasosUso_ConsultaPagare_CambioInstruccionesInversionCrecienteyPagare.</p> <p>FDS-006CasosUso_Cambio_NIP (débito y crédito).</p> <p>FDS-006CasosUso_Cancelación de Tarjeta (débito y crédito).</p> <p>FDS-006CasosUso_Desbloqueo de Tarjeta (débito y crédito).</p> <p>FDS-006CasosUso_Reposición de Tarjeta (débito y crédito).</p> <p>FDS-006CasosUso_MantenimientoDeHuella.</p> <p>FDS-006CasoUso_Captura Billetes_Falsos.</p> <p>FDS-006CasoUso_Asignacion_Numero_Guia.</p> <p>FDS-006CasoUso_Consulta_Billetes_Falsos.</p>
Resultado Final	Obtener la correcta Consulta para los diferentes procesos y validaciones que conforman el aplicativo de Cambio de Plataforma Sucursales Web.
Consideraciones Generales	<p>Que el BUS funcione adecuadamente durante la etapa de implementación a pruebas.</p> <p>Migración de 350 funcionalidades desarrolladas en Visual Basic a un ambiente web desarrollado bajo la tecnología de Java.</p>

Tabla 11 Descripción General del Programa

Diagrama de Flujo del Programa

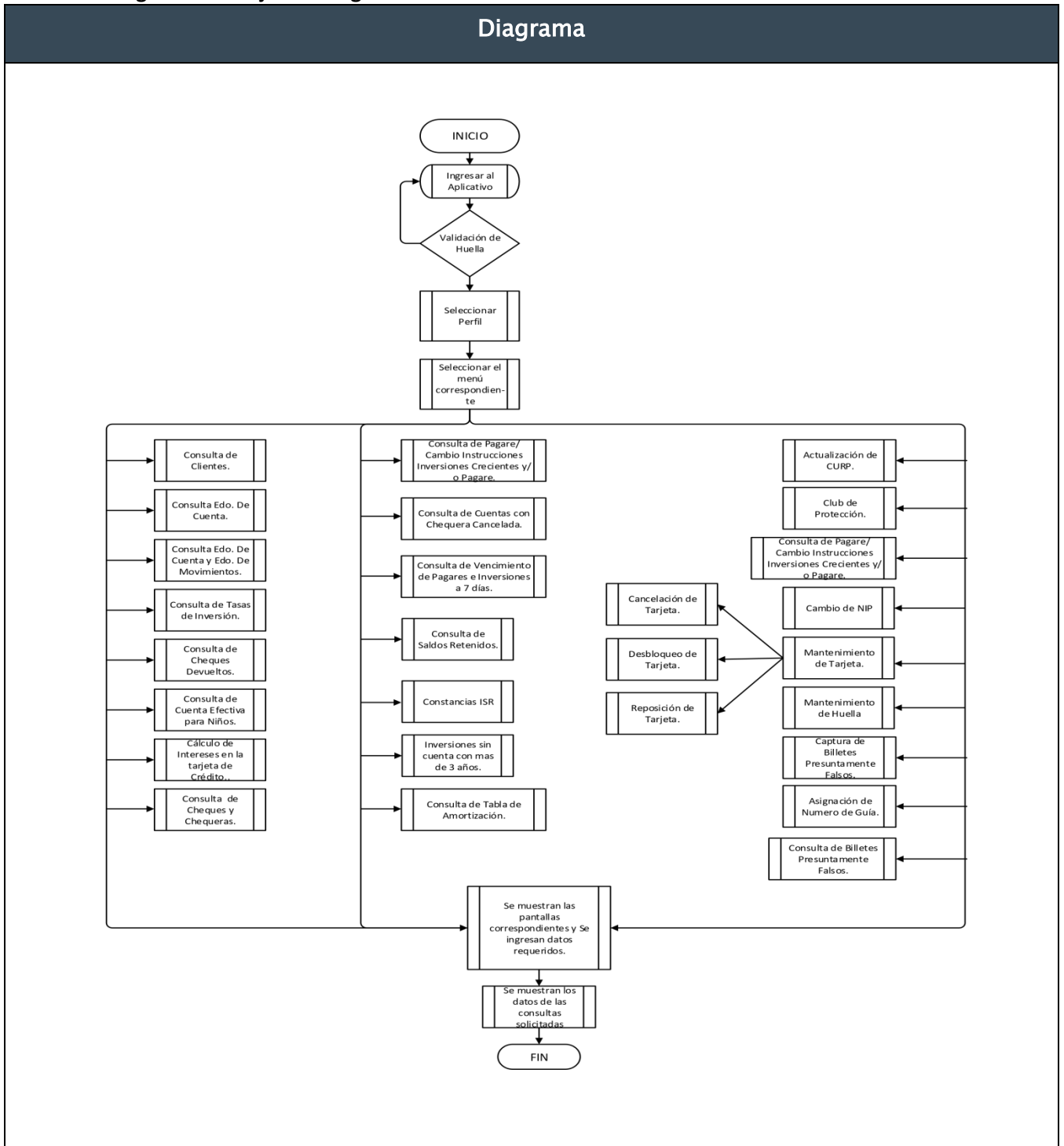


Ilustración 15: Flujo del programa

INVENTARIO DE CLASES

Clase 1 [cajacontroller] 01 Apertura de inicio de día

Análisis de Clases

Nombre de la Clase	Cajacontroller		
Descripción	APERTURA DE INICIO DE DÍA		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad a la transacción de apertura de inicio de día		
Atributos	Descripción	Tipo	Valor
panelPrincipal	Panel principal y root de la pantalla sobre la cual está armado	private	AnchorPan
panelCaja	-	private	AnchorPan
panelMenu	-	private	AnchorPan
btnMenu	-	Private	SButton
btnAceptar	Botón que realiza la acción de aceptar	Private	SButton
btnCancelar	Botón que realiza la acción de cancelar	private	SButton
btnSalir	Botón que realiza la acción de salir	Private	SButton
btnConsultaCliente	Botón que realiza la acción de consulta cliente	Private	SButton
btnCalculadora	Botón que realiza la acción de ejecutar la función calculadora	Private	SButton
imgBtnAceptar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
ImgBtnCancela	- Imagen para el botón cancelar que	private	ImageView

	se reemplazara en tiempo de ejecución		
ImgBtnSalir	Imagen para el botón salir que se reemplazara en tiempo de ejecución	Private	ImageView
ImgBtnConsulta	Imagen para el botón consulta que se reemplazara en tiempo de ejecución	Private	ImageView
ImgBtnCalculadora	Imagen para el botón calculadora que se reemplazara en tiempo de ejecución	private	ImageView
ImgEstacionCerrada	Imagen para el botón salir estación cerrada	Private	ImageView
modal	Modal sobre el cual se pintaran los mensajes de erros, confirmación, Información, etc... De cada pantalla	Private	SolserModalSameStage
servicio	-	Private	CajaService
mantoServiciosService	-	Private	MantoServiciosService
generalRestCallService	-	private	GeneralRestCallService
lblTransaccion	-	private	Label
lblCuentas	-	Private	Label
txtTransaccion	-	Private	STextField
cmbCuentas	-	private	SComboBox<String>
abrirMenu	-	Private	TranslateTransition
cerrarMenu	-	Private	TranslateTransition
mostrarMenu	-	Private	FadeTransition

ocultarMenu	-	Private	FadeTransition
-------------	---	---------	----------------

Tabla 12 Inventario de clases Cajacontroller

Diseño de Clases

Métodos y Funciones

public void initialize(final URL location, final ResourceBundle resources)

public void resetView()

public void ejecutar()

public void cancelarTransaccion()

mx.com.solser.cplasu.interfaces.Callback salirCallBack()

mx.com.solser.cplasu.interfaces.Callback cancelarTransaccionCallBack()

Objetos (Instancias)

ReqSusuarios();

SusuariosDTO();

ReqActualizaSusuarios

VDiariosDTO

DepurarCampaniasDTO

SpConsMaecamReplicaDTO

SpConscomespreplicaDTO

SpPaseSucDTO

SimpleDateFormat

ReqArqueoCajero

ReqArqueo

Clase 2 [CierreTemporalEstacionController] 02 Cierre temporal de estación

Análisis de Clases

Nombre de la Clase	CierreTemporalEstacionController		
Descripción	Clase controlador		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad a la transacción cierre temporal de estación		
Atributos	Descripción	Tipo	Valor
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	private	AnchorPane
btnAceptar	Botón que realiza la acción de la transacción	private	SButton
btnSalir	Botón que realiza la acción de cancelar la transacción llevándonos a la De ventanilla	private	SButton
imgBtnAceptar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
imgBtnCancelar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
modal	Modal sobre el cual se pintaran los mensajes de erros, confirmación, Información, etc... De cada pantalla	private	SolserModalSameStage
Servicio	Servicio comun de la caja	private	CajaService
respuesta	-	private	boolean

Tabla 13 Inventario de clases CierreTemporalEstacionController

Diseño de Clases

Métodos y Funciones

public void initialize(URL location, ResourceBundle resources)

public void resetView()

Objetos (Instancias)

ReqActualizaSusuarios()

SusuariosDTO()

VDiariosDTO()

Dependencias

CierreTemporalEstacionController

Clase 3 [AperturaDeEstacionController] 03 Apertura de estación controller

Análisis de Clases

Nombre de la Clase	AperturaDeEstacionController		
Descripción	Clase Controlador		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad a la transacción Apertura de estación del cajero principal		
Atributos	Descripción	Tipo	Valor
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	private	AnchorPane
btnAceptar	Botón que realiza la acción de la transacción	private	SButton
btnSalir	Botón que realiza la acción de cancelar la transacción llevándonos a la Pantalla principal de ventanilla	private	SButton
imgBtnAceptar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
imgBtnCancelar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
modal	Modal sobre el cual se pintaran los mensajes de erros, confirmación, Información, etc... De cada pantalla	private	SolserModalSameStage

servicio	Servicio común de la caja	private	CajaService
----------	------------------------------	---------	-------------

Tabla 14 Inventario de clases AperturaDeEstacionController

Diseño de Clases

Métodos y Funciones

```

public void initialize(URL location, ResourceBundle resources) {
this.modal = SolserApplicationConfig.getSolserModalSameStage();
this.servicio = new CajaService();
this.imgBtnAceptar.setImage(GenericosCaja.ObtenerBtnAceptar());
this.imgBtnCancelar.setImage(GenericosCaja.ObtenerBtnSalir());

    public void resetView() {

        NavegacionControlesService.getInstance().creaNavegacion();

        this.ponFoco(this.btnAceptar);

        private void validaEstatusEstacion()

```

Objetos (Instancias)

```

ReqSusuarios()

Callback()

```

Dependencias

AperturaDeEstacionController

Clase 4 [CapturaTarjetasController] 500 Captura de tarjetas

Análisis de Clases

Nombre de la Clase	CapturaTarjetasController		
Descripción	Clase controlador		
Responsabilidades	Clase controlador que hereda de AbstractPanelFactory		
Atributos	Descripción	Tipo	Valor
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	private	AnchorPane
btnAceptar	Botón que realiza la acción de aceptar	Private	SButton
btnSalir	Botón que realiza la acción de salir	Private	SButton
btnCaptura	Botón que realiza la acción de capturar	private	SButton
btnAgregar	Botón que realiza la acción de agregar	private	SButton
imgBtnAceptar	Imagen para el botón aceptar	private	ImageView
imgBtnCancelar	Imagen para el botón cancelar	Private	ImageView
imgBtnCaptura	Imagen para el botón captura	Private	ImageView
imgBtnAgregar	Imagen para el botón agregar	Private	ImageView
modal	-	private	SolserModalSameStage
servicio	-	private	CajaService
cajeroPrincipalService	-	private	CajeroPrincipalService
txtCaptura	-	Private	STextField
txtSucursal	-	private	STextField

txtLote	-	private	STextField
txtTarjetaInicial	-	private	STextField
txtTarjetaFinal	-	Private	STextField
tblTarjetas	-	private	STableView<LoteTarjeta>

Tabla 15 Inventario de clases CapturaTarjetasController

Diseño de Clases

Métodos y Funciones

// METODOS PUBLICOS

public void initialize(URL location, ResourceBundle resources)

public void resetView()

// METODOS PRIVADOS

// Configura el estado inicial de los controles de la pantalla

private void configuracionInicalDePantalla()

// Configuracion incial de la tabla

private void configuracionInicialTabla()

// reactiva inputs de ingreso de datos

private void activarInputs(final Boolean campoCantidad, final Boolean camposInserccion, final Boolean btnAgregar)

// Limpia el contenido de los inputs

private void formateaCampos()

// Contadores de caracteres para los campos

public EventHandler<KeyEvent> filtroCantidadCaracteres(final Integer i, STextField sigFoco)

// Sobrecargar del metodo para poner foco en boton

public EventHandler<KeyEvent> filtroCantidadCaracteres(final Integer i, SButton sigFoco)

// Validacion campo sucursal

private void validacionSucursal()

// Validacion campo lote

```
private void validacionLote()
// Validacion campo numero de tarjeta inicial
private void validacionTarjetaInicial()
// Validacion campo numero de tarjeta final
private void validacionTarjetaFinal()
// Validacion campo sucursal correspondiente
private void validacionSucursalCorrespondiente(String sucursal)
// Metodo de adiccion de lotes a tabla para preparar inserccion
private void ingresaLotesTablas()
```

Objetos (Instancias)

CajeroPrincipalService

CajaService

panelNavegable

btnAceptar

btnSalir

btnCaptura

btnAgregar

imgBtnAceptar

imgBtnCancelar

imgBtnCaptura

imgBtnAgregar

modal

servicio

CajeroPrincipalService

txtCaptura

txtSucursal

txtLote

txtTarjetaInicial
txtTarjetaFinal
tblTarjetas
lblCajasFaltantes

Dependencias
CapturaTarjetasController

Clase 5 [CierreEstacionFinDiaController] 04 Cierre de estación fin de día controller

Análisis de Clases

Nombre de la Clase	CierreEstacionFinDiaController		
Descripción	Clase controlador		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad a la transacción cierre de estación fin de día		
Atributos	Descripción	Tipo	Valor
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	private	AnchorPane
btnAceptar	Botón que realiza la acción de la transacción	private	SButton
btnSalir	Botón que realiza la acción de cancelar la transacción llevándonos a la Pantalla principal de ventanilla	private	SButton
imgBtnAceptar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
imgBtnCancelar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
modal	Modal sobre el cual se pintaran los mensajes de erros, confirmación, Información, etc... De cada pantalla	private	SolserModalSameStage

servicio	Servicio común de la caja	private	CajaService
----------	---------------------------	---------	-------------

Tabla 16 Inventario de clases CierreEstacionFinDiaController

Diseño de Clases

Métodos y Funciones

public void initialize(URL location, ResourceBundle resources)

public void resetView()

private void validaCueadreControllInterno()

private void validaRembPorRecibir()

private void validaDotaPorEntregar()

private void validaCajerosMixtosSinCierre()

private void validaCajeroConPrivilegio()

private void validaConcentracionesTombola()

private void validaTotalEfectivoCajeros()

private void guardaResguaroTombolaHist()

private void confirmarCierre()

private void ejecutarTransaccion()

Objetos (Instancias)

CajaService()

ReqFnCtaContable()

ReqDotaRemPendientes()

ReqCajMixSinCierre()

ReqCajConPriv()

ReqConcentracionTombola()

ReqConcentracionTombola()

ReqArqueoCajero()

ReqActualizaSusuarios()

SusuariosDTO()

VDiariosDTO()

Dependencias

CierreEstacionFinDiaController

Clase 6 [CapturarConcentracionTombolaController] 46 Captura de concentración de Tómbola controller

Análisis de Clases

Nombre de la Clase	CapturarConcentracionTombolaController		
Descripción	Clase controlador		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad a la transacción captura concentración en tómbola		
Atributos	Descripción	Tipo	Valor
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	Private	AnchorPane
btnAceptar	Botón que realiza la acción de la transacción	Private	SButton
btnSalir	Botón que realiza la acción de cancelar la transacción llevándonos a la * pantalla principal de ventanilla	Private	SButton
imgBtnAceptar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	Private	ImageView
imgBtnCancelar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	Private	ImageView
imgBtnCancelar	Imagen para el botón cancelar que se reemplazara en tiempo de ejecución	Private	ImageView
modal	Modal sobre el cual se pintaran los	Private	SolserModalSameStage

	mensajes de erros, confirmación, * información, etc... De cada pantalla		
cajeroServicio	Servicio común del cajero principal	Private	CajeroPrincipalService
arqueo;	-	Private	ArqueoCajero
txtFolio	-	Private	STextField
txtImpConcentracion	-	Private	STextField
controles;	-	Private	LinkedList<Node>
tombola	-	Private	ResTombola
desglose	-	Private	RetornoDesglose

Tabla 17 Inventario de clases CapturarConcentracionTombolaController

Diseño de Clases

Métodos y Funciones

public void initialize(URL location, ResourceBundle resources)

public void resetView()

private void llamaDesglose()

private Callback arqueoWarningCallback = callback ->

Objetos (Instancias)

CajeroPrincipalService()

ResTombola()

DesgloceDeMonedaNacionalDTO()

RetornoDesglose()

Callback

RetornoDesglose

ReqTombola

Dependencias

CapturarConcentracionTombolaController

Clase 7 [SolicitudDeDotacionBilleteController] 27 Solicitud de dotación billete

Análisis de Clases

Nombre de la Clase	SolicitudDeDotacionBilleteController		
Descripción	SOLICITUD DE DOTACIÓN BILLETE		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad a la transacción		
Atributos	Descripción	Tipo	Valor
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	private	AnchorPane
btnAceptar	Botón que realiza la acción de la transacción	Private	SButton
btnSalir	Botón que realiza la acción de cancelar la transacción llevándonos a la Pantalla principal de ventanilla	private	SButton
imgBtnAceptar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	Private	ImageView
imgBtnCancelar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	Private	imgBtnCancelar
modal	Modal sobre el cual se pintaran los mensajes de erros, confirmación, información, etc... De cada pantalla	private	SolserModalSameStage

servicio	Servicio común de la caja	private	CajaService
retorno	-	private	RetornoDesglose
arqueo	-	private	ArqueoCajero
generalRestCallService	-	private	GeneralRestCallService
secuencia	-	private	GeneralRestCallService
txtimporteDotacion	-	private	STextField
txtefectivoRecibido	-	private	STextField

Tabla 18 Inventario de clases SolicitudDeDotacionBilleteController

Diseño de Clases

Métodos y Funciones

public void initialize(URL location, ResourceBundle resources)

public void resetView()

private void mostrarMensaje(final MessageConfigs mensajeUsuario, String numFolio,
final Callback callback)

private ReqValidaDatosGuardados llenarValidaDatosguardados()

private ReqAfectaTablas llenarAfectaTablas()

private ReqSolicitudDotacionBillete llenarBeanDeTicket(ReqAfectaTablas req,

ReqValidaDatosGuardados reqValidaDatosGuardados)

private void mostrarFolio()

Objetos (Instancias)

CajaService()

GeneralRestCallService()

ReqValidaDatosGuardados()

ReqAfectaTablas()

ReqSolicitudDotacionBillete()

HashMap<Integer, String>()

ReqSoldoCta()

DesgloceDeMonedaNacionalDTO()

ReqMensaje()

int[1]

Dependencias

SolicitudDeDotacionBilleteController

Clase 8 [SolicitudDeDotacionDeBilleteBusinessImpl] Solicitud de dotación de billete businessImpl

Análisis de Clases

Nombre de la Clase	SolicitudDeDotacionBilleteBusinessImpl		
Descripción	Extends del controlador		
Responsabilidades	Implementa funciones heredadas del controlador		
Atributos	Descripción	Tipo	Valor
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	private	AnchorPane
btnAceptar	Botón que realiza la acción de la transacción	Private	SButton
btnSalir	Botón que realiza la acción de cancelar la transacción llevándonos a la Pantalla principal de ventanilla	private	SButton
imgBtnAceptar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	Private	ImageView
imgBtnCancelar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	Private	imgBtnCancelar

Tabla 19 Inventario de clases SolicitudDeDotacionBilleteBusinessImpl

Diseño de Clases**Métodos y Funciones**

```
public ResSoldoCta spSoldoCtaWeb(ReqSoldoCta req) throws  
BusinessException
```

```
public RespuestaBase imprimeTicket(ReqImprimeTicketSolicitudDotacionBillete  
req) throws BusinessException
```

Objetos (Instancias)

Sistema

Usuario

afectaTablas

spSoldoCtaWeb

certificadoraBusiness

ReqSistema

RespuestaBase

ReqSistema

Estado

N/A

Dependencias

SolicitudDeBilleteBusinessImpl

Clase 9 [SolicitudDeDotacionDeBilleteBusiness] Solicitud de dotación de billete Bussines

Análisis de Clases

Nombre de la Clase	SolicitudDeDotacionDeBilleteBusiness		
Descripción	Interfaz		
Responsabilidades	Imprime Ticket de la solicitud de dotación de billete		
Atributos	Descripción	Tipo	Valor
sistema	-	private	SistemaBusiness
usuario	-	private	UserSession
afectaTablas	-	private	AfectaTablasBusiness
spSoldoCtaWeb	-	private	CajeroPrincipalWSDAO
certificadoraBusiness	-	private	CertificadoraBusiness
storedProcedureDao	-	private	StoredProcedureDao
vdesgmon	-	private	VdesgmonDao
reverso	-	private	ReversoBusiness

Tabla 20 Inventario de clases SolicitudDeDotacionDeBilleteBusiness

Diseño de Clases

Métodos y Funciones

```
private void iniciaDaos()
```

```
public ResSoldoCta spSoldoCtaWeb(ReqSoldoCta req) throws BusinessException
```

```
public RespuestaBase solicitudBilleteBusiness(ReqSolicitudDotacionBillete req)
```

```
private ReqReverso llenarReverso(ReqSolicitudDotacionBillete req)
```

Objetos (Instancias)

```
CajaService
```

```
CajeroPrincipalService
```

Dependencias

```
SolicitudDeDotacionDeBilleteBusiness
```

Clase 10 [EliminacionDeSobrantesDeCajaController] 20 ELIMINACION DE SOBRANTES DE CAJA CONTROLLER

Análisis de Clases

Nombre de la Clase	EliminacionDeSobrantesDeCajaController		
Descripción	Clase controlador del módulo eliminación de sobranante de caja		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad al modulo		
Atributos	Descripción	Tipo	Valor
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	private	AnchorPane
btnAceptar	Botón que realiza la acción de la transacción	private	SButton
btnSalir	Botón que realiza la acción de cancelar la transacción llevándonos a la Pantalla principal de ventanilla	private	SButton
imgBtnAceptar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
imgBtnCancelar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
modal	Modal sobre el cual se pintaran los mensajes de erros, confirmación, Información, etc... De cada pantalla	private	SolserModalSameStage

lblNombreNomina	-	private	Label
lblImporte	-	private	Label
lblEfectivoPagar	-	private	Label
txtNoNomina	-	private	TextField
txtNoReferencia	-	private	STextField
txtImporte	-	private	STextField
txtEfectivoAPagar	-	private	STextField
sobrante	-	private	ReqSobranteCaja
desglose	-	private	RetornoDesglose

Tabla 21 Inventario de clases EliminacionDeSobrantesDeCajaController

Diseño de Clases

Métodos y Funciones

```
public void initialize(URL location, ResourceBundle resources) public void resetView()
```

```
public void resetView()
```

```
public void ejecutar()
```

```
private ChangeListener<String> txtNoNominaChangeListener = (ObservableValue<?  
extends String> observable,
```

```
String oldValue, String newValue)
```

```
private ChangeListener<String> txtNoReferenciaChangeListener =  
(ObservableValue<? extends String> observable,
```

```
String oldValue, String newValue)
```

```
private void habilitarCampos(boolean activo)
```

```
private void habilitarCamposReferencia(boolean activo)
```

Objetos (Instancias)

DesgloceDeMonedaNacionalDTO()

ArrayList<Integer>()

panelNavegable

Dependencias

EliminacionDeSobrantesDeCajaController

Clase 11 [FaltantePorRoboAsaltoController] 45 Faltante por robo o asalto

Análisis de Clases

Nombre de la Clase	FaltantesPorRoboAsaltoController		
Descripción	Clase principal		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad al modulo		
Atributos	Descripción	Tipo	Valor
panelNavegable	Método de la instancia Paneles y root de la pantalla	Privado	AnchorPane
btnAceptar	Botón que realiza la acción transacción (Aceptar)	Privado	SButton
btnSalir	Botón que realiza la acción de cancelar la ventanilla	Privado	SButton
imgBtnAceptar	Imagen para el botón aceptar que se reemplaza en tiempo de ejecución	Privado	ImageView
imgBtnCancelar	Imagen para el botón cancelar	Privado	ImageView
lblNombreNomina	Inputs de las propiedades de la clase	Privado	Label
txtNoNomina	Inputs de las propiedades de la clase	Privado	STextField
txtImporte	Inputs de las propiedades de la clase	Privado	STextField
txtEfectivo	Inputs de las propiedades de la clase	Privado	STextField
servicio	-	Privado	CajaService
servicioCP	-	Privado	CajeroPrincipalService

retorno	-	Privado	RetornoDesglose
arqueo	-	Privado	ArqueoCajero
secuenciaError	-	Privado	String
usuarioValido	-	Privado	String
secuencia	-	Privado	String

Tabla 22 Inventario de clases FaltantesPorRoboAsaltoController

Diseño de Clases

Métodos y Funciones

```

public class FaltantePorRoboAsaltoController

public void initialize(URL location, ResourceBundle resources)

public void resetView()

public void ejecutar()

private ChangeListener<String>

private void habilitar(boolean activo)

private void ejecutarTran()

```

Objetos (Instancias)

```

afectaTabla

vJourDiarDTO

vArqueoDTO

vDesgmonDTO

grabarConFalSuc

```

Dependencias

```

FaltantePorRoboAsaltoController

```

Clase 12 [SobranesDeCajaController] 19 Sobrante de caja

Análisis de Clases

Nombre de la Clase	SobranesDeCajaController		
Descripción	Controlador sobrante de caja		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad al modulo		
Atributos	Descripción	Tipo	Valor
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	FXML	-
btnAceptar	Botón que realiza la acción de la transacción	FXML	-
btnSalir	Botón que realiza la acción de cancelar la transacción llevándonos a la Pantalla principal de ventanilla	FXML	-
imgBtnAceptar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	FXML	-
imgBtnCancelar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	FXML	-
modal	Modal sobre el cual se pintaran los mensajes de erros, confirmación,	FXML	-

	Información, etc... De cada pantalla		
--	---	--	--

Tabla 23 Inventario de clases SobrantesDeCajaController

Diseño de Clases

Métodos y Funciones

public void initialize(URL location, ResourceBundle resources)

public void resetView()

public void ejecutar()

private ChangeListener<String>

private void habilitarCampos(boolean activo)

private void mostrarMensaje(final MessageConfigs mensajeUsuario, String numFolio,
final Callback callback)

Objetos (Instancias)

CajaService

CajeroPrincipalService

Dependencias

SobrantesDeCajaController

Clase 13 [DevolucionACajaEnGeneralController] 06 Devolución a caja en general controller

Análisis de Clases

Nombre de la Clase	DevolucionACajaEnGeneralController		
Descripción	Clase controlador de la transacción Dotación a caja en general controller		
Responsabilidades	Devolución a caja general, es cuando un cajero mixto requiere regresar efectivo excedente al cajero principal		
Atributos	Descripción	Tipo	Valor
panelNavegable	-	Private	panelNavegable
btnAceptar	-	Private	SButton
btnSalir	-	Private	SButton
imgBtnAceptar	-	Private	ImageView
imgBtnCancelar	-	Private	ImageView
modal	-	Private	SolserModalSameStage
cajaService	-	Private	CajaService
servicioDevolucionCajaGeneral;	-	Private	DevolucionCajaGeneralService
generalRestCallService	-	Private	GeneralRestCallService
servicioDesgloceDeMonedaNacionalModal	-	Private	DesgloceDeMonedaNacionalModalService
transaccionDTO	-	Private	TransaccionDTO
vDiariosDTO	-	Private	VDiariosDTO
arqueoSajero	-	Private	ArqueoCajero
retornoDesglose	-	Private	RetornoDesglose
txtnoFolio	-	Private	STextField
txtimporteDevolucion	-	Private	STextField
txtefectivoDevuelto	-	Private	STextField

Tabla 24 Inventario de clases DevolucionACajaEnGeneralController

Diseño de Clases

Métodos y Funciones

```
public void initialize(URL location, ResourceBundle resources),
final Callback callback)

public void resetView()

private void validaExistencialtran(final int transaccion)

private void consVdiariosEstadoPorRef(final String tranro, final String folio, final String
estado)

private void tranRequiereHuellaGerente(final String proceso, final String transaccion)

private void concentraDotacionCajaGeneral(final RetornoDesglose retornoDesgloseP, final
String montoDot, final String fecha, final String numfolioServicio, final String
numfolioSucursal)

private void consultaFechaFolio(final String numfolio)

private void mostrarMensaje(final MessageConfigs mensajeUsuario, final Callback
callback)

private boolean validarFolio()

private void disableCopyPaste()

private void formateaCampos()

private void inicializaCampos()

private Map<Integer, String> generaDesgloseTicket(final ArqueoCajero arqueo)

private String generaEspacios(int cadenalzquierda, String cadenaDerecha)
```

Objetos (Instancias)

ReqAfectaTablas()

ArrayList<Integer>()

ReqDevCajaGen()

DesgloceDeMonedaNacionalDTO()

DesgloceDeMonedaNacionalDTO()

ReqEliminaVtempemision()

StringBuilder()

DecimalFormat("#0.00")

DesgloceDeMonedaNacionalDTO()

Dependencias

DevolucionACajaEnGeneralController

Clase 14 DesgloceDeMonedaNacionalModalController] COGEN01 Desglose de moneda nacional modal controller
Análisis de Clases

Nombre de la Clase	DesgloceDeMonedaNacionalModalController		
Descripción	Clase controlador de la transacción Desglose de moneda nacional		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad al modulo		
Atributos	Descripción	Tipo	Valor
desgloceInicial	-	private	BigDecimal
diferenciaActual	-	private	BigDecimal
acumulado	-	private	BigDecimal
acumuladoEgreso	-	private	BigDecimal
importeActual	-	private	BigDecimal
egresoActual	-	private	BigDecimal
tipoDesglose	-	private	TipoDesglose
transaccion	-	private	String
panelNavegable	-	private	AnchorPane
lblDenominaciones	-	private	Label
txtCantidadInicial	-	private	STextField
lblDiferencia	-	private	Label
txtDiferencia	-	private	STextField
txtCantidadIngreso1000	-	private	STextField
txtTotalIngreso1000	-	private	STextField
txtCantidadIngreso500	-	private	STextField
txtTotalIngreso500	-	private	STextField

txtCantidadIngreso200	-	private	STextField
txtTotalIngreso200	-	private	STextField
txtCantidadIngreso100	-	private	STextField
txtTotalIngreso100	-	private	STextField
txtCantidadIngreso50	-	private	STextField
txtTotalIngreso50	-	private	STextField
txtCantidadIngreso20	-	private	STextField
txtTotalIngreso20	-	private	STextField
txtCantidadIngresoM	-	private	STextField
txtTotalIngresoM	-	private	STextField
txtCantidadEgreso100 0	-	private	STextField
txtTotalEgreso1000	-	private	STextField
txtCantidadEgreso500	-	private	STextField
txtTotalEgreso500	-	private	STextField
txtCantidadEgreso200	-	private	STextField
txtTotalEgreso200	-	private	STextField
txtCantidadEgreso100	-	private	STextField
txtTotalEgreso100	-	private	STextField
txtCantidadEgreso50	-	private	STextField
txtTotalEgreso50	-	private	STextField
txtCantidadEgreso20	-	private	STextField
txtTotalEgreso20	-	private	STextField
txtCantidadEgresoM	-	private	STextField
txtTotalEgresoM	-	private	STextField

txtTotalIngreso	-	private	STextField
txtTotalEgreso	-	private	STextField
btnAceptarModal	-	private	SButton
imgBtnAceptar	-	private	ImageView
btnSalirModal	-	private	SButton
imgBtnSalir	-	private	ImageView
txtEgresos	-	private	STextField
modalSS	-	private	SolserModalSameStage
arqueoSajero	-	private	ResArqueoCajero
retornoDesglose	-	private	RetornoDesglose
templateEmpleado	-	private	TemplateHuella
numCajeroPrincipal	-	private	String
validaHuella	-	private	boolean
validaHuellaCajeroPrincipal	-	private	boolean
validaHuellaCajeroPrincipalGerente	-	private	boolean

Tabla 25 Inventario de clases DesgloceDeMonedaNacionalModalController

Diseño de Clases

Métodos y Funciones

```
public void initialize(URL location, ResourceBundle resources)
```

```
public void reload()
```

```
public void setEntrada(DesgloceDeMonedaNacionalDTO entrada)
```

```
private void initDenominaciones()
```

```
private void obtieneArqueoCajero(String numCajero)
```

```
private boolean obtieneTransaccionParametro(String valorParametro)
```

```
public Optional<Boolean> getResponse()

private void btnSalirListener()

private void btnAceptarListener()

private void validaEgreso()

private boolean validaDesgloseEgreso()

private void procesaAmbos()

private void mostrarMensaje(final MessageConfigs mensajeUsuario,
                             final Callback callback)

private void mostrarTransaccionExitosa(TipoDesglose tipoDesglose)

private void validaHuellaCajeroPGerente(Callback callbackMensajeH)

private void consultaParametroWeb(String valor, Callback callback)
```

Objetos (Instancias)

```
BigDecimal

DesgloseDeMonedaNacionalModalService()

ResArqueoCajero()

RetornoDesglose()

TemplateHuella()

DecimalFormat

Image
```

ReqArqueoCajero()

ReqConsultaParametro()

BigDecimal(0)

Dependencias

DesgloceDeMonedaNacionalModalController

Clase 15 [DotacionDeCajaGeneralController] 05 Dotación de caja general controller

Análisis de Clases

Nombre de la Clase	DotacionDeCajaGeneralController		
Descripción	Controlador Dotación de caja general controller		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad al modulo		
Atributos	Descripción	Tipo	Valor
folioDota	-	private	String
Desglose	-	private	RetornoDesglose
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	private	AnchorPane
btnAceptar	Botón que realiza la acción de aceptar	private	SButton
btnSalir	Botón que realiza la acción de salir	private	SButton
imgBtnAceptar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
imgBtnCancelar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
modal	Modal sobre el cual se pintaran los mensajes de erros, confirmación, Información, etc... De cada pantalla	private	SolserModalSameStage
servicio	-	private	DotaCajaGralService

generalRestCallService	-	private	GeneralRestCallService
txtNumeroFolio	-	private	STextField
txtImporteDotacion	-	private	STextField
txtEfectivoRecibido	-	private	STextField
txtImporteFaltante	-	private	STextField
txtImporteSobrante	-	private	STextField
secuencia	-	private	STextField

Tabla 26 Inventario de clases DotacionDeCajaGeneralController

Diseño de Clases

Métodos y Funciones

public void initialize(URL location, ResourceBundle resources)

public void resetView()

private void mostrarMensaje(final MessageConfigs mensajeUsuario, final Callback callback)

private Map<Integer, String> generaDesgloseTicket(final ArqueoCajero arqueo)

private String generaEspacios(int cadenalzquierda, String cadenaDerecha)

private void inicializaCampos()

private void formateaCampos()

private ChangeListener<String> listener = (ObservableValue<? extends String> observable, String oldValue, String newValue)

public void btnAceptarListener(ActionEvent event)

private void btnSalirListener(ActionEvent event)

Objetos (Instancias)

DotaCajaGralService()

GeneralRestCallService()

HashMap<Integer, String>()
ReqValFolioDotaCajaGen()
DesgloceDeMonedaNacionalDTO()
ReqRecepDotaCajaGen()
ReqAfectaTablas()
ReqDotacionCajaGeneral()
ArrayList<Integer>()
ReqMensaje()

Dependencias

DotacionDeCajaGeneralController

Clase 16 [TotalesCifrasSucursalController] 21 Totales cifras sucursal controller

Análisis de Clases

Nombre de la Clase	TotalesCifrasSucursalController		
Descripción	Controlador Totales cifras sucursal controller		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad al modulo		
Atributos	Descripción	Tipo	Valor
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	private	AnchorPane
btnAceptar	Botón que realiza la acción de la transacción	private	SButton
btnSalir	Botón que realiza la acción de cancelar la transacción llevándonos a la Pantalla principal de ventanilla	private	SButton
imgBtnAceptar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
imgBtnCancelar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
modal	Modal sobre el cual se pintaran los mensajes de erros, confirmación,	private	SolserModalSameStage

	Información, etc... De cada pantalla		
servicio	Servicio común de la caja	private	CajaService
txtOperador	-	private	STextField
cmbSistema	-	private	SComboBox<String>
tblTotales	-	private	STableView<TotalCifrasS ucursalTableModelDTO>
totalesModel	-	private	TotalCifrasSucursalTable ModelDTO

Tabla 27 Inventario de clases TotalesCifrasSucursalController

Diseño de Clases

Métodos y Funciones

```
public void initialize(URL location, ResourceBundle resources)
```

```
public void resetView()
```

```
private void tblTotalesConfig()
```

Objetos (Instancias)

```
CajaService();
```

```
ReqTotalesCifrasSucursal()
```

Dependencias

```
TotalesCifrasSucursalController
```

Clase 17 [SolicitudDeDotacionMorrallaController] 28 Solicitud de dotación de morralla

Análisis de Clases

Nombre de la Clase	SolicitudDeDotacionMorrallaController		
Descripción	Solicitud de dotación de morralla		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad al modulo		
Atributos	Descripción	Tipo	Valor
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	private	AnchorPane
btnAceptar	Botón que realiza la acción de la transacción	private	SButton
btnCancelarModal		private	SButton
btnSalir		private	SButton
imgBtnAceptar		private	ImageView
imgBtnCancelar		private	ImageView
modal		private	SolserModalSameStage
servicio		private	CajaService
cajeroPrincipalService		private	CajeroPrincipalService
retorno		private	RetornoDesglose
arqueo		private	ArqueoCajero
generalRestCallService		private	GeneralRestCallService
secuencia		private	String
txtImporteDotacion		private	STextField
txtEfectivoRecibido		private	STextField

Tabla 28 Inventario de clases SolicitudDeDotacionMorrallaController

Diseño de Clases

Métodos y Funciones

```
public void initialize(URL location, ResourceBundle resources)

public void resetView()

private void mostrarMensaje(final MessageConfigs mensajeUsuario, String numFolio, final
Callback callback)

private void mostrarFolio()

private ReqAfectaTablas llenarAfectaTablas()

private ReqValidaDatosGuardados llenarValidaDatosguardados()

private ReqSolicitudDotacionBillete llenarBeanDeTicket
```

Objetos (Instancias)

```
CajaService();

CajeroPrincipalService();

GeneralRestCallService();

ReqSoldoCta();

ReqAfectaTablas();

ReqValidaDatosGuardados();

ReqSolicitudDotacionBillete();

HashMap<Integer, String>();

DesgloceDeMonedaNacionalDTO();

ReqValidaDatosGuardados();
```

Dependencias

```
SolicitudDeDotacionMorrallaController
```

Clase 18 [PagoDeBilleteAutenticoController] 51 Pago de billete autentico controller

Análisis de Clases

Nombre de la Clase	PagoDeBilleteAutenticoController		
Descripción	Controlador Pago de billete autentico		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad al modulo		
Atributos	Descripción	Tipo	Valor
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	private	AnchorPane
btnAceptar	Botón que realiza la acción de la transacción	private	SButton
btnSalir	Botón que realiza la acción de cancelar la transacción llevándonos a la pantalla principal de ventanilla	private	SButton
imgBtnAceptar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
imgBtnCancelar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
modal	-	private	SolserModalSameStage
cajeroPrincipalServicio	-	private	CajeroPrincipalService
generalRestCallService	-	private	GeneralRestCallService

vbCampos	-	private	VBox
txtNoCuenta	-	private	STextField
secuencia	-	private	String
fechaSistema	-	private	Date
tipoPago	-	private	int
gerente	-	private	String
numeroCuenta	-	private	String
tipoCuenta	-	private	String
formaPago	-	private	PagoBilleteAutenticoModalDTO
desglose	-	private	RetornoDesglose
retorno	-	private	RetornoDesglose
arqueo	-	private	ArqueoCajero

Tabla 29 Inventario de clases PagoDeBilleteAutenticoController

Diseño de Clases

Métodos y Funciones

```
public void initialize(URL location, ResourceBundle resources)

public void resetView()

private void mostrarMensaje(final MessageConfigs mensajeUsuario, final Callback callback)

private void validaNoCuenta()

private void actualizaPiezasBym()

private void llamaDesglose()

private void mostrarMensaje(final MessageConfigs mensajeUsuario, String numFolio, final Callback
callback)

private void getCapturaReciboModal()

private String generaCodigoSamBPF()
```

Objetos (Instancias)

```
PagoBilleteAutenticoModalDTO();

CajeroPrincipalService();

GeneralRestCallService();

ReqSpConsSdos2Web();

ReqSpActualizaPiezaBymWeb();

DesgloceDeMonedaNacionalDTO();

SimpleDateFormat

ReqArqueoCajero();
```

Dependencias

```
PagoDeBilleteAutenticoController
```

Clase 19 [PagoDeBilleteAutenticoController] Pago de billete autentico controller
Análisis de Clases

Nombre de la Clase	PagoDeBilleteAutenticoController		
Descripción	Controlador Pago de billete autentico		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad al modulo		
Atributos	Descripción	Tipo	Valor
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	private	AnchorPane
btnAceptar	Botón que realiza la acción de la transacción	private	SButton
btnSalir	Botón que realiza la acción de cancelar la transacción llevándonos a la pantalla principal de ventanilla	private	SButton
imgBtnAceptar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
imgBtnCancelar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
modal	-	private	SolserModalSameStage
cajeroPrincipalServicio	-	private	CajeroPrincipalService
generalRestCallService	-	private	GeneralRestCallService

vbCampos	-	private	VBox
txtNoCuenta	-	private	STextField
secuencia	-	private	String
fechaSistema	-	private	Date
tipoPago	-	private	int
gerente	-	private	String
numeroCuenta	-	private	String
tipoCuenta	-	private	String
formaPago	-	private	PagoBilleteAutenticoModalDTO
desglose	-	private	RetornoDesglose
retorno	-	private	RetornoDesglose
arqueo	-	private	ArqueoCajero

Tabla 30 Inventario de clases PagoDeBilleteAutenticoController

Diseño de Clases

Métodos y Funciones

```
public void initialize(URL location, ResourceBundle resources)
```

```
public void resetView()
```

```
private void mostrarMensaje(final MessageConfigs mensajeUsuario, final Callback callback)
```

```
private void validaNoCuenta()
```

```
private void actualizaPiezasBym()
```

```
private void llamaDesglose()
```

```
private void mostrarMensaje(final MessageConfigs mensajeUsuario, String numFolio, final Callback callback)
```

```
private void getCapturaReciboModal()
```

```
private String generaCodigoSamBPF()
```


Objetos (Instancias)

PagoBilleteAutenticoModalDTO();

CajeroPrincipalService();

GeneralRestCallService();

ReqSpConsSdos2Web();

ReqSpActualizaPiezaBymWeb();

DesgloceDeMonedaNacionalDTO();

SimpleDateFormat

ReqArqueoCajero();

Dependencias

PagoDeBilleteAutenticoController

Clase 20 [ReimpresionDeTicketController] 11 Reimpresión de ticket controller

Análisis de Clases

Nombre de la Clase	ReimpresionDeTicketController		
Descripción	Clase controlador de la transacción Reimpresión de Ticket		
Responsabilidades	Esta Clase se encargara de Reimprimir Tickets cuando el tipo de transacción así lo permita		
Atributos	Descripción	Tipo	Valor
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	private	AnchorPane
btnAceptar	Botón que realiza la acción de la transacción	private	SButton
btnSalir	Botón que realiza la acción de cancelar la transacción llevándonos a la pantalla principal de ventanilla	private	SButton
imgBtnAceptar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
imgBtnCancelar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
modal	-	private	SolserModalSameStage
CajeroPrincipalServicio	-	private	CajeroPrincipalService
txtnoFolio	-	Private	STextField
txtnoNomina	-	Private	STextField

Tabla 31 Inventario de clases ReimpresionDeTicketController

Diseño de Clases

Métodos y Funciones

public void initialize(URL location, ResourceBundle resources)

public void resetView()

public void testWebServices()

Objetos (Instancias)

ReqVDesgmon()

VdesgmonDTO()

ReqReimpresion()

RespuestaBase()

Callback()

Dependencias

ReimpresionDeTicketController

Clase 21 [EliminacionFaltantesDeCajaController] 18 Eliminación faltantes de caja controller

Análisis de Clases

Nombre de la Clase	EliminacionFaltantesDeCajaController		
Descripción	Clase controlador de la transacción Eliminación faltantes de caja controller		
Responsabilidades	Controlador de la clase encargado de dar funcionalidad al modulo		
Atributos	Descripción	Tipo	Valor
panelNavegable	Panel principal y root de la pantalla sobre la cual está armado	private	AnchorPane
btnAceptar	Botón que realiza la acción de la transacción	private	SButton
btnSalir	Botón que realiza la acción de cancelar la transacción llevándonos a la pantalla principal de ventanilla	private	SButton
imgBtnAceptar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
imgBtnCancelar	Imagen para el botón aceptar que se reemplazara en tiempo de ejecución	private	ImageView
modal	-	private	SolserModalSameStage
servicio	-	private	CajaService

lblNombreNomina	-	private	Label
txtNoNomina	-	private	STextField
txtReferencia	-	private	STextField
txtMontoAdeudo	-	private	STextField
txtImporte	-	private	STextField
txtEfectivoRecibido	-	private	STextField
txtDevEfectivo	-	private	STextField
adeudo	-	private	float
sldfaltantes	-	private	String
dFechaReg	-	private	Date
sSecuencia	-	private	String
sFecha	-	private	String
sNoNomina	-	private	String
TotalIngreso	-	private	String
arqueoIngreso	-	private	ArqueoCajero
arqueoEgreso	-	private	ArqueoCajero

Tabla 32 Inventario de clases EliminacionFaltantesDeCajaController

Diseño de Clases

Métodos y Funciones

```
public void initialize(URL location, ResourceBundle resources)
```

```
public void resetView
```

```
public void ejecutar()
```

```
private ChangeListener<String> inputTextChangeListener = (ObservableValue<? extends String> observable, String oldValue, String newValue)
```

```
private void consultaFaltante()
```

```
private void fnGuardarAbono()
```

```
private void fnReverso()
```

```
private void deshabilitaCortaryPegar()
```

```
private void validaDesgloseyhuella()
```

Objetos (Instancias)

```
CajaService();
```

```
ReqConsultaEmpleado();
```

```
CajaEliminaFaltantesService();
```

```
ReqConsultaFaltanteEmp();
```

```
CajaEliminaFaltantesService();
```

```
ReqAbonoFaltante();
```

```
ArrayList<Integer>();
```

```
CajaEliminaFaltantesService();
```

```
ReqGrabaReversoSuc();
```

```
CajaEliminaFaltantesService();
```

```
DesgloceDeMonedaNacionalDTO();
```

Dependencias

```
EliminacionFaltantesDeCajaController
```

MATRIZ DE TRAZABILIDAD

Objetivo del documento

El propósito de este documento es presentar los artefactos de las clases del proyecto CPlasu OLA 4, mostrando un listado de componentes (Import). La matriz de trazabilidad vincula los componentes con su origen. Luego los monitorea durante la evolución del proyecto ya que cuando el proyecto es muy grande o complejo es difícil saber que clase o diseños cubren cada una de las especificaciones o requerimientos del proyecto.

Componente vs Componente	AperturaDeDia CAJACONTROLLER	CierreTemporalEstacionController	AperturaDeEstacionController	CierreEstacionFinDiaController	Dotación de caja general	Devolución de caja general	Reimpresión de Ticket	Recepción de dotación	Faltante en caja	Eliminación de faltante de caja	SobrantesDeCajaController	EliminaciónDeSobrantesDeCaja	TotalesCifrasSucursalController	SolicitudDeDotacionBilleteController	Solicitud de dotación de morralla	FaltantePorRoboAsaltoController	CapturaConcentracionTombolaController	Pago de billete autentico en efectivo	CapturaTarjetasController	Desglose de moneda nacional																		
mx.com.solser.controls.SButton;	X	X	X	X	X	X	X			X	X	X	X	X	X	X	X	X	X	X																		
mx.com.solser.controls.SComboBox	X									X	X	X	X	X	X	X	X	X	X	X																		
mx.com.solser.controls.STextField;	X									X	X	X	X	X	X	X	X	X	X	X																		
mx.com.solser.controls.api.NavegacionControlesService;	X	X	X	X	X	X	X			X	X	X	X	X	X	X	X	X	X	X																		
mx.com.solser.cplasu.beans.dto.Cliente;	X																																					
mx.com.solser.cplasu.beans.dto.RespuestaBase;	X			X	X	X	X			X				X	X																							
mx.com.solser.cplasu.beans.dto.altanica.altaproductos.capta	X																																					
mx.com.solser.cplasu.beans.req.ReqMensaje;	X				X	X								X	X																							
mx.com.solser.cplasu.beans.res.altanica.productocaptacion.	X																																					
mx.com.solser.cplasu.util.Constantes;	X				X	X				X	X	X	X	X	X	X	X	X	X	X																		
mx.com.solser.cplasu.caja.beans.dto.ArqueoDTO;	X					X																																
mx.com.solser.cplasu.caja.beans.dto.DepurarCampaniasDTO;	X																																					

Ilustración 16: Matriz de trazabilidad

CAPÍTULO 4. RESULTADOS Y CONCLUSIONES

La aparición de nuevas tecnologías, obligan a las organizaciones a migrar sus sistemas Heredados hacia las nuevas plataformas y tecnologías, adaptando las funcionalidades de los sistemas, para poder dar un mejor funcionamiento.

Para la realización de un proyecto de migración de un sistema se tiene que realizar la gestión del mismo apegándose a una metodología de desarrollo y una guía para su gestión.

Tras la gestión de este proyecto se han podido comprender, más a profundidad distintos conceptos relacionados con el ciclo de vida de un proyecto así como de las normas para gestionarlo.

4.1 Resultados

Para realizar la modernización y optimización del sistema del cliente se contemplaron estrategias encaminadas a lograr los beneficios y entregables en cortos plazos que agreguen valor agregado al negocio y permitan migrar los riesgos que conllevan la implementación de proyectos.

Durante el desarrollo del proyecto se fue viendo la importancia de la gestión de proyectos, basándose en metodologías para el estudio previo de todas las características del proyecto y así obtener un mejor resultado.

Los resultados obtenidos a raíz de este proyecto son los siguientes:

- Con este proyecto se consiguió el monitoreo continuo del cronograma y de la gestión del trabajo para cumplir con los tiempos estipulados. Además se logró culminar con las fases del desarrollo de módulos junto con los entregables adecuados y establecidos a la etapa correspondiente (OLA 4).
- El diseño e implementación de una nueva plataforma WEB centralizada, encamina a la modernización de las funcionalidades actuales.
- La implementación de esta propuesta cumple con los puntos del alcance mencionado anteriormente (que se describen en el primer capítulo apartado 1.7 Alcances y limitaciones), así mismo contempla la migración de las 350 funcionalidades.

El proyecto se desarrolla en base a la metodología ágil SCRUM como lo propone el cliente y además, complementa a la documentación del proyecto con algunos artefactos de la metodología SOLSER basada en CMMI nivel 3 para desarrollo (versión 1.3).

El proyecto se desarrollará utilizando los estándares de desarrollo de SOLSER verificados por el cliente.

4.2 Trabajos Futuros

Una mejora requerida en el futuro es la posibilidad de implementar las siguientes funciones:

- Adquisición de los periféricos utilizados por las Sucursales
- Adquisición de hardware para la arquitectura de solución propuesta
- Implementación del ambiente de trabajo
- El despliegue de la aplicación para ambiente de Producción

Para dar seguimiento al proyecto los desarrolladores, analistas e involucrados con el proyecto deberán informar los avances realizados. El líder de proyecto deberá comunicar a ODP los avances llevados hasta ese momento, así como cualquier incidencia que pueda ocurrir durante el ciclo de vida del proyecto. Al término de cada fase del proyecto se realizará una reunión donde participará el equipo de trabajo y en la cual se revisarán dichos informes, además de las actividades que se plasman en el cronograma de trabajo y se dará monitoreo y seguimiento de los riesgos detectados, así como nuevos riesgos que puedan surgir.

4.3 Recomendaciones

El soporte que se le da al proyecto es a través de la solución de los incidentes que se generan en la fase de pruebas de Solser y las pruebas que realiza el cliente.

- Que el BUS funcione adecuadamente durante la etapa de implementación a pruebas.
- Migración de 350 funcionalidades desarrolladas en Visual Basic a un ambiente web desarrollado bajo la tecnología de Java.
- Las funcionalidades tendrán mejor desempeño al estar implementadas bajo la tecnología Java

Como servicio de valor agregado se proporciona el uso de la herramienta JIRA, con el uso del módulo ágil para la gestión del proyecto, le asignaremos al cliente 5 usuarios para dar seguimiento al proyecto y para organizar, planear y ejecutar de forma ordenada el proyecto.

También se proporciona el uso de nuestro software de control de incidentes COS (Centro de operaciones SOLSER), donde se da seguimiento de forma automática y se guardan las pruebas y el seguimiento de los incidentes. Se proporcionan los usuarios que el cliente solicite.

ANEXOS

Este anexo corresponde a la presentación de:

Inventario de Pantallas


 Certificación CMMiN3	FDS-022 Inventario de Pantallas <V 1.2> Fecha Liberación del Formato <20/07/2011> S057-PRYE20150929-022/CPIaSu			
Historial de Cambios				
Versión	Descripción	Elaboró	Autorizó	Fecha
1.1	Se crea Archivo	Reyes Rojas Fredy Arturo	Mota León Rubén	20/03/2018

Ilustración 17: Inventario de pantalla

Plataforma:	Windows.
--------------------	----------

2 . Detalle de Pantallas

2.1 . [Actualización de CURP]

2.1.1 .Descripción General de la Pantalla

ID	001
Nombre de Pantalla	Actualización de CURP.
Funcionalidad	Permite realizar el correcto mantenimiento de "Actualización de CURP" de un Cliente Bancario.
Flujo :	ActoresFDS-006CasosUso_Actualizacion_de_CURP.
Módulo o Trx a la que Pertenece:	Consultas
Pantallas asociadas	NA
Tipo de pantalla	Normal

2.1.2 Campos de la Pantalla

Descripción del Campo	Tipo	Longitud	Valores Default	Tabla/N° de Campo
Nombre	BOOLEANO	NA	NA	NA
Cuenta/Tarjeta	BOOLEANO	NA	NA	NA
No. de Cliente	BOOLEANO	NA	NA	NA
No. de Teléfono	BOOLEANO	NA	NA	NA
Cuenta/ Tarjeta.	INT	9	NA	NA
No. de Cliente	INT	16	NA	NA

Ilustración 18: Inventario de pantalla

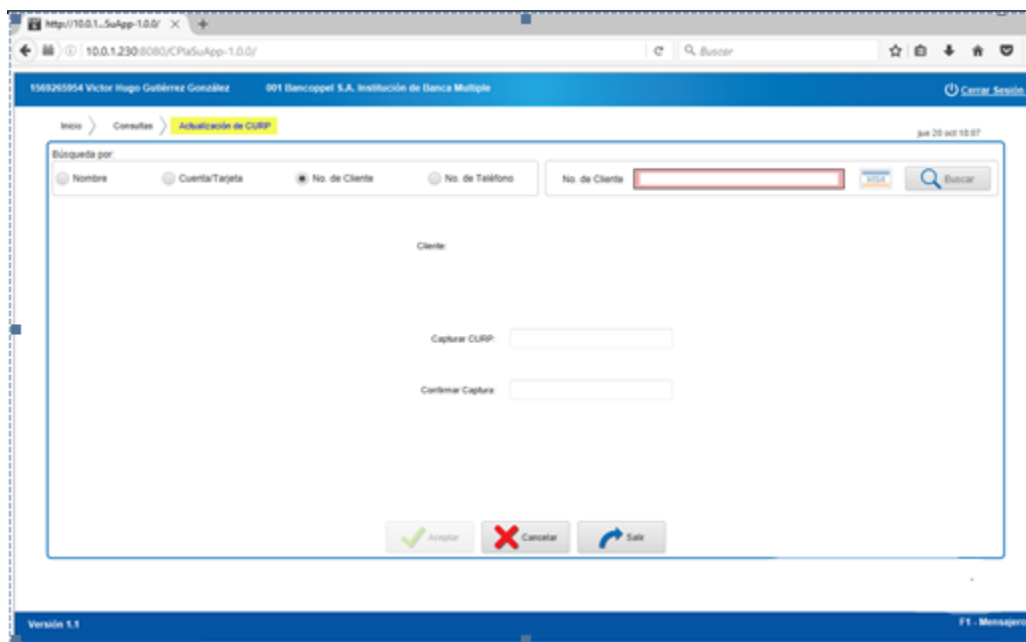


Ilustración 19: Inventario de pantalla

2.2 . [Búsqueda por Nombre]

2.2.1 .Descripción General de la Pantalla

ID	002
Nombre de Pantalla	Búsqueda por Nombre
Funcionalidad	Permite realizar la Actualización de CURP, por Nombre ingresando Apellido Paterno, Apellido Materno, Nombre(s) y Fecha de Nacimiento.
Flujo:	FDS-006CasosUso_Actualizacion_de_CURP
Módulo o Trx a la que Pertenece:	Actualización de CURP.
Pantallas a sociadas	Consultas
Tipo de pantalla	Normal

2.2.2 .Campos de la Pantalla

Descripción del Campo	Tipo	Longitud	Valores Default	Tabla/N° de Campo
Nombre del Cliente	BOOLEANO	NA	NA	NA
Razón Social	BOOLEANO	NA	NA	NA
Apellido Paterno	STRING	15	NA	NA
Apellido Materno	STRING	15	NA	NA
Primer Nombre	STRING	15	NA	NA
Segundo Nombre	STRING	15	NA	NA

Ilustración 20: Inventario de pantalla

Ilustración 21: Inventario de pantalla

Búsqueda por Nombre X


Nombre del Cliente
 Razon Social

Razon Social

Nombre del Cliente	No. de Cliente	RFC	
No existe información			

Ilustración 22: Inventario de pantalla

Modelado de datos



SOLSER
INFORMATION TECHNOLOGY
Certificación CMMiN3

FDS-026 Modelado de Datos <V 1.2>
Fecha Liberación del Formato <20/07/2011>
S057-PRYE20150929-022/CPLaSuc

Historial de Cambios

Versión	Descripción	Elaboró	Autorizó	Fecha
1.1	Se crea documento	Reyes Rojas Fredy Arturo	Pozas Naranjo Isabel	09/02/2018

Ilustración 23: Modelado de datos

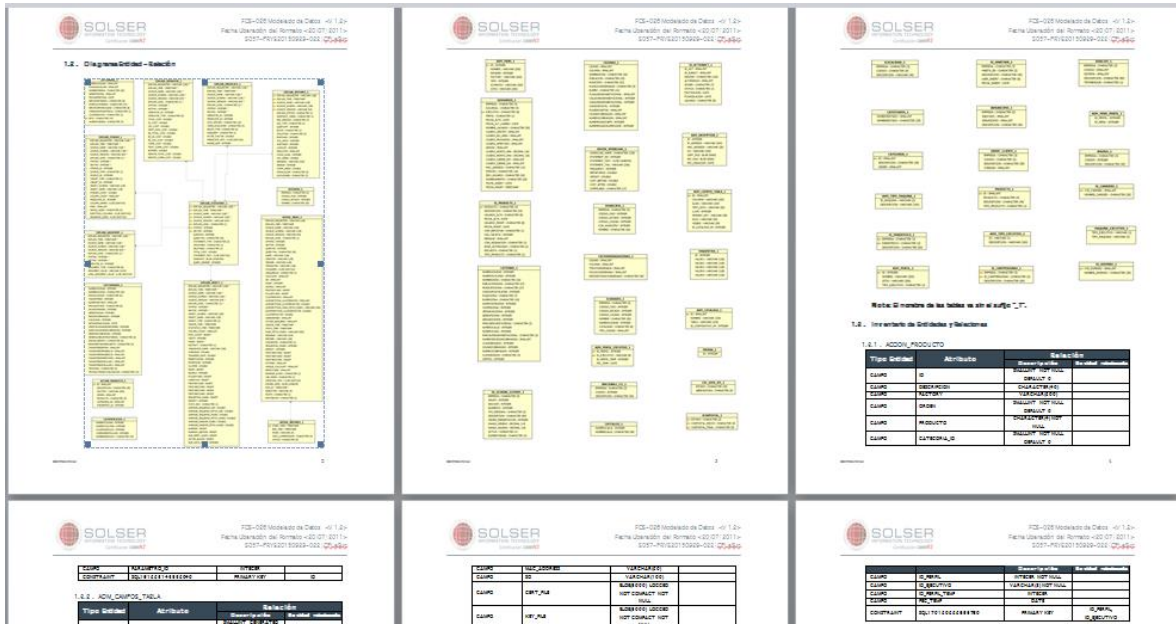


Ilustración 24: Modelado de datos

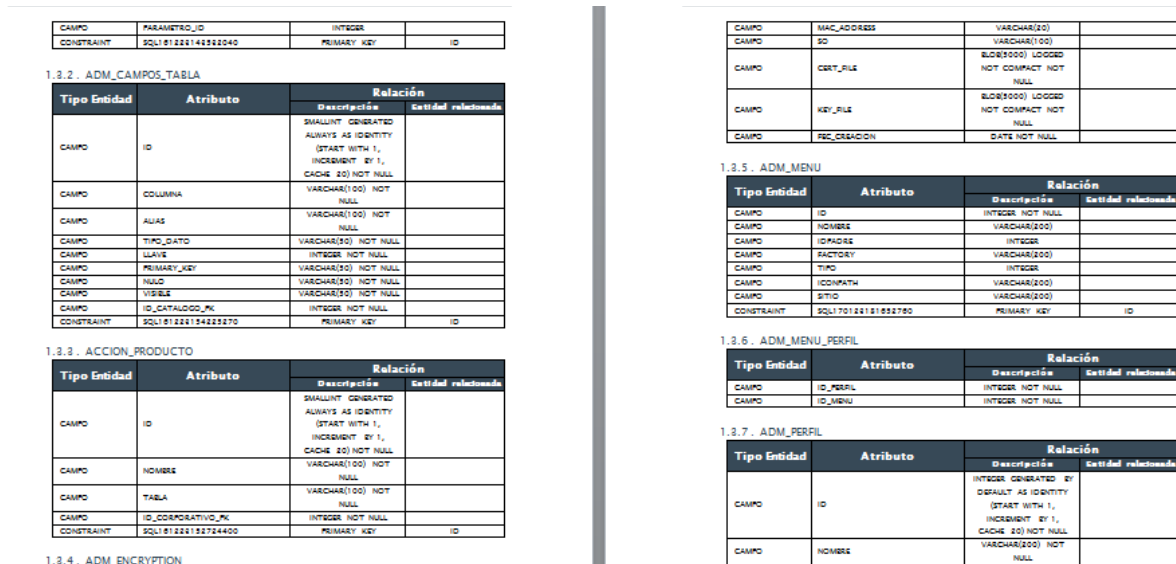


Ilustración 25: Modelado de datos

BIBLIOGRAFÍA

- 3.0, C. C. (16 de 12 de 2016). *Ingeniería del software*. Obtenido de <http://www.um.es/docencia/barzana/IAGP/lagp1.html>
- Amo, F. A. (1995). *Entornos y metodologías de programación*. España: Thomson-Paraninfo.
- Gomez, C., & Jimenez, J. (Junio de 2015). *SlideShare*. Recuperado el 05 de Febrero de 2018, de <https://es.slideshare.net/JesusJimenez123/migracion-de-sistemas-computacionales>
- Piattini, M. y. (2004). *"Análisis y diseño de aplicaciones informáticas de gestión"*. Madrid: RA-MA.
- Profit, W. (2018). *Gestiopolis*. Recuperado el 12 de Febrero de 2018, de <https://www.gestiopolis.com/terminos-de-uso/>
- Segura, M. A. (06 de 26 de 2012). *Operacionalización de variables*. Obtenido de http://bvsper.paho.org/videosdigitales/matedu/2012investigacionsalud/20120626Operacionalizacion_MoisesApolaya.pdf?ua=1
- UCIPFG. (2016). *Ucipfg Repositorio*. Obtenido de [http://www.ucipfg.com/Repositorio/MIA/MIA-01/MIA-JC/BLOQUE-ACADEMICO/Unidad-1/lecturas%20obligatorias/Guia_del_Proyecto_de_Curso_y_Ejemplo_de_un_Estudio_de_Prefactibilidad%20\(1\).pdf](http://www.ucipfg.com/Repositorio/MIA/MIA-01/MIA-JC/BLOQUE-ACADEMICO/Unidad-1/lecturas%20obligatorias/Guia_del_Proyecto_de_Curso_y_Ejemplo_de_un_Estudio_de_Prefactibilidad%20(1).pdf)

