



Reporte Final de Estadía

Diego Altamirano Suarez

CPLASU Caja



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 Information Technology

Nombre del proyecto
“CPLASU Caja”

Presenta
Diego Altamirano Suarez

Cuitláhuac, Ver., a 4 de abril de 2018.



Universidad Tecnológica del Centro de Veracruz

Universidad Tecnológica del Centro de Veracruz



Programa Educativo
Tecnologías de la Información

Nombre del Asesor Industrial
Ing. Abraham Guerrero Sierra

Nombre del Asesor Académico
MSC. López Hernández Jesús Leonardo

Jefe de Carrera
Lic. Aldaraca Juárez César

Nombre del Alumno
Altamirano Suarez Diego

AGRADECIMIENTOS

El mérito de haber terminado este ciclo no solo me pertenece -a mi sino también a aquellas personas que contribuyeron a mi formación, como son los docentes los cuales me brindaron las herramientas para poder desempeñarme en el ámbito laboral, pero también a aquellas empresas y personal de estas que me brindaron la oportunidad de experimentar y aprender de nuevas tecnologías y lo que es desempeñarse en un trabajo formalmente, cabe destacar que sin apoyo de mis familiares y amigos esto no hubiera sido posible ya que estuvieron apoyándome sin importar las consecuencias.



RESUMEN

El proyecto CPLASU Caja fue desarrollado dentro de la empresa SOLSER, dicho proyecto es una migración del sistema anterior de ventanilla, el cual fue desarrollado en Visual Basic 6.0, este documento tiene como finalidad justificar la razón de ser de la migración de los módulos reimpresión de ticket, consulta de cifras en sucursal y consulta de cuentas de saldos en captación a un nuevo lenguaje de programación como lo es Java, todo esto con el objetivo de disminuir la pérdida de datos, así como también permitir la actualización de los equipos en los cuales sea utilizado el aplicativo.



Contenido

Agradecimientos

Resumen

CAPÍTULO 1. INTRODUCCIÓN	1
1.1 Estado del Arte	1
1.2 Planteamiento del Problema	3
1.3 Objetivo general	3
1.4 Objetivos	4
1.5 Definición de variables	4
1.6 Hipótesis.....	4
1.7 Justificación del Proyecto	4
1.8 Limitaciones y Alcances	5
1.9 Solser Information Technology.....	5
CAPÍTULO 2. METODOLOGÍA.....	7
2.1 Planificación	8
2.2 SCRUM Diario	8
2.3 Trabajo de desarrollo	9
2.4 Revisión del sprint	9
2.5 Retrospectiva del sprint.....	9
CAPÍTULO 3. DESARROLLO DEL PROYECTO	10
CAPÍTULO 4. RESULTADOS Y CONCLUSIONES	19
4.1 Resultados.....	19
4.2 Trabajos Futuros.....	21
4.3 Recomendaciones	21



Glosario..... 22

Tabla de ilustraciones

Ilustración 1-1 Diagrama CMMI 3 Solser	6
Ilustración 1-2 Diagrama SCRUM Solser	7
Ilustración 2-1 Metodología SCRUM [6]	8
Ilustración 3-1 Herramienta MeisterTask	10
Ilustración 3-2 Formato para estimación de tiempo en Excel	11
Ilustración 3-3 Caso de uso consulta de saldos en cuenta de captación.	12
Ilustración 3-4 Diagrama de flujo Consulta saldos de cuenta.	13
Ilustración 3-5 Interfaz del módulo consulta de saldos en cuenta de captación.	13
Ilustración 3-6 Interfaz del módulo consulta de saldos en cuenta de captación.	14
Ilustración 3-7 Interfaz del módulo consulta de saldos en cuenta de captación.	14
Ilustración 3-8 Caso de uso del módulo Consulta cifras sucursal.	15
Ilustración 3-9 Caso de uso del módulo Reimpresión de ticket.	15
Ilustración 3-10 Pantalla del módulo Reimpresión de ticket.	16
Ilustración 3-11 Interfaz del repositorio SmartCVS	17
Ilustración 3-12 Interfaz del sistema COS	18
Ilustración 4-1 Tabla del sistema en Visual Basic	19
Ilustración 4-2 Tabla en sistema Java	20
Ilustración 4-3 Log del sistema en Java.	20
Ilustración 5-1 Log de backend	23
Ilustración 5-2 Log de Frontend.	24

CAPÍTULO 1. INTRODUCCIÓN

El proyecto se desarrolló dentro de la empresa SOLSER el cual tiene como nombre CPLASU Caja, este tiene como objetivo la migración del sistema de ventanilla de una empresa bancaria, dicho sistema es el encargado de realizar y gestionar las distintas operaciones que realizan los usuarios del banco, así como sus colaboradores.

El sistema antiguo de la empresa bancaria fue desarrollado en el lenguaje de programación Visual Basic 6.0 este sistema fue proporcionado por cliente al igual que un análisis sobre el funcionamiento de cada módulo esto debido a que al ser un proyecto de migración el sistema desarrollado en Java debe contar con la misma funcionalidad que el anterior.

Hasta hoy en día el proyecto general ha alcanzado diversos objetivos principales como lo es la creación de la pantalla principal del sistema, el establecimiento de la arquitectura del mismo, la elaboración del análisis de diversos módulos y el desarrollo de múltiples componentes tanto genéricos como específicos. Una de las principales peticiones del cliente fue la de respetar la funcionalidad del sistema anterior, por lo tanto el sistema actual no contara con nueva funcionalidad o distinta a la establecida, ya que sólo se llevó acabo el cambio de arquitectura y la mejora de procesos el cual vuelve al sistema seguro y escalable, así como también al contar con una tecnología más actual a la hora de realizar estos el sistema es más eficaz, de igual manera al tener una mejor infraestructura y disminución en cuanto a la líneas de código las operaciones se realizan de manera más fácil y rápida, disminuyendo la perdida de datos por transacción, el sistema podrá ser utilizado en cualquier equipo con navegador sin importar el sistema operativo ya que este es web.

1.1 Estado del Arte

Se define como proyecto de migración al proceso por el cual un entorno informático basado en software se adapta y traslada a algún otro software de uso libre, asegurando una adecuada transacción entre las mismas y garantizar que la funcionabilidad de este sea igual que el sistema anterior unos ejemplos de estos proyectos son:

Proyecto	Características	Año	Metodología	
Migración sistema de Wal-Mart[1]	<ul style="list-style-type: none">Migración del sistema Wal-Mart	<ul style="list-style-type: none">2002	<ul style="list-style-type: none">UML	<ul style="list-style-type: none">Se terminó

	<p>del lenguaje PHP a Java 1.5.</p> <ul style="list-style-type: none"> • Mejora de herramientas de monitoreo. 			<p>el proceso de migración en el año 2002.</p>
Migración de PRENDAMEX[2]	<ul style="list-style-type: none"> • Migración del lenguaje .NET 1.1 a NET 3.5 	<ul style="list-style-type: none"> • 2011 	<ul style="list-style-type: none"> • UML 	<ul style="list-style-type: none"> • El proyecto termino en el año 2012.
Migración de sistema de información para el ministerio de justicia.[3]	<ul style="list-style-type: none"> • Migración de PHP a Java. • Configuración del ambiente de desarrollo JBOSS 5.1 GA 	<ul style="list-style-type: none"> • 2002 	<ul style="list-style-type: none"> • SCRUM 	<ul style="list-style-type: none"> • 2002
Proyecto NACA migración de COBOL a Java [4]	<ul style="list-style-type: none"> • Migración del sistema COBOL que corría en una aplicación de IBM/Mainframe a Java. • Se migraron 4 millones de líneas de código fuente COBOL. 	<ul style="list-style-type: none"> • 2009 	<ul style="list-style-type: none"> • SCRUM 	<ul style="list-style-type: none"> • Termino en el 2009
Migración de banca electrónica.(Solser)	<ul style="list-style-type: none"> • Migración de Java antiguo 1.3 a Java 1.6 	<ul style="list-style-type: none"> • 2017 	<ul style="list-style-type: none"> • SCRUM 	<ul style="list-style-type: none"> • En progreso.

	<ul style="list-style-type: none"> • Implementación de arquitectura MVC. 			
--	---	--	--	--

El lenguaje Visual Basic el cual fue lanzado en el año 1991 fue muy bien recibido debido a que con este se podría realizar una aplicación para Windows con mayor facilidad y rapidez, debido a que su interfaz gráfica e intuitiva era fácil de implementar los controles sobre un formulario solamente arrastrando estos, es por esto que actualmente muchas aplicaciones creadas para distintas empresas se encuentran desarrolladas bajo este lenguaje.

En [5] se describe como tras el pasar de los años el lenguaje de programación VISUAL BASIC ha pasado a ser obsoleto a comparación de distintos lenguajes de programación como C++ o Cobol los cuales se han mantenido a la altura y siguen siendo opciones viables para diversas empresas bancarias.

Actualmente han surgido nuevas tecnologías que traen ventajas de diversos lenguajes y ninguna desventaja, algunos de estos pueden ser JAVA, C# etc. JAVA actualmente el más conocido este por ser multiplataforma, así como multifacético, este lenguaje puede ser la opción perfecta para realizar la migración por cuestiones como despliegue, documentación, arquitectura y los Frameworks hacen de este una de las opciones más robustas para desarrollar.

1.2 Planteamiento del Problema

La empresa bancaria cuenta con un sistema para el área de ventanilla que se encarga de llevar acabo todas las transacciones de la empresa, por ejemplo, movimientos de cuentas, depósitos, entre otros. Sin embargo, este sistema se encuentra desarrollado en una tecnología obsoleta como lo es Visual Basic 6.0, actualmente dicha tecnología no cuenta con soporte esto da como resultado una complicada mantenibilidad del sistema y una gran pérdida de datos por transacción, aunado a su inadaptabilidad en equipos modernos.

1.3 Objetivo general

Realizar la migración de los módulos “Cifras totales en sucursales”, “Reimpresión de ticket”, “Cambio de efectivo”, “Consultas de saldo tarjeta de crédito y débito” del lenguaje de programación

Visual Basic al lenguaje de programación orientado a web como lo es Java, ofreciendo así la posibilidad de ser un producto capaz de ser adaptable en diferentes dispositivos.

1.4 Objetivos

- Dividir en capas la funcionalidad del sistema para cada módulo.
- Disminuir la pérdida de datos por transacción.
- Optimización de procesos para registro de datos, implementando Beans de sesión para no repetir consultas como nombre de operador, sucursal y caja.
- Simplificación de validaciones para cada módulo.

1.5 Definición de variables

La empresa bancaria cuenta con un sistema desarrollado actualmente en el lenguaje Visual Basic que carece de arquitectura alguna, dicho lenguaje ya no cuenta con soporte desde hace más de 10 años por lo cual para realizar un mantenimiento al sistema es muy complicado sin mencionar que no es posible utilizarlos en equipos modernos.

Variables directas

- Implementación de una nueva arquitectura aunado a la implementación de una nueva tecnología.
- Disminución en pérdida de datos por transacción.

Variables indirectas.

- Disminución de tiempo en la ejecución de cada transacción.

1.6 Hipótesis

Al llevar a cabo la implementación de una nueva arquitectura, aunado a la implementación de una nueva tecnología, se espera que la pérdida de datos por transacción se reduzca considerablemente por el control de transacciones descentralizado.

1.7 Justificación del Proyecto

La migración del sistema bancario desarrollado en Visual Basic a la plataforma JAVA orientado web traerá consigo múltiples beneficios como agilidad en los procesos, mejoras de seguridad, mayor

rendimiento, escalabilidad y mantenibilidad, ya que la migración del proyecto permitirá comprender el análisis y elaborar una documentación apropiada para futuros casos, de igual manera al contar con una arquitectura definida la comprensión para futuros desarrolladores que interactúen y se relacionen con el sistema se hará más fácil.

1.8 Limitaciones y Alcances

Limitaciones:

- Diseño gráfico establecido sin opciones de mejora.
- El uso de este sistema será exclusivo para la empresa para la cual se desarrolla.

Alcances:

- Permitir que el sistema sea utilizado en cualquier equipo y sistema operativo.
- Mejorar los procesos dentro del sistema.
- Disminuir la pérdida de datos por transacción.
- Facilitar la mantenibilidad del sistema migrado a Java.

1.9 Solser Information Technology

Historia De La Empresa

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 horas y con capacidad para 180,000 y contamos con oficinas comerciales en la Ciudad de México y Cancún Quintana Roo.

Misión, Visión y Valores

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.

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.

Valores:

- 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 los 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.

Procesos

Al contar con la certificación en el modelo CMMI nivel 3 todos los procesos de la empresa se encuentran en una constante mejora, la cual requiere el estricto seguimiento de cada una de las actividades realizadas, existen diversos procesos dentro de la empresa cada uno acorde a las necesidades de cada área, pero en este caso nos enfocaremos en el departamento de desarrollo ya que es considerado el que más flujo de trabajo genera.

Durante la concepción del proyecto se encuentran 3 procesos principales compuestos por diversas actividades estos procesos son.

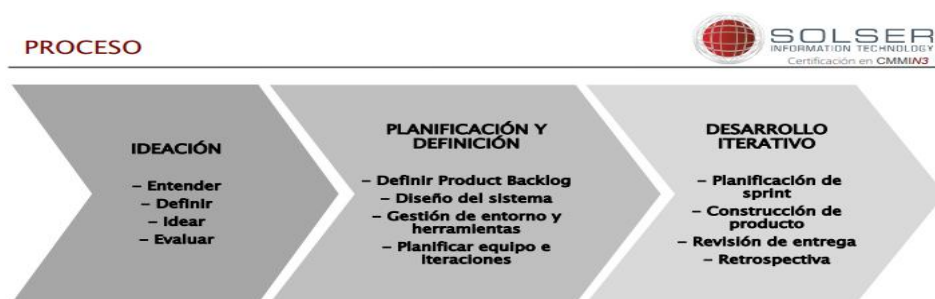


Ilustración 1-1 Diagrama CMMI 3 Solser

Una vez que se tiene el proyecto definido y se comienza con la construcción es aquí cuando el equipo sigue la metodología SCRUM con sus diversos procesos a realizar.

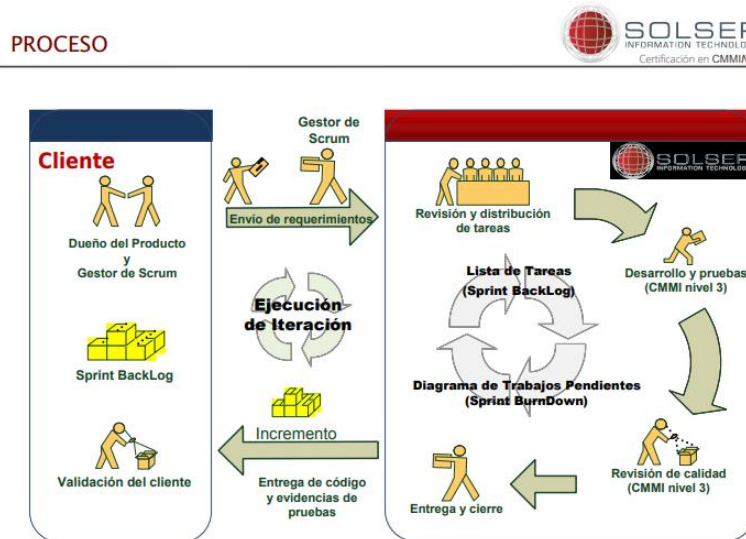


Ilustración 1-2 Diagrama SCRUM Solser

Mercado De Impacto

Para la industria del software existen diversos mercados que pueden ser explotados, como el de entretenimiento, transporte, turismo, telecomunicaciones, Pymes y el sector financiero siendo este último el sector de mayor aprovechamiento para la empresa Solser, ya que cuenta con más de diez productos orientados a este ramo los cuales suelen ser los más lucrativos y de mayor reto.

Impacto En El Área De Las TIC

Dentro de la empresa bancaria el sistema de ventanilla es uno de los que más carga de trabajo tiene día con día, por lo cual es de vital importancia que este tenga un correcto funcionamiento y un nivel de eficiencia óptimo, es por eso que con la migración tecnológica serán ofrecidos múltiples beneficios para poder cumplir con las condiciones mencionadas con anterioridad, otro añadido importante que viene con el cambio de lenguaje es el soporte del mismo, esto quiere decir que con el paso del tiempo el lenguaje contará con diversas APIS que ayudarán al sistema a realizar diversas acciones de maneras más simples y eficientes.

CAPÍTULO 2. METODOLOGÍA

La metodología bajo la cual se desarrollan los distintos proyectos de la empresa SOLSER es SCRUM el cual es un proceso completo que aplica de manera constante buenas prácticas para el trabajo en equipo, todo esto para la obtención del mejor resultado posible del proyecto. Estas prácticas son apoyadas unas por otras y busca volver a los equipos altamente productivos.

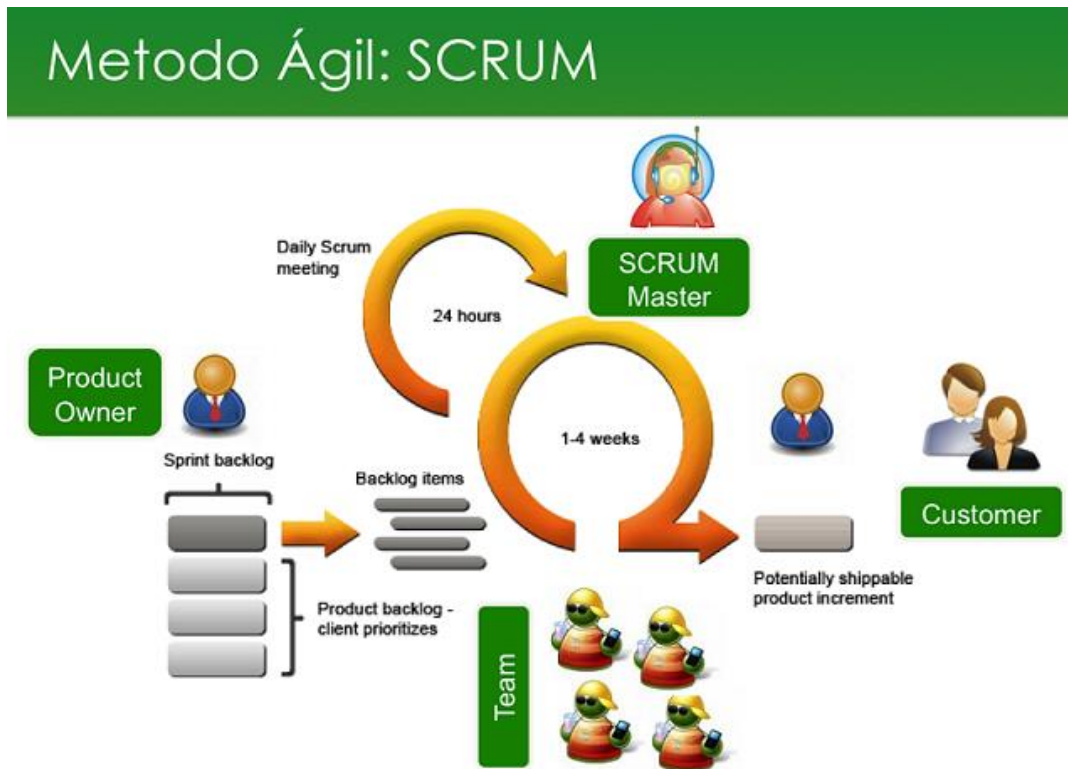


Ilustración 2-1 Metodología SCRUM [6]

2.1 Planificación

En esta fase se reúnen todos los involucrados en el desarrollo y se decide cuáles serán los requerimientos o tareas que se le asignarán a cada miembro del equipo, así como la especificación del tiempo que sea prudente para realizar el trabajo, con base a esta determinación se establece la duración del sprint.

2.2 SCRUM Diario

Se llevan a cabo reuniones con el equipo de desarrollo de un máximo de 15 minutos, para la sincronización de actividades y creación de un plan para las próximas 24 horas, esta fase se realiza con la finalidad de evaluar el progreso del sprint y evaluar la tendencia del mismo.

El Sprint Backlog es una lista de tareas que contiene datos como: nombre de la tarea, quien la desarrolla y estado de esta, (por ejemplo, progreso, revisión y completada con las horas por cada tarea.).

2.3 Trabajo de desarrollo

Cuando el sprint está en curso se debe asegurar que no se realizan cambios que afecten el objetivo del sprint, en esta fase el cliente revisa cada uno de los elementos dentro del product backlog el cual son un listado de todas las tareas que se pretenden hacer durante el desarrollo del proyecto esto con la finalidad de esclarecer cualquier duda que pueda surgir por parte de los desarrolladores también funciona para volver a estimar el tiempo y esfuerzo dedicado a cada requerimiento.

2.4 Revisión del sprint

Los miembros del equipo y el cliente se reúnen para mostrar el trabajo desarrollado de software que se ha completado, el cliente identifica lo que se ha hecho y lo que no se ha hecho, se lleva a cabo un análisis del estado actual del product backlog y estima fechas de finalización basado en el progreso hasta esa fecha.

2.5 Retrospectiva del sprint

En esta fase se realiza una reunión para dialogar sobre lo ocurrido durante el sprint por ejemplo sobre ¿Qué se hizo bien?, ¿Qué se hizo mal?, ¿Qué inconvenientes se encontraron? Esto con la finalidad de crear un plan para la implementación de mejoras con respecto a cómo el equipo hace su trabajo.

CAPÍTULO 3. DESARROLLO DEL PROYECTO

Debido a que la empresa trabajo bajo la metodología SCRUM el desarrollo del sistema se llevó a cabo basado en esta, es decir que para la realización de cada módulo se hizo como primer paso un análisis del sistema, en base a esto se estimó el tiempo para la creación de cada módulo, y posteriormente se sometió a una revisión por parte del departamento de calidad de la empresa SOLSER esto con el motivo de que el entregable llegara al cliente con el menor número de detalles posibles.

Estimación del tiempo: Durante la etapa de estimación se recaban todas las funciones que realiza el módulo asignándole un tiempo de realización en la herramienta MeisterTask así como un formato de Excel en el que el líder del proyecto aprueba esta estimación.

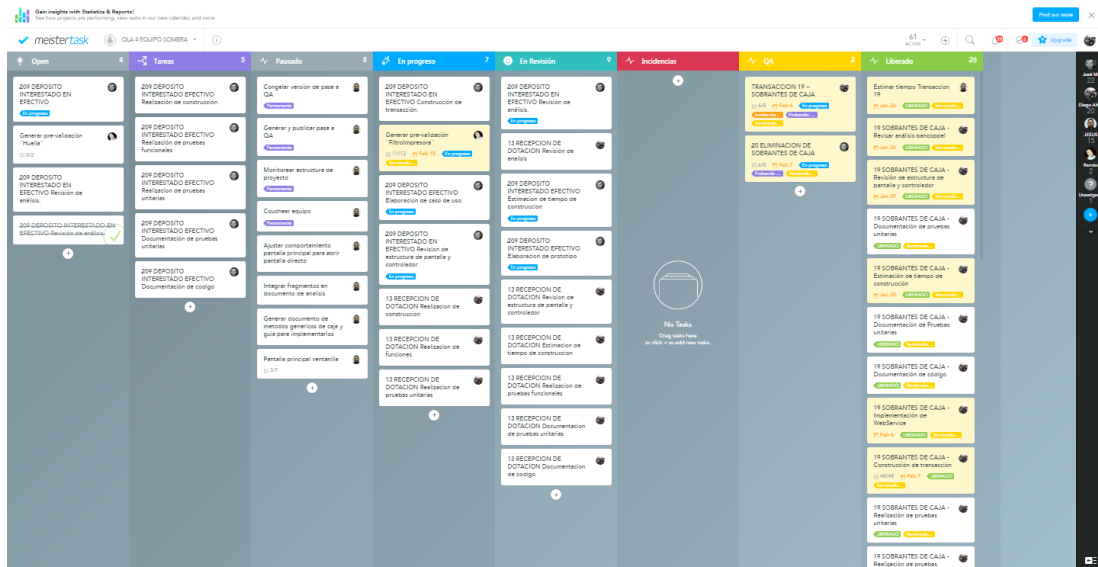


Ilustración 3-1 Herramienta MeisterTask

En esta se crean los diferentes trabajos que se realizaran en el desarrollo del módulo desde la creación de la documentación hasta la entrega al área de QA.

Tarea	Horas (Si aplica)	Justificación
Validación de datos de pantalla	3.00	Son 2 controles que necesitan validacion de longitud y valor así como tambien que no vallan vacios y que funcionen de manera correcta es decir que sean navegables .
Validación de reglas de negocio	5.00	Debido a que se deben realizar consultas para validar desde el Folio de dotacion ingresado pertenezca a una dotacion realizada en la transaccion 12.
Consultas a base de datos DB2 (ltran, itrandet, vdesgmon, varqueo, etc)	8.00	Se realizan consultas a las funciones fnvalidadotacion, fndota caja, fnvalidadenominacionesarqueo, fnValidaDotacionesReembolsos, fnvalidadatosguardados, fnvalidamovimientosdotacion, a la tabla vdesgmon, vdiarios, vdotacaja, .
Funciones DB2 a ejecutar	3.00	Se realizan 2 funciones, para validar el usuario y verificar que la transaccion se realizo de manera correcta.
Afectación a tablas (incluir tablas varqueo, vdiarios, vjourdjar, vdesgmon y las necesarias)	3.00	Se realizan registros de datos en las tablas vdiarios, vdesgmon, vdotacaja
Integración de Web Services	0.00	
Confirmación de fin de transacción	2.00	Se deben validar que la transacción se haya realizado de manera correcta.
Elementos (controladores, servicios, clases helpers, métodos, etc.)	4.00	Esto debido a la realización de las distintas clases por realizar.
Documentación	3.00	Debido a que son varias funciones controladores y clases que se deben realizar.
Pruebas unitarias	4.00	Se debe verificar que todas las funciones se realisen de manera correcta.
Total de horas	35.00	
DEPENDENCIAS		

Ilustración 3-2 Formato para estimación de tiempo en Excel

Documentación del módulo: Esto se hace basándose en el análisis y la máquina virtual proporcionada por el cliente, para la realización de esta la empresa SOLSER te proporciona una plantilla de cómo se debe realizar, esta se llena con una imagen de casos de uso, un diagrama de flujo, la descripción de las funciones que realizan en orden cronológico, las tablas en las que lleva un registro y las columnas en las que lo hace, así como también una imagen del prototipo de la transacción. Para la realización de esta documentación es necesario debuggear la aplicación en VISUAL BASIC y compararlo con lo descrito en los documentos del cliente, para llevar acabo los diagramas es necesario utilizar la aplicación STAR UML y enviárselos en el formato de la misma al líder del proyecto para que este tenga un respaldo de cada módulo y su respectivo diagrama, posteriormente se realiza el prototipo de la pantalla con la herramienta SCENE BUILDER la cual debe contar con las medidas establecidas por la empresa y los controles desarrollados por la misma.

Ejemplo de los diagramas e interfaces de cada transacción.

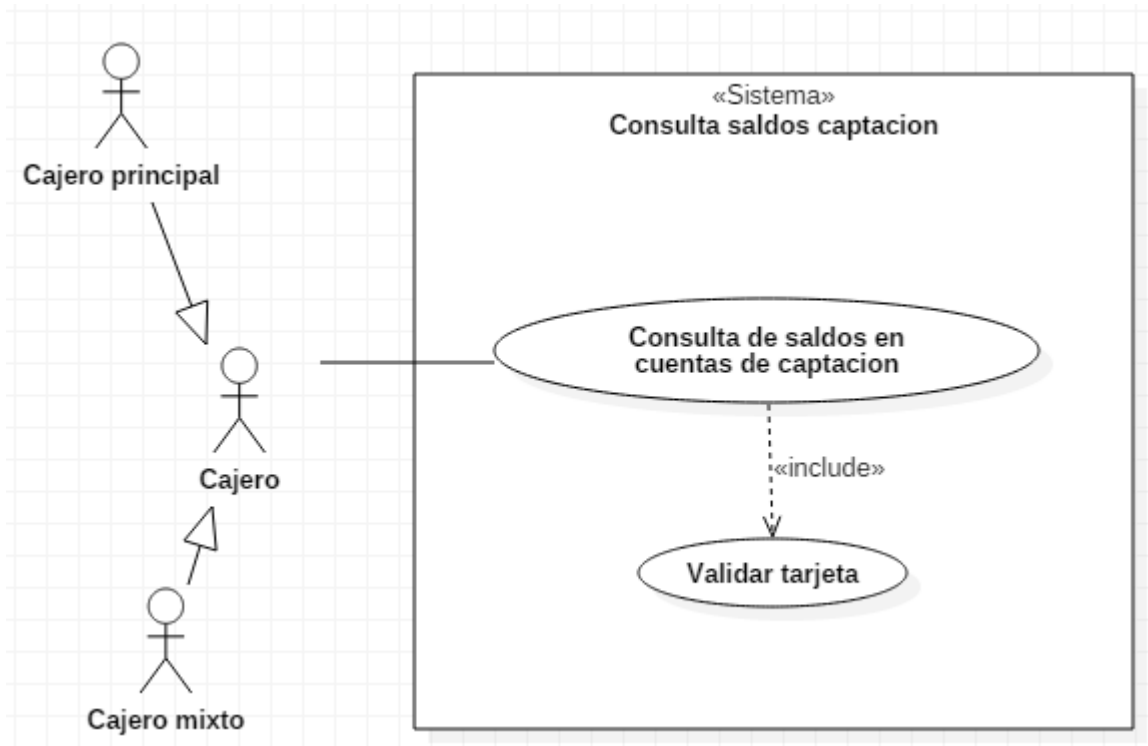


Ilustración 3-3 Caso de uso consulta de saldos en cuenta de captación.

Caso de uso consulta saldos en cuentas de captación, módulo desarrollado durante mi estadía en la empresa, la función necesita obligatoriamente de la función desliza tarjeta para mostrar a los clientes su saldo en la cuenta.

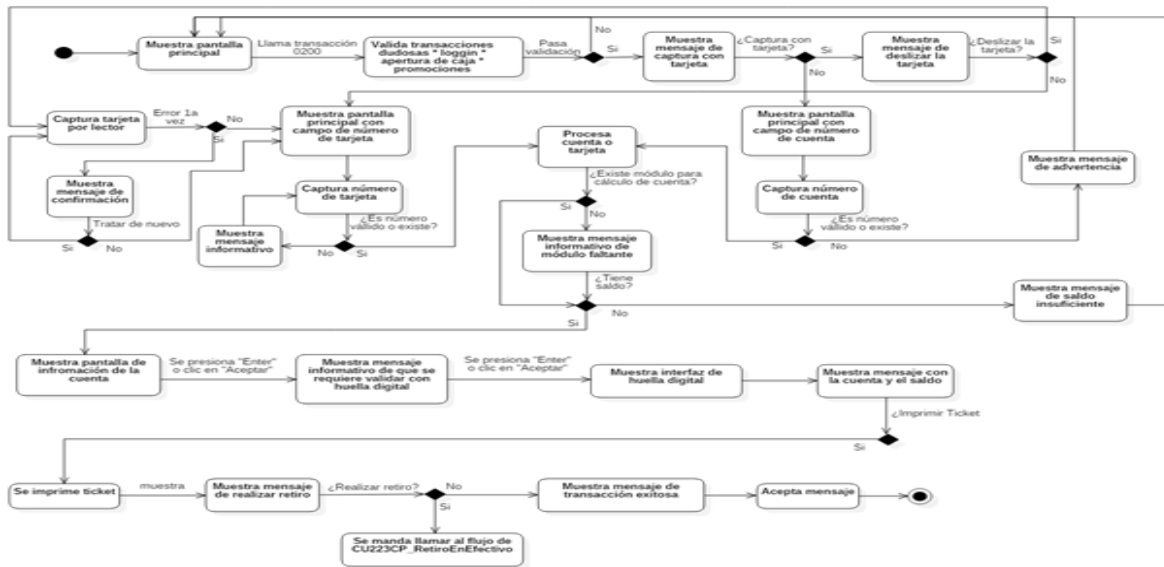


Ilustración 3-4 Diagrama de flujo Consulta saldos de cuenta.

Diagrama de flujo de la misma transacción este fue recortado por motivos de confidencialidad.

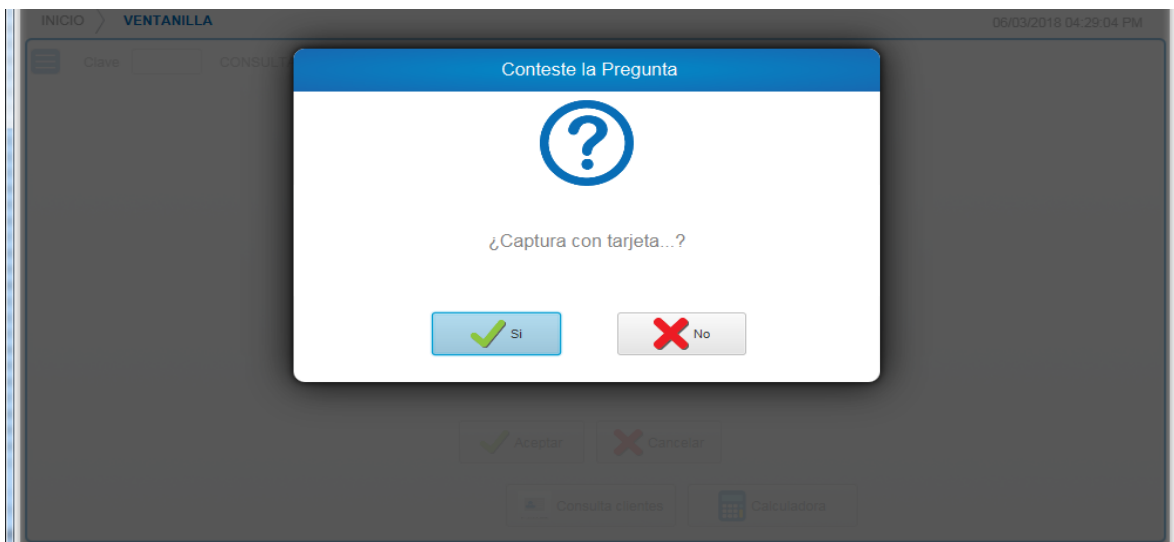


Ilustración 3-5 Interfaz del módulo consulta de saldos en cuenta de captación.

Solicita si se va a deslizar la tarjeta para la consulta

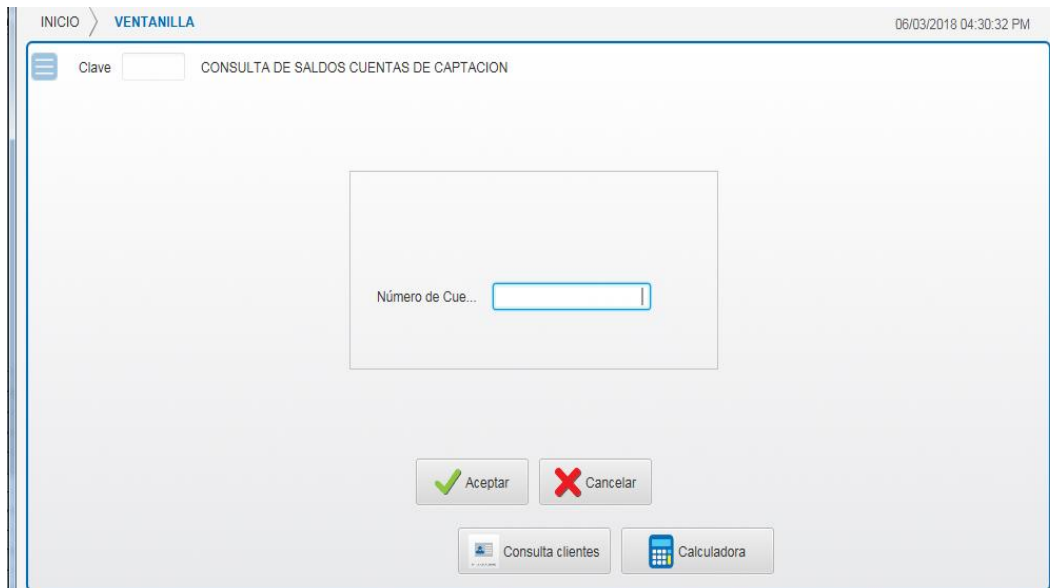


Ilustración 3-6 Interfaz del módulo consulta de saldos en cuenta de captación.

Interfaz del módulo consulta de cuentas de captación si selecciona no en la modal anterior en este se ingresa el número de cuenta.

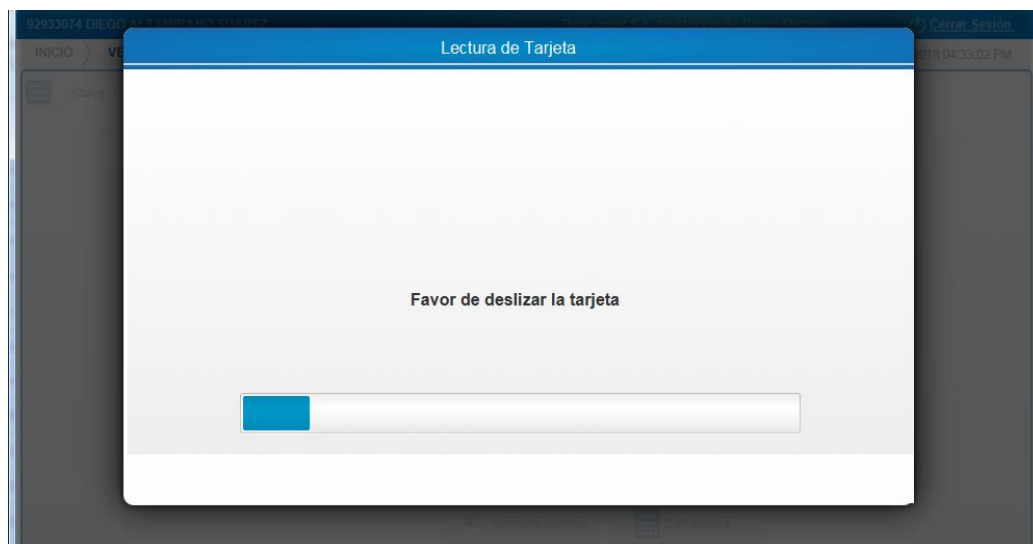


Ilustración 3-7 Interfaz del módulo consulta de saldos en cuenta de captación.

Si se selecciona la opción si en la modal empezara a utilizar el lector para validar la tarjeta.

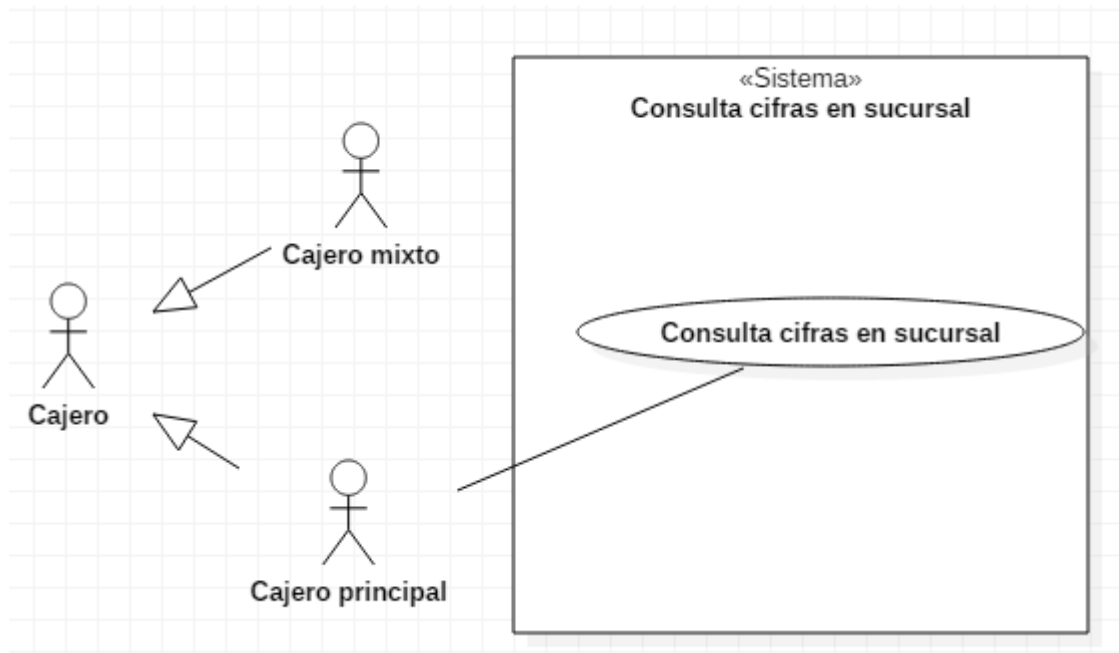


Ilustración 3-8 Caso de uso del módulo Consulta cifras sucursal.

En este módulo solo se puede visualizar cuanto efectivo ingreso a la caja.

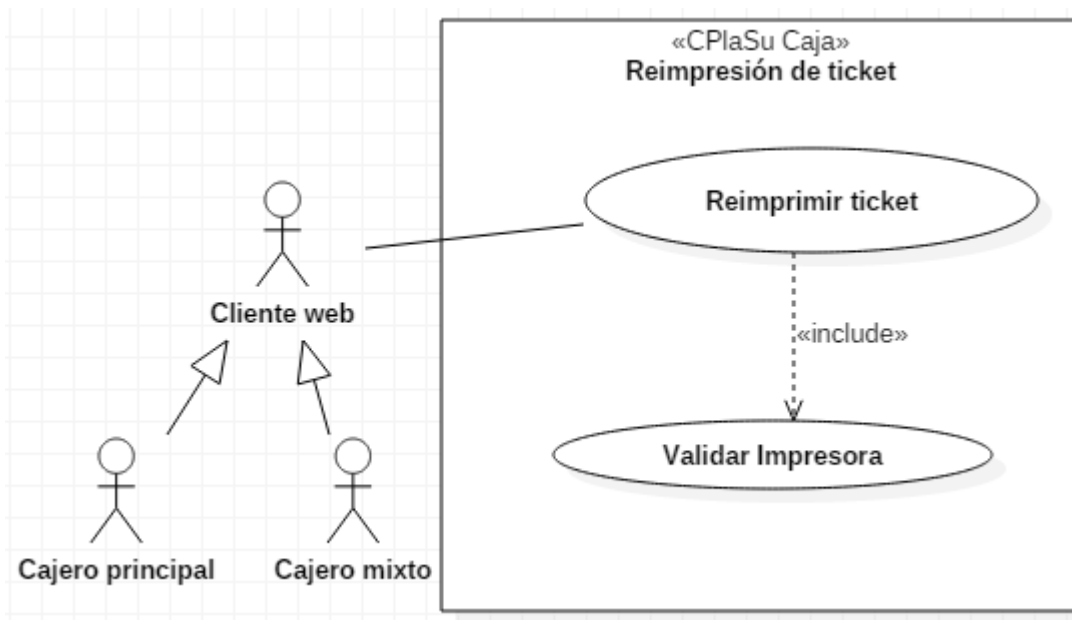


Ilustración 3-9 Caso de uso del módulo Reimpresión de ticket.

Este módulo sirve para recuperar el ticket en caso de que el cliente lo haya perdido.

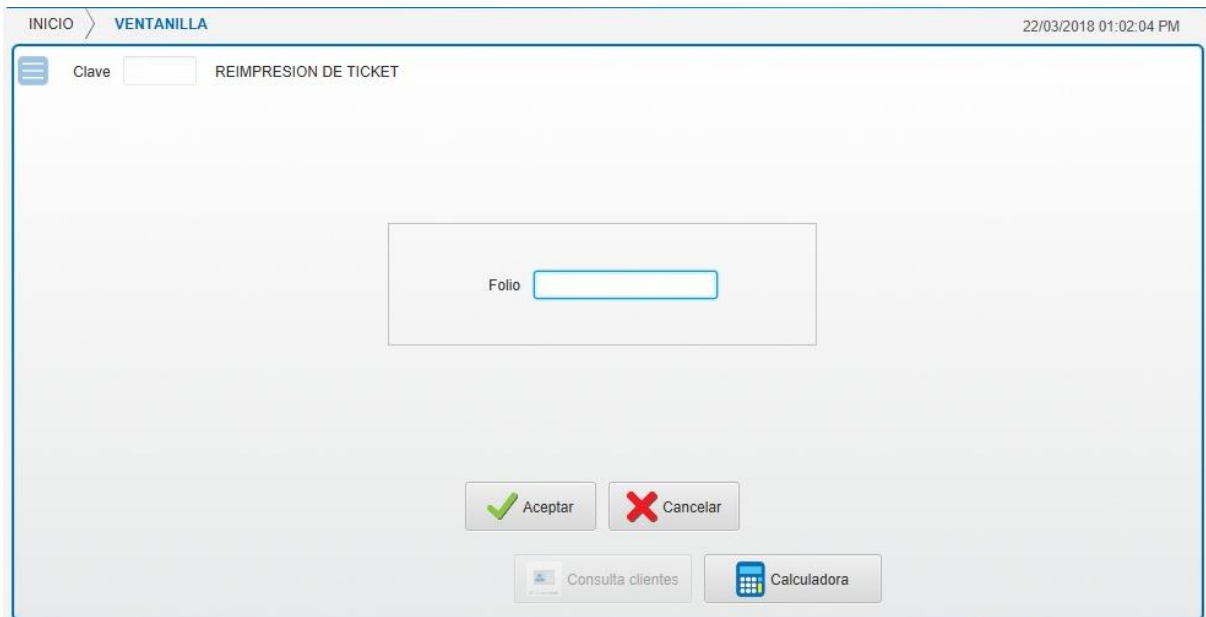


Ilustración 3-10 Pantalla del módulo Reimpresión de ticket.

Funciona ingresando el folio de la transacción.

Revisión del análisis del módulo: Una vez realizada la documentación del módulo este es guardado en un repositorio de la empresa mediante la utilización de la herramienta SMART CVS en la carpeta destinada al proyecto, con un nombre descriptivo para el mismo por ejemplo FDS_006CasosUso_Ventanilla_”CU”NUMEROTRANSACCION”CP”_NOMBRETRANSACCION donde se entiende como FDS_006 al formato proporcionado por la empresa, CU “CASO DE USO” y CP lo cual hace alusión a “CAJERO PRINCIPAL”, al llevar acabo el guardado del documento, el área de QA Aseguramiento de calidad o por sus siglas en inglés QUALITY ASSURANCE lleva a cabo una revisión de lo descrito en el módulo y lo realizado en la aplicación de VISUAL BASIC este departamento, lleva acabo comentarios para la compostura del mismo en caso de que no esté documentado o lo libera para su construcción en java.

Como se muestra en la ilustración 3-8 la documentación del módulo es guardada en un repositorio dependiendo el proyecto en este caso el OLA 4

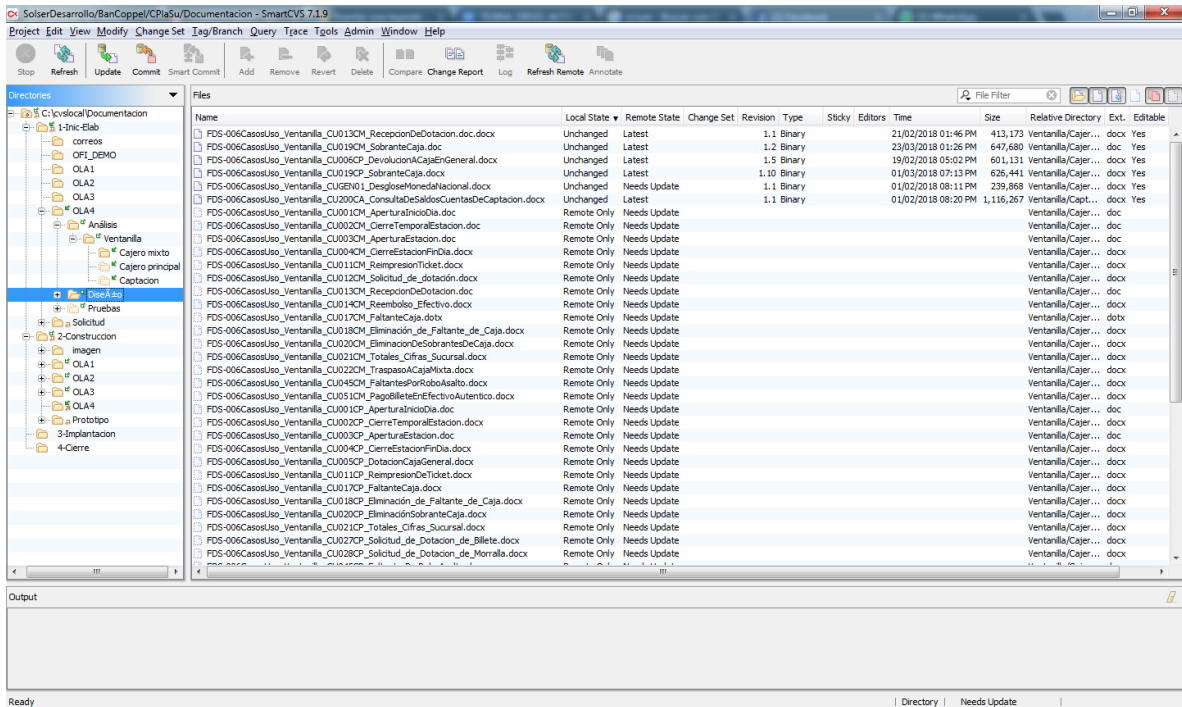


Ilustración 3-11 Interfaz del repositorio SmartCVS

Construcción del módulo: Cuando el área de QA te brinda el visto bueno del caso de uso, se procede con la construcción de la transacción, para ello se utiliza la aplicación eclipse el cual es un IDE para java developers, así como también la aplicación AQUA DATA STUDIO en el cual se encontrarán las tablas que se requieren el sistema y los STORED PROCEDURES que se ocupan para afectar a estas ya sea para realizar consulta, actualización y/o eliminación, si el módulo cuenta con web services(SERVICIOS WEB) se ocupa la aplicación SOAP UI para probar, simular y generar código de manera sencilla, en la construcción del módulo se realizan las siguientes clases:

- Controller del módulo. En esta clase se agrega toda la funcionalidad del módulo en cuanto al front end y la conexión con el back end.
- Cajero principal services. La clase sirve para realizar la conexión entre el frontend y el backend se debe agregar información del módulo como son los request (Datos que envía) y los response (La respuesta que recibe) así como validaciones de que hacer en caso de que no se lleve a cabo el guardado, actualización o eliminación de datos ya que la respuesta volverá al front end.

- URL constantes. Esta es al igual que la anterior una clase genérica donde solamente se agrega el nombre de la transacción en mayúsculas sin espacios en su lugar se ocupan guiones bajo (_).
- Bussiness. Esta es una interfaz donde se especifica al método que se dirige y los datos que va enviar el cual puede ser solamente el request de la transacción o de algunos otros.
- Bussiness Impl. En la clase de bussiness impl se lleva acabo toda la implementación de lógica del sistema en cuanto ha guardado en base de datos utilización de web services o stored procedures.
- Request y Response. Estos son los beans de los datos que se van a mandar en cuanto al request y los datos que se van a recibir esto para el response.

Revisión del módulo: Una vez terminado el módulo se lleva acabo documentación de este tanto en el código como en una plantilla proporcionada por la empresa en la cual se incluyen imágenes de cómo se lleva a cabo la transacción con sus diferentes flujos alternos, este documento es subido al repositorio antes mencionado con el nombre PR_NOMBREDETRANSACCION#_CP donde significa (Pruebas) y CP(Cajero principal) posteriormente el sistema es liberado para el área QA donde realizan pruebas de funcionamiento en cuanto al documento y lo que realiza el módulo, si se encuentran incidencias o bugs se suben con evidencia a la plataforma de la empresa COS donde los desarrolladores podemos visualizar cual fue el error y corregirlo subiendo prueba de la corrección y pasando a estado terminado esta incidencia.

ID	TÍTULO	FECHA CREACIÓN	FECHA ACTUALIZACIÓN	ESTATUS	PRIORIDAD	VER
117	P_FinDia_C20	21-03-2018 07:23 PM	03-04-2018 10:44 AM	RESUELTO	MEDIA	
116	P_FinDia_C18	21-03-2018 07:21 PM	04-04-2018 09:44 AM	CANCELADO	MEDIA	
115	P_FinDia_C17	21-03-2018 07:19 PM	03-04-2018 10:41 AM	RESUELTO	MEDIA	
114	P_FinDia_C11	21-03-2018 07:17 PM	03-04-2018 06:33 PM	RESUELTO	MEDIA	
113	P_TotalCifrasSuc_C21	21-03-2018 06:56 PM	22-03-2018 11:20 AM	RESUELTO	ALTA	
112	P_TotalCifrasSuc_C18	21-03-2018 06:52 PM	22-03-2018 01:37 PM	RESUELTO	ALTA	
111	P_TotalCifrasSuc_C18	21-03-2018 06:49 PM	22-03-2018 01:34 PM	RESUELTO	BAJA	
110	P_TotalCifrasSuc_C14	16-03-2018 07:04 PM	21-03-2018 01:04 PM	RESUELTO	MEDIA	
109	P_TotalCifrasSuc_C13	16-03-2018 07:02 PM	21-03-2018 01:21 PM	RESUELTO	MEDIA	
108	P_TotalCifrasSuc_C00	16-03-2018 06:58 PM	21-03-2018 04:03 PM	RESUELTO	BAJA	
107	P_TotalCifrasSuc_C02	16-03-2018 06:39 PM	21-03-2018 04:06 PM	RESUELTO	BAJA	
106	P_TotalCifrasSuc_C01	16-03-2018 06:36 PM	21-03-2018 04:25 PM	RESUELTO	BAJA	
105	P_ElimFaltanteCaja_C011	16-03-2018 05:09 PM	26-03-2018 10:37 AM	RESUELTO	MEDIA	
104	P_DistacionCajaCajal_C8	16-03-2018 04:07 PM	28-03-2018 03:48 PM	RFU(RFUT)	MF79A	

Ilustración 3-12 Interfaz del sistema COS

CAPÍTULO 4. RESULTADOS Y CONCLUSIONES

El banco anterior mente ya contaba con un sistema, para el funcionamiento de sus operaciones, pero al ser un sistema antiguo no contaba con las actuaciones correspondiente, por tal motivo se hizo una actualización del mismo, utilizando las nuevas tecnologías de desarrollo, haciendo una interfaz amigable y que las ejecuciones de los usuarios sean más eficientes reduciendo el tiempo de respuesta, El nuevo sistema es adaptable a diferentes navegadores, del mismo modo que funciona en diferentes sistemas operativos. Su funcionamiento es en red por qué varias sucursales están conectadas gracias al nuevo sistema.

4.1 Resultados

Se disminuyó la perdida en el volumen de datos de los módulos realizados esto basado en consultas realizadas a la base de datos después de realizar cada transacción en el sistema de Visual Basic.

Nombre del módulo.	Datos perdidos en sistema de Visual Basic.
Consultas de cuentas de captación.	<ul style="list-style-type: none">• Perdida del código del cliente.• Perdida del código del empleado.• Perdida del Código de gerente.
Consultas de cifras de sucursal	<ul style="list-style-type: none">• Perdida de numero de sucursal.• Perdida de usuarios que registraron las ventas.
Reimpresión de ticket.	<ul style="list-style-type: none">• Perdida de numero de sucursal.• Perdida del código de empleado.• Perdida del folio.

Como se muestra en la ilustración 4-1 al llevar acabo la transacción recepción de dotación en el sistema de Visual Basic se perdió el valor de la columna nro_cuenta a diferencia de la ilustración 4-2 donde al realizar la misma transacción se insertaron de manera correcta todos los datos.

empresa	cod_usuario	secuencia	nro_cuenta	fecha_valida
001	96669454	17154602	0	16/11/2017 12:00:00 AM
001	96669454	17193902	0	16/11/2017 12:00:00 AM
001	96669454	17393202	0	16/11/2017 12:00:00 AM
001	96669454	11344002	0	16/11/2017 12:00:00 AM
001	96669454	11453105	0	16/11/2017 12:00:00 AM
001	96669454	14060602	0	16/11/2017 12:00:00 AM

Ilustración 4-1 Tabla del sistema en Visual Basic

empresa	cod_usuario	secuencia	nro_cuenta	fecha_valida
001	96669454	18411504	12000002648	16/11/2017 12:00:00 AM
001	96669454	18432707	12000002648	16/11/2017 12:00:00 AM
001	96669454	18443610	12121212121	16/11/2017 12:00:00 AM

Ilustración 4-2 Tabla en sistema Java

Para disminuir la perdida de datos se logró gracias a la implementación de log los cuales informan cuales datos se están enviando y con esto realizar las validaciones pertinentes para que solo se realicen las transacciones una vez que todos los datos se inserten de manera correcta. Una muestra es la siguiente imagen en la cual se visualizan los campos de la tabla Vdesgmon(Rojo) en esta se hace una consulta y posterior a esto inserta los valores con la función afecta tablas(Azul) a la tabla desglose de moneda y por ultimo inserta los nuevos registros en la tabla Vdesgmon (Verde).

```

22:09:14,775 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:09:14.775 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.persistence.cplasucaja.jdbc.db2tables.impl.VdesgmonDaoImpl - SELECT
EMPRESA,SUCURSAL,TRAN_NRO,ESTADO,DUDOSO,COD_USUARIO,SECUENCIA,TIPO_CUENTA,NRO_CUENTA,TIPO_MON
EDA,IMPORTE_TRAN,TOTAL_EFECTIVO,CANTIDAD_1,CANTIDAD_2,CANTIDAD_3,CANTIDAD_4,CANTIDAD_5,CANTIDAD_
6,CANTIDAD_7,CANTIDAD_8,CANTIDAD_9,CANTIDAD_10,CANTIDAD_11,CANTIDAD_12,CANTIDAD_13,CANTIDAD_
14,CANTIDAD_15,CANTIDAD_EGRESO_1,CANTIDAD_EGRESO_2,CANTIDAD_EGRESO_3,CANTIDAD_EGRESO_
4,CANTIDAD_EGRESO_5,CANTIDAD_EGRESO_6,CANTIDAD_EGRESO_7,CANTIDAD_EGRESO_8,CANTIDAD_EGRESO_
9,CANTIDAD_EGRESO_10,CANTIDAD_EGRESO_11,CANTIDAD_EGRESO_12,CANTIDAD_EGRESO_13,CANTIDAD_EGRESO_
14,CANTIDAD_EGRESO_15,TOTAL_DENOMINAC_1,TOTAL_DENOMINAC_2,TOTAL_DENOMINAC_3,TOTAL_DENOMINAC_
4,TOTAL_DENOMINAC_5,TOTAL_DENOMINAC_6,TOTAL_DENOMINAC_7,TOTAL_DENOMINAC_8,TOTAL_DENOMINAC_
9,TOTAL_DENOMINAC_10,TOTAL_DENOMINAC_11,TOTAL_DENOMINAC_12,TOTAL_DENOMINAC_
13,TOTAL_DENOMINAC_14,TOTAL_DENOMINAC_15,MEDIO_TARJ FROM DB2INST1.VDESGMON WHERE COD_USUARIO
=:usuario AND EMPRESA = :empresa AND SECUENCIA = :secuencia AND SUCURSAL = :sucursal

22:09:14,810 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:09:14.810 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.rest.business.cplasucaja.AfectaTablasBusinessImpl - AFECTA TABLAS -- DESGLOSE -- GENERANDO
DESGLOSE DE MONEDA

22:09:14,811 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:09:14.811 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.rest.business.cplasucaja.AfectaTablasBusinessImpl - AFECTA TABLAS -- DESGLOSE -- INSERTANDO
DESGLOSE DE MONEDA

22:09:14,812 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:09:14.812 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.persistence.cplasucaja.jdbc.db2tables.impl.VdesgmonDaoImpl - Query:insert into VDESGMON
(EMPRESA,SUCURSAL,TRAN_NRO,ESTADO,DUDOSO,COD_USUARIO,SECUENCIA,TIPO_CUENTA,NRO_CUENTA,TIPO_MO
NEDA,IMPORTE_TRAN,TOTAL_EFECTIVO,CANTIDAD_1,CANTIDAD_2,CANTIDAD_3,CANTIDAD_4,CANTIDAD_
5,CANTIDAD_6,CANTIDAD_7,CANTIDAD_8,CANTIDAD_9,CANTIDAD_10,CANTIDAD_11,CANTIDAD_12,CANTIDAD_
13,CANTIDAD_14,CANTIDAD_15,CANTIDAD_EGRESO_1,CANTIDAD_EGRESO_2,CANTIDAD_EGRESO_
3,CANTIDAD_EGRESO_4,CANTIDAD_EGRESO_5,CANTIDAD_EGRESO_6,CANTIDAD_EGRESO_7,CANTIDAD_EGRESO_
8,CANTIDAD_EGRESO_9,CANTIDAD_EGRESO_10,CANTIDAD_EGRESO_11,CANTIDAD_EGRESO_12,CANTIDAD_EGRESO_
13,CANTIDAD_EGRESO_14,CANTIDAD_EGRESO_15,TOTAL_DENOMINAC_1,TOTAL_DENOMINAC_
2,TOTAL_DENOMINAC_3,TOTAL_DENOMINAC_4,TOTAL_DENOMINAC_5,TOTAL_DENOMINAC_6,TOTAL_DENOMINAC_
7,TOTAL_DENOMINAC_8,TOTAL_DENOMINAC_9,TOTAL_DENOMINAC_10,TOTAL_DENOMINAC_
11,TOTAL_DENOMINAC_12,TOTAL_DENOMINAC_13,TOTAL_DENOMINAC_14,TOTAL_DENOMINAC_15,MEDIO_TARJ)
values
(:empresa,:sucursal,:transaccion,:estado,:dudoso,:cajero,:secuencia,:tipoCuenta,:numeroCuenta,:tipomoneda,:importetr
ansaccion,:totalefectivo,:cantidad1,:cantidad2,:cantidad3,:cantidad4,:cantidad5,:cantidad6,:cantidad7,:cantidad8,:cantid
ad9,:cantidad10,:cantidad11,:cantidad12,:cantidad13,:cantidad14,:cantidad15,:egreso1,:egreso2,:egreso3,:egreso4,:egr
eso5,:egreso6,:egreso7,:egreso8,:egreso9,:egreso10,:egreso11,:egreso12,:egreso13,:egreso14,:egreso15,:totaldenom1,:
totaldenom2,:totaldenom3,:totaldenom4,:totaldenom5,:totaldenom6,:totaldenom7,:totaldenom8,:totaldenom9,:totald
enom10,:totaldenom11,:totaldenom12,:totaldenom13,:totaldenom14,:totaldenom15,:mediotarjeta)

22:09:14,906 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:09:14.906 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.rest.business.cplasucaja.AfectaTablasBusinessImpl - AFECTA TABLAS -- ARQUEO -- GENERANDO ARQUEO

```

Ilustración 4-3 Log del sistema en Java.

En esta imagen se presenta los campos que va a afectar y en que tabla de igual manera los valores que va a agregar.

Se disminuyó el tiempo de respuesta por cada transacción en el lenguaje de Java a comparación de las desarrolladas en Visual Basic.

Nombre de modulo	Visual Basic	Java
Consulta de saldos en cuenta de captación.	1 Minuto.	0.5 Minutos.
Consulta de cifras sucursal.	1.5 Minutos.	0.6 Minutos
Reimpresión de ticket.	2 Minutos.	1.2 Minutos.

Se facilitó la mantenibilidad de los módulos realizados.

El sistema al ser web puede ser utilizado en cualquier sistema operativo que cuente con navegador.

Las interfaces, aunque mantienen el mismo funcionamiento que el antiguo sistema cuentan con mejoras tanto en el diseño como en la funcionabilidad.

4.2 Trabajos Futuros

Se llevará a cabo la realización del sistema de cajero mixto para la misma empresa ya que el explicado en este documento es el de cajero principal de igual manera se llevarán a cabo módulos de captación que se implementaran tanto en el sistema de cajero principal como el de mixto.

4.3 Recomendaciones

Debido a que la empresa posee una gran cantidad de proyectos futuros y la mayoría de estos son de migración de sistemas es necesario analizar el lenguaje más apropiado por lo cual se debe tomar en cuenta el tiempo de vida del mismo y el soporte que se le puede brindar a este mediante actualizaciones esto con el fin de que se pueda llevar acabo a largo plazo, de igual manera definir una arquitectura de software para facilitar la actualización y mantenibilidad del sistema , de igualmente se debe tener en consideración el servidor de la base de datos con el objetivo de que no presente problemas funcionales como perdida de conexión , caída de servidor o soporte muy escaso

Glosario

- Business Implement: Implementación del negocio.
- Business: Negocio
- Controller: Controlador
- Developers: Desarrolladores
- Frontend: Interfaz
- Request: Solicitud
- Response: Respuesta
- Services: Servicios
- Stored Procedures: Procedimientos almacenados.
- Web services: Servicios web

ANEXOS

En este apartado se muestran los Log más detallados tanto del backend como del frontend implementados en los módulos realizados durante mi estancia en la empresa SOLSER.

- Log de Backend y Frontend.

```
22:09:04.768 [Thread-29] DEBUG org.jboss.resteasy.resteasy_jaxrs.i18n - Interceptor Context:
org.jboss.resteasy.core.interception.jaxrs.ClientReaderInterceptorContext, Method : proceed
22:09:04.768 [Thread-29] DEBUG org.jboss.resteasy.resteasy_jaxrs.i18n - MessageBodyReader:
org.jboss.resteasy.spi.ResteasyProviderFactory$SortedKey
22:09:04.768 [Thread-29] DEBUG org.jboss.resteasy.resteasy_jaxrs.i18n - MessageBodyReader:
org.jboss.resteasy.plugins.providers.StringTextStar
22:09:04.779 [JavaFX Application Thread] INFO mx.com.solser.fx.service.general.BusyService - Quita: 0
Numero cajero principal: 92933066
22:09:09.736 [JavaFX Application Thread] INFO mx.com.solser.fx.util.rest.RestCall - URL-
REST::http://localhost:8081/CPlaSuCajaService-1.0.0/obtieneArqueoCajero.go
22:09:09.740 [Thread-35] DEBUG org.jboss.resteasy.resteasy_jaxrs.i18n - Interceptor Context:
org.jboss.resteasy.core.interception.jaxrs.ClientWriterInterceptorContext, Method : proceed
22:09:09.740 [Thread-35] DEBUG org.jboss.resteasy.resteasy_jaxrs.i18n - MessageBodyWriter:
org.jboss.resteasy.spi.ResteasyProviderFactory$SortedKey
22:09:09.740 [Thread-35] DEBUG org.jboss.resteasy.resteasy_jaxrs.i18n - MessageBodyWriter:
org.jboss.resteasy.plugins.providers.StringTextStar
22:09:09.857 [JavaFX Application Thread] INFO mx.com.solser.fx.service.general.BusyService - Agrega: 1
22:09:09.891 [Thread-35] INFO mx.com.solser.fx.util.rest.TaskRest -
Peticion::http://localhost:8081/CPlaSuCajaService-1.0.0/obtieneArqueoCajero.go::Exit: HTTP CODE
200 ::
22:09:09.892 [Thread-35] DEBUG org.jboss.resteasy.resteasy_jaxrs.i18n - Interceptor Context:
org.jboss.resteasy.core.interception.jaxrs.ClientReaderInterceptorContext, Method : proceed
22:09:09.892 [Thread-35] DEBUG org.jboss.resteasy.resteasy_jaxrs.i18n - MessageBodyReader:
org.jboss.resteasy.spi.ResteasyProviderFactory$SortedKey
22:09:09.892 [Thread-35] DEBUG org.jboss.resteasy.resteasy_jaxrs.i18n - MessageBodyReader:
org.jboss.resteasy.plugins.providers.StringTextStar
22:09:09.946 [JavaFX Application Thread] INFO mx.com.solser.fx.util.rest.RestCall - URL-
REST::http://localhost:8081/CPlaSuCajaService-1.0.0/obtieneParametroWeb.go
22:09:09.946 [JavaFX Application Thread] INFO mx.com.solser.fx.service.general.BusyService - Quita: 0
22:09:09.949 [JavaFX Application Thread] INFO mx.com.solser.fx.service.general.BusyService - Agrega: 1
22:09:09.950 [Thread-37] DEBUG org.jboss.resteasy.resteasy_jaxrs.i18n - Interceptor Context:
org.jboss.resteasy.core.interception.jaxrs.ClientWriterInterceptorContext, Method : proceed
22:09:09.950 [Thread-37] DEBUG org.jboss.resteasy.resteasy_jaxrs.i18n - MessageBodyWriter:
org.jboss.resteasy.spi.ResteasyProviderFactory$SortedKey
22:09:09.950 [Thread-37] DEBUG org.jboss.resteasy.resteasy_jaxrs.i18n - MessageBodyWriter:
org.jboss.resteasy.plugins.providers.StringTextStar
22:09:10.033 [Thread-37] INFO mx.com.solser.fx.util.rest.TaskRest -
Peticion::http://localhost:8081/CPlaSuCajaService-1.0.0/obtieneParametroWeb.go::Exit: HTTP CODE
```

Ilustración 5-1 Log de backend.

```

22:08:50,722 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:08:50.722 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.persistence.cplasu.db2.SecurityDaoImpl - SecurityDao :: validaSesionMac

22:08:50,762 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:08:50.762 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.persistence.cplasu.ws.WebServiceDao - WEBSERVICE - CONFIGURANDO PETICION

22:08:50,762 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:08:50.762 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.persistence.cplasu.ws.WebServiceDao - WEBSERVICE - REALIZANDO PETICION

22:08:50,763 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:08:50.763 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.persistence.cplasu.ws.ClienteSoapService - request: <soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"><soapenv:Header /><soapenv:Body>
<adm:AdmConsEjecutivo xmlns:adm="http://www.bancoppel.com/Login/AdmConsEjecutivo/"><cabecera>
<idTrxGlobal></idTrxGlobal><sistemaOrigen></sistemaOrigen></cabecera><eEjecut>92933066</eEjecut>
<eMac>408D5C9A12DF</eMac><eSuc>1000</eSuc></adm:AdmConsEjecutivo></soapenv:Body></soapenv:Envelope>

22:08:50,765 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:08:50.765 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.persistence.cplasu.ws.WebServiceDao - insertando bitacora de actividades: null - 0000000000

22:08:50,809 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:08:50.809 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.persistence.cplasu.ws.ClienteSoapService - response: <?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"><soapenv:Body>
<ns2:AdmConsEjecutivoResponse xmlns:ns2="http://www.bancoppel.com/Login/AdmConsEjecutivo/"><ejecutivo>
<codRetorno>00000</codRetorno></ejecutivo></ns2:AdmConsEjecutivoResponse></soapenv:Body>
</soapenv:Envelope>

22:08:50,809 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:08:50.809 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.persistence.cplasu.ws.WebServiceDao - actualizando bitacora de actividades: null - 0000000000

22:08:50,809 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:08:50.809 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.persistence.cplasu.ws.WebServiceDao - WEBSERVICE - VALIDANDO CODIGO ERROR

22:08:50,810 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:08:50.810 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.persistence.cplasu.ws.WebServiceDao - WEBSERVICE - VALIDANDO CODIGO DE RETORNO

22:08:50,810 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:08:50.810 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.persistence.cplasu.ws.WebServiceDao - WEBSERVICE - VALIDANDO CONTENIDO RESPONSE

22:08:50,810 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:08:50.810 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.persistence.cplasu.ws.WebServiceDao - WEBSERVICE - CONSTRUYENDO BEAN RESPUESTA

22:08:50,810 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:08:50.810 [http-localhost/127.0.0.1:8081-1] INFO
mx.com.solser.persistence.cplasu.ws.WebServiceDao - WEBSERVICE - AÑADIENDO CODIGO RETORNO A BEAN
RESPUESTA

22:08:50,810 INFO [stdout] (http-localhost/127.0.0.1:8081-1) 22:08:50.810 [http-localhost/127.0.0.1:8081-1] INFO

```

Ilustración 5-2 Log de Frontend.

BIBLIOGRAFÍA

- [1] Eugenio Ciurana. (2002). Migración a Java 5 en walmart.com. 2008, de SG Sitio web:
<https://sg.com.mx/content/view/93>

[2] Gustavo Rivera Cabrera. (2012). Migración de framework y desarrollo del módulo plata para el sistema de operaciones de préstamos prendarios (SOPREP) de la empresa Prendamex. Teziutlán Puebla: Teziutlán.

[3] All. (2015). Migración de sistema de información desde PHP a Java para el Ministerio de Justicia. 2015, de Desarrollolibre.cl Sitio web: <http://www.desarrollolibre.cl/drupal7/migración-de-sistema-de-información-desde-php-java-para-el-ministerio-de-justicia>

[4] Benek. (2009). Proyecto NACA: Migración de código COBOL a Java. 07/11/2009, de Java México Sitio web: http://www.javamexico.org/blogs/jb/proyecto_naca_migracion_de_codigo_cobol_java

[5] Jorge Saavedra. (2008). Historia de Visual Basic. 16/04/2008, de Jorge Saavedra Sitio web: <https://jorgesaavedra.wordpress.com/2008/04/16/historia-de-visual-basic/>

[6] N. Guerrero. (2016) ¿Conoces en que consiste la Metodología Ágil en Scrum? 28/06/2016, de Programa En Línea Sitio web: <http://programaenlinea.net/conoces-en-que-consiste-la-arquitectura-de-software-en-scrum/>