



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

Universidad Nacional Autónoma de México

Facultad de Ingeniería

División de Ingeniería Eléctrica

“Implementación de un evolutivo en Oracle
Siebel CRM para una institución bancaria
chilena.”

Informe experiencia profesional

Que para obtener el título de

INGENIERA EN COMPUTACIÓN

Presenta:

Celia Karen Huaracha Magaña

Director:

M.C. Alejandro Velázquez Mena

2015



Agradecimientos

A mis padres, las personas más importantes en mi vida, a quienes simplemente debo todo, por preocuparse y dar todo de sí mismos para mi formación personal y profesional. Por dar siempre prioridad a mi superación, por guiarme con su amor infinito y disciplina por el que ahora entiendo que es el mejor camino. Y sobre todo, por hacerlo con el ejemplo de las personas integra que son.

A mi hermano Luis, quien siempre ha sido ejemplo, guía y consejero certero. A quien admiro y que seguramente seguirá siendo referente profesional y compañero incondicional para toda la vida.

A Dann, por ser mi cómplice y mejor amigo. Por creer en mí, por ser apoyo y motivador de mis sueños, e incluso por compartir algunos. Además de ser un catalizador de muchas de las mejores cosas que hay en mí.

A mi familia, tan grande y única, quienes de distintas formas han estado presentes a lo largo de mi vida y han sido factores importantes en este camino y en ese “todo” que me define. A mis abuelitos Celia y Macedonio, Luis y Josefina, por todo el cuidado, el amor y la enseñanza inagotable. A Janet, por el oído que nunca se cansa y porque nos une más que la sangre. A Omar, por tanto aprendizaje compartido de toda índole. A Liz y a Brenda, por las risas y los años que seguiremos compartiendo. A Norma, Alan y todos con quienes crecí, porque no pudo ser mejor.

A mis amigos, con quienes he compartido muchos de los mejores años de mi vida y estoy segura seguiremos compartiendo.

A la Facultad de Ingeniería y a todos los profesores que forman parte de ella y que colaboraron en mi camino, a ustedes agradeceré siempre la preocupación por formar ingenieros de la más alta calidad al mismo tiempo que ciudadanos íntegros, conscientes y críticos al servicio de la sociedad.

A la Universidad Nacional Autónoma de México, por ser mi segunda casa, la institución que me dio herramientas para la vida mientras me formaba como profesional. Por ser un espacio de inclusión para el libre pensamiento, al que siempre será un gusto volver para aportar un poco de todo lo que me dio.

Contenido

Introducción.....	1
Capítulo 1.- La empresa.....	4
1.1. Organigrama	4
1.2. Descripción del puesto	4
Capítulo 2. Breve descripción de participación en otros proyectos.	6
Capítulo 3. Implementación de un evolutivo en Oracle Siebel CRM para una institución bancaria chilena.	10
3.1. Antecedentes generales	10
3.2. Objetivos y alcances generales	11
3.3. Metodología de implementación	14
3.4. Desarrollo del evolutivo	15
3.4.1. Marco teórico	15
3.4.2. Planteamiento del problema	20
3.4.3. Objetivos y alcances del evolutivo.....	20
3.4.4. Especificación de requerimientos funcionales y no funcionales.....	21
3.4.5. Diseño funcional	24
3.4.6. Diseño técnico	32
3.4.7. Construcción	38
3.4.8. Pruebas.....	56
3.4.9. Ambiente QA y pase a producción.....	61
3.5. Resultados.....	65
Conclusiones.....	66
Referencias	67

Tablas e ilustraciones

Figura 1. Organigrama de consultoría TI en everis	4
Figura 2. . Ciclo de vida de un evolutivo Siebel con interacción cliente y equipo desarrollador.....	13
Figura 3. Fases del proyecto y entregables CMMI	14
Figura 4. Arquitectura web de Siebel	15
Figura 5. Arquitectura de una aplicación de Siebel	17
Figura 6. Cómo se define un business component	18
Figura 7. Elementos en la interfaz de usuario de Siebel.	18
Figura 8. Interacción entre las tres capas de la arquitectura de una aplicación Siebel	19
Figura 9. Representación general del evolutivo	21
Figura 10. Situación actual de visibilidad CI.....	23
Figura 11. Situación futura de visibilidad CI	23
Figura 12. Pantalla productos/servicios con CI	24
Figura 13. Vista más información con CI	25
Figura 14. Vista detalle de productos/servicios con CI.....	25
Figura 15. Pantalla empresas con CI	25
Figura 16. Vista visión global con CI.....	26
Figura 17. Pantalla productos/servicios con RUT empresa muestra CI.....	26
Figura 18. Vista productos con RUT empresa muestra CI	27
Figura 19. Pantalla productos/servicios derivada del drill down de familia de productos.....	27
Figura 20. Vista resumen cliente.....	28
Figura 21. Pantalla personas con CI	28
Figura 22. Vista visión global de personas con CI en productos/servicio	29
Figura 23. Vista producto en personas con familia de CI	29
Figura 24. Vista resumen cliente para personas	29
Figura 25. Lógica de asignación de responsabilidades.....	30
Figura 26. Flujo del filtro	31
Figura 27. Orden en el diseño de la solución	32
Figura 28. Estación de trabajo	39
Figura 29. Camino del evolutivo de desarrollo a producción	41
Figura 30. Orden de construcción del evolutivo.	42
Figura 31. Campos para join faltante	44
Figura 32. Relación de business component con applets y toggle applets.....	46
Figura 33. Relacionar business component con business objects	47
Figura 34. Links para relacionar BO y BC.....	47
Figura 35. Campos calculados para filtro.....	48
Figura 36. Multi Value Link	49
Figura 37. Links con las listas de valores.....	49
Figura 38. Filtro en las especificaciones de búsqueda del BC	51
Figura 39. Relación M:M entre vistas, responsabilidades y usuarios	53

Figura 40. Asignación de responsabilidades a usuarios.....	54
Figura 41. Asignación de responsabilidades a las vistas.	54
Figura 42. Relación de las vistas a las pantallas.....	55
Figura 43. Agregar pantallas a la aplicación en Siebel Tools	56
Figura 44. Pantalla productos/servicios sin CI	59
Figura 45. Pantalla productos/servicios vista más información que sólo muestra registros con clase cuenta distintos de CIN Y CIV.	59
Figura 46. Pantalla personas, vista visión global, que no muestra CI en productos/servicios.	60
Figura 47. Pantalla personas, vista productos sin mostrar CI.....	60
Figura 48. Ejemplo de la evidencia en la certificación de un objeto.	63

Tabla 1. Descripción del campo RUT.....	22
Tabla 2. Cruce de posibilidades entre responsabilidad y RUT	22
Tabla 3. Análisis de elementos para productos/servicios	33
Tabla 4. Análisis de elementos para productos/servicios	33
Tabla 5. Análisis de elementos para productos/servicios	33
Tabla 6. Análisis de elementos para empresas.....	34
Tabla 7. Análisis de elementos para empresas.....	34
Tabla 8. Análisis de elementos para empresas.....	34
Tabla 9. Análisis de elementos para personas.....	35
Tabla 10. Análisis de elementos para personas.....	35
Tabla 11. Análisis de elementos para personas.....	35
Tabla 12. Análisis de los business component principales	37
Tabla 13. Análisis de los business component principales.....	37
Tabla 14. Análisis de los business component principales.....	37
Tabla 15. Ambientes por los que pasará el evolutivo	40
Tabla 16. Lista de valores implementada para el filtro.	43
Tabla 17. Evaluación del filtro.	52
Tabla 18. Casos de prueba.....	57
Tabla 19. Ejecución de una prueba	58
Tabla 20. Objetos .sif que conforman el evolutivo y se van a exportar	61

Introducción

Las empresas han pasado las últimas dos décadas reorganizando sus procesos internos para mejorar su rendimiento. El beneficio aportado por estas estrategias residía en la reducción de los costos de transacción, pero desde hace unos años se ha empezado a perseguir un nuevo objetivo: el incremento de los beneficios y las oportunidades de negocio. El cambio de paradigma es claro e implica un giro de la visión desde dentro hacia el exterior de la empresa, utilizando como estrategia la organización de la información, que ha pasado a ser parte del valor añadido del producto y las empresas se empiezan a diferenciar unas de otras por la forma de obtener, organizar, almacenar y presentar esa información de manera que resulte accesible para todos.

La evolución del internet está cambiando radicalmente la forma de hacer negocios, la automatización de los procesos de una empresa por este medio, ha dado como resultado un nuevo concepto: La empresa eBusiness, cuyos sistemas informáticos se encuentran sobre una arquitectura distribuida, lo que permite la conexión entre empleados, clientes, proveedores y distribuidores, es decir la integración de la cadena de valor, usando todos los canales disponibles. Obteniendo un alcance global, a precios locales. Un sistema de eBusiness completo integra soluciones para la administración de la cadena del suministro (SCM), la administración de los recursos de la empresa (ERP) y por supuesto la gestión de las relaciones con el cliente (CRM) para ayudar a las empresas a mejorar su eficiencia operativa y adaptación al entorno, explotando el uso de tecnologías basadas en la web.

El CRM, no es un concepto nuevo, hace 59 años Peter Drucker, filósofo de la administración dijo: “El verdadero negocio de cualquier compañía es crear y mantener a sus clientes”. Sin embargo, es una idea que ha tenido un verdadero auge en los últimos años gracias a la posibilidad que otorgan las tecnologías de la información para relacionarnos individualmente con cada cliente.

Por supuesto, el cambio en el mercado no puede atribuirse únicamente al avance en materia de tecnologías de la información (TI), también ha sido necesaria la combinación de otros factores importantes, para la renovación de los procesos de negocio de una empresa, tales como:

- La globalización de los mercados: las barreras comerciales han desaparecido, ahora la competencia de una empresa viene de cualquier parte, y el mercado se ha saturado de productos y servicios homogéneos, obligando a las empresas a enfocarse en ofrecer un valor agregado.
- La evolución del consumidor: El cliente se ha vuelto más exigente y requiere soluciones cada vez más personalizadas. La forma tradicional de fragmentar un mercado ya no es efectiva, ya que ahora existen muchos más mercados por lo que cada uno de ellos es menos numeroso.

- Nuevos canales de comunicación: Con la aparición de nuevos dispositivos y formas de comunicación cada vez más personales y portátiles, los medios de comunicación en masa y las estrategias de marketing para estos, han resultado cada vez menos efectivos y se ha vuelto más difícil asegurar que la información llega a quien fue dirigida.

Entendiendo la situación del mercado actual, no podemos decir que un CRM, sea únicamente un software y mucho menos una solución instantánea. Hablar de un CRM es hablar de una estrategia de negocio completa, habilitada por las tecnologías de la información, donde el cliente y sus necesidades se vuelven la columna vertebral de todo proceso, y donde es necesario integrar a toda la organización de la empresa. Por lo que muchas veces implica un rediseño total de sus procesos de negocio.

Un CRM, mediante el análisis centralizado de los datos de un cliente, busca entender, anticipar y satisfacer sus necesidades, para construir relaciones sólidas de largo plazo que generen un valor económico a la empresa.

Con una estrategia CRM la empresa será capaz de:

- Almacenar, integrar, segmentar y analizar toda la información de sus clientes.
- Personalizar y optimizar sus interacciones con el cliente siendo capaz de devolver los mensajes de marketing y servicio al cliente con el formato y contenido preferido por éste.
- Coordinar todos los canales de interacción con el cliente para que estén disponibles en cualquier momento y a cualquier hora.
- Evaluar las mejoras de interacción con el cliente.
- Personalización de la oferta y el servicio asociado al producto.

Una estrategia CRM se basa en el poder de la información, y en el hecho comprobado de que retener un cliente y crecer con él, es mucho más rentable que conseguir nuevos. Sin embargo existen otras ventajas al adoptar dicha estrategia, como por ejemplo:

- ✓ Hace posible controlar y medir los resultados de un negocio, en tiempo real o mediante históricos.
- ✓ Ayuda a una empresa a anticiparse a las prioridades cambiantes y a la dinámica del mercado, mediante la recopilación de información de valor.
- ✓ Permite a las organizaciones alinear a las distintas funciones existentes con un objetivo común.
- ✓ Incremento de la eficiencia y productividad y con ello, reducción de costos.
- ✓ Aumentar la rentabilidad de los clientes.
- ✓ Disponer de relaciones más duraderas con los clientes.
- ✓ Incrementar su competitividad en el mercado

Tipo de CRM

Un CRM, es un sistema integrado de información, cuyo valor central para la empresa consiste en mejorar la rentabilidad, los ingresos con un enfoque de satisfacción al cliente, sin embargo, existen diversos tipos de CRM que pueden ser implementados para alcanzar un objetivo específico.

- CRM Operacional: apoya los procesos front-office y administra todos los puntos de contacto a través de los que se recoge información sobre los clientes.
- CRM Estratégico: Suele ser el núcleo de la estrategia de negocio centrada en el cliente, lo cual resulta en ganar y conservar clientes más benéficos.
- CRM Analítico: comprende una serie de herramientas que analizan la información existente sobre clientes con objeto de generar perfiles, identificar patrones de compra, determinar el grado de satisfacción y hacer posible la segmentación. Incluye herramientas como almacenamiento y minería de datos.
- CRM Colaborativo: un CRM es capaz de integrar con otros sistemas tecnológicos existentes en la empresa, para dar mayor valor a la compañía, socios y clientes.

Tendencias tecnológicas en CRM.

- E-CRM: Utiliza el internet como canal preferente para el establecimiento y desarrollo de relaciones con los clientes. Permite la disponibilidad de la información en todos los puntos de contacto internos y externos de la compañía.
- Mobile CRM: incluye comunicación unidireccional o interactiva entre la empresa y el cliente utilizando un dispositivo móvil, lo que ofrece autonomía espacial y temporal en la interacción con los clientes.

Capítulo 1.- La empresa

Everis es una consultora multinacional que ofrece soluciones de negocio, estrategia, desarrollo y mantenimiento de aplicaciones tecnológicas, y outsourcing. La compañía, desarrolla su actividad en los sectores de telecomunicaciones, entidades financieras y de seguros, industria, energía, administración pública y salud. La consultora pertenece al grupo NTT DATA, la sexta compañía de servicios TI del mundo, con 70.000 profesionales y presencia en Asia-Pacífico, Oriente Medio, Europa, Latinoamérica y Norteamérica.

1.1. Organigrama

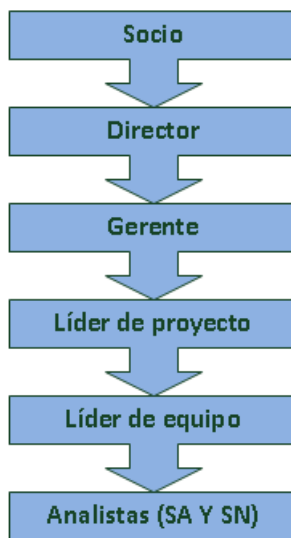


Figura 1. Organigrama de consultoría TI en everis

1.2. Descripción del puesto

Dentro de la compañía existen distintos modelos de carrera profesional, los cuales funcionan como itinerarios naturales de desarrollo, en función de cada una de las categorías que reúnen funciones, conocimiento y competencias requeridas.

La carrera de Consultoría en Tecnologías de la Información se desarrolla en el ámbito de una gran gama de servicios de TI, aplicados al ciclo de vida de integración, desarrollo y mantenimiento de aplicaciones y soluciones tecnológicas de mercado.

Algunas de las funciones establecidas para un consultor de nivel Solutions Assistant (SA), como es mi caso durante la realización del proyecto que elegí para este reporte, son las siguientes:

- ✓ Realizar el diseño técnico de las soluciones, programas y/o módulos.
- ✓ Diseñar pruebas unitarias.
- ✓ Programar/Parametrizar programas y/o módulos.
- ✓ Realizar pruebas unitarias y participar en las integradas.
- ✓ Documentar.
- ✓ Realizar tareas de apoyo a la ejecución del proyecto.
- ✓ Estimar esfuerzos de las tareas.

Además dentro de esta categoría se exigen las siguientes aptitudes:

- Trabajo en equipo
- Flexibilidad
- Autonomía
- Solvencia
- Fiabilidad

Además de las carreras, existen también las unidades de negocio (UDN), para enfocar a los profesionales a la especialización dentro de alguna de las áreas o industrias. Pertenecer a una unidad de negocio, determina los proyectos a los que se asignará a una persona.

En la unidad de negocio de tecnología, de la que formo parte, tenemos como meta facilitar a nuestros clientes el cumplimiento de sus objetivos de negocio por medio de la exitosa implantación de un gran abanico de soluciones tecnológicas, metodologías y/o arquitecturas. Algunas de las cuales se mencionan a continuación, por ser las más desarrolladas en muchos proyectos dentro de la compañía.

- Big Data
- Definición e implantación de arquitecturas tecnológicas
- Business Intelligence
- Oracle Specialization Group que incluye las líneas de productos bajo tecnología Oracle:
 - ✓ SOA (Arquitectura orientada a los servicios)
 - ✓ Oracle Business Intelligence (OBI)
 - ✓ e-Business Suite
 - ✓ Siebel CRM
 - ✓ Oracle BPM suite (Gestión de los procesos de negocio)
- Cómputo en la nube

En el presente reporte de experiencia profesional, se desarrollará un proyecto realizado en la unidad de negocio (UDN) de tecnología y su práctica de Oracle, utilizando su aplicación más robusta de CRM, Siebel.

Capítulo 2. Breve descripción de participación en otros proyectos.

Durante el tiempo que he ejercido mi carrera de forma profesional he participado en proyectos para clientes de diversas industrias como Banca, Seguros, Telecomunicaciones y empresas de venta masiva de productos (Retail). Lo que me ha permitido adquirir conocimientos específicos, no sólo en el terreno de la informática, sino del negocio en particular, lo cual aporta mucho valor a un profesional de la ingeniería.

- **Implementación de CRM On Demand en empresa de Telefonía Móvil**

El proyecto de CRM On Demand en dicha empresa de telecomunicaciones, surgió como respuesta a la necesidad de mejorar el desempeño en la fuerza de ventas, para impulsar un área de reciente creación dentro de la compañía.

Se trató de un proyecto corto por las características y ventajas que ofrece una herramienta On Demand que busca la menor cantidad de configuraciones en comparación con una solución a la medida. Las actividades que realicé fueron desde el análisis del proceso de venta, la optimización, su implementación en la herramienta y la formulación de reportes, hasta la capacitación de los usuarios mediante cursos y manuales de usuario. Los clientes a lo largo de su relación con la empresa pasan por distintas etapas, el conocimiento de la fase del ciclo de vida en la que se encuentra el cliente, permite entre otras cosas, personalizar acciones de marketing, obteniendo mayor rentabilidad. Para cumplir con esto, la empresa deberá realizar propuestas concretas a cada cliente individual, ofreciendo la misma cara en todo el ciclo de venta.

Un ciclo de venta clásico comienza con las actividades de marketing realizadas sobre los prospectos (clientes potenciales), para convertirlos en oportunidades y finalmente en clientes. Antes de cerrar una venta, el cliente pasa por una serie de actividades que dan como resultado la oferta de productos y servicios a éste, seguidas de las actividades de servicio destinadas a fidelizar al mismo. Mediante el análisis de estas interacciones podremos lanzar campañas ofreciendo productos y ofertas adicionales que puedan interesar al cliente, cerrando el círculo y colocando de nuevo al cliente en su posición natural de prospecto.

Los distintos estados por los que pasará el cliente dentro de un ciclo de vida, deben ser analizados de manera que la empresa sea capaz en el futuro de reutilizar esa información en nuevas interacciones.

- **Implementación de MDM en la base de datos de clientes para una empresa de Retail.**

El proyecto de Master Data Management (MDM) sobre la base de datos de clientes de dicha empresa de retail, tuvo como objetivo asegurar la calidad y la consistencia de los datos maestros, es decir de la información estática de la organización. El resultado de una base de datos maestros consistente y fiable, se refleja en todos los procesos donde se generan datos transaccionales, afectando directamente el cumplimiento de los objetivos de negocio de toda organización. Algunas de las tareas básicas de un proceso de MDM son: Identificar el origen de los datos y quien los consume, eliminar duplicados, elegir una estrategia de estandarización y un gobierno de datos así como una estrategia de mantenimiento y estandarización de los nuevos datos, etc.

Mi participación en el proyecto consistió en la coordinación y ejecución de las tareas de identificación de registros duplicados o redundantes, lo cual se realizó mediante procesos Batch, utilizando Siebel UCM como motor de deduplicación. Los procesos encargados de la deduplicación ejecutaban las siguientes tareas: Se segmenta un flujo de datos entrante, se identifican los segmentos de datos de manera exclusiva y, luego, se comparan con los datos almacenados anteriormente. Si el segmento es único, se almacena en el disco. Sin embargo, si un segmento de datos entrante es un duplicado de uno ya almacenado, se crea una referencia a éste y el segmento no se almacena nuevamente.

Después de la deduplicación se obtuvo una mejora significativa en requerimientos de almacenamiento en la base de datos de clientes, contribuyendo significativamente con el objetivo global de la administración de los datos maestros.

- **Proyecto de Modelado de procesos e implementación BPM en empresa de seguros.**

Este proyecto de gestión de procesos de negocio (BPM), surgió como necesidad de la redefinición estratégica del área de Gastos Médicos con foco en la administración de los siniestros, lo cual incluía la renovación de su plataforma tecnológica.

Durante mi participación en este proyecto, lo primero que se realizó fue el modelado de procesos As Is (¿cómo funciona ahora?), con el fin de entender cómo se hacían las cosas al momento del inicio del programa de reestructuración del sector. Para esto, tuve asignados 4 de los ocho procesos y participé directamente en las sesiones de definición de proceso que se llevaron a cabo con el cliente, específicamente con los dueños de proceso y usuarios involucrados, para modelar cada tarea a lo largo del flujo, identificar actores, documentos de entrada y salida y puntos de integración con otros sistemas, principalmente. Todo esto con el objetivo de posteriormente poder realizar un rediseño de procesos que cubra la necesidad actual pero de forma mucho más eficiente y acorde a las nuevas soluciones tecnológicas previstas.

Estas soluciones, incluían un nuevo Core (núcleo de sistemas), que englobará el resto de los sistemas distribuidos que se tenían hasta el momento, además de una solución BPM con los procesos actuales rediseñados y un nuevo Front. Con las ventajas que la nueva estrategia BPM implica, como la capacidad de obtener información para la generación de reportes de Business Intelligence dentro de la operación de gastos médicos.

Una vez modelados los procesos As Is, iniciamos el rediseño de dichos procesos, para llevarlos al modelo To Be, que plasmaba la nueva ruta de cada proceso. Este modelo tuvo que validarse de nuevo con las mismas personas con quienes se definió el As Is en un nuevo ciclo de workshops, donde se recogieron opiniones, se hicieron correcciones al modelo propuesto, hasta llegar a un diseño que tuviera el visto bueno de todas las partes involucradas. Todo este modelado se hizo bajo el estándar mundial de BPM (BPMN), ya que posteriormente esto sería implementado en IBM BPM.

Después de que los procesos fueron aprobados también me encargue del análisis y diseño de KPI (indicadores clave de desempeño) para cada proceso que del mismo modo, se presentaron al cliente para su retroalimentación, ajustes y posterior aceptación. Estos KPI, fueron utilizados para desarrollar 5 tableros de control que proporcionaban información mucho más fiable y útil sobre el rendimiento del proceso.

- **Proyecto de mantenimiento correctivo y evolutivo de Oracle Siebel CRM 8 en otra institución bancaria chilena.**

Al igual que el proyecto principal que se desarrolla en el presente informe, este proyecto fue realizado en Santiago de Chile, para otra institución bancaria chilena. El CRM en el banco es utilizado para gestionar las solicitudes de servicio asociadas a todos los clientes y productos en el banco. La iniciativa del proyecto surge a raíz de la necesidad de cambiar ciertas capacidades en la funcionalidad actual que utiliza el banco en su herramienta Siebel versión 8.

Algunos de los evolutivos que realicé, consistieron en la incrustación de elementos externos a Siebel, como una página Web o el portal de otro sistema utilizado al interior del banco, esto con la finalidad de condensar la mayor cantidad de elementos necesarios para atender las solicitudes del cliente, en el mismo sistema para mejorar la atención y el rendimiento en los procesos. Todos estos desarrollos evolutivos implicaron una etapa de análisis de requerimientos, diseño técnico y funcional, construcción, pruebas y soporte.

El proyecto también incluyó una gran cantidad de tareas relacionadas al mantenimiento correctivo, es decir, a la corrección de incidencias reportadas por los usuarios del banco. Las incidencias se presentaban a distintos niveles y con diversas prioridades, es decir, existieron incidencias funcionales cuyo error residía en la definición de alguna funcionalidad o bien incidencias técnicas que no permitían el correcto desempeño del proceso de atención a las solicitudes de servicio, por errores en la configuración de la herramienta.

Capítulo 3. Implementación de un evolutivo en Oracle Siebel CRM para una institución bancaria chilena.

3.1. Antecedentes generales

Durante un periodo de casi 9 meses, tuve la oportunidad de viajar y vivir en el extranjero para participar en un proyecto en Santiago de Chile, en la segunda institución bancaria más grande de este país sudamericano.

Este proyecto, fue desarrollado en una de las líneas más relevantes dentro de la Unidad de Tecnología en la oficina de everis Chile, la denominada Enterprise Business Applications en la que se desarrollan proyectos de Oracle Siebel CRM y eBusiness Suite (ERP), principalmente.

La institución bancaria donde laboré, que a partir de ahora llamaré: el Banco, actualmente cuenta con la herramienta Oracle Siebel dentro de su núcleo de negocio para manejar a los clientes y sus productos. Sin embargo, se detectó la necesidad de realizar mejoras a las soluciones existentes que podrían derivar en la implantación de nuevas soluciones para mejorar el desempeño y por ende la rentabilidad del negocio. Además se han reportado nuevos requerimientos que hasta ahora no se consideran en el sistema CRM y que se han vuelto indispensables en la operación del banco.

Con ello, surge este proyecto que de acuerdo a la clase de actividades requeridas, incluyó mantenimiento evolutivo y correctivo.

El mantenimiento evolutivo, consiste en la realización de todas aquellas tareas encaminadas a la implementación de desarrollos y/o configuraciones que hagan evolucionar las funcionalidades ya implantadas en la herramienta, o bien, una nueva funcionalidad o proceso. Para ello, se conformó un equipo de trabajo que permita evolucionar en base a los requerimientos levantados por las áreas de negocio del banco.

Se considera mantenimiento correctivo a la realización de todas las tareas encaminadas a resolver errores en la operación, errores por parte del usuario y errores en la aplicación. En general, esto se entiende como la resolución de incidencias con intervención sobre el sistema, pero sin alterar las especificaciones funcionales o técnicas de la aplicación.

Dichas incidencias pueden afectar a las funcionalidades utilizadas por los procesos o usuarios, o simplemente a cuestiones técnicas que impidan un normal funcionamiento de operación del sistema.

Posterior al paso de este proyecto a producción, todavía fue necesario contar con un proceso de mantenimiento para seguir generando mejoras sobre la aplicación y nuevos desarrollos acorde con el lineamiento de negocio y normativas del Banco.

3.2. Objetivos y alcances generales

Dentro de la realización de todo proyecto, uno de los puntos más importantes y por ende más controversiales sobre todo cuando se trata de un servicio de consultoría, es la definición de objetivos y alcances en la implementación de las soluciones acordadas. Esta definición debe hacerse con pleno conocimiento de lo que se hará, sin dejar de considerar riesgos y sobre todo haciendo énfasis en las expectativas tecnológicas y de negocio de todas las partes involucradas.

Los objetivos generales del presente proyecto fueron los siguientes:

1. Contar con un equipo que sea capaz de evolucionar la aplicación, a través de la construcción de nuevos requerimientos en la herramienta, apoyando al Banco en el ciclo de vida completo de estos requerimientos.
2. Entregar soporte en el paso a producción y post producción a los evolutivos que desarrolle everis dentro del servicio.
3. Supervisar y asegurar la calidad mediante la ejecución de pruebas para garantizar el correcto funcionamiento de la aplicación, antes de su pase a producción.
4. Catalogar y archivar la documentación (entregables técnicos, funcionales y de gestión del proyecto) y de los componentes desarrollados (fuentes del código).
5. Apoyar en el modelo de operación basado en peticiones (ordenes de trabajo) priorizadas y administradas por el Banco.
6. Contar con una pirámide concreta que permita escalar de forma clara los requerimientos, en caso que no se cuente con la respuesta en base a los SLA definidos y la criticidad que éstas tengan para el negocio.

Para el servicio de mantenimiento evolutivo y correctivo en el Banco, se definieron los siguientes alcances:

1. No se considera la compra de hardware ni de software en el transcurso del servicio.
2. No se contempla la redefinición de la arquitectura tecnológica incluyendo el diseño y funcionamiento de servidores, instancias, telecomunicaciones, etc.
3. Actualmente, el Banco no cuenta con una herramienta para el control de las fuentes (objeto original), por lo tanto, se debe acordar e informar todos los objetos y/ componentes a modificar.
4. El equipo Siebel es responsable de entregar componentes y su documentación de instalación para todos los ambientes distintos al de desarrollo, pero no es responsable de instalarlos. El Banco es responsable de las instalaciones en el ambiente QA (quality assurance, ambiente de aseguramiento de la calidad) y en el ambiente de producción.
5. El Banco es responsable del equipo QA y de los tiempos considerados para la certificación de los evolutivos en el ambiente de certificación.

6. Las pruebas del equipo Siebel se realizarán únicamente en ambiente de desarrollo.
7. En cualquier estado del evolutivo, el Banco tiene la facultad de cancelar el requerimiento en curso.
8. La criticidad de las incidencias será determinada por el equipo de QA, en función de los antecedentes proporcionados por el usuario que registre el problema.

Acorde con este alcance establecido y dado que el proyecto fue pactado como un proyecto de mantenimiento correctivo y evolutivo a continuación se detallan las actividades a realizar por el equipo Siebel:

Mantenimiento correctivo:

- Validar funcionalmente y a nivel de documentación, los pasos a producción por cambios ocasionados por incidentes.
- Análisis de fallas y propuesta de soluciones de contingencia que minimicen el impacto de la incidencia, ya sea correcciones a las parametrizaciones o cambios realizados con Siebel Tools.
- Diagnóstico de problemas.
- Documentar los cambios realizados para resolver los problemas.
- Coordinar con el cliente el paso a producción de las correcciones.
- Realización de las actividades del Ciclo de Vida del Desarrollo del Software necesarias para generar una nueva versión.

Mantenimiento evolutivo:

- Gestión, Control y Seguimiento del Servicio.
- Plan de Mejora Continua.
- Soporte evolutivo a la implementación de Siebel, incluyendo:
 - Parametrizaciones de Siebel que permitan soportar nuevas funcionalidades.
 - Construcciones con Siebel Tools que permitan modificar la actual funcionalidad o incorporar nuevas funcionalidades.
 - Documentar las modificaciones y nuevas funcionalidades.
 - Coordinar con cliente el paso a producción de las nuevas funcionalidades.

El ciclo de vida de todos los nuevos evolutivos y mejoras que marcará las obligaciones de cada una de las partes y será una guía para las actividades que deberán considerarse en la estimación de cada evolutivo se muestra a continuación:

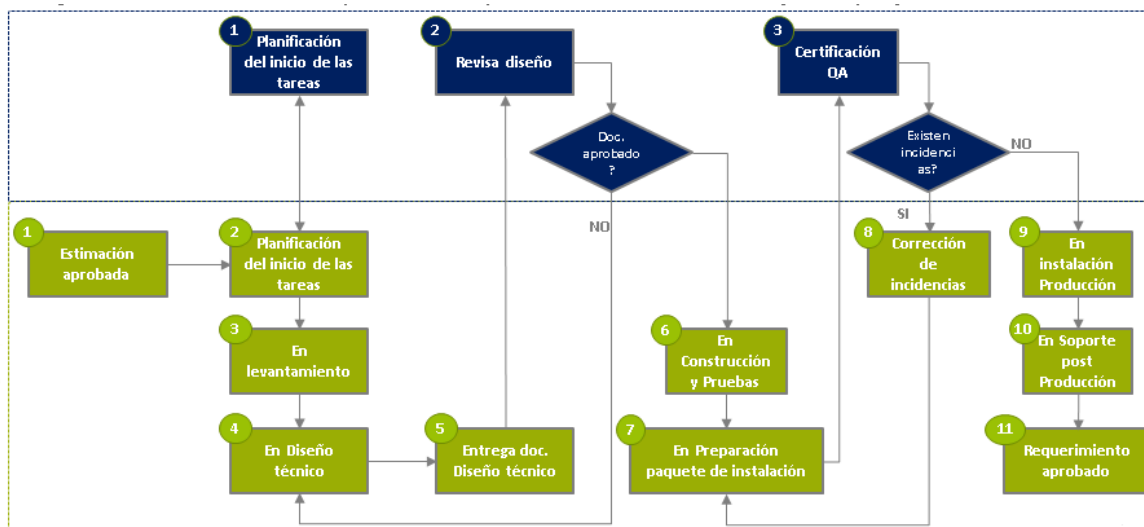


Figura 2. . Ciclo de vida de un evolutivo Siebel con interacción cliente y equipo desarrollador

Además de todo lo anterior, las funciones de cada una de las personas involucradas en el equipo Siebel, está determinada por su rol dentro del proyecto. En mi caso, mis funciones fueron las correspondientes a un analista funcional Siebel y también como desarrollador Siebel. El analista funcional, es el perfil del equipo Siebel encargado de la interpretación de los requerimientos por parte del banco, para su diseño funcional, donde se plasmen los problemas a resolver por cada requerimiento, y lo que debe cumplirse, también es quien válida por parte del equipo Siebel que la solución construida cumple plenamente lo solicitado.

Una vez que se tiene el diseño funcional, es el programador Siebel quien continúa con el desarrollo, iniciando con el diseño técnico de la solución, lo cual implica un análisis de todo lo que se hará en las tres capas que considera la herramienta: capa de datos, capa de negocio y capa de interfaz de usuario (Estas capas se explicarán a detalle más adelante durante el desarrollo del evolutivo) con el objetivo de cumplir con todo lo planteado en el diseño funcional. También es competencia del programador Siebel, construir, configurar, realizar las cargas manuales y todas las tareas requeridas que resulten en una nueva funcionalidad para el CRM del Banco. Por último, las tareas del programador Siebel terminan en el diseño y la ejecución de las pruebas en el ambiente de desarrollo y el paso del evolutivo al ambiente de pruebas. También fue de mi competencia el apoyo en los pases a producción para verificar la correcta instalación de los objetos de desarrollo y las cargas manuales.

3.3. Metodología de implementación

La metodología utilizada en la gestión y ejecución del proyecto fue CMMI (Integración de modelos de madurez de capacidades). Esto se ejecuta a través de un repositorio en el servidor de control de versiones el cual contiene la información relevante con respecto a los pasos y soluciones entregadas en los distintos desarrollos de evolutivos.

En el siguiente diagrama se enlistan algunos de los entregables propios de la metodología en cada etapa del proyecto:

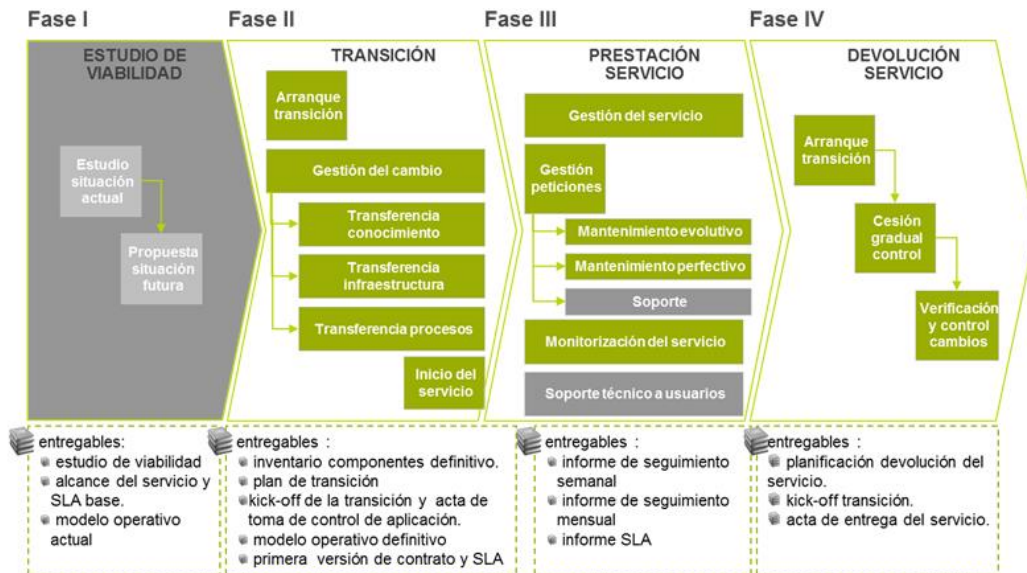


Figura 3. Fases del proyecto y entregables CMMI

Además de CMMI en la administración del proyecto, para los documentos entregables de cada evolutivo, se utilizó el formato propio de la metodología AIM de Oracle, a petición del banco. Dichos documentos fueron:

- Diseño Funcional: Formato MD050
- Diseño Técnico: Formato MD070
- Casos de prueba: Formato TE040

3.4. Desarrollo del evolutivo

3.4.1. Marco teórico

El presente proyecto, se desarrolló en su totalidad sobre la plataforma Oracle CRM Siebel versión 7.7. Por ello, es importante iniciar el desarrollo del mismo con una introducción a Siebel, donde se destaquen los conocimientos más importantes que facilitarán el entendimiento del resto del proyecto realizado.

Siebel CRM, es una aplicación web empresarial adquirida por Oracle, que se caracteriza por ser una de las más robustas de su tipo en el mercado. Es una tecnología que requiere la reestructuración del negocio para enfocar los procesos en el cliente ya que permite administrar y potenciar las relaciones entre los clientes y los empleados mediante múltiples canales de comunicación. Sincroniza todos los puntos de contacto entre ellos mediante un repositorio central de información, una base de datos única, un conjunto de herramientas de desarrollo y una arquitectura propia.

En su versión “vanilla” (sin personalización o también llamado “out of the box”) tiene un enfoque orientado a las ventas y el marketing, sin embargo, es totalmente configurable y cuenta con una gran cantidad de herramientas para ello, lo que permite que se ajuste a casi cualquier negocio y necesidad.

- **Arquitectura Siebel**

La arquitectura web de Siebel a un alto nivel se compone de los siguientes elementos:

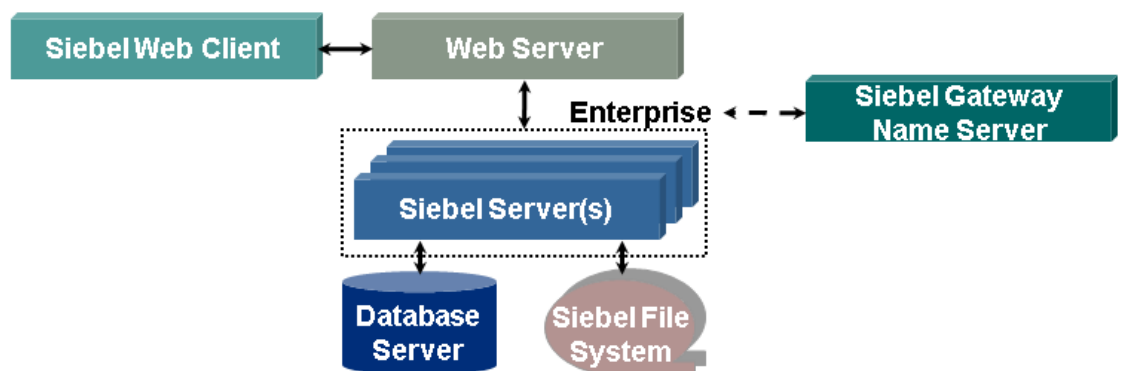


Figura 4. Arquitectura web de Siebel

- ✓ Siebel Web Client:

Despliega la parte interactiva y visual de Siebel. Se puede ejecutar en una gran variedad de ambientes e incluso dispositivos móviles.

- ✓ Web server :

Identifica y lleva las peticiones del Siebel Web Client a los servidores de Siebel.

Entrega al Siebel Web Client la aplicación HTML completa.

✓ Siebel Gateway Name Server

En Windows es un servicio, en UNIX es un demonio encargado de registrar dinámicamente a los servidores y la disponibilidad de los componentes, también almacena información de conectividad.

✓ Siebel Enterprise

Es una colección lógica de los servidores de Siebel que soporta el acceso de usuarios a un solo servidor y una base de datos única y a un solo sistema de archivos.

✓ Siebel Servers

Ejecuta tareas interactivas, de segundo plano o en proceso batch mediante componentes del servidor que administran los datos del negocio.

Dentro de este arreglo de servidores se encuentran los componentes del servidor. Uno de los más importantes es el Object Manager encargado de leer los parámetros de configuración y de los componentes así como las especificaciones de la aplicación.

Todo esto da como resultado la página web completa de la aplicación, que es enviada al Web server.

✓ Siebel File System

Es el directorio compartido que almacena los archivos utilizados por las aplicaciones Siebel, como los atributos de usuario.

✓ Database server

Almacena datos utilizados en las aplicaciones Siebel bajo un esquema de base de datos predefinido. Esto garantiza la consistencia de los datos. Soporta una gran cantidad de manejadores de bases de datos relacionales.

- **Arquitectura de una aplicación Siebel**

Una aplicación Web de Siebel completa tiene los siguientes elementos:

Una base de datos con la información de negocio.

Archivos de configuración (CFG): Tiene toda la información de configuración de la aplicación como localización de los servidores, puertos de conexión al servidor, al enterprise y al object manager, configuración de seguridad, etc.

Archivo de repositorio Siebel (SRF): Es un archivo binario que define el comportamiento de la aplicación y la lógica del negocio. Especifica la presentación de los datos, todos los procesos y reglas de negocio, todos los objetos de

compilación y su interacción, así como la organización y el almacenamiento de los datos físicos.

Plantillas de interfaz web (SWT): plantillas de HTML que especifican como se presentará la interfaz de usuario.

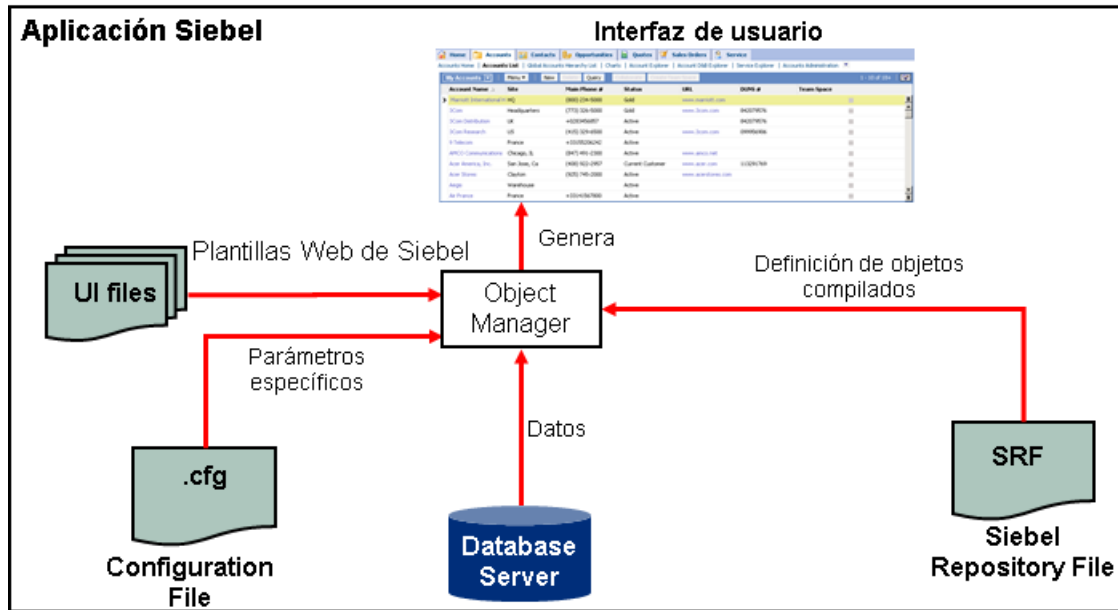


Figura 5. Arquitectura de una aplicación de Siebel

Derivado de todos los elementos que se requieren para poder tener una aplicación CRM en Siebel, cuando se habla de configuración y personalización de la herramienta, se mencionan 3 niveles de trabajo:

- Capa de datos (Data layer)
 - Especifica la estructura lógica de la base de datos.
 - Las definiciones son metadatos, no datos.
 - Los objetos que se definen en esta capa son las tablas y sus columnas.

 - Capa de negocio (Business layer)
 - Se especifica la lógica de negocio para la aplicación.
 - Los dos objetos principales son: Business Objects y Business Components.
- ✓ Un Business Component (BC) representa la unidad de negocio fundamental en la compañía. Está compuesto de diversos campos y agrupa lógicamente los datos de una o varias tablas, haciendo referencia a las columnas de la base de datos.

- ✓ Un Business Object (BO) es una colección de Business Components relacionados entre sí. Representa un área de negocio más grande dentro de la compañía.

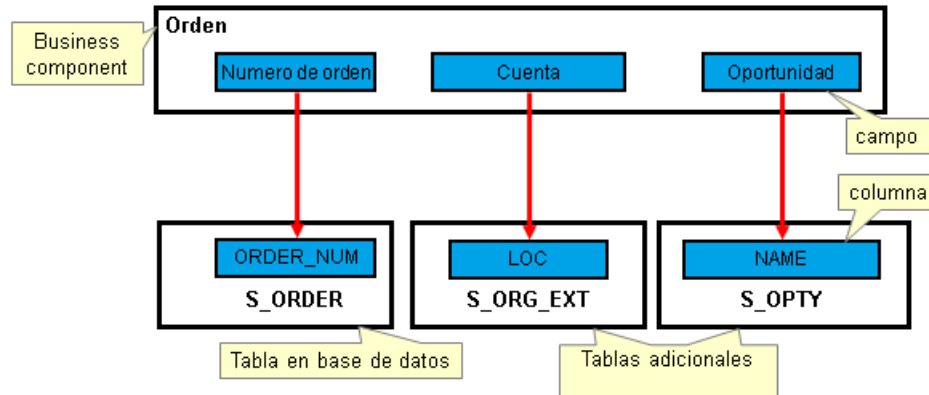


Figura 6. Cómo se define un business component

- Capa de interfaz de usuario (UI layer)

En la última capa de la arquitectura de una aplicación Siebel, se encuentran las representaciones visuales de los objetos de la capa de negocio, permitiendo a los usuarios manipular los datos. Dentro de la capa interfaz de usuario, existen 5 elementos básicos a manipular:

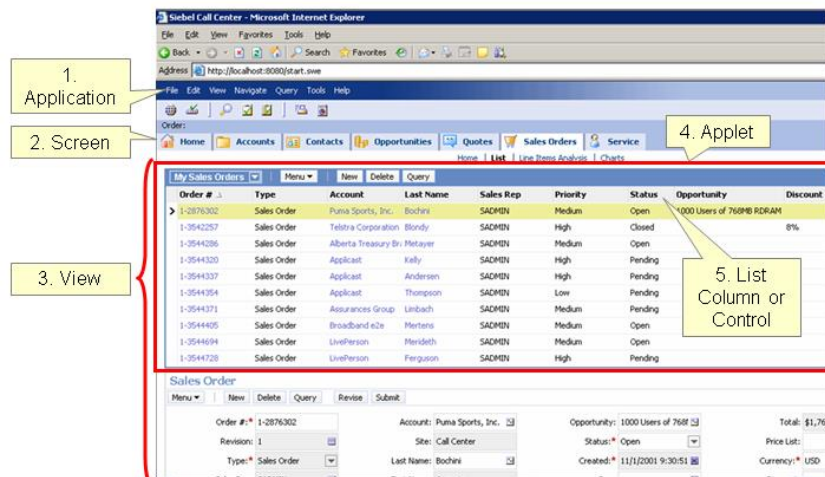


Figura 7. Elementos en la interfaz de usuario de Siebel.

- List column or control: muestra y permita manipular los datos ya sea en una columna o en un campo.
- Applet: es la sección de una vista que hace referencia a los datos contenidos en un Business Components.

- Vista: es el compendio de applets dentro de la aplicación, por lo tanto hace referencia a diversos Business Components, por lo tanto una vista hace referencia directamente a los Business Objects.
- Pantalla: Contiene múltiples vistas que definen las pantallas de la aplicación. Dentro del negocio se asocia a un área de negocio aún mayor.
- Aplicación: es una colección particular de pantallas disponibles en una aplicación Siebel.

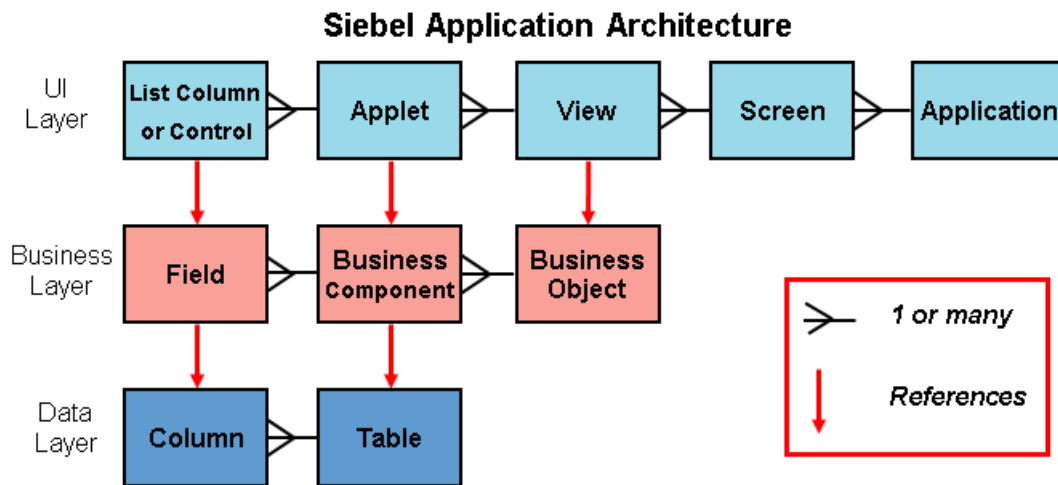


Figura 8. Interacción entre las tres capas de la arquitectura de una aplicación Siebel

Para la configuración de los objetos de las tres capas, Siebel tiene su propio ambiente de desarrollo: Siebel Tools, que no es un ambiente de programación, sino una herramienta de configuración de aplicaciones declarativa que permite configurar el conjunto de objetos de la capa de datos negocio e interfaz de usuario sin necesidad de modificar el código fuente (aunque es posible hacerse también a este nivel). Posee una interfaz distinta al Web Client. Dentro de Siebel tools, se compilan todos los objetos y se genera el SRF que alimenta la aplicación.

3.4.2. Planteamiento del problema

El Banco requiere que las áreas comerciales y operativas que tienen contacto directo con los clientes, no visualicen las Cuentas internas que se utilizan para gestión. Para eso, tendrían que cumplirse ciertas condiciones que formarán parte de la lógica de filtrado de información, así se garantice que ciertos roles dentro de la organización no puedan cometer el error de informar a un cliente sobre una cuenta que él no ha solicitado.

De lo anterior, es que se requiere generar una solución en Siebel de tal forma que aquellas Cuentas internas (CI) creadas para propósitos administrativos internos solo puedan ser visualizadas por áreas operativas permitidas que no tienen contacto directo con los clientes. Cada cuenta corriente posee un campo llamado "Clase de Cuenta", que permite la clasificación de las cuentas corrientes en el banco y así poder determinar qué cuentas están dentro del alcance del proyecto.

A la fecha existen aproximadamente 25,000 de ellas, considerando todas las clases de cuentas internas, de éstas se requiere inhibir la visibilidad de un estimado de 20,000 que cumplen con todas las condiciones para su inhibición. Con la solución propuesta no se considera un aumento en la creación de este tipo de cuentas internas, seguirán el curso normal.

3.4.3. Objetivos y alcances del evolutivo

Como se describió en el punto anterior de este documento, el objetivo del proyecto es permitir que sólo aquellas áreas operativas donde su función diaria está relacionada al tratamiento y revisión de cuentas internas puedan verlas. Con ellos se garantiza que las áreas que tienen contacto directo con cliente no tengan acceso a información interna, haciendo más eficiente y certero el servicio al cliente, sin importar el punto de contacto con éste. Lo cual es el objetivo global de implementar en el banco una herramienta como Siebel.

En cuanto al alcance de este evolutivo se pactó lo siguiente:

Un pre-requisito de la solución propuesta es el envío del campo "Clase Cuenta" a Siebel. Dicho pre-requisito sería cumplido mediante otro proyecto.

En Siebel deberá existir la responsabilidad destinada para visualizar todas las cuentas internas. Esta responsabilidad será, en producción, la asociada a los usuarios de las CI, permitiendo una visualización sin filtros. Con la solución propuesta no se considera la creación de cuentas internas. A la fecha las CI que se inhibirán representan el 86% del total.

El alcance del requerimiento sólo abarca la inhibición de cuentas internas en el sistema Siebel, no considerando otros canales electrónicos internos.

3.4.4. Especificación de requerimientos funcionales y no funcionales

La principal necesidad que posee el negocio es:

Inhibir la visualización de Cuentas Contables Internas del Banco para aquellas áreas que posean un contacto directo con el cliente, ya sea que éste sea persona o empresa. Se deben crear pantallas con filtros para garantizar que aquellos usuarios de Siebel que posean determinadas responsabilidades (de acuerdo a sus funciones en el banco), no visualicen información restringida. Cada cuenta corriente interna en Siebel posee una clasificación que se indica en el campo clase de cuenta y un RUT (rol único tributario). Debe ser dicho RUT otro dato necesario para determinar si se aplicará al usuario un filtro de información o no.

Actualmente, mediante las pantallas de Siebel “Productos/Servicios”, “Personas” y “Empresas”, es posible obtener información de las CI, lo cual debe ser inhibido en determinados casos. Por lo tanto, se requiere trabajar con responsabilidades de Siebel asociadas a pantallas como las actuales, pero con filtros aplicados.

Se deben considerar las pantallas, vistas y applets relacionados con las distintas formas de acceder a la información de las CI en Siebel.

De forma general el evolutivo puede conceptualizarse con el siguiente diagrama:

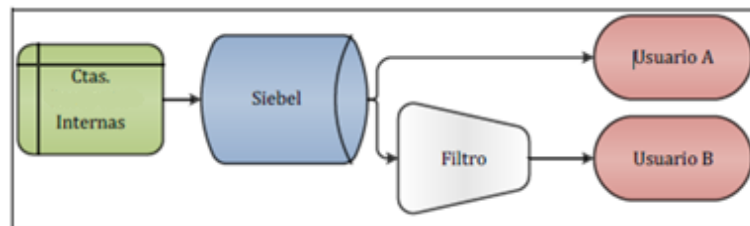


Figura 9. Representación general del evolutivo

En la figura anterior se observa, por ejemplo, que el usuario B posee una visión parcial (restringida) de los datos de las cuentas corrientes, ya que existe un filtro al acceder a la información de Siebel. La cantidad o tipo de usuarios impactados por el proyecto depende de las responsabilidades existentes.

Requisitos para el filtro:

Las condiciones que se deben ser evaluadas para determinar si corresponde inhibir la visualización de una cuenta corriente interna son:

- Clase de Cuenta: es el tipo de cuenta corriente informada para determinar si será inhibido o no.

Debe garantizarse que pueda ser modificable a futuro si es necesario, para no requerir de desarrollos adicionales ni afectar la lógica de filtrado si se cambia el valor de Clase Cuenta

Sólo será evaluada la posible inhibición de las cuentas internas que posean la clase de cuenta informada por el banco, originalmente el valor de clase cuenta a considerar es: CIN (pago de nómina) o CIV (postergación de cuotas).

- RUT: es un valor que informa el RUT del Banco que debe estar informado para el proyecto y debe garantizarse que pueda ser modificable a futuro si es necesario, sin alterar el desempeño del filtro para no requerir de desarrollos adicionales. Este valor debe evaluarse tanto en el campo RUT empresa como en el campo RUT persona.

Nombre	Tipo dato	Campo
Rut Empresa	String	Account RUT
Rut Persona	String	Person ID

Tabla 1. Descripción del campo RUT

Realizando un cruce de posibilidades de acuerdo a las condiciones, no considerando Clase Cuenta, ya que lo que NO sea lo informado, no pasa a evaluarse su inhibición. Entonces, se tiene:

Responsabilidad informada	RUT	Inhibir visualización
SI	SI	NO
NO	SI	NO
SI	NO	NO
NO	NO	SI

Tabla 2. Cruce de posibilidades entre responsabilidad y RUT

Revisando el primer caso, se tiene que si un usuario pertenece a responsabilidad informada (SI) y además la cuenta corriente consultada está asociada a un RUT BH (Valor del Rol Unico Tributario proporcionado por el banco para considerar en el filtro) (SI), entonces la inhibición no se realiza (NO). Se asume que cuando un usuario pertenezca a una responsabilidad informada, deberá estar asociado a la responsabilidad que tiene acceso a visión completa de las cuentas corrientes.

En resumen, puede afirmarse que:

- Las responsabilidades informadas deben visualizar todas las cuentas internas existentes.
- Las cuentas internas asociadas al “RUT BH” deben estar visibles para todas las responsabilidades.

• **Situación actual**

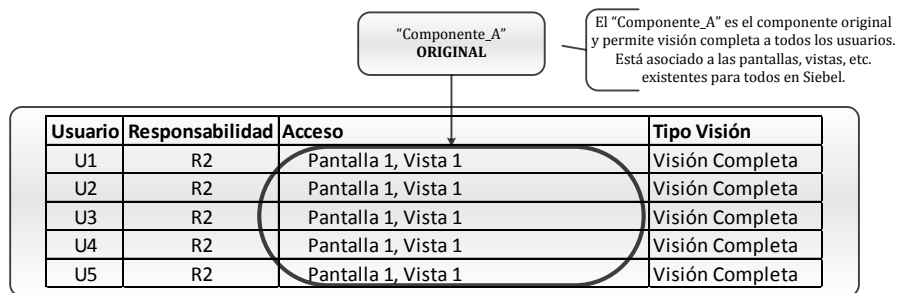


Figura 10. Situación actual de visibilidad CI

• **Situación futura**

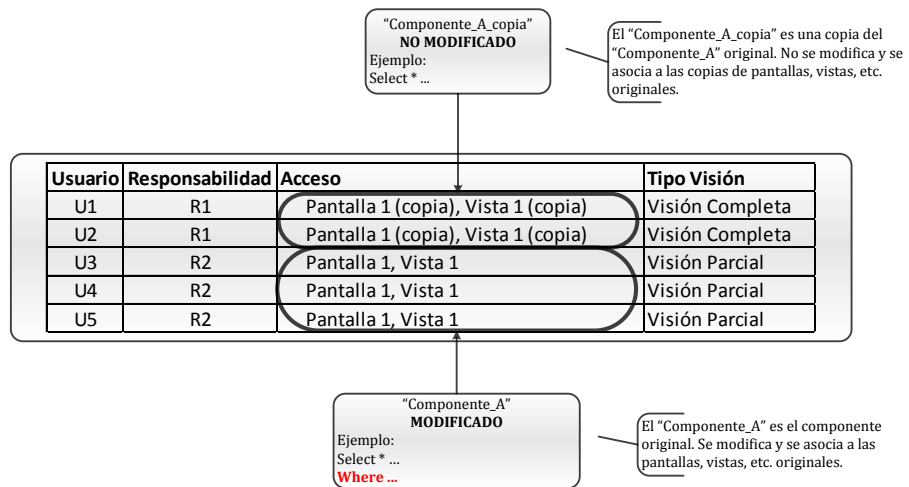


Figura 11. Situación futura de visibilidad CI

Del diagrama futuro se puede deducir que:

- Los usuarios (ej. U1, U2, U3, etc.) actualmente tienen visión completa, sin embargo, sólo el U1 y U2 pertenecen a una responsabilidad informada, por lo tanto, se requiere que estén asociados a una nueva responsabilidad (R1) que les permita un acceso a pantallas y vistas (ejemplo: Pantalla 1 (copia) y Vista 1 (copia)), las que a su vez están asociadas al “Componente_A_copia” que no tendrá cambios (No Modificado).
- Actualmente la responsabilidad utilizada por el banco es R2, la cual permite que todos los usuarios tengan una visión completa. La responsabilidad seguirá estando asociada al “Componente_A”, pero con modificaciones que permitirán tener una visión parcial.

3.4.5. Diseño funcional

3.4.5.1. Lógica de pantallas

A continuación se identifican los lugares donde es posible obtener la información de las cuentas CI, esto con el objetivo de identificar todos los objetos y componentes de negocio o interfaz que deben modificarse. El usuario posee actualmente una serie de pantallas que permiten la consulta y visualización de información. Aquellas áreas que no requieran visualizar cuentas internas, deben estar filtradas. Los procedimientos necesarios para visualizar una CI en las pantallas de empresas, personas y productos servicios son:

Navegación a pantalla productos/servicios: En este applet no debería visualizarse la CI seleccionada.

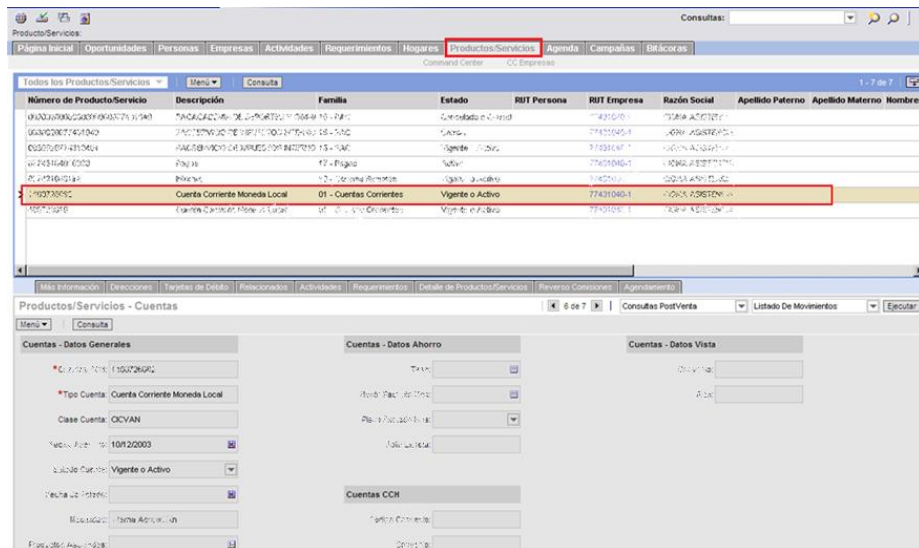


Figura 12. Pantalla productos/servicios con CI

Dentro de esta misma pantalla, si se navega a las vistas “Más Información” y “Detalle de productos/servicios” también aparece la CI, donde en cuyo caso no debería visualizarse.

a) Vista más información

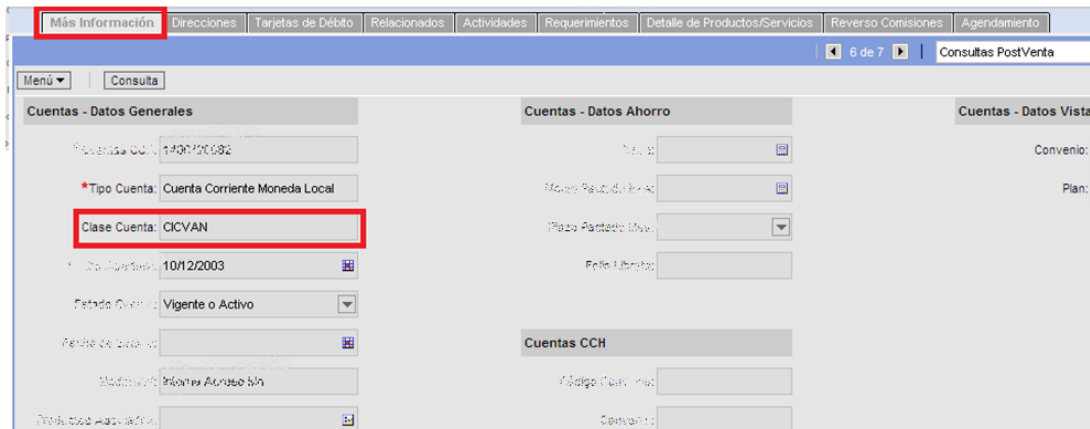


Figura 13. Vista más información con CI

b) Vista Detalle de productos/servicios: en este caso, la CI no debería visualizarse en la sección “Datos Cuenta”.

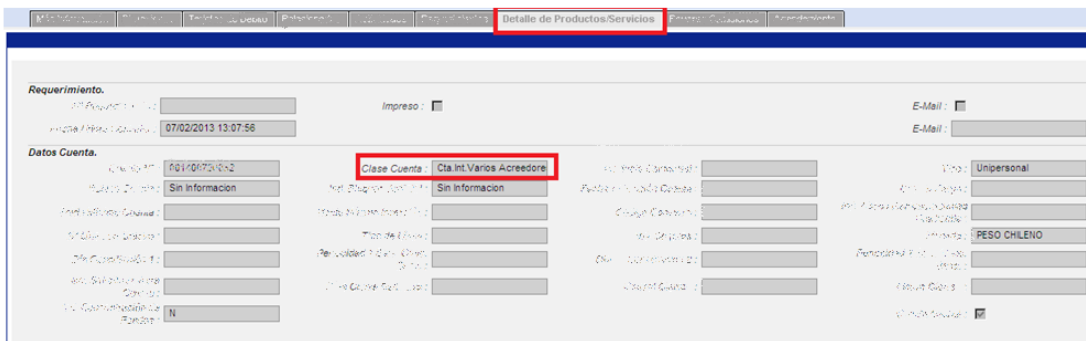


Figura 14. Vista detalle de productos/servicios con CI

Navegación a pantalla empresas: Búsqueda del cliente por su RUT.



Figura 15. Pantalla empresas con CI

Al navegar a las Vistas “visión global”, “productos” y “resumen cliente” también aparece la CI, donde en cuyo caso no debería visualizarse.

a) Vista visión global

Familia	Cantidad de Productos
> 01 - Cuentas Corrientes	2
01 - Cuentas Corrientes	1
01 - Cuentas Corrientes	1

Figura 16. Vista visión global con CI

En este caso se indica que existen 2 cuentas corrientes, y al hacer click en la Familia, se deriva a la pantalla “productos/servicios”, y es aquí donde la CI se visualiza.

Número de Producto/Servicio	Descripción	Familia	Estado	RUT Empresa	Razón Soci
> 1400726010	Cuenta Corriente Moneda Local	01 - Cuentas Corrier	Vigente o Activo	774310611	COPSA, A.S.
408727011	Cuenta Corriente Moneda Local	01 - Cuentas Corrier	Vigente o Activo	774310611	COPSA, A.S.

Cuentas - Datos Generales		Cuentas - Datos Admno	
*Cuentas C.C.I:	1400726010	Tasa:	
*Tipo Cuenta:	Cuenta Corriente Moneda Local	Valor Periodo Tas:	
Clase Cuenta:	CICVAN	Clase Restado Mon:	
Fecha Apertura:	10/12/2003	RUT Cliente:	
Estado Cuenta:	Vigente o Activo		
Fecha Cierre:			
Moneda:	Moneda Admision		

Figura 17. Pantalla productos/servicios con RUT empresa muestra CI

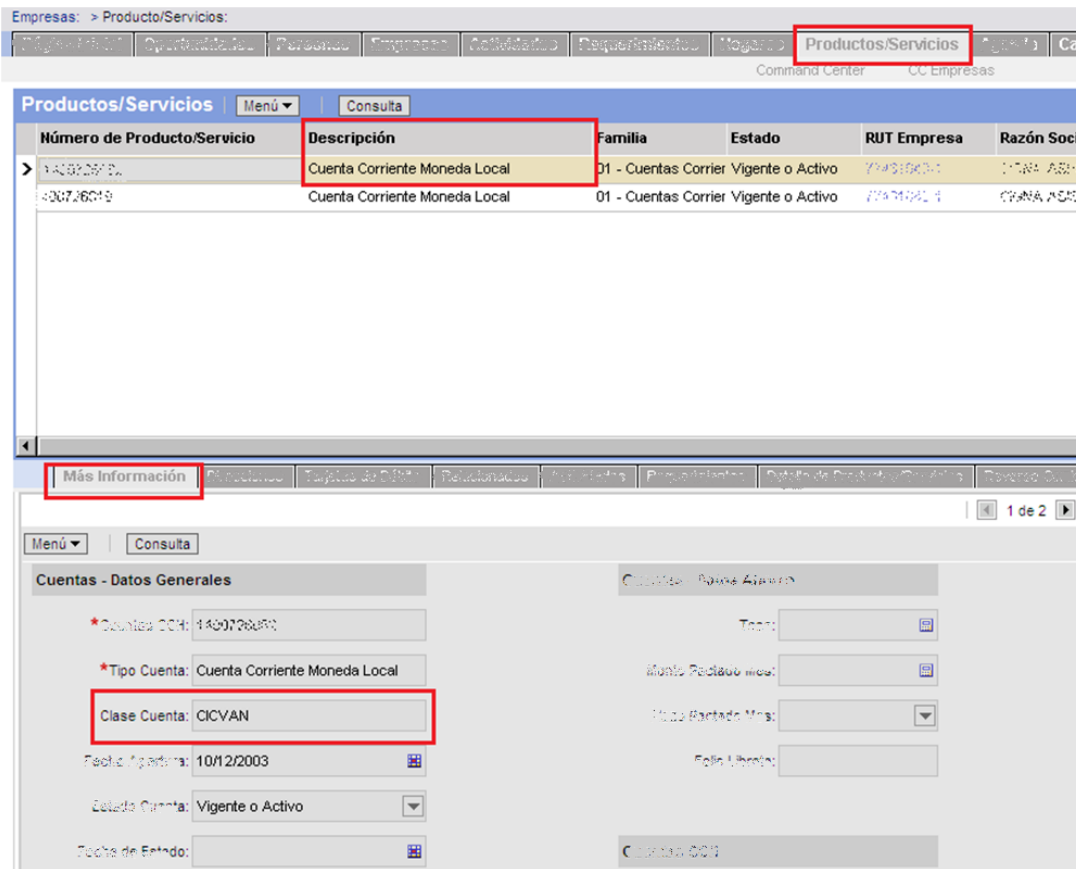
b) Vista productos



Familia	Cantidad de Productos
01 - Cuentas Corrientes	2

Figura 18. Vista productos con RUT empresa muestra CI

En este caso aparece en la sección “Familia”, donde al hacer click deriva a la pantalla “Productos/Servicios”, y es aquí donde la CI se visualiza.



Número de Producto/Servicio	Descripción	Familia	Estado	RUT Empresa	Razón Soci
1450128000	Cuenta Corriente Moneda Local	01 - Cuentas Corrier Vigente o Activo		779215001	CI - NA - ASIS
436736000	Cuenta Corriente Moneda Local	01 - Cuentas Corrier Vigente o Activo		779215001	CI - NA - ASIS

Más Información

Cuentas - Datos Generales

* Cuenta CCH: 1450128000

* Tipo Cuenta: Cuenta Corriente Moneda Local

Clase Cuenta: CICVAN

Fecha Agendar: 10/12/2003

Estado Cuenta: Vigente o Activo

Fecha de Estado:

Figura 19. Pantalla productos/servicios derivada del drill down de familia de productos

c) Vista resumen cliente

Numero Cuenta	Tipo Operacion
101010001	Av. A
101010002	Pagos
101010003	Regime
101010004	Canal de Remesas
101010002	Cuentas Corrientes
101010005	Canal de Remesas

Figura 20. Vista resumen cliente

En esta vista la CI aparece en la sección “otros productos”, en este caso debiera inhibirse.

Navegación a pantalla personas: búsqueda por Rut Cliente

RUT	Apellido Paterno	Apellido Materno	Nombres	RUT Cóny
101010002

Figura 21. Pantalla personas con CI

En este caso se muestran las cuentas al navegar a las vistas “Visión Global”, “Productos” y “Resumen Cliente”.

a) Vista visión global

Número de Producto/Servicio	Descripción	Estado	Bloqueo
136495207	Cuenta Corriente Moneda Local	Vigente o Activo	
01176430204	Línea De Crédito Personas	Vigente o Activo	
005197113158001919	Cuenta de Ahorro Entidad	Vigente o Activo	
013741315800	Línea De Crédito Personas	Vigente o Activo	
013741315800	Ahorro	Vigente o Activo	
013741315800	Cuenta Ahorro	Vigente o Activo	

Figura 22. Vista visión global de personas con CI en productos/servicio

b) Vista Producto

Número de Producto	Descripción	Estado	Bloqueo	Saldo Promedio Mensual/Saldo Insoluto	Utilizado Promedio Mes	Monto Autorizado/Monto Otorgado	Familia
136495207	Cuenta Corriente Moneda Local	Vigente o Activo					01 - Cuentas Corrientes
01176430204	Línea De Crédito Personas	Vigente o Activo				411.785 \$	04 - Línea de Crédito
005197113158001919	Cuenta de Ahorro Entidad	Vigente o Activo					07 - Cuenta Ahorro Entidad
013741315800	Línea De Crédito Personas	Vigente o Activo					05 - Línea de Crédito Personas
013741315800	Ahorro	Vigente o Activo					13 - Cuenta Ahorro
013741315800	Cuenta Ahorro	Vigente o Activo					13 - Cuenta Ahorro

Figura 23. Vista producto en personas con familia de CI

c) Vista resumen de cliente: En esta vista aparece en la sección “cuenta corriente”.

Nro. de Operación	Rut Deudor	Nombre Deudor	Clasificación	Deuda Total(M\$)
<p>AVALA A ACEPTA A AFIANZA A</p>				
Cuenta Corriente -				
Nro Cuenta		Saldo Disponible	Saldo Contable	
136495207		0,0	0,0	

Figura 24. Vista resumen cliente para personas

3.4.5.2. Lógica de visibilidad según las responsabilidades.

Los componentes con filtro se asignarán a las responsabilidades actuales que hoy en día pueden ver toda la información, esto ahorra la asignación manual de todas las responsabilidades existentes, limitando dicha asignación de los elementos copia a sólo una responsabilidad nueva que si tendrá visibilidad completa. La lógica que se empleará se explica a continuación.

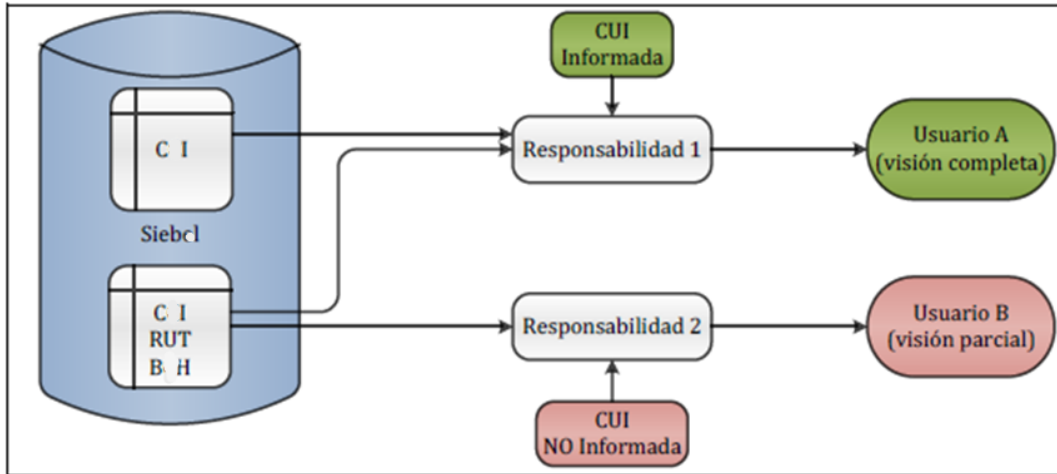


Figura 25. Lógica de asignación de responsabilidades

El diagrama presenta dos responsabilidades de ejemplo, donde se tiene:

- a. Responsabilidad 1: deberá ser una responsabilidad nueva, ya que será asociada a componentes nuevos de Siebel, creados a partir de los componentes originales. Los componentes asociados a la Responsabilidad 1 NO tendrán cambios, ya que deben permitir visualizar todas las cuentas internas.
- b. Responsabilidad 2: corresponde a la responsabilidad que filtra información. Estará asociada a los componentes originales, pero con filtro.

3.4.5.3. Lógica de filtrado.

La lógica a seguir para seleccionar cuando se debe presentar la información filtrada y sin acceso a las Cuentas internas, se basa en las siguientes condiciones:

a) Primero se evalúa si se trata de una responsabilidad informada o no, cuya determinación está fuera del alcance del proyecto. Ya que se tiene como dato de entrada, proporcionado por el Banco, la responsabilidad a la que se asocian las pantallas, vistas y applets filtrados y la responsabilidad a la que se asocia los objetos sin filtro como se encuentran actualmente.

Por lo que no hay ninguna operación dentro de este flujo de filtrado para evaluar las responsabilidades.

b) El siguiente parámetro es el RUT de las Cuentas Corrientes, ya que si se tiene un valor diferente al del Banco, no será una CI visible para todas las responsabilidades.

c) Por último, se evalúa la característica de las CI llamada Clase Cuenta, si es del tipo CIN (pago de nómina) o CIV (postergación de cuota), no serán visibles para las responsabilidades sin permisos de visualización de CI.

A continuación se muestra en un diagrama de flujo la lógica de filtrado, evaluando las condiciones necesarias en cada caso:

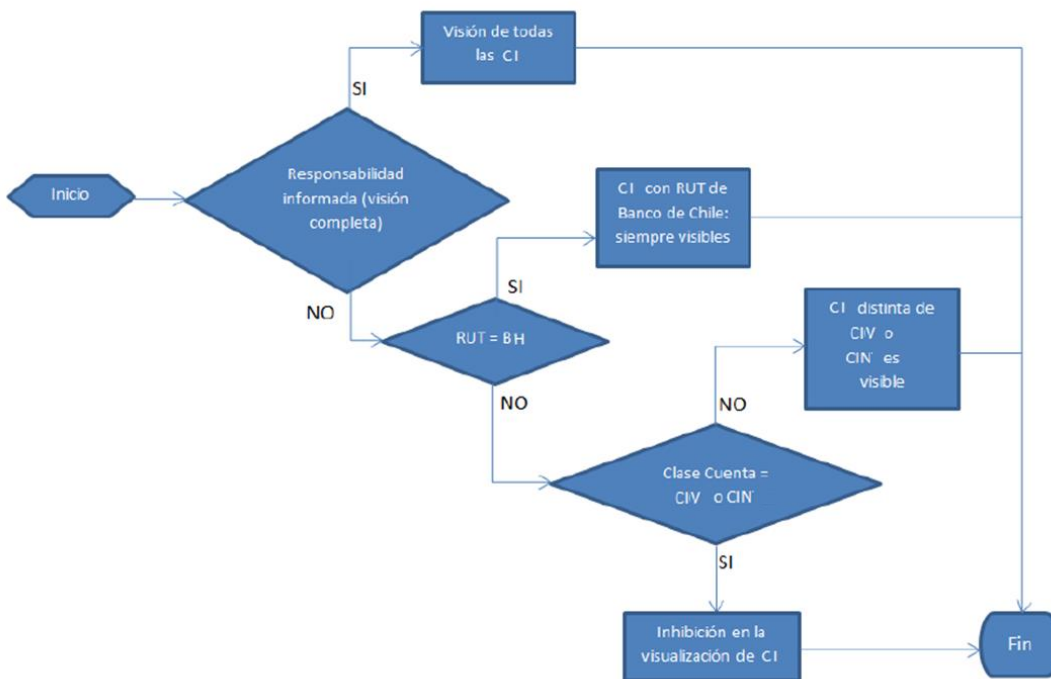


Figura 26. Flujo del filtro

3.4.6. Diseño técnico

Conforme a las buenas prácticas Siebel, existe una estrategia de configuración que indica que los cambios deben diseñarse comenzando por la capa de usuario, después por la capa de negocio y por último por la capa de datos. De esta forma, primero se puede diseñar toda la funcionalidad de la aplicación para satisfacer los requerimientos, y comenzar a descender de nivel tomando en cuenta todos los cambios necesarios en capa de negocio para poder ejecutar lo planteado en capa de interfaz y de la misma forma planear todo lo necesario en capa de datos para hacer posible la solución a nivel capa de negocio y capa de interfaz.

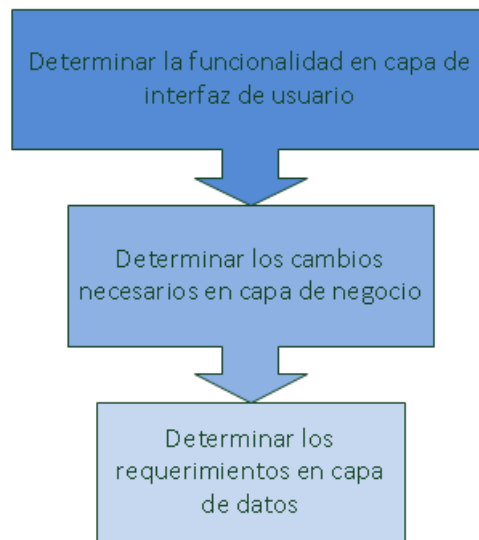


Figura 27. Orden en el diseño de la solución

Según esto, y acorde también con lo que se explicó en el marco teórico de este evolutivo, a continuación se explica la solución desglosada en dichos niveles.

3.4.6.1. Capa de interfaz de usuario

La información de las CI en Siebel se consulta y visualiza mediante las pantallas Productos/Servicios, Personas y Empresas. Mediante un filtro en los elementos de dichas pantallas es posible inhibir la visualización de información a las responsabilidades informadas.

Para el cumplimiento de este requerimiento y siguiendo la lógica de filtrado, se utilizarán campos calculados que evalúen si se tiene el RUT BH y el tipo de Clase Cuenta correspondiente para restringir la visualización, todo esto mediante una lista de valores para garantizar que no existe dependencia al valor a evaluar, sino que queda paramétrico para futuros cambios en dichos valores.

Es importante recordar que los filtros se montarán sobre los objetos originales, pero es necesario crear los objetos copia que mantengan la funcionalidad actual.

Descripción de pantallas, vistas y applets principales:

Se cambia el nombre real y técnico de las Vistas, Business Component y applets por privacidad de datos, de esta parte es importante ilustrar la estructura actual de la aplicación para ellos se utiliza una descripción funcional.

❖ Para la pantalla productos/servicios:

VISTA				
Nombre	Descripción	Título	Business Component	Applets
Service Products view	Vista principal de productos y servicios	Productos/ Servicios	CRP Account	PyS principal List Applet
				PyS principal Form Applet

Tabla 3. Análisis de elementos para productos/servicios

VISTA				
Nombre	Descripción	Título	Business Component	Applets
More info view	Vista correspondiente a "Más información"	Más información	CRP Account	PyS More Info List Applet
				PyS Current Account Form Applet

Tabla 4. Análisis de elementos para productos/servicios

VISTA				
Nombre	Descripción	Título	Business Component	Applets
Service Products Detail View	Vista correspondiente a "Detalle de Productos y servicios"	Detalle de Productos / Servicios	CRP Account	Account Agent Form Applet
			CRP Account Web	Account Web Applet

Tabla 5. Análisis de elementos para productos/servicios

En la pantalla anterior, existe un web applet que aunque es parte de la vista implicada, no será afectado por el filtro dado que los web applet sirven para incrustar contenido web, por lo que no consume información del Business Component.

❖ Para la pantalla empresas:

VISTA				
Nombre	Descripción	Título	Business Component	Applets
Company Summary View	Vista correspondiente a "Empresas" – "Visión Global"	Empresas	Account	Account Entry Applet
			CRP Account	Company Financial Accounts Summary List Applet

Tabla 6. Análisis de elementos para empresas

VISTA				
Nombre	Descripción	Título	Business Component	Applets
Company Financial Account View	Vista correspondiente a "Empresas" – "Productos"	Empresas	Account	Account Entry Applet
			CRP Account Header	Account Header List Applet

Tabla 7. Análisis de elementos para empresas

VISTA				
Nombre	Descripción	Título	Business Component	Applets
Summary Client Account View	Vista correspondiente a "Empresas" – "Resumen Cliente"	Resumen Cliente	Account	Account Entry Applet
			Web Applet BC	Account Resumen Web Applet

Tabla 8. Análisis de elementos para empresas

En esta última vista, también se tiene un web applet por lo que aplica el mismo caso mencionado en las vistas de productos/servicios.

❖ Para la pantalla personas:

VISTA				
Nombre	Descripción	Título	Business Component	Applets
Contact Summary View	Vista correspondiente a "Personas" – "Visión Global"	Persona	Contact	Contact Form Applet
			BC CRP Account	Contact Financial Accounts Summary List Applet

Tabla 9. Análisis de elementos para personas

VISTA				
Nombre	Descripción	Título	Business Component	Applets
Contact Account View	Vista correspondiente a "Personas" – "Productos"	Persona	Contact	Contact Form Applet
			BC CRP Account	Main Account List Applet

Tabla 10. Análisis de elementos para personas

VISTA				
Nombre	Descripción	Título	Business Component	Applets
Summary Client View	Vista correspondiente a "Personas" – "Resumen Cliente"	Persona	Contact	Contact Basic Form Applet
			Web Applet BC	Contact Resumen Web Applet

Tabla 11. Análisis de elementos para personas

3.4.6.2. Capa de negocio

El requerimiento solicitado por el Banco se resuelve con la modificación de objetos a nivel capa de negocio, principalmente en los Business Component que asocian la información correspondiente a personas, empresas y productos/servicio.

Los Business Component involucrados en el desarrollo y que actualmente tienen visión completa de CI son:

- CRP Account (productos/servicios)
- Account (empresas)
- Contact (personas)

Estos BC son los encargados de almacenar la información de los productos/servicios, Personas y Empresas. Sin embargo, siempre puede existir el caso en donde dentro de un mismo contexto de negocio, los applets consuman información de un BC distinto. Es decir, para la vista Empresas, no es un hecho que todos los applets que se encuentran ahí consuman información del BC de empresas, sino que pueden hacerlo de algún otro BC que contenga información que es útil también para empresas. Para este evolutivo se encontraron los siguientes BC en ese caso, debido a los toggle applets existentes:

- CRP Account (Light)
- CRP Account Header

Ya que se identificaron, el desarrollo implica, realizar una copia de cada uno de estos Business Component, ya que algunos mantendrán la funcionalidad original y otros tendrán el filtro aplicado, por lo que después de la implantación del evolutivo se tendría:

- CRP Account
- Copia de CRP Account
- Account
- Copia de Account
- Contact
- Copia de Contact
- CRP Account (Light)
- Copia de CRP Account (Light)
- CRP Account Header
- Copia de CRP Account Header

La lógica de desarrollo indica que sobre los componentes originales, se aplican los nuevos filtros y quedan asociados a las responsabilidades que actualmente existen.

Mientras que los componentes copia de FINCORP Account, copia de Account y copia de Contact, permanecen con la configuración actual y se asocian a una única responsabilidad nueva que no tiene restricción en la visualización de Cuentas internas.

A continuación se describen las características de los Business Component principales:

Business Component					
Nombre	Descripción	Tabla Asociada	No delete	No Insert	No update
CRP Account	Business Component para mapear datos de Productos y Servicios	S_ASSET	FALSE	FALSE	FALSE

Tabla 12. Análisis de los business component principales

Business Component					
Nombre	Descripción	Tabla Asociada	No delete	No Insert	No update
Contacts	Business Component para mapear datos de Personas	S_PARTY	FALSE	FALSE	FALSE

Tabla 13. Análisis de los business component principales

Business Component					
Nombre	Descripción	Tabla Asociada	No delete	No Insert	No update
Accounts	Business Component para mapear datos de Empresas	S_PARTY	FALSE	FALSE	FALSE

Tabla 14. Análisis de los business component principales

Para garantizar la funcionalidad de la solución que se planteó a nivel de Business Component se tiene que considerar que los Business Component dependen de los Business Object, por lo que es necesario cuidar esa relación para que los datos sean congruentes entre los objetos originales y los objetos copia. Esta relación implica la asignación de los Business Component que no existían en su correspondiente Business Object y la creación de links para relacionarlos.

3.4.6.3. Capa de datos

Para el cumplimiento de los requerimientos solicitados, no es necesario llevar a cabo modificaciones importantes a nivel capa de datos, la estructura de datos permanecerá como hasta ahora dentro de la base de datos. Ya que la solución estará basada en un filtro que únicamente limite la visibilidad de las cuentas corrientes, jamás se considera la eliminación de ningún registro.

De igual forma, la inclusión del campo “Clase cuenta” que se requiere para el filtrado, es resultado de la puesta en producción de otro proyecto. El alcance de este evolutivo considera la existencia de dicho campo.

Además los campos calculados que se requerirán, son configuraciones a nivel capa de negocio, ya que estos campos no se almacenan en la base de datos, su valor se calcula en tiempo de ejecución.

El único cambio que se considera a este nivel, es para el caso de uno de los Business Component que no cuenta con los campos Clase Cuenta y RUT BH, necesarios para ejecutar la lógica del filtro según las condiciones establecidas. Sin embargo, es responsable de mostrar Cuentas internas, por lo que debe llevar un filtro. La solución planteada será agregar dichos campos a este BC para poder usar la misma lógica de filtrado del resto de los Business Component.

Es importante recordar que esta adaptación deberá ser realizada al principio del evolutivo, para cuando se esté desarrollando a nivel de objetos de negocio, ya se tengan disponibles esos campos.

3.4.7. Construcción

3.4.7.1. Tareas previas al desarrollo

Por tratarse de un proyecto de tipo evolutivo y correctivo, no fue necesaria la instalación de la herramienta o de sus ambientes de desarrollo, ya que el banco ya contaba con ella dentro de su núcleo de negocio. Sin embargo, si fueron necesarios adecuar el ambiente de desarrollo y pruebas para cada uno de los participantes en el proyecto.

La tecnología con que cuenta el banco para su CRM se describe a continuación:

- a)** La versión de Siebel es la 7.7.
- b)** Para la conexión con la base de datos se tiene la versión Oracle 9i.

Para el desarrollo el banco proporcionó una máquina virtual con la instalación completa del Siebel Client y el Siebel Tools, sin embargo, para cada usuario desarrollador, es necesario generar un extracto local de la base de datos que permita realizar configuraciones y pruebas en el servidor de desarrollo. Para lo cual se requieren las siguientes actividades:

- a) Solicitar al DBA o encargado, la creación del usuario nuevo directamente en base de datos.
- b) Crear una cuenta en Siebel con el mismo ID de base de datos y asignarle una responsabilidad. También es necesario crear el mismo usuario en los clientes móviles dentro de la aplicación.
- c) Conectar a Siebel Server, indicando en ID de usuario y Contraseña el usuario creado más arriba.
- d) Ejecutar tarea para extraer el File System y la Base de Datos del servidor y usarla localmente en Siebel Tools.

En el Web Client, ejecutar primero un trabajo para crear la base de datos con los siguientes parámetros:

- Componente/Trabajo: Generate New Database
- SQL Anywhere Database: sse_utf8.dbf

Una vez terminado exitosamente dicho trabajo es necesario crear uno nuevo con los siguientes parámetros:

- Componente/trabajo = Database Extract
- Client Name = Id de usuario creado
- Extract All Repository = True

Se ejecuta el trabajo y al finalizar el extract es posible ingresar a Siebel tools y Siebel Web Client con el usuario creado, al momento de iniciar sesión se instalará el extract generado, se importarán todas las tablas y será posible comenzar a desarrollar con base de datos local.

Una vez en Siebel Tools es necesario descargar los proyectos de del servidor al Siebel Tools que se van a trabajar en el ambiente local, para ello haremos un proceso de Check Out, lo que implica que podremos modificar localmente los objetos necesarios, mientras los mantenemos bloqueados en el servidor para que nadie más pueda modificarlos, así se evita la discrepancia de la información cuando se trabaja en equipo sobre los mismos objetos.

El objetivo de todas estas tareas es que para comenzar a desarrollar la solución se debe tener una estación de trabajo como se explica en la imagen:

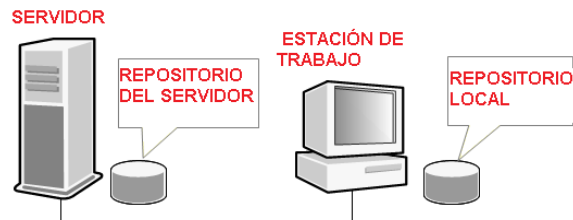


Figura 28. Estación de trabajo

3.4.7.2. Entregables de la solución

Una vez que se tiene listo todo el ambiente de desarrollo: la base de datos local y los proyectos bloqueados en el servidor y descargados en Siebel Tools, se puede iniciar la construcción de la solución. Para ello es necesario tomar en cuenta no sólo los requisitos de negocio sino las actividades y los alcances propios de cada ambiente, para generar en tiempo y forma todo lo necesario para lograr la implementación de la solución en el ambiente productivo.

La siguiente tabla, explica los ambientes por los que nuestro evolutivo pasará hasta ser llevado al ambiente productivo.

Ambiente	Descripción
DESARROLLO	Ambiente inicial en el que se modifica el código fuente, se realiza la compilación y las pruebas unitarias e integradas.
QA	Ambiente en el que se realizan las pruebas de certificación por parte de los usuarios. Considera pruebas de stress, inspección de código para asegurar que la versión está completamente preparada para salir a producción. Este ambiente debe ser idéntico al entorno de producción.
PRODUCCION	Ambiente al que tienen acceso los usuarios para hacer uso de la aplicación.

Tabla 15. Ambientes por los que pasará el evolutivo

El siguiente diagrama muestra el proceso de desarrollo, la interacción entre los servidores de desarrollo y QA y nos indica lo que se debe generar durante la construcción en el servidor de desarrollo.

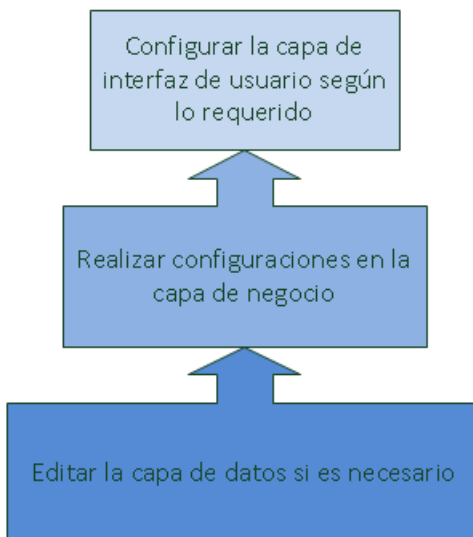


Figura 30. Orden de construcción del evolutivo.

Por ello, a partir de ahora la solución estará dividida por requerimiento y dentro de éste, se especificarán las capas a las que afecta, respetando el orden anterior. Los requerimientos serán presentados en el orden en que tuvieron que ser desarrollados.

Requerimiento: El filtro debe ser independiente al valor de BH y Clase Cuenta. (Valores paramétricos)

A. Capa de datos

No aplica

B. Capa de negocio

El primer requerimiento considerado, es el hecho de que el filtro de información no dependa del valor que tomen los campos involucrados, sino que sea independiente y que sea posible modificar el valor sin afectar la el funcionamiento del filtro o que esto lleve a un desarrollo correctivo posterior. Entonces, los valores a considerar en el filtro para los campos RUT BH y Clase cuenta, serán administrados desde una lista de valores.

En Siebel, una lista de valores soporta las listas estáticas que se muestran en el Siebel Web Client. En el caso específico de este evolutivo, no se mostrarán los valores de esta lista de valores, simplemente servirá para consultar los valores que hay en ella y utilizarlos como valores a tomar en cuenta para el filtrado de la información. De esta forma, un usuario de Siebel con permisos de administrador, podría modificar esta lista de valores para cambiar el valor de alguno de los campos

considerados, y esto no afectaría el filtro, que simplemente rescataría el valor que se encuentre en dicha lista para funcionar.

Las listas de valores se consideran objetos de capa de negocio, porque a pesar de haber sido configurada a nivel aplicación y no en Siebel Tools queda almacenada en un Business Component nativo de la herramienta, el llamado ‘List of Values’.

La lista de valores implementada fue la siguiente:

Proyecto	Fecha	Tipo	Valor visible	Código que no depende del idioma	Parent LIC	Activo	Descripción	Idioma
Inhibición CI	08/10/2013	INHIBICION_CI	CLASE CUENTA	CIV	inhibición	Y	Se agregan nuevos valores a LOV para el filtro de inhibición CI	Español (moderno)
Inhibición CI	08/10/2013	INHIBICION_CI	CLASE CUENTA	CIN	inhibición	Y		Español (moderno)
Inhibición CI	08/10/2013	INHIBICION_CI	RUT	972840980-5	inhibición	Y		Español (moderno)
Inhibición CI	08/10/2013	LOV_TYPE	INHIBICION_CI	Inhibición	inhibición	Y		Español (moderno)

Tabla 16. Lista de valores implementada para el filtro.

De dicha lista, el filtro consultará el campo “valor visible” y hará las comparaciones precisas con lo que está en “Código que no depende del idioma”, por lo que el filtro no estará en función de lo que contiene ese campo y si en el futuro los valores de clase cuenta o RUT cambian, no requiere ningún desarrollo extra, simplemente actualizar esta lista de valores y el filtro seguirá funcionando de la misma forma.

C. Capa de interfaz de usuario

No aplica

Requerimiento: Las Cuentas internas que cumplen con las condiciones de filtrado no deben ser visibles.

A. Capa de datos

Como se mencionó en el diseño técnico, existe un Business Component con una condición particular, el CRP Account (Light), que no permite utilizar la misma estrategia de filtrado, ya que no cuenta con los campos Clase Cuenta y RUT BH, necesarios para ejecutar la lógica del filtro según las condiciones establecidas. Sin embargo, es responsable de mostrar Cuentas internas, por lo que se tiene que tener

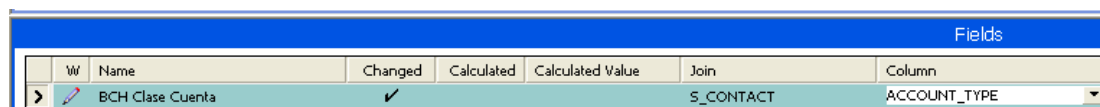
alguna estrategia que permita aplicar el filtro también pues se vuelve más práctico que diseñar una estrategia única para este BC.

Para lograrlo, se planea utilizar tres campos en base de datos que no estén siendo ocupados, la forma de saberlo fue haciendo uso de un manejador externo de base de datos, conectar la base de datos de Siebel, y mediante una serie de consultas SQL, determinar 3 campos existentes pero vacíos en la misma tabla que alimenta dicho BC.

Una vez que se tienen estos campos, serán utilizados para que contengan la información de Clase Cuenta y el RUT BH para persona y empresa.

Esta adaptación también implica una operación JOIN con la tabla de contactos que contiene los valores de Clase Cuenta y RUT. Esta operación combinará los campos deseados que se encuentran en tablas distintas. Este JOIN se hace en Siebel Tools, ya no es necesario utilizar ningún manejador de base de datos extra.

La modificación dentro del Business Component para cada campo quedaría de la siguiente forma:



W	Name	Changed	Calculated	Calculated Value	Join	Column
>	BCH Clase Cuenta	✓			S_CONTACT	ACCOUNT_TYPE

Figura 31. Campos para join faltante

Donde se especifica el nombre del campo libre encontrado y se hace el join con la tabla correcta y la columna que contiene la información que necesitamos.

Con estos campos, es posible continuar con la implementación del filtro que se explica en el siguiente punto.

B. Capa de negocio

Los campos que pueden verse en la aplicación y que se muestran en un applet, pertenecen a un Business Component.

Los Business Components son las entidades encargadas de relacionar y agrupar todos los campos que se encuentran en una o varias tablas dentro de la base de datos y llevar estos datos a un objeto de negocio.

Por lo que si se requiere un filtro de ciertos registros mostrados en el applet, éste tendría que colocarse a nivel Business Component y con ello se aplica a los applets que consuman campos pertenecientes a dicho BC. Recordando que los applets únicamente organizan los registros para mostrarse en la interfaz de usuario y aunque también pueden aplicarse filtros a este nivel, en este caso hacerlo a nivel de BC implica mayor facilidad de implementación y dado que el CRM del banco es un

sistema en constante movimiento se garantiza la simplicidad para futuras adaptaciones.

Además se pidió en los requerimientos que las CI se inhibieran en todos los lugares donde se puede obtener información de ella, así que el hecho de que aplicando el filtro en BC afecte todos los applets que dependen de él nos proporciona esa funcionalidad.

Pero antes de montar el filtro y como se explicó en la etapa de diseño de este proyecto, se requieren crear copias de los componentes existentes. En Siebel, la copia de los elementos se realiza en cascada, es decir que se copian también todos los objetos que dependan de dicho componente que se copiará. Lo cual es muy conveniente para el desarrollo, ya que se requiere conservar toda la funcionalidad actual y sólo agregar los requerimientos propios de este evolutivo. Por lo que se garantiza que todos los componentes actuales que no están involucrados en el filtro seguirán idénticos en los elementos copia.

Otra consideración que hay que tener en cuenta al momento de copiar elementos es la jerarquía de estos a nivel interfaz de usuario, sobre todo de los applets, ya que existen applets que dependen de otros applets, estos se llaman toggle applets, y se despliegan desde el applet padre, según algunas condiciones. Por ejemplo, si existiera un formulario para llenar datos sobre una persona, y en el applet padre se pudiera elegir entre una persona física o moral, se puede implementar un toggle applet que se despliegue con los campos requeridos si se eligió persona física o un toggle applet con los campos propios de la persona moral, siendo estos applets independientes entre sí. Tomando en cuenta esto, y haciendo un análisis de la jerarquía de elementos que muestran la información que se debe filtrar existen toggle applets que se deben duplicar por separado de sus applets padre, ya que si sólo se duplican los padres, como ya se mencionó se duplicaran los hijos (por copia en cascada) pero lo hijos a diferencia de los applets padre seguirán consumiendo información del BC original, es decir se presentará una incongruencia entre la información que recibe el applet padre con respecto a la de los toggle applets.

Para evitar esto, una vez copiados los Business Component, los applet padre y los toggle applets, se debe hacer una asociación manual, entre los BC copia con los applet copia y estos a su vez con los toggle applets copia. Para simplificar el entendimiento de este punto y por tratarse de un aspecto vital en la solución, se explica con el siguiente diagrama:

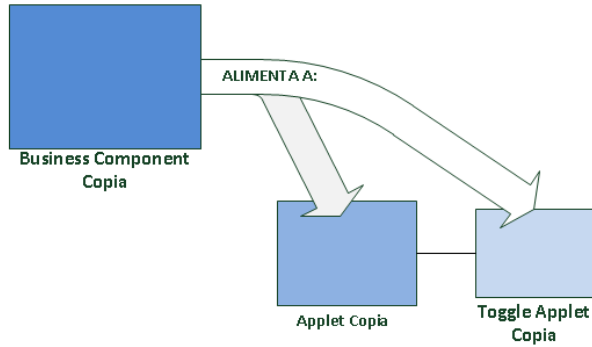


Figura 32. Relación de business component con applets y toggle applets.

Después del copiado a los elementos copia de los agregó la palabra CI en el nombre para distinguirlos de los elementos originales recordando que son los elementos originales los que llevarán el filtro. A continuación un conteo de los elementos originales involucrados en todo el desarrollo:

Componentes que deben copiarse:

- ✓ Business Components: 5
- ✓ Applets: 11
- ✓ Toggle applet: 23

Componentes que no deben copiarse:

- ✓ Business Object: 3
- ✓ Aplicación: 1
- ✓ Pantallas: 3
- ✓ Vistas: 9

Los elementos que no van a copiarse son objetos superiores en la jerarquía Siebel, que únicamente requieren que se hagan referencias correctas a los objetos originales y copiados que contendrán, para asegurar la congruencia de la información que proporcionan a dichos objetos hijos.

En esta capa de negocio, uno de los pasos vitales es la referencia de los Business Objects con los nuevos Business Component copiados.

Las vistas referencian a los Business Object, y aunque cada applet que está dentro de la vista obtiene sus datos de un BC, es obligatorio que dichos Business Component estén referenciados al Business Object que contiene la vista donde se quiere mostrar los Applets.

Para ello, el primer paso es agregar cada BC copia con el BO al que su objeto original pertenecía.

En Siebel Tools, dentro del Business Object se agrega un nuevo registro de Business Object Component con el nombre del BC a utilizar, como se muestra en la siguiente pantalla:

The screenshot shows two tables in Siebel Tools. The top table, 'Business Objects', lists components like AICC SCORM Import, AMS Ref - Activity, and AMS Ref - Branch. The bottom table, 'Business Object Components', lists components like AMS Branch, AMS Service Center, and Internal Product, with links between them.

W	Name	Changed	Project	Primary Business Component	Query List Business Component
	AICC SCORM Import		eTraining Lessons	Training Course Lessons	Query List
	AMS Ref - Activity		FINS AMS Integrator	AMS Activity Def	Query List
	AMS Ref - Area Code		FINS AMS Integrator	AMS Area Code	Query List
	AMS Ref - Branch		FINS AMS Integrator	AMS Service Center	Query List

W	Bus Comp	Changed	Inactive	Link	Comments
	AMS Branch			AMS Service Center/AMS Branch	
	AMS Service Center				
	FIN Product Rates			Internal Product/FIN Product Rates	
	Internal Product			AMS Branch/Internal Product	

Figura 33. Relacionar business component con business objects

También se tiene que crear un link entre el BO/BC utilizando como campo fuente y destino el Row Id, es decir, dejando esos campos en blanco para que Siebel automáticamente tome el Row Id.

En la siguiente imagen se ilustra cómo deben quedar los links que se utilizarán para ligar el Business Object "Contact" con un Business Component original y su copia.

The screenshot shows the 'Links' table in Siebel Tools, detailing the relationship between a parent Business Component and a child Business Component.

W	Name	Changed	Project	Parent Business Component	Child Business Component	Source Field
	Contact/BCH FINCORP Account (Light) CCI	✓	BCH FINS Portfolio	Contact	BCH FINCORP Account (Light) CCI	
	Contact/BCH FINCORP Account (Light)	✓	Contact	Contact	BCH FINCORP Account (Light)	

Figura 34. Links para relacionar BO y BC

Una vez referenciados correctamente todos los objetos de negocio, el primer paso en la configuración del filtro como tal, consiste en la creación de una serie de campos calculados dentro de cada uno de los Business Component involucrados en la inhibición.

Un campo calculado, es un campo que pertenece a un Business Component, pero no se almacena en la base de datos. Obtiene su valor de otros campos del mismo BC o de algún BC padre, o del resultado de alguna operación, función o expresión que se ejecutará en tiempo real. No existe ninguna revisión sintáctica para los campos calculados, por lo que su éxito depende completamente de la habilidad del configurador.

En la solución se utilizarán campos calculados para evaluar las condiciones de filtrado, comparando los campos de Clase Cuenta y RUT empresa/persona con la lista de valores.

Los campos calculados por cada Business Component fueron creados de la siguiente forma:

The screenshot shows two tables from a software interface. The top table, titled 'Business Components', has columns: W, Name, Changed, Project, Cache Data, Class, Data Source, Dirty Reads, Distinct, and Enclosure ID Field. It contains one row: CRP Account, with 'Changed' checked, 'Project' as 'FINS Portfolio', and 'Class' as 'CSSB:Case'. The bottom table, titled 'Fields', has columns: W, Name, Changed, Calculated, Calculated Value, Join, Column, and Currency Code. It contains three rows of calculated fields:

W	Name	Changed	Calculated	Calculated Value	Join	Column	Currency Code
>	BH Clase Cuenta CI	✓	✓	IIF(Count(BH Inhibicion CI)=0, 'N', 'Y')			
	BH RUT empresa CI	✓	✓	IIF(Count(BH Inhibicion RUT)=0, 'N', 'Y')			
	BH RUT persona CI	✓	✓	IIF(Count(BH Inhibicion RUT P)=0, 'N', 'Y')			

Figura 35. Campos calculados para filtro

Como se aprecia en la imagen, los campos calculados tienen el siguiente valor calculado:

IIF(Count(BH Inhibicion CI)=0, 'N' , 'Y')

Donde, IIF es una función condicional que evalúa el resultado de la función Count(BH Inhibicion CI). La función Count tiene como argumento un Multi Value Link (MVL).

Un Multi Value Link, por jerarquía pertenece al Business Component y es una unión para establecer una relación 1:M entre un Business Component Padre y un Business Component Hijo por medio de un Link para conectar ambos Business Components. Como resultado de esta relación se obtiene un Multi Value Field que permite al Business Component Padre hacer uso de información del Business Component Hijo como si fuera propia. Un Multi Value Link se tiene que definir sobre el Business Component Padre.

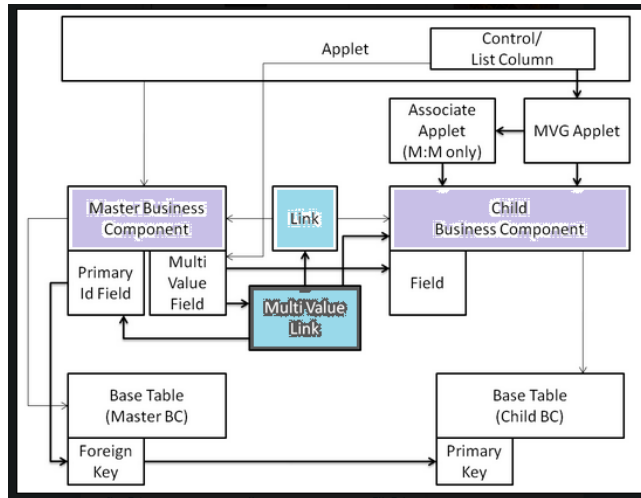


Figura 36. Multi Value Link

Para el desarrollo de nuestra solución, esta funcionalidad nos permitirá hacer comparaciones entre los campos de clase cuenta y RUT propios de Empresas, Personas y Productos / Servicios, con los valores proporcionados para cada caso por el banco y que activan el filtro. Recordando que dichos valores fueron implementados en una lista de valores para darle al filtro autonomía con respecto al valor especificado en cada caso.

Es importante recordar que para la lista de valores creada, el Business Component 'List of Values' será el que se debe utilizar en los Links y Multi Value Links.

Con esto, quedan identificados los Business Components involucrados y la relación que se tiene que establecer a nivel Multi Value Link, se explica a continuación en función de los parámetros del objeto en Siebel Tools. Un MVL tiene que tener al menos:

- ✓ **Nombre:** El nombre con que identificaremos al MVL, en el evolutivo es el que aparece dentro de la función Count en el campo calculado.
- ✓ **Business Component Destino:** es el Business Component hijo que se relacionará. Para el filtro siempre será: "List of Values"
- ✓ **Link Destino:** Indica el nombre del Link que relaciona específicamente los BC por sus campos. Se crea como un objeto aparte, fuera del BC en la jerarquía de Siebel Tools.

Los links por cada Multi Value Link quedan como en la siguiente imagen:

Links							
W	Name	Changed	Project	Parent Business Component	Child Business Component	Source Field	Destination Field
>	CRP Account/List Of Values	✓	FINS Portfolio	CRP_Account	List Of Values	BH Folio Libreta	Name
	CRP Account/List Of Values RUT	✓	FINS Portfolio	CRP_Account	List Of Values	BH Account RUT	Name
	CRP Account/List Of Values RUT P	✓	FINS Portfolio	CRP_Account	List Of Values	BH Person UID	Name

Figura 37. Links con las listas de valores

En el nombre del link, siempre debe ser el BC padre / BC hijo, para este evolutivo tuve que agregar un distintivo al final del nombre, como se muestra en la imagen, para distinguir el link de clase cuenta de los link de RUT persona y empresa, ya que Siebel no permite que dos objetos se llamen de la misma forma.

También se especifica el BC padre en el campo que se muestra como: Parent Business Component, éste será el BC que contendrá el MVL que manda llamar este link.

Lo más importante del link es la unión entre los campos, ya que así se especifica cual será el campo fuente y el campo destino, que en el contexto de nuestro filtro, indicamos qué campos se comparan en la función Count. Según la imagen: será el campo "BH Folio Libreta" contra el campo "Name" de la lista de valores.

Para ilustrarse un poco mejor, el campo BH Folio Libreta contiene el identificador de la clase cuenta y éste va a ligarse con el Name de la lista de valores que se implementó en el punto anterior, es decir, cuando dicho MVL se aplique a la función Count, esta función buscará las coincidencias del contenido del campo en el BC padre, con el contenido del campo relacionado en el BC hijo.

Entonces, una vez que se entiende cómo un MVL y su link relacionan dos campos de dos BC distintos, es posible entender que la función IIF devuelve el valor que quedará en nuestro campo calculado según la siguiente relación:

'N' cuando la función Count da como resultado un 0, es decir que no se encontraron coincidencias entre los campos comparados. En contexto de negocio, se entiende que los campos de Clase cuenta o RUT según el caso, no contienen ninguno de los valores necesarios para filtrar la información.

'Y' si el resultado de la función Count es mayor a 0, es decir que se encontraron coincidencias entre los campos comparados. En contexto de negocio significa que los campos de Clase Cuenta o RUT según el caso, contienen los valores necesarios para requerir del filtro de información.

Como se observa en la imagen donde se muestran los campos calculados del Business Component, existe uno por cada criterio para determinar si aplica el filtro o no. Es decir, un campo calculado que evalúa que el campo Clase Cuenta contenga el valor indicado por el banco para filtrar, uno para evaluar que se tenga el Rut del banco en el campo Rut Persona y uno para evaluar que se tenga el Rut del banco en el campo Rut empresa. Esto debido a que se requiere un multi value link y un link por cada relación entre los campos del Business Component padre e hijo.

Una vez que se efectúan las funciones antes especificadas para cada Persona, Empresa o Producto/Servicio que se vaya a mostrar en pantalla y el campo calculado de su Business Component correspondiente contiene 'N' o 'Y', se puede decir que ya se sabe qué condiciones se tienen para cada caso y con ellas determinar si se van a filtrar las Cuentas internas o no.

Como ya se explicó, el filtro será puesto a nivel de Business Component, para ello se modificará una propiedad de dicho objeto llamada 'Search Specification' o especificación de búsqueda, que consiste en una serie de condiciones que deben cumplirse para mostrar o no un registro. De acuerdo a la arquitectura de la aplicación Siebel, por cada registro que se vaya a mostrar extraído desde el BC se ejecutará la especificación de búsqueda para determinar si se muestra o no.

La especificación de búsqueda se compone de sentencias sencillas con operadores lógicos, cuyo resultado final es Verdadero o Falso.

Si es Verdadero, entonces el registro evaluado será mostrado. Mientras que si el resultado es Falso, entonces el registro evaluado no será mostrado. Por lo que la lógica de nuestra expresión debe concluir en Falso cuando se cumplan las condiciones de filtrado.

El filtro montado queda como se muestra a continuación:

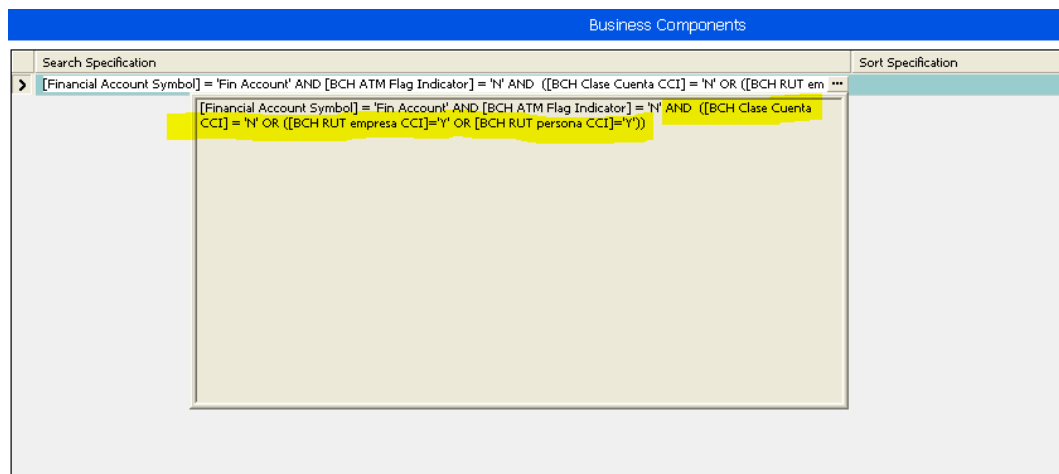


Figura 38. Filtro en las especificaciones de búsqueda del BC

En todos los Business Component involucrados, ya existía una especificación de búsqueda previa, por lo que sólo se va a agregar con un operador lógico AND la especificación correspondiente al filtro de las Cuentas internas. Utilizando el operador AND se respeta el cumplimiento de la especificación de búsqueda existente, y se agregan las condiciones nuevas, subrayadas arriba:

AND ([BH Clase Cuenta CI] = 'N' OR ([BH RUT empresa CI] = 'Y' OR [BH RUT persona] = 'Y'))

Esta sentencia cubre todas las combinaciones que pueden presentarse con los 3 criterios a evaluar. Como se demuestra en la siguiente tabla (prestar atención a la existencia de signos de agrupación):

[BH Clase Cuenta CI] = 'N'	OR	([BH RUT empresa CI] = 'Y'	OR	[BH RUT persona] = 'Y')	Resultado
F	OR	(F	OR	F)	F
V	OR	(F	OR	F)	V
F	OR	(V	OR	F)	V
V	OR	(V	OR	F)	V
F	OR	(F	OR	V)	V
V	OR	(F	OR	V)	V
F	OR	(V	OR	V)	V
V	OR	(V	OR	V)	V

Tabla 17. Evaluación del filtro.

Donde se observa que la única combinación que inhibe la visualización, es decir que da como resultado Falso es cuando:

- BH Clase Cuenta CI = 'Y' lo que significa que la Clase Cuenta del registro es alguna de las proporcionadas por el banco para inhibir las Cuentas Corrientes, lo cual es la primera condición de filtrado. Recordando que esto se obtuvo de la función Count que comparó la lista de valores válidos, con el valor del campo.
- BH RUT empresa CI o BH RUT persona CI = 'N' lo que significa que ni en RUT empresa ni en RUT persona se tiene el RUT del banco, segunda condición que debía cumplirse para filtrar las cuentas corrientes.

Por último, se cumplen ambas condiciones de filtrado, que de acuerdo con la lógica de la expresión se tiene $F \text{ OR } F = F$ y eso en la especificación de búsqueda indica que el registro consultado debe inhibirse para las responsabilidades no informadas

Requerimiento: Las cuentas internas que cumplan las condiciones de filtrado deben ser inhibidas para las responsabilidades no informadas.

Una de las principales cualidades de un CRM es la restricción de información según el rol de los empleados dentro de las empresas, esto permite eficiencia en su labor y privacidad en ciertos datos sensibles y por ende se mejora indirectamente el servicio al cliente. Sin embargo, no todas las herramientas lo hacen de la misma forma ni permiten la misma flexibilidad. En el caso de Siebel, es posible restringir datos o vistas (disposición de un conjunto de datos) de forma independiente.

De acuerdo a la solución que se construyó, la nueva funcionalidad (el filtro) se encuentra a nivel de Business Component, por lo que esto afecta directamente a los applets que consumen información de ellos. Y por jerarquía de Siebel a nivel capa de interfaz de usuario, los applets se agrupan en vistas, es por eso que para asignar la nueva funcionalidad a las personas indicadas dentro de la compañía, únicamente es necesario hacer la restricción a nivel de vistas.

Para ello se utilizan las responsabilidades, que son colecciones de vistas permitidas a un rol real en la empresa. Una responsabilidad no representa un rol de trabajo de la empresa, ya que una misma responsabilidad podría estar asignada a muchos puestos de trabajo distintos o al revés, un puesto de trabajo puede tener asignadas muchas responsabilidades. Únicamente es la forma en que se agrupan los permisos de un usuario. Y la relación entre vistas, responsabilidades y usuarios es muchos a muchos:



Figura 39. Relación M:M entre vistas, responsabilidades y usuarios

Los usuarios con múltiples responsabilidades ven la unión de los datos de sus vistas.

Los usuarios sin responsabilidades no pueden ver nada.

Las vistas sin responsabilidades asignadas no cumplen ninguna función dentro de la aplicación, se consideran basura y deben depurarse.

La asignación de responsabilidades a una vista se lleva a cabo dentro de la aplicación de Siebel, no en Siebel Tools, ya que es una tarea del administrador de la herramienta, no de un encargado de la configuración. Por ello, la asignación se reporta como una carga manual que debe ser ejecutada nuevamente por el administrador de cada ambiente al que pase el evolutivo (desarrollo, QA, producción).

Como se especificó durante los requerimientos de este evolutivo, no está dentro del alcance la determinación de las responsabilidades que verán o no las vistas con filtro. El banco proporciona el grupo A de responsabilidades que pueden ver la información sin filtro y el grupo B de responsabilidades que no pueden ver toda la información.

Sabiendo esto, el administrador de la aplicación debe asignar a cada usuario la responsabilidad correspondiente, como se muestra en la siguiente pantalla:

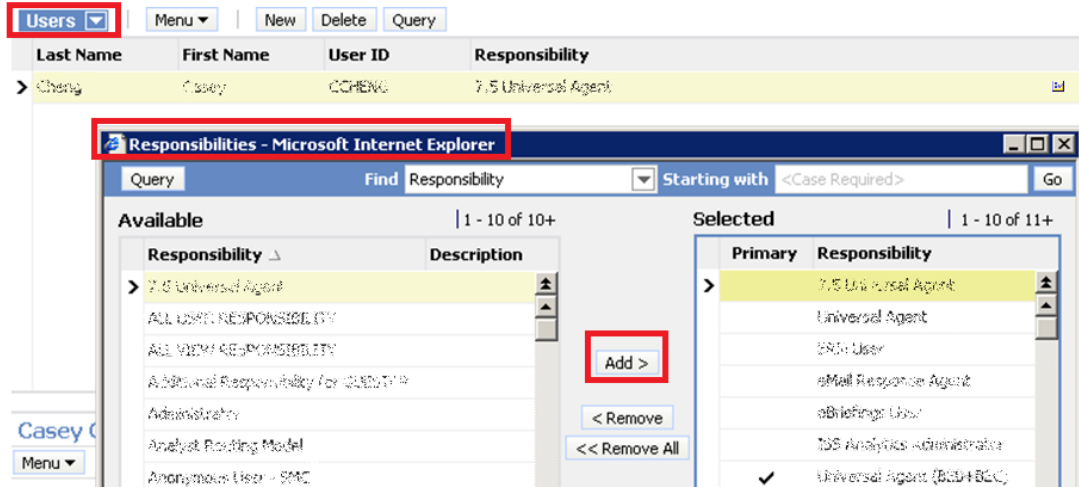


Figura 40. Asignación de responsabilidades a usuarios

Después se asignan todas las vistas que podrá ver cada responsabilidad

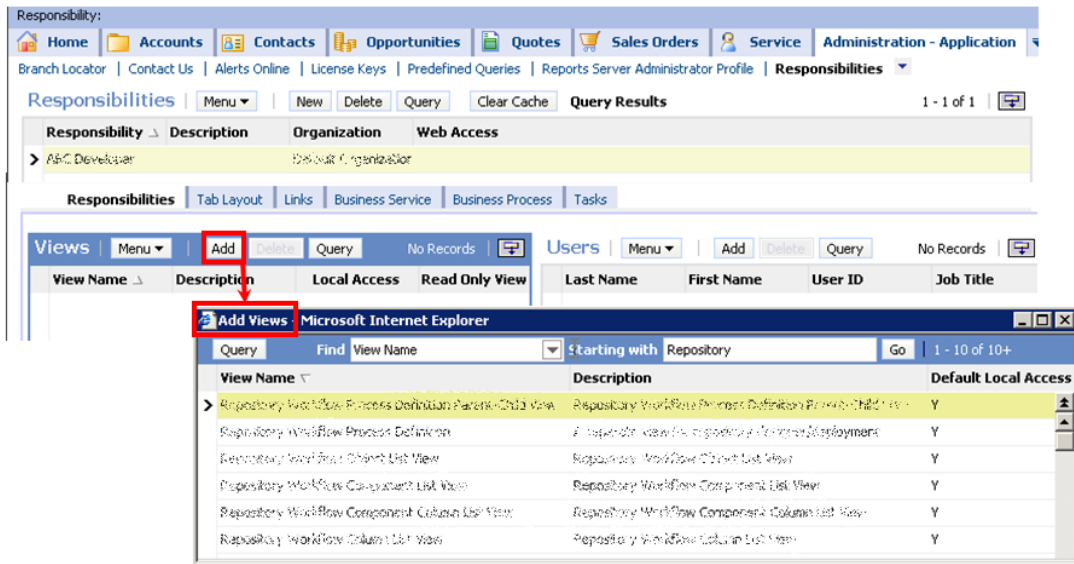


Figura 41. Asignación de responsabilidades a las vistas.

Para ello, se determinó que las vistas originales que contienen el filtro se asignarán a las responsabilidades que no pueden ver las Cuentas internas, y que las vistas copia se asignarán a las responsabilidades con visibilidad total sobre dicha información.

Es sumamente importante que esta asociación entre usuarios, responsabilidades y vistas se haga de forma correcta en todos los ambientes de Siebel, ya que esto

podría generar incidencias que no tendrían nada que ver con el desempeño de la herramienta sino con la administración de la misma, permitiendo ver información sensible a personas sin esa facultad o viceversa. Lo cual va en contra de las necesidades del banco y anula todos los objetivos de este evolutivo.

A continuación y según la jerarquía de objetos a nivel capa de negocio de Siebel, es necesario asignar las vistas a la pantalla (Screen).

No es necesario tener copias de las pantallas (Screen) que actualmente funcionan, ya que este objeto sólo funciona como contenedor de vistas, y son las responsabilidades las que determinan que usuario puede verlas. Por lo que se asignarán todas las vistas a las pantallas (Screen) correspondientes para personas, empresas y productos/servicios sin importar si son vistas con información restringida o no.

La asignación de vistas a las pantallas si es una tarea propia de los configuradores Siebel, ya que se realiza desde Siebel Tools, lo cual significa que después de asignarles todas sus vistas, las pantallas (Screen) deben compilarse y exportarse en un archivo .SIF para llevarse a otros ambientes.

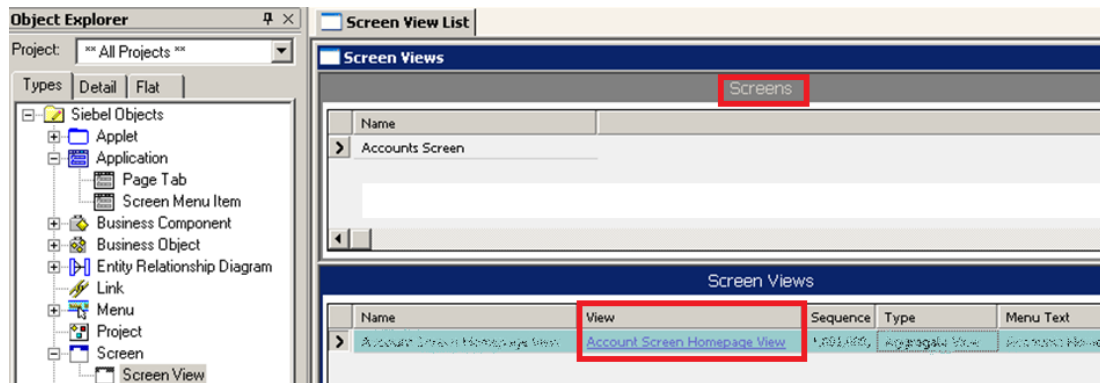


Figura 42. Relación de las vistas a las pantallas.

Y por último, siguiendo con la jerarquía de objetos de interfaz de usuario se tiene que recompilar la aplicación, que contiene las pantallas (Screen) que recién fueron modificadas, para que los cambios sean visibles. La aplicación es el objeto padre de todos los objetos de interfaz de usuario. Por lo que en el evolutivo sólo se involucra una aplicación que incluye el total de las funcionalidades del CRM del banco.

Las Pantallas se hacen visibles en la aplicación agregándolas en el Screen Menu Item como se aprecia en la ilustración:

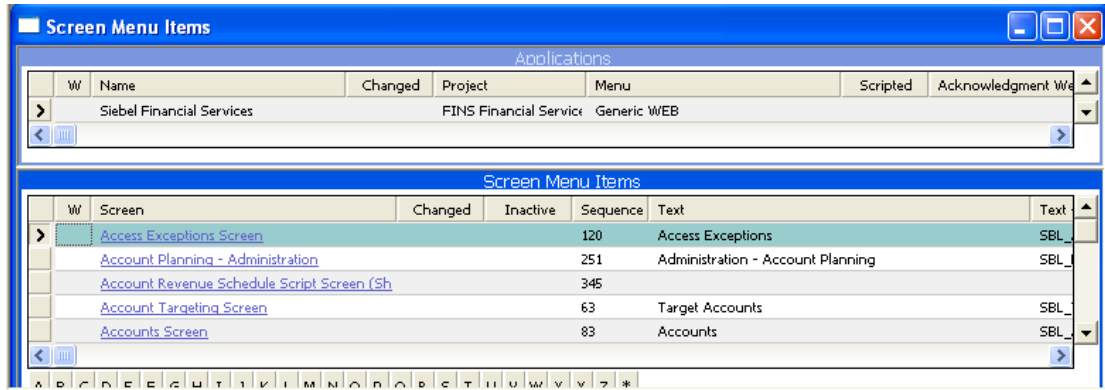


Figura 43. Agregar pantallas a la aplicación en Siebel Tools

3.4.8. Pruebas

Una vez que se completó la solución, inicia la etapa de pruebas. En ellas se deben considerar todas las casuísticas posibles, que ayuden a garantizar la calidad del desarrollo y la satisfacción de todos los requerimientos solicitados.

Las pruebas que se planifican a continuación son todas unitarias, pues sólo se centran en la funcionalidad del evolutivo de inhibición CI e involucran únicamente los componentes modificados. También es importante recordar que todas las pruebas se realizarán únicamente en el servidor de desarrollo. Ya que las pruebas en ambiente QA no competen a nuestro equipo, por protocolos de calidad.

El total de las pruebas que se realizarán se describen en la siguiente tabla:

#	Nombre Prueba	Responsabilidad	Condiciones filtrado	Resultado Esperado
1	Visualización de Cuentas internas para usuarios con responsabilidad informada en Productos/Servicios.	Informada	RUT BH = Indiferente CLASE CUENTA = Indiferente	En esta sección de igual forma se muestran otros datos sobre la CI seleccionada.
2	Visualización de Cuentas internas para usuarios con responsabilidad informada en Empresas	Informada	RUT BH = Indiferente CLASE CUENTA = Indiferente	En esa sección se mostrará el número de cuenta de las Cuentas Corrientes
3	Visualización de Cuentas internas para usuarios con responsabilidad informada en Personas	Informada	RUT BH = Indiferente CLASE CUENTA = Indiferente	En esa sección se mostrará el número de cuenta de las Cuentas Corrientes.

4	Visualización de Cuentas internas para usuarios con responsabilidad no informada pero CI con RUT BH en Productos/Servicios.	No Informada	RUT BH = Igual al RUT Banco CLASE CUENTA = Indiferente	En esta sección de igual forma se muestran otros datos sobre la CI con RUT BH seleccionada.
5	Visualización de Cuentas internas para usuarios con responsabilidad no informada pero CI con RUT BH en Empresas	No Informada	RUT BH = Igual al RUT Banco CLASE CUENTA = Indiferente	En esa sección se mostrará el número de cuenta de las Cuentas Corrientes con RUT BH.
6	Visualización de Cuentas internas para usuarios con responsabilidad no informada pero CI con RUT BH en Personas.	No Informada	RUT BH = Igual al RUT Banco CLASE CUENTA = Indiferente	En esa sección se mostrará el número de cuenta de las Cuentas internas con RUT BH
7	Visualización de Cuentas internas para usuarios con responsabilidad no informada pero CI con Clase Cuenta distinta de CIV y CIN en Productos/Servicios.	No Informada	RUT BH = Diferente a RUT Banco CLASE CUENTA = Distinta a CIV Y CIN	En esta sección de igual forma se muestran otros datos sobre la CI con RUT BH o Clase Cuenta distinta de CIV o CIN.
8	Visualización de Cuentas internas para usuarios con responsabilidad no informada pero CI con Clase Cuenta distinta de CIV y CIN en Empresas.	No Informada	RUT BH = Diferente a RUT Banco CLASE CUENTA = Distinta a CIV Y CIN	En esa sección se mostrará el número de cuenta de las Cuentas Corrientes con RUT BH o Clase Cuenta distinta de CIV o CIN
9	Visualización de Cuentas internas para usuarios con responsabilidad no informada pero CI con Clase Cuenta distinta de CIV y CIN en Personas.	No Informada	RUT BH = Diferente a RUT Banco CLASE CUENTA = Distinta a CIV Y CIN	En esa sección se mostrará el número de cuenta de las Cuentas Corrientes con RUT BH o Clase Cuenta distinta de CIV o CIN.
10	Visualización de Cuentas internas para usuarios con responsabilidad no informada y todas las condiciones de filtrado en Productos/Servicios	No Informada	RUT BH = Diferente a RUT Banco CLASE CUENTA = Igual a CIV Y CIN	En esta sección de igual forma no hay otros datos sobre la CI.
11	Visualización de Cuentas internas para usuarios con responsabilidad no informada y todas las condiciones de filtrado en Empresas.	No Informada	RUT BH = Diferente a RUT Banco CLASE CUENTA = Igual a CIV Y CIN	En esa sección no se mostrará el número de cuenta de las Cuentas internas.
12	Visualización de Cuentas internas para usuarios con responsabilidad no informada y todas las condiciones de filtrado en Personas.	No Informada	RUT BH = Diferente a RUT Banco CLASE CUENTA = Igual a CIV Y CIN	En esa sección no se mostrará el número de cuenta de las Cuentas internas

Tabla 18. Casos de prueba

Importante recordar que si se indica que la prueba se realizara en Productos/servicios, eso implica que se debe seguir la navegación completa por todas las vistas de Productos/Servicios, esa navegación se especifica en el diseño Funcional. A continuación se muestra un ejemplo de cómo se realizó la secuencia de pruebas:

Sec #	Nombre de la Prueba	Ruta	Descripción	Resultado
1	Visualización de CI		Utilizar estos procedimientos para obtener información sobre CI	
1.1	Personas	Pantalla Personas	<ul style="list-style-type: none"> • Ingresar a la pantalla de Empresas. • Seleccionar "Todos los Clientes" • Realizar una búsqueda por medio del RUT persona. 	Se visualizan el registro empresa consultado mediante el RUT persona.
1.2	Visión Global	Pantalla Personas → Vista Visión Global	<ul style="list-style-type: none"> • Una vez seleccionado un registro, ingresar a la vista "Visión Global" • En Applet "Productos y Servicios" se muestra la columna "Familia" y "Cantidad de productos" 	Ahí se muestra la familia "Cuentas Corrientes" y la suma total de CI y Cl. Aunque sólo se puede acceder a las Cuentas Corrientes que no sean internas
1.3	Hipervínculo Productos/Servicios	Pantalla Personas → Vista Visión Global → Applet productos y servicios	<ul style="list-style-type: none"> • Presionar sobre el hipervínculo en la columna "familia" llamado "Cuentas Corrientes" <ul style="list-style-type: none"> • Traslado a pantalla "Productos/Servicios" • En caso de que haya cuentas corrientes y se pueda activar el hipervínculo, se aplica el flujo de la secuencia de prueba "Visualización de Cuentas internas para usuarios con responsabilidad no informada y todas las condiciones de filtrado en Productos/Servicios." 	No es posible visualizar más información de la Cuenta Corriente Interna accediendo a través del hipervínculo a su registro en la pantalla de Productos/Servicios ni en sus vistas correspondientes ("Más información" y "Detalle de producto y servicio").
1.4	Productos	Pantalla Personas → Vista Productos	<ul style="list-style-type: none"> • Presionar sobre el hipervínculo en la columna "familia" llamado "Cuentas Corrientes" <ul style="list-style-type: none"> • Traslado a pantalla "Productos/Servicios" • En caso de que haya cuentas corrientes y se pueda activar el hipervínculo, se aplica el flujo de la secuencia de prueba "Visualización de Cuentas internas para usuarios con responsabilidad no informada y todas las condiciones de filtrado en Productos/Servicios." 	Al igual que el caso anterior, no es posible visualizar más información de la Cuenta Corriente Interna accediendo a través del hipervínculo a su registro en la pantalla de Productos/Servicios ni en sus vistas correspondientes ("Más información" y "Detalle de producto y servicio").
1.5	Resumen Cliente	Pantalla Personas → Vista Resumen Cliente	<ul style="list-style-type: none"> • Al entrar a la Pantalla Personas y a la vista "Resumen Cliente" • Ir a la sección "Otros productos". 	En esa sección no se mostrará el número de cuenta de las Cuentas internas

Tabla 19. Ejecución de una prueba

Durante la etapa de pruebas, se detectaron y corrigieron diversas incidencias, todas por errores en la configuración, afortunadamente no hubo errores funcionales o conflictos relacionados con la definición o el cumplimiento de los requerimientos.

Por cada incidencia corregida es recomendable volver a ejecutar las pruebas para asegurar el funcionamiento de los evolutivos. En cuanto se hayan completado todas las pruebas exitosamente, es requisito documentar y evidenciar.

Algunas de las evidencias del resultado del evolutivo en servidor de desarrollo:

Número de Producto/Servicio	Descripción	Familia	Estado	RUT Persona	RUT Empresa	Razón Social
15 - PAC	PAC CADA VEZ DE DEPORTES Y O...	15 - PAC	Cancelado o Cancel	774311041	774311041	COMPAÑIA ASISTENTE P...
15 - PAC	PAC SERVICIO DE RESPUESTA RÁPIDA...	15 - PAC	Cerrado	774311041	774311041	COMPAÑIA ASISTENTE P...
17 - Pagos	PAGO SERVICIO DE RESPUESTA RÁPIDA...	17 - Pagos	Abierto	774311041	774311041	COMPAÑIA ASISTENTE P...
13 - Canales Remotos	PAGO SERVICIO DE RESPUESTA RÁPIDA...	13 - Canales Remotos	Abierto o Activo	774311041	774311041	COMPAÑIA ASISTENTE P...

Figura 44. Pantalla productos/servicios sin CI

Más Información

1 de 1 Consultas PostVenta Listado De Movimientos Ejecutar

Cuentas - Datos Generales

Cuenta: CMNOCH

*Tipo Cuenta: Cuenta Corriente Moneda Local

Clase Cuenta: CMNOCH

Fecha Apertura: 02/01/2014

Estado Cuenta: Abierto o Activo

Fecha de Expiración:

Cuentas - Datos Abierto

Código:

Datos Producto:

Código Producto:

Fecha Límite:

Clase Cuenta:

Cuentas - Datos Vista

Item:

Fecha:

Figura 45. Pantalla productos/servicios vista más información que sólo muestra registros con clase cuenta distintos de CIN Y CIV.

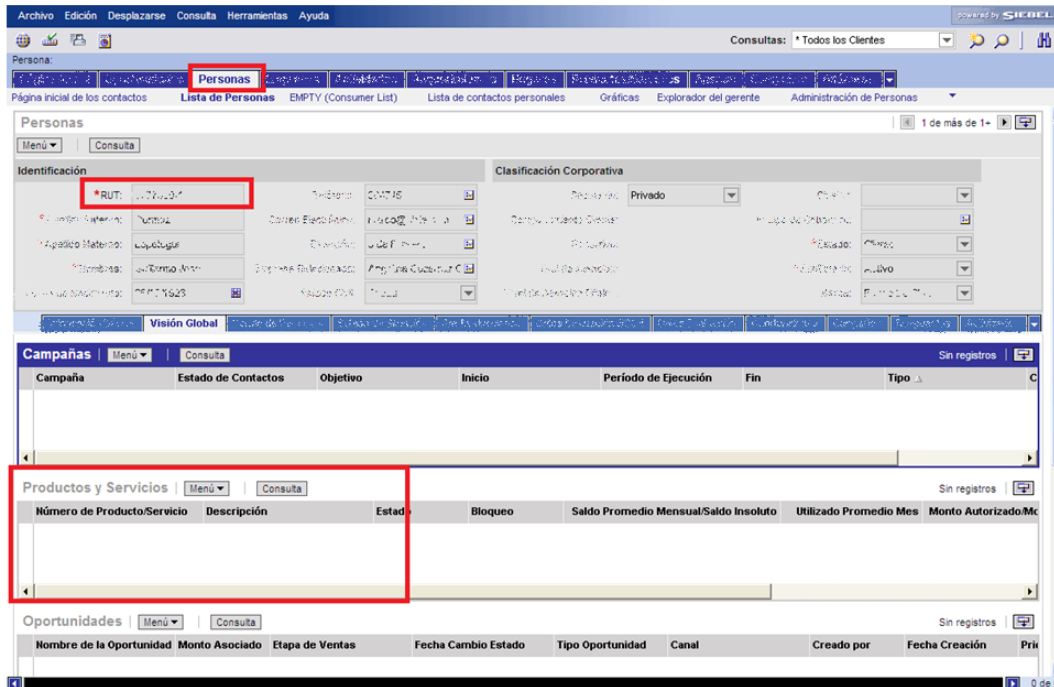


Figura 46. Pantalla personas, vista visión global, que no muestra CI en productos/servicios.

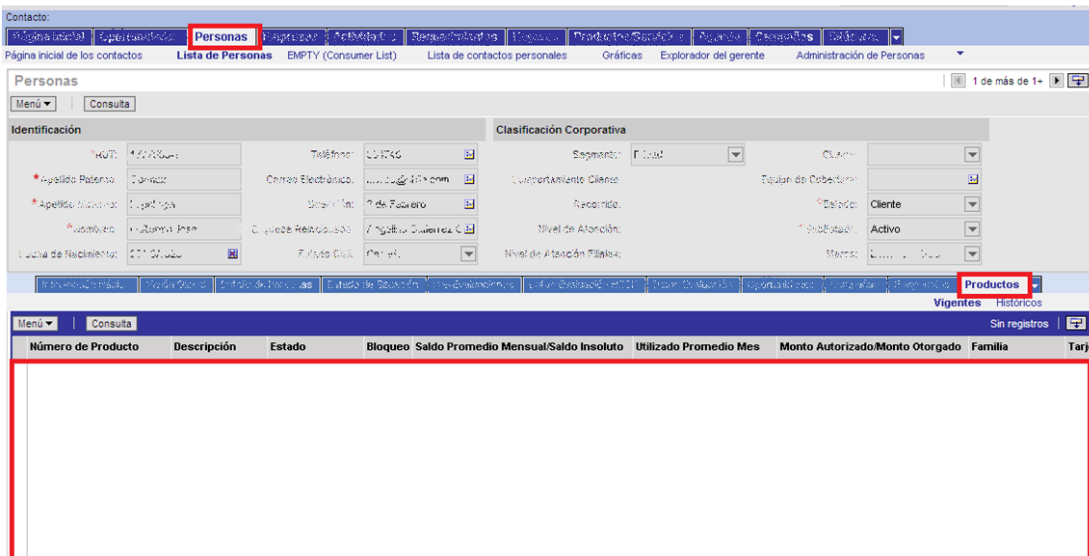


Figura 47. Pantalla personas, vista productos sin mostrar CI.

3.4.9. Ambiente QA y pase a producción.

3.4.9.1. Exportación del ambiente de desarrollo.

Al final del desarrollo y las pruebas, se exportan todos los objetos .SIF involucrados que se mencionan a continuación:

Tipo de Objeto	Objetos totales	Originales	Copias
Aplicación	1	1	0
Pantalla (Screen)	3	3	0
Business Object	3	3	0
Vista (View)	9	9	0
Business Component	10	5	5
Applet	22	11	11
Toggle Applets	46	23	23
Links			0

Tabla 20. Objetos .sif que conforman el evolutivo y se van a exportar

Además de los archivos .SIF, otro elemento importante para la migración a cualquier ambiente son las cargas manuales, es decir, el detalle de todo aquello que se configuró a nivel aplicación y que no se compila por lo tanto no está incluido en los archivos .SIF.

Las cargas manuales se reportan en un documento y es responsabilidad del administrador de la herramienta, asegurarse de crearlas manualmente en cada uno de los ambientes a los que se vaya a migrar el evolutivo.

En el caso de la inhibición de cuentas CI, las cargas manuales generadas son:

- Lista de Valores
- Vistas dadas de alta en la aplicación
- Relación de Vistas – Responsabilidades - Usuarios

En el documento donde se notifican todas las cargas manuales existentes, es necesario incluir todos los campos que el administrador debe cargar en la aplicación. Es muy importante que este proceso sea cuidado muy de cerca por el desarrollador del evolutivo, ya que las cargas manuales siempre implican un riesgo por errores humanos que puede afectar por completo el desempeño del desarrollo. Por supuesto la correcta documentación de las mismas es una forma de mitigar dicho riesgo.

3.4.9.2. *Migración a ambiente QA y certificación*

Después de la etapa de pruebas, una vez que todas fueron alcanzadas con éxito y que se puede comprobar la máxima calidad posible en el ambiente de desarrollo, es momento de migrar el evolutivo al ambiente QA donde se ejecutarán pruebas unitarias e integrales, así como pruebas de estrés y todas las necesarias para asegurar que el evolutivo puede ir a producción.

Yo no participé directamente en esta tarea, pues no era competencia del equipo everis su realización. En el banco existe un equipo alterno de otra empresa que se encarga del ambiente de pruebas, esto es una práctica común para elevar los estándares de calidad y reducir el riesgo de errores en producción.

El proceso de pruebas en QA se mantiene en constante contacto con desarrollo, ya que uno o varios objetos de compilación del evolutivo podrían regresar a desarrollo si se presenta alguna incidencia que afecte el funcionamiento particular o global de la aplicación, por eso es importante que las pruebas en desarrollo abarquen todas las casuísticas posibles. En el peor de los casos incluso podrían presentarse errores referentes al cumplimiento de los requerimientos, pero esto no debiese pasar pues el usuario responsable del banco, da su visto bueno al evolutivo en el ambiente de desarrollo y antes del paso a QA.

Las incidencias en QA que requieran revisión en desarrollo son tratadas con prioridad alta, sobre todo si afectan la funcionalidad general.

Si se recibe una incidencia de QA, el equipo de desarrollo está encargado de su solución, poniendo en práctica el mantenimiento correctivo del que se habló. Para ello, el procedimiento general de solución de incidencias en QA comienza con el reporte por parte del equipo de pruebas, después el equipo de desarrollo lo analiza y diseña una solución. Se reporta al banco y al equipo QA cuáles serán las modificaciones pertinentes a nivel de objetos .SIF o cargas manuales.

Una vez generados los nuevos objetos con las incidencias corregidas y nuevamente realizadas las pruebas en desarrollo, se realiza un respaldo de los objetos de QA y el SRF Actual, enseguida el equipo a cargo del ambiente se encarga de compilar el SRF, siempre de forma total, en el ambiente de pruebas no se hacen compilaciones incrementales.

Otra de las actividades que se realizan cuando el evolutivo llega a QA es el proceso de certificación.

La certificación es un proceso donde se validan todos los cambios realizados en cada uno de los objetos, a nivel del código fuente que se entrega en cada archivo .SIF. La comparación se realiza con respecto al archivo fuente y se utiliza alguna herramienta de comparación de código. Los archivos fuente son la última versión certificada y aprobada de cada objeto.

Es importante recordar que los cambios no se realizaron a nivel de código, por buenas prácticas y porque Siebel es una herramienta que busca evitar la configuración a ese nivel, sin embargo, una vez generado el .SIF se tiene acceso al código en un lenguaje declarativo llamado Siebel Visual Basic.

Siebel VB es muy similar al Visual Basic de Microsoft, por lo que no resulta complicado para un programador que haya trabajado alguna vez con dicho lenguaje. Sin embargo, existen algunas diferencias:

- ✓ Siebel VB no permite la inclusión de ningún objeto de control de interfaz de usuario como botones o estilos. Ya que en Siebel eso se modifica gráficamente en Siebel Tools mediante el diseñador de applets. Y en el código simplemente se hace referencia a él.
- ✓ En Siebel VB no existe el tipo de dato Boolean, si se requiere hay que simularlo.

El proceso de certificación se documenta con la evidencia de cada objeto como se ejemplifica a continuación:

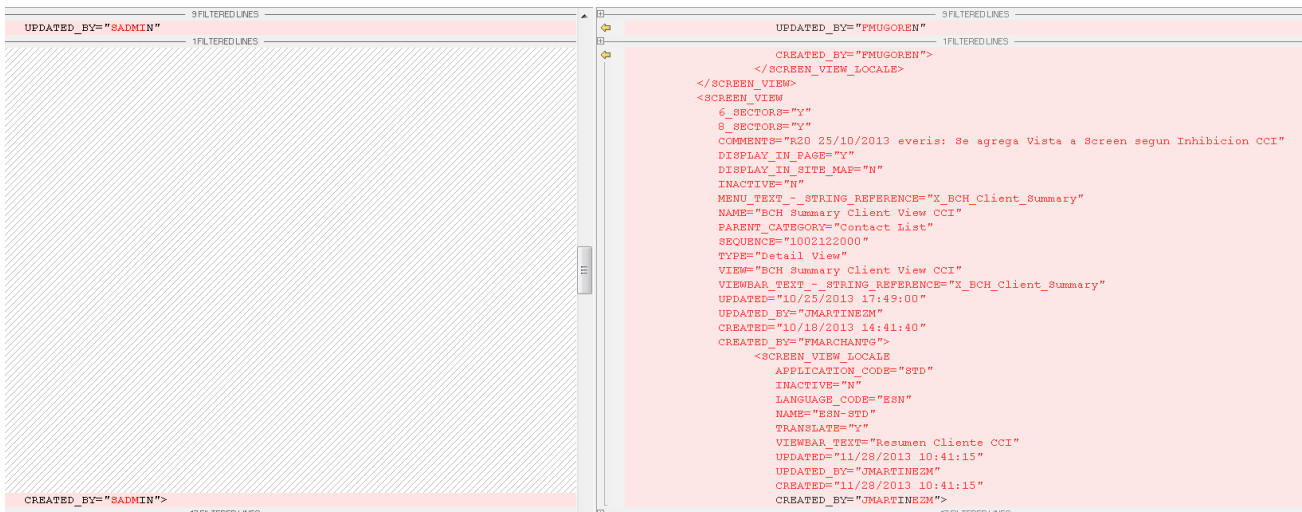


Figura 48. Ejemplo de la evidencia en la certificación de un objeto.

Del lado izquierdo se muestra un fragmento del código del objeto fuente, en este caso una pantalla (Screen), mientras que del lado derecho se muestra la misma sección de código del objeto después del desarrollo con la diferencia de que se agregaron vistas propias del evolutivo.

3.4.9.3. Pase a producción

Después del proceso de certificación en QA y una vez que no se presentaron más incidencias se considera que los evolutivos se encuentran estables, entonces el banco verifica y valida nuevamente los evolutivos y da su visto bueno para iniciar la preparación para el paso a producción.

En el caso de este proyecto de mantenimiento evolutiva y correctiva en el banco, se desarrollaron muchos evolutivos de forma paralela, donde Inhibición CI es sólo uno de ellos.

El banco determinó las mejoras que debían pasar a producción de forma conjunta. Por lo que para llegar a producción después de su estabilización y certificación en QA, inhibición CI tuvo que esperar que el resto de los evolutivos llegaran al mismo estado.

Una vez estable el total de las mejoras, comienza la preparación del pase a producción.

Lo cual implica la generación de todos los componentes que ya fueron certificados en sus archivos .SIF para su exportación del ambiente de QA a producción. También implica la consolidación de las cargas manuales y los objetos que se importarán para todos los evolutivos, teniendo especial cuidado en la claridad de la documentación de los mismos ya que los encargados de realizar las cargas manuales e importaciones en producción son personas del banco, que únicamente conocen el ambiente Siebel pero no los desarrollos realizados.

El banco estableció la fecha de pase a producción que se reservó para un fin de semana por las noches, para no interferir con el correcto desempeño de la herramienta, ya que esto afectaría directamente la operación bancaria. En primer lugar se instalaron los nuevos objetos .SIF y se hicieron las cargas manuales e importaciones. Cuando el nuevo SRF estaba también instalado en producción se inició la etapa de pruebas, que a diferencia de las pruebas en desarrollo y QA, se desarrollaron todas en los 3 días destinados al pase a producción con la presencia del usuario responsable de cada proceso por parte del banco, para dar su aprobación final.

Una semana antes del paso a producción los ambientes se congelan y se hacen las exportaciones de objetos.

Mi participación en los pases a producción fue en las actividades de cargas manuales directamente en el ambiente productivo, pero sólo en un rol supervisor, ya que se pidió que únicamente el personal autorizado por el banco realizara la instalación y configuración.

El pase a producción de todas estas mejoras se dio en 2 intentos de instalación/configuración en producción.

No se realizó el paso a producción la primera vez dado que existe una casuística no considerada en uno de los evolutivos (no en inhibición CI), funcional y técnica por lo tanto la usuaria responsable no entrega su aprobación. Por esta razón se dio marcha atrás a todo lo realizado en producción, se regresó el evolutivo a QA para su revisión y a desarrollo para su corrección, lo que provocó la postergación del segundo intento de pasar a producción por algunas semanas

El nuevo intento de pase a producción implicó el mismo procedimiento que el interior, salvo porque quienes habían desarrollado el evolutivo que falló tuvieron que realizar el procedimiento de resolución de incidencias con prioridad urgente. Este segundo intento se realizó con éxito y los usuarios dieron su aprobación, por lo que las mejoras ya se utilizan en el CRM del banco desde entonces.

3.5. Resultados

Al hacer la petición de este evolutivo, el banco busca mejorar su servicio, filtrando información interna para áreas que tienen trato directo con el cliente, esto en el contexto de la herramienta de CRM que tienen implementada en su estrategia tecnológica y de negocio.

Después de la implementación del evolutivo y haciendo una retrospectiva, se logró implementar una solución lo menos invasiva posible, ya que ésta se integró con el resto de la funcionalidad al pasar a producción. A pesar de que fue necesario copiar elementos, lo cual puede representar un riesgo a nivel de referencias era mucho más viable dicha estrategia que otras como la utilización de Toggle Applets para mostrar o no información de Cuentas internas, ya que dicha solución es mucho más costosa en cuestión de rendimiento para la aplicación. Además con la solución elegida, se simplificó el trabajo de la nueva asignación de responsabilidades informadas o no informadas. También se logró que el filtro fuera independiente del valor que actualmente se considera para que una CI se muestre o no en pantalla, esto es importante por los constantes cambios que sufre la herramienta tanto en definición, como funcionalidad y desempeño.

El resultado final del evolutivo en el ambiente de QA fue de pocas incidencias registradas, todas de menor impacto que pudieron ser resueltas sin afectar la funcionalidad global del ambiente. Estas incidencias surgieron en pruebas integrales y fueron todas relacionadas al mayor riesgo que implicaba la solución, las referencias de los objetos al copiarse. Por lo que sólo fue necesario revisar y ajustar algunas referencias y relaciones sobre todo entre Business Objects, Business Components y Applets, con sus respectivos Links.

Una vez que el desarrollo se estabilizó en QA, pasó a producción sin mayor inconveniente donde tampoco presentó dificultades durante la etapa de soporte post productivo. Lo cual comprueba que la solución fue diseñada correctamente.

Con esta mejora ya funcionando en producción desde hace algunos meses, el banco ganó eficiencia en su trato con el cliente, pues quienes se encargan directamente de su atención sólo tienen acceso a la información que les compete. Aunado a las ventajas que se lograron gracias a esta funcionalidad, se tienen algunas otras características importantes como la alta disponibilidad en que opera Siebel según sus estándares, porque es una herramienta que cuenta con planes de contingencia, esto se traduce en un buen tiempo de respuesta en el funcionamiento de la aplicación, lo cual a su vez mejora considerablemente el tiempo de respuesta de cara al cliente, cumpliendo con los niveles de servicio requeridos por el negocio.

Conclusiones

Para terminar, con base en el proyecto realizado, quiero destacar lo enriquecedor que es participar en proyectos de consultoría, que permiten cambiar constantemente de ambientes, herramientas, tecnologías, equipos multidisciplinarios y modelos de negocio, lo cual abre el panorama para cualquier profesional, sobre todo en las áreas de ingeniería cuyo rango de acción es tan amplio. Además, se añade el hecho de tener la oportunidad de participar en proyectos en otro país, donde el entorno general del proyecto cambia y representa un reto en todos los aspectos, ya que aunque el lenguaje de la ingeniería es universal, la forma de entenderlo varía culturalmente y al final de la experiencia, todo se traduce en aprendizaje integral.

La razón por la que elegí este proyecto de entre los que he realizado hasta ahora, para desarrollar mi informe de experiencia profesional, fue que me permitió involucrarme libremente en su solución, es decir, tuve la facultad de pensar y plantear una solución integral para el proyecto en base a mi criterio y conocimiento, desde el análisis y diseño funcional y técnico, hasta su construcción y aunque la solución aquí presentada luce sencilla, hay que recordar que únicamente se está mostrando el resultado final. Llegar a ella implicó un arduo trabajo tanto teórico como práctico donde evalué diversas soluciones, hasta llegar a la más óptima tomando en cuenta todos los factores: aspectos de rendimiento, implicaciones a futuro, impacto al resto de la funcionalidad y a la herramienta, riesgos, etc. Siempre alineado a cumplir con los requerimientos del cliente. Hasta llegar a la solución que implementé, que fue la que tenía una mejor relación entre eficiencia y satisfacción de lo solicitado.

Trabajar con una herramienta que ya está construida para ciertos fines, no hace más sencillo alcanzar el objetivo para el que se implementó en una organización, en comparación con la construcción de soluciones tecnológicas desde cero. Únicamente implica un trabajo distinto cuando se quieren implementar o cambiar funcionalidades, ya que siempre se requiere un trabajo de ingeniería inversa para entender cómo está funcionando actualmente la aplicación, así como un conocimiento pleno de la arquitectura, la estructura de datos, las transacciones, sus capacidades y limitantes aunado al dominio del negocio al que está aplicada. Siempre tomando en cuenta el tipo de solución tecnológica que es, en este caso un CRM, para no perder el objetivo que se busca al elegirla, ya que como se revisó en el informe, más que una solución de paquete, implica una estrategia de negocio y debe funcionar como tal.

En conclusión, todo este trabajo de análisis, diseño y construcción que llevan a la solución de un problema, y su implantación final en el ambiente productivo es la oportunidad perfecta para poner en práctica todas las habilidades que se adquirieron durante la carrera universitaria y a través de la experiencia en otros proyectos.

Referencias

“Bookshelf for Oracle's Siebel Business Applications Version 7.7” [En Línea]:

Oracle 2012.- Consultado en Agosto 2013

http://docs.oracle.com/cd/E05553_01/homepage.htm

“Siebel Essentials” [Curso]: Oracle 2007

“Oracle Siebel CRM Developer’s Handbook” [Libro] / aut. Alexander Hasal, Packt Publishing, 2011.