



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**PROYECTO DE IMPLEMENTACION DE SAP
ERP**

INFORME DE ACTIVIDADES PROFESIONALES

QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN

P R E S E N T A

Alejandro Hernández Guadarrama

ASESORA DE INFORME

M. en I. Norma Elva Chávez

Ciudad Universitaria, Ciudad de México, Abril 2016



Tabla de contenido

Introducción	7
Capítulo 1. SAP y AltriaTec	8
SAP	9
La empresa y su historia.	9
El ERP.	9
Asociaciones.	10
Metodología de implementación SAP.	10
Metodología ASAP estándar	12
Preparación del proyecto	12
Documentación (Business blueprint)	12
Realización	12
Preparación final	12
Soporte Go-live	13
Operación	13
Metodología ASAP rápida y simplificada.....	13
Preparación del proyecto	13
Validación del alcance	13
Realización	13
Preparación final	13
Soporte Go-live	13
Operación	14
Metodología ASAP ágil	14
Preparación del proyecto	14
Blueprint	14
Realización	14
Preparación final	14
Soporte Go-live	15
Operación	15
AltriaTec	15
Historia	15
Estructura de la empresa.....	15
Capítulo 2. ABAP y marco técnico.....	16

Arquitectura del sistema.	17
Estructura del servidor de aplicación	18
Procesos de negocio Cross system	19
Application Link Enabling (ALE).....	20
Conexiones con otros sistemas: RFC y BAPI	21
RFC Remote Function Call.....	21
Repositorio de objetos (BOR) y BAPI.....	22
El lenguaje ABAP	23
Historia	23
Tipos de programas en ABAP.....	23
Flujo de procesamiento de un programa ABAP.....	24
Repositorio de información.	28
El navegador de objetos.	28
Organización de desarrollos.	29
Capítulo 3. Implementación de sistema SAP.....	31
Preparación del proyecto y validación del alcance.....	32
Alcance funcional.	33
Alcance técnico	39
Realización	42
Tipos de desarrollos.	42
Estrategia de desarrollo.....	44
Seguimiento de avances.	45
Preparación final.	47
Procedimiento.....	47
Plan de cutover para el proyecto.	48
Go-live y soporte.	50
Soporte	51
Capítulo 4. Resultados.	52
PI	53
ERP.	53
CRM.....	53
SLA.....	55
Deducciones.	57

Ejemplos de aplicaciones.	60
Conclusiones.	65
Apéndice 1. Metodología SCRUM.	66
Historia.	66
Características.	66
Principales roles.	68
Product Owner.	68
SCRUM máster.	68
Equipo de desarrollo.	68
Reuniones de SCRUM.	68
Reunión diaria.	68
Reunión de SCRUMs.	68
Reunión planificadora de sprint.	69
Reunión de revisión del sprint.	69
Beneficios del SCRUM.	69
Apéndice 2. Ejemplos de tipos de desarrollos.	71
Tabla de diccionario.	71
Programa de carga.	71
Programa tipo reporte.	72
Presentación de un reporte.	77
Formulario.	78
Fuentes de información.	79
Glosario.	80

Figuras y tablas

Figura1: Fases de la metodología de implementación ASAP.	¡Error! Marcador no definido.
Figura2: Fases de la metodología estándar ASAP.....	¡Error! Marcador no definido.
Figura3: Fases de la metodología ASAP ágil.	¡Error! Marcador no definido.
Figura4: Estructura organizativa de AltriaTec.	¡Error! Marcador no definido.
Figura 5: Arquitectura del servidor SAP NetWeaver	¡Error! Marcador no definido.
Figura6: Procesos de trabajo y despachadores.	¡Error! Marcador no definido.
Figura7: Distribución de procesos utilizando ALE.....	¡Error! Marcador no definido.
Figura8: BAPI	¡Error! Marcador no definido.
Figura 9: Flujo de programa (1)	¡Error! Marcador no definido.
Figura 10: Flujo de programa (2)	¡Error! Marcador no definido.
Figura 11: Flujo de programa (3)	¡Error! Marcador no definido.
Figura 12: Flujo de programa (4)	¡Error! Marcador no definido.
Figura 13: Flujo de programa (5)	¡Error! Marcador no definido.
Figura 14: Flujo de programa (6)	¡Error! Marcador no definido.
Figura 15: Flujo de programa (7)	¡Error! Marcador no definido.
Figura 16: Repositorio de información y arquitectura R3.....	¡Error! Marcador no definido.
Figura 17: Panel de navegación en ABAP workbench.....	¡Error! Marcador no definido.
Figura 18: Sistema de transportes.....	¡Error! Marcador no definido.
Tabla 1: Componentes considerados para CCO.	¡Error! Marcador no definido.
Tabla 2: Componentes considerados para coordinación de operaciones.	¡Error! Marcador no definido.
Tabla 3: Componentes considerados para coordinación de gestión...	¡Error! Marcador no definido.
Figura 19: Módulos y sub módulos del ERP implementados.	¡Error! Marcador no definido.
Figura 20: PI como punto central de omunicaciones.	¡Error! Marcador no definido.
Tabla 4: Desarrollos por módulo.	¡Error! Marcador no definido.
Tabla 5: Tipos de desarrollos	¡Error! Marcador no definido.
Figura 21: Tablero SCRUM	¡Error! Marcador no definido.
Tabla 6: Plan de cutover.....	¡Error! Marcador no definido.
Figura 22: Plan de go- live.	¡Error! Marcador no definido.
Figura 23: Ciclo de vida de la solicitud de servicio.....	¡Error! Marcador no definido.
Figura 24. Ciclo de SLA.	¡Error! Marcador no definido.

Figura 25: Definición de tiempos en un ticket.....	¡Error! Marcador no definido.
Figura 26: Ejemplo de reporte, interfaz de usuario.....	60
Figura 27: Ejemplo de reporte, ejecución.....	61
Figura 28: Ejemplo de factura generada	62
Figura 29. Ejemplo de factura generada.	¡Error! Marcador no definido.
Figura 30. Ejemplo de reporte.....	¡Error! Marcador no definido.
Figura 31: Ejemplo de reporte.....	¡Error! Marcador no definido.
Figura 32: Ejemplo de formulario	¡Error! Marcador no definido.

Introducción

En el presente documento se presenta mi experiencia laboral obtenida a partir de los conocimientos adquiridos como egresado de la Facultad de Ingeniería en la carrera de Ingeniería en Computación, en el periodo 2002 – 2008, concluida en el mes de junio con un promedio de 8.45.

Casi inmediatamente después de haber terminado, me incorporé como consultor técnico a la empresa AltriaTec S.A. de C.V. Sin embargo, fue hasta diciembre de ese mismo año que comencé la carrera de consultor técnico ABAP, implementando el ERP SAP.

Con base en lo anterior, este documento se enfocará en dar a conocer un proceso de implementación del ERP de SAP y cómo es que aporté conocimientos a dicha implementación como consultor técnico, de esta manera, presentaré información acerca de lo que es SAP, así como su arquitectura y la metodología que maneja, además de un breve informe de lo que es el lenguaje ABAP.

Con los elementos anteriores, se describirá lo que se realizó en un proyecto típico de implementación de un ERP en una empresa de construcción, en donde se instalaron las soluciones ERP y CRM de SAP para gestionar las necesidades de la administración de recursos y solicitudes de servicios.

Capítulo 1. SAP y AltriaTec

SAP

La empresa y su historia.

SAP es una empresa multinacional de software de origen alemán, cuyo principal objetivo es crear software empresarial para administrar operaciones de negocio y relaciones con el cliente. Tiene sus oficinas centrales en Waldorf, Alemania, y presencia regional en 130 países, entre los que se encuentra México.

Xerox apuntó a salir de la industria de las computadoras para 1975 y pidió ayuda a IBM para migrar sus sistemas de negocios a dicha tecnología (IBM). Como parte de la compensación por la migración, IBM se quedó con los derechos del Sistema de Datos Científicos (SDS por sus siglas en inglés), reportado con un contrato crediticio por \$80,000 USD.

5 ingenieros del departamento de inteligencia artificial, se encontraban trabajando en un sistema empresarial basado en este software, el cual, se indicó que ya no sería necesario, sin embargo, en lugar de abandonar el proyecto, decidieron dejar IBM e iniciar una nueva compañía.

En junio de 1972, estos 5 ingenieros fundaron la compañía “System Analysis and Program Development” como una asociación privada bajo el código civil alemán. Su primer cliente fue la división alemana de Industrias químicas imperiales, en Östringen, donde desarrollaron programas de mainframe para nómina y contabilidad.

El ERP.

En 1973, se lanza el primer producto comercial: el SAP R/98, ofreciendo un sistema común para múltiples tareas. Este sistema permitió el uso de almacenamiento de datos centralizado, mejorando la administración de datos.

En 1979, se lanza SAP R/2, expandiendo las capacidades del sistema a otras áreas, como Gestión de materiales (MM) y Planeación de producción (PP). Fue hasta 1992 cuando SAP introdujo su sistema R/3, el cual sufrió varios cambios de versión hasta 1995, y para estas fechas, SAP se unió a la tendencia de cambiar la computación centralizada por una arquitectura cliente – servidor.

Asociaciones.

Los socios (Partners) de SAP, incluyen:

- Socios de servicios globales con capacidades de consultoría multinacional entre industrias.
- Socios de software globales que proporcionan productos integrados que complementan las soluciones de la suite de soluciones de negocio SAP.
- Socios de tecnología globales que proveen a las compañías con una amplia gama de productos para apoyar a la tecnología de SAP, incluyendo vendedores de hardware, bases de datos, sistemas de almacenamiento, redes y tecnología móvil.

Existe un número pequeño de compañías que se denomina Extensions partners, los cuales proporcionan funcionalidades que complementan las capacidades de los productos SAP. Estos complementos cumplen con los más altos estándares y están certificados, apoyados y son vendidos por SAP directamente. Algunos de estos socios son:

- Adobe
- CA technologies
- GK software
- HP
- IDS Scheer
- OpenText

Metodología de implementación SAP.

SAP utiliza un conjunto central de metodologías y herramientas diseñadas para entregar resultados de una manera rápida y confiable, y de esta manera ayudar a los clientes a maximizar los resultados para sus soluciones. Esto incluye el marco de la metodología ASAP (el cual es seguido por el equipo de trabajo para implementar el software de SAP de una manera eficiente), y la suite de gestión SAP Solution Manager (SOLMAN).

Esta metodología ASAP entrega una metodología de contenido estructurado –procesos, procedimientos, aceleradores, listas de verificación, ligas a documentación de SAP, ect.- necesario para la implementación de soluciones SAP.

La última versión de la metodología ASAP (v8) se basa en los fundamentos sólidos de la versión anterior que incluye la transparencia de realización de valor, a través de casos de negocio. Una guía eficiente de SOA, BPM y proyectos de implementación tradicional a lo largo del ciclo de vida completo de un proyecto, que va desde la evaluación, pasando por la entrega y hasta la gestión de la solución después de la implementación y operaciones. De

esta manera se entrega contenido revisado en todas las áreas que se necesitan para hacer más eficientes a los equipos del proyecto:

- Gestión
- Administración
- Cambio organizacional
- Capacitación
- Documentación
- Configuración
- Pruebas
- Plan de instalación (cutover)
- Ejecución



Figura1: Fases de la metodología de implementación ASAP.

La figura 1 muestra las principales fases en un proyecto típico de implementación de SAP, sin embargo, adicionalmente a la metodología estándar, existen 2 variantes del modelo de implementación ASAP:

- Simplificado y rápido.
- Ágil

Metodología ASAP estándar



Figura2: Fases de la metodología estándar ASAP.

En la figura 2, podemos ver las fases principales que componen la metodología estándar de una implementación, a continuación, una breve explicación de cada una de ellas.

Preparación del proyecto

Durante esta fase, el equipo debe hacer la planeación y preparación inicial para el proyecto SAP.

Documentación (Business blueprint)

El propósito de esta fase es alcanzar un entendimiento común de cómo la compañía pretende implementar y operar su negocio con SAP. En la metodología estándar, el resultado es el documento denominado como Business blueprint - BBP de aquí en adelante – que contiene detalles del análisis realizado durante los talleres con el personal correspondiente asignado por el cliente.

Realización

El propósito de esta fase es implementar todos los requerimientos de los procesos de negocio, basados en el BBP. La configuración del sistema en la metodología ASAP estándar se realiza en 2 bloques de trabajo:

- Configuración base (o baseline), la cual se refiere a un alcance mayor y generalizado.
- Configuración final.

Durante esta fase, también se realizan pruebas al sistema

Preparación final

El propósito de esta fase es afinar los detalles faltantes de la realización, incluyendo pruebas técnicas, unitarias e integrales, ciclos de capacitaciones a los usuarios finales, gestión del sistema y actividades de cutover, para finalizar la preparación con miras en la puesta en productivo (go-live).

Esta fase también sirve para resolver los detalles críticos que hayan surgido. Una vez completada esta fase, el negocio se encuentra listo para poder correr sus operaciones con SAP.

Soporte Go-live

El propósito de esta fase es moverse de un ambiente orientado al proyecto / pre productivo a la operación en un ambiente de producción.

Operación

En esta fase el sistema se encuentra ya en operación con la ayuda de la plataforma central, conformada por el personal capacitado por parte del cliente, SAP SOLMAN y la documentación generada durante las fases anteriores.

Metodología ASAP rápida y simplificada

Preparación del proyecto

Esta fase proporciona planeación y preparación inicial para el proyecto. Cada proyecto tiene sus objetivos, alcance y prioridades únicos. Los entregables descritos en esta fase ayudan a completar los pasos de iniciación y planeación de una manera eficaz y eficiente.

Validación del alcance

El propósito de esta fase es alcanzar un entendimiento común de cómo la compañía pretende correr SAP en sus negocios. Se enfoca en una configuración rápida de un ambiente que estará disponible para talleres de validación con los usuarios del negocio para confirmar el alcance y determinar los requerimientos que implicarán un cambio y que se realizarán en la siguiente fase del proyecto para mejorar y complementar el baseline proporcionado por los Sistemas de rápido desarrollo (RDS por sus siglas en inglés).

Realización

El propósito de esta fase es implementar y configurar los requerimientos de cambio definidos durante la fase de validación del alcance. El equipo configura, desarrolla, prueba y documenta la solución en series de iteraciones calendarizadas. Antes de poder liberar la solución para la siguiente fase, se deben realizar pruebas unitarias, e integrales end-to-end, y éstas deben ser aceptadas por el usuario.

Preparación final

El propósito de esta fase es completar las actividades de cutover (incluyendo pruebas técnicas y carga de datos maestros, capacitación al usuario gestión del sistema y actividades de simulación e cutover) para finalizar y afinar detalles con miras a la puesta en producción. Esta fase también sirve para resolver todos los incidentes críticos que se hayan presentado. En una implementación exitosa de esta fase, el sistema quedará listo para correr en producción.

Soporte Go-live

El propósito de esta fase es moverse de un ambiente orientado al proyecto / pre productivo a la operación en un ambiente de producción.

Operación

Esta fase tiene como propósito afinar los ciclos de vida de aplicaciones, procesos y procedimientos establecidos durante el proyecto, y alinearlos con las necesidades del negocio. La plataforma central de operación de SAP, será SOLMAN, el cual gestiona la documentación para la operación de los sistemas.

Metodología ASAP ágil



Figura3: Fases de la metodología ASAP ágil.

A continuación, la descripción de las fases de una metodología ASAP ágil, mostradas en la figura 3.

Preparación del proyecto

Esta fase proporciona planeación y preparación inicial para el proyecto. Cada proyecto tiene sus objetivos, alcance y prioridades únicos. Los entregables descritos en esta fase ayudan a completar los pasos de iniciación y planeación de una manera eficaz y eficiente.

Blueprint

El propósito de esta fase es alcanzar un entendimiento común de cómo la compañía pretende correr SAP en sus negocios. Se enfoca en una configuración rápida del entorno de validación para los talleres con los usuarios de negocio para confirmar el alcance y determinar los requerimientos de cambio que serán realizados en la siguiente fase del modelo.

Realización

El propósito de esta fase es implementar y configurar los requerimientos de cambio definidos durante la fase de blueprint. El equipo configura, desarrolla, prueba y documenta la solución en series de iteraciones calendarizadas. Antes de poder liberar la solución para la siguiente fase, se deben realizar pruebas unitarias, e integrales end-to-end, y éstas deben ser aceptadas por el usuario.

Preparación final

El propósito de esta fase es completar las actividades de cutover (incluyendo pruebas técnicas y carga de datos maestros, capacitación al usuario gestión del sistema y actividades de simulación e cutover) para finalizar y afinar detalles con miras a la puesta en producción. Esta fase también sirve para resolver todos los incidentes críticos que se hayan presentado. En una implementación exitosa de esta fase, el sistema quedará listo para correr en producción.

Soporte Go-live

El propósito de esta fase es moverse de un ambiente orientado al proyecto / pre productivo a la operación en un ambiente de producción.

Operación

Esta fase tiene como propósito afinar los ciclos de vida de aplicaciones, procesos y procedimientos establecidos durante el proyecto, y alinearlos con las necesidades del negocio. La plataforma central de operación de SAP, será SOLMAN, el cual gestiona la documentación para la operación de los sistemas.

AltriaTec

Historia

Es una empresa mexicana fundada el 8 de abril de 2002 por Felipe Martínez y Luz Mery Rodríguez, dedicada a dar soluciones en cuestión de logística de negocio y empresarial basada en la tecnología que ofrece SAP.

Estructura de la empresa

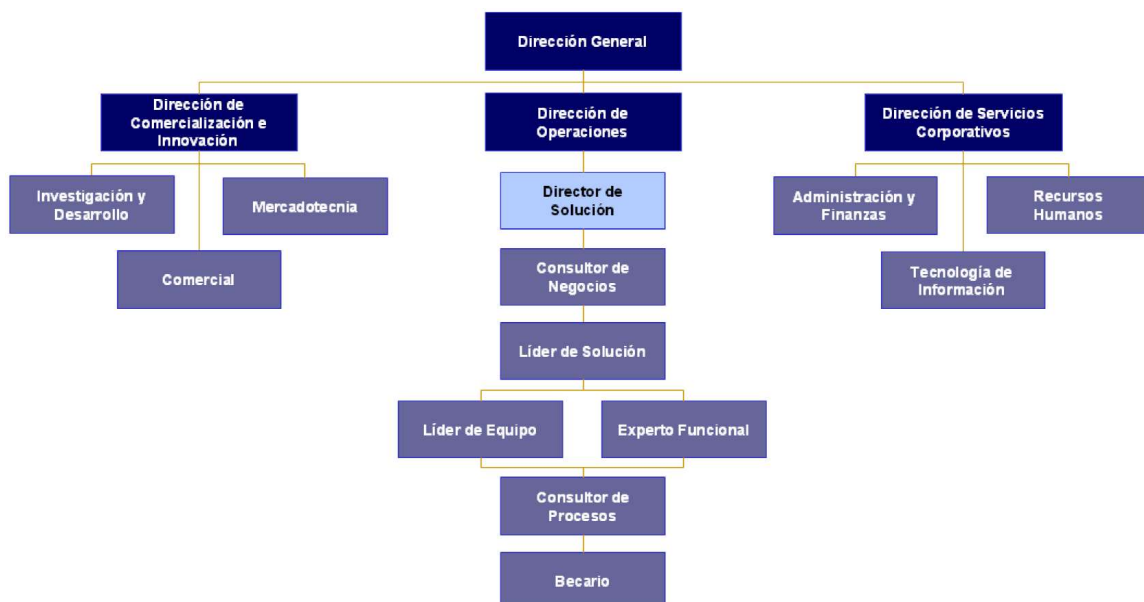


Figura4: Estructura organizativa de AltriaTec.

Capítulo 2. ABAP y marco técnico.

Arquitectura del sistema.

Como se muestra en la figura 5, el servidor de aplicación de SAP NetWeaver tiene una arquitectura modular que sigue el principio de cliente servidor, dicha arquitectura se compone de 3 capas:

- Presentación
- Aplicación
- Datos

Estas capas pueden ser asignadas a diferentes sistemas, lo que sirve como base para la escalabilidad del sistema.

El nivel más bajo es el de datos, en donde se administra con ayuda del sistema de gestión de base de datos relacional (RDBMS por sus siglas en inglés). Esta información incluye, aparte de los datos de aplicación, los programas y metadatos que el sistema SAP requiere para auto gestionarse.

Los programas ABAP corren en el nivel de servidor de aplicación, estos programas son los que proporciona SAP y los que se crean por los desarrolladores en el transcurso del proyecto, estos programas leen información de la base de datos, la procesan y almacenan nueva si es necesario.

El tercer nivel es el del servidor de presentación, el cual contiene la interfaz de usuario, en donde cada uno de ellos puede acceder a los programas, introducir nueva información y recibir los resultados de un proceso de trabajo (WP por sus siglas en inglés).

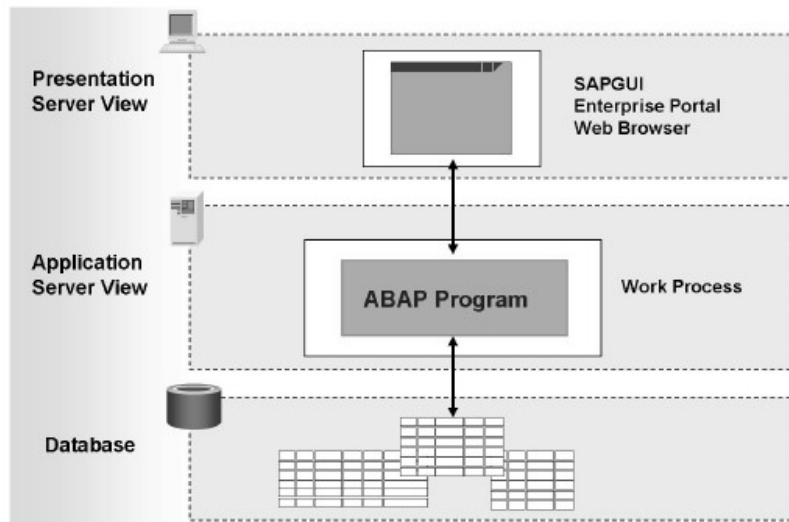


Figura 5: Arquitectura del servidor SAP NetWeaver

La distribución técnica del software es independiente de su alojamiento físico en el hardware. Hablando “verticalmente”, todos los niveles pueden ser instalados en un solo servidor, aunque lo recomendable es que se encuentren cada uno en un servidor diferente. “Horizontalmente”, los servidores de presentación y aplicación pueden ser divididos entre cualquier cantidad de computadoras.

Estructura del servidor de aplicación

Todos los servidores de aplicación de ABAP incluyendo el servidor de mensajes, representan la capa de aplicación de la arquitectura de un sistema basado en ABAP, estos servidores ejecutan las aplicaciones ABAP y se comunican con los componentes de presentación, base de datos y también entre ellos mismos y con otros sistemas utilizando el servidor de mensajes.

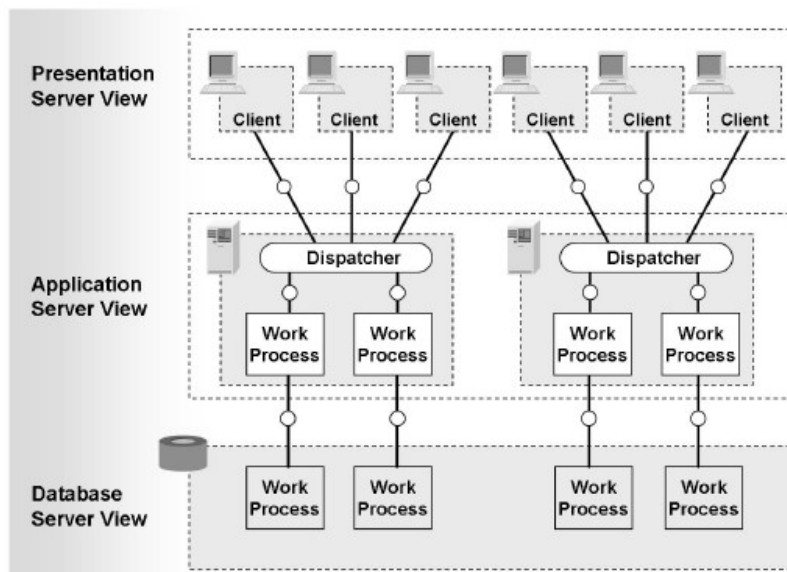


Figura6: Procesos de trabajo y despachadores.

Adicionalmente a los procesos de trabajo, cada servidor de ABAP contiene un despachador, un Gateway y una memoria compartida, como se muestra en la figura 6.

- **Procesos de trabajo:** Son componentes capaces de ejecutar una aplicación, cada uno de estos procesos, está ligado a un área de memoria que contiene el contexto de la aplicación que se está ejecutando, el contexto tiene la información actual del programa de aplicación.
- **Despachador:** Es la liga entre los procesos de trabajo y los usuarios firmados en el servidor de aplicación. Su tarea es recibir peticiones de los pasos de diálogo desde el SAP GUI (servidor de presentación) y direccionarlos a un proceso de trabajo que se encuentre libre. Del mismo modo, direcciona la salida de pantalla proveniente del WP hacia el usuario adecuado.
- **Gateway:** Es la interfaz para los protocolos de comunicación del servidor de aplicación, puede comunicar con otros servidores, ya sean SAP o no SAP.
- **Memoria compartida:** Todos los procesos de trabajo en el servidor de aplicación de ABAP usan una memoria principal común llamada memoria compartida, para salvar los contextos o para almacenar en buffer datos constantes localmente.

Procesos de negocio Cross system

Definamos los Cross Systems (CS en adelante) con un ejemplo: En una compañía dada, el sistema de recursos humanos está separado del resto del software del negocio, obviamente, los sistemas no pueden estar separados del todo, dado que contabilidad

necesita la información nominal de los empleados. En esta situación, se necesita una aplicación de negocios CS para intercambio de la información relevante.

Los procesos de negocio CS se usan por ejemplo, si dos empresas desean colaborar de cerca órdenes conjuntas a un vendedor. Los departamentos de TI de las empresas necesitan comunicarse el uno con el otro para consolidar las cantidades que serán solicitadas, y en este caso, los procesos de negocio no romperán los límites de comunicación entre sistemas solamente, sino también entre las compañías.

Application Link Enabling (ALE)

ALE es una forma de crear y operar aplicaciones distribuidas. El concepto básico de ALE es asegurar la operación de un diseño de sistema distribuido, pero integrado, lo que involucra intercambio de mensajes controlado entre los diferentes sistemas utilizando información consistente y libre entre los sistemas de aplicación involucrados. Las aplicaciones están integradas a través de comunicación síncrona y asíncrona, no así por medio de una base de datos central.

Los sistemas que usan ALE para intercambiar información pueden estar alojados en la misma compañía o pertenecer a distintas. Una de las características de ALE es que sistemas diferentes pueden ser conectados en términos del negocio por medio de transferencia de información consistente y segura.

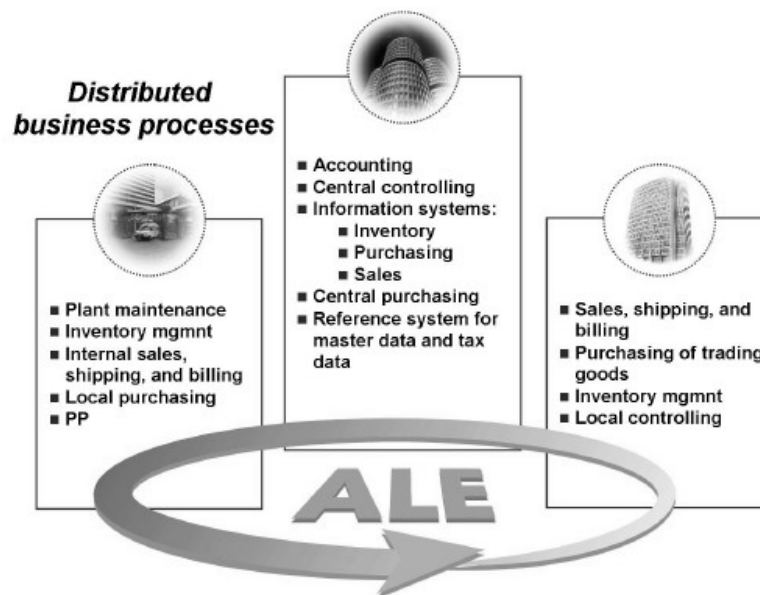


Figura7: Distribución de procesos utilizando ALE.

Para implementar ALE es necesario que queden claro los siguientes puntos:

1. Identificar los procesos y objetos involucrados.
2. Identificar la información que será transmitida.
3. Especificar el formato de la información que será transmitida.
4. Decidir la tecnología de transferencia que se utilizará.
5. Decidir el tipo de transferencia.
6. Especificar el destino de la información.

Dentro de un sistema SAP la información se identifica utilizando un objeto de negocio y su BAPI (método de un objeto).

Para la estructura de transferencia entre sistemas SAP – no SAP, se utilizan principalmente los IDocs, (Documentos intermedios), de los cuáles existen varios tipos y formatos diferentes con los cuáles se puede intercambiar la información.

En cuanto a la tecnología de transferencia de información, se puede seleccionar aquella que se encuentre dentro de los estándares de cada sistema, en SAP existen las RFC (Remote Function Call por sus siglas en inglés) y los protocolos HTTP o HTTPS.

Existen 2 principales tipos de transferencia:

- Síncrona
- Asíncrona

[Conexiones con otros sistemas: RFC y BAPI](#)

SAP echa mano de diferentes tecnologías como ALE, EDI, HTTP, SMTP, SOAP y XML para conectarse con otros sistemas, pero de las tecnologías que se usan, hay 2 que sobresalen por su capacidad: RFC y BAPI.

[RFC Remote Function Call](#)

La RFC es un protocolo de interfaz de SAP basado en CPI-C y TCP/IP, el cual simplifica la programación de los procesos de comunicación entre diferentes sistemas. Las RFC permiten llamar y ejecutar funciones predefinidas en un sistema remoto o dentro del mismo sistema, además de gestionar los procesos de comunicación, transferencia de parámetros y manejo de errores.

La RFC describe una interfaz, no el lenguaje de programación en el cual se ejecuta la función, es por esto que las RFC se utilizan también para llamar programas en sistemas no SAP.

Si se desea iniciar un programa externo de manera remota, se necesita una interfaz RFC fuera del sistema SAP, el cual puede ser por ejemplo un DLL simple, tomando en cuenta que cada interfaz de RFC puede ser bidireccional, la DLL podrá usar una función que se encuentre en el sistema SAP.

Para realizar el llamado de una RFC desde un sistema SAP, se necesita conocer los parámetros de entrada y salida (definidos en el entorno de construcción de funciones), y debe existir una conexión entre los 2 sistemas, la cual se denomina Conexión o Destino RFC.

Repositorio de objetos (BOR) y BAPI

Una BAPI (Business Application Programming Interface por sus siglas en inglés) es una interfaz estandarizada que facilita el acceso interno y externo a los procesos de negocio y a los datos en sistemas SAP, éstas se definen en el BOR como métodos de los objetos de negocio de SAP y permiten una vista orientada a objetos de la información del negocio en los sistemas SAP. En la siguiente figura (8), se muestra la relación del BOR con los BO y las BAPI.

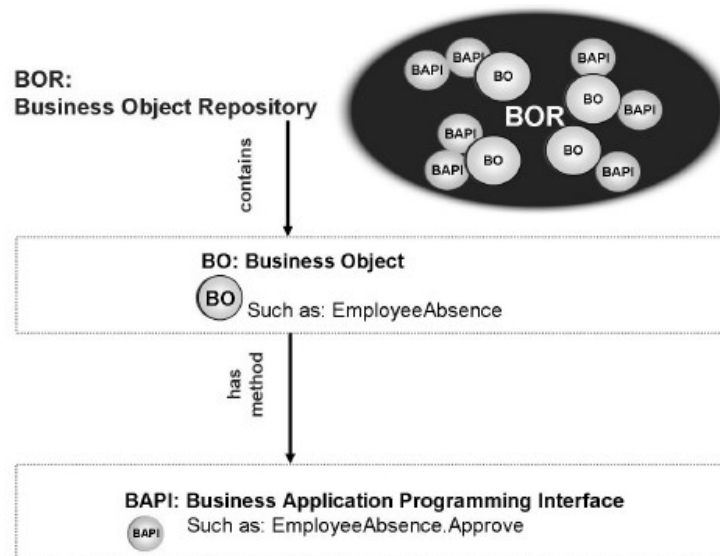


Figura8: BAPI

Las BAPIs, que representan métodos para los objetos de negocio en el sistema SAP, se usan en una gran variedad de contextos, como por ejemplo:

- Para ligar procesos de negocio entre sistemas (cuando se usa ALE por ejemplo)
- Para integrar varias soluciones en el marco de trabajo de la suite de SAP.
- Para conectar un sistema SAP a internet.
- Para trabajar con SAP workflow.
- Para realizar conexiones con programas externos.

El lenguaje ABAP

ABAP (Advanced Business Application Programming) es un lenguaje de cuarta generación, propiedad de SAP, que se utiliza para programar la mayoría de sus productos. Utiliza sentencias de OpenSQL para conectarse con prácticamente cualquier base de datos. Cuenta con un gran número de funciones para el manejo de archivos, bases de datos, fechas, etc. Permite conexiones remotas que conectan los sistemas SAP con cualquier otro sistema ya sea SAP o no.

Historia

ABAP fue desarrollado por SAP como un lenguaje de reportes para SAP R/2, en los años 80, una plataforma que permitía a las grandes corporaciones construir aplicaciones de negocios para gestión de materiales y finanzas. En sus inicios, ABAP incluyó el concepto de bases de datos lógicas, que suministraba un alto nivel de abstracción para el acceso a bases de datos.

Fue pensado como un lenguaje de programación para que los usuarios finales pudieran manipular la información, sin embargo, al ser un lenguaje de cuarta generación, se volvió complicado para los usuarios normales, por lo que fue necesario tener programadores experimentados para realizar desarrollos.

Se mantuvo como lenguaje de desarrollo para la siguiente versión que adoptó el modelo cliente – servidor y el R/3, que fue lanzada al mercado en 1992, para esta versión, prácticamente todo el sistema estaba escrito en ABAP, salvo las llamadas básicas al sistema. En 1999, con el lanzamiento de la versión 4.6 de R/3, SAP lanzó una extensión orientada a objetos denominada ABAP Objects, es decir, ABAP es un lenguaje que soporta tanto la programación estructurada, como la orientada a objetos.

Tipos de programas en ABAP

Existen principalmente 2 tipos de programas que pueden ser creados en lenguaje ABAP:

- Reportes / Ejecutables: Siguen un modelo de programación relativamente simple, en donde el usuario introduce una serie de parámetros y el programa los usa para producir un reporte de forma de lista interactiva, cabe mencionar que estos reportes no son meramente de visualización de información, sino que se tiene la bondad de poder modificar la información directamente hacia la base de datos.
- Module pool: Definen patrones más complejos de interacción con el usuario a través de una colección de pantallas, denominadas dynpros, que se refieren a la imagen que los usuarios verán en el front end. Cada una de las pantallas tiene un flujo lógico, el cual se divide en procesamiento antes de mostrar (PBO por sus siglas en inglés: Process Before Output) y el procesamiento una vez introducidas las instrucciones del usuario (PAI por sus siglas en inglés: Process After Input).

Los tipos de programas no ejecutables en ABAP, son:

- Includes
- Grupos de funciones
- Clases de objetos
- Interfaces

Flujo de procesamiento de un programa ABAP

1. El sistema carga el contexto del programa en el servidor de aplicación, este contexto tendrá las áreas de memoria para las variables y objetos, información de las pantallas para los usuarios y bloques de procesamiento ABAP. El sistema de tiempo de ejecución obtendrá toda la información necesaria para el programa del repositorio, que es una parte especial de la base de datos. Para este ejemplo de flujo de programa, se tendrá una pantalla de selección como el diálogo con el usuario, una variable y una estructura como objetos de datos y un bloque de procesamiento ABAP. La lista que se mostrará será creada dinámicamente en tiempo de ejecución.

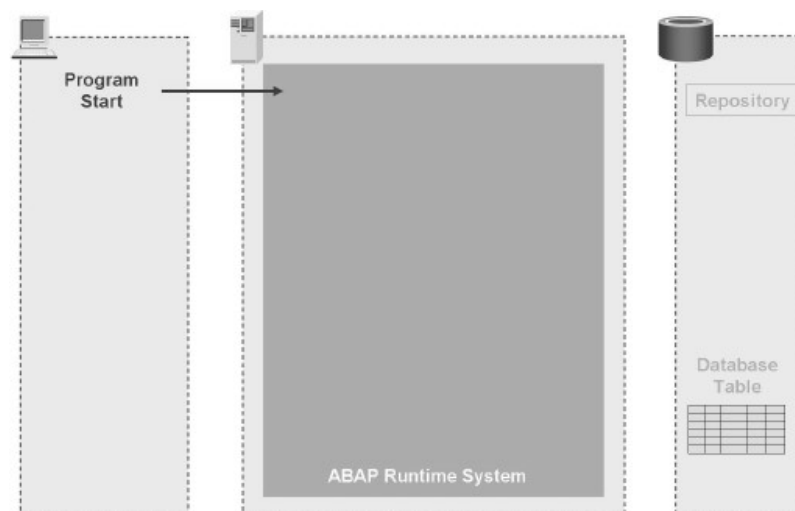


Figura 9: Flujo de programa (1)

2. Como el programa contiene una pantalla de selección, el sistema de tiempo de ejecución la envía al servidor de presentación, el cual controla el flujo mientras el usuario no haya terminado de introducir la información en los campos de entrada.

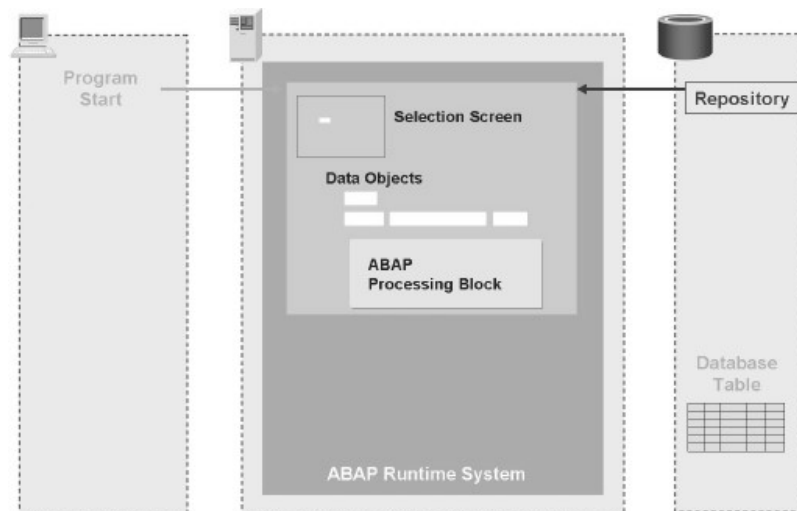


Figura 10: Flujo de programa (2)

3. Tan pronto como el usuario haya terminado de introducir los datos en la pantalla de selección, éste puede disparar el procesamiento del programa ejecutando la función correcta. Es aquí cuando la información que se introdujo se coloca automáticamente en los objetos de datos correspondientes en el programa y el entorno de tiempo de ejecución de ABAP reanuda el control del procesamiento. Si el usuario no introdujo los datos correctamente, el sistema muestra un mensaje de error automáticamente hasta que estos datos sean corregidos.

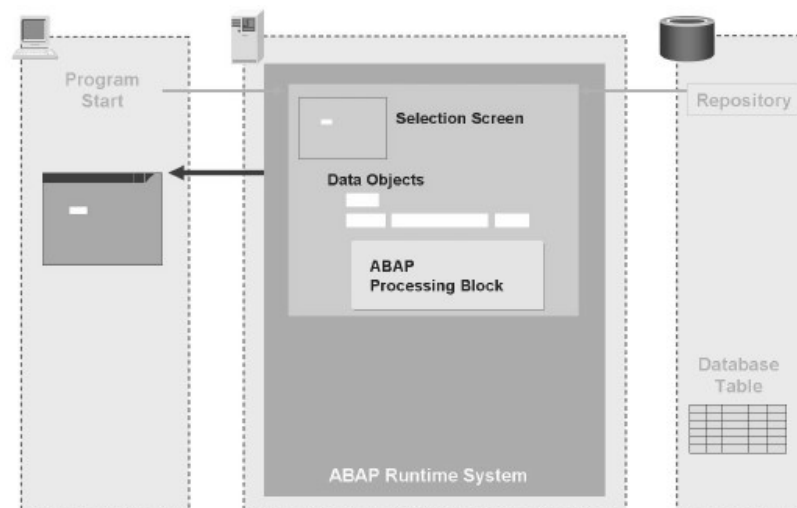


Figura 11: Flujo de programa (3)

- En este momento se hace uso del repositorio por medio de llamados a unidades reusables, por ejemplo: funciones o métodos de clases globales, los cuales se leen y se cargan al servidor de aplicación. Una vez llevada al servidor de aplicación, esta unidad reusable se ejecuta de manera síncrona.

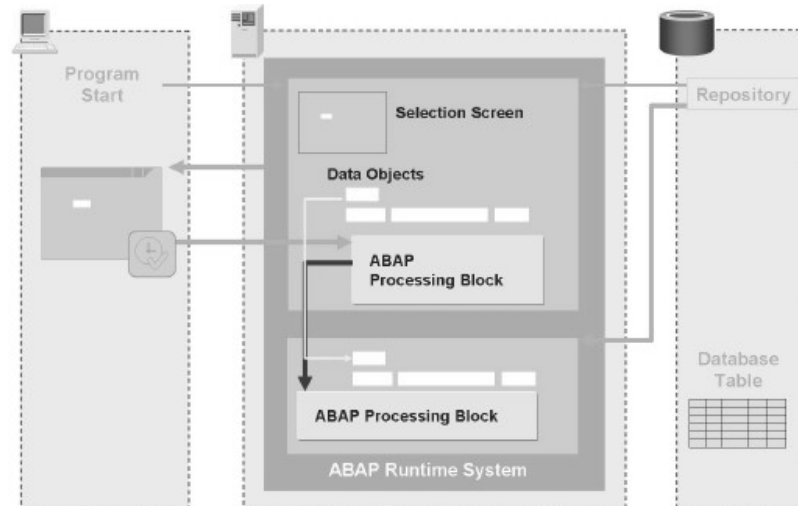


Figura 12: Flujo de programa (4)

- Se realizan los llamados y lecturas a la base de datos programados en la unidad reusable. De esta forma, la información que se requiere de la base de datos, incluyendo tabla y registros(s), se pasa a la BD por medio del manejador de la base de datos, incluido en el servidor de aplicación.

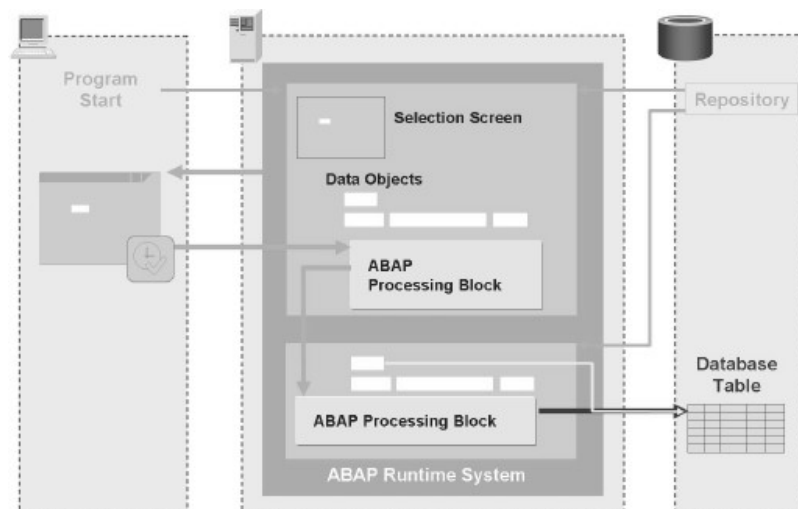


Figura 13: Flujo de programa (5)

6. Se regresa la información de la base de datos y se almacena en los objetos de datos creados para dicho propósito, de esta manera, puede ser un campo, una estructura, o una tabla. De esta manera concluye el uso de la unidad reusable y se cede el control al programa principal.

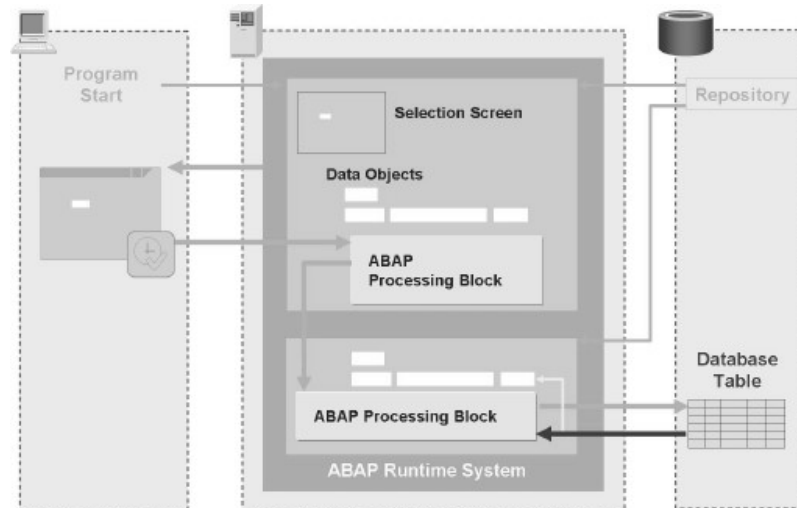


Figura 14: Flujo de programa (6)

7. Después del llamado a la unidad reusable, el bloque de procesamiento ABAP recibe las sentencias para estructurar la lista o reporte y la muestra en el servidor de presentación, es decir, al usuario final.

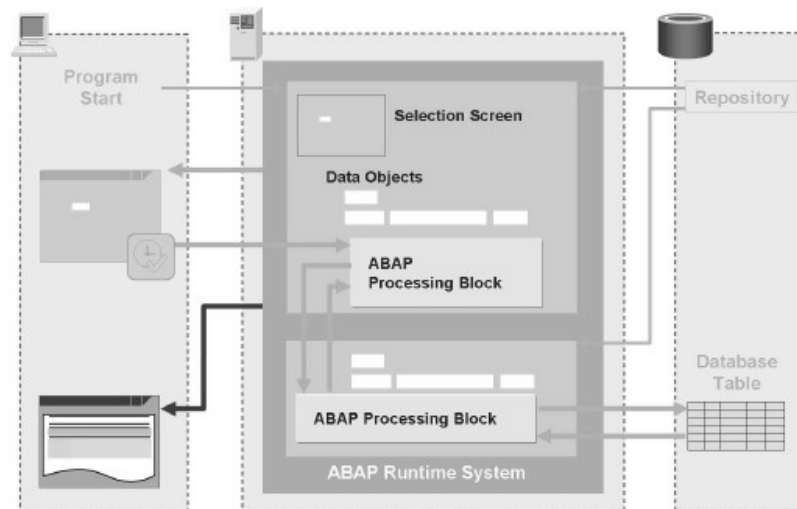


Figura 15: Flujo de programa (7)

Repositorio de información.

El repositorio consiste de todos los objetos de desarrollo del sistema: programas, módulos de funciones, definiciones de tablas de base de datos, etcétera. En este repositorio, se tienen objetos entregados por SAP así como algunos definidos por el cliente, reside en la base de datos y es siempre independiente de cliente, lo que significa que se puede acceder a un objeto de repositorio desde cualquier mandante

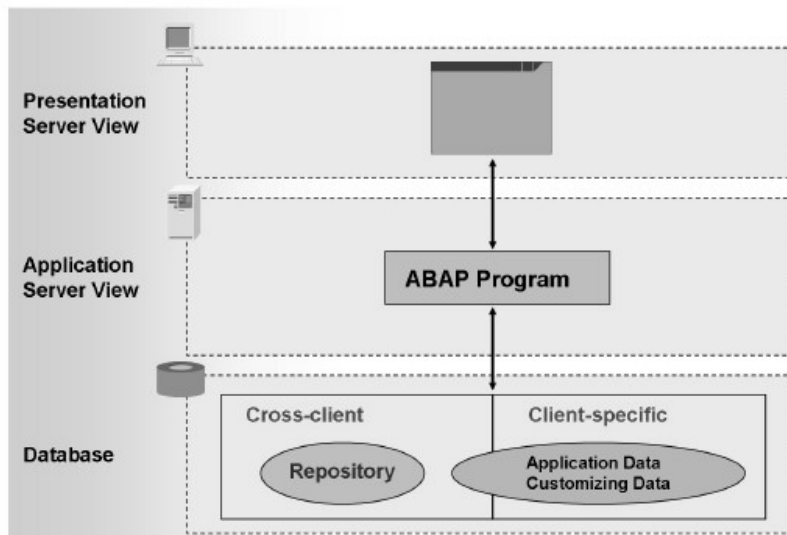


Figura 16: Repositorio de información y arquitectura R3

Como se observa en la figura 16, el repositorio se subdivide de acuerdo a los componentes de aplicación, dentro de cada componente de aplicación (por ejemplo Gestión de Materiales – MM), existen varios paquetes que contiene objetos relevantes para una mejor y más detallada subdivisión. Cada que se crea un objeto de repositorio, éste debe ser asignado a un paquete.

El navegador de objetos.

El entorno de trabajo de ABAP incluye todas las herramientas necesarias para crear y editar objetos del repositorio, las cuáles cubren el ciclo de desarrollo completo en su totalidad, algunas de estas herramientas son:

- El editor de ABAP para crear código fuente.
- El diccionario de ABAP para crear y editar definiciones de tablas, tipos de datos centrales y demás elementos.
- El editor de pantallas (screen painter) para configurar las pantallas.
- El editor de menús para diseñar las interfaces de usuario (barras de menú, barras de herramientas, barras de aplicaciones y accesos rápidos)

- El constructor de funciones para generar módulos y grupos de funciones.
- El constructor de clases para administrar las clases e interfaces globales.

Cada una de estas herramientas se pueden llamar explícitamente por medio de una transacción y después cargar un objeto del repositorio para su procesamiento, pero existe una aplicación que centraliza todas estas herramientas: El Object navigator.

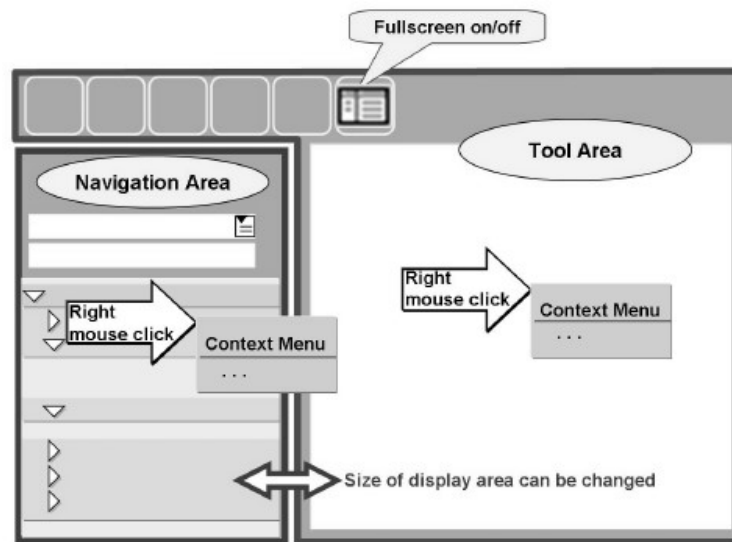


Figura 17: Panel de navegación en ABAP workbench.

Las 2 principales áreas, como se puede ver en la figura 17, son:

- El área de navegación para mostrar una lista de objetos jerárquica.
- El área de herramientas para mostrar y editar un objeto de desarrollo utilizando la herramienta apropiada.

Dentro del área de navegación podemos acceder jerárquicamente a paquetes, el diccionario de datos, clases globales, programas, grupos y módulos de funciones, y en el área de herramientas, se puede visualizar y/o editar el objeto en cuestión.

Organización de desarrollos.

Los proyectos de desarrollo se llevan a cabo en un ambiente o sistema de desarrollo (DEV), los cuales tienen objetos que son creados o editados y deben ser transportados a los ambientes subsecuentes, es decir, pruebas (QAS) y producción (PRD). Al inicio del proyecto de desarrollo, se deben crear órdenes de transporte dentro de las cuales se incluirán todos los objetos de desarrollo mencionado anteriormente en este documento.

Cuando un objeto de desarrollo se crea o se modifica, el responsable de dicho objeto, asigna el mismo a la orden de transporte, que estará lista para ser transportada al ambiente subsiguiente, tal y como se muestra en la figura 18.

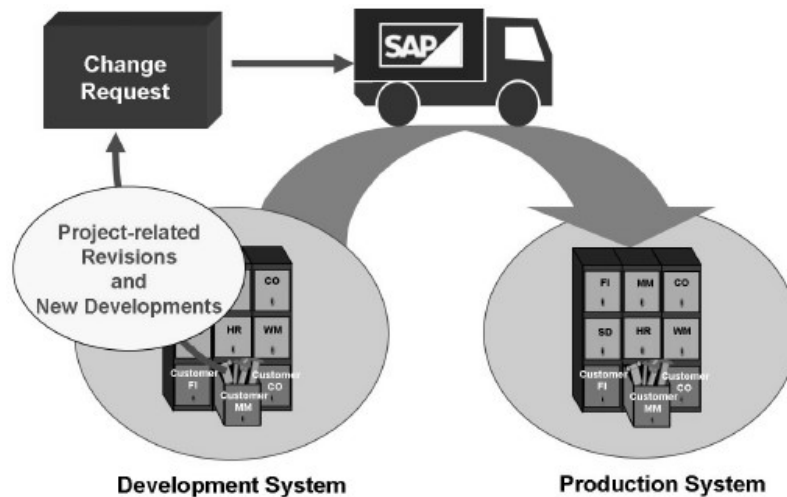


Figura 18: Sistema de transportes.

El uso de órdenes de transporte tiene las siguientes ventajas:

- Cada persona puede monitorear sus actividades específicas.
- Los respectivos objetos de desarrollo se pueden procesar por todos los empleados involucrados en el proyecto, y para aquellos desarrolladores que no pertenecen al equipo, los objetos de repositorio permanecen bloqueados hasta que se libere la orden de transporte, o se agreguen como empleados a la misma.
- El transporte en conjunto de los objetos procesados en el proyecto al momento de la liberación de la orden, se asegura automáticamente asignando los objetos a la orden, de esta manera, aunque un programa pertenezca a un paquete, puede pertenecer a diferentes proyectos.

Capítulo 3. Implementación de sistema SAP.

Las implementaciones del sistema SAP, como se ha visto anteriormente, requieren una metodología minuciosa para garantizar la satisfacción del cliente y el correcto funcionamiento del sistema de acuerdo a los casos de negocio que se van presentando a lo largo del ciclo de vida del proceso empresarial, es por esto que en este caso de estudio, se presentará la metodología usada y que se utilizó para implementar el ERP de SAP en una empresa dedicada a la construcción, cuya razón social permanecerá anónima, debido a cuestiones de confidencialidad.

Preparación del proyecto y validación del alcance.

Para estas 2 primeras fases del proyecto, se encontró que los complejos que la empresa se encontraba desarrollando en ese momento, se veían rebasados por la demanda de espacio y se tenía un cumplimiento deficiente de las normas de higiene y habitabilidad, por lo que, como parte de la estrategia de implementación, se integraría un modelo estratégico para ampliar la capacidad administrativa.

Por lo tanto, con base en los contratos que se gestionaron y obtuvieron, se limitaría a proporcionar servicios de apoyo a la administración, dentro de los que se establece brindar una plataforma tecnológica (ERP de SAP) que soporte la administración de las operaciones, el control de los servicios a proveer y el seguimiento a la resolución de los eventos que afecten los acuerdos del nivel de servicio (SLA).

Dado lo anterior, se encontraron las siguientes áreas de negocio y sus respectivos requerimientos:

- Gerencia de logística.
 - Departamento de alimentos.
 - Departamento de limpieza integral y manejo de residuos.
 - Departamento de lavandería y ropería.
 - Departamento de fumigación y control de plagas y fauna nociva.
 - Departamento de laboratorio clínico.
 - Departamento de almacenes.
 - Departamento de transporte.
- Gerencia de ingeniería.
 - Departamento de seguridad y comunicaciones.
 - Departamento de jardinería y mantenimiento de espacios exteriores.
 - Departamento de conservación de mobiliarios y equipos menores.
 - Departamento de conservación y mantenimiento de equipo industrial.
 - Departamento de tecnologías de información.

- Departamento de administración de servicios públicos, conservación y ahorro de energía.
- Gerencia de gestión.
 - Departamento de recursos humanos.
 - Departamento de contabilidad.
 - Departamento de egresos.
 - Departamento de ingresos.
 - Departamento de adquisiciones.
 - Departamento de nóminas.
- Centro de coordinación de operaciones.
 - Seguimiento a eventos y requerimientos propios de las SLA.
 - Generación de información.

Alcance funcional.

A continuación, se describen los procesos que se consideran dentro del alcance de esta iniciativa indicando, en cada uno de ellos, las actividades que se implementarán en soluciones SAP y las consideraciones que se contemplan.

Centro de coordinación operativa (CCO).

Se consideran las siguientes actividades con las siguientes sub tareas:

- Tablero de seguimiento a los acuerdos de nivel de servicio (SLA).
 - El registro de los parámetros de cada uno de los SLA acordados.
 - La construcción de los indicadores de desempeño, en relación con los programas que se establecen para cada servicio relevante para los SLA.
 - El establecimiento de la comunicación entre los diversos eventos considerados como alertas y los indicadores de desempeño.
 - La construcción del tablero de seguimiento a SLA que comprenderá los indicadores de desempeño antes mencionados.
- Modelo de control de gestión.
 - Registro de los eventos relevantes para la medición de SLA que será implementado en SAP CRM.
 - El tablero de seguimiento antes descrito.
 - La facilidad de modificar los indicadores de desempeño o los parámetros de los niveles de servicio acordados, de conformidad con los acuerdos que entre las partes responsables se vayan realizando durante la operación de los complejos.
- Seguimiento a eventos.

- Registro de los eventos y alarmas en SAP CRM por medio de SAP BPM, tanto aquellos generados automáticamente por los sistemas de control que son relevantes para la operación de la constructora, como aquellos que se registren manualmente.
- La notificación de eventos y alarmas a las unidades responsables de cada servicio.
- El registro de los avances en la resolución de eventos y alarmas.
- El registro del cierre de los eventos y alarmas resueltos.
- Cuestiones de seguridad y confidencialidad de la información.
 - El acceso al tablero de seguimiento a SLA y SAP CRM para el registro manual de eventos se limitará únicamente a los usuarios que cumplan con el perfil de autorización necesario.
 - Únicamente el personal de la constructora, podrá acceder al tablero de control en el que se miden los SLA internos.
 - El usuario que dé de alta un evento podrá acceder sólo al evento que le corresponde.
 - El CCO determinará los usuarios y los perfiles de autorización para los tableros de control.
- Reportes.
 - Reportes del nivel de servicio otorgado.
 - Reportes de manejo de materiales.
 - Reportes de operación.
- Cálculo de deducciones.
 - Programación de las fórmulas de deducciones propias de cada servicio con SLA.
 - Cálculo de deducciones al cierre del periodo.

Componentes SAP considerados.

Componente	Módulo	Sub módulo	Función
SAP CRM	-	-	Registro de eventos relevantes para los SLA
SAP NW-PI	-	-	Integración de procesos
SAP BPM	-	-	Integración de alarmas y eventos con CRM
SAP ERP	-	-	Sistema de información
SAP MII	-	-	Integración con dispositivos no SAP

Tabla 1: Componentes considerados para CCO.

Coordinación de operaciones.

La coordinación de operaciones está conformada por las gerencias de logística y de servicios de ingeniería, mismas que consideran los siguientes departamentos:

- Gerencia de logística.
 - Departamento de alimentos.
 - Departamento de limpieza integral y manejo de residuos.
 - Departamento de lavandería y ropería.
 - Departamento de fumigación y control de plagas y fauna nociva.
 - Departamento de laboratorio clínico.
 - Departamento de almacenes.
 - Departamento de transportes.
- Gerencia de ingeniería.
 - Departamento de seguridad y comunicaciones.
 - Departamento de jardinería y mantenimiento de espacios exteriores.
 - Departamento de conservación de mobiliarios y equipos menores.
 - Departamento de conservación y mantenimiento de equipo industrial.
 - Departamento de tecnologías de información.
 - Departamento de administración de servicios públicos, conservación y ahorro de energía.

Componentes SAP considerados.

Componente	Módulo	Sub módulo	Función
SAP ERP	MM	CBP	Planificación de requerimientos basada en consumo
		PUR	Compras de bienes y servicios
		SRV	Administración de servicios
		IM	Administración de inventarios
	QM	-	Control de calidad
	PM	-	Mantenimiento y reparación
	AA	-	Administración de activos fijos
	FI	GL	Contabilidad general
		AP	Cuentas por pagar
	CO	-	Costos
	HCM	-	Recursos humanos
	CRM	-	Registro de eventos y alarmas

Tabla 2: Componentes considerados para coordinación de operaciones.

A continuación, se describen los procesos que pertenecen a cada uno de los departamentos que integran las gerencias de logística y tecnologías de la información.

GL - Departamento de alimentos.

- Proceso de contratación del servicio de alimentos a un tercero calificado.
- Proceso de preparación y mantenimiento de las instalaciones de cocina y comedor.
- Proceso de planificación de dietas y adquisición de ingredientes para su preparación.
- Proceso de preparación de dietas.
- Proceso del servicio de comedor.
- Proceso de comidas no programadas.
- Clasificación del servicio.

GL - Departamento de limpieza integral y manejo de residuos.

- Proceso de contratación del servicio de limpieza a un tercero calificado.
- Proceso de preparación del servicio.
- Servicio de limpieza integral y manejo de residuos.
- Calificación del servicio.

GL - Departamento de lavandería y ropería.

- Proceso de contratación del servicio de lavandería a un tercero calificado.
- Proceso de provisión de ropa.
- Proceso de asignación de ropa.
- Proceso de preparación y mantenimiento de las instalaciones de lavandería.
- Proceso de adquisición de insumos de lavandería.
- Proceso de mantenimiento de prendas.
- Calificación del servicio.

GL - Departamento de laboratorio clínico.

- Proceso de contratación de personal calificado.
- Proceso de preparación de las instalaciones del laboratorio clínico.
- Proceso de adquisición de insumos de laboratorio.
- Proceso de registro del historial clínico.

GL - Departamento de almacenes.

- Proceso de administración de bienes.
- Proceso de taller de reparación.

GL - Departamento de transportes.

- Proceso de adquisición de vehículos.
- Proceso de asignación y uso de vehículos.
- Proceso de mantenimiento y reparación de vehículos.
- Proceso de taller de reparación de vehículos.

- Proceso de validación de consumo de combustible y desgaste.

GSI - Departamento de jardinería y mantenimiento de espacios exteriores.

- Proceso de contratación del servicio a un tercero calificado.
- Proceso de preparación para el servicio.
- Servicio de jardinería y mantenimiento de espacios exteriores.
- Calificación del servicio.

GSI - Departamento de conservación de mobiliario y equipos menores.

- Proceso de preparación.
- Proceso de adquisición de materiales para la gestión del mantenimiento.
- Proceso de ejecución del servicio.
- Calificación del servicio.

GSI - Departamento de conservación y mantenimiento de equipo industrial.

- Proceso de preparación.
- Proceso de adquisición de materiales para la gestión del mantenimiento.
- Proceso de ejecución del servicio.
- Calificación del servicio.

GSI - Departamento de tecnología de la información.

- Proceso de preparación.
- Proceso de adquisición de materiales para la gestión del mantenimiento.
- Proceso de ejecución del servicio.
- Calificación del servicio.

Coordinación de gestión.

Esta coordinación está conformada por los departamentos administrativos. El alcance para los procesos involucrados se describe a continuación y será soportado por los siguientes componentes:

Componentes SAP considerados

Componente	Módulo	Sub módulo	Función
SAP ERP	FI	GL	Contabilidad general
		AP	Cuentas por pagar
		AR	Cuentas por cobrar
	MM	PUR	Compras de bienes y servicios
		SRV	Administración de servicios
		IM	Administración de inventarios
	AA	-	Administración de activos fijos
	CO	-	Costos
	HCM	-	Recursos humanos
		PY	Nómina
SAP CRM	-	-	Registro de eventos y alarmas

Tabla 3: Componentes considerados para coordinación de gestión.

A continuación, se listan los procesos por departamentos.

Departamento de contabilidad.

- Estructura organizacional.
- Proceso de la contabilidad general.
- Procesos de la contabilidad fiscal.

Departamento de egresos.

- Procesos del control presupuestal de egresos.
- Procesos financieros.
- Procesos de cuentas por pagar.

Departamento de ingresos.

- Procesos del control presupuestal de ingresos.
- Procesos de facturación por servicios prestados
- Procesos de cuentas por cobrar.

Departamento de adquisiciones.

- Estructura organizacional.
- Procesos de adquisición de bienes y servicios.

Departamento de recursos humanos.

- Estructura orgánica.
- Procesos de selección y reclutamiento.
- Procesos relativos al personal.
- Procesos en la administración del tiempo.

- Procesos administrativos.

Departamento de nóminas.

- Procesos administrativos.

Alcance técnico

Como se detalló en la sección anterior, el alcance funcional comprende una serie de procesos los cuáles deben ser cubiertos ya sea por configuraciones y adecuaciones del ERP SAP en conjunto con los componentes que actuarán en conjunto para satisfacer todos y cada uno de ellos. A continuación, se muestra una breve reseña de lo que es cada sistema mencionado, y el rol que desempeña en la solución tecnológica propuesta para la empresa.

ERP (Enterprise Resource Planning)

Este componente incorpora las funciones y procesos de negocio claves de una organización, como son operaciones, finanzas y recursos humanos, los 3 principales macro procesos que definen un proceso de negocios, y que en la arquitectura del ERP, se dividen de la manera que se muestra en la figura 19:

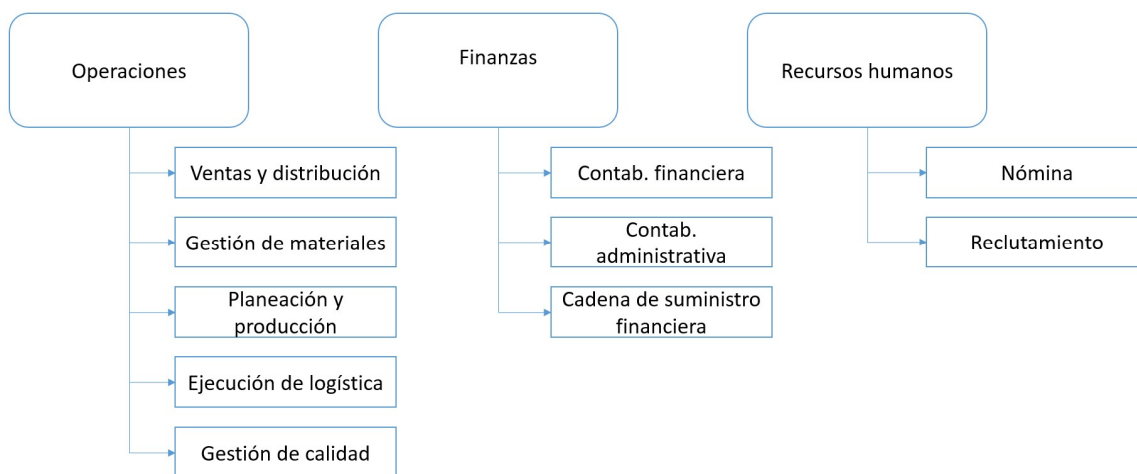


Figura 19: Módulos y sub módulos del ERP implementados.

PI (Process Integration)

SAP Netweaver Process Integration, es el software de integración de aplicaciones empresariales, un componente de Netweaver que se utiliza para facilitar el intercambio de información entre los diversos sistemas de una empresa, se puede ver en la figura 20 que centraliza las comunicaciones entre diversos sistemas, ya sea con software interno o sistemas de proveedores, es decir, SAP PI es compatible con otros sistemas SAP y con productos de software desarrollados por otras compañías.

SAP ha denominado a PI como un agente de integración, porque sirve como intermediario entre diversas entidades con varios requerimientos en términos de conectividad, formato y protocolos.

Una de las principales características, es el Single-Sign-On, que permite que un usuario del sistema, no tenga que estar introduciendo credenciales para firmarse dentro de cada uno de los sistemas que integran la solución empresarial.

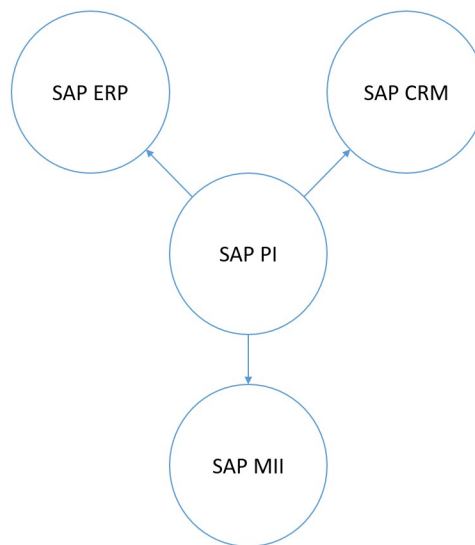


Figura 20: PI como punto central de comunicaciones.

MII (Manufacturing Integration and Intelligence)

Esta aplicación de SAP permite procesos adaptativos de manufactura mediante la sincronización de operaciones de manufactura con sistemas back end y de negocios. Esta vista unificada de manufactura, funciona de 2 diferentes maneras:

- Inteligencia de manufactura: Un motor de análisis en tiempo real que totaliza y entrega visualización de eventos, alertas, KPI y ayuda a las decisiones con base en tableros basados en roles.
- Integración de manufactura: Una sola aplicación compatible con diversos estándares que integra la conectividad del ERP con aplicaciones de la planta en tiempo real, ya sea una sola planta o varias, con una vista normalizada del desempeño a lo largo de los sistemas que integra.

Las alertas disparadas por eventos en SAP MII, permite reacciones rápidas ante la identificación y resolución de problemas, Proporciona respuestas basadas en workflow, dentro de los cuales, se puede configurar para ciertas alertas.

CRM (Customer Relationship Management)

Es la herramienta de SAP que permite gestionar las relaciones y canales de comunicación con sus clientes, brindando elementos e información para la generación de reportes interactivos. Esta solución, incluye varios módulos que respaldan áreas funcionales claves como:

- Ventas
- Mercadeo
- Servicios
- Analíticos

BPM (Business Process Management)

Es un componente de Netweaver que se usa para el modelado y ejecución de los procesos de sistema y usuario, lo cual se logra por medio de la visualización o esquematización basada en procesos.

BPM ve el negocio desde una perspectiva de procesos que atraviesan los silos. Proporciona herramientas que permiten a la compañía modelar los procesos de principio a fin para un mejor análisis y gobernabilidad, promoviendo la consistencia con políticas y mejores prácticas, así como el fomento de la innovación por medio de la visibilidad lógica de los negocios.

En resumen, el alcance técnico, comprende la instalación y puesta en marcha de los componentes antes mencionados, así como su integración por medio de SAP PI, obteniendo de esta forma un sistema robusto, que soportará todos los procesos y actuará de forma eficiente para cubrir las necesidades y requerimientos de la empresa.

Para cubrir dichas necesidades, se puede ajustar el sistema por medio de configuraciones de los procesos que ya cubre el ERP, sin embargo, existen otros requerimientos puntuales y muy específicos, que tuvieron que ser cubiertos por medio de mejoras al sistema y desarrollos específicos del cliente

Lo anterior, se realizó con el lenguaje ABAP y sus diferentes herramientas que fueron listadas en el capítulo 2 de este documento.

Realización

Como objeto de estudio del presente documento, me enfocaré en explicar la realización técnica, pues es este apartado en donde se utiliza el lenguaje ABAP y las habilidades e consultoría técnica que son el principal tópico a tratar.

Una vez obtenido el alcance funcional, podemos ver que está dividido en procesos de muchos tipos que son los que cubrirán las necesidades del negocio, sin embargo, hay muchas de ellas que tras desglosar los procesos de una manera más detallada, se encontró que no se tiene manera de cubrirlos por medio de configuraciones estándar del ERP, es por esto que se tienen que identificar dichas necesidades en conjunto con el equipo de implementación funcional y tener una visibilidad más amplia de lo que será el conjunto de desarrollos necesarios.

Tipos de desarrollos.

Se tiene ya una plantilla de desarrollos necesarios dentro de un proyecto, conocido como RICEFW, acrónimo en inglés para Reports, Interface, Conversion, Enhancements, Forms and Workflow, que comprende los siguientes tipos de desarrollo:

Reportes

Un reporte en SAP es un programa ejecutable que obtiene información de la base de datos y genera una salida basada en filtros y criterios de selección proporcionados por el usuario. Este tipo de programas casi nunca hace actualizaciones a la base de datos. SAP cuenta ya con un conjunto de reportes estándares que muestran información típica de los procesos de operación, sin embargo, existen necesidades que no son cubiertas, por lo cual debe existir un desarrollo.

Interfaces.

En algunas organizaciones, algunos de los procesos como lo que se han visto hasta ahora, se ejecutan o administran en sistemas externos, ya sean SAP o no SAP, y dentro del ERP o algún componente de SAP como CRM, necesitamos tener esa información a la mano o enviarla para su procesamiento. Esto se hace típicamente por medio de IDocs, aunque SAP también soporta la ejecución de servicios web, o RFC. El encargado principal de realizar estos intercambios de información, es el componente SAP PI.

Programas de conversión.

Estos programas se necesitan típicamente cuando la información debe homologarse entre diferentes sistemas. De esta manera, cuando se implementa SAP, se reemplazarán algunos o todos los sistemas legados, pero la información deberá permanecer, así que se extrae esta información de los sistemas legados y se pone en algunos archivos para así cargarla al sistema SAP por medio de programas destinados a este fin y de este modo, se evitan errores de carga en la fase de go-live.

Mejoras.

Las mejoras son programas por medio de los cuales se agregarán o modificarán funcionalidades existentes a las aplicaciones de negocio estándares de SAP, es decir, la funcionalidad estándar no cubre el requerimiento en su totalidad, o no contiene toda la información que se requiere para cierto proceso, es así que por medio de mejoras como son user-exits, customer-exits o BADIs, se puede acceder al código del sistema y modificarlo para adecuarlo completamente a lo que requerimos.

Formularios.

Los formularios, son impresiones que contienen información extraída de SAP, como órdenes de compras, facturas o certificados de impresión de lotes, por mencionar algunos. Por lo general, los formularios que proporciona SAP siempre deben adecuarse a las necesidades del cliente o empresa, dado que hay datos que se requieren y no están incluidos en él, además de logos de impresión o códigos de barras.

Workflow.

Son flujos de trabajo, es decir, una secuencia de actividades conectadas que da como resultado el intercambio y propagación de información.

En todos y cada uno de los anteriores tipos de desarrollo, es fundamental que el equipo funcional especifique a los desarrolladores las características de los requerimientos, como por ejemplo:

- Las columnas que debe contener cada reporte.
- El mapeo de los campos que son susceptibles de conversiones.
- La lista de puntos a mejorar del sistema estándar.
- La información que debe estar contenida en los formularios.
- Los diagramas de flujo de información.

Programas de carga.

Adicionalmente y como primeros desarrollos se tienen las cargas de datos maestros, este tipo de desarrollos, tienen 2 formas de realizarse:

- LSMW: Legacy System Migration Workbench por sus siglas en inglés. En los grandes proyectos, existe incluso un equipo dedicado solamente a la carga de datos maestros, sin embargo, en algunas ocasiones, es responsabilidad del equipo funcional, cargar esta información, tal como datos de clientes, proveedores, activos fijos, etc. LSMW es una herramienta estándar de SAP que paso a paso toma la información que deberá cargarse de archivos fuentes y la introduce en el sistema con llenado de estructuras propias de cada dato maestro.

- Programa personalizado: Cuando las cargas de datos por LSMW no son suficientes y se deben de tener ciertas consideraciones como rutinas de conversión o validaciones de datos, se procede a crear un programa de carga en el cual se puede obtener la información de archivos de texto, o en formato Excel, y se carga por medio de una BAPI en el sistema. Se pueden utilizar también cuando hay que modificar o actualizar la información que ya está cargada en el sistema.

Estrategia de desarrollo.

Al ser un proyecto de implementación del sistema, la cantidad de desarrollos fue muy grande, como lo pudimos observar en el alcance funcional, los procesos que fueron cubiertos eran demasiados, por lo que, el equipo funcional identificó todas y cada una de las necesidades que existían, y el equipo de desarrollo del cual yo fui el encargado, dividió los desarrollos por módulo funcional y por tipo de desarrollo, llegando a cerca de 40 desarrollos, distribuidos como se muestra en la tabla 4, esto, sin contar 80 programas de carga de datos, debido a que en su mayoría, se realizaron por medio de la herramienta LSMW.

Módulo	Desarrollos solicitados
BPM	7
CRM	13
FI	7
MM	7
QM	1
RH	1

Tabla 4: Desarrollos por módulo.

Para los desarrollos restantes, se asignaron de acuerdo a los perfiles de los miembros del equipo, dado que no todos contábamos con los mismos conocimientos ni habilidades, es decir, los desarrollos con mayor grado de complejidad, se asignaron a los consultores con mayor experiencia, esto con el fin de optimizar tiempos de entrega y minimizar riesgos en cuestión de retrasos.

Para la etapa de desarrollo, la creación de código se fue realizando de acuerdo a la importancia de cada requerimiento conforme las etapas del proyecto, no así su importancia en la operación, es decir, se realizaron primero aquellos desarrollos que creaban las bases y los cimientos de los diferentes componentes, para así dar paso a la creación del código fuente de aquellos desarrollos más complejos y que en algunos casos, tenían dependencias de otros sencillos (por llamarlos de alguna manera) y sin ellos, les era imposible operar.

Dividiendo por tipo de requerimiento e importancia, la estrategia de desarrollo quedó de la siguiente manera:

Prioridad	Tipo de desarrollo
1	Carga de datos
2	Interfaces
3	Flujos de procesos
4	Reportes
5	Formulario

Tabla 5: Tipos de desarrollos

Como se puede ver en la tabla 5, los desarrollos de carga de información son los más importantes, no así los más complejos, pues todos los demás desarrollos, dependen de una etapa de carga de datos exitosa.

En segundo y tercer lugar, tenemos las interfaces y flujos de procesos, pues éstos representan la comunicación entre los sistemas y la principal vertiente de información. Debemos recordar que este proyecto contó con varios componentes de SAP, así, existieron varios canales de comunicación, principalmente entre los componentes ERP y CRM, PI, al ser el principal integrador, funciona a manera de servidor, en donde se procesan peticiones.

Algunos ejemplos de interfaces que se realizaron son:

- Homologación de tablas personalizadas en la base de datos.
- Sincronización de datos maestros, tales como clientes.
- Comunicación entre CRM y módulos de ERP como MM y PP.
- Envío de información de facturación electrónica entre MM y sistema legado.

Por último, pero no menos importante, se tienen los reportes y los formularios, pues esta es la parte del sistema que se presentará al usuario, sin embargo, no se puede desarrollar antes, puesto que existen dependencias con los anteriores desarrollos, sin datos maestros ni comunicación entre los módulos y componentes, no se puede llegar al producto final que es lo que se presenta al cliente.

Seguimiento de avances.

Para el seguimiento de avances por parte del equipo de desarrollo, se adoptó la metodología SCRUM, sin embargo, se adaptó de acuerdo a las necesidades del equipo y el proyecto, a continuación la metodología utilizada:

- Se designó un SCRUM máster, quien era el encargado de dirigir las sesiones de revisión de avances.
- Un secretario que llevaba las anotaciones que surgían de la sesión.

- Se definieron periodos de metas que iban de 3 a 4 semanas dependiendo la carga de trabajo y la complejidad de los desarrollos que se consideraban para ese periodo.
- Todos los días, a primera hora se realizaba una sesión no mayor a 20 minutos, en donde se trataban 3 puntos principales con cada miembro del equipo de desarrollo:
 - Estatus actual de los requerimientos.
 - Actividades planeadas para el día.
 - En caso de existir retrasos, explicar el motivo y lo que se requería para solucionar el mismo.

De esta manera, se garantizaba un completo control sobre el estatus de los desarrollos y sobre las acciones que debían tomarse para no generar ningún retraso. Del mismo modo, se podía generar una secuencia de desarrollo por programador, y detectar posibles puntos débiles y posibles reasignaciones de requerimientos.

La manera de llevar control de las actividades antes descritas fue por medio de una pizarra con notas adhesivas que presentaba la estructura que se muestra en la figura 21.

TABLERO PRINCIPAL				
Nombre	Actividades asignadas	En proceso	Terminados	Stoppers
Desarrollador 1	NOTA NOTA NOTA NOTA NOTA	NOTA NOTA NOTA	NOTA NOTA NOTA NOTA NOTA	NOTA NOTA
Desarrollador 2	NOTA NOTA NOTA NOTA NOTA	NOTA NOTA	NOTA NOTA NOTA NOTA	NOTA
Desarrollador m	NOTA NOTA NOTA NOTA	NOTA NOTA NOTA NOTA	NOTA	
Desarrollador n	NOTA NOTA NOTA NOTA NOTA	NOTA	NOTA NOTA NOTA	

Figura 21: Tablero SCRUM

Se utilizó código de colores para las notas adhesivas, para hacer más interactiva la metodología SCRUM, así, existían 4 columnas principales además del nombre de los desarrolladores:

- Actividades asignadas: Las notas pertenecientes a esta columna, son aquellos requerimientos que se asignaron a cada desarrollador cada iteración de la

metodología, éstas eran determinadas por los arquitectos de solución y el líder de desarrollo.

- En proceso: Todos aquellos requerimientos que se encontraban en etapa de desarrollo.
- Terminados: Los requerimientos que se encontraban en esta columna, son aquellos que ya habían pasado por un ciclo de pruebas unitarias por parte del consultor técnico y funcional.
- Stoppers: Aquellas actividades o puntos que impedían al desarrollador continuar con alguna tarea, estos debían ser atendidos lo antes posible.

Preparación final.

El propósito de esta tarea es planear las actividades previas al arranque en la correcta secuencia, para garantizar que todo se encuentre completo y que el personal necesario esté disponible cuando se requiera. Las actividades de preparación final, o plan de cutover cubre la instalación e inicialización del ambiente de producción, y debe cubrir la información de aplicación y operación, tal como:

- Datos maestros.
- Información transaccional.
- Objetos propios de cliente y repositorio del ERP.

Procedimiento.

Crear plan de cutover.

Preparar el plan para migrar el sistema y la organización hacia el ambiente de producción, es lo que se denomina plan de cutover. Este plan debe concentrarse en las actividades, tareas y sincronización de los últimos días del proyecto, cuyo principal beneficio es asegurar una transición óptima hacia el ambiente de producción. Hacer referencia a este plan en caso de existir dificultades, sirve como guía de acción, puesto que incluye una lista de verificación que revisa los puntos de preparación y proporciona las bases de aprobación para el progreso de las actividades.

La planeación del cutover debe cubrir los procedimientos para cerrar los sistemas legados e introducir la información en el nuevo sistema, además, se debe incluir una pausa en las actividades entre el inicio de operaciones y la puesta a punto del sistema de producción, porque puede necesitarse para resolver los problemas de último minuto, y debe durar entre 12 y 24 hrs.

El plan de cutover debe incluir:

- Conversiones de información.
- Tiempos estimados de cuándo serán realizadas las conversiones.
- Líderes de equipo para el cutover, tanto funcional como técnico.

- Roles y responsabilidades del núcleo del equipo de proyecto, usuarios clave, usuarios, etc.
- Asignaciones del equipo y horarios de trabajo y disponibilidad.
- Involucrar a la administración de la compañía y designar quiénes tomarán decisiones.
- Procedimientos para apagar los sistemas legados.
- Procesos de reconciliación para asegurar que las transacciones de la compañía sean introducidas en los nuevos sistemas.
- Procesos de reconciliación para asegurar que la información sea convertida a los formatos del nuevo sistema.

El plan de cutover debe ser revisado y aprobado por el líder de proyecto, líderes de equipo y por parte de la administración de la compañía.

[Crear una lista de comprobación final.](#)

Esta lista debe revisar los puntos de preparación y proporcionar los indicadores de progreso.

[Crear un plan de contingencia.](#)

El plan de cutover debe incluir un plan de contingencia para retrasos, el cual debe especificar cuanto tiempo (en horas y/o días) puede estar detenida la instalación del nuevo sistema de producción, y el tiempo en que los sistemas legados pueden ser reestablecidos. Es una práctica estándar, tener un punto de no retorno para hacer operativo a un sistema, es decir, después de un cierto número de horas o días, no podría ser posible regresar la operación a los sistemas legados.

[Determinar tiempos y horarios de conversiones.](#)

Determinar el tiempo y horario de la conversión y transferencia final de información, pretende estimar cuánto tiempo puede tomar cada conversión, tanto de datos maestros como de información transaccional, incluyendo la ejecución de programas de conversión y tiempos de conversión manual, en caso de existir. También se debe determinar quién y cuándo reconcilia la información, así como el tiempo requerido para re ejecutar un programa en caso de un fallo.

[Probar la operación en el nuevo sistema.](#)

El paso final del proceso de conversión y preparación del sistema es probar la operación en el sistema de producción, asegurar que las transacciones están funcionando adecuadamente y que los usuarios tienen acceso al sistema de manera adecuada.

[Plan de cutover para el proyecto.](#)

Por el tipo de implementación que se llevó a cabo, en donde no hubo usuarios involucrados en el proceso de definición, se diseñó un plan de cutover, que se muestra en la tabla 6, de acuerdo a las necesidades identificadas, de esta manera, se presenta a continuación el plan con el que se pusieron a punto las funcionalidades técnicas en los sistemas. Recordemos

que no existieron sistemas legados de donde extraer información, y por lo tanto, la fase de conversión de datos no existe.

Secuencia	Actividad	Equipo responsable
1	Instalación de instancias para ambiente PRD	Basis
	Actualización de sistemas e instalación de parches para homologar con ambiente QAS	
2	Paso de transportes de ambiente de QAS a PRD	Basis
3	Validación de transportes de configuración y desarrollo	
	Verificar que las configuraciones del sistema estándar, se encuentren listas en el nuevo ambiente	Funcional
	Verifica que los elementos de diccionario y programas se encuentren en PRD.	Técnico
	Verificar conexiones entre los diferentes componentes	Basis
4	Carga de datos de tablas de cliente	
	Carga manual de datos de cliente	Funcional, técnico
	Ejecución de programas de carga para datos de cliente	Funcional
	Ejecución de interfaces para homologación de información entre los distintos componentes	Técnico
5	Carga de datos maestros	
	Verificación de archivos y formatos de carga	Funcional
	Ejecución de LSMW	Funcional
	Ejecución de programas de carga	Funcional
	Carga de logos e imágenes	Técnico
6	Verificación de roles y perfiles	
	Se prueba el acceso o restricción a transacciones	Basis

Tabla 6: Plan de cutover.

Las ventajas de tener una implementación a una nueva empresa, con respecto al plan de instalación, son las siguientes:

- El plan de instalación se fue llevando de manera gradual, es decir, no se tuvieron pausas en la operación para poder poner a punto los sistemas.
- Las conexiones entre los componentes se fueron realizando de manera que fueran estables al final del ciclo de pruebas.
- El paso de transportes se llevó a cabo en un periodo de una semana aproximadamente, garantizando que las configuraciones y desarrollos quedaran correctamente instalados en el nuevo ambiente.

- La carga de datos maestros se realizó de manera gradual, sin el riesgo de que los programas dejaran de funcionar inesperadamente, pues en caso de que esto sucediera, se tenía tiempo suficiente para corregir los problemas y realizar la carga de manera satisfactoria.

Así, el plan de instalación se realizó sin mayor contratiempo pues al ser un plan de instalación que se ejecuta gradualmente, todos los riesgos se reducen o minimizan por la ventana de tiempo que se tiene para una instalación exitosa.

Al final de la ejecución de dicho plan, se contó con un sistema limpio y puesto a punto con datos maestros cargados, componentes en línea y conectados, y roles y perfiles funcionando correctamente.

Riesgos.

Como se mencionó anteriormente, los riesgos quedan minimizados por la gran ventana de tiempo con la que se contó, sin embargo, minimizar no significa eliminar, y entre los riesgos que se identificaron, están los siguientes:

- Roles y perfiles. Al crear los roles y perfiles sin la intervención de los usuarios, la matriz de roles queda definida teóricamente, y cuando la operación se pone en marcha, pueden existir requerimientos y necesidades distintas tanto para accesos como para restricciones, así que los roles y perfiles, deben ser modificados en cuanto se notifiquen los cambios.
- Formato de reportes. Puede ocurrir que al iniciar la operación, algún o algunos reportes creados, carezcan de información, o presenten más de lo necesario, es por esto que se debe tener en consideración, el desarrollo de reportes dinámicos en donde la disposición de columnas, sea completamente dinámica.

Go-live y soporte.

La etapa más importante, pues aquí se pone en operación todo lo realizado durante la implementación, en este punto del proyecto, debe haberse ejecutado ya el plan de cutover exitosamente, puesto que la compañía se encontrará ya en operación.

Para este particular caso, se tomó la estrategia de la figura 22:

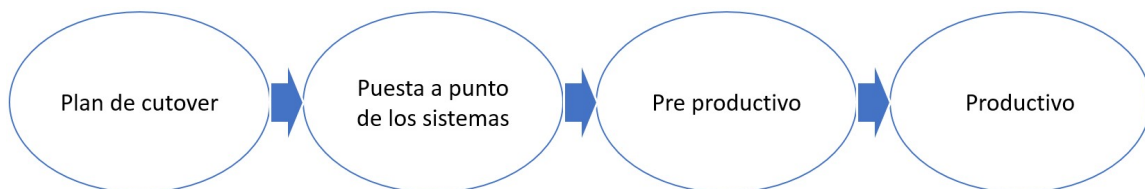


Figura 22: Plan de go- live.

Las etapas de instalación de cutover se detallaron en la sección anterior, sin embargo, podemos ver que se incluyó una etapa de pre productivo, con el objetivo de que los usuarios se adapten a la nueva operación, así como una estabilización tanto en el sistema como en la operación. Este periodo de estabilización, se refiere a comenzar a operar de manera normal, en un ambiente similar al de QAS, que posteriormente se tomará como base para crear el nuevo ambiente de PRD, es decir, la información transaccional se tomará de este ambiente de pre PRD, dado que esta operación se llevó a cabo durante un par de meses.

Así, al final de este periodo de pre producción, se contaba con un sistema completamente estabilizado y poblado ya en su totalidad con datos maestros e información transaccional proveniente de la operación diaria.

Al final de este periodo de pre producción, se realiza una copia de mandante para tener el ambiente destinado a PRD en el servidor asignado para tal tarea, y es aquí en donde se da la ventana de mantenimiento de 12 a 24 horas para realizar las copias y pasar la operación a los sistemas que quedarán operando.

Soporte

Esta etapa del proyecto está dada en su totalidad por las condiciones contractuales que se definieron al principio del proyecto.

Dada la estructura y estrategias que se han mencionado a lo largo de este documento, la etapa de soporte, quedó definida junto con la etapa de pre producción, en donde se atendieron las necesidades y requerimientos finales en sitio.

Por las condiciones del proyecto, se tomó un equipo de consultoría base para atender y afinar los requerimientos y dar soporte a la operación en sitio, siguiendo estos pasos:

- Se crea equipo base de consultoría.
- Se traslada a sitio, es decir, se trabaja desde las oficinas en donde se lleva a cabo la operación, en caso de ser necesario, el equipo de consultoría se divide geográficamente.
- Los consultores estarán verificando que la operación se lleve a cabo de manera adecuada, en caso de existir algún error, o modificación por solicitud del cliente, se deberá corregir.
- Es responsabilidad del consultor, garantizar la mejor estrategia e implementación del sistema, y en caso de encontrar un punto de mejora, se deberá implementar una solución adecuada y que cumpla con las necesidades de la compañía.

De esta manera, se garantiza una operación fluida y con el apoyo del equipo de consultoría integrado con los usuarios finales.

Capítulo 4. Resultados.

Como resultado de la implementación de un sistema ERP para la gestión de los procesos en la empresa, se tiene un sistema que gestionará los procesos de negocio de principio a fin, estos procesos, son todos aquellos que se mencionaron en el capítulo anterior, todo esto con ayuda de los módulos y componentes que se instalaron y configuraron de manera adecuada para cubrir las necesidades y requerimientos extras que el cliente necesitó.

Así, mediante la integración de los diferentes componentes, se tiene un ciclo de vida del proyecto completamente controlado, y gestionado por SAP.

Los principales sistemas actores de la gestión de los procesos son: ERP, CRM y PI, teniendo cada uno las siguientes funciones:

PI

- Es el componente central de comunicación, en donde se lleva a cabo la transferencia y transformación (en caso de existir) de la información.

ERP.

- Contiene la configuración de los módulos principales: FI, PP, MM, QM, es decir, el flujo de los procesos y los resultados, son tomados desde este componente.
- Los datos maestros de la operación están contenidos en este componente, de esta manera, CRM y los demás, se sincronizarán y se garantizará la integridad de la información.

CRM.

- Es el componente desde donde se llevará a cabo la operación, es decir, cuando los usuarios ingresen al sistema, entrarán directamente a CRM.
- La manera de realizar peticiones, es por medio de una pantalla en donde se levantarán tickets de servicio, dependiendo de la necesidad, pueden ser procesos de lavandería, alimentos, jardinería, etc.
- Por medio de este componente, se dará seguimiento a las peticiones realizadas a través de un tablero de control, en donde se pueden ver todos los tickets que existen y su estatus de acuerdo a los KPI y SLA definidos.
- El monitoreo de este sistema por medio del tablero, es esencial, puesto que si no se cumplen los SLA definidos, se generarán penalizaciones por la disponibilidad y cumplimiento de los servicios, de esta manera, un tablero de control en donde existen señalizaciones tipo semáforo, es crítico cuando los tiempos de respuesta y planeación de actividades están completamente definidos.
- Se tienen reportes de cumplimiento de SLA, en donde se determinan las penalizaciones (en caso de existir) por cada incumplimiento de SLA.

Al ser la parte central del procesamiento de solicitudes de servicios (jardinería, mantenimiento, alimentos, etcétera), CRM alberga el flujo principal de servicios y solicitudes, interactuando mayormente con el ERP, dado que dicho componente contiene

la base de datos e información que CRM necesita para calcular tiempos y recursos necesarios para completar las operaciones y solicitudes de servicio que se crean por medio de tickets.

El proceso de solicitud de servicio, se describe en la figura 23:

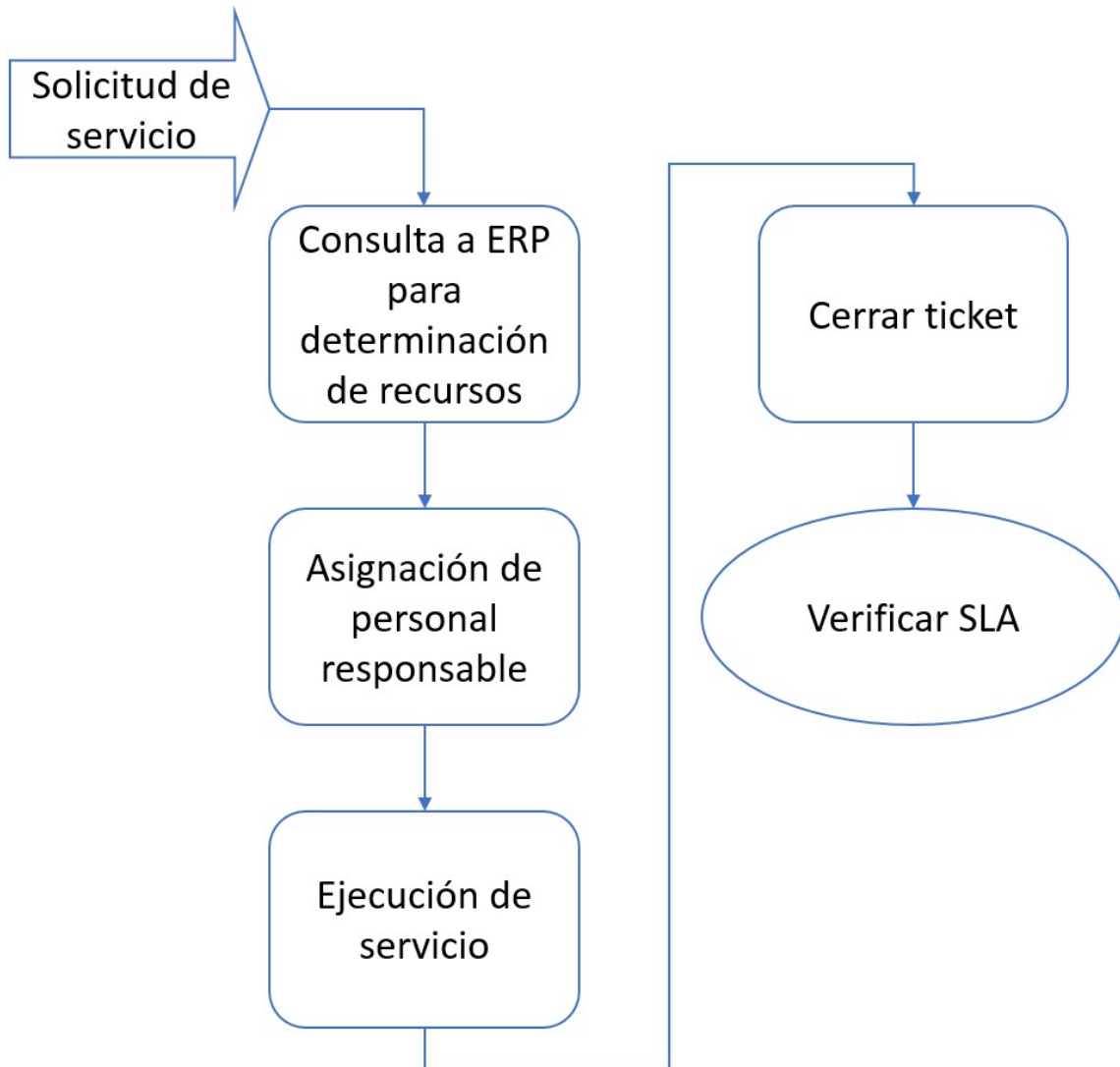


Figura 23: Ciclo de vida de la solicitud de servicio.

1. Solicitud de servicio: Desde la plataforma central, se solicita un servicio con toda la información necesaria para el cumplimiento del mismo, es decir, se debe proporcionar el tipo de servicio solicitado, el área en donde se solicita, el departamento que solicita, y el sistema determinará automáticamente la fecha y hora de solicitud, pues esta información es muy importante para el cumplimiento de los SLA.

2. Determinación de recursos: Mediante una conexión al ERP, por medio de PI, el sistema CRM, determinará los materiales necesarios para la ejecución de dicha actividad, por ejemplo, si se solicita un servicio de lavandería, los materiales necesarios incluirán artículos como detergente, cestas para ropa, planchas, etc. De esta forma, no se deja algún paso sin cubrir. Adicionalmente, existen hojas de instrucciones para la ejecución de cada uno de los servicios descritos con anterioridad en este documento, lo que reduce la posibilidad de errores y retrasos considerablemente.
3. Asignación de personal responsable: El personal responsable de los servicios será notificado por su área correspondiente que existe un servicio solicitado en espera de su ejecución, así, la entidad responsable de notificar las solicitudes a las áreas correspondientes, es el centro de control.
4. Ejecución del servicio: Una vez asignado el personal, se tiene un determinado tiempo de respuesta en dónde se deberá proporcionar el servicio solicitado en tiempo y forma. Para la ejecución de cada servicio, existe una hoja de secuencia que contiene paso a paso las acciones que deben llevarse a cabo para el correcto cumplimiento del servicio, así como los materiales que se necesitan para la ejecución del mismo.
5. Cerrar el ticket: Una vez que se ha ejecutado el servicio, se debe cerrar el ticket indicando si fue ejecutado correctamente y en el tiempo establecido, para así poder verificar el cumplimiento de los SLA.
6. Verificar los SLA: Esta es la parte más delicada y que más atención requiere en el proceso, debido a que el incumplimiento de los servicios, o una mala calidad en la entrega de los mismos, genera penalizaciones económicas a las empresas proveedoras de los mismos. Dicha verificación deberá determinar si el servicio se cumplió de acuerdo a las hojas de instrucciones preestablecidas para dicho proceso, así, si un SLA no se cumplió, existirán penalizaciones económicas por incumplimiento.

SLA

Los SLA juegan un rol importante y esencial en el proceso de negocio, pues toda la configuración y trabajo que se realiza, está enfocado en optimizar los procesos tanto como sea posible, para de esta manera, cumplir con los objetivos y evitar cualquier tipo de penalización o el incumplimiento parcial o total de algún servicio, principalmente.

El tiempo de ejecución de cualquier servicio, se encuentra determinado por tabuladores establecidos por la empresa de acuerdo a las necesidades de tiempo de respuesta, basados también en las hojas de instrucciones para cada uno de los procesos asignados a los servicios.

El objetivo principal, es que los SLA siempre sean cumplidos en tiempo y forma, sin embargo, pueden existir inconvenientes o imprevistos que retrasen la ejecución de los servicios, y algunos de ellos, no son imputables a la empresa prestadora de servicios, a continuación se muestra un esquema que explica brevemente el ciclo de vida de un SLA.

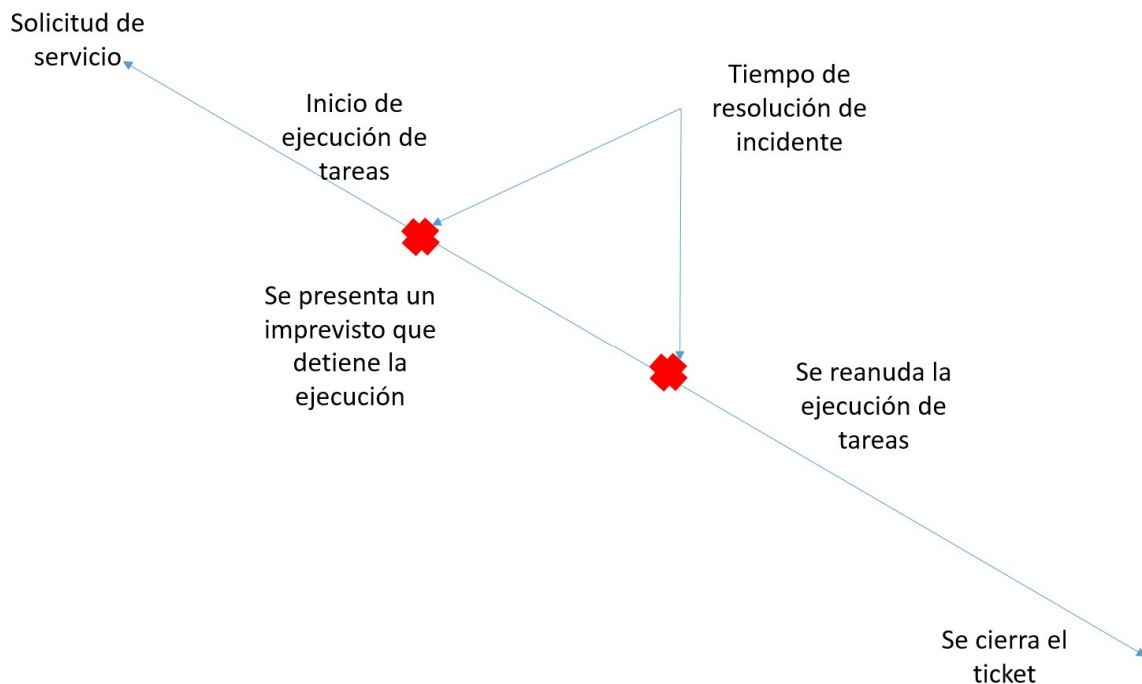


Figura 24. Ciclo de SLA.

Como se puede observar en la figura 24, el proceso puede verse interrumpido por causas de fuerza mayor, por ejemplo: no se cuenta con la pieza de maquinaria que está fallando y no puede ser reemplazada, así, el proveedor de servicio no puede llevar a cabo la tarea, pues es obligación de la empresa tener almacenes surtidos con los materiales necesarios para reparar cualquier desperfecto que se pueda presentar en las instalaciones.

Si lo anterior llegara a ocurrir, se deberá notificar al centro de control que algo ocurrió durante la ejecución del servicio, y de esta forma, detener el tiempo del SLA, lo que detonará un cambio de estatus en el ticket a "Detenido por cliente".

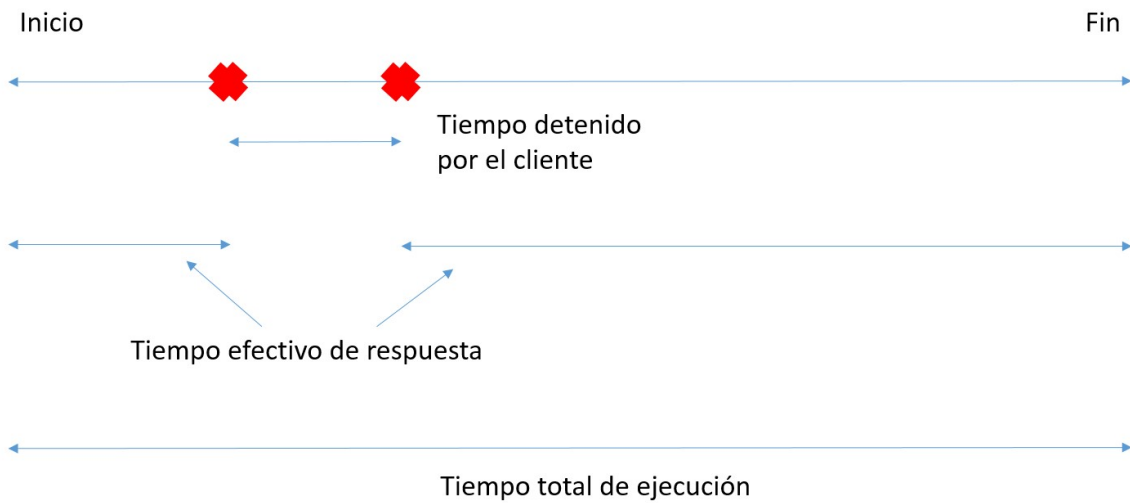


Figura 25: Definición de tiempos en un ticket.

Como ya se mencionó anteriormente, y como se muestra en la figura 25, cualquier evento previsto o no, que ocurra en el ciclo de ejecución y cumplimiento de un servicio, se detonará en un cambio de estatus en el ticket, lo cual determinará los tiempos que se contabilizarán para determinar si el SLA fue concluido o no satisfactoriamente. Los estatus que puede tomar un ticket, son:

- Solicitado.
- En ejecución.
- Detenido por cliente.
- Cerrado.

Para efectos de determinación de cumplimiento de los SLA, se tomará en cuenta solo el tiempo que el ticket se encuentre en ejecución, pues no se puede penalizar a la prestadora de servicios por alguna falta por parte de la empresa, de esta manera, se solicitó un reporte en donde se presentaran todas las deducciones que existan por periodo de tiempo, en este caso, mensual, para así determinar los costos de operación y las penalizaciones que la empresa puede aplicar a las prestadoras de servicios.

Deducciones.

Se ha hablado reiteradamente del concepto de “deducciones”, y en este apartado, se presentará una visión general de lo que este concepto significa, pues es de los más importantes debido a su naturaleza económica.

Existen diferentes tipos de deducciones, las cuáles deben estar completamente definidas, delimitadas y contar con una fórmula para su cálculo, a continuación, se presentan las deducciones que existen y en algunos casos, las fórmulas que se programaron para su cálculo.

- Deducción aplicable.
- Deducción máxima aplicable en un mes determinado.
- Deducción pendiente por aplicar.
- Deducción por fallas de calidad.
- Deducción por fallas de disponibilidad.
- Deducción por fallas en la prestación del servicio.
- Deducción por fallas generales del servicio.
- Deducción por reiteración de fallas.
- Deducción por reiteración de fallas de calidad.
- Deducción por reiteración de fallas de disponibilidad.
- Deducción por reiteración de fallas generales del servicio.
- Deducción por errores u omisiones en la información.
- Deducción total por fallas en la prestación del servicio.

Los conceptos generales para entender los tipos de deducciones son los siguientes:

- Falla de disponibilidad: Ocurre cuando una unidad funcional (área de las instalaciones especificadas) no cumple con la condición de disponibilidad dentro del tiempo de rehabilitación.
- Falla de calidad: Ocurre cuando una unidad funcional no cumple con la condición de calidad dentro del tiempo de respuesta, tiempo de rehabilitación parcial, en caso que aplique, o tiempo de rehabilitación correspondiente.
- Falla general de servicio: Ocurre cuando una de las actividades definidas dentro del catálogo de servicios generales no cumple con los estándares definidos.
- Tiempo de rehabilitación y rehabilitación parcial: Para cada área funcional y sus respectivas unidades funcionales, dependiendo de la categoría de área de la misma dentro de las instalaciones, corresponde un tiempo de rehabilitación de la condición de disponibilidad. De igual manera se especifica para cada actividad dentro de los estándares establecidos, el tiempo de rehabilitación de la condición de calidad, así como el tiempo de rectificación de los estándares generales.

Un ejemplo de fórmula que se utiliza para el cálculo de las deducciones es:

- Deducción por fallas de disponibilidad

$$DTFD = \sum_{d=1}^D DFD_d^i$$

$$DFD_d^i = \frac{PMR_i}{dm_i} * pdfd_j * ppa_m * ppu_n * \frac{utfd_d^i}{ut_n}$$

En donde:

DFD = Deducción por fallas de disponibilidad en el mes i derivada de una falla de disponibilidad en una unidad funcional n dentro de un área funcional m .

PMR = Pago mensual de referencia del mes i .

dm = número de días en que haya sido prestado el servicio integral de capacidad en el mes i .

pdfd = Porcentaje de deducción por fallas de disponibilidad.

ppa = Coeficiente de ponderación de área funcional m a la que corresponde la unidad funcional n .

ppu = Coeficiente de ponderación de la unidad funcional n dentro del área funcional m .

utfd = Número de turnos en los que se presenta el incumplimiento con la condición de disponibilidad d en el mes i .

ut = Número de turnos existentes para cada día atribuibles a una unidad funcional n .

Ejemplos de aplicaciones.

Adicional, en el sistema quedan contenidos los programas necesarios para la correcta operación de la empresa, desde reportes, funciones, formulario, etc.

Interface Contabilidad Electronica SAT

Selección de Reporte:

- Catálogo De Cuentas
- Balanza De Comprobación
- Pólizas

Datos de Selección:

Sociedad:

Plan de Cuentas:

Estructura de balance:

Periodo:

Ejercicio:

Selección de Reporte:

Tipo de Requerimiento

Tipo de solicitud:

Número de Orden:

Número de Trámite:

Descarga de Archivos.

Ruta de descarga:

Tipo de Ejecución

- Mostrar ALV (Modo Test)
- Generar XML en Fondo

SAP | PMD (1) 500 | pdmxerpd | INS

Figura 26. Ejemplo de reporte, interfaz de usuario.

Sistema Ayuda

Asignación de Fondos por Obra-Semana

Soc.	Ejercicio	Cod. obra	...	Σ	Asignado Σ	Gastado Σ	In. Asign Σ	In. Gasta	Mon.
0100	2013	FS-11-16	19				0.00	0.00	MXN
	0000						0.00	0.00	

Figura 27. Ejemplo de reporte, ejecución.

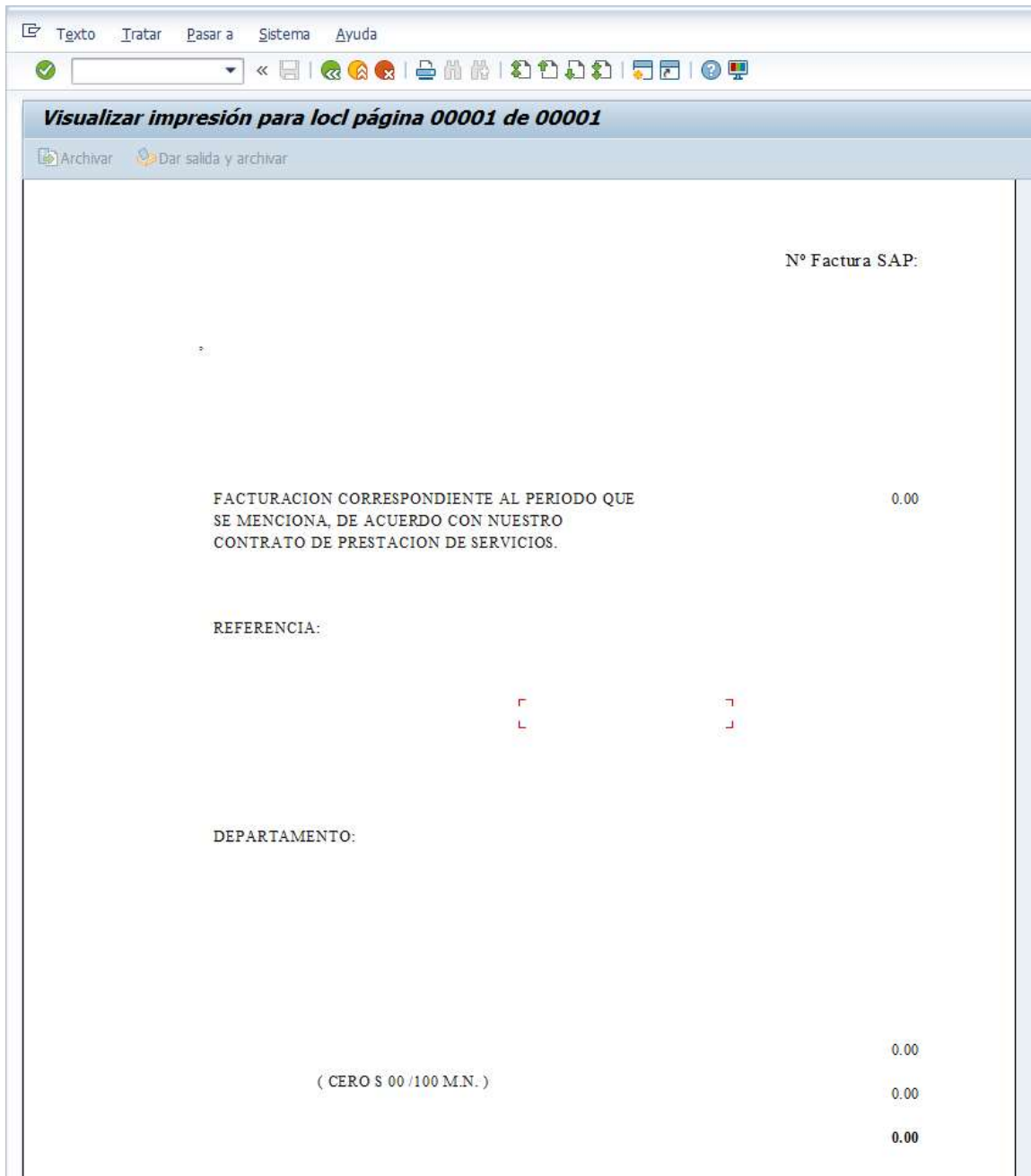


Figura 28. Ejemplo de factura generada.

Texto Tratar Pasar a Sistema Ayuda

Visualizar impresión para la página 00001 de 00001

Archivar Dar salida y archivar

Mexico D. F. a 00 de 00 de 0000
 No. Factura SAP:

RAZON SOCIAL:
 DIRECCION: COL
 CIUDAD:
 CODIGO POSTAL:
 R. F. C.:

RECIBI DEL

COMO PAGO AL AVANCE ESTIMADO DE OBRA NUMERO 0
 CORRESPONDIENTE A LOS TRABAJOS DE:

MONTO A ESTIMAR	\$	0.00
ANTICIPO	\$	0.00
FONDO	\$	0.00
SUBTOTAL	\$	0.00
IMPUESTO AL VALOR AGREGADO	\$	0.00
DERECHOS Y APORTACIONES	\$	0.00
TOTAL	\$	0.00

Figura 29. Ejemplo de factura generada.

Sistema Ayuda

Historial de pagos y autorizaciones

OBRA/ACREEDOR	Nº doc.	Pos.	MOD	Bloq...	TCODE	Usuario	Fecha	Importe	Pagado	Mon...	Refer...
0005000311		0						280,000.00- 0.00			
100329		0						30,000.00- 0.00			
1900000002	1900000002	1 1	S		ZSOLICIT...	ADERIO	21.09.2009	15,000.00- 0.00		MXN	F.5050
1900000002	1900000002	1 2			ZAUTORIZ...	ADERIO	21.09.2009	15,000.00- 0.00		MXN	F.5050
100333		0						150,000.00- 0.00			
1900000003	1900000003	1 1	S		ZSOLICIT...	ADERIO	21.09.2009	75,000.00- 0.00		MXN	F-53313
1900000003	1900000003	1 2			ZAUTORIZ...	ADERIO	21.09.2009	75,000.00- 0.00		MXN	F-53313
100396		0						100,000.00- 0.00			
0007000301		0						5,244.00- 0.00			
0007000401		0						244.26- 0.00			
0007100401		0						60,529.15- 0.00			
1000308		0						1,600,000.00- 0.00			
CCBC01		0						575.00- 0.00			
100451		0						575.00- 0.00			
5100002499	5100002499	1 1	M		ZSOLICIT...	OCCONTG	29.05.2009	575.00- 0.00		MXN	63374
CCEX01		0						1,985.00- 0.00			
CCEX04		0						1,410.00- 0.00			
CCEX05		0						43,844.24- 0.00			
CCMC01		0						590.00- 0.00			
CCMC02		0						590.00- 0.00			
CCMC03		0						590.00- 0.00			
CCMT02		0						5,321.01- 0.00			
100451		0						5,321.01- 0.00			
5100002492	5100002492	1 1	M		ZSOLICIT...	OCCONTG	11.05.2009	1,800.00- 0.00		MXN	62982
5100002493	5100002493	1 1	M		ZSOLICIT...	OCCONTG	12.05.2009	1,781.01- 0.00		MXN	62987
5100002506	5100002506	1 1	M		ZSOLICIT...	OCCONTG	29.05.2009	1,740.00- 0.00		MXN	63374
CCMT03		0						575.00- 0.00			
CCPERF-01		0						163,372.50- 0.00			
CCRE01		0						54,264.30- 0.00			
CCRE03		0						3,496.02- 0.00			
CCRE04		0						3,267.86- 0.00			
CCRE05		0						4,137.85- 0.00			
CCRE06		0						2,677.86- 0.00			
CCRM01		0						6,973.91- 0.00			

SAP PMD (1) 500 pdmxxrpd INS

Figura 30. Ejemplo de reporte

Conclusiones.

Se instaló un sistema robusto con diversos componentes, como se mencionó a lo largo del documento, en donde los procesos solicitados quedaron completamente mapeados y configurados en el sistema, además de funcionalidades específicas creadas para facilitar y agilizar la operación de la empresa.

SAP proporcionó la herramienta adecuada y la infraestructura tecnológica para el cumplimiento y correcta ejecución de los servicios deseados, de esta manera, la empresa tiene una visibilidad completa del proceso de negocio, es decir, desde la solicitud de un servicio, hasta la conclusión del mismo, pasando por monitoreo de variables y estándares de calidad.

El equipo de consultoría que estuvo a cargo de la implementación, tuvo que basarse en las metodologías pre establecidas por SAP, y utilizar diversas herramientas para la consecución de los objetivos en tiempo y forma, es decir, se utilizaron las metodologías ASAP para llevar un control de las diferentes etapas del proyecto, y así lograr concluir exitosamente cada una de las etapas. Se echó mano también de la metodología SCRUM, con la cual, la gestión del equipo de desarrollo fue completa,

Utilizando y aplicando los conocimientos adquiridos a lo largo de la carrera de Ingeniería en Computación, se logró crear un conjunto de desarrollos robustos y que cubren y satisfacen la necesidades del cliente, académicamente hablando, las habilidades que se aplicaron fueron principalmente conocimientos en programación, tanto estructurada como orientada a objetos, a pesar que ABAP no es un lenguaje que haya aprendido en mi estancia en la Facultad de Ingeniería, apliqué los paradigmas de programación aprendidos en lenguajes como C, Java, Perl y otros tantos con los que interactué a lo largo de la carrera en diversos proyectos y asignaturas.

De la misma manera, habilidades como el trabajo en equipo, la gestión de recursos, la administración del tiempo y las capacidades para estimar tiempos y distribuirlos óptimamente para cumplir con los objetivos establecidos en las fechas pactadas, son cualidades de un buen consultor y que tuve la fortuna de aprender, y adquirir en el transcurso de la carrera y con la ayuda de los profesores, pues de una manera indirecta, fomentaron el aprendizaje mediante los proyectos, ya sea individuales o en equipo, lo que da la capacidad para distribuir las tareas a ejecutar, otras actividades como prácticas, clases, proyectos y demás, hacen que se despierte el sentido de urgencia y que el trabajo bajo presión no desbalancee ni ponga en peligro la entrega de las asignaciones.

De esta manera, la carrera de Ingeniería en Computación, me dio las bases, conocimientos y carácter para desempeñarme en el mundo laboral de este país, que día con día exige profesionales que cumplan con los más altos estándares de desempeño, calidad, productividad y proactividad dentro de sus respectivas áreas de trabajo.

Apéndice 1. Metodología SCRUM.

SCRUM es el nombre con el que se denomina a los marcos de desarrollo ágiles caracterizados por:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o en cascada.

Historia.

Este modelo fue identificado y definido por Ikujiro Nonaka e Hirotaka Takeuchi a principios de los años 80, al analizar cómo desarrollaban los nuevos productos las principales empresas de manufactura tecnológica: Fuji-Xerox, Canon, Honda, NEC, Epson, Brother, 3M y Hewlett-Packard (Nonaka & Takeuchi, *The New New Product Development Game*, 1986).

En su estudio, Nonaka y Takeuchi compararon la nueva forma de trabajo en equipo, con el avance en formación de melé (SCRUM en inglés) de los jugadores de Rugby, a raíz de lo cual quedó acuñado el término SCRUM para referirse a ella.

Aunque esta forma de trabajo surgió en empresas de productos tecnológicos, es apropiada para cualquier tipo de proyecto con requisitos inestables y para los que requieren rapidez y flexibilidad, situaciones frecuentes en el desarrollo de determinados sistemas de software.

En 1995, Ken Schwaber presentó el SCRUM Development Process en OOPSLA 1995 (Object-Oriented Programming Systems & Applications conference), un marco de reglas para desarrollo de software, basado en los principios de SCRUM, y que él había empleado en el desarrollo de Delphi, y Jeff Sutherland en su empresa Easel Corporation (compañía que en los macrojuegos de compras y fusiones, se integraría en VMARK, y luego en Informix y finalmente en Ascential Software Corporation).

Características.

SCRUM es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en SCRUM son el SCRUM Master, que procura facilitar la aplicación de la metodología y gestionar cambios, el Product Owner, que representa a los interesados externos o internos, y el equipo, que ejecuta el desarrollo y demás elementos relacionados. Durante cada sprint, un periodo entre una y cuatro

semanas (la magnitud es definida por el equipo y debe ser lo más corta posible), el equipo crea un incremento de software potencialmente entregable. El conjunto de características que forma parte de cada sprint viene del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar (PBI, Product Backlog Item). Los elementos del PB que forman parte del sprint se determinan durante la reunión de planeación de sprint. Durante esta reunión, el Product Owner identifica los elementos del Product Backlog que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo conversa con el Product Owner buscando la claridad y magnitud adecuadas para luego determinar la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint.

SCRUM permite la creación de un equipo auto organizado impulsando la localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

Un principio clave de SCRUM es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan, y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, esta metodología adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

Las características más marcadas que se logran notar en SCRUM son:

- Gestión regular de las expectativas del cliente.
- Resultados anticipados.
- Flexibilidad y adaptación.
- Retorno de inversión.
- Mitigación de riesgos.
- Productividad y calidad.
- Alineamiento entre cliente y equipo.
- Equipo motivado.

Cada uno de estos puntos mencionados, permiten que el SCRUM sea utilizado de manera regular en un conjunto de buenas prácticas para el trabajo en equipo y de esa manera obtener resultados posibles.

Existen varias implementaciones de sistemas para gestionar el proceso de SCRUM, que van desde sencillos post-its y pizarras hasta paquetes de software. Una de las mayores ventajas de SCRUM es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar. Así, si se utiliza una pizarra con notas autoadhesivas cualquier miembro del equipo

podrá ver tres columnas: trabajo pendiente, tareas en proceso y hecho. De un solo vistazo, una persona puede ver en qué están trabajando los demás en un momento determinado.

Principales roles.

Product Owner.

Representa al cliente, y es el encargado de asegurar que el equipo trabaje de forma adecuada desde la perspectiva del negocio.

SCRUM máster.

Su principal función es eliminar los obstáculos que impiden que el equipo alcance el objetivo de cada sprint, no es el líder del equipo, sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El SCRUM máster se asegura de que el proceso se utilice como es debido y hace que las reglas se cumplan.

Equipo de desarrollo.

Tiene la responsabilidad de entregar el producto.

Reuniones de SCRUM.

Reunión diaria.

Cada día de un sprint, se realiza la reunión sobre el estado del proyecto, la cual sigue estos lineamientos.

- La reunión debe comenzar puntualmente.
- Todos son bienvenidos, pero solo los involucrados en el proyecto pueden hablar.
- La reunión tiene una duración fija de 15 minutos, independientemente del tamaño del equipo.
- Debe llevarse a cabo en el mismo lugar y hora todos los días.

Durante esta reunión, cada miembro del equipo debe contestar 3 preguntas:

- ¿Qué has hecho desde ayer?
- ¿Qué es lo que harás para mañana?
- ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo?

Lo anterior tiene como principal propósito saber si cada miembro del equipo está cumpliendo con los plazos planeados para conseguir un sprint exitoso.

Reunión de SCRUMs.

Este tipo de sesiones se realizan cuando en la organización, existen varios equipos de SCRUM, y se les permite discutir su trabajo, enfocándose especialmente en áreas de solapamiento e integración. Se hace normalmente después de las reuniones diarias o como máximo cada 2 días, a estas reuniones solamente asiste una persona asignada por cada equipo de SCRUM.

La agenda es la misma que en las reuniones diarias, además de las siguientes 4 preguntas:

- ¿Qué ha hecho tu equipo desde la última reunión?
- ¿Qué hará tu equipo antes de la próxima reunión?
- ¿Hay algo que demore o estorbe a tu equipo?
- ¿Estás a punto de poner algo en el camino de otro equipo?

Reunión planificadora de sprint.

Al inicio de cada ciclo de sprint, se lleva a cabo una reunión de planificación, en donde se pretende lo siguiente:

- Seleccionar el trabajo a realizar durante el periodo.
- Preparar con el equipo completo el tiempo que llevará hacer el trabajo.
- Identificar y comunicar cuánto del trabajo es probable que se realice durante el actual sprint.
- La sesión debe llevar 8 horas como tiempo máximo.

Reunión de revisión del sprint.

El objetivo de estas reuniones es atender los siguientes puntos en un espacio no mayor a 4 horas:

- Revisar el trabajo que fue completado y no completado.
- Presentar el trabajo completado a los interesados.

Beneficios del SCRUM.

- Flexibilidad a cambios: Gran capacidad de reacción ante los cambios de requerimientos generados por las necesidades del cliente o evolución del mercado. El marco de trabajo está diseñado para adecuarse a las nuevas exigencias que implican los proyectos.
- Reducción del tiempo de fabricación de software: El cliente puede empezar a utilizar las características más importantes del proyecto antes de que éste se encuentre completamente terminado.
- Mayor calidad en el software: El trabajo metódico y la necesidad de obtener una versión de trabajo funcional después de cada iteración, ayuda la obtención de un software de alta calidad.
- Mayor productividad: Se logra, entre otras razones, por la eliminación de la burocracia y la motivación del equipo proporcionado por el hecho de que pueden estructurarse de manera autónoma.
- Maximiza el retorno de inversión: Creación de software solamente con las prestaciones que contribuyen a un mayor valor de negocio gracias a la priorización por retorno de inversión.

- Predicciones de tiempos: A través de este marco de trabajo se conoce la velocidad media del equipo por sprint, con lo que es posible estimar de manera fácil, cuando se podrá hacer uso de una determinada funcionalidad.
- Reducción de riesgos: El hecho de desarrollar, en primer lugar, las funcionalidades de mayor valor y de saber la velocidad a la que el equipo avanza en el proyecto, permite despejar riesgos efectivamente de manera anticipada.

Apéndice 2. Ejemplos de tipos de desarrollos.

En esta sección se mostrarán algunos ejemplos de desarrollos y sus elementos principales.

Tabla de diccionario.

ZCXCM

Campos de salida SAPLMBGB

Row	Field name	Position	Key	Data element	Domain	Datatype	Length	Domain text
1	CNT02	1		MB_BUSTW_CNT	NUM03	NUMC	3	Contador consecutivo string contab.p.valores (gest.stocks)
2	CURTP	2		CURTP	CURTP	CHAR	2	Tipo de moneda y vista de valoración
3	WAERS	3		WAERS	WAERS	CUKY	5	Clave de moneda
4	DMBTR	4	X	DMBTR	WERT7	CURR	13	Importe en moneda local
5	SHKZG	5	X	SHKZG	SHKZG	CHAR	1	Indicador debe/haber
6	SHKZM	6	X	SHKZG	SHKZG	CHAR	1	Indicador debe/haber
7	BUALT	7		BUALT	WERTV7	CURR	13	Importe contab.en control alternativo de precios
8	BUZID	8		BUZID	CHAR1	CHAR	1	Identificación del apunte
9	VKWR7	9		VKWR7	WERT7	CURR	13	Valor p.precios de venta con IVA
10	VORSL	10		VORSL	VORSL	CHAR	3	Clave de operación para determinación de cuentas
11	XUMBW	11		XUMBW	XFELD	CHAR	1	Revaloración
12	FWBAS	12		FWBAS	WERT7	CURR	13	Importe base impuesto moneda documento
13	LBBTR	13		LBBTR	WERT7	CURR	13	Val.material facilitado en mon.local
14	XSUBS	14		XFELD	XFELD	CHAR	1	Casilla de selección

Como se puede ver en la figura anterior, una tabla en SAP, se crea proporcionando la siguiente información.

- Campo: Nombre del campo de la tabla
- Campo clave: Se marca sólo si el campo pertenece a la llave de la tabla.
- Elemento de datos: Este contiene el dominio, es decir, el tipo de dato y su longitud.
- Tipo de dato: Se puede proporcionar directamente el tipo de dato siempre y cuando no se proporcione el elemento, estos tipos de datos son los predeterminados de SAP.
- Longitud: Auto explicativo y trabaja en conjunto con el tipo de dato.
- Texto explicativo: Descripción del campo.

Programa de carga.

```
*&-----*
*& Report ZCARGA_MONEDA
*-----*
```

```
REPORT zcarga_moneda.
*** DECLARACION DE VARIABLES
DATA: l_ruta_s TYPE string.
TYPE-POOLS: truxs.

DATA: it_raw TYPE truxs_t_text_data.
```

```
TYPES: BEGIN OF st_zfs_moneda.
INCLUDE STRUCTURE zfs_moneda.
TYPES: END OF st_zfs_moneda.
```

```
*** DEFINICION DE PARAMETROS DE SELECCION.
SELECTION-SCREEN BEGIN OF BLOCK bk1 WITH FRAME TITLE
text-006.
PARAMETERS: p_file LIKE ibiparms-path DEFAULT "
OBLIGATORY LOWER CASE.
SELECTION-SCREEN END OF BLOCK bk1.
```

```

*** EVENTOS
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_file.
CALL FUNCTION 'F4_FILENAME'
  EXPORTING
    field_name = 'P_FILE'
  IMPORTING
    file_name = p_file.

*****
*****
*** INICIO DE EJECUCIÓN DE PROGRAMA
START-OF-SELECTION.

  PERFORM exportfile.

*&-----*
*& Form EXPORTFILE
*&-----*
* text
*-----*
FORM exportfile.
  DATA: it_zfs_moneda TYPE st_zfs_moneda OCCURS 0 WITH
  HEADER LINE.
  CLEAR it_zfs_moneda. REFRESH it_zfs_moneda.

  DELETE FROM zfs_moneda.

```

```

CALL FUNCTION 'TEXT_CONVERT_XLS_TO_SAP'
  EXPORTING
    * I_FIELD_SEPARATOR =
    * =i_line_header = 'X'
    i_tab_raw_data = it_raw " WORK TABLE
    i_filename = p_file
  TABLES
    i_tab_converted_data = it_zfs_moneda[] "ACTUAL DATA
  EXCEPTIONS
    conversion_failed = 1
    OTHERS = 2.
IF sy-subrc NE 0.
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
  WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.
DELETE FROM zfs_moneda.

MODIFY zfs_moneda FROM TABLE it_zfs_moneda[].

IF sy-subrc NE 0.
  MESSAGE 'PROBLEMA CON LA CARGA DE DATOS' TYPE 'E'.
ELSE.
  MESSAGE 'CARGA DE DATOS CONCLUIDA CON EXITO' TYPE
  'S'.
ENDIF.
ENDFORM.          "CONCEPTOS

```

Programa tipo reporte.

```

*&-----*
*& Report ZFI_RE_MULTICASH
*&
*&-----*
*&
*&
*&-----*
REPORT zfi_re_multicash.

INCLUDE: zfi_re_multicash_top,
        zfi_re_multicash_f01,
        zfi_re_multicash_e01.

*&-----*
*& Include ZFI_RE_MULTICASH_TOP
*&-----*

TYPES: BEGIN OF ty_file,
        line(1000) TYPE c,
      END OF ty_file,

      BEGIN OF ty_header,
        c01(12) TYPE c, "Clave de banco
        c02(24) TYPE c, "Cuenta bancaria
        c03(4) TYPE c, "Número del estado de cuenta
        c04(8) TYPE c, "Fecha del estado de cuenta
        c05(3) TYPE c, "Moneda
        c06(21) TYPE c, "Saldo anterior
        c07(21) TYPE c, "Total de cargos
        c08(21) TYPE c, "Total de abonos
        c09(21) TYPE c, "Saldo final
        c10(35) TYPE c, "Titular de la cuenta

```

```

c11(35) TYPE c, "Descripción de la cuenta
c12(1) TYPE c, "No se usa
c13(1) TYPE c, "No se usa
c14(1) TYPE c, "No se usa
c15(1) TYPE c, "No se usa
c16(1) TYPE c, "No se usa
c17(1) TYPE c, "No se usa
c18(5) TYPE c, "Número de movimientos
END OF ty_header,

BEGIN OF ty_detail,
d01(12) TYPE c, "Clave de banco
d02(24) TYPE c, "Cuenta bancaria
d03(4) TYPE c, "Número del estado de cuenta
d04(8) TYPE c, "Fecha valor
d05(10) TYPE c, "No se usa
d06(27) TYPE c, "Referencia (doc. de pago del R3)
d07(21) TYPE c, "Leyenda / código de operación del
banco
d08(1) TYPE c, "No se usa
d09(1) TYPE c, "No se usa
d10(16) TYPE c, "Número de cheque / número de folio
d11(21) TYPE c, "Importe
d12(1) TYPE c, "No se usa
d13(1) TYPE c, "No se usa
d14(8) TYPE c, "Fecha de contabilización
d15(1) TYPE c, "No se usa
d16(1) TYPE c, "No se usa
d17(27) TYPE c, "Nota 2
d18(27) TYPE c, "Nota 3
d19(27) TYPE c, "Nota 4
d20(27) TYPE c, "Nota 5
d21(27) TYPE c, "Nota 6
d22(27) TYPE c, "Nota 7

```



```

d23(27) TYPE c,          "Nota 8
d24(27) TYPE c,          "Nota 9
d25(27) TYPE c,          "Nota 10
d26(27) TYPE c,          "Nota 11
d27(27) TYPE c,          "Nota 12
d28(27) TYPE c,          "Nota 13
d29(27) TYPE c,          "Nota 14
d30(27) TYPE c, "Número de cliente (opcional)
d31(27) TYPE c, "Socio de negocio (opcional)
d32(12) TYPE c, "Clave de banco del socio de negocio
(opcional)
d33(24) TYPE c, "Cuenta bancaria del socio de negocio
(opcional)
d34(3) TYPE c, "Código de transacción del banco
d35(1) TYPE c, "No se usa
END OF ty_detail,

```

```

BEGIN OF ty_input11,
i01(2) TYPE c, "Registro
i02(4) TYPE c, "Clave país
i03(4) TYPE c, "Sucursal
i04(10) TYPE c, "Cuenta
i05(6) TYPE c, "Fecha inicial
i06(6) TYPE c, "Fecha final
i07(1) TYPE c, "Saldo 2[+] / 1[-]
i08(14) TYPE c, "Saldo inicial 1
i09(3) TYPE c, "Moneda alfabética
i10(1) TYPE c, "Dígito cuenta clabe
i11(23) TYPE c, "Titular de la cuenta
i12(3) TYPE c, "Plaza cuenta clabe
i13(3) TYPE c, "Libre
END OF ty_input11,

```

```

BEGIN OF ty_input22,
i14(2) TYPE c, "Registro
i15(4) TYPE c, "Clave país
i16(4) TYPE c, "Sucursal
i17(6) TYPE c, "Fecha operación
i18(6) TYPE c, "Fecha valor
i19(2) TYPE c, "Para uso futuro
i20(3) TYPE c, "Clave leyenda
i21(1) TYPE c, "Cargo (1) / Abono (2)
i22(14) TYPE c, "ImporteDato
i23(10) TYPE c, "Dato
i24(28) TYPE c, "Concepto
END OF ty_input22,

```

```

BEGIN OF ty_input23,
i25(2) TYPE c, "Registro
i26(2) TYPE c, "Código dato
i27(38) TYPE c, "Referencia ampliada
i28(38) TYPE c, "Referencia
END OF ty_input23,

```

```

BEGIN OF ty_input32,
i29(2) TYPE c, "Registro
i30(3) TYPE c, "Clave país
i31(2) TYPE c, "Subcódigo de registro
i32(35) TYPE c, "Información 1
i33(35) TYPE c, "Información 2
END OF ty_input32,

```

```

BEGIN OF ty_input33,
i34(2) TYPE c, "Registro
i35(4) TYPE c, "Clave país
i36(4) TYPE c, "Sucursal

```

```

i37(10) TYPE c, "Cuenta
i38(5) TYPE c, "No. de cargos
i39(14) TYPE c, "Importe total de cargos
i40(5) TYPE c, "No. de abonos
i41(14) TYPE c, "Importe total de abonos
i42(1) TYPE c, "Saldo 2[+] / 1[-]
i43(14) TYPE c, "Saldo final
i44(3) TYPE c, "Moneda alfabética
END OF ty_input33.

```

```
DATA: filename TYPE string,
```

```
data_tab TYPE STANDARD TABLE OF ty_file.
```

```
DATA: vg_header TYPE ty_header,
vg_detail TYPE ty_detail,
```

```
tg_header TYPE STANDARD TABLE OF ty_header,
tg_detail TYPE STANDARD TABLE OF ty_detail,
```

```
vg_input11 TYPE ty_input11,
vg_input22 TYPE ty_input22,
vg_input23 TYPE ty_input23,
vg_input32 TYPE ty_input32,
vg_input33 TYPE ty_input33,
```

```
vg_flag TYPE c,
vg_cargos TYPE p DECIMALS 2,
vg_abonos TYPE p DECIMALS 2.
```

```
CONSTANTS: cg_colon(1) TYPE c VALUE ',',
cg_point(1) TYPE c VALUE '.',
cg_minus(1) TYPE c VALUE '-',
cg_plus(1) TYPE c VALUE '+'.

```

```
PARAMETERS: p_infile LIKE ibipparms-path OBLIGATORY.
```

```

*&-----*
*& Include      ZFI_RE_MULTICASH_F01
*&-----*

*&-----*
*& Form UPLOAD_FILE
*&-----*
* text
*-----*
* --> p1 text
* <-- p2 text
*-----*

```

```
FORM upload_file .
```

```
DATA: vl_space TYPE c.
```

```
filename = p_infile.
vl_space = cl_abap_char_utilities=>horizontal_tab.
```

```
CALL METHOD cl_gui_frontend_services=>gui_upload
EXPORTING
filename          = filename
has_field_separator = vl_space
CHANGING
data_tab          = data_tab
EXCEPTIONS
file_open_error   = 1

```

```

file_read_error      = 2
no_batch             = 3
gui_refuse_filetransfer = 4
invalid_type         = 5
no_authority         = 6
unknown_error        = 7
bad_data_format      = 8
header_not_allowed   = 9
separator_not_allowed = 10
header_too_long      = 11
unknown_dp_error     = 12
access_denied        = 13
dp_out_of_memory     = 14
disk_full            = 15
dp_timeout           = 16
not_supported_by_gui = 17
error_no_gui         = 18
OTHERS               = 19.

IF sy-subrc <> 0.
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
    WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

ENDFORM.          " UPLOAD_FILE

*&-----*
*& Form TRANSFER_FILE
*&-----*
* text
*-----*
* --> p1 text
* <-- p2 text
*-----*
FORM transfer_file .

DATA: vl_data LIKE LINE OF data_tab.

LOOP AT data_tab INTO vl_data.

CASE vl_data+0(2).
  WHEN '11'.

    IF vg_flag IS INITIAL.
      MOVE: vl_data+0(2) TO vg_input11-i01,
            vl_data+2(4) TO vg_input11-i02,
            vl_data+6(4) TO vg_input11-i03,
            vl_data+10(10) TO vg_input11-i04,
            vl_data+20(6) TO vg_input11-i05,
            vl_data+26(6) TO vg_input11-i06,
            vl_data+32(1) TO vg_input11-i07,
            vl_data+33(14) TO vg_input11-i08,
            vl_data+47(3) TO vg_input11-i09,
            vl_data+50(1) TO vg_input11-i10,
            vl_data+51(23) TO vg_input11-i11,
            vl_data+74(3) TO vg_input11-i12,
            vl_data+77(3) TO vg_input11-i13.
    ELSE.

      MOVE: vl_data+0(2) TO vg_input11-i01,
            vl_data+2(4) TO vg_input11-i02,
            vl_data+6(4) TO vg_input11-i03,
            vl_data+10(10) TO vg_input11-i04,
            vl_data+20(6) TO vg_input11-i05,
            vl_data+26(6) TO vg_input11-i06,
            vl_data+32(1) TO vg_input11-i07,
            vl_data+33(14) TO vg_input11-i08,
            vl_data+47(3) TO vg_input11-i09,
            vl_data+50(1) TO vg_input11-i10,
            vl_data+51(23) TO vg_input11-i11,
            vl_data+74(3) TO vg_input11-i12,
            vl_data+77(3) TO vg_input11-i13.
    ENDIF.

    WHEN '22'.
      MOVE: vl_data+0(2) TO vg_input22-i14,
            vl_data+2(4) TO vg_input22-i15,
            vl_data+6(4) TO vg_input22-i16,
            vl_data+10(6) TO vg_input22-i17,
            vl_data+16(6) TO vg_input22-i18,
            vl_data+22(2) TO vg_input22-i19,
            vl_data+24(3) TO vg_input22-i20,
            vl_data+27(1) TO vg_input22-i21,
            vl_data+28(14) TO vg_input22-i22,
            vl_data+42(10) TO vg_input22-i23,
            vl_data+52(28) TO vg_input22-i24.
    WHEN '23'.
      MOVE: vl_data+0(2) TO vg_input23-i25,
            vl_data+2(2) TO vg_input23-i26,
            vl_data+4(38) TO vg_input23-i27,
            vl_data+42(38) TO vg_input23-i28.

      PERFORM append_line_to_file.

    WHEN '32'.
      MOVE: vl_data+0(2) TO vg_input32-i29,
            vl_data+2(3) TO vg_input32-i30,
            vl_data+5(2) TO vg_input32-i31,
            vl_data+7(35) TO vg_input32-i32,
            vl_data+42(35) TO vg_input32-i33.
    WHEN '33'.
      MOVE: vl_data+0(2) TO vg_input33-i34,
            vl_data+2(4) TO vg_input33-i35,
            vl_data+6(4) TO vg_input33-i36,
            vl_data+10(10) TO vg_input33-i37,
            vl_data+20(5) TO vg_input33-i38,
            vl_data+25(14) TO vg_input33-i39,
            vl_data+39(5) TO vg_input33-i40,
            vl_data+44(14) TO vg_input33-i41,
            vl_data+58(1) TO vg_input33-i42,
            vl_data+59(14) TO vg_input33-i43,
            vl_data+73(3) TO vg_input33-i44.
    ENDCASE.

  ENDLOOP.
ENDFORM.          " TRANSFER_FILE

*&-----*
*& Form CREATE_FILES
*&-----*
* text
*-----*

```

```

* --> p1    text
* <-- p2    text
*-----*
FORM create_files.

TYPES: BEGIN OF ty_line,
       line(1000) TYPE c,
       END OF ty_line.

DATA: vl_line TYPE ty_line.

DATA: vl_t012k TYPE t012k,
       vl_t012  TYPE t012,
       vl_febko TYPE febkko,

       vl_file TYPE string,

       tl_line TYPE STANDARD TABLE OF ty_line,
       tl_file TYPE STANDARD TABLE OF string.

SELECT SINGLE *
FROM t012k
INTO vl_t012k
WHERE bankn EQ vg_input11-i04.

SELECT SINGLE *
FROM t012
INTO vl_t012
WHERE bukrn EQ vl_t012k-bukrn AND
       hbkid EQ vl_t012k-hbkid.

IF sy-subrc EQ 0.

       vg_header-c01 = vl_t012-bankl.    "Clave del banco
       vg_header-c02 = vg_input11-i04.    "Cuenta bancaria

       PERFORM get_next_number USING vg_input11-i04
           CHANGING vg_header-c03.

       CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
           EXPORTING
               input = vg_header-c03
           IMPORTING
               output = vg_header-c03.

       CONCATENATE vg_input11-i05+4(2) cg_point vg_input11-
           i05+2(2) cg_point vg_input11-i05+0(2) cg_point
           INTO vg_header-c04.

*       vg_header-c05 = vg_input11-i09.
       vg_header-c05 = vl_t012k-waers.
       CONCATENATE vg_input11-i08+0(12) cg_point vg_input11-
           i08+12(2) INTO vg_header-c06.
*       vg_header-c06 = vg_input11-i08.
       CONCATENATE vg_input33-i39+0(12) cg_point vg_input33-
           i39+12(2) INTO vg_header-c08.
*       vg_header-c07 = vg_input33-i35.
       IF vg_abonos NE vg_header-c08.
           MESSAGE 'La suma de abonos no coincide con el archivo
               de entrada' TYPE 'E'.
       ENDIF.
       CONCATENATE vg_input33-i41+0(12) cg_point vg_input33-
           i41+12(2) INTO vg_header-c07.
*       vg_header-c08 = vg_input33-i37.
       IF vg_cargos NE vg_header-c07.

```

```

           MESSAGE 'La suma de cargos no coincide con el archivo de
               entrada' TYPE 'E'.
       ENDIF.
       CONCATENATE vg_input33-i43+0(12) cg_point vg_input33-
           i43+12(2) INTO vg_header-c09.
*       vg_header-c09 = vg_input33-i39.
       vg_header-c10 = vg_input11-i11.
       vg_header-c18 = vg_input33-i38 + vg_input33-i40.

       CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
           EXPORTING
               input = vg_header-c18
           IMPORTING
               output = vg_header-c18.

       CONCATENATE: 'C:\Multicash\H\ vl_t012-bankl vg_header-
           c04 '.txt' INTO vl_file,

           vg_header-c01 vg_header-c02 vg_header-c03
           vg_header-c04 vg_header-c05 vg_header-c06
           vg_header-c07 vg_header-c08 vg_header-c09
           vg_header-c10 vg_header-c11 vg_header-c12
           vg_header-c13 vg_header-c14 vg_header-c15
           vg_header-c16 vg_header-c17 vg_header-c18
           INTO vl_line SEPARATED BY cg_colon.

       APPEND vl_line TO tl_line.

       CALL FUNCTION 'GUI_DOWNLOAD'
           EXPORTING
               filename = vl_file
           TABLES
               data_tab = tl_line.

       IF sy-subrc <> 0.
           MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
               WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
       ENDIF.

       CLEAR vl_file.
       CONCATENATE 'C:\Multicash\D\ vl_t012-bankl vg_header-
           c04 '.txt' INTO vl_file.

       CLEAR: vl_line, tl_line.
       LOOP AT tg_detail INTO vg_detail.
           vg_detail-d03 = vg_header-c03.
           CONCATENATE vg_detail-d01 vg_detail-d02 vg_detail-d03
               vg_detail-d04 vg_detail-d05 vg_detail-d06
               vg_detail-d07 vg_detail-d08 vg_detail-d09
               vg_detail-d10 vg_detail-d11 vg_detail-d12
               vg_detail-d13 vg_detail-d14 vg_detail-d15
               vg_detail-d16 vg_detail-d17 vg_detail-d18
               vg_detail-d19 vg_detail-d20 vg_detail-d21
               vg_detail-d22 vg_detail-d23 vg_detail-d24
               vg_detail-d25 vg_detail-d26 vg_detail-d27
               vg_detail-d28 vg_detail-d29 vg_detail-d30
               vg_detail-d31 vg_detail-d32 vg_detail-d33
               vg_detail-d34 vg_detail-d35
               INTO vl_line SEPARATED BY cg_colon.
           APPEND vl_line TO tl_line.
       ENDLOOP.

       CALL FUNCTION 'GUI_DOWNLOAD'
           EXPORTING
               filename = vl_file
           TABLES

```

```

        data_tab = tl_line.
ELSE.
    MESSAGE 'La cuenta no existe' TYPE 'E'.
ENDIF.

ENDFORM.          " CREATE_FILES
*&-----*
*& Form GET_NEXT_NUMBER
*&-----*
*   text
*-----*
*   -->P_RANGE text
*   -->P_OBJECT text
*   -->P_0431 text
*   <--P_VG_HEADR_C03 text
*-----*
FORM get_next_number USING p_cuenta
    CHANGING p_next.

DATA: vl_num TYPE i,
      vl_rango TYPE zfi_tt_rango_cta.

SELECT SINGLE *
FROM zfi_tt_rango_cta
INTO vl_rango
WHERE bankn EQ p_cuenta.

CALL FUNCTION 'NUMBER_RANGE_ENQUEUE'
EXPORTING
    object      = vl_rango-object
EXCEPTIONS
    foreign_lock    = 1
    object_not_found = 2
    system_failure  = 3
    OTHERS          = 4.
IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
        WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

CALL FUNCTION 'NUMBER_GET_NEXT'
EXPORTING
    nr_range_nr      = vl_rango-range
    object           = vl_rango-object
IMPORTING
    number           = vl_num
EXCEPTIONS
    interval_not_found    = 1
    number_range_not_intern = 2
    object_not_found      = 3
    quantity_is_0         = 4
    quantity_is_not_1     = 5
    interval_overflow      = 6
    buffer_overflow        = 7
    OTHERS                 = 8.
IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
        WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

CALL FUNCTION 'NUMBER_RANGE_DEQUEUE'
EXPORTING

```

```

    object      = vl_rango-object
EXCEPTIONS
    object_not_found = 1
    OTHERS          = 2.
IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
        WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

p_next = vl_num.

ENDFORM.          " GET_NEXT_NUMBER
*&-----*
*& Form APPEND_LINE_TO_FILE
*&-----*
FORM append_line_to_file.

DATA: vl_t012k TYPE t012k,
      vl_t012 TYPE t012.

SELECT SINGLE *
FROM t012k
INTO vl_t012k
WHERE bankn EQ vg_input11-i04.
SELECT SINGLE *
FROM t012
INTO vl_t012
WHERE bukrs EQ vl_t012k-bukrs AND
    hbkid EQ vl_t012k-hbkid.

vg_detail-d01 = vl_t012-bankl.
vg_detail-d02 = vg_input11-i04.
vg_detail-d03 = vg_header-c03.
CONCATENATE vg_input22-i17+4(2) cg_point vg_input22-
i17+2(2) cg_point vg_input22-i17+0(2)
    INTO vg_detail-d04.
* vg_detail-d04 = vg_input22-i16.
vg_detail-d06 = vg_input22-i23.
vg_detail-d07 = vg_input22-i19.
vg_detail-d10 = vg_input23-i28.

CONCATENATE vg_input22-i22+0(12) cg_point vg_input22-
i22+12(2) INTO vg_detail-d11.
IF vg_input22-i21 EQ 1.
    vg_abonos = vg_abonos + vg_detail-d11.
    CONCATENATE vg_detail-d11 cg_minus INTO vg_detail-d11.
ELSE.
    vg_cargos = vg_cargos + vg_detail-d11.
ENDIF.
* IF vg_input22-i22 LT 0.
*   CONCATENATE vg_detail-d11 cg_minus INTO vg_detail-
d11.
* ELSE.
*   CONCATENATE vg_detail-d11 cg_plus INTO vg_detail-d11.
* ENDIF.

CONCATENATE vg_input22-i17+4(2) cg_point vg_input22-
i17+2(2) cg_point vg_input22-i17+0(2)
    INTO vg_detail-d14.
* vg_detail-d14 = vg_input22-i16.
vg_detail-d30 = vg_input22-i24.
vg_detail-d34 = vg_input22-i20.

APPEND vg_detail TO tg_detail.

```

```

ENDFORM.                " APPEND_LINE_TO_FILE                field_name = 'P_INFILE'
                                                                IMPORTING
                                                                file_name = p_infile.

*&-----*
*& Include      ZFI_RE_MULTICASH_E01
*&-----*

AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_infile.

CALL FUNCTION 'F4_FILENAME'
EXPORTING
  program_name = syst-cprog
  dynpro_number = syst-dynnr

                                                                *** Solo para pruebas
                                                                PERFORM: create_files.

```

Presentación de un reporte.

Pos.	Cl.	Tp.	GCP	Hist.ped.	Fecha doc.	Proveedor/Centro suministrador	Gpo.artic.	B	P	I	Ce.	Alm.	Cantidad	UMP	Cantidad	UMA	Prc.neto	Mon.	Por	Cantidad	CtdPrevVen	Por entr.	Por entr. Po
Sin material																							
Documento compras 4500000007																							
10	FO	F	100		24.02.2016	300001 Proveedor interno 01	09		K	1000			1	SRV			90.00	MXN	1	0	0	1	0.00
Documento compras 4500000009																							
10	NB	F	100		25.02.2016	300001 Proveedor interno 01	09		F	K	1000		1	UP			100.00	MXN	1	0	0	0	0.00
Documento compras 4500000018																							
10	NB	F	100		26.02.2016	300025 Compras intercía	09		F	K	1000		1	UP			890.00	MXN	1	0	0	0	0.00
Material 1 BAR-7																							
Documento compras 4500000003																							
10	NB	F	100		22.02.2016	300001 Proveedor interno 01	01				1000		10	PI	10	PI	90.00	MXN	1	0	0	10	900.00
Documento compras 4500000004																							
10	NB	F	100		22.02.2016	300001 Proveedor interno 01	01				1010		10	PI	10	PI	90.00	MXN	1	0	0	10	900.00
Documento compras 4500000005																							
10	NB	F	100		22.02.2016	300001 Proveedor interno 01	01				1000		9	PI	9	PI	90.00	MXN	1	0	0	9	810.00
Material 9 Báscula Peso																							
Documento compras 4500000001																							
10	NB	F	100		22.02.2016	300022 Yongkang Donglong Import & Export C	01				1000		9	PI	9	PI	85.00	USD	1	0	0	0	0.00
Documento compras 4500000002																							
10	ZPI	F	100		22.02.2016	300022 Yongkang Donglong Import & Export C	01				1000		10	PI	10	PI	85.00	USD	1	0	0	10	850.00
Documento compras 4500000019																							
10	NB	F	101		29.02.2016	300022 Yongkang Donglong Import & Export C	01				1000		10	PI	10	PI	85.00	USD	1	0	0	0	0.00
Material 118 Báscula Peso-Precio 40kg / 2g Plus																							
Documento compras 4500000010																							
10	NB	F	100		25.02.2016	100000 Dummy	01				1000		9	PI	9	PI	90.00	MXN	1	0	0	0	0.00
Documento compras 4500000011																							

Figura 31: Ejemplo de reporte

Formulario

Petición de oferta

Información del Proveedor
Proveedor interno 01 Almacén Calle de la paz 67124 Distrito federal México

Información
ESTO NO ES UNA ORDEN
Número de Documento: 6000000000
Proveedor: 300001
Grupo de Compras: Punto de Venta
Plazo de la Oferta: 16.02.2016
Fecha de la Oferta: 15.02.2016
Teléfono
Fax
email
Pagina 1 de 1
Por favor regresar la cotización al Depto. de Compras a la siguiente dirección.

Información de la cotización presentada por proveedor:
 Nombre Impreso _____
 Firma _____

Dirección de envío	<div style="border: 1px solid black; width: 80px; height: 25px;"></div>
	MEXICO
Términos de Entrega :	FH (Franco domicilio) / Puesto en Puerto
Términos de pago :	Pagadero inmediatamente sin DPP

Artículo	Material/Descripción	Cantidad	UM	Precio Unitario	Valor Bruto
10	H20 Mercancia,Punto de Reorden, No Serial No. Int . Artículo 4012345500103	9.00	PI		

Instrucciones al Vendedor:
1. ENTREGA - Todos los precios incluidos en esta cotización deben considerar la entrega en FOB, salvo indicación. 2. CONDICIONES DE PAGO - A menos que otro término de pago sea específicamente acordado, las condiciones de pago serán las que se presentan por credenciales del sistema. 3. Alertamos las sugerencias, alternativas o reducción de costos. Cuando presente la alternativa deberá incluir suficiente información para coportar el cambio. 4. IMPUESTOS - A menos que se requiera específicamente, no incluir ningún impuesto en la cotización. 6. GARANTÍA - A menos que se especifique lo contrario por el proveedor, la garantía será de un año desde la recepción del producto o servicio. 8. REQUISITOS Y CONDICIONES DE COMPRA - Son incluidos aquí para referencia. La copia de los términos y condiciones pueden ser solicitados al Departamento de Compras.

Puesto:
 Director de
 Departamento

Nombre:

Depto.:
 Compras

Tel.
 Ext. 101
 Email

Figura 32: Ejemplo de formulario

Fuentes de información.

- Manual TAW10_1 ABAP workbench fundamentals. Manual del instructor. Versión del curso 63. Número de material 50089085. SAP AG 2008.
- https://en.wikipedia.org/wiki/SAP_implementation
- https://en.wikipedia.org/wiki/SAP_ERP
- <http://scn.sap.com/docs/DOC-8032>
- <https://es.wikipedia.org/wiki/ABAP>
- http://help.sap.com/saphelp_nw70/helpdata/en/fc/eb2e8a358411d1829f0000e829fbfe/content.htm
- https://en.wikipedia.org/wiki/SAP_NetWeaver_Process_Integration
- <http://wiki.scn.sap.com/wiki/display/ESpackages/SAP+MII>
- https://en.wikipedia.org/wiki/SAP_CRM
- <http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/d014cef6-37cf-2b10-e8ae-871324d54d8d?QuickLink=index&overridelayout=true>
- <http://scn.sap.com/docs/DOC-55201>
- <http://scn.sap.com/docs/DOC-46251>
- <https://es.wikipedia.org/wiki/Scrum>
- http://help.sap.com/saphelp_45b/helpdata/EN/33/d9da8ab29d11d189ca0000e829fbbd/content.htm
- https://es.wikipedia.org/wiki/Indicador_clave_de_rendimiento
- http://help.sap.com/saphelp_46c/helpdata/en/c5/e4a930453d11d189430000e829fbbd/content.htm
- Documentación del caso de estudio.

Glosario.

- Mandante: Cada una de las instancias creadas dentro del ERP de SAP, cada ambiente (DEV, QAS y PRD) puede tener las instancias necesarias. Generalmente estos mandantes se crean para crear configuraciones, datos maestros y desarrollos de cliente por separado.
- Paquete: Es un contenedor de órdenes de transporte, sirve solamente como un agrupador para tener diferenciados los módulos del ERP y los desarrollos.
- Sistema back end: Es la parte que procesa la entrada de datos que se efectuó desde la interfaz de usuario, es decir, son los procesos que utiliza el administrador del ERP para resolver las peticiones del usuario.
- KPI: Key Performance Indicator por sus siglas en inglés, es una medida del nivel de desempeño de un proceso. El valor del indicador está directamente relacionado con un objetivo fijado de antemano y normalmente se expresan en porcentaje.
- Workflow: Es una herramienta del sistema SAP que se usa para definir procesos que no están mapeados directamente en el ERP, estos pueden ser desde simples liberaciones o procesos de aprobación (principalmente), o procesos de negocio más complejos, tales como crear datos maestros de materiales y la coordinación de departamentos involucrados que se asocia a este proceso. Esta herramienta se utiliza en situaciones en las que los procesos deben ejecutarse varias veces, paralelamente o dependan de eventos en el sistema.
- RICEFW: Reports, Interface, Conversion, Enhancements, Forms and Workflow por sus siglas en inglés. Aplica a todos los módulos, puesto que los desarrollos de estos elementos se lleva a cabo por separado.
- Consultor/equipo basis: Son los miembros del equipo que se encargan de la administración de los sistemas y conexiones, ellos garantizan que los sistemas se encuentren funcionando, se la creación de mandantes, instalación de los ambientes y puesta a punto de los sistemas para que los otros equipos puedan trabajar sobre dichas instancias.
- Consultor/equipo funcional: Miembros encargados de atender y entender las necesidades del cliente, así como de configurar el sistema base para cubrir el mayor número de requerimientos y solicitudes de manera estándar, es decir, sin ayuda de un desarrollo o modificación al sistema.
- Consultor/equipo técnico: Es el grupo de personas que se encargan de crear los desarrollos que cubrirán todas aquellas necesidades del cliente que no se puedan atender de manera estándar, o de modificar el código del sistema para ayudar a cubrir otras.