

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**“ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UNA
INTERFACE PARA EL CONTROL Y GESTIÓN DE
MATERIAS PRIMAS (QUÍMICOS Y COLORANTES)
DESARROLLADO EN ABAP/4 LENGUAJE DE
PROGRAMACIÓN DE SAP”**

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS E INFORMÁTICA

POR: KLEVER MARCELO BENAVIDES CALLE

SANGOLQUÍ, 15 de octubre de 2012

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. KLEVER MARCELO BENAVIDES CALLE como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

Sangolquí, 09 de Octubre de 2012

INGENIERO MAURICIO CAMPAÑA

DIRECTOR

DEDICATORIA

Dedico este trabajo a mis amadas esposa e hija María José y Mía Isabella quienes han sido mi fuerza motivadora para culminar este proyecto, a mi madre Rosita y mi padre Hernando, ambos ejemplo de trabajo incansable y mi único apoyo durante el transcurso mi vida, a mi hermano David el mejor amigo que tengo y del cual espero cumpla todas sus metas, del mismo modo a mis hermanos Xavier y Marco a los que quiero con todo mi corazón, por ultimo pero no menos importantes a mis sobrinos hermosos Camila y Matías que con sus ocurrencias iluminan mi vida.

KLEVER MARCELO BENAVIDES CALLE

AGRADECIMIENTOS

Agradezco a Dios por darme la salud y capacidad necesarias, a mis padres Rosita y Hernando quienes han luchado y sufrido todos estos años por ver este trabajo finalizado. A mis muñecas hermosas María José y Mía Isabella que son el combustible que impulsa mi vida y por quienes lucho cada día. A mi Director y Codirector, los ingenieros Mauricio Campaña y Carlos Caizaguano que con su orientación, dedicación y exigencia me permitieron sacar adelante esta tesis.

KLEVER MARCELO BENAVIDES CALLE

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN	II
DEDICATORIA.....	III
AGRADECIMIENTOS	IV
ÍNDICE DE CONTENIDOS	V
ÍNDICE DE FIGURAS	VII
RESUMEN.....	VIII
CAPÍTULO 1	2
1 GENERALIDADES	2
1.1 INTRODUCCIÓN.....	2
1.2 JUSTIFICACIÓN E IMPORTANCIA	4
1.3 OBJETIVOS.....	6
1.3.1 Objetivo General	6
1.3.2 Objetivos Específicos	6
1.4 ALCANCE.....	6
1.5 METODOLOGÍA	8
1.6 DESCRIPCIÓN DE LAS FASES.	11
1.7 FACTIBILIDAD DEL PROYECTO	11
Recurso Técnico.....	11
Recursos Económicos.....	13
Recursos Operativos	14
CAPÍTULO 2	15
2 MARCO TEÓRICO	15
2.1 GENERALIDADES ACERCA DEL SISTEMA SAP R/3.....	15
2.1.1 Historia	15
2.1.2 Principales Características.....	16
2.1.3 Arquitectura del sistema.....	17
2.1.4 Desarrollos dentro de SAP	19
2.2 MÓDULOS SAP	20
2.3 ¿QUÉ ES ABAP/4?	20
2.4 ESTRUCTURA DE UN PROGRAMA ABAP/4	23
2.4.1 Como crear un programa en ABAP/4	23
2.5 TRANSACCIONES	28
2.5.1 Estructura de un programa de diálogo.....	29
2.5.2 Atributos de los campos.....	31
2.5.3 Diálogos de error	31
2.6 DYNPRO	32
2.6.1 Atributos de la pantalla	32
2.6.2 Formato	33
2.6.3 Lógica de flujo	33
2.7 ASIGNACIÓN DEL CÓDIGO DE TRANSACCIONES A LOS PROGRAMAS	33
2.8 MANEJO DE DATOS.....	34
2.8.1 Declaración de datos.....	34
2.9 CONTROL DE FLUJO	39

2.9.1	Programación de expresiones lógicas	40
2.9.2	Programación de ramificaciones y bucles	41
2.10	TÉCNICAS ESPECIALES DE PROGRAMACIÓN	42
2.10.1	Creación y procesamiento de tablas internas	42
2.10.2	Propósito de las tablas internas	42
2.10.3	Estructura de las tablas internas	44
2.10.4	Trabajando con tablas internas.....	44
2.11	MODULARIDAD DE LOS PROGRAMAS ABAP/4	46
2.11.1	Módulos de código fuente	47
2.11.2	Definición de subrutinas	47
2.11.3	Invocando subrutinas internas	48
2.11.4	Módulos de función	49
CAPÍTULO 3		51
3. METODOLOGÍA XP (Programación Extrema)		51
3.1	PLANIFICACIÓN	51
3.1.1	Especificación de requerimientos	51
3.1.2	Descripción general	55
3.1.4	Plan de Entregas.....	58
3.1.5	Establecimiento de objetivos	59
3.1.6	Iteraciones	59
3.1.7	Rotaciones	60
3.1.8	Reuniones	60
3.2	DISEÑO.....	61
3.2.1	Seleccionar tipo de producto de conocimiento.....	61
3.2.2	Soluciones Puntuales	63
3.2.3	Funcionalidad Mínima	63
3.3	DESARROLLO	64
3.3.1	Flujo de trabajo	64
3.3.2	Interfaces de usuario.....	65
3.3.3	Plantillas	67
3.3.4	Programación.....	68
3.3.5	Integración	69
3.4	PRUEBAS	69
3.4.1	Procedimiento de Manejo	69
3.4.2	Pruebas al sistema	69
3.4.3	Pruebas de aceptación.....	69
CAPÍTULO 4		71
4. CONCLUSIONES Y RECOMENDACIONES		71
4.1.	CONCLUSIONES.....	71
4.2.	RECOMENDACIONES	72
4.3.	ANEXO A: INSTRUCTIVO DE INSTALACIÓN DE SAP GUI PARA WINDOWS	72
4.4.	ANEXO B: INSTRUCTIVO DEL PROGRAMADOR	72
BIBLIOGRAFIA		73
GLOSARIO.....		¡Error! Marcador no definido.

ÍNDICE DE FIGURAS

Figura 1.1 Fases de la Metodología XP	10
Figura 2.1 Características SAP	16
Figura 2.2 Arquitectura de sistema SAP	18
Figura 2.3 Módulos del sistema SAP R/3	20
Figura 2.4 Menu Workbench ABAP	23
Figura 2.5 Definición Programa ABAP	24
Figura 2.6 Crear Programa ABAP	24
Figura 2.7 Portal de registro SAP	25
Figura 2.8 Clave de registro SAP	25
Figura 2.9 Atributos de programa ABAP.....	26
Figura 3.1 Plan de entregas de Historias de Usuario.....	59
Figura 3.10 Ventana de transacción ZENVIO_INTERFACES.....	66
Figura 3.11 Ventana de datos para envío.....	67
Figura 3.2 Plan de Iteraciones.....	60
Figura 3.3 Flujo de trabajo.....	64
Figura 3.4 Etiqueta de Texto.....	65
Figura 3.5 Caja de Texto	65
Figura 3.6 Radio Botones	65
Figura 3.7 Check Box.....	65
Figura 3.8 Botón	66
Figura 3.9 Data Grids	66
Figura A.1: Marketplace Descarga SAP GUI	75
Figura A.10: Ingreso de datos de usuario.....	79
Figura A.11: Pantalla inicial SAP	80, 83, 84
Figura A.2: Icono de instalación	75
Figura A.3: Inicio de la instalación	76
Figura A.4: Selección de componentes a instalar.....	76
Figura A.5: Proceso de instalación.....	77
Figura A.6: Fin de instalación	77
Figura A.7: Inicio de Sap Logon.....	78
Figura A.8: Parámetros de conexión.....	78
Figura A.9: Conexión a Sap Jersey	79
Figura B.1: Unidad de Red Colorantes.....	83
Figura B.2: Unidad de Red Químicos Líquidos	84
Figura B.3: SAP GUI Ingreso Transacción SE38	84
Figura B.4: Modificación de un programa ABAP	85
Figura B.5: Interfaz ZENVIO_INTERFACES.....	86
Figura B.6: Interfaz ZPP_01	92
Figura B.7: Interfaz Z_INTERFAZ_SOLIDOS.....	97
Figura B.7: Interfaz Z_INTERFAZ_SOLIDOS I.....	97
Figura B.8: Interfaz Z_INTERFAZ_SOLIDOS II.....	97, 102
Figura B.9: Interfaz de entradas Manuales	102

RESUMEN

En el desarrollo del presente proyecto se puede encontrar una completa descripción de una de las principales metodologías usadas en la desarrollo de sistemas de software, esta es la programación Extrema (XP), la principal característica de este tipo de metodología es el enfoque que da al usuario, así como la manera de capturar los requerimientos, la realización del diseño del software y el manejo de las pruebas unitarias.

Se podrá observar también como el proyecto se basa en el proceso metodológico que establece la Programación Extrema, para crear una solución que permitirá la mejora del proceso de tintura dentro de la empresa S.J. Jersey Ecuatoriano, con el fin de encontrar los diferentes componentes lógicos como son; cadenas de conexión, capa de datos, capa de negocios, que pueden servir de base para la creación de nuevos proyectos.

Además encontrará la manera de implementar y automatizar una de las características más importantes de Programación Extrema como son el uso de las Pruebas Unitarias.

CAPÍTULO 1

1 GENERALIDADES

1.1 INTRODUCCIÓN

Desde algún tiempo atrás y en la actualidad, con el fin de afrontar las necesidades que surgen en el núcleo de las empresas para lograr una productividad más eficiente y eficaz, existen una gran cantidad de alternativas de software que se aplica a toda la cadena que recorre un producto. Una de las más utilizadas, y en las que se basan otros sistemas, es el denominado SAP R/3, cuyos orígenes se remontan a Alemania.

El sistema SAP en Ecuador y mundialmente es muy conocido, pero debido a sus elevados costos resulta casi imposible para una empresa adquirirlo, es de conocimiento general que SAP es uno de los líderes en ERP, sin embargo en estos tres últimos semestres SAP ha brindado facilidades para que medianas empresas puedan adquirir su sistema en una campaña de masificación del producto y en Ecuador algunas compañías han adquirido SAP como sistema principal, abriendo así campo laboral a consultores y para que el sistema sea conocido a profundidad, además de que su lenguaje de programación ABAP/4 sea difundido y muy tomado en cuenta por los desarrolladores.

El SAP R/3 es un sistema de tipo ERP (Enterprise Resource Planning) que ha sido diseñado en base a una plataforma que ofrece una gran versatilidad de

programación, facilidad de uso, y precisión en el manejo total de los datos recolectados.

Su nombre se remite a dos factores importantes que caracterizan a dicho software. En principio, la R se refiere al procesamiento en tiempo real, y el número 3 a los tres niveles de la arquitectura de procesos con los que trabaja: bases de datos, servidor de aplicaciones y cliente.

Mediante este interesante sistema de gestión es posible realizar un control exhaustivo de todos los procesos y operaciones que se realizan dentro de una compañía, almacenando la información que puede ser utilizada por cualquier sector de la organización.

El sistema R/3 fue diseñado y este se encuentra dividido en módulos claramente diferenciados, los cuales corresponden a las necesidades propias de cada una de las áreas dentro de una empresa.

Dentro de cada una de la organizaciones es necesario automatizar ciertos procesos de negocio, los cuales son críticos y en ciertos casos son peligrosos cuando la mano del hombre interviene, ya que puede causar errores de digitación o a veces suele pasar que simplemente por un olvido por parte del operario un proceso no se alimenta con la información adecuada y en tiempo real.

Son en estas situaciones cuando un desarrollo a medida es requerido y SAP como un sistema completo e integrado nos da la posibilidad de modificar o desarrollar dentro su código fuente en su lenguaje propio de programación (ABAP/4). Los desarrollos que se realizan para una empresa determinada pueden servir para muchas otras y así ahorrar tiempo en desarrollar repetitivamente lo mismo.

1.2 JUSTIFICACIÓN E IMPORTANCIA

Las empresas en el Ecuador, especialmente las de grande y mediano tamaño, vienen demandando cada vez más, un manejo eficaz y eficiente de todos sus procesos ya sean estos críticos o no. Estos procesos debido a su complejidad o manejo son necesarios automatizarlos con el fin de evitar varios problemas comunes que por intervención de la mano del hombre no se llevan de una manera adecuada, es por esta razón resulta necesario llevar un control total de los descargos y dosificaciones de materias primas, este no es solo el problema de una empresa, al contrario varias empresas padecen el mismo, con lo cual este planteamiento de solución es aplicable a cualquier empresa de producción dentro del Ecuador o del mundo que cuente con SAP como ERP.

Por otro lado, como alternativa de solución al rápido crecimiento de las empresas y la necesidad de tener una mejor administración de recursos y procesos en este mundo globalizado, vienen surgiendo herramientas como los sistemas ERP (Sistemas de Planificación de Recursos Empresariales) que ayudan a las empresas a coordinar mejor sus activos, procesos y recursos.

Entre todas ellas, evidentemente el sistema que mayor éxito ha alcanzado a nivel mundial es el SAP (Sistemas, Aplicaciones y Productos), cuyo principal producto de la compañía es el R/3, que permite controlar todos los procesos que se llevan a cabo en un empresa a través de sus diferentes módulos: finanzas, ventas y distribución, contabilidad, gestión de materiales, planeamiento de producción, recursos humanos y gestión de almacenes.

Sin embargo, el SAP R/3, específicamente en el módulo de Planificación de la producción (PP), no cuenta con una sección para la gestión de químicos y colorantes, por cuanto no prevé la capacidad de control de estas dosificaciones y descargos de materiales de las bodegas de inventario. En ese sentido, resulta necesario realizar adaptaciones del Sistema SAP R/3 a este tipo de realidades, si se quiere contribuir a una mejor administración y gestión de materias primas usadas en la elaboración de un producto.

Por los motivos mencionados anteriormente, el tema de este plan de tesis propone una alternativa para la gestión adecuada de materias primas (Químicos y colorantes, etc.), dentro del sistema SAP, adaptada a la realidad de cualquier empresa, ya sea esta grande como mediana, tomando en cuenta que este puede ser usado en cualquier empresa del Ecuador y el mundo que cuente con básicas características de producción.

Este proyecto además tiene la finalidad de elaborar la tesis para obtener el título de Ingeniería en Sistemas e Informática, mediante el desarrollo de una aplicación en lenguaje ABAP/4 propietario de SAP.

1.3 OBJETIVOS.

1.3.1 Objetivo General

Analizar, diseñar, desarrollar e implementar una interfase con SAP, utilizando el lenguaje de programación ABAP/4.

1.3.2 Objetivos Específicos

- Revisar los conceptos relacionados con SAP y ABAP/4.
- Aplicar la metodología XP para el análisis y diseño de la interfase con SAP.
- Aplicar la metodología XP para el desarrollo e implementación de la interfase con SAP.
- Realizar las pruebas de manejo y aceptación de la interfase con SAP.

1.4 ALCANCE.

El principal fin de este desarrollo es brindar una herramienta que ayude a las empresas al control y la gestión de materias primas (Químicos, colorantes, etc.), es por eso que para este proyecto se ha dividido estas en tres grandes grupos que son:

- Químicos sólidos.
- Químicos líquidos.
- Colorantes.

Para alcanzar los objetivos descritos anteriormente, se va a proceder a investigar y analizar la información concerniente al proceso de dosificación y descargo de inventario de materias primas, para luego diseñar y construir la aplicación, mediante la utilización de herramientas de diseño y desarrollo que SAP provee dentro de su core de soluciones.

Utilizar las herramientas de desarrollo que están incluidas como transacciones dentro de SAP, por este motivo no es necesaria la instalación de una herramienta de desarrollo independiente.

Durante las fases de cumplimiento del presente proyecto se va a recopilar toda la información necesaria, la cual será utilizada posteriormente en el desarrollo y la realización de pruebas del mismo, todo esto bajo parámetros y estándares de desarrollo, garantizando así la obtención de un sistema de calidad.

Las interfaces de usuario GUI a utilizarse en esta solución serán amigables y estarán definidas de acuerdo a los estándares que SAP utiliza para la presentación al usuario final.

Sera necesario definir unas nuevas tablas dentro de la base de datos de SAP, además de la creación de reportes e impresiones de etiquetas.

Dentro de la interfaz principal, será necesario desarrollar sub-interfaces para las materias primas antes mencionadas, las cuales enviarán y recibirán información de SAP.

1.5 METODOLOGÍA

SAP utiliza dos metodologías de desarrollo, Rational Unified Process (RUP) para proyectos medianos y grandes, y Extreme Programming (XP) para el caso de los proyectos de corta duración.

A continuación se presenta una breve reseña de ambas metodologías de desarrollo:

- **Rational Unified Process (RUP)**

RUP es un proceso que define claramente quien, cómo, cuándo y qué debe hacerse y, como su enfoque está basado en modelos utiliza un lenguaje bien definido para tal fin, el UML. Éste aporta herramientas como los casos de uso, que definen los requerimientos y permiten la ejecución iterativa del proyecto y del control de riesgos.

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software. Cada Fase tiene definido un conjunto de objetivos y un punto de control específico, como se puede ver en la tabla 1.1

Tabla 1.1 Fases de la Metodología RUP

Fase	Objetivos	Puntos de Control
Inicio	Definir el alcance del proyecto	Objetivo del proyecto
Elaboración	Construir una versión ejecutable de la arquitectura de la aplicación	Arquitectura de la aplicación
Construcción	Completar el esqueleto de la Aplicación con la funcionalidad	Versión Operativa inicial de la Aplicación
Transición	Disponibilidad de la Aplicación para usuarios finales	Liberación de la versión de la Aplicación

- **Programación Extrema (XP)**

Es una de las metodologías de desarrollo de software más exitosas en los últimos años utilizadas para proyectos de corto plazo, equipo y cuyo plazo de entrega es el menor tiempo posible.

La Programación Extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado.

Las cuatro variables

- Coste: Máquinas, especialistas y oficinas.
- Tiempo: Total y de Entregas.
- Calidad: Externa e Interna.
- Alcance: Intervención del cliente.

Fases de la Metodología XP (Extreme Programming)

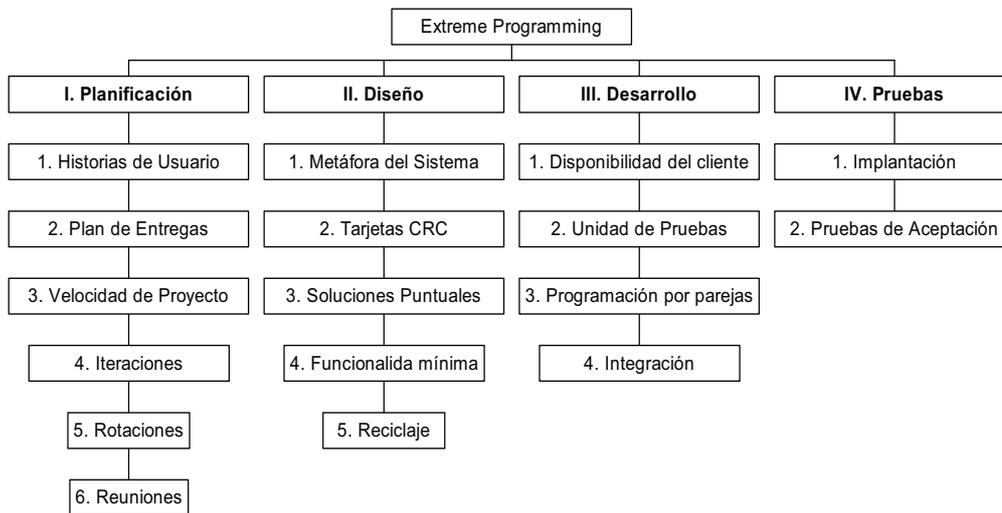


Figura 1.1 Fases de la Metodología XP

Para el desarrollo de este proyecto se va a utilizar la metodología XP (Extreme Programming), debido a que no es un proyecto extremadamente grande como para usar RUP, además la metodología XP es la que mas similitud tiene respecto a las metodologías de desarrollo tradicionales, ver la Figura 1.1

1.6 DESCRIPCIÓN DE LAS FASES.

CAPÍTULO 1. GENERALIDADES: Muestra una breve descripción del proyecto y los objetivos del mismo.

CAPÍTULO 2. MARCO TEÓRICO: Presenta todo lo referente al concepto de SAP como ERP y su arquitectura, además de una introducción al lenguaje de programación ABAP/4 propio de SAP.

CAPÍTULO 3. METODOLOGÍA XP (Extreme Programming): Muestra el análisis necesario para la creación del sistema utilizando la información recopilada por diferentes medios y se realiza el diseño utilizando esta metodología.

CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES: Presenta las conclusiones alcanzadas y las diferentes recomendaciones que ayudaran al desarrollo de este tipo de aplicaciones, por otro lado se muestra la diversas fuentes bibliográficas utilizadas en el desarrollo del proyecto.

1.7 FACTIBILIDAD DEL PROYECTO

Recurso Técnico

Para este proyecto, se cuenta con la información necesaria adquirida por el autor en un proyecto de implementación de SAP, además se contará con el apoyo de profesionales de la ESPE en lo referente a la parte técnica y metodológica.

Hardware:

CANTIDAD	IDENTIFICACIÓN	DESCRIPCIÓN
1	Computador (Servidor SAP)	Core 2 Duo 2.8 Ghz, 32GB RAM a 1067 MHz, disco duro 500GB, DVD-RW, teclado, mouse, monitor, tarjeta de red.
1	Computador (Portátil)	Core 2 Duo 2.8, 4GB RAM, Disco 200GB, DVDRW
1	Impresora	Epson 20 ppm

Tabla 1.2 Recurso Técnico Hardware

Software:

El software a instalar en las computadoras para el desarrollo de proyecto será:

- Microsoft Windows 7.
- Microsoft Office 2007
- SAP GUI 7.20
- SAP R/3

Recursos Económicos

Hardware:

IDENTIFICACIÓN	CANTIDAD	VALOR	TOTAL
Servidor SAP(Jersey)	1	0	0
Computador (Portátil)	1	2500	2500
Impresora Epson CX5400	1	150	150
Subtotal Hardware:			2650

Tabla 1.3 Recurso Económico Hardware

Software:

IDENTIFICACIÓN	CANTIDAD	VALOR	TOTAL
SAP R/3 Versión 6.0	1	Sin Costo	0
Subtotal Software:			0

Tabla 1.4 Recurso Económico Software

Humano:

CONCEPTO	TIEMPO (MESES)	SUELDO MENSUAL	TOTAL
1 Egresado de Ingeniería en Sistemas	4	Sin Costo	0
Subtotal Humano:			0

Tabla 1.5 Recurso Económico Humano

Otros:

CONCEPTO	CANTIDAD	VALOR	TOTAL
Conexión a Internet (Banda Ancha)	5 Meses	20	100
Papel	3 (Resma)	6	18
Medios Magnéticos	10	0.50	5
Subtotal Otros:			123

Tabla 1.6 Recurso Económico Otros

El costo total del proyecto a realizar es de **\$2773 dólares**, el cual será cubierto en su totalidad por el autor del mismo.

Recursos Operativos

Para el desarrollo del Proyecto de Tesis propuesto se encargará el autor: Klever Marcelo Benavides Calle, egresado de la Facultad de Sistemas e Informática de la Escuela Superior Politécnica del Ejército, por otra parte se contará con la ayuda de Ingenieros especializados en desarrollo de sistemas, los cuales intervendrán en calidad de Director y Codirector de Tesis respectivamente.

CAPÍTULO 2

2 MARCO TEÓRICO

2.1 GENERALIDADES ACERCA DEL SISTEMA SAP R/3

SAP es la principal empresa de software empresarial del mundo. Cuenta con más de 29.800 clientes en más de 120 países, 12.000 millones de usuarios, 100.600 instalaciones de software y más de 1.500 partners, tal es así que se ha convertido en el tercer proveedor de software independiente más importante a nivel mundial.

2.1.1 Historia

SAP fue fundada el 01 de Abril de 1972 por cinco personas: Wellenreuther, Hopp, Hector, Plattner y Táchira, cinco programadores de IBM que decidieron apostar por una idea nueva. Siendo empleados en la empresa multinacional IBM habían desarrollado un paquete de contabilidad financiera que funcionaba en bloques para un cliente de IBM (Naturin), como el producto funcionaba muy bien solicitaron a la compañía que la adoptara y promoviera como una aplicación de la empresa pero no aceptaron. De esta manera los 5 colegas decidieron emprender su camino de manera independiente y luchar por lo que creían. “Nos fuimos con el único objetivo de desarrollar software estándar”. SAP compró los derechos a Naturin y empezó con el diseño e implementación de un sistema financiero en tiempo real. Simultáneamente desarrollaron un sistema de administración de materiales para una empresa cliente, pero esta vez se reservaron los derechos de propiedad para SAP. Con el dinero obtenido llegaron

a culminar con el desarrollo del sistema financiero contable. Posteriormente, el sistema de administración de materiales se convirtió en un paquete estándar. Estos dos sistemas desarrollados se constituyeron en los primeros módulos de lo que se llamó el sistema R, que posteriormente se renombró R/1 para distinguirlo mejor de sus sucesores R/2 y R/3. La letra "R" viene de 'real-time data processing' (procesamiento de datos en tiempo real) que es la base de la filosofía SAP.

2.1.2 Principales Características

En la figura 2.1, se muestra las principales características del sistema SAP, que a continuación se explican.

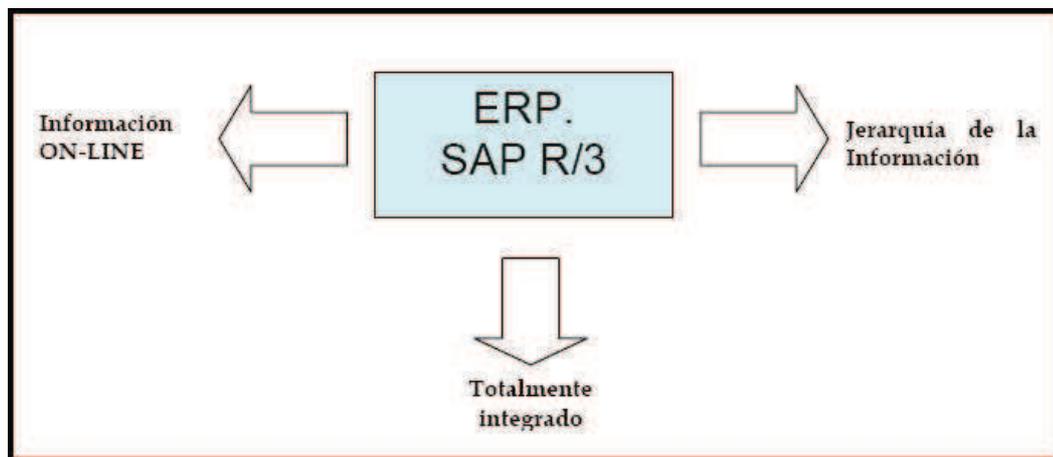


Figura 2.1 Características SAP

- **Información "on-line".**

Esta característica significa que la información se encuentra disponible al momento, sin necesidad de esperar largos procesos de actualización y procesamiento habituales de otros sistemas.

- **Jerarquía de la información.**

Gracias a esta forma de organizar la información, se tiene la posibilidad de obtener informes desde diferentes vistas.

- **Integración.**

Es la característica más destacable e importante de SAP R/3 y significa que la información se comparte entre todos los módulos que lo requieran; es decir, la información circula a través de todas las áreas de la empresa.

Esta integración se logra por la alimentación de una base de datos común, por lo tanto hay que tener en cuenta que toda información que se introduce repercutirá en tiempo real, a todos los demás usuarios que cuenten con acceso a la misma.

El sistema R/3 está formado por distintos módulos que funcionan de forma integrada. Además, dispone de su propio lenguaje de programación de cuarta generación denominado ABAP/4.

2.1.3 Arquitectura del sistema

La arquitectura que se describe a continuación es una fusión y resumen de varios capítulos de los libros titulados *Implementación de SAP R/3* [IMP 2000] y *Edición especial SAP R/3* [EES 1999], que se toman como referencia.

El sistema SAP R/3 está basado en un diseño sólido que utiliza el principio de software modular cliente/servidor de varios niveles y sus componentes arquitectónicos son los procesos y módulos de software capaces de ofrecer servicios de cliente/servidor y de aplicaciones.

El software básico de R/3 es el conjunto de programas y herramientas que funciona como interfaz con el sistema operativo de la computadora, con la base de datos, con los protocolos de comunicaciones y con las interfaces de presentación. Este software permite que las aplicaciones de R/3 (FI, CO, SD, PP, etc.) funcionen exactamente del mismo modo, independientemente del sistema operativo o de la base de datos que estén instalados en el sistema. El software básico de R/3 es una capa independiente que garantiza la integración técnica de todos los módulos de aplicación. En la figura 2.2 se muestra la Arquitectura del Sistema SAP.

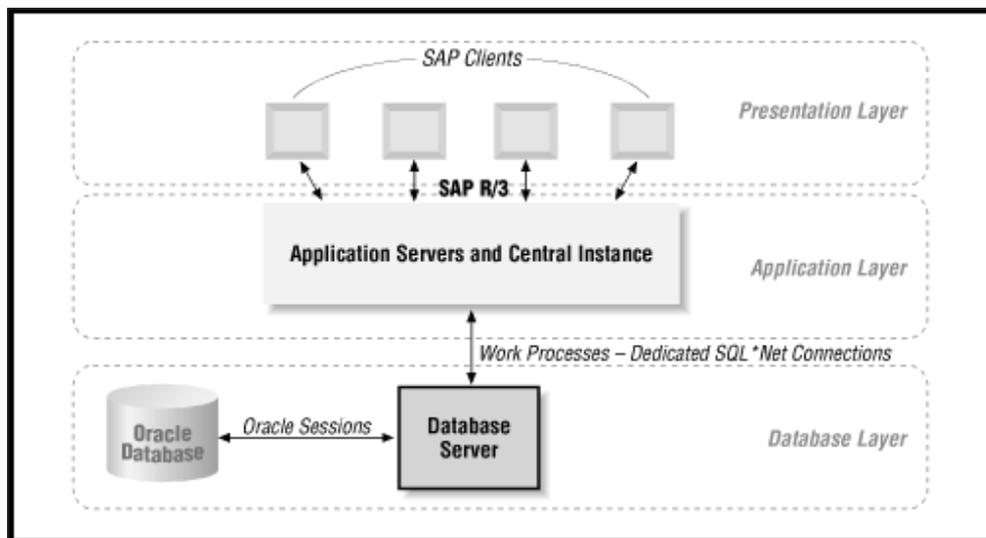


Figura 2.2 Arquitectura de sistema SAP

2.1.4 Desarrollos dentro de SAP

ABAP Workbench es un conjunto plenamente integrado de herramientas de desarrollo, funciones, lenguaje de programación y diccionario de datos. Está diseñado como un entorno de desarrollo que cubre todas las fases de un proyecto de desarrollo. Cuenta con las herramientas necesarias para transferir con facilidad desarrollos entre distintos sistemas, haciendo que las aplicaciones de R/3 sean completamente portables y de esa manera aseguran la integridad de la información. La integración de todos los componentes del entorno de desarrollo implica que los cambios realizados en cualquier parte tienen un efecto directo e inmediato en todas las aplicaciones que utilicen dichos componentes. El entorno contiene además una importante biblioteca de funciones de gestión, con ella los clientes pueden encontrar información bastante precisa, teniendo en cuenta sus necesidades específicas.

El lenguaje de desarrollo de SAP se denomina ABAP que es el acrónimo de “Advanced Business Application Programming” (programación avanzada para aplicaciones de gestión) y es un lenguaje de cuarta generación, por lo que su nombre completo es ABAP/4 sin embargo, debido a que ha incorporado muchas características de los lenguajes orientados a objeto se ha quedado simplemente con el nombre de ABAP.

ABAP es el lenguaje de programación que SAP ha utilizado para desarrollar todos los módulos y aplicaciones de negocio de R/3 incluyendo las funciones de administración del sistema.

Precisamente, gracias a él, los clientes y diseñadores pueden ampliar las funciones de SAP y adaptarlas a sus necesidades particulares a través de dicho lenguaje.

2.2 MÓDULOS SAP

Los módulos de los que consta SAP R/3 se muestran en la figura 2.3:

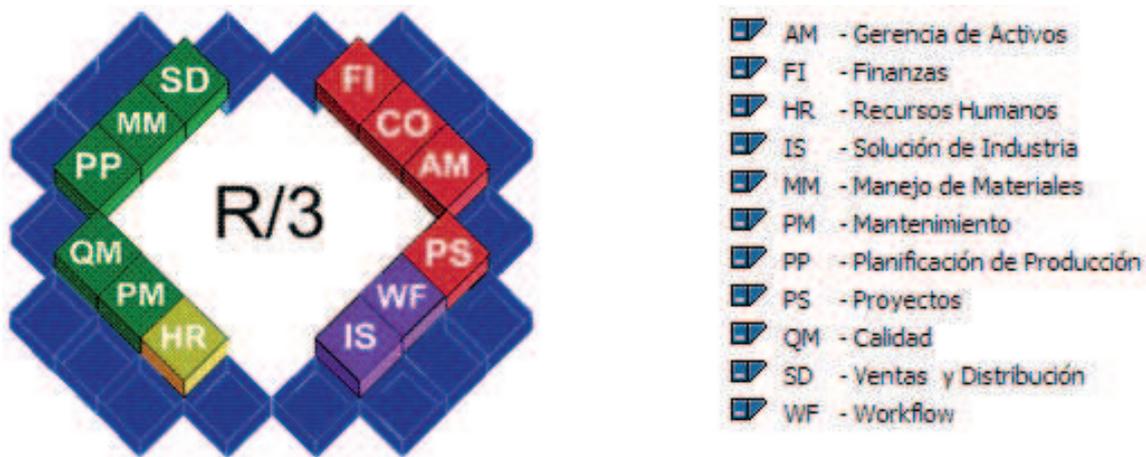


Figura 2.3 Módulos del sistema SAP R/3

2.3 ¿QUÉ ES ABAP/4?

ABAP/4 es un lenguaje de cuarta generación que soporta programación estructurada. ABAP/4 incorpora todas las estructuras de control y conceptos de modularización más usuales.

Las funciones básicas de ABAP/4 incluyen lo siguiente:

- Elementos para declarar datos con varios tipos y estructuras.
- Elementos operacionales para la manipulación de datos.
- Elementos de control para controlar el flujo del programa.
- Elementos de eventos para poder reaccionar a eventos externos.

ABAP/4 soporta varios lenguajes. Los elementos de tipo texto (por ejemplo títulos, encabezados y otros textos) se almacenan en forma separada del código del programa.

ABAP/4 soporta varios tipos de datos y operaciones comunes en los negocios. Es posible efectuar cálculos con campos especiales de fecha y hora, el sistema ejecuta todas las conversiones de tipo que sean necesarias en forma automática.

ABAP/4 provee un rango de funciones para el procesamiento de estructuras de caracteres a strings. También contiene una parte de SQL llamada Open SQL. Open SQL permite leer y acceder tablas de bases de datos sin importar qué sistema de bases de datos se esté utilizando.

ABAP/4 permite la definición y proceso de tablas internas que existen solo mientras el programa se está ejecutando. Estas tablas internas hacen más fácil trabajar con tablas de bases de datos y usar las estructuras complejas de datos en un programa.

ABAP/4 permite definir y llamar subrutinas. Asimismo, es posible hacer llamadas a las subrutinas de otros programas. Los parámetros se pueden pasar de/a las subrutinas de maneras diferentes.

ABAP/4 contiene una clase especial de subrutina conocida como un módulo de función. Se pueden crear y mantener módulos de función en una librería central. Los módulos de función tienen una interfaz de datos claramente definida entre el programa que los invoca y la subrutina. Estos pueden probarse de modo independiente del programa que los invoca.

Existen dos tipos principales de programas de ABAP/4:

- **Programas de Reportes**

Los programas de reportes son utilizados para analizar datos de tablas de bases de datos. Los resultados de tales análisis pueden desplegarse en una pantalla o enviarse a una impresora. Los programas de reportes son soportados por bases de datos lógicas, que no son más que programas especiales de ABAP/4 que evitan a los programadores tener que codificar todos los accesos a la base de datos.

- **Programas de Dialogo**

Los programas de diálogo están organizados como bancos de módulos, los cuales contienen módulos de diálogos. Cada "programa dinámico" o dynpro (Dynamic Process), el cual consiste de una pantalla y la lógica de su flujo está basado exactamente en un programa de diálogo de ABAP/4.

2.4 ESTRUCTURA DE UN PROGRAMA ABAP/4

2.4.1 Como crear un programa en ABAP/4

La creación de un programa sencillo en ABAP/4 incluye los siguientes pasos básicos:

1. Dar un nombre al programa.
2. Especificar los atributos del programa.
3. Escribir el código del programa.
4. Probar el programa.

Para crear los módulos para una nueva transacción, se puede proceder de una manera diferente que para un programa de reportes, así:

a. Dar un nombre al programa

Para crear un programa en ABAP/4, se debe proceder como sigue:

- I. Escoger Herramientas ABAP/4 Workbench en la pantalla inicial en R/3 de SAP. Aparece, Entonces, la pantalla de Herramientas de Desarrollo ABAP/4, ver figura 2.4

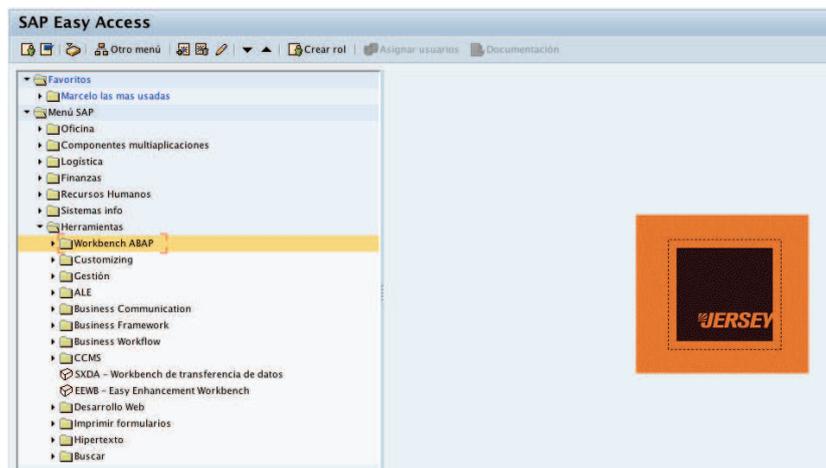


Figura 2.4 Menú Workbench ABAP

Escoger el Editor de ABAP/4. La pantalla inicial del Editor de ABAP/4 aparece como se muestra a continuación en al figura 2.5:

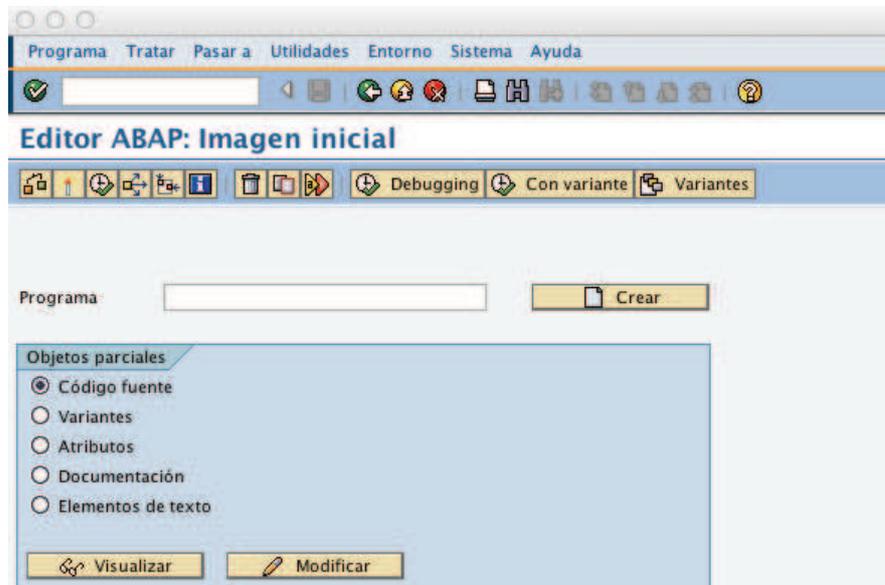


Figura 2.5 Definición Programa ABAP

- II. Ingresar un nombre para el programa que se desea crear en el campo Programa.
- III. Escoger crear, ver figura 2.6

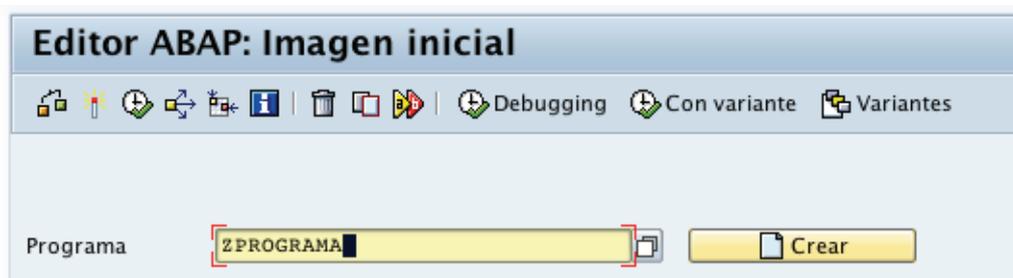


Figura 2.6 Crear Programa ABAP

Es importante tomar en cuenta que para la creación de un programa se debe registrar el objeto en la página de SAP <https://websmp102.sap-ag.de/support> allí nos dará la clave de registro para empezar a crear nuestro programa, ver figura 2.7 y 2.8

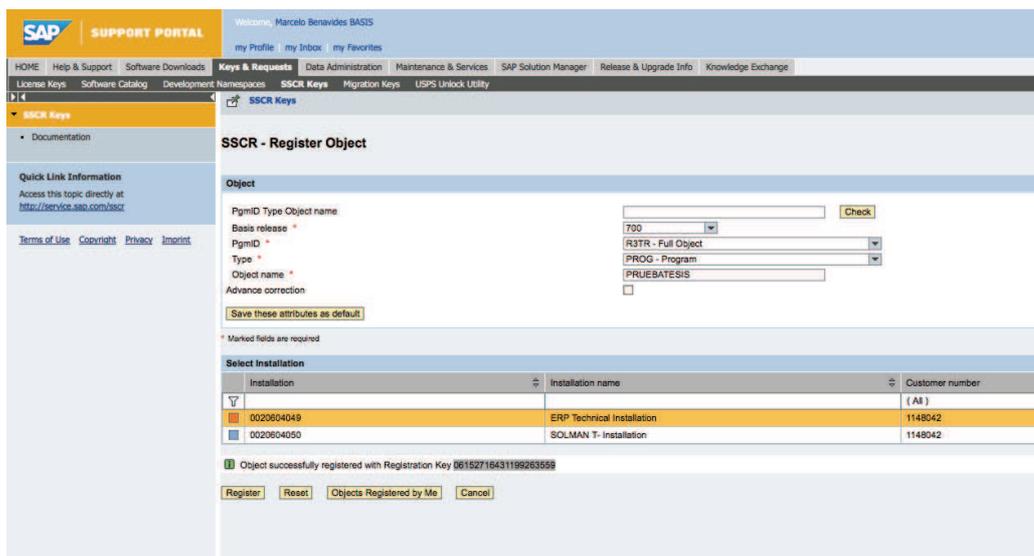


Figura 2.7 Portal de registro SAP

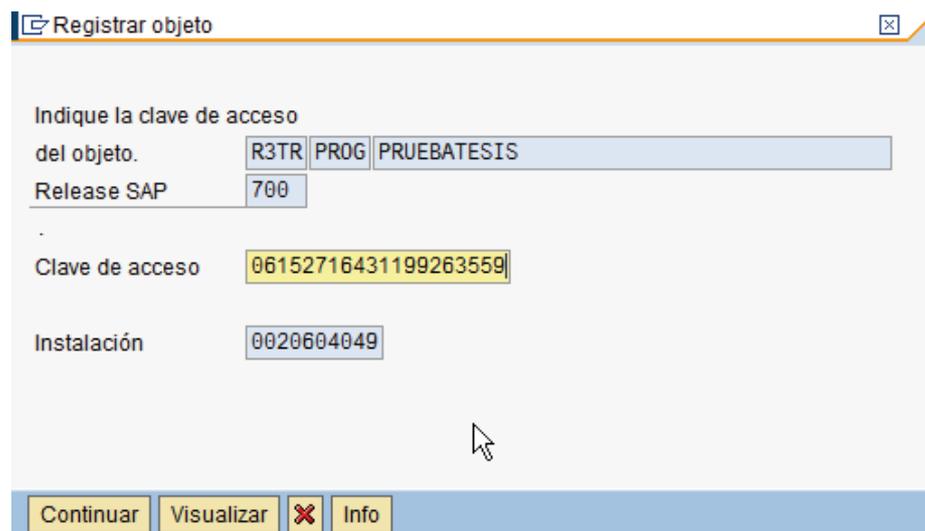


Figura 2.8 Clave de registro SAP

b. Especificar los atributos del programa

Los atributos de un programa determinan a que aplicación pertenece un programa, así como la base de datos lógica a la cual esta enlazado. Al momento de asignar un nombre al programa y seleccionar Crear en el Editor de ABAP/4, aparece entonces la pantalla de los atributos del programa. Para ingresar estos atributos se debe proceder de la siguiente manera:

- Ingresar un titulo para el programa en el campo Título. Se debe de escoger un nombre que describa la función del programa, ver figura 2.9
- Completar todos los atributos

Título	Programa de Prueba Tesis		
Idioma maestro	ES	Español	
Creado por	27.10.2011	MMORENO	
Última modific.			
Status	Nuevo(revisado)		
Atributos			
Tipo	Programa ejecutable		
Status	Programa estándar SAP productivo		
Aplicación	Multiaplicaciones		
Grupo autorizaciones			
Base de datos lógica			
Vers.imagen selección			
<input type="checkbox"/> Bloqueo de editor	<input checked="" type="checkbox"/> Cálculo de coma fija		
<input checked="" type="checkbox"/> Verif.unicode activas	<input type="checkbox"/> Iniciar mediante variante		

Figura 2.9 Atributos de programa ABAP

c. Escribir el código del programa.

El programador escribe el programa en el Editor de ABAP/4. Se puede enlazar el Editor directamente desde la pantalla de atributos del programa. Los pasos son los siguientes:

- I. Seleccionar **Ir a Código Fuente** o escoger Código Fuente en la pantalla de atributos del programa, o se debe seleccionar Código Fuente y escoger Modificar en la pantalla inicial del ABAP/4. Aparece entonces, la pantalla del Editor de ABAP/4. El sistema automáticamente ingresa la primera instrucción al programa, por ejemplo, **REPORT**<nombre del reporte> a **PROGRAM**<nombre del programa>. Para el nombre del programa, el sistema utiliza el nombre que el programador ingresó en la pantalla inicial del Editor de ABAP/4.

Las declaraciones **PROGRAM** y **REPORT** tienen exactamente la misma función. Estas permiten al sistema reconocer el programa de reportes o cualquier otro tipo de programa de ABAP/4 y permite al programador especificar ciertos criterios para la lista de salida.

Las declaraciones **REPORT** o **PROGRAM** pueden tener diferentes parámetros, por ejemplo LINE-SIZE, LINE-COUNT o NO STANDARD PAGE HEADING. Estos parámetros son utilizados principalmente en programas de reportes que sirven para analizar datos, y proporcionan como salida los resultados en una lista.

- II. Ingresar el código del programa.

III. Chequear y revisar la sintaxis, seleccionando la opción **Check**. El sistema entonces escanea el código del programa para encontrar errores de sintaxis e incompatibilidades. Si se detecta un error en este procedimiento, aparece un mensaje para reportarlo, y si es posible propone una solución. El cursor se coloca automáticamente en la posición apropiada.

IV. Salvar el código seleccionando **Save**.

d. Probar el programa.

El programa se debe de chequear para determinar si el código funciona correctamente. Para ejecutar el programa únicamente para revisión, se debe seleccionar **Ejecutar** en la pantalla de edición del Editor de ABAP/4. El sistema ejecuta el programa exactamente como si se hubiera empezado desde la pantalla inicial del Editor de ABAP/4.

2.5 TRANSACCIONES

Una transacción es un programa que conduce un diálogo con el usuario. En un diálogo típico, el sistema despliega una pantalla en la cual el usuario puede ingresar o requerir información. Como consecuencia de lo que el usuario ingrese, el programa ejecuta las instrucciones apropiadas, es decir, pasa a la siguiente pantalla, despliega una salida o realiza cambios en la base de datos.

Ejemplo:

Un agente viajero desea reservar un vuelo. El agente ingresa los datos correspondientes en la pantalla. Posteriormente el agente puede tomar dos acciones: primero, confirmar la solicitud requerida al sistema, es decir, reservar el vuelo y viajar el día deseado en el asiento reservado al destino elegido, o segundo, el sistema despliega la información de que el vuelo ya ha sido reservado.

Para cumplir con tales requerimientos, un programa de diálogo debe de ofrecer:

- Una interfaz amigable.
- Revisión de formato y consistencia para los datos ingresados por el usuario.
- Fácil corrección de errores de ingreso de datos.
- Fácil acceso a los datos, almacenándolos en una base de datos.

2.5.1 Estructura de un programa de diálogo

Un programa de diálogo consiste de los siguientes componentes básicos:

Pantallas (dynpros)

Cada diálogo en un sistema de SAP es controlado por un dynpro (Dynamic Program). Un dynpro consiste de una pantalla y su lógica de flujo, además controla exactamente un paso en el diálogo. La lógica del flujo determina cual proceso toma lugar antes de desplegar una pantalla, es decir, se ejecuta un

PBO (Process Before Output) que es un proceso que se ejecuta antes de dar los resultados o salida, y luego de recibir los datos que el usuario haya ingresado en la pantalla, es decir, se ejecuta entonces un PAI (Process After Input), que es un proceso que se ejecuta después de haber recibido los datos que el usuario ha ingresado.

Banco de módulos de ABAP/4

Cada pantalla (dynpro) se refiere como se mencionó anteriormente, exactamente a un programa de diálogo de ABAP/4. Este programa es llamado también un "Banco de Módulos", ya que contiene módulos interactivos. La lógica del flujo de un dynpro contiene llamadas de módulos desde el banco de módulos correspondiente. Los módulos interactivos que se invocan en el evento PBO se utilizan para preparar la plantilla de la pantalla de acuerdo al texto, por ejemplo, colocando los contenidos de los campos o suprimiendo campos que no son necesarios para el despliegue. Los módulos interactivos que se invocan en el evento PAI se utilizan para revisar los datos ingresados por el usuario, y para enlazar y ejecutar los pasos del diálogo que sean apropiados, tales como actualizar.

Todos los dynpros que se invocan dentro y desde una transacción se refieren a un banco de módulos en común. Estos dynpros están numerados. El sistema almacena, para cada dynpro, el dynpro que será desplegado inmediatamente después. La secuencia o cadena de los dynpros o puede ser lineal así como cíclica. Desde una cadena de dynpros, es posible invocar otra cadena de dynpros, y luego de procesarla, regresar a la cadena original.

No es posible escribir campos de datos utilizando la instrucción WRITE dentro del programa. El sistema, a cambio, transfiere los datos comparando los nombres de los campos en la pantalla con los nombres de las variables en el programa. Si ambos nombres son iguales, se transfieren los valores de los campos en la pantalla a los campos del programa, y viceversa. Esto sucede inmediatamente antes e inmediatamente después de desplegar la pantalla.

2.5.2 Atributos de los campos

Para todos los campos de una pantalla, los atributos se definen en el Screen Painter o Dibujador de Pantallas. Si el nombre de un campo en la pantalla corresponde al nombre de un campo dentro del Diccionario de ABAP/4, el sistema establece automáticamente una referencia entre estos dos campos. Estos atributos, junto con los elementos de datos y el dominio del diccionario asignado forman la base para las funciones estándar que un dynpro ejecuta en un dialogo (por ejemplo, chequeo automático del formato para los campos de la pantalla, chequeo automático del rango de valores, ayuda en línea, etc.)

2.5.3 Diálogos de error

Otra función del procesador de dynpros es conducir los diálogos de error. El chequeo automático de los datos de entrada se realiza utilizando tablas de chequeo del Diccionario de ABAP/4, o por el programa en sí. Además, solo los campos cuyo contenido está relacionado con el error, y para los cuales existe una corrección, pueden aceptar modificaciones.

2.6 DYNPRO

Cada pantalla contiene campos utilizados para desplegar o requerir información. Estos campos pueden ser cadenas de caracteres, campos de entrada o salida de datos, botones redondos, botones simples, etc.

Un dynpro de SAP consiste de varios componentes:

- Lógica de flujo: utilizada para invocar los módulos de ABAP/4 para una pantalla.
- Formato de la pantalla: posiciones de los textos, campos, botones, y todo lo necesario para una pantalla.
- Atributos de la pantalla: número de la pantalla, número de la pantalla subsecuente, y otros. Atributos del campo: definición de los atributos de campos individuales en una pantalla.

2.6.1 Atributos de la pantalla

Desde el punto de vista del usuario, una transacción es una secuencia de pantallas, desplegadas una después de otra. ¿Cómo se determina esa secuencia? Los atributos de las transacciones determinan qué pantalla debe desplegarse primero. Los atributos de los dynpros individuales determinan cuál pantalla desplegar después de la actual. También es posible colocar el número de las pantallas subsecuentes en forma dinámica desde el programa.

2.6.2 Formato

Si se escoge Fullscreen, se traslada el control al editor de la pantalla. Aquí es posible determinar la forma de la pantalla.

2.6.3 Lógica de flujo

El código para el control de flujo de un dynpro consiste de unas pocas declaraciones que sintácticamente se parecen a las de ABAP/4. Sin embargo, no es posible utilizar palabras clave de control en ABAP/4 y viceversa. El código para el control de flujo se ingresa en el Screen Painter como un componente del dynpro.

2.7 ASIGNACIÓN DEL CÓDIGO DE TRANSACCIONES A LOS PROGRAMAS

Es posible asignar los códigos de transacciones a programas de tipo 1, y manejarlos como tales en adelante. Para esto, se debe proceder de la siguiente manera:

- Crear un programa.
- Seleccionar Development Further Tools Transacciones en la pantalla de Herramientas de Desarrollo de ABAP/4.
- Completar el nombre de la transacción y seleccionar Create en la pantalla de mantenimiento de transacciones.
- Seleccionar Report Transacción en la ventana de dialogo que aparece.
- Llenar los campos de texto requeridos en la pantalla de transacciones Create Report.

2.8 MANEJO DE DATOS

2.8.1 Declaración de datos

Para declarar datos, el usuario debe estar familiarizado con el concepto de ABAP/4 que define la relación entre los tipos de datos y los objetos de datos.

2.8.1.1 Introducción a los tipos de datos y objetos

Los tipos de datos y los objetos de datos son componentes esenciales en el concepto de tipos de ABAP/4. Ambos pueden ser declarados y mantenidos por el usuario. Esto contrasta con otros lenguajes de programación como C, Fortran, o Pascal, donde el usuario puede declarar los objetos de datos pero está restringido a los tipos de datos predefinidos. En ABAP/4, el usuario puede procesar tipos de datos de manera independiente de los objetos de datos, utilizando una sintaxis paralela a la declaración normal de datos. También es posible almacenar tipos de datos centralizadamente, fuera de un programa específico.

Las principales funciones de los tipos de datos y objetos son las siguientes, entre otras:

Tipos de datos (elementales o estructurados)

- Los tipos de datos son puramente descripciones.
- No hay memoria asociada con los tipos de datos.
- El tipo de dato es un atributo de un objeto de datos, y está íntimamente conectado con este.

- Los objetos de datos son las unidades físicas que un programa utiliza en tiempo.
- Cada objeto de datos tiene un tipo de datos específico asignado.

2.8.1.1.1 Tipos de datos.

Los tipos de datos en ABAP/4 pueden definirse de acuerdo a su estructura y definición.

Los tipos de datos pueden ser:

- Elementales (no estructurados) o estructurados.
- Predefinidos o definidos por el usuario.

Los tipos de datos de ABAP/4 se describen en la siguiente tabla:

Tabla 2.1 Tipos de Datos de ABAP/4

	PREDEFINIDOS	DEFINIDOS POR EL USUARIO
Elementales	C, D, F, I, N, P, T, X: ABAP/4 contiene ocho tipos de datos elementales predefinidos.	Los tipos de datos elementales definidos por el usuario están basados en los tipos de datos predefinidos
Estructurados	TABLA: Este tipo de dato estructurado y predefinido s utiliza solo para teclear parámetros formales y símbolos.	Cadenas de campos y tablas internas: Estos tipos de datos estructurados pueden utilizarse para objetos de datos y son definidos por el usuario.

2.8.1.1.2 Objetos de datos.

En ABAP/4 es posible trabajar con varias clases de objetos de datos, tales como:

- Objetos de datos internos

Los objetos de datos internos se crean para uso en un programa particular. No tienen validez fuera de ese programa. Los objetos de datos internos incluyen:

- **Literales:** Una literal es un valor compuesto. ABAP/4 distingue entre literales de texto y literales numéricas.
- **Literales de texto:** Las literales de texto son secuencias de caracteres alfanuméricos incluidos en comillas simples o apóstrofes. Por ejemplo:

'Antonio Pérez'

69190 Walldorf'

Las literales de texto pueden tener una longitud de hasta 254 caracteres. Si una literal de texto contiene un apóstrofe, se debe repetir (colocar doble apóstrofe) para que el sistema 10 pueda reconocer como parte del contenido de la literal y no como la terminación de ésta.

Ejemplo:

WRITE: / 'Esta es la bicicleta de Juan D'Arcy'

Esta instrucción genera la siguiente salida:

Esta es la bicicleta de Juan D'Arcy

- **Variables:** Una variable se utiliza para almacenar y hacer referencia a datos bajo cierto nombre y cierto formato. Las variables pueden diferir en:

- **Nombre**
- **Tipo**
- **Longitud**
- **Estructura**

Las variables se declaran en el programa con la instrucción **DATA**.

Ejemplo:

```
DATA: S1 TYPE I,  
      S2 TYPE I,  
      SUM TYPE I.  
  
SUM = S1+S2
```

- **Constantes:** Una constante es un objeto de datos que contiene un valor que el usuario puede componer en el momento de su inicialización. Las constantes no pueden ser modificadas durante la ejecución del programa. Las constantes se declaran en el programa con la instrucción **CONSTANTS**. Si se trata de modificar o cambiar la constante en algún otro lugar del programa, ya sea el chequeo de sintaxis o el sistema en tiempo de corrida desplegará un mensaje de error.

Ejemplo:

```
CONSTANTS P1 TYPE P DECIMALS 10 VALUE  
'3.1415926536'.
```

- **Tipos de datos externos:** Los tipos de datos externos existen independientemente de los programas. No se puede trabajar directamente con ellos, pero se pueden copiar a objetos de datos. Los tipos de datos externos pueden utilizarse globalmente a lo largo del ambiente del sistema.

ABAP/4 almacena los objetos de datos en tablas definidas en el Diccionario de ABAP/4. Para acceder estos datos desde un programa, se declaran las tablas en el programa con la instrucción **TABLES**.

- **Objetos de datos definidos por el sistema:** Además de los objetos de datos definidos por el usuario, algunos objetos se definen automáticamente en el sistema. Dentro de estos están:
 - Objetos especiales de datos.
 - Parámetros.

2.9 CONTROL DE FLUJO

Para ejecutar los diferentes componentes de un programa dependiendo de ciertas condiciones, o para combinar secuencias de instrucciones repetitivas juntas en ciclo o "loops", se puede usar las palabras clave estándar que ABAP/4 provee para controlar el flujo de un programa.

ABAP/4 también contiene una función adicional importante: el flujo de un programa de ABAP/4 puede ser controlado interna y externamente. La diferencia entre el control interno y el externo se explican a continuación.

Concepto del control de flujo en ABAP/4

Al igual que otros lenguajes de programación (como FORTRAN, PASCAL, etc.), ABAP/4 provee palabras clave estándar para controlar el flujo de un programa.

Estas palabras clave se pueden utilizar para:

- ramificar (IF, CASE)
- hacer ciclos (DO, WHILE)

Sin embargo, ABAP/4 difiere de estos otros lenguajes de programación en que este tiene un control interno y externo para el flujo del programa.

El control interno se maneja con palabras clave estándar según se mencionó anteriormente.

El control externo se maneja por eventos. Los eventos se generan, ya sea desde otros programas de ABAP/4 (programas del sistema o programas del usuario), o desde entradas interactivas del usuario (como por ejemplo el utilizar el mouse para hacer un clic en la pantalla). El sistema no necesariamente procesa las instrucciones en la misma secuencia como son listadas en el programa de ABAP/4. Esto hace de ABAP/4, un lenguaje de programación conducido por eventos, similarmente a Microsoft Visual Basic.

2.9.1 Programación de expresiones lógicas

Es posible controlar el flujo interno de un programa a través de condiciones. Para formularlas, se utilizan expresiones lógicas que comparan los campos de datos, como sigue:

Sintaxis

... <f1 ><operador><f2>...

Esta expresión compara dos campos. Puede ser verdadera o falsa. Se utilizan las expresiones lógicas en instrucciones de condición con las palabras **IF**, **CHECK**, Y **WHILE**.

Dependiendo de los tipos de dato de los operándos<f1> Y <f2>, el usuario puede utilizar diferentes operadores lógicos. Estos le permiten hacer:

- Comparaciones con todos los tipos de campos.
- Comparaciones con cadenas de caracteres y cadenas de números.

2.9.2 Programación de ramificaciones y bucles

2.9.2.1 CASE

Para ejecutar diferentes bloques de instrucciones, dependiendo del contenido de campos particulares de datos se utiliza la instrucción CASE como sigue:

Sintaxis:

CASE <f>.

WHEN

<f1>.

<Bloque de instrucciones>

WHEN <f2>.

<Bloque de instrucciones>

WHEN <f3>.

<Bloque de instrucciones>

WHEN...

WHEN OTHERS.

<Bloque de instrucciones>

ENDCASE.

2.9.2.2 DO

Si se quiere procesar un bloque de instrucciones más de una vez, se puede programar un ciclo utilizando la instrucción **DO**, como sigue:

Sintaxis:

```
DO [<n> TIMES] [VARYING <f> FROM <f1> NEXT <f2>].
```

<Bloque de instrucciones>

```
ENDDO.
```

2.9.2.3 WHILE

Si se desea procesar un bloque de instrucciones más de una vez, siempre y cuando la condición sea verdadera, es posible programar un ciclo con la instrucción **WHILE**, como sigue:

Sintaxis:

```
WHILE <condición> [VARY <f> FROM <f1> NEXT <f2>]
```

<Bloque de instrucciones>

```
ENDWHILE.
```

2.10 TÉCNICAS ESPECIALES DE PROGRAMACIÓN

2.10.1 Creación y procesamiento de tablas internas

Además de los campos de tipo string, las tablas internas constituyen otro de los tipos de datos estructurados que ABAP/4 provee.

2.10.2 Propósito de las tablas internas

En ABAP/4, se trabaja mayormente con tablas. Las tablas son las estructuras de datos esenciales en el sistema R/3. La información de larga vida se almacena en tablas de bases de datos relacionales.

Además de las tablas de bases de datos, es posible crear tablas internas, las cuales existen sólo durante el tiempo de corrida del programa. ABAP/4 provee varias operaciones para trabajar con tablas internas, como por ejemplo, buscar líneas específicas, insertar o borrar líneas.

Dependiendo de los requerimientos, el sistema incrementa el tamaño de las tablas internas durante el tiempo de corrida. Si, por ejemplo, el usuario desea leer una tabla en una tabla interna, no es necesario que sepa el tamaño de la tabla de la base de datos por anticipado. Esta función hace que el trabajar con tablas internas sea una tarea fácil, al mismo tiempo que soporta programación dinámica.

Otro uso de las tablas internas es reorganizar el contenido de las tablas de bases de datos de acuerdo a las necesidades del programa. Por ejemplo, es posible leer datos relevantes para la creación de una lista de teléfonos de una o varias tablas grandes de clientes, dentro de una tabla interna.

Durante el tiempo de corrida del programa, el usuario puede acceder esta lista directamente sin tener que hacer una búsqueda en la base de datos, consumiendo tiempo.

Además de usarlas al trabajar con tablas de bases de datos, las tablas internas son una importante función en ABAP/4 para la implementación de estructuras complejas de datos en el programa.

2.10.3 Estructura de las tablas internas

En ABAP/4, es posible distinguir entre una tabla interna con tipo de datos, la cual define la estructura de una tabla interna, y los objetos de datos de tablas internas, que son en sí las tablas internas y pueden llenarse con datos. Una tabla interna con tipo de datos es una definición abstracta de una estructura que puede ser utilizada para declarar objetos de datos como tablas internas. Ver Tipos de datos y Objetos de Datos, en Declaración de Datos.

2.10.4 Trabajando con tablas internas

Para crear un tipo de datos como una tabla interna, se debe utilizar la instrucción **TYPES**, como sigue:

Sintaxis

```
TYPES <t><tipo> OCCURS <n>.
```

Esto lo que hace es crear una tabla interna con un tipo de datos <t>, utilizando la opción OCCURS de la instrucción TYPES. Las líneas de una tabla interna tienen el tipo de datos especificado en <tipo>. Para especificar el tipo de datos de las Líneas, se puede utilizar, ya sea el parámetro TYPE o el parámetro LIKE.

Al utilizar el parámetro LIKE para hacer referencia a un objeto definido en el Diccionario de ABAP/4, se pueden crear tablas internas, las cuales tendrán la misma estructura de las líneas que los objetos almacenados en el Diccionario, y reflejarán la estructura de las tablas de bases de datos.

<n> especifica un número inicial de líneas. La memoria está, entonces, reservada para el número de líneas especificado, tan pronto como se escribe la primera línea en un objeto de datos como una tabla interna creado con tipo <t>. Si se agregan mas líneas de las especificadas en <n> a la tabla interna, la memoria reservada se expande automáticamente. Si no hay espacio suficiente en la memoria para una tabla interna, entonces se escribe a un buffer o al disco (área de paginado).

TYPES VECTOR TYPE I OCCURS 10.

Este ejemplo crea una tabla interna con tipo de datos VECTOR, el cual tiene líneas consistentes del campo de tipo elemental I.

TYPES: BEGIN OF LINE,

COLUMN1 TYPE I,

COLUMN2 TYPE I,

COLUMN3 TYPE I,

END OF LINE.

TYPES ITAB TYPE LINE OCCURS 10.

Este ejemplo crea una tabla interna con tipo de datos ITAB, la cual contiene líneas con la misma estructura del campo LINE.

Para inicializar una tabla interna con o sin una línea de encabezado, se utiliza la instrucción REFRESH, como sigue:

Sintaxis:

REFRESH <itab>

Esta instrucción regresa la tabla interna al estado en que estaba antes de que fuera llenada.

Esto significa que la tabla no contiene ninguna línea. Si se está trabajando con una tabla interna sin un área de trabajo, se puede utilizar la instrucción **CLEAR** en vez de la instrucción **REFRESH**, de la siguiente forma:

Sintaxis:

CLEAR <itab>

Si se está trabajando con una tabla interna con una línea de encabezado, la instrucción **CLEAR** limpia únicamente el área de trabajo de la tabla. Para limpiar toda la tabla interna sin limpiar el área de trabajo de la tabla, se utiliza **REFRESH** o **CLEAR**.

2.11 MODULARIDAD DE LOS PROGRAMAS ABAP/4

Si un programa ABAP/4 contiene los mismos, o similares bloques de instrucciones, o si se desea procesar la misma función varias veces, se puede evitar redundancia utilizando algunas técnicas de modularización. Modularizando los programas de ABAP/4, se puede lograr que éstos sean fáciles de leer. Así mismo, los programas modularizados son más fáciles de mantener y actualizar. ABAP/4 provee las siguientes técnicas de modularización:

2.11.1 Módulos de código fuente

Si se modulariza el código fuente, los programas serán más fáciles de mantener. El usuario debe usar estas técnicas de modularización, si, por ejemplo, desea evitar tener que repetir las mismas instrucciones varias veces en el programa. Sin embargo, no se debe utilizar este procedimiento para modularizar cualquier tarea o función. Para esto, se debe utilizar subrutinas y módulos de función.

Es posible crear módulos invocables de código dentro del programa de ABAP/4 por medio de la definición de Macros.

Definición e invocación de Macros

Para definir una macro que contenga parte del código fuente, se utiliza la instrucción **DEFINE**, Como sigue:

Sintaxis:

DEFINE <macro>.

<Instrucciones>

END-OF-DEFINITION.

2.11.2 Definición de subrutinas

Una subrutina es un bloque de código que inicia con **FORM** y concluye con **ENDFORM**. Para definir una subrutina, se utiliza la siguiente sintaxis:

Sintaxis:

FORM <subr> [<pasar>]

<Bloque de instrucciones>

ENDFORM.

El nombre de la subrutina está definido por <subr>. En la opción <pasar>, se especifica cómo pasar datos de y a la subrutina.

Una subrutina no puede contener bloques anidados de **FORM-
ENDFORM**.

2.11.3 Invocando subrutinas internas

Para llamar una subrutina interna, se debe utilizar la instrucción **PERFORM**, como se muestra a continuación:

Sintaxis:

PERFORM <subr> [<pasar>].

Con esta instrucción se invoca la subrutina <subr>. En la opción <pasar> se especifica cómo pasar datos desde y hacia la subrutina. Asimismo, si no se utiliza, esta opción, la subrutina podrá acceder todos los tipos de datos y objetos declarados en el programa principal de ABAP/4. Estos datos llamados son datos globales, y son visibles para una subrutina si no están escondidos por definiciones de datos locales con los mismos nombres.

Por ejemplo:

PROGRAM SAPMZTST.

DATA: NUM1 TYPE I,

```

        NUM2 TYPE I,
        SUM TYPE I.

NUM1 = 2. NUM2 = 4.

PERFORM ADDIT.

NUM1 = 7. NUM2 = 11.

PERFORM ADDIT.

FORM ADDIT.

        SUM = NUM1 + NUM2.

        PERFORM OUT.

ENDFORM.

FORM OUT.

        WRITE: / 'La suma de', NUM1, 'y', NUM2, 'es', SUM.

ENDFORM.

```

La salida es como sigue:

La suma de La suma de 2 y 4 es6

La suma de La suma de 7 y 11es18

ADDIT se invoca desde el programa, y OUT se invoca desde ADDIT. Las subrutinas tienen acceso en forma automática a los campos NUM1 y NUM2.

2.11.4 Módulos de función

Para programar un módulo de función, se deben incluir las instrucciones entre **FUNCTION** y **ENDFUNCTION**, como sigue:

Sintaxis:

```
FUNCTION <modulo>.
```

<instrucciones>

ENDFUNCTION.

Para ingresar el código del programa se debe proceder de la misma manera que para las subrutinas normales, pero con las siguientes excepciones:

- **Manejo de datos en módulos de función:**

No se debe declarar los parámetros para importar y exportar en el código fuente del módulo de función. El sistema realiza la tarea automáticamente en un programa **INCLUDE**.

- **Invocar subrutinas desde los módulos de función**

Se pueden invocar diferentes subrutinas desde los módulos de función.

- Se puede escribir el código del programa para las subrutinas internas directamente antes de la instrucción **ENDFUNCTION** de un módulo. Estas subrutinas pueden invocarse desde todos los módulos del grupo de funciones.
- Para crear subrutinas internas que pueden llamarse desde todos los módulos de un grupo de funciones <fgrp>, se debe utilizar programas **INCLUDE** especiales. Se pueden abrir estos programas **INCLUDE** haciendo doble clic sobre sus nombres en el programa principal **SAPL**<fgrp> luego de seleccionar **Main program** en la librería de funciones.

CAPÍTULO 3

3. METODOLOGÍA XP (Programación Extrema)

3.1 PLANIFICACIÓN

3.1.1 Especificación de requerimientos

Se requiere realizar una interfaz para los controles de materias primas (químicos y colorantes) de la siguiente manera:

Los componentes químicos de la orden están asociados a la cabecera del grupo Dyeing y esta al ser liberada genera una única reserva para los diferentes químicos. Colorantes. Químicos sólidos y químicos líquidos.

Los tres tipos de químico se entregan en diferentes áreas en donde cada usuario entrega determinados materiales.

Para realizar la interfaz es necesario filtrar por cada tipo de químico que se debe entregar, para ello se cuenta con los siguientes grupos de materiales.

1002 Colorantes

1003 Químicos líquidos

1005 Químicos Sólidos (está por definir el tipo con MM).

Para la entrega de estos tres tipos de químicos son necesarias tres interfaces independientes.

1. Interfaz Colorantes

La Interfaz será de ida y vuelta.

La Interfaz enviará desde SAP Envió la siguiente información:

- Número de Orden
- Número de reserva
- Posiciones de la reserva.
- Componentes asociados a la reserva filtrados por el grupo de materiales 1002. (al software de la báscula le llegara la reserva SOLO con los materiales Colorantes.)
- Cantidad de cada reserva.

2. Interfaz Químicos líquidos

La interfaz será de ida y vuelta.

La Interfaz enviara desde SAP Envió la siguiente información:

- Número de Orden
- Número de reserva
- Posiciones de la reserva.

- Componentes asociados a la reserva filtrados por el grupo de materiales 1003 Químicos líquidos (al software legacy le llegarán la reserva SOLO con los materiales químicos líquidos.)
- Cantidad de cada reserva.

3. Interfaz Otros Químicos Sólidos

Se leerá por barcode el número de la orden y traerá las posiciones filtradas por el grupo 1005. (Solo podrá ver la reserva SOLO con los materiales otros químicos sólidos)

Se requiere que la cantidad real pesada en la báscula se almacenara en el campo ERFMG tabla MSEG de la reserva de material. Y se salvara la transacción con impresión automáticamente de la etiqueta.

La interfaz envía de la báscula la siguiente información:

Cantidad real pesada. Para cada una de las posiciones de la reserva.

Se requiere que la cantidad real pesada en la báscula se almacenara en el campo ERFMG tabla MSEG de la reserva de material. Y se salvara la transacción con impresión automáticamente de la etiqueta.

Las interfaces reciben de la Legacy la siguiente información:

Cantidad real entregada automáticamente en cada proceso. Para cada una de las posiciones de las reservas (filtradas).

Se requiere que la cantidad real pesada en la báscula se almacenara en el campo ERFMG tabla MSEG de la reserva de material. Y se salvara la transacción con impresión automáticamente de la etiqueta.

Para cada entrega en la reserva se genera en un evento independiente, en donde:

La entrega de las cantidades de la reserva de la orden ZP01 DYE, no se ingresarán a la orden ZP01 DYE sino que serán prorrateadas al mismo componente de las reservas de cada orden.

Buscará en la tabla ZPPGRUPOS el campo DYE y buscara campo OFAB las órdenes asociadas y buscara campo Tabla RESB el campo AUFNR Orden de acuerdo a al campo BDMNG Cantidad necesaria prorratea el campo ERFMG tabla MSEG de la reserva de material.

3.1.2 Descripción general

El Diseño de la interfaz se basa principalmente en un requerimiento para cubrir una necesidad muy puntual dentro del proceso productivo de la empresa S.J. Jersey la cual es tener el control del las dosificaciones de químicos y colorantes, este proceso al momento no cuenta con ninguna automatización y el desperdicio o perdida del producto llevaron a que se solicite esta solución.

Para el diseño de la interfaz, se consideró básicamente la experiencia del usuario a fin de que la aplicación tenga el éxito deseado.

3.1.3 Historias de usuario.

Las Historias de usuario dentro de la metodología XP otorgan al usuario facilidad para que éste pueda escribir sus requerimientos, y una vez que lo escribió el programador transforma en código, el mismo que procesa la petición y entrega la respuesta.

Cada usuario puede escribir más de una historia, y no tiene importancia escribir en un orden, ya que el plan de entregas ordena sus requerimientos.

Las historias que se detallan a continuación desde la tabla 3.1 hasta la 3.8 fueron elaboradas y entregadas por el personal que labora en la S.J. Jersey en el área de Bodegas y Producción.

Tabla 3.1 Historia de usuario 1

		HISTORIA DE USUARIO N. 01			
FECHA:	07/05/2012	TIPO ACTIVO:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA	PRIORIDAD TÉCNICA:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
REFERENCIA:		RIESGO:	<input checked="" type="checkbox"/> NINGUNO <input type="checkbox"/> MEDIO <input type="checkbox"/> ALTO	PRIORIDAD USUARIO:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
USUARIO:	FERNANDO LOZANO	CARGO:	BODEGUERO	CALSIFICACIÓN:	CONSULTA DE MATERIALES
DESCRIPCIÓN:					
PARA FACILITAR EL PROCESO DE TINTURA DE TELA ES NECESARIO CONTROLAR EL PROCESO DE DOSIFICACIÓN AUTOMÁTICA DE QUÍMICOS Y COLORANTES, CONSULTAR MATERIALES POR ORDEN DYE					
SE NECESITA CONSULTAR TODOS LOS MATERIALES DE UNA ORDEN DYE A SER DOSIFICADOS, EL FILTRO DEBE SER POR: GRUPO DE ARTICULOS (1002 COLORANTES), (1003 QUÍMICOS LÍQUIDOS), (1005 QUÍMICOS SÓLIDOS) POR NÚMERO DE ORDEN ESPECÍFICA POR TIPO (SIN ENVIAR), (ENVIADOS), (TODOS)					
NOTAS:					
SEGUIMIENTO:					
ESTADO:	<input checked="" type="checkbox"/> ATENDIDO <input type="checkbox"/> EN PROCESO <input type="checkbox"/> FINALIZADO	COMENTARIOS:		ÚLTIMO SEGUIMIENTO:	

Tabla 3.2 Historia de usuario 2

		HISTORIA DE USUARIO N. 02			
FECHA:	09/05/2012	TIPO ACTIVO:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA	PRIORIDAD TÉCNICA:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
REFERENCIA:		RIESGO:	<input checked="" type="checkbox"/> NINGUNO <input type="checkbox"/> MEDIO <input type="checkbox"/> ALTO	PRIORIDAD USUARIO:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
USUARIO:	FERNANDO LOZANO	CARGO:	BODEGUERO	CALSIFICACIÓN:	ENVÍO DE MATERIALES
DESCRIPCIÓN:					
DE LA CONSULTA DE MATERIALES SE DEBE MOSTRAR EN UNA TABLA LA SIGUIENTE INFORMACIÓN: ESTADO, NÚMERO DE DYE, RESERVA, MATERIAL, DESCRIPCIÓN DEL MATERIAL, GRUPO, CANTIDAD, UNIDAD, ALMACEN, CANTIDAD DOSIFICADA, LOTE SE DEBE PODER SELECCIONAR UNA O VARIAS LINEAS Y ENVIAR LA TRANSMISIÓN A LAS MÁQUINAS DE DOSIFICACIÓN.					
NOTAS:					
SEGUIMIENTO:					
ESTADO:	<input checked="" type="checkbox"/> ATENDIDO <input type="checkbox"/> EN PROCESO <input type="checkbox"/> FINALIZADO	COMENTARIOS:		ÚLTIMO SEGUIMIENTO:	

Tabla 3.3 Historia de usuario 3

		HISTORIA DE USUARIO N. 03			
FECHA:	11/05/2012	TIPO ACTIVO:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA	PRIORIDAD TÉCNICA:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
REFERENCIA:		RIESGO:	<input checked="" type="checkbox"/> NINGUNO <input type="checkbox"/> MEDIO <input type="checkbox"/> ALTO	PRIORIDAD USUARIO:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
USUARIO:	FERNANDO LOZANO	CARGO:	BODEGUERO	CALSIFICACIÓN:	GENERACIÓN DE ARCHIVO COLORANTES
DESCRIPCIÓN:					
PARA LA DOSIFICACIÓN DE COLORANTES SE DEBE GENERAR UN ARCHIVO CON FORMATO *.TMP LLAMADO MAGAZZ.TMP EL CUAL VA A SER COLOCADO EN UNA UNIDAD DE RED CON DIRECCIÓN "Q:\WINMAGPASAY" CON LAS ESPECIFICACIONES DEL EQUIPO DE COLORANTES, SE ENTREGARÁ UN ARCHIVO CON EL FORMATO Y TIPO DE CAMPOS NECESARIOS					
NOTAS:					
SEGUIMIENTO:					
ESTADO:	<input checked="" type="checkbox"/> ATENDIDO <input type="checkbox"/> EN PROCESO <input type="checkbox"/> FINALIZADO	COMENTARIOS:		ÚLTIMO SEGUIMIENTO:	

Tabla 3.4 Historia de usuario 4

		HISTORIA DE USUARIO N. 04			
FECHA:	14/05/2012	TIPO ACTIVIDAD:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA	PRIORIDAD TÉCNICA:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
REFERENCIA:		RIESGO:	<input checked="" type="checkbox"/> NINGUNO <input type="checkbox"/> MEDIO <input type="checkbox"/> ALTO	PRIORIDAD USUARIO:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
USUARIO:	FERNANDO LOZANO	CARGO:	BODEGUERO	CALSIFICACIÓN:	GENERACION DE ARCHIVO QUIMICOS LIQUIDOS
DESCRIPCIÓN:					
PARA LA DOSIFICACION DE QUIMICOS LIQUIDOS SE DEBE GENERAR UNA ARCHIVO CON FORMATO *.TMP LLAMADO RECIPETMP EL CUAL VA A SER COLOCADO EN UNA UNIDAD DE RED CON DIRECCIÓN "z:\dps95win\DATA\\" CON LAS ESPECIFICACIONES DEL EQUIPO DE QUIMICOS LIQUIDOS, SE ENTREGARA UN ARCHIVO CON EL FORMATO Y TIPO DE CAMPOS NECESARIOS					
NOTAS:					
SEGUIMIENTO:					
ESTADO:	<input checked="" type="checkbox"/> ATENDIDO <input type="checkbox"/> EN PROCESO <input type="checkbox"/> FINALIZADO	COMENTARIOS:		ULTIMO SEGUIMIENTO:	

Tabla 3.5 Historia de usuario 5

		HISTORIA DE USUARIO N. 05			
FECHA:	16/05/2012	TIPO ACTIVIDAD:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA	PRIORIDAD TÉCNICA:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
REFERENCIA:		RIESGO:	<input checked="" type="checkbox"/> NINGUNO <input type="checkbox"/> MEDIO <input type="checkbox"/> ALTO	PRIORIDAD USUARIO:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
USUARIO:	FERNANDO LOZANO	CARGO:	BODEGUERO	CALSIFICACIÓN:	GENERACION DE ARCHIVO QUIMICOS SOLIDOS
DESCRIPCIÓN:					
PARA LA DOSIFICACION DE QUIMICOS SOLIDOS NO ES NECESARIO GENERAR ARCHIVO ALGUNO YA QUE LO HACE DIRECTAMENTE DESDE LA BALANZA SOLO SE SOLICITA CAMBIAR EL STATUS DE LA CONSULTA DE MATERIALES PARA SABER QUE FUE ENVIADO					
NOTAS:					
SEGUIMIENTO:					
ESTADO:	<input checked="" type="checkbox"/> ATENDIDO <input type="checkbox"/> EN PROCESO <input type="checkbox"/> FINALIZADO	COMENTARIOS:		ULTIMO SEGUIMIENTO:	

Tabla 3.6 Historia de usuario 6

		HISTORIA DE USUARIO N. 06			
FECHA:	18/05/2012	TIPO ACTIVIDAD:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA	PRIORIDAD TÉCNICA:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
REFERENCIA:		RIESGO:	<input checked="" type="checkbox"/> NINGUNO <input type="checkbox"/> MEDIO <input type="checkbox"/> ALTO	PRIORIDAD USUARIO:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
USUARIO:	FERNANDO LOZANO	CARGO:	BODEGUERO	CALSIFICACIÓN:	DESCARGO DE MATERIALES
DESCRIPCIÓN:					
SE SOLICITA REALIZAR EL DESGARGO DE MATERIALES DOSIFICADOS DEL INVENTARIO CON UN MOVIMIENTO DE INVENTARIO TIPO 261					
NOTAS:					
SEGUIMIENTO:					
ESTADO:	<input checked="" type="checkbox"/> ATENDIDO <input type="checkbox"/> EN PROCESO <input type="checkbox"/> FINALIZADO	COMENTARIOS:		ULTIMO SEGUIMIENTO:	

Tabla 3.7 Historia de usuario 7

		HISTORIA DE USUARIO N. 07			
FECHA:	21/05/2012	TIPO ACTIVIDAD:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA	PRIORIDAD TÉCNICA:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
REFERENCIA:		RIESGO:	<input checked="" type="checkbox"/> NINGUNO <input type="checkbox"/> MEDIO <input type="checkbox"/> ALTO	PRIORIDAD USUARIO:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
USUARIO:	FERNANDO LOZANO	CARGO:	BODEGUERO	CALIFICACIÓN:	IMPRESIÓN DE ETIQUETAS
DESCRIPCIÓN:					
PARA LA ENTREGA DE COLORANTES y QUIMICOS SOLIDOS ES NECESARIO REALIZAR LA IMPRESIÓN DE ETIQUETAS AL MOMENTO DE ENTREGAR LOS MATERIALES A PRODUCCION					
NOTAS:					
SEGUIMIENTO:					
ESTADO:	<input checked="" type="checkbox"/> ATENDIDO <input type="checkbox"/> EN PROCESO <input type="checkbox"/> FINALIZADO	COMENTARIOS:	ÚLTIMO SEGUIMIENTO:		

Tabla 3.8 Historia de usuario 8

		HISTORIA DE USUARIO N. 08			
FECHA:	25/10/2012	TIPO ACTIVIDAD:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA	PRIORIDAD TÉCNICA:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
REFERENCIA:		RIESGO:	<input checked="" type="checkbox"/> NINGUNO <input type="checkbox"/> MEDIO <input type="checkbox"/> ALTO	PRIORIDAD USUARIO:	<input checked="" type="checkbox"/> NUEVA <input type="checkbox"/> CORRECCIÓN <input type="checkbox"/> MEJORA
USUARIO:	FERNANDO LOZANO	CARGO:	BODEGUERO	CALIFICACIÓN:	INGRESOS MANUALES
DESCRIPCIÓN:					
EN CASO DE ERROR DE ALGUNA DE LAS INTERFACES SE SOLICITA PODER REALIZAR DESCARGOS MANUALES DE LOS MATERIALES					
NOTAS:					
SEGUIMIENTO:					
ESTADO:	<input checked="" type="checkbox"/> ATENDIDO <input type="checkbox"/> EN PROCESO <input type="checkbox"/> FINALIZADO	COMENTARIOS:	ÚLTIMO SEGUIMIENTO:		

3.1.4 Plan de Entregas

Cuando se han obtenido las historias de usuarios, y además determinar la necesidad en el orden de realización, se procedió a ubicar los tiempos de las entregas de estas, con el fin de estructurar el plan de iteraciones.

El tiempo que tomó definir todas las historias de usuario de la solución propuesta fue de 15 días laborables, y el tiempo que tomará desarrollar estas historias se definió en 28 días laborables, como se muestra en la figura 3.1

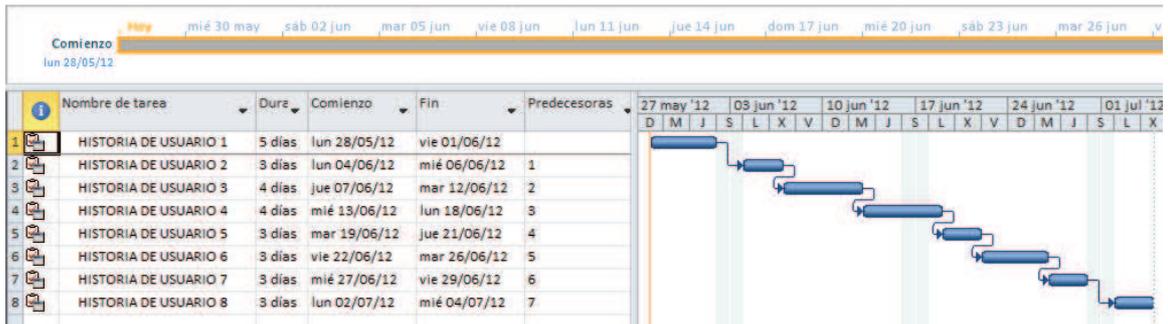


Figura 3.1 Plan de entregas de Historias de Usuario

3.1.5 Establecimiento de objetivos

En base al plan de entregas, surgió la necesidad de ir evaluando permanentemente como avanza el proceso a fin realizar los reajustes necesarios para cumplir con el tiempo establecido y satisfacer la necesidad del usuario.

3.1.6 Iteraciones

Descritas ya las historias de usuario, éstas serán agrupadas técnicamente en 3 iteraciones, bajo el criterio del tiempo que llevó en desarrollarse cada una de las historias.

Dentro del plan de entregas previsto se desarrollo la iteración numero 1 comprendida ésta por las historias 1, 2 utilizándose 8 días laborables; así mismo la iteración numero 2 compuesta por las historias 3, 4, 5,6, se desarrolló en los 14 días posteriores a la iteración numero 1 y la iteración numero 3 compuesta por las historias 7,8, se desarrolló en los días restantes, es decir, en los 6 días subsiguientes de haber concluido la iteración 2.

Las iteraciones sistematizadas fueron codificadas respetando la necesidad del usuario, aspecto fundamental para llegar al producto final, es decir la solución.

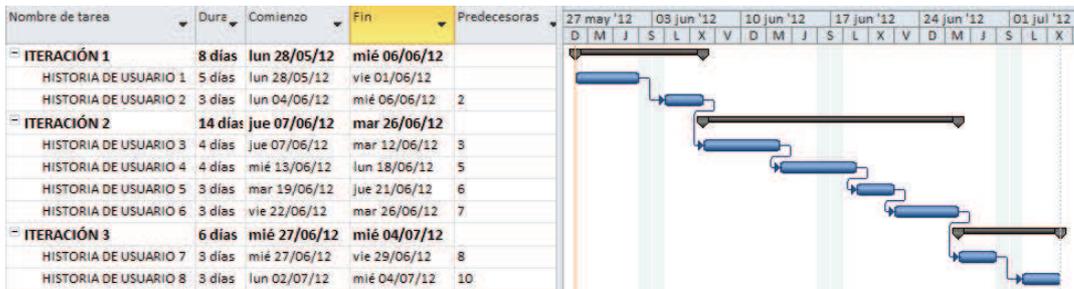


Figura 3.2 Plan de Iteraciones.

3.1.7 Rotaciones

La metodología XP como uno de sus pasos principales contempla la realización de rotaciones, ya que ella permite al personal que integra el equipo de desarrollo rotar para cumplir las diferentes actividades del desarrollo de software.

Por esta razón no es posible cumplir con este paso ya que el desarrollo será realizado por una sola persona.

3.1.8 Reuniones

Para alcanzar los objetivos y el alcance definidos se estableció realizar sesiones de trabajo continuas con el fin de evaluar el avance y el correcto desarrollo, identificando claramente las necesidades que nos lleven a construir una solución efectiva y de calidad.

Para este proyecto se definió las reuniones semanales cada lunes a las 8 AM.

3.2 DISEÑO

3.2.1 Seleccionar tipo de producto de conocimiento

Metáfora del Sistema

La solución creada permitirá llevar un control adecuado de las dosificaciones realizadas en los procesos de tintura dentro de la empresa S.J. Jersey. Esta información se extraerá de la base de datos central, la cual contiene la información correspondiente a las ordenes de producción, cantidades y materiales, estos datos llegan a una transacción para el envío a los diferentes sistemas, una vez realizado el proceso de dosificación se realiza la descarga de materiales del inventario y se imprime una etiqueta con los datos necesarios para finalmente se proceder a prorratear las cantidades dentro de las ordenes de producción hijas.

Tarjetas CRC

Uno de los pasos dentro de la metodología XP, contempla la necesidad de contar con tarjetas CRC en razón de que ellas reflejan la interacción de las clases que tiene el sistema.

Para este caso puntal se han levantado las siguientes tarjetas CRC:

Tabla 3.9: TARJETA CRC ENVÍO

ENVÍO	
Responsabilidad	Colaboración
1. Buscar las ordenes Enviadas, Sin enviar, o Todas, por grupo de artículos o numero de orden	Operario
2.- Realizar la transmisión a los equipos de dosificación	Operario
3. Verificación si fue o no enviada	Operario

La tarjeta CRC ENVÍO, tiene como función el buscar y enviar todas las órdenes a ser dosificadas.

Tabla 3.10: TARJETA MANUAL

MANUAL	
Responsabilidad	Colaboración
1. Buscar todas las ordenes Sin enviar, por grupo de artículos o numero.	Operario
2.- Realizar el descargo Manual del inventario por transacción Z	Operario
3. Verificación si fue o no descargada	Operario

La tarjeta CRC MANUAL, tiene como función realizar el descargo manual de inventario cuando hay algún problema con el descargo automático.

3.2.2 Soluciones Puntuales

La solución desarrollada será Multiplataforma, y está incluida dentro del estándar de SAP, se asignará códigos de transacción Z personalizadas, y estará sujeta a las seguridades por roles manejada por SAP como estándar.

La solución planteada permite:

- Consulta de órdenes.
- Verificación.
- Transmisión.
- Generación de archivos *.txt para maquinas de dosificación (Colorantes, Químicos Líquidos)
- Descarga automática de inventario (Uso BAPI)
- Impresión etiquetas.

3.2.3 Funcionalidad Mínima

El sistema planteado cuenta con una funcionalidad mínima la cual permite al usuario obtener lo que él solicita en la información descrita en las historias de usuario, esta respuesta tiene un aspecto operativo fundamental el de entregar solo características puntuales y no cosas adicionales que el desarrollador desea.

3.2.4 Reciclaje

La metodología XP establece que en lo posible se mantenga la menor cantidad de código comentado, esto facilita que el código sea de fácil entendimiento para cualquier integrante del equipo de desarrollo.

El código manejado en la presente solución se encuentra comentado al mínimo, tratando de dar solo una descripción breve en las cabeceras del cada programa, tratando de dar una continuidad y fluidez al código para comprender y en lo posible evitar complicaciones al desarrollador.

3.3 DESARROLLO

3.3.1 Flujo de trabajo

El flujo de trabajo para la solución planteada, inicia con las historias de usuario definidas en la fase de Planificación, con estas historias de usuario se definió en el plan de entregas en 28 días laborables.

El cliente procedió a seleccionar y priorizar las historias de usuario, a continuación se estimaron y elaboraron las pruebas correspondientes. Al desarrollo y conclusión de cada historia se realizaron entregas pequeñas, el flujo de la metodología se muestra en la figura 3.3

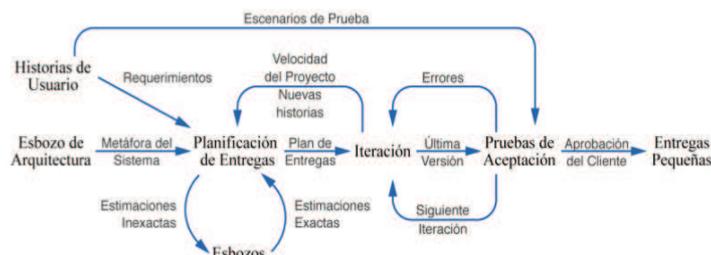


Figura 3.3 Flujo de trabajo

3.3.2 Interfaces de usuario

El sistema está desarrollado dentro del GUI de SAP, el cual es compatible con ambientes Unix y Windows, el software para la interfaz de usuario fue diseñado considerando su utilidad y presentación, así por ejemplo en la interfaz de usuario se utiliza:

- Etiquetas de texto: Las etiquetas se las usa para identificar el contenido de las cajas de texto, figura 3.4

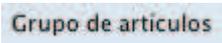


Figura 3.4 Etiqueta de Texto

- Cajas de texto: se utiliza cajas de texto para el ingreso y despliegue de información, figura 3.5



Figura 3.5 Caja de Texto

- Radio botones: Los radios botones son utilizados para las opciones de selección, figura 3.6

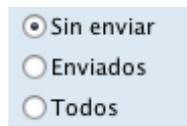


Figura 3.6 Radio Botones

- Check box: Los Check box sirven para seleccionar o no un dato para ser procesado, figura 3.7

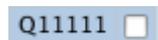


Figura 3.7 Check Box

- Botones: Los Botones sirven para realizar una acción. Ej. Guardar, Procesar, Enviar, etc, figura 3.8



Figura 3.8 Botón

- Data Grids: Los datagrids son utilizados para realizar acciones, desconectados de la fuente de datos, figura 3.9

Estado	Orden DYE	Reserva	Grupo artículos	Lote	...
<input type="checkbox"/>	1000010616	0000167592	1003	Q11111	<input type="checkbox"/>
<input type="checkbox"/>	1000010783	0000168975	1003	Q11111	<input type="checkbox"/>
<input type="checkbox"/>	1000010830	0000169609	1003	Q11111	<input type="checkbox"/>
<input type="checkbox"/>	1000010861	0000169922	1003	Q11111	<input type="checkbox"/>
<input type="checkbox"/>	1000010910	0000170683	1003	Q11111	<input type="checkbox"/>
<input type="checkbox"/>	1000010949	0000171215	1003	Q11111	<input type="checkbox"/>
<input type="checkbox"/>	1000010957	0000171419	1003	Q11111	<input type="checkbox"/>

Figura 3.9 Data Grids

A continuación en las figuras 3.10 y 3.11 se muestran algunas de las interfaces desarrolladas para este sistema

Figura 3.10 Ventana de transacción ZENVIO_INTERFACES

Estado	Orden DYE	Reserva	Pos.lista mat.	Material	Denominación	Grupo articulos	Ctd.necesaria	UM base	Almacén	Cantidad toma	Lote	...
COO	1000010549	0000166400	2090	1250040	SOLOFENIL TURQUEZA BRLE 400K	1002	13,300	G	2002	13,300	C11111	
COO		0000166400	2170		SOLOFENIL TURQUEZA BRLE 400K	1002	10,330	G	2002	10,330	C11111	
COO		0000166400	2160	1250043	TERASIL AMARILLO 4G	1002	1	G	2002	1	VU1019V9	
COO		0000166400	2080	1250046	TERASIL AZUL BGE-01 200K	1002	39,900	G	2002	39,900	Q5017500	
COO		0000166400	2150		TERASIL AZUL BGE-01 200K	1002	48,310	G	2002	48,310	Q5017500	
COO	1000010560	0000166572	1080		TERASIL AZUL BGE-01 200K	1002	0,150	G	2002	0,150	Q5017500	
COO		0000166572	1070	1250054	TERASIL ROJO FBN CONC.	1002	0,280	G	2002	0,280	WY0026V0	
COO	1000010570	0000166625	2090	1250008	CIBACRON TURQUEZA H-CN	1002	66,500	G	2002	66,500	Q6690600	
COO		0000166625	2080	1250022	REMAZOL AMARILLO BTE 3GL	1002	98,920	G	2002	98,920	QAC70005	
COO	1000010616	0000167592	3080	1250008	CIBACRON TURQUEZA H-CN	1002	23,660	G	2002	23,660	Q6690600	
COO		0000167592	3070	1250022	REMAZOL AMARILLO BTE 3GL	1002	64,900	G	2002	64,900	QAC70005	
COO	1000010633	0000167809	2100	1250008	CIBACRON TURQUEZA H-CN	1002	0,320	G	2002	0,320	Q6690600	
COO		0000167809	2110	1250022	REMAZOL AMARILLO BTE 3GL	1002	6,400	G	2002	6,400	QAC70005	
COO	1000010662	0000168109	3080	1250001	CIBACRON AMARILLO S-3R	1002	5,170	G	2002	5,170	24785400	
COO		0000168109	3130		CIBACRON AMARILLO S-3R	1002	15	G	2002	15	24785400	
COO		0000168109	3090	1250026	REMAZOL NARANJA RGB gran	1002	91	G	2002	91	AA10237	
COO		0000168109	3140		REMAZOL NARANJA RGB gran	1002	16	G	2002	16	AA10237	
COO	1000010799	0000169250	3060	1250001	CIBACRON AMARILLO S-3R	1002	35,660	G	2002	35,660	24785400	
COO		0000169250	3050	1250004	CIBACRON AZUL OSCURO W-R	1002	467,230	G	2002	467,230	27642300	

Figura 3.11 Ventana de datos para envío

3.3.3 Plantillas

- **Disponibilidad Del Cliente**

Durante el desarrollo del software, el cliente (Usuario Final) estuvo presente en todas las etapas del proceso, esta situación facilitó la realización de las pruebas de funcionamiento del sistema y a la vez se aclararon inquietudes manifiestas en la etapa de planificación.

La circunstancia descrita permitió la consolidación y fortificación del funcionamiento del software, lo que quiere decir que el cliente tuvo un nivel excepcional en el proceso.

- **Unidades de Pruebas**

En función al código se probó la eficacia y eficiencia del software, este código permitió identificar los errores de funcionamiento del sistema, en base a esto se definió las siguientes pruebas:

- Búsqueda de materiales a dosificar.

Esta prueba consistió en conocer en que estatus esta el material y si es posible volverlo a enviar nuevamente a dosificación o no.

El resultado de la prueba unitaria aplicada a la búsqueda de materiales generó datos satisfactorios.

- Generación de archivo plano para colorantes.

Esta prueba nos permitió identificar errores de formato en la creación del archivo plano necesario para la transferencia a la máquina de dosificación.

- Descarga de materiales de inventario.

Esta prueba no demostró el correcto funcionamiento del programa, además de la correcta utilización de la BAPI, que genera un movimiento de inventario automático tan solo con darle parámetros básicos necesarios.

3.3.4 Programación

- **Programación en parejas.**

Este paso dentro de la metodología XP no fue aplicado para este proyecto en particular, la razón principal radica en que este proyecto de tesis está siendo desarrollado por una sola persona.

3.3.5 Integración

En razón de que el desarrollo es individual, todo el código fue elaborado y unificado por una sola persona.

3.4 PRUEBAS

3.4.1 Procedimiento de Manejo

En Jersey por política interna se solicita que se realicen instructivos para que en base a estos se realicen las pruebas de implantación y aceptación, en el Anexo B encontrará detallado el instructivo.

3.4.2 Pruebas al sistema

Las pruebas de Implantación y de Aceptación, facilitaron obtener un producto final eficiente y un buen desempeño de la solución desarrollada.

Con la ayuda de ABAP unit transacción SCI dentro de SAP se pudo probar cada una de las pruebas unitarias realizadas, obteniendo una solución bastante estable y con errores controlados.

3.4.3 Pruebas de aceptación

El actor principal en este tipo de pruebas es el usuario final, en vista de que es quien indica que desea probar y está pendiente de los resultados que la solución le presente.

Consulta de Materiales

Resultado de la prueba: Satisfactorio para el usuario.

Generación de archivos para interfaces

Resultado de la prueba: Satisfactorio para el usuario.

Impresión de etiquetas

Resultado de la prueba: Satisfactorio para el usuario.

Descarga de inventarios

Resultado de la prueba: Satisfactorio para el usuario.

Descarga Manual de Materiales

Resultado de la prueba: Satisfactorio para el usuario.

Los resultados obtenidos de las pruebas realizadas dejaron al usuario final seguro de que el software desarrollado cumplió con todos los requerimientos.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- SAP al poseer herramientas de desarrollo dentro de su entorno permite la creación de todo tipo de solución personalizada, estas están orientadas a satisfacer la necesidad del cliente.
- La elaboración de las historias de usuario en la fase de diseño permite la materialización óptima y la concreción para abstraer y representar los requerimientos del mismo, ya que el usuario es el sujeto que sistematiza su propia solución al requerimiento.
- El uso simultáneo del plan de entregas e iteraciones dentro de la Metodología XP en la fase de diseño facilita al usuario el conocimiento del tiempo y de cuándo se finalizará toda o parte de la solución.
- En la etapa de análisis, la integración del usuario final dentro del equipo de desarrollo optimiza la productividad de los programadores o desarrolladores y despeja inquietudes que en el proceso se presenta.
- En la fase de implementación, la utilización de herramientas de pruebas unitarias integrada, permite al programador mayor concentración para escribir el código que da solución al negocio.

4.2. RECOMENDACIONES

- La Metodología utilizada en el presente proyecto, es factible de ejecutar tanto en proyectos medianos como en grandes, siempre y cuando se cuente con el compromiso decidido del cliente.
- Para la realización de soluciones a medida se recomienda llevar a cabo un minucioso levantamiento de información, este constará en la especificación de requerimientos.
- Es importante la correcta utilización del instructivo de uso para garantizar el funcionamiento óptimo del sistema.
- Es fundamental e importante para el programador la participación e integración del usuario en el proceso a fin de que verifique su necesidad y garantice la solución.
- En la carrera de Ingeniería de sistemas es necesario formar a estudiantes y profesores en el manejo de ERP'S.

4.3. ANEXO A: INSTRUCTIVO DE INSTALACIÓN DE SAP GUI PARA WINDOWS

4.4. ANEXO B: INSTRUCTIVO DEL PROGRAMADOR

BIBLIOGRAFIA

WWW.EXTREMEPROGRAMMING.ORG

WWW.PROGRAMACIONEXTREMA.ORG

WWW.XPROGRAMMING.COM

WWW.APOLOSOFTWARE.COM

WWW.SAP.COM

WWW.FOROSAP.COM

WWW.SAP4.COM