



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**MÉTODOS DE CONSULTA, EXTRACCIÓN, IMPORTACIÓN DE
INFORMACIÓN E INTERCOMUNICACIÓN DE SISTEMAS EXTERNOS A SAP**

José Fernando Ocheita Castro

Asesorado por el Ing. Nery José Marroquín Izaguirre

Guatemala, mayo de 2013

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**MÉTODOS DE CONSULTA, EXTRACCIÓN, IMPORTACIÓN DE
INFORMACIÓN E INTERCOMUNICACIÓN DE SISTEMAS EXTERNOS A SAP**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

JOSÉ FERNANDO OCHEITA CASTRO

ASESORADO POR EL ING. NERY JOSÉ MARROQUÍN IZAGUIRRE

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, MAYO DE 2013

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Walter Rafael Véliz Muñoz
VOCAL V	Br. Sergio Alejandro Donis Soto
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Marlon Antonio Pérez Türk
EXAMINADORA	Inga. Floriza Felipa Ávila Pesquera de Medinilla
EXAMINADORA	Inga. Sonia Yolanda Castañeda
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

MÉTODOS DE CONSULTA, EXTRACCIÓN, IMPORTACIÓN DE INFORMACIÓN E INTERCOMUNICACIÓN DE SISTEMAS EXTERNOS A SAP

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha del 16 de enero del 2013.



José Fernando Ocheita Castro

Guatemala, 13 de marzo del 2013

Ingeniero
Carlos Azurdia
Coordinador de Tesis

Estimado Ingeniero:

Por este medio le informo que el trabajo de tesis Titulado "**METODOS DE CONSULTA, EXTRACCION, IMPORTACION DE INFORMACION E INTERCOMUNICACION DE SISTEMAS EXTERNOS A SAP**", el cual está a cargo el estudiante **JOSE FERNANDO OCHEITA CASTRO** con carné **200113425**, de la carrera de Ingeniería en Ciencias y Sistemas, comprendido en el periodo de tiempo del 16 de enero del 2013 al 15 de marzo del 2013. El presente trabajo de tesis fue finalizado satisfactoriamente.

Agradeciendo la atención a la presente y quedando a sus órdenes para cualquier información adicional.

Atentamente,



*Ing. Nery Marroquín
Colegiado 9848*

Ing. Nery José Marroquín Izaguirre
Asesor



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 20 de Marzo de 2013

Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **JOSÉ FERNANDO OCHEITA CASTRO** carné 200113425, titulado: "METODOS DE CONSULTA, EXTRACCIÓN, IMPORTACIÓN DE INFORMACIÓN E INTERCOMUNICACIÓN DE SISTEMAS EXTERNOS A SAP", y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



E
S
C
U
E
L
A

D
E

C
I
E
N
C
I
A
S

Y

S
I
S
T
E
M
A
S

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación "MÉTODOS DE CONSULTA, EXTRACCIÓN, IMPORTACIÓN DE INFORMACIÓN E INTERCOMUNICACIÓN DE SISTEMAS EXTERNOS A SAP", realizado por el estudiante JOSÉ FERNANDO OCHEITA CASTRO, aprueba el presente trabajo y solicita la autorización del mismo.

"ID Y ENSEÑAD A TODOS"

Ing. Murlon Antonio Pérez Turk
Director, Escuela de Ingeniería en Ciencias y Sistemas



Guatemala, 29 de mayo 2013



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **MÉTODOS DE CONSULTA, EXTRACCIÓN, IMPORTACIÓN DE INFORMACIÓN E INTERCOMUNICACIÓN DE SISTEMAS EXTERNOS A SAP**, presentado por el estudiante universitario: **José Fernando Ocheita Castro**, procede a la autorización para la impresión del mismo.

IMPRÍMASE.

Ing. Murphy Olympo Paiz Recinos
Decano



Guatemala, mayo de 2013

/cc

ACTO QUE DEDICO A:

Dios	Porque me dio la vida, la oportunidad de estudiar una carrera universitaria y las fuerzas para poder concluirla.
Mis padres	Jorge Eduardo Ocheita y María Elizabeth Castro. Por su apoyo incondicional y por enseñarme con su ejemplo a esforzarme al máximo y no rendirme nunca.
Mis hermanos	Luis Eduardo, Madelaine, Estefania y Sebastian Ocheita Castro, porque siempre he podido contar con ustedes en cualquier situación, los quiero mucho.
Mis amigos	Nery Marroquín, Kristhian Herrera, Hugo Argueta y a todos los demás que no menciono. Por ser amigos verdaderos y personas en quien he podido confiar. Que Dios los bendiga.
Empresa EEGSA	Por permitir mí desarrollo profesional.

AGRADECIMIENTOS A:

Dios	Por darme la bendición de estudiar en la universidad.
Universidad de San Carlos de Guatemala	Por ser importante influencia en mi carrera, entre otras cosas.
Facultad de Ingeniería	Por ser una importante influencia en mi carrera entre otras cosas.
Mis padres	María Elizabeth Castro y Jorge Eduardo Ocheita, por brindarme su apoyo incondicional.
Mis amigos de la facultad	Nery Marroquín, Kristhian Herrera y a todos los demás que no menciono. Porque juntos avanzamos en esta carrera. Qué Dios los bendiga.
Mis amigos de la Empresa Eléctrica	Por haberme apoyado en el desarrollo del tema de tesis y por la confianza que me han brindado. Qué Dios los bendiga.
Los ingenieros	Marlon Türk, Carlos Azurdia y Floriza Avila, por toda la ayuda y orientación que me brindaron en estos últimos años.

1.3.1.	Transacción Se11.....	41
1.3.2.	Transacción Se16.....	49
1.3.3.	SQL ABAP.....	52
1.3.3.1.	Sentencia SELECT.....	54
1.3.3.2.	Sentencia INSERT.....	55
1.3.3.3.	Sentencia UPDATE.....	56
1.3.3.4.	Sentencia DELETE.....	57
1.3.3.5.	Sentencias COMMIT Y ROLLBACK....	57
1.4.	Activación de programas u objetos.....	58
1.5.	Cómo crear un programa.....	60
1.6.	Utilización del Debugger.....	70
1.7.	Como crear una orden de transporte.....	73
2.	IMPORTACIÓN, EXPORTACIÓN Y CONSULTA DE INFORMACIÓN A LOS MÓDULOS DE SAP.....	81
2.1.	Transacciones para consultas.....	81
2.1.1.	Transacción SQVI.....	81
2.2.	Procesos batch input.....	92
2.2.1.	Transacción SHDB.....	94
2.2.2.	Transacción SM35.....	100
2.2.3.	Ejemplo de creación de batch input.....	104
2.2.4.	Código del programa.....	108
2.3.	Transacciones para monitoreo.....	117
2.3.1.	Transacción ST05, Análisis de Rastreo SQL (SQL Trace).....	118
2.3.2.	Operaciones de base de datos medidas por el SQL Trace.....	125
2.4.	Winshuttle.....	125
2.4.1.	WinShuttle Transaction.....	126

2.4.2.	WinShuttle Query	142
2.5.	Análisis y comparación de ambos métodos Abap Vrs WinShuttle	150
2.5.1.	Transaction Shuttle y Abap	151
2.5.2.	Query Shuttle y Abap	152
3.	INTERCOMUNICACIÓN ENTRE SISTEMAS EXTERNOS HACIA SAP.....	153
3.1.	Consumo de Webservice de .Net desde SAP	153
3.1.1.	Ejemplo de consumo de Webservice.....	154
3.2.	Remote Function Call (RFC).....	190
3.2.1.	Configuración de destinos RFC despliegue y mantenimiento	192
3.2.1.1.	Transacción SM59	192
3.2.1.2.	Creación de función en Transacción SE37	196
3.3.	Análisis de utilización de Webservice y RFC (Remote Function Call)	212
3.3.1.	Análisis de Webservices en SAP.....	212
3.3.2.	Análisis de RFC (Remote Function Call).....	214
	CONCLUSIONES	215
	RECOMENDACIONES.....	217
	BIBLIOGRAFÍA.....	219

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Diagrama de capas de infraestructura sugerida para SAP	4
2.	Pantalla para escoger mandante ingresar en este caso es (Desarrollo)	5
3.	Pantalla de Login, para ingresar usuario y contraseña	6
4.	Escritorio de SAP	7
5.	Ingreso de comandos y utilización del menú de favoritos general	8
6.	Arquitectura Cliente Servidor	9
7.	Modelo petición respuesta	10
8.	Diagrama de cómo funciona la arquitectura de SAP	11
9.	Diagrama de arquitectura de SAP con dos servidores de aplicación, un servidor de base de datos y varios clientes	12
10.	Diagrama que presenta de cómo es el flujo de información en SAP .	13
11.	Diagrama como es el flujo de datos, cuando se ejecuta un programa en ABAP	14
12.	Diagrama cómo funcionan los procesos cuando se ejecuta una aplicación en ABAP	14
13.	Estructura general de una sentencia de ABAP	16
14.	Esquema de Programación en ABAP	17
15.	Utilización de comentarios	17
16.	Esquema de declaración de variables en ABAP	20
17.	Sentencia de control IF simple	21
18.	Sentencia de control IF con ELSE	21
19.	Sentencia de control IF con ELSEIF y ELSE	22

20.	Sentencia de control CASE WHEN.....	22
21.	Sentencia de control CASE WHEN con sentencia IF anidada	23
22.	Instrucción de bucle WHILE.....	24
23.	Instrucción de bucle WHILE con sentencia IF anidada	24
24.	Sentencia de bucle WHILE	25
25.	Sentencia de bucle DO	25
26.	Utilización de la instrucción CONTINUE	26
27.	Utilización de la instrucción EXIT.....	26
28.	Ventana de acceso a mandantes para acceder	42
29.	Ingreso de Mandante, Usuario, Contraseña e idioma	42
30.	Escritorio de SAP e ingreso a la transacción se11.....	43
31.	Ventana principal de la transacción se11 e ingreso de la tabla AUFK (Datos maestros de órdenes) para visualización	44
32.	Visualización de la tabla AUFK (Datos maestros de órdenes)	45
33.	Tablas relacionadas con la AUFK (Datos maestros de órdenes)	46
34.	Diagrama entidad relación, tablas relacionadas con la tabla AUFK (Datos maestros de órdenes)	47
35.	Selección de relación entre dos tablas	48
36.	Información de la relación entre las tablas AUFK y ONR00	48
37.	Información de tabla escogida dentro el diagrama entidad relación ..	49
38.	Ingreso de la transacción SE16	50
39.	Pantalla principal de la transacción se16 e ingreso de la tabla AUFK (Datos maestros de órdenes)	50
40.	Pantalla de selección de datos de la tabla AUFK (Datos maestros de órdenes)	51
41.	Información de la tabla AUFK (Datos maestros de órdenes) y selección de un registro	51
42.	Visualización de la información seleccionado de forma vertical	52
43.	Estructura de la tabla KNA1 (Maestro de clientes).....	53

44.	Diagrama de objetos de desarrollo en estado inactivo hacia estado activo	59
45.	Diagrama de generación de objetos de ejecución	60
46.	Ingreso a la transacción se38	61
47.	Ventana principal de la transacción Se38	61
48.	Pantalla de creación de un nuevo programa ABAP	62
49.	Pantalla para seleccionar el paquete al cual pertenecerá el programa o también puede ser objeto local.....	63
50.	Editor de programas ABAP en SAP	65
51.	Verificación de Sintaxis utilizando Ctrl + F2.....	66
52.	Barra de estado, donde se despliegan los mensajes de SAP	66
53.	Ventana de activación de programas utilizando Ctrl + F3.....	67
54.	Mensaje en la barra de estado que los objetos han sido activados ...	67
55.	Ejecución del programa prueba e ingreso de parámetros	68
56.	Resultado de la ejecución del programa de prueba.....	68
57.	Menú de acceso a la opción verificación de programas ampliada	69
58.	Ingreso de tipos de verificaciones a realizar	69
59.	Resultado del análisis de verificación de programa ampliada	70
60.	Ingreso de un break point.....	71
61.	Ejecución de programa para realizar una sesión DEBUG	71
62.	Entorno para DEBUG, del lado izquierdo el código ejecutándose y del lado derecho inspección de variables	72
63.	Inspección de variables.....	72
64.	Resultado del programa luego de ejecutarlo con F7	73
65.	Ingreso de la transacción Se10	73
66.	Pantalla principal de la transacción Se10.....	74
67.	Listado de órdenes de transporte, procesadas y no procesadas.....	75
68.	Creación de una nueva orden de transporte, para un programa nuevo	75

69.	Ingreso de información de una nueva orden de transporte, para un programa nuevo	76
70.	Actualización del listado de órdenes de transporte	77
71.	Liberación de orden de transporte seleccionada.....	77
72.	Ingreso de la transacción STMS	78
73.	Pantalla principal de la transacción STMS	78
74.	Selección de mandante (DEV, PRD, QAS) en la transacción STMS	79
75.	Orden de transporte no importada para el mandante PRD	79
76.	Ingreso de Fecha, Hora de la importación de la orden de transporte	80
77.	Pantalla principal de la transacción SQVI	82
78.	Pantalla de creación de una nueva consulta, para luego guardarla ..	82
79.	Área de trabajo para ingreso de tablas para consulta en la transacción SQVI.....	83
80.	Ingreso de la tabla a incluir en la consulta	83
81.	Vista de la tabla en el área de trabajo, en la transacción SQVI.....	84
82.	Ingreso de una tabla relacionada.....	84
83.	Diagrama entidad relación de tablas ingresadas en la transacción SQVI.....	85
84.	Navegación y vista del diagrama entidad relación de tablas ingresadas.....	86
85.	Pantalla de selección de campos de selección y de vista en la consulta a crear	86
86.	Campos a seleccionar	87
87.	Vista de selección de campos por tabla del lado izquierdo interfaz para escoger campos y del lado derecho los campos seleccionados.	88
88.	Ingreso de datos para consulta.....	88
89.	Resultado de consulta	89
90.	Selección de resultado en estilo de lista para luego hacer una exportación a Excel	89

91.	Exportar en Excel.....	90
92.	Selección de formato de salida, del resultado de consulta	90
93.	Tipo de salida si en tabla o tabla pivote.....	91
94.	Resultado en formato de Excel	91
95.	Diagrama de procesos general de batch input	94
96.	Pantalla principal de la transacción SHDB	96
97.	Cuadro de diálogo de nueva grabación de SHDB	96
98.	Información del contrato 475837 a modificar.....	97
99.	Ingreso de datos en la transacción ES21	97
100.	Pantalla de datos por contrato en la transacción ES21	98
101.	Modificación de dato, marcado en amarillo	98
102.	Código generado por la transacción SHDB después de parar la grabación	99
103.	Pantalla principal de la transacción SM35.....	100
104.	Estadística general de batch input	101
105.	Log general del estado de todos los batch input del sistema.....	101
106.	Selección de batch input a monitorear	102
107.	Información de batch input escogido	102
108.	Dynpros ejecutados	103
109.	Log de salida luego de la ejecución del batch input	103
110.	Diagrama de procesos del programa	104
111.	Grabación SHDB de la transacción IL01	105
112.	Tabla de verificación de existencia de información para el batch input.....	105
113.	Ejecución del programa.....	106
114.	Escoger el archivo a cargar.....	106
115.	Formato del archivo de entrada a cargar a la base de datos	107
116.	Salida del Log de errores I	107
117.	Salida del Log de errores II	108

118.	Declaración de tablas internas.....	108
119.	Inicio del Programa, llamando a las subrutinas Obtener_Archivo, Modo_Test, Procesa, End_Selection	109
120.	Subrutina Obtener_Archivo.....	109
121.	La subrutina Modo_Test	110
122.	Procedimiento Verifica_UT	111
123.	Procedimiento Procesa.....	111
124.	Procedimiento parámetros_batch	112
125.	Procedimiento batch input	112
126.	Código SHDB implementado en ABAP	113
127.	Procedimiento DynPro	114
128.	Llamada a la función Llamar transacción.....	114
129.	Procedimiento Llamar Transacción.....	115
130.	Función de generación de log de batch input	116
131.	Despliegue de log de batch input.....	117
132.	Pantalla inicial de la transacción	118
133.	Pantalla Filter Trace List	119
134.	Pantalla de resultados de las sentencias SQL	120
135.	Pantalla de sumarización.....	121
136.	Pantalla de monitoreo de una transacción en específico	122
137.	Pantalla de leyenda del reporte	122
138.	Leyenda del reporte.....	123
139.	Verificación de sentencias sql idénticas.....	124
140.	Pantalla de resultados de sentencias de sql idénticas	124
141.	Se tienen los datos empresariales en Excel	127
142.	Pantalla principal de Transaction Shuttle o WinShuttle Transaction.....	127
143.	Proceso de creación de transacción en Transaction Shuttle.....	128
144.	Se ingresa el código de transacción y luego se selecciona el módulo de SAP a acceder	129

145.	Login a SAP en Transaction Shuttle.....	130
146.	Transacción de cambio de material de SAP.....	130
147.	Selección de datos básicos para la grabación de la transacción	131
148.	Inicio de grabación de ingreso de datos en la transacción	132
149.	Mapeo de datos	133
150.	Código de la transacción y archivo de Excel juntos.....	134
151.	Proceso de transposición de datos entre Excel y SAP o viceversa .	135
152.	Carga de datos en tiempo real de Excel hacia SAP	136
153.	Ejecución de los errores del archivo de Excel	137
154.	Realización de carga de datos erróneos	138
155.	Log Viewer de Transaction Shuttle.....	139
156.	Planificador de carga automático Transaction Shuttle.....	140
157.	Utilización de plantillas de carga	141
158.	Carga y validación de datos en tiempo real.....	142
159.	Selección de nueva consulta.....	143
160.	Pantalla de Login	144
161.	Selección de tablas de SAP.....	144
162.	Selección de campos de la tabla seleccionada	145
163.	Tablas seleccionadas para realizar la consulta	145
164.	Vista preliminar de la consulta.....	146
165.	Guardar la plantilla de consulta	146
166.	Asignación de los campos seleccionados al archivo de salida	147
167.	Selección de campos de la consulta al archivo de salida de Excel..	148
168.	Exportación del reporte a un archivo de Excel	149
169.	Grabación de archivo de salida Query Shuttle	149
170.	Reporte en archivo de Excel	150
171.	Diagrama de flujo de consumo de web service con SAP	153
172.	Creación de nuevo proyecto	154
173.	Creación de Servicio Web.....	155

174.	Código servicio Web AspNet	156
175.	Deshabilitación de los métodos HttpGet y HttpPost al protocolo Http.	157
176.	Publicación del servicio Web	157
177.	Publicación del servicio web en sistema de archivos	158
178.	Publicación del servicio web, utilizando IIS	158
179.	Servicio web publicado wsConexionSAP	159
180.	Opción de Examinar Aplicación	159
181.	Servicio web publicado	160
182.	Método Prueba_Comunicacion_A_SAP	160
183.	Resultado del método Prueba_Comunicacion_A_SAP	161
184.	Cambio del namespace en el servicio web	162
185.	Transacción SE80	163
186.	Creación de objeto de cliente proxy para comunicación con un servicio web.....	163
187.	Selección de fuente de Servicio.....	164
188.	Ingreso del link del WSDL del servicio Web en SAP.....	165
189.	Pantalla de ingreso de paquete para el cliente proxy.....	166
190.	Pantalla de finalización de creación de proxy	167
191.	Creación de tablas internas de SAP, luego de reconocer el servicio web.....	167
192.	Guardando y activando el servicio web.....	168
193.	Creación de puerto lógico en SAP con la transacción LPCONFIG, para consumo de servicio web	168
194.	Creación del puerto lógico	169
195.	Ingreso de parámetros de puerto lógico en SAP.....	170
196.	SoapAction del método Prueba_Comunicacion_A_SAP.....	171
197.	Definición del método Prueba_Comunicacion_A_SAP en la puerta lógica	171
198.	Parametrizaciones globales, mantenimiento de status	171

199.	Mensaje de información de parametrización global en Ipconfig.....	172
200.	Mensaje de mantenimiento de status en Ipconfig	172
201.	Cheque de mantenimiento de status en Ipconfig	172
202.	Mensaje de éxito de grabación y activación del puerto lógico	173
203.	Prueba del servicio Web, presionando el botón F8	173
204.	Ingreso del nombre del puerto lógico	174
205.	Búsqueda del método a ejecutar	174
206.	Salida de la ejecución del método del Servicio Web, ejecución del SOAP ACTION	175
207.	Resultado de ejecución del método del servicio web	175
208.	Método de multiplicación, pidiendo parámetros de entrada	176
209.	Resultado de ejecución del método de multiplicación	176
210.	Editor XML para cambiar parámetros del archivo a enviar al servicio web	177
211.	Cambio de parámetros para la llamada del método del servicio web	177
212.	Resultado de la llamada del método con parámetros	178
213.	Diagrama de flujo de consumo de webservice con ABAP	178
214.	Creación de programa para consumo de servicio web	179
215.	Parámetros para creación del programa de consumo de servicio web	180
216.	Grabación del paquete y objeto local	180
217.	Editor de Abap, para la programación del consumo del servicio web	181
218.	Icono para ver el listado de objetos locales	181
219.	Listado de objetos locales	182
220.	Botón para visualizar el listado de objetos que pertenecen al paquete	182
221.	Selección del Enterprise Services en SAP	183

222.	Código para la llamada del servicio web	184
223.	Llamada al método PRUEBA_COMINICACION_A_SAP del servicio web	185
224.	Plantilla de programación ABAP, de la llamada del método del servicio web.....	186
225.	Código de la llamada del método PRUEBA COMUNICACIÓN A SAP	187
226.	Resultado de la llamada del método PRUEBA COMUNICACIÓN A SAP	187
227.	Código de llamada del método del servicio web	189
228.	Código de inicio de selección del programa.....	189
229.	Ejecución del programa	190
230.	Resultado de la multiplicación, utilizando servicio web	190
231.	Diagrama de procesos de funcionamiento de RFC.....	191
232.	Diagrama de flujo para consumir una RFC	191
233.	Pantalla de configuración de conexiones RFC.....	192
234.	Creación de Conexión ABAP	193
235.	Pantalla de ingreso de Opciones Técnicas del RFC Destino	194
236.	Pantalla de ingreso de seguridad del destino RFC	195
237.	Prueba de conexión de Destino RFC.....	196
238.	Log de prueba de conexión de destino RFC.....	196
239.	Transacción SE37	197
240.	Creación de función.....	197
241.	Atributos de la función de acceso remoto	198
242.	Ingreso de parámetros en la función.....	199
243.	Ingreso de parámetro de salida de la función	199
244.	Código fuente de función RFC.....	200
245.	Ingreso de parámetros de la función.....	201
246.	Resultado de la función	201

247.	Creación de nuevo proyecto en Visual Studio .Net	202
248.	Aplicación WEB vacías de ASP.NET	203
249.	Ingreso de archivos DLL en la carpeta del proyecto	203
250.	Agregar referencia de archivos DLL	204
251.	Selección de archivos DLL del proyecto.....	205
252.	Agregando nuevo elemento en el proyecto	206
253.	Creación de clase en Visual Studio .Net	207
254.	Clase de conexión a SAP y consumo de funciones RFC	208
255.	Diseño de página web de utilización de RFC	210
256.	Creación de la instancia y ejecución de la función RFC	211
257.	Resultado de la llamada a la función RFC de SAP.....	212

TABLAS

I.	Tipos de datos en ABAP	19
II.	Sentencias Open SQL ABAP	53
III.	Características y ventajas de Transaction Shuttle Vrs Abap	151
IV.	Características y ventajas de Query Shuttle Vrs Abap	152
V.	Características y ventajas de WebServices en SAP.....	213
VI.	Características y ventajas de RFC	214

GLOSARIO

ABAP	Lenguaje avanzado para programación de aplicaciones de negocios (Advanced Business Application Programming), lenguaje nativo de SAP.
APPEND	Comando para manejo de base de datos, el cual consiste en agregar uno o más registros nuevos al final de la tabla que se está operando.
BATCHINPUT	Es un proceso para carga masiva de datos utilizando las transacciones que fueron grabadas con la transacción SHDB.
DELETE	Comando que se utiliza en el lenguaje Abap, el cual sirve para poder eliminar registros de una tabla interna.
DEV	Desarrollo, mandante o ambiente para el desarrollo de aplicaciones ABAP.
GUI	Interfaz gráfica de usuario (Graphic User Interface).
HTML	Lenguaje de marcado de hipertexto (HyperText Markup Language).
INSERT	Comando para manejo de base de datos, el cual consiste en anexar un registro al final de una tabla que contiene los valores del campo especificados.

JAVA	Lenguaje de programación, libre de plataforma.
MODIFY	Comando que se utiliza en el lenguaje Abap, el cual cambia los valores dentro de una tabla interna.
MOVE	Comando que se utiliza en el lenguaje Abap, el cual es el operador de asignación de contenido a cualquier variable, es equivalente al igual.
PRD	Productivo, mandante o ambiente en el cual los desarrollos ya se están utilizando por el usuario final ya con el aseguramiento de calidad.
QAS	Aseguramiento de calidad (Quality Assurance), mandante o ambiente para pruebas de aplicaciones ABAP.
RFC	Función llamada de forma remota (Remote Function Call).
SAP	Productos para procesamiento de información y Sistemas de aplicación (Systems Applications and Products for data processing).
SAPOPENSQ	Lenguaje estructurado de consultas hacia la base de datos (Structured Query Language, database query language).
SE10	Transacción que permite organizar los ordenes de transporte. Desde esta transacción el administrador puede organizar todas las tareas relacionadas con los ordenes así como crearlas, modificarlas, ver los registros y la pantalla de todos los transportes que el cliente entrega.

SE11	Transacción para la administración de todas las definiciones de datos usadas en R/3.
SE16	Transacción que permite buscar los datos en las tablas especificadas.
SE37	Transacción para la creación, modificación y eliminación de funciones desarrolladas en ABAP.
SE38	Transacción para poder crear, modificar y eliminar desarrollos de programas ABAP.
SHDB	Transacción para grabación de eventos para batch input.
SM35	Transacción para monitoreo de Sesiones de Batch Input, mostrando estadísticas de cada uno de ellos.
SQVI	Transacción en la cual está diseñada para que el usuario pueda cubrir sus propias necesidades de reportes o de información puntual creando sus propias consultas.
ST05	Transacción para ver el análisis de rendimiento de consultas SQL en programas estándar o desarrollados en SAP.
STMS	Transacción para la administración de órdenes de transporte hacia otros mandantes.
UNIX	Sistema operativo portable, multitarea y multiusuario.
WINDOWS	Sistema operativo desarrollado por Microsoft, es multitarea y multiusuario.

WRITE

Comando que se utiliza en el lenguaje Abap, el cual sirve para imprimir en pantalla valores, mensajes, etc.

RESUMEN

El trabajo está dirigido a usuarios finales e ingenieros en sistemas que no tienen conocimiento de SAP, pero si tienen los conceptos básicos de sistemas.

En el capítulo 1 se muestra una breve historia de la creación de SAP, manejo básico de la interfaz de usuario, arquitectura de SAP y programación básica en ABAP. Esto es con el objetivo de tener conocimiento rápido de SAP sin estudiar grandes contenidos de estudio y para el entendimiento de los siguientes capítulos.

Los capítulos 2 y 3, están dirigido a ingenieros en sistemas que tienen una gran experiencia en análisis y desarrollo de software, ya que pueden tener un buen juicio de las diferencias, ventajas y desventajas de los métodos de importación, exportación de información e intercomunicación de sistemas externos a SAP.

Este tema de tesis esta también dirigido a personas que no tienen mayor experiencia en análisis y desarrollo de software, ya que se muestra en cada uno de los capítulos una serie de pasos a seguir para importar, exportar e intercomunicar utilizando los ejemplos mostrados, ya sea utilizando la interfaz de usuario de SAP o programación en ABAP. Estos ejemplos fueron desarrollados y puestos en prueba para la elaboración de este tema de tesis, por lo que se asegura que se puede utilizar este documento para poder realizar los desarrollos y ponerlos en productivo.

OBJETIVOS

General

Presentar una propuesta en la cual pretende dar una solución que sea de muy bajo costo con respecto a la intercomunicación de sistemas externos a SAP, así como también brindar las ventajas y desventajas que con lleva a la hora de extracción e importación de información a SAP .

Específicos

1. Mostrar las ventajas y desventajas para importación y extracción de información utilizando:
 - a. Transacciones de SAP, BatchInputs
 - b. Utilización de Winshuttle
 - c. Mostrar la flexibilidad que muestra Winshuttle para el manejo de información de SAP.

2. Mostrar las formas de intercomunicación de sistemas externos hacia SAP:
 - a. WebServices
 - b. Remote Function Call
 - c. Mostrar que la intercomunicación entre sistemas externos y SAP, es confiable y sencilla.

INTRODUCCIÓN

El siguiente trabajo de investigación pretende brindar una breve descripción de lo que es SAP R/3, así como también dar a conocer las diferentes formas de importación, exportación y consulta de información a los módulos de SAP, incluyendo sus ventajas, desventajas y en qué momentos utilizar la vía más adecuada.

Esto es de gran utilidad ya que permite saber que método utilizar para realizar operaciones de manera efectiva, tomando en cuenta el factor económico.

La intercomunicación de sistemas externos hacia SAP es muy importante ya que permite que se minimicen costos en implementación de otros módulos de SAP.

1. INTRODUCCIÓN A SAP

SAP AG es una empresa multinacional alemana dedicada al diseño de productos informáticos de gestión empresarial, tanto para empresas como para organizaciones y organismos públicos.

SAP (Systems Applications and Products for data processing), la fundaron cuatro ingenieros de IBM, Dietmar Hopp, Klaus Tschira, HansWerner Hector, Hasso Plattner y Claus Wallenreuther. Aunque en un principio la compañía evolucionó a partir de su software de gestión, poco a poco se han ido añadiendo nuevos productos y compañías que han completado un gran conglomerado de soluciones. Sin embargo, todos sus productos han tenido siempre un enfoque común, la mejora de la gestión empresarial.

Su breve historia está marcada por los siguientes sucesos:

- 1972 Ex empleados de IBM fundan SAP en Manheim, Alemania
- 1973 Se lanza la versión SAP R/1
- 1976 Se funda SAP GmbH. Traslada su sede a Walldorf
- 1979 Se lanza la versión de SAP R/2
- 1988 SAP GmbH se convierte en la corporación SAP AG
- 1992 Se libera la primera versión R/3
- 1995 Se empieza a cotizar en el índice DAX Alemán
- 2003 SAP comienza a cotizar en el Dow Jones Stoxx 50
- 2003 SAP anuncia su plataforma integrada SAP Netweaver
- 2007 SAP compra Business Objects

- 2010 SAP compra Sybase
- 2010 SAP libera la primera versión de SAP HANA
- 2011 Se anuncia la compra de SuccessFactors
- 2012 Se anuncia la compra de Ariba Inc.

1.1. **SAP R/3**

SAP R/3 es un grupo de módulos integrados que sirven para poder procesar datos para grandes corporaciones y su propósito es de proveer un grupo de módulos empresariales integrados para poder administrar corporaciones.

SAP R/3 está compuesto por los siguientes módulos:

- PP (Production Planning, Planificación de la Producción)
- MM (Materials Management, Gestión de Materiales)
- SD (Sales and Distribution, Ventas y Distribución)
- FI (Financial Accounting, Contabilidad Financiera)
- CO (Controlling, Controlador)
- AM (Fixed Assets Management, Gestión de Activos Fijos)
- PS (Project System, Proyecto del Sistema)
- WF (WorkFlow, Flujos de Trabajo)
- IS (Industry Solutions, Soluciones de industria)
- HR (Human Resources, Recursos Humanos)
- PM (Plant Maintenance, Mantenimiento de planta)
- QM (Quality Management, Gestión de Calidad).

Cada uno de estos módulos son necesarios entre ellos mismos, por ejemplo: el módulo de Gestión de Materiales necesita los vínculos del módulo de Ventas y Distribución y el módulo de Contabilidad Financiera.

1.1.1. Navegación en el sistema

El SAP GUI está basado en el estilo Windows y está disponible para diferentes plataformas, brindando las mismas funcionalidades para cada una. Si se llega a aprender a utilizar el SAP GUI en una plataforma no hay ningún problema en utilizar SAP en otra plataforma diferente.

Los diferentes tipos de SAP GUI disponibles son:

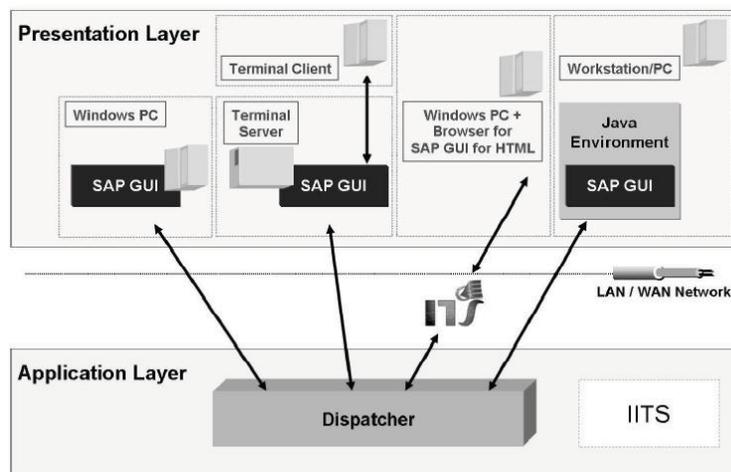
- SAP GUI en la plataforma de Windows
 - Este ambiente está desarrollado con C o C++, su flujo de información entre la capa de presentación y la capa de aplicación es de forma muy efectiva pues solo viajan a través de la red pocos KB, por lo que es muy efectiva la conexión entre los servidores u otras redes.
- SAP GUI en la plataforma de JAVA
 - Esta ambiente está desarrollado en JAVA, es independiente de la plataforma ya que se puede utilizar en Windows, Linux, Solaris, etc. También su flujo de información entre capas y redes es muy eficiente.
- SAP GUI para HTML.
 - Este consiste en el web server de SAP el cual se llama Internet Transaction Server (ITS). Este servidor requiere convertir la información desde la web hacia el SAP WEB APPLICATION

SERVER y viceversa. El SAP GUI HTML utiliza el modelo petición respuesta.

- Una de las ventajas de este entorno es que no es necesario la instalación del SAP GUI en cada máquina ya que solo se acceso a una dirección desde el buscador (Internet Explorer, Firefox..., etc.)

A continuación se presenta la versatilidad de conexión que brinda SAP.

Figura 1. **Diagrama de capas de infraestructura sugerida para SAP**



Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 36.

1.1.2. Ingreso al sistema

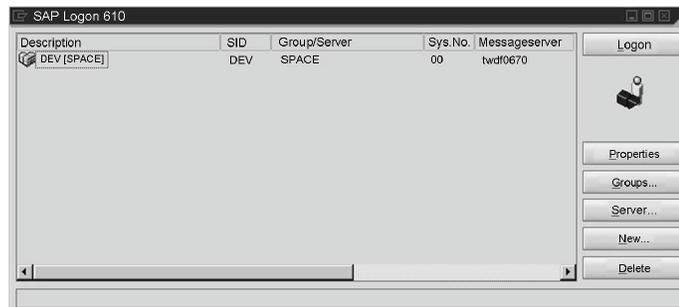
Para iniciar el SAP GUI se puede iniciar desde línea de comandos escribiendo SAP LOGON, luego puede acceder al proceso de ingreso al sistema. Para ello le muestra una pantalla la cual le presenta los diferentes entornos de SAP, los cuales son:

- DEV (Desarrollo)

- Entorno de desarrollo de programas ABAP
- QAS (Quality Assurance)
 - Entorno de pruebas de programas desarrollados en ABAP
- PRD (Productivo)
 - Programas ya en productivo, listos para el usuario final.

Entonces se escoge el entorno y luego se le da al botón de logon.

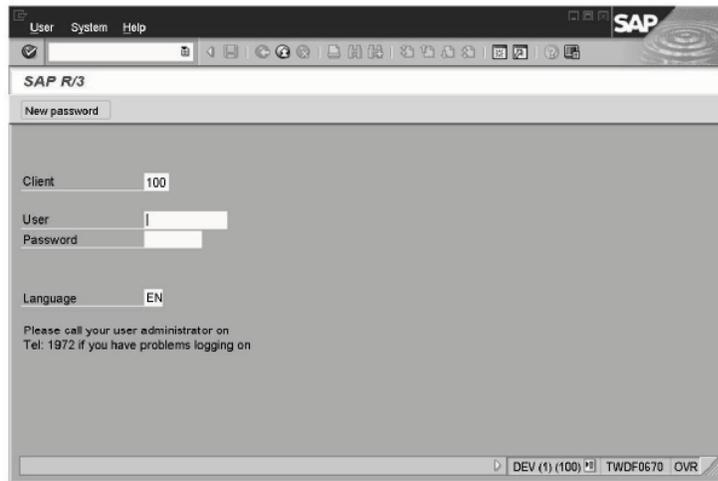
Figura 2. **Pantalla para escoger que mandante ingresar en este caso es DEV (Desarrollo)**



Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 37.

Luego presenta la ventana de ingreso de Usuario y Contraseña.

Figura 3. **Pantalla de Login, para ingresar usuario y contraseña**



Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 38.

Si el usuario ingresa al sistema más de una vez desde diferentes ubicaciones, el sistema despliega mensajes de advertencia el cual presenta las siguientes opciones:

- Continuar con este ingreso y terminar los demás ingresos al sistema.
- Continuar con este ingreso sin terminar los demás ingresos al sistema (este es el ingreso actual).
- Terminar este ingreso.

Inicialmente la contraseña asignada cuando se ingresa al sistema por primera vez, durante el ingreso el sistema pide un cambio de contraseña.

Esto es por razones de seguridad y tiene sentido cambiar de contraseña de vez en cuando.

Luego de ingresar el usuario y contraseña adecuados, el sistema presenta el menú llamado SAP EASY ACCESS.

Figura 4. Escritorio de SAP



Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 49.

En donde está señalado que dice ROLE-BASED Menú es el cual muestra las transacciones asignadas a las que pueda acceder el usuario según el rol que se le haya asignado por medio del administrador.

En donde se muestra FAVORITES LIST es un listado en el cual el usuario final puede guardar las transacciones que hace uso, se podría decir que son accesos directos.

En el Easy Access en la parte superior donde dice Command Field como se muestra en la imagen se puede ingresar:

- Códigos de transacción.
- /n para cancelar la transacción actual (/nse38)

- /o para desplegar las sesiones abiertas
- /oXXXX para llamar la transacción XXXX en una nueva sesión
- /nend para terminar la sesión con confirmación
- /nex para terminar la sesión sin confirmación
- /i para borrar la sesión que se está utilizando actualmente.

Figura 5. Ingreso de comandos y utilización del menú de favoritos y menú general



Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 51.

1.1.3. Arquitectura principal

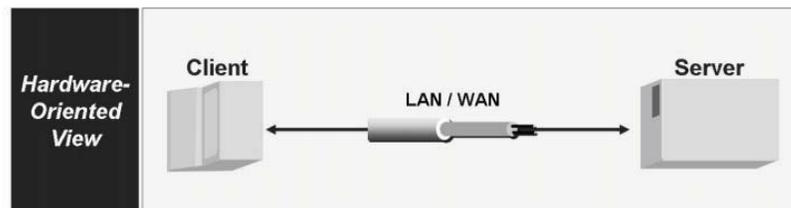
SAP es utilizado para llevar a cabo los procesos o aplicaciones del negocio. Estas deben de ser implementadas independientemente del ambiente de software utilizado (sistema operativo, bases de datos) y hardware.

SAP brinda el ambiente de programación ABAP (Advanced Business Application Programming) Runtime Environment, el propósito de este lenguaje es permitir desarrollar aplicaciones de negocio de alta escalabilidad. Mayormente los clientes utilizan este lenguaje para el desarrollo de aplicaciones totalmente nuevas y también para mejorar, modificar aplicaciones estándar de SAP.

1.1.3.1. Configuración cliente/servidor

Desde el punto de vista de hardware el término servidor significa tener un servidor central en la red que provee información, memoria, recursos para todas las estaciones de trabajo (clientes).

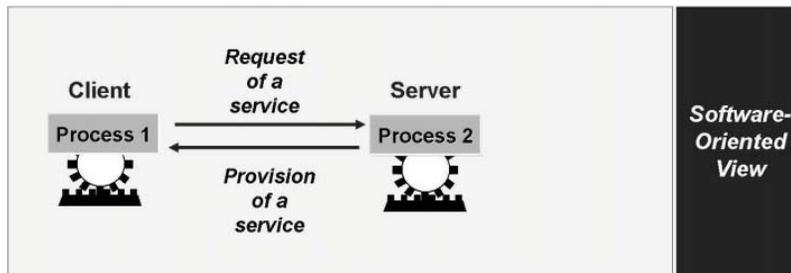
Figura 6. **Arquitectura Cliente Servidor**



Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 82.

Desde el punto de vista de software, el cliente y el servidor están definidos en el nivel de servicios, estos consisten en un proceso o un grupo de procesos, que estos son pedidos por el cliente y el servidor da la respuesta a dichos pedidos.

Figura 7. **Modelo petición respuesta**



Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 82.

La arquitectura de SAP se distribuye en diferentes capas las cuales son:

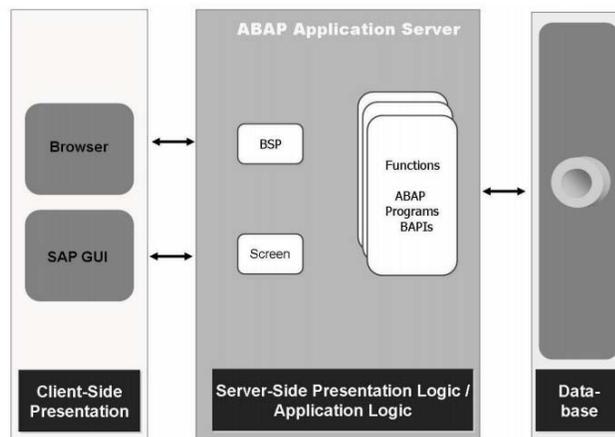
- Capa de presentación (procesos de la interfaz de usuario)
- Capa de aplicación (procesos para la ejecución de programas)
- Capa de datos (procesos para gestionar y organizar la base de datos)

Para poder configurar SAP se debe de decidir cómo se van a distribuir los procesos de acuerdo al hardware disponible. Hay varias configuraciones de un nivel o varios niveles, dependiendo del número de clientes. SAP recomienda utilizar algunas de estas configuraciones que se presentan a continuación:

- Configuración de una sola capa: esto significa que todos los procesos (base de datos, aplicación y presentación) estén configurados en una sola máquina. Se puede decir que esta es la forma clásica, pero no es muy recomendable ya que hay sobrecarga tanto en memoria como en procesador, por lo que el funcionamiento sería de bajo rendimiento. Pero esta configuración se utiliza para demostraciones, por ejemplo en una laptop.

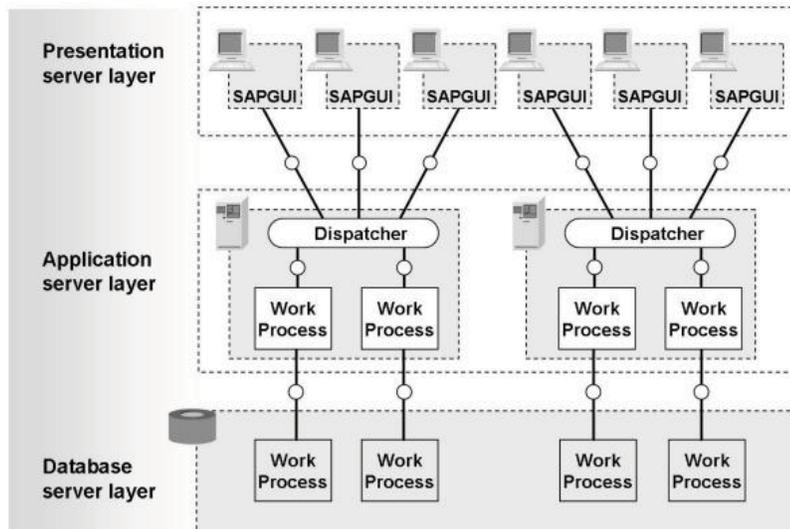
- Configuración de dos capas: la capa de presentación queda instalada en los clientes y las capas de aplicación y datos están en un servidor. Esta configuración mantiene un buen funcionamiento manteniendo a varios usuarios trabajando simultáneamente. Pero aumenta los costos ya que cada máquina del cliente debe tener buena capacidad para tener instalado el GUI de SAP, a la vez si el número de clientes aumenta más allá de la capacidad que el servidor puede atender, se corre el riesgo que suceda un cuello de botella. Por lo que la solución para atender este problema es añadir otro servidor que se ocupe en la capa de aplicación.
- Configuración de tres capas: cada capa funciona en su propio servidor, esta configuración da la ventaja que todo puede funcionar de manera simultánea, por lo que el funcionamiento es muy eficiente.

Figura 8. **Diagrama de cómo funciona la arquitectura de SAP**



Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 84.

Figura 9. **Diagrama de arquitectura de SAP con dos servidores de aplicación, un servidor de base de datos y varios clientes**



Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 172.

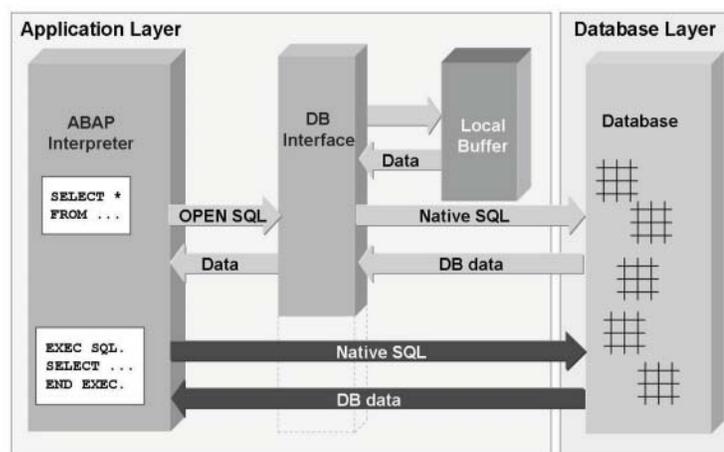
1.1.3.2. **Comunicación de SAP R/3 con la base de datos**

SAP puede ser utilizado por variedad de sistemas operativos y sistemas de bases de datos. Para poder acceder a la información almacenada en la base de datos se debe de realizar por medio de la capa de aplicación utilizando programas ABAP, pero estos deben de tener las sentencias necesarias para poder realizar las consultas a la base de datos.

Con ABAP se puede utilizar SAP Open SQL (Structured Query Language, database query language). La interface de la base de datos traduce las sentencias Open SQL programadas con ABAP al lenguaje correspondiente SQL Nativo de la base de datos que se tiene instalada.

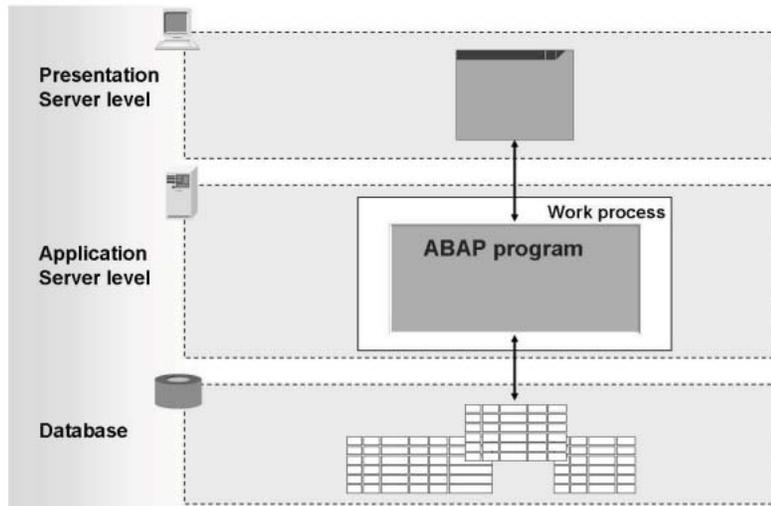
Cuando se está interpretando las sentencias Open SQL programadas con ABAP, la interface de base de datos de SAP verifica la sintaxis de estas sentencias y automáticamente asegura la utilización óptima de los Buffers de SAP que es memoria compartida.

Figura 10. **Diagrama que presenta de como es el flujo de información en SAP**



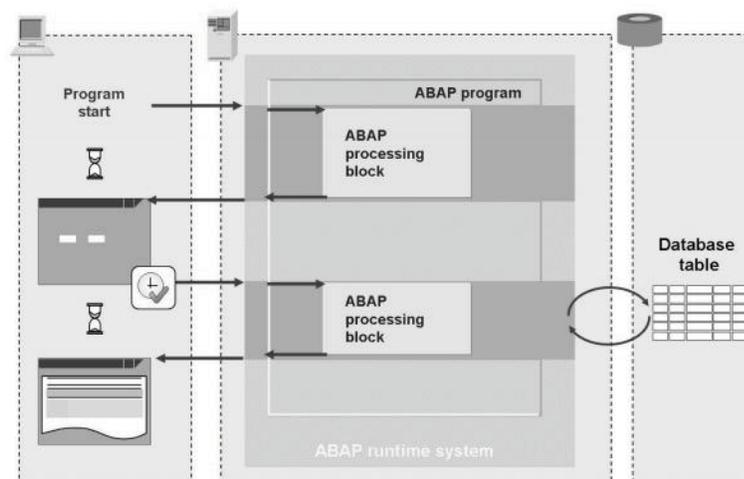
Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 111.

Figura 11. **Diagrama como es el flujo de datos, cuando se ejecuta un programa en ABAP**



Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 173.

Figura 12. **Diagrama cómo funcionan los procesos cuando se ejecuta una aplicación en ABAP**



Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 175.

1.2. 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 (R/3, MySap, Business Suite...). Utiliza sentencias OPEN SQL para conectarse con prácticamente cualquier base de datos. Cuenta con miles de funciones para el manejo de archivos, bases de datos, fechas, etc.

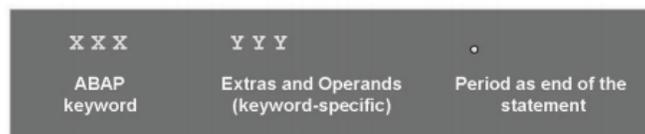
1.2.1. Características de ABAP

ABAP es un lenguaje el cual contiene las siguientes características:

- Es tipado: significa que soporta procesamiento de tipos de datos (carácter, entero, real, etc.) conversión de tipos por ejemplo de carácter a entero o viceversa.
- Permite aplicaciones con diferentes lenguajes (Inglés, Español...) utilizando elementos de texto traducibles.
- Permite acceso a SQL: con las sentencias de ABAP y Diccionario de datos, se puede tener acceso directo a la base de datos.
- Un lenguaje orientado a objetos
- Independiente de la plataforma: esto significa que siempre tiene la misma funcionalidad o significado sin depender del lenguaje de programación SQL de la base de datos y el sistema operativo. Por ejemplo si se tienen varios desarrollos en ABAP que tienen acceso a la base de datos y la base de datos es Oracle 10g, se puede hacer un cambio de motor de base de datos a SQL Server que esto no afectaría los desarrollos en ABAP. Al igual si se cambia de sistema operativo de un UNIX a Windows que tampoco afectarían los desarrollos en ABAP.

- Es especialmente diseñado para aplicaciones basadas en dialogo: esto significa que se pueden programar formularios para que el usuario pueda ingresar información utilizando una lógica del negocio.
- Compatible hacia arriba: esto significa que los programas de ABAP se pueden ejecutar en versiones posteriores.

Figura 13. **Estructura general de una sentencia de ABAP**



Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 202.

- Cada sentencia debe de terminar con un punto.
- Las palabras deben de estar al menos separadas por un espacio.
- La primera palabra en la sentencia se llama palabra llave o palabra reservada.
- Las sentencias pueden llevar más de una línea.

Ejemplo de un programa en ABAP:

Figura 14. Esquema de Programación en ABAP

```
...  
DATA wa_scarr TYPE scarr .  
  
SELECT * FROM scarr  
      INTO wa_scarr .  
NEW-LINE .  
WRITE wa_scarr-carrid .  
WRITE wa_scarr-carrname .  
WRITE wa_scarr-url .  
ENDSELECT .
```

Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 202.

Figura 15. Utilización de comentarios

```
* comments ... }  
* comments ... } Comments  
* comments ... } (whole lines)  
  
DATA wa_scarr TYPE scarr. " comments ... }  
SELECT * FROM scarr      " comments ... } rest-of-line  
      INTO wa_scarr.     " comments ... } comments  
NEW-LINE .  
WRITE : wa_scarr-carrid , }  
      wa_scarr-carrname , } Chained statement  
      wa_scarr-url .  
ENDSELECT .
```

Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 203.

- Lo más recomendable para poder realizar un comentario es utilizar las dobles comillas “ya que estas son reconocidas en cualquier parte del

código, mientras que utilizar los asteriscos *, estos deben de ir al principio de la línea del código de lo contrario el sistema no detecta el asterisco como un comentario.

- La variable SY-SUBRC sirve para que el sistema escriba el código de retorno e indica información del éxito de la ejecución de la sentencia realizada o ejecutada anteriormente. El valor 0 indica que la sentencia fue ejecutada con éxito de lo contrario hay algún error.

1.2.2. Tipos de datos

La longitud esta medida en número de caracteres no en número de *bytes*.

Tabla I. Tipos de datos en ABAP

TIPOS ELEMENTALES:	Tipo	Long. Defecto	Long. Máx. (nº char)	Valor inicial
Alfanumérico (Char, string)	C	1	de 1 a 65535	SPACE
Fecha(Date)	D	8	8	'00000000'
Coma flotante (Float)	F	8	8	0
Entero (Integer)	I	4	4	0
Numéricos (Natural)	N	1	de 1 a 65535	'0...0'
Empaquetados (Packed)	P	8	de 1 a 16	0
Hora (Time)	T	6	6	'000000'
Hexadecimales (Hex)	X	1	de 1 a 65535	X'00'

Fuente: http://www.abap.es/centro_Manual_03_tipos.htm. Consulta: 18 de enero del 2013.

Todas las variables deben ser declaradas antes de ser utilizadas. La forma de declarar una variables es: nombre, tipo, longitud (si aplica), modificadores adicionales (por ejemplo, número de decimales para los datos empaquetados) y opcionalmente el valor inicial.

Figura 16. **Esquema de declaración de variables en ABAP**

```
DATA: contador TYPE i,  
      acumulador TYPE i VALUE 0,  
      factor(2) TYPE p DECIMALS 2,  
      nombre(20) TYPE c.
```

Fuente: http://www.todoabap.com.ar/tipos_datos.html. Consulta: 19 de enero del 2013.

La palabra reservada VALUE sirve para asignar un valor a la variable en el momento de declaración.

La palabra reservada DECIMALS sirve para especificar el número de decimales que lleva la variable.

En la declaración de la variable factor no es una variable si no un arreglo de números de decimales con precisión 2.

1.2.3. **Cálculo de expresiones aritméticas**

En ABAP se puede programar expresiones aritméticas anidarse hasta cualquier profundidad:

- + Suma
- - Resta
- * Multiplicación
- / División
- ** Exponenciación
- DIV División sin residuo
- MOD Residuo de una división

1.2.4. Sentencias de control

Para ejecutar instrucciones sólo si se cumplen ciertas condiciones ABAP tiene dos bloques de instrucciones:

El primero es IF...ENDIF

Esta instrucción comprueba si se cumple una condición antes de pasar a la siguiente instrucción. Si no se cumple ejecuta la instrucción que sigue al ENDIF, por ejemplo

Figura 17. Sentencia de control IF simple

```
IF x > 0.  
    WRITE / 'x es un número positivo'.  
ENDIF.
```

Fuente: <http://es.scribd.com/doc/25289651/SAP-Instrucciones-de-Control-y-Bucles-en-ABAP>.

Consulta: 20 de enero del 2013.

Para poder añadir una opción por defecto se puede utilizar la instrucción ELSE, Por ejemplo:

Figura 18. Sentencia de control IF con ELSE

```
IF x > 0.  
    WRITE / 'x es un número positivo'.  
ELSE  
    WRITE / 'x es un número negativo'.  
ENDIF.
```

Fuente: <http://es.scribd.com/doc/25289651/SAP-Instrucciones-de-Control-y-Bucles-en-ABAP>,

Consulta el 20 de enero del 2013.

También podría darse el caso de que la variable X fuera cero. Si queremos comprobar si se cumple una segunda condición usamos la instrucción ELSEIF, Por ejemplo:

Figura 19. **Sentencia de control IF con ELSEIF y ELSE**

```
IF x GT 0.  
    WRITE / 'x es un número positivo'.  
ELSEIF x EQ 0.  
    WRITE / 'x es igual a cero'.  
ELSE  
    WRITE / 'x es un número negativo'.  
ENDIF.
```

Fuente: <http://es.scribd.com/doc/25289651/SAP-Instrucciones-de-Control-y-Bucles-en-ABAP>,
Consulta: 20 de enero del 2013.

La segunda opción para comprobar si se cumple una condición es mediante el bloque de instrucciones CASE...ENDCASE. Por ejemplo para el caso anterior.

Figura 20. **Sentencia de control CASE WHEN**

```
CASE x.  
    WHEN x GT 0.  
        WRITE / 'x es un número positivo'.  
    WHEN x EQ 0.  
        WRITE / 'x es cero'.  
    WHEN OTHERS.  
        WRITE / 'x es un número negativo'.  
ENDCASE.
```

Fuente: <http://es.scribd.com/doc/25289651/SAP-Instrucciones-de-Control-y-Bucles-en-ABAP>,
Consulta: 20 de enero del 2013.

Se puede comprobar si se cumplen varias condiciones a la vez mediante la expresión AND.

También se puede comprobar si se cumple alguna de las condiciones mediante la expresión OR.

Figura 21. **Sentencia de control CASE WHEN con sentencia IF anidada**

```
CASE x.  
  WHEN x GT 0 AND x LE 100.  
    WRITE / 'x es un número positivo entre uno y cien'.  
  WHEN x EQ 0 OR x LT 0.  
    WRITE / 'x es cero o un número negativo'.  
ENDCASE.
```

Fuente: <http://es.scribd.com/doc/25289651/SAP-Instrucciones-de-Control-y-Bucles-en-ABAP>,
Consulta: 20 de enero del 2013.

1.2.5. Sentencias de bucle

En ABAP existen dos tipos de bucles. Los bucles condicionales y los no condicionales.

Los bucles condicionales se crean con el grupo de instrucciones WHILE...ENDWHILE

Por ejemplo:

Que se pide por pantalla al usuario que escriba un número, el cual se almacenara en la variable x, hasta el ingreso el valor de 0 el bucle finalizara.

Figura 22. **Instrucción de bucle WHILE**

```
WHILE x NE 0.  
    WRITE / 'Por favor escriba otro número'.  
ENDWHILE.
```

Fuente: <http://es.scribd.com/doc/25289651/SAP-Instrucciones-de-Control-y-Bucles-en-ABAP>,
Consulta: 20 de enero del 2013.

En cada iteración el sistema comprueba que se cumple la condición antes de entrar de nuevo en el bucle. Si no se cumple salta hasta la sentencia ENDWHILE y continúa ejecutando el programa a partir de esa instrucción.

Al igual que en las expresiones condicionales, en los bucles se puede comprobar más de una condición mediante las expresiones AND u OR.

Figura 23. **Instrucción de bucle WHILE con sentencia IF anidada**

```
WHILE x <> 0 AND x < 100.  
    WRITE / 'Por favor escriba otro número'.  
ENDWHILE.
```

Fuente: <http://es.scribd.com/doc/25289651/SAP-Instrucciones-de-Control-y-Bucles-en-ABAP>.
Consulta: 20 de enero del 2013.

En este caso se le pide al usuario que escriba otro número siempre que este no sea cero y que sea menor que cien.

También se puede anidar varios bucles o condiciones, por ejemplo:

Figura 24. **Sentencia de bucle WHILE**

```
WHILE x <> 0.  
  PARAMETER x TYPE i.  
  IF x MOD 2 = 0.  
    WRITE / 'Es un número par'.  
  ELSE  
    WRITE / 'Es un número impar'.  
  ENDIF  
ENDWHILE.
```

Fuente: <http://es.scribd.com/doc/25289651/SAP-Instrucciones-de-Control-y-Bucles-en-ABAP>.
Consulta: 20 de enero del 2013.

Los bucles NO condicionales se crean con las instrucciones DO...ENDDO.

En este caso se indica el número de veces que se desea que se ejecute un grupo de instrucciones, por ejemplo:

Figura 25. **Sentencia de bucle DO**

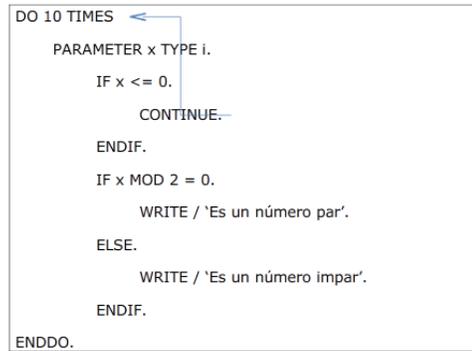
```
DO 10 TIMES  
  WRITE 'Hola amigos'.  
ENDDO.
```

Fuente: <http://es.scribd.com/doc/25289651/SAP-Instrucciones-de-Control-y-Bucles-en-ABAP>.
Consulta: 20 de enero del 2013.

En este caso aparecerá 10 veces la frase Hola amigos.

Se puede terminar una vuelta con la instrucción CONTINUE

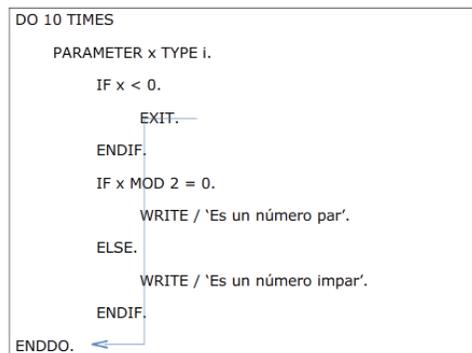
Figura 26. **Utilización de la instrucción CONTINUE**



Fuente: <http://es.scribd.com/doc/25289651/SAP-Instrucciones-de-Control-y-Bucles-en-ABAP>.
Consulta: 20 de enero del 2013.

Por último se puede utilizar la instrucción EXIT para salir del bucle.

Figura 27. **Utilización de la instrucción EXIT**



Fuente: <http://es.scribd.com/doc/25289651/SAP-Instrucciones-de-Control-y-Bucles-en-ABAP>.
Consulta: 20 de enero del 2013.

En este caso si el usuario introduce un número negativo el programa lleva inmediatamente a la instrucción ENDDO y continúa ejecutando la instrucción que viene a continuación.

1.2.6. Subrutinas

Es un bloque de código introducido por las palabras reservadas FORM y ENDFORM. La sintaxis es la siguiente:

```
FORM <nombre de las subrutina> [USING ... [VALUE(<pi>[])] [TYPE <t>|LIKE  
<f>]... ] [CHANGING... [VALUE(<pi>[])] [TYPE <t>|LIKE <f>]... ].  
<Código de la subrutina>  
ENDFORM.
```

La palabra reservada CHANGING sirve para especificar el valor de retorno de la subrutina.

Por ejemplo:

```
FORM escribe_hola.  
    Write:/ Hola'.  
ENDFORM.
```

Y la subrutina se llama con la palabra reservada PERFORM.

```
PERFORM escribe_hola.
```

Ahora se mostrará el efecto de CHANGING cuando los parámetros se pasan por valor. Si está la cláusula CHANGING especificada en la cláusula

PARAMETERS que se pasan por valor, el valor de los parámetros actuales también cambia.

```
Data: d_sum type I, d_num1 type I, d_num2 type I.  
d_num1 = 5.  
d_num2 = 10.  
  
perform pass_val_chcg using d_num1 d_num2.  
write:/ d_sum, d_num1, d_num2.  
  
FORM pass_val_chcg USING VALUE(P_D_NUM1) CHANGING VALUE(P_D_NUM2).  
  
P_D_NUM1 = 1001.  
P_D_NUM2 = 2002.  
  
ENDFORM.
```

La salida del programa es la siguiente:

```
15, 5, 2002
```

Esto es porque se tiene especificada la cláusula CHANGING solo para PARAMETER P_D_NUM2 y por lo tanto el valor actual del parámetro d_num2 cambia.

Si la subrutina concluye exitosamente, cuando ocurre ENDFORM o cuando la subrutina termina a través de una clausula EXIT, el valor actual del parámetro formal es copiado dentro del parámetro actual.

Si la subrutina termina prematuramente con un error de mensaje, el valor no es pasado.

1.2.7. Procesamiento de tablas internas

Las tablas internas son objetos de datos estructurados que permiten el almacenamiento de datos en memoria. Además de las tablas de la base de datos, un programa puede gestionar datos sobre tablas internas en el programa. Los datos almacenados en las tablas internas solo existirán en tiempo de ejecución de un programa. Con una tabla interna se puede, por ejemplo añadir, borrar o seleccionar registros. Dependiendo de los requerimientos del programa el sistema incrementara el tamaño de la tabla interna en tiempo de ejecución.

Un tipo de dato de tabla interna es una definición de una estructura de datos, la cual puede ser utilizada en la declaración de un objeto de dato. Las declaraciones de los tipos de datos se realizaran con la sentencia TYPES, mientras que la declaración de los objetos de datos se realiza con la sentencia DATA.

Para acceder a registros de una tabla interna el sistema proporciona los siguientes métodos:

- Índice de la tabla interna: el índice es un número secuencial de cada registro de la tabla interna. No es un campo de la tabla. Esta creado y gestionado por el sistema. El índice puede ser utilizado para borrar, insertar o modificar registros de la tabla interna. El campo del sistema SY-TABIX contiene el índice del registro.
- Clave definida por sentencia: cuando se leen registros de una tabla interna con la sentencia READ se puede identificar una combinación de campos para marcar el criterio de lectura.

- Clave estándar: este puede ser utilizada en sentencias como COLLECT, READ, SORT y SUM. Si la clave estándar la forman los primeros campos de una tabla interna, las sentencias anteriores serán más eficientes.

Con una tabla interna se puede realizar las siguientes operaciones:

- Insertar registros en tablas internas: para rellenar una tabla interna, o se añaden datos línea a línea, o se copia el contenido de otra tabla. Para introducir registros en una tabla interna, uno a uno se puede utilizar las sentencias APPEND e INSERT.
 - APPEND: esta instrucción añade registros al final de una tabla interna.
 - INSERT: utilizada para insertar nuevos registros antes de una línea existente en la tabla interna.

Por ejemplo:

```
DATA: BEGIN OF TABLA OCCURS 10,
      COLUMNA1 TYPE C,
      COLUMNA2 TYPE I,
      END OF TABLA.
APPEND INITIAL LINE TO TABLA.
DO 3 TIMES.
      TABLA-COLUMNA1 = SY-INDEX.
      TABLA-COLUMNA2 = SY-INDEX ** 2.
APPEND TABLA. ENDDO.
```

El contenido de la tabla interna es el siguiente:

1	1
	4
2	
3	9

La diferencia entre SY-INDEX y SY-TABIX

El SY-INDEX se actualiza en los loops con DO - ENDDO y WHILE - ENDWHILE. En estos LOOP, el campo SY-TABIX no es actualizado.

El SY-TABIX se actualiza con LOOPS en tablas internas: LOOP AT - ENDLOOP. En este LOOP el campo SY-INDEX no es actualizado.

Así que no es cosa de cuál quieras usar, sino en qué tipo de LOOP te encuentres es el campo que debes usar.

Si estás en un LOOP de una tabla interna (LOOP AT), debes usar SY-TABIX.

Si está en otro LOOP (DO o WHILE), debes usar el SY-INDEX.

```
DATA: BEGIN OF TABLA OCCURS 10,  
      COLUMNA1 TYPE C,  
      COLUMNA2 TYPE I,  
END OF TABLA.  
INSERT INITIAL LINE INTO TABLA INDEX 1.  
TABLA-COLUMNA1 = 'A'.  
TABLA COLUMNA2 = 3.  
INSERT TABLA INDEX 2.  
TABLA-COLUMNA1 = 'B'.  
TABLA COLUMNA2 = 5.  
INSERT TABLA INDEX2.
```

EL contenido de la tabla interna es el siguiente:

B	5
A	3

La última sentencia INSERT inserta la línea entre las dos ya existentes.

- Copiar tablas internas: las sentencias que nos permiten realizar esta función son las siguientes.
 - APPEND: también puede ser utilizada para copiar el contenido de tablas internas.
 - INSERT: también puede ser utilizada para copiar el contenido de tablas internas.

Por ejemplo:

```
DATA: BEGIN OF LINEA.  
      COL1 TYPE C,  
      COL2 TYPE I,  
END OF LINEA,  
TABLA1 LIKE LINEA OCCURS 10,  
TABLA2 LIKE LINEA OCCURS 10.  
TABLA-COL1 = 'A'. TABLA-COL2 = 1. APPEND TABLA1.  
TABLA-COL1 = 'B'. TABLA-COL2 = 2. APPEND TABLA1.  
TABLA-COL1 = 'C'. TABLA-COL2 = 3. APPEND TABLA2.  
TABLA-COL1 = 'D'. TABLA-COL2 = 4. APPEND TABLA2.  
  
APPEND LINES OF TABLA 1 FROM 1 TO 1 TO TABLA2.
```

El contenido de la tabla interna TABLA2 será el siguiente:

C	3
D	4
A	1

El mismo código anterior solo que cambiando la línea que está en negrillas por esta línea:

```
INSERT LINES OF TABLA1 INTO TABLA2 INDEX 2.
```

La TABLA1 se inserta antes de la posición 2 de la tabla TABLA2. El contenido de la tabla interna TABLA2 será el siguiente:

C	3
A	1
B	2
D	4

- Lectura de registros de tablas internas: las sentencias que nos permiten realizar esta función son las siguientes.
 - Loop .. endloop: lectura de registros de tablas internas, registro a registro.
 - Read: lectura de un registro de una tabla interna utilizando un índice o una clave.
 - Sum: Suma los campos numéricos de los registros de una tabla interna.

Por ejemplo:

```
DATA: BEGIN OF LINEA,  
      COL1 TYPE I,  
      COL2 TYPE I,  
END OF LINEA.  
TABLA LIKE LINEA OCCURS 10.
```

```

DO 30 TIMES.
    LINEA-COL1 = SY-INDEX.
    LINEA-COL2 = SY-INDEX * SY-INDEX.
APPEND LINEA TO TABLA.
ENDDO.
LOOP AT TABLA FROM 10 TO 25 WHERE COLUMNA2>400.
    WRITE: / SY-TABIX, LINEA-COL2.
ENDLOOP.

```

Salida:

21	441
22	448
23	529
24	576
25	625

En el primer bucle DO se ingresan 30 registros en la tabla, luego en el segundo bucle LOOP realizamos la lectura y el despliegue de la información generada. Solo a partir de la línea 10 hasta la línea 25 son leídas. De las líneas leídas, solo para aquellas en que se cumpla la expresión lógica de la cláusula WHERE, es decir, en aquellas cuyo campo COL2 sea mayor a 400.

Del código anterior sustituimos el segundo loop por el siguiente código:

```

READ TABLE TABLA INDEX 7.
WRITE: SY-SUBRC, SY-TABIX, TABLA-COL2.
READ TABLE TABLA INDEX 31.
WRITE: / SY-SUBRC.

```

Salida:

0	7	49
4		

Con la primera sentencia READ se lee la línea con el índice 7. Como la línea existe, SY-SUBRC es 0. La segunda sentencia READ intenta leer la línea con índice 31. Como no existe, SY-SUBRC es 4.

```
DATA: BEGIN OF LINEA,  
      COL1 TYPE I,  
      COL2 TYPE I,  
END OF LINEA.  
TABLA LIKE LINEA OCCURS 10.  
DO 3 TIMES.  
      LINEA-COL1 = SY-INDEX.  
      LINEA-COL2 = SY-INDEX * *2.  
APPEND LINEA TO TABLA.  
ENDDO.  
LOOP AT TABLA INTO LINEA.  
      WRITE: / LINEA-COL1 LINEA-COL2.  
      SUM.  
      WRITE: / LINEA-COL1 LINEA-COL2.  
ENDLOOP.
```

Salida:

1	1
6	14
2	4
6	14
3	9
6	14

La tabla interna es creada y rellena con tres entradas. En el bucle LOOP. ENDLOOP, el contenido del área de trabajo LINEA es escrito antes y después de la sentencia SUM. Antes de la sentencia SUM, LINEA tiene los valores correspondientes a la línea tratada en el bucle. Después de la sentencia SUM, LINEA tiene la suma de todas las entradas de la tabla interna.

- Modificación de registros de tablas internas: las sentencias que nos permiten realizar esta función son las siguientes:
 - MODIFY: modifica el contenido de un registro de la tabla interna.
 - DELETE: borra registros de una tabla interna.

```

DATA: BEGIN OF LINEA,
      COL1 TYPE I,
      COL2 TYPE I,
END OF LINEA.
TABLA LIKE LINEA OCCURS 10.
DATA: LIN TYPE I,
      OCC TYPE I.
DO 3 TIMES.
      LINEA-COL1 = SY-INDEX.
      LINEA-COL2 = SY-INDEX * *2.
APPEND LINEA TO TABLA.
ENDDO.
LINEA-COL1 = 10.
LINEA-COL2 = 10 ** 2.
MODIFY TABLA FROM LINEA INDEX 2.
LOOP AT TABLA INTO LINEA.
      WRITE: / SY-TABIX, LINEA-COL1, LINEA-COL2.
ENDLOOP.

```

Salida:

1	1	1
2	10	100
3	3	9

Con la sentencia MODIFY se modifica la línea con el índice 2. Por último se ejecuta un bucle de lectura de la tabla interna. En el bloque del bucle se escribe la línea tratada.

```

DATA: BEGIN OF LINEA,
      COL1 TYPE I,
      COL2 TYPE I,
END OF LINEA.
TABLA LIKE LINEA OCCURS 10.
DO 30 TIMES.
    LINEA-COL1 = SY-INDEX.
    LINEA-COL2 = SY-INDEX * *2.
APPEND LINEA TO TABLA.
ENDDO.
LOOP AT TABLA INTO LINEA.
    IF LINEA -COL1 < 28.
        DELETE TABLA.
    ENDIF.
ENDLOOP.
LOOP AT TABLA INTO LINEA.
    WRITE: / SY-TABIX, LINEA-COL1, LINEA-COL2.
ENDLOOP.

```

Salida:

1	28	784
2	29	841
3	30	900

En este ejemplo se crea una tabla interna y se rellena con 30 líneas. En el primer bucle LOOP se borran todas aquellas líneas que tengan un valor inferior a 28 en el componente de línea COL1. En el último bucle loop se escriben las líneas que han quedado en la tabla interna.

```

DATA: BEGIN OF LINEA,
      COL1 TYPE I,
      COL2 TYPE I,
END OF LINEA.

```

```

TABLA LIKE LINEA OCCURS 10.
DO 30 TIMES.
    LINEA-COL1 = SY-INDEX.
    LINEA-COL2 = SY-INDEX * *2.
APPEND LINEA TO TABLA.
ENDDO.
DELETE TABLA INDEX: 2, 3, 4.
WRITE: 'SY-SUBRC', SY-SUBRC.
LOOP AT TABLA INTO LINEA.
    WRITE: / SY-TABIX, LINEA-COL1, LINEA-COL2.
ENDLOOP.

```

Salida:

Sy-subrc		4
1	1	1
2	3	9
3	5	25

En este ejemplo se crea una tabla interna y se rellena con cinco líneas. Se ejecuta a continuación la sentencia DELETE con la cláusula INDEX tres veces. Las dos primeras se completan, ya que existen las líneas con el índice 2 y 3. La última ejecución de la sentencia DELETE no puede ser completada. La tabla no tiene ninguna línea con el índice 4.

```

DATA: BEGIN OF LINEA,
    COL1 TYPE I,
    COL2 TYPE I,
END OF LINEA.
TABLA LIKE LINEA OCCURS 10.
DO 40 TIMES.
    LINEA-COL1 = SY-INDEX.
    LINEA-COL2 = SY-INDEX * *2.
APPEND LINEA TO TABLA.
ENDDO.

```

```

DELETE TABLA FROM 3 TO 38 WHERE COL2 > 20.
WRITE: 'SY-SUBRC', SY-SUBRC.
LOOP AT TABLA INTO LINEA.
    WRITE: / LINEA-COL1, LINEA-COL2.
ENDLOOP.

```

Salida:

1	1
2	4
3	9
4	16
39	1521
40	1600

La primera parte del ejemplo define una tabla interna y carga 40 líneas en ella. Con la sentencia DELETE, el sistema borra las entradas entre la línea 3 y 38, el campo COL2 es mayor a 20.

- Iniciación de tablas internas: las sentencias que nos permiten realizar esta función son las siguientes.
 - REFRESH: inicia el contenido de una tabla interna.
 - CLEAR: limpia la línea de cabecera de la tabla interna.

Por ejemplo:

```

DATA: BEGIN OF TABLA OCCURS 10.
    CAMPO(3) TYPE C,
END OF TABLA.
TABLA-CAMPO = 'ABC'. APPEND TABLA.
TABLA-CAMPO = 'DEF'. APPEND TABLA.
REFRESH TABLA.
WRITE TABLA-CAMPO.
LOOP AT TABLA.
    WRITE: / TABLA-CAMPO.
ENDLOOP.

```

Salida:

DEF

La tabla interna, después de la sentencia REFRESH, no contiene ninguna línea, pero la línea de cabecera de la tabla no cambia, de ahí que permanezca el último valor asignado.

El mismo código anterior solo que cambiando la línea que está en negrillas por esta línea:

Clear tabla.

Salida:

ABC

DEF

La tabla interna, después de la sentencia CLEAR permanece sin cambios, solo se inicializa la línea de cabecera de la tabla interna.

1.3. Diccionario de datos

Es el centro donde se describen y manejan todas las definiciones de datos usadas en el sistema. El Diccionario ABAP soporta la definición de tipos definidos por el usuario (elementos de datos, estructuras y tipos de tablas).

También se puede definir la estructura de objetos de la base de datos (tablas, índices y vistas). Estos objetos pueden ser creados automáticamente en la base de datos con esta definición.

Los objetos más importante en el Diccionario ABAP son tablas, vistas, tipos (elementos de datos, estructuras, tipos de tablas), dominios.

Para la utilización del diccionario de datos en un programa ABAP se puede mencionar que hay 2 categorías de tipos los cuales son:

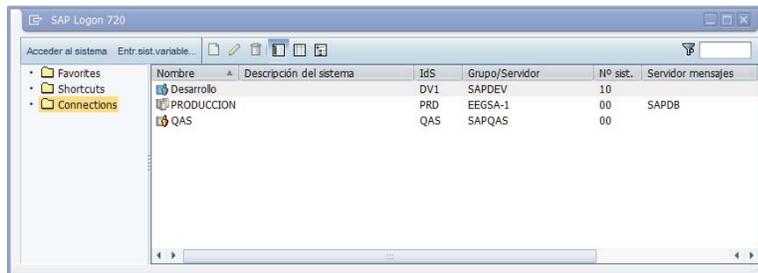
- Local: estos tipos de datos solo pueden ser utilizados en el programa donde fueron declarados. Solo se deberían de utilizar exclusivamente en el programa donde son definidos y su información semántica no tiene relación con los objetos del negocio.
- Global: estos tipos de datos se pueden utilizar en todo el sistema. Aparte de ellos brinda las siguientes ventajas:
 - La utilización de variables globales incrementa la consistencia y su reutilización reduce el mantenimiento requerido en el sistema.
 - Adicionalmente a la información técnica, los tipos de datos globales contienen información correspondiente de las descripciones de los objetos del negocio.

Para acceder al diccionario de datos se utilizan las transacciones SE11 y SE16.

1.3.1. Transacción Se11

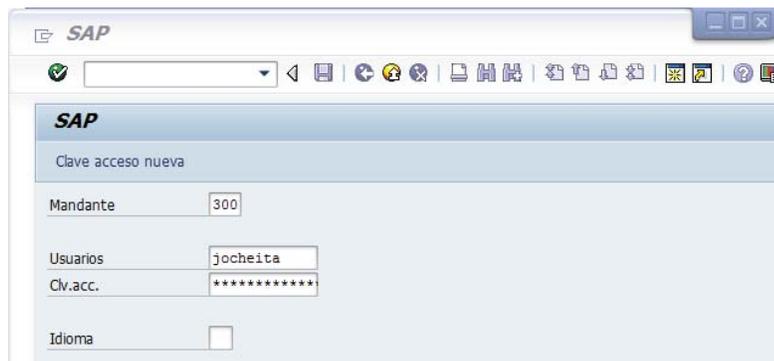
La Transacción Se11 permite la administración (Crear, Actualizar y Eliminar) de todos los objetos (tablas, vistas, tipos de datos, dominios, tipos de grupos, etc.) usados en SAP R/3.

Figura 28. **Ventana de acceso a mandantes para acceder**



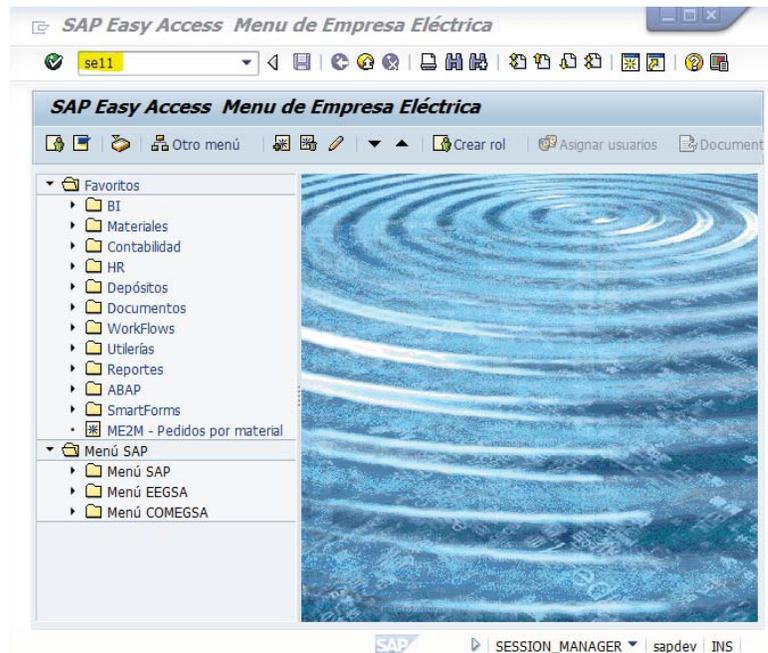
Fuente: elaboración propia, con base en SAP R/3.

Figura 29. **Ingreso de Mandante, Usuario, Contraseña e idioma**



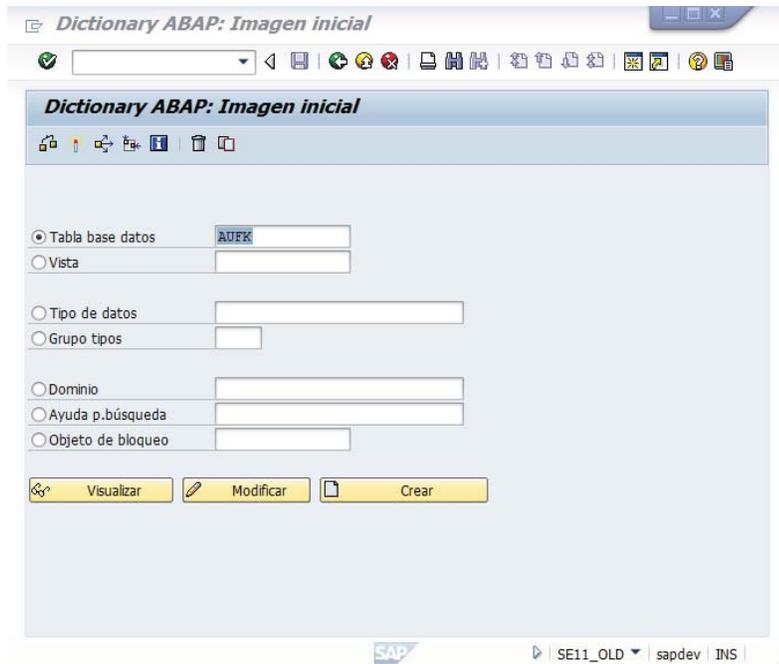
Fuente: elaboración propia, con base en SAP R/3.

Figura 30. Escritorio de SAP e ingreso a la transacción se11



Fuente: elaboración propia, con base en SAP R/3.

Figura 31. **Ventana principal de la transacción se11 e ingreso de la tabla AUFK (Datos maestros de órdenes) para visualización**



Fuente: elaboración propia, con base en SAP R/3.

Figura 32. Visualización de la tabla AUFK (Datos maestros de órdenes)

The screenshot shows the SAP R/3 interface for viewing the structure of table AUFK. The title is "Dict: Visualizar tabla". Below the title, there are navigation icons and a search bar. The table name "AUFK" is entered in the "Tabla transparente" field, and its description "Datos maestros de órdenes" is shown below. The interface has several tabs: "Atributos", "Entrega y actualización", "Campos", "Ayuda p./Verif.entr.", and "Campos de moneda/cantidad". The "Campos" tab is active, displaying a table of fields.

Campo	C.	V...	Elem.datos	Tipo ...	Long.	Dec...	Descripción breve
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Mandante
AUFNR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	AUFNR	CHAR	12	0	Número de orden
AUART	<input type="checkbox"/>	<input type="checkbox"/>	AUART	CHAR	4	0	Clase de orden
AUTYP	<input type="checkbox"/>	<input type="checkbox"/>	AUTYP	NUMC	2	0	Tipo de orden
REFNR	<input type="checkbox"/>	<input type="checkbox"/>	AUFREFNR	CHAR	12	0	Número de orden de referencia
ERNAM	<input type="checkbox"/>	<input type="checkbox"/>	AUFERNAM	CHAR	12	0	Nombre del autor
ERDAT	<input type="checkbox"/>	<input type="checkbox"/>	AUFERFDT	DATS	8	0	Fecha entrada

Fuente: elaboración propia, con base en SAP R/3.

Al presionar la combinación de teclas Ctrl +Shift+ F11, el sistema muestra todas las tablas relacionadas con la AUFK.

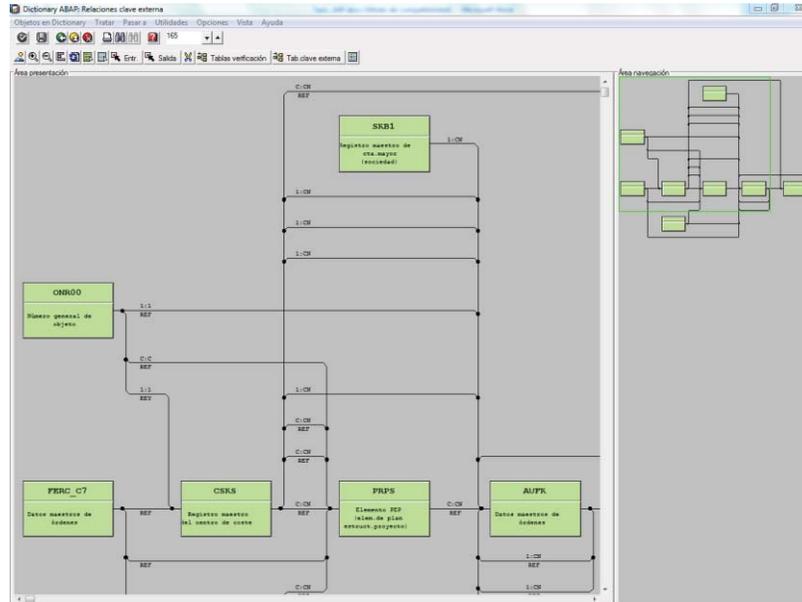
Figura 33. **Tablas relacionadas con la AUFK (Datos maestros de órdenes)**

Tabla	Descripción breve
<input type="checkbox"/> CEPC	Tabla de datos maestros del profit center
<input type="checkbox"/> CSKB	Clases de coste (datos dependientes de la sociedad CO)
<input type="checkbox"/> CSKS	Registro maestro del centro de coste
<input type="checkbox"/> FERC_C7	
<input type="checkbox"/> QNR00	Número general de objeto
<input type="checkbox"/> PRFS	Elemento PEP (elem.de plan estruct.proyecto) datos maes
<input type="checkbox"/> SKB1	Registro maestro de cta.mayor (sociedad)
<input type="checkbox"/> T000	Mandantes
<input type="checkbox"/> T001	Sociedades
<input type="checkbox"/> T001W	Centros/Sucursales
<input type="checkbox"/> T0030	Clases de orden

Fuente: elaboración propia, con base en SAP R/3.

Se puede seleccionar las tablas las tablas deseadas. Luego al dar clic en el botón de Tomar, el sistema muestra el modelo entidad relación.

Figura 34. **Diagrama entidad relación, tablas relacionadas con la tabla AUFK (Datos maestros de órdenes)**

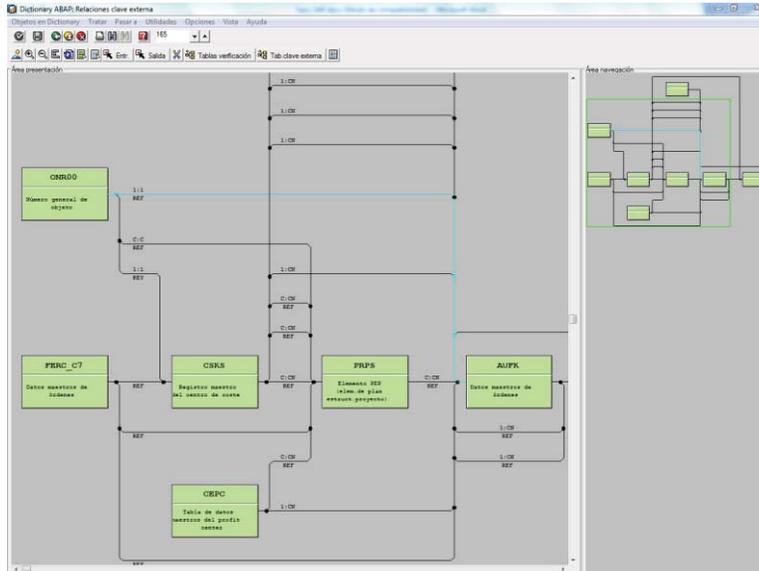


Fuente: elaboración propia, con base en SAP R/3.

Del lado derecho se tiene el visor de búsqueda y del lado izquierdo se tiene una visualización cercana del modelo.

Al darle doble clic a una relación muestra la siguiente pantalla.

Figura 35. Selección de relación entre dos tablas



Fuente: elaboración propia, con base en SAP R/3.

Figura 36. Información de la relación entre las tablas AUFK y ONR00

The screenshot shows the 'Visualizar clave externa AUFK-ONR00' dialog box. It contains the following information:

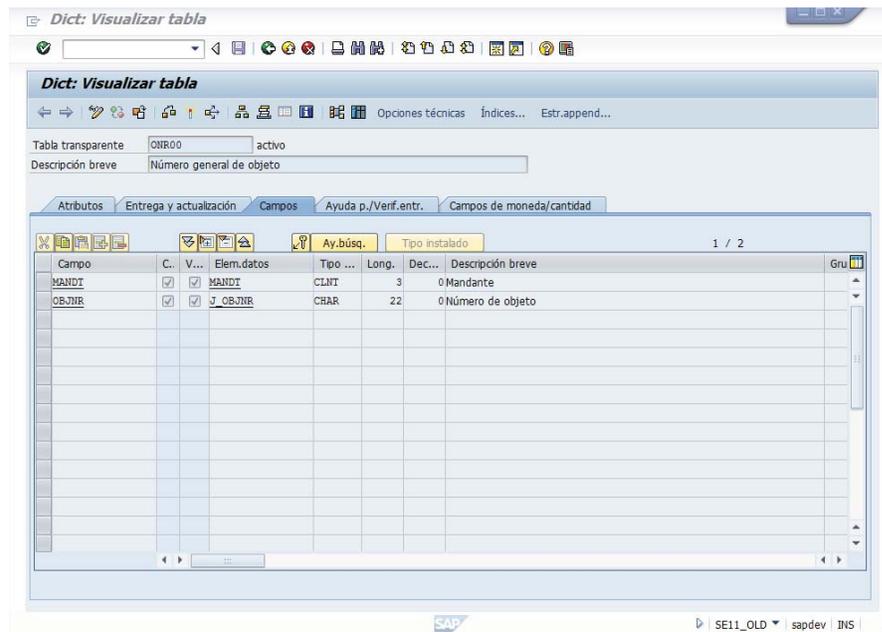
- Descripción breve:** Empty field.
- Tabla verificación:** ONR00
- Campos clave externa:**

Tab.verif.	CpoTabVerif	Tabla cla...	Campo clave ext.	genérico	Constante
ONR00	MANDI	AUFK	MANDI	<input type="checkbox"/>	
ONR00	OBJNR	AUFK	OBJNR	<input type="checkbox"/>	
- Verificación dynpro:**
 - Verificación deseada
 - Mensaje de error: N°Mje AFunc
- Propiedades semánticas:**
 - Clase de campos clave externa:
 - Sin especificar
 - Sin campos/candidatos clave
 - Campos/candidatos de clave
 - Campos clave de una tabla de texto
 - Cardinalidad: 1 : 1

Fuente: elaboración propia, con base en SAP R/3.

Al darle doble clic a un recuadro verde o una tabla muestra la información de la misma.

Figura 37. Información de tabla escogida dentro el diagrama entidad relación



Fuente: elaboración propia, con base en SAP R/3.

1.3.2. Transacción Se16

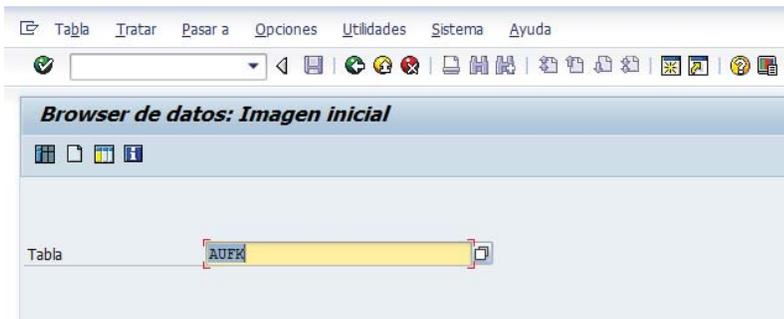
La Transacción SE16 permite realizar consultas a tablas de la base de datos de SAP R/3. Esta transacción no permite realizar consultas avanzadas, solo permite realizar búsquedas en tablas específicas y poder realizar filtros de búsqueda en ellas.

Figura 38. Ingreso de la transacción SE16



Fuente: elaboración propia, con base en SAP R/3.

Figura 39. Pantalla principal de la transacción se16 e ingreso de la tabla AUFK (Datos maestros de órdenes)



Fuente: elaboración propia, con base en SAP R/3.

Figura 40. **Pantalla de selección de datos de la tabla AUFK (Datos maestros de órdenes)**



Fuente: elaboración propia, con base en SAP R/3.

Al dar clic en el botón del reloj o al presionar el botón F8 se muestra la información.

Figura 41. **Información de la tabla AUFK (Datos maestros de órdenes) y selección de un registro**

MANDT	AUFNR	AUART	AUTYP	REFNR	ERNAM	ERDAT	AENAM	AEDAT	KTEXT	LTEXT	BUKRS	WERKS	GSBER	KOKRS	C
300	\$ORDESTAD	44	03	\$ORDINV	PBRVVO	19.07.2000	GBARDON	28.07.2003						1000	1
300	000010000000	E101	01	\$ORDESTAD	RMEJIA	21.07.2000	MBAZARIEGOS	02.01.2003	1000-Terrenos para subestaciones		1000			1000	1
300	000010000001	E102	01	\$ORDESTAD	RMEJIA	21.07.2000	RMEJIA	31.07.2000	3000-Terrenos para subestaciones		3000			1000	1
300	000010000002	E103	01	\$ORDESTAD	RMEJIA	21.07.2000	RMEJIA	31.07.2000	1000-Otros terrenos		1000			1000	1
300	000010000003	E104	01	\$ORDESTAD	RMEJIA	21.07.2000	RMEJIA	31.07.2000	3000-Otros Terrenos		3000			1000	1
300	000010000004	E105	01	\$ORDESTAD	RMEJIA	21.07.2000	WMARROQUIN	02.01.2001	1000-Derechos de via		1000			1000	1
300	000010000005	E106	01	\$ORDESTAD	RMEJIA	21.07.2000	RMEJIA	31.07.2000	3000-Derechos de via		3000			1000	1
300	000010000006	E107	01	\$ORDESTAD	RMEJIA	21.07.2000	JQUIROZ	25.07.2006	1000-Edificios		1000			1000	1
300	000010000007	E108	01	\$ORDESTAD	RMEJIA	21.07.2000	RMEJIA	31.07.2000	2000-Edificios		2000			1000	1
300	000010000008	E109	01	\$ORDESTAD	RMEJIA	21.07.2000	RMEJIA	31.07.2000	4000-Edificios		4000			1000	1
300	000010000009	E110	01	\$ORDESTAD	RMEJIA	21.07.2000	RMEJIA	31.07.2000	5000-Edificios		5000			1000	1

Fuente: elaboración propia, con base en SAP R/3.

Al seleccionar un registro de la tabla se puede visualizar la información respectiva.

Figura 42. **Visualización de la información del registro seleccionado de forma vertical**



The screenshot shows a SAP interface titled 'Tabla AUFK Visual'. Below the title is a sub-header 'Tabla verificación'. The main content is a vertical list of fields and their corresponding values:

MANDT	800
AUFNR	10000000
AUART	EI01
AUTYP	1
REFNR	ØORDESTAD
ERNAM	RMEJIA
ERDAT	21.07.2000
AENAM	MMAZARIEGOS
AEDAT	02.01.2003
KTEXT	1000-Terrenos para subestaciones

Fuente: elaboración propia, con base en SAP R/3.

1.3.3. SQL ABAP

SAP denomina OPEN SQL, que no es más que el conjunto de sentencias propias del lenguaje ABAP que encapsulan las sentencias SQL. SAP ha creado un conjunto de sentencias SQL llamadas sentencias OPEN SQL. OPEN SQL contiene un pequeño subconjunto de las sentencias SQL estándar. Con estas sentencias se puede acceder a cualquier base de datos soportada por SAP sin necesidad de modificar el programa.

A continuación se describen algunas sentencias OPEN SQL, se puede decir que son las más importantes:

Tabla II. Sentencias Open SQL ABAP

Sentencia	Descripción
SELECT	Lectura de datos de las tablas de la base de datos
INSERT	Añade líneas en tablas de la base de datos
UPDATE	Modifica líneas en tablas de la base de datos.
DELETE	Borra líneas en tablas de la base de datos.
COMMIT WORK	Confirmación de modificaciones sobre la base de datos.
ROLLBACK WORK	Deshace las modificaciones sobre la base de datos.

Fuente: elaboración propia.

En los ejemplos se utilizara la tabla KNA1 (datos maestros de cliente a nivel de mandante).

Figura 43. Estructura de la tabla KNA1 (Maestro de clientes)

Campo	C.	V...	Elem.datos	Tipo ...	Long.	Dec...	Descripción breve
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Mandante
KUNNR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	KUNNR	CHAR	10	0	Nº de cliente 1
LAND1	<input type="checkbox"/>	<input type="checkbox"/>	LAND1_GP	CHAR	3	0	Clave de país
NAME1	<input type="checkbox"/>	<input type="checkbox"/>	NAME1_GP	CHAR	35	0	Nombre 1
NAME2	<input type="checkbox"/>	<input type="checkbox"/>	NAME2_GP	CHAR	35	0	Nombre 2
ORT01	<input type="checkbox"/>	<input type="checkbox"/>	ORT01_GP	CHAR	35	0	Población
PSTLZ	<input type="checkbox"/>	<input type="checkbox"/>	PSTLZ	CHAR	10	0	Código postal
REGIO	<input type="checkbox"/>	<input type="checkbox"/>	REGIO	CHAR	3	0	Región (Estado federal, "land", provir

Fuente: elaboración propia, con base en SAP R/3.

1.3.3.1. Sentencia SELECT

A continuación se presenta las diferentes formas de poder utilizar la sentencia SELECT para consultar la base de datos:

```
REPORT Z0001.  
TABLES KNA1.  
SELECT * FROM KNA1 WHERE KUNNR > '00156100'.  
    WRITE: / KNA1-KUNNR, KNA1-NAME1, KNA1-PSTLZ, KNA1-ORT01.  
ENDSELECT.
```

En este ejemplo se ejecuta el bucle de lectura SELECT..ENDSELECT. En el bloque de sentencias se escriben varios campos de la estructura leída.

```
REPORT Z0001.  
TABLES KNA1.  
SELECT * FROM KNA1.  
    SELECT SINGLE * FROM *KNA1 WHERE KUNNR = KNA1-KUNNR.  
    DELETE *KNA1.  
ENDSELECT.
```

Esta combinación de sentencias SELECT borra todos los registros existentes en la tabla KNA1.

```
REPORT Z0001.  
TABLES KNA1.  
DATA TABLA LIKE KNA1 OCCURS 10 WITH HEADER LINE.  
SELECT * FROM KNA1 INTO TABLE TABLA WHERE KUNNR > '00156100'.
```

Ejemplo de clausula INTO TABLE. Todos los registros que cumplan que el código de cliente sea mayor al valor 00156100 se almacenan en la tabla interna tabla.

```

REPORT Z0001.
TABLES KNA1.
DATA: BEGIN OF TABLA OCCURS 10,
      DKUNNR LIKE KNA1-KUNNR,
      HKUNNR LIKE KNA1-KUNNR,
END OF TABLA.
TABLA-DKUNNR = '0001561001'. TABLA-HKUNNR = '0001561100'. APPEND TABLA.
TABLA-DKUNNR = '0001561340'. TABLA-HKUNNR = '0001561400'. APPEND TABLA.
SELECT * FROM KNA1 FOR ALL ENTRIES IN TABLA
WHERE KUNNR >= TABLA-DKUNNR AND KUNNR <= TABLA-HKUNNR.
...
ENDSELECT.

```

Los registros serán aquellos clientes cuyo código este comprendido entre los valores '1561001' y '1561100' o entre los valores '1561340' y '161400'.

En las sentencia SELECT se puede utilizar las cláusulas ORDER BY ASCENDING que orden de forma ascendentemente según el campo especificado, también ORDER BY DESCENDING que lo realiza de forma descendentemente y la cláusula GROUP BY que esta agrupa el conjunto de datos según el campo especificado.

1.3.3.2. Sentencia INSERT

A continuación se presenta las diferentes formas de poder utilizar la sentencia INSERT para ingreso de registros en la base de datos:

```

REPORT Z0001.
TABLES KNA1.
DATA AREA LIKE KNA1.
DATA TABLA LIKE KNA1 OCCURS 10 WITH HEADER LINE.
MOVE: '0001561100' TO AREA-KUNNR, 'MEDIAPARTE' TO AREA-NAME1.

```

INSERT KNA1 FROM AREA.
MOVE: '0001561400' TO KNA1-KUNNR, 'Carat Iberica' TO KNA1-NAME1.
INSERT KNA1.
MOVE: '0001561200' TO TABLA-KUNNR, '28901' TO TABLA-PSTLZ. APPEND TABLA.
MOVE: '0001561300' TO TABLA-KUNNR, '28902' TO TABLA-PSTLZ. APPEND TABLA.
MOVE: '0001561400' TO TABLA-KUNNR, '28903' TO TABLA-PSTLZ. APPEND TABLA.
MOVE: '0001561500' TO TABLA-KUNNR, '28904' TO TABLA-PSTLZ. APPEND TABLA.
INSERT KNA1 FROM TABLE TABLA ACCEPTING DUPLICATE KEYS.

Distintos ejemplos de la sentencia INSERT con las distintas clausulas.

1.3.3.3. Sentencia UPDATE

A continuación se presenta las diferentes formas de poder utilizar la sentencia UPDATE para actualización de registros en la base de datos:

REPORT Z0001.
TABLES KNA1.
DATA AREA LIKE KNA1.
DATA TABLA LIKE KNA1 OCCURS 10 WITH HEADER LINE.
MOVE: '0001561100' TO AREA-KUNNR, '2810' TO AREA-PSTLZ.
UPDATE KNA1 FROM AREA.
MOVE: '0001561400' TO KNA1-KUNNR, '28229' TO KNA1-PSTLZ.
UPDATE KNA1.
UPDATE KNA1 SET SPRAS = 'S' TELTX = ' ' WHERE KUNNR > '0001561100'.
MOVE: '0001561200' TO TABLA-KUNNR, '28901' TO TABLA-PSTLZ. APPEND TABLA.
MOVE: '0001561300' TO TABLA-KUNNR, '28902' TO TABLA-PSTLZ. APPEND TABLA.
MOVE: '0001561400' TO TABLA-KUNNR, '28903' TO TABLA-PSTLZ. APPEND TABLA.
MOVE: '0001561500' TO TABLA-KUNNR, '28904' TO TABLA-PSTLZ. APPEND TABLA.
UPDATE KNA1 FROM TABLE TABLA.

1.3.3.4. Sentencia DELETE

A continuación se presenta las diferentes formas de poder utilizar la sentencia DELETE para eliminar de registros en la base de datos:

```
REPORT Z0001.  
TABLES KNA1.  
DATA AREA LIKE KNA1.  
DATA TABLA LIKE KNA1 OCCURS 10 WITH HEADER LINE.  
MOVE: '0001561100' TO AREA-KUNNR.  
DELETE KNA1 FROM AREA.  
MOVE: '0001561400' TO KNA1-KUNNR.  
DELETE KNA1.  
DELETE FROM KNA1 SET SPRAS = 'D'.  
MOVE: '0001561200' TO TABLA-KUNNR.  
MOVE: '0001561300' TO TABLA-KUNNR.  
MOVE: '0001561400' TO TABLA-KUNNR.  
MOVE: '0001561500' TO TABLA-KUNNR.  
DELETE KNA1 FROM TABLE TABLA.
```

1.3.3.5. Sentencias COMMIT Y ROLLBACK

A continuación se presenta la utilización de las sentencia COMMIT para poder confirmar una operación en la base de datos y la sentencia ROLLBACK para cancelar una operación en la base de datos:

```
REPORT Z00001.  
TABLES KNA1.  
DATA FLAG.  
MOVE: '0001561100' TO KNA1-KUNNR, 'CLIENTE 1' TO KNA1-NAME1. INSERT KNA1.  
IF SY-SUBRC <> 0.  
    MOVE 'X' TO FLAG.  
ENDIF.
```

```
MOVE: '0001561101' TO KNA1-KUNNR, 'CLIENTE 1' TO KNA1-NAME1. INSERT KNA1.  
IF SY-SUBRC <> 0.  
    MOVE 'X' TO FLAG.  
ENDIF.  
IF FLAG = 'X'.  
    ROLLBACK WORK.  
ELSE.  
    COMMIT WORK. ENDIF.
```

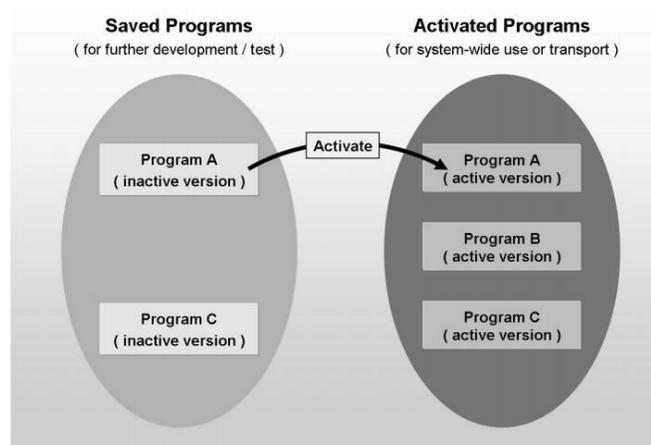
Si las tres operaciones de inserción se realizan satisfactoriamente, se confirma la actualización de la base de datos con la sentencia COMMIT WORK. En caso contrario, se deshacen los cambios con la sentencia ROLLBACK WORK.

1.4. Activación de programas u objetos

El propósito de activar los programas u objetos, es de mantenerlos seguros en el sistema y que no se pueden realizar cambios en ellos hasta que sean activados. Por lo que esto ayuda a que el sistema se mantenga estable y aparte de ello permite transportar programas u objetos a otros ambientes como QAS y PRD.

El proceso que realiza el sistema para la activación de programas u objetos es de guardarlos en un repositorio antes de recibir la orden de ser activados. Como se muestra a continuación:

Figura 44. **Diagrama de objetos de desarrollo en estado inactivo hacia estado activo**



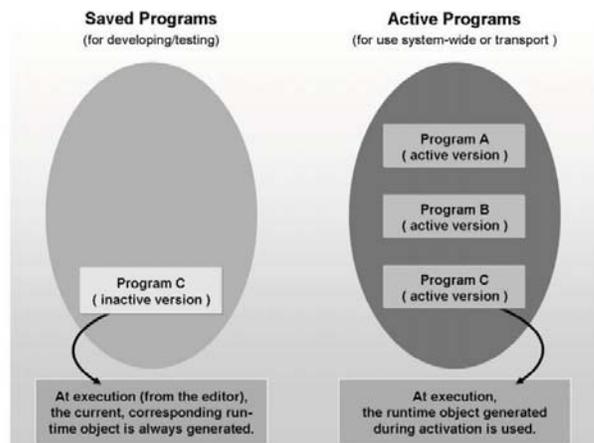
Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 207.

Cuando el usuario activa el programa u objeto, el sistema despliega el listado de todos los objetos inactivos, entonces en el listado se debe de escoger el objeto a activar.

La activación de programas u objetos incluyen las siguientes funcionalidades:

- Guardar el objeto como una versión inactiva
- Verificación de sintaxis y consistencia del programa u objeto de la versión inactiva.
- Sobrescribir previamente la versión activa con la versión inactiva, siempre y cuando esta tenga una sintaxis y consistencia exitosa.
- Generación de objetos de ejecución para ejecuciones del programa u objeto posteriormente.

Figura 45. Diagrama de generación de objetos de ejecución



Fuente: Sap Netweaver 2004. Abap workbench fundamentals, participant handbook. p. 208.

Cuando se genera el objeto de desarrollo, en este caso el sistema crea por separado un objeto de carga en compilación y lo guarda en un repositorio. Esto genera el objeto de ejecución para luego ser ejecutado.

Este proceso permite tener un control de versiones de todos los programas u objetos, por lo que es muy importante realizarlo.

1.5. Cómo crear un programa

Se debe de ingresar la transacción se38.

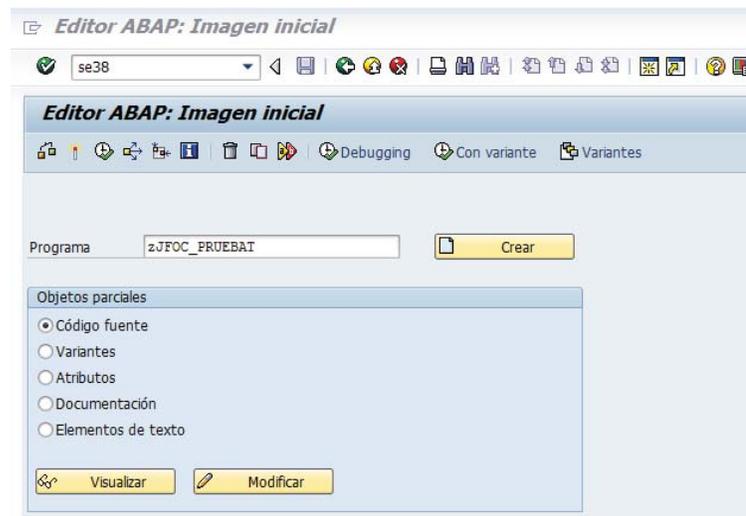
Figura 46. Ingreso a la transacción se38



Fuente: elaboración propia, con base en SAP R/3.

Luego despliega la siguiente ventana:

Figura 47. Ventana principal de la transacción Se38



Fuente: elaboración propia, con base en SAP R/3.

Si ya existe un programa creado solo se ingresa el nombre del programa que se ha creado y se presiona el botón de modificar.

Si no existe proyecto y es uno nuevo, se ingresa el nuevo nombre y se presiona crear.

Figura 48. **Pantalla de creación de un nuevo programa ABAP**

ABAP: Propiedades de programa ZIFOC_PRUEBAT modificar

Titulo prueba para aprender abap

Idioma maestro ES Español

Creado por 06.02.2013 JOCHEITA

Última modif. [] []

Status Nuevo(revisado)

Atributos

Tipo Programa ejecutable

Status Programa de test

Aplicación Aplicación desconocida

Grupo autorizaciones []

Base de datos lógica []

Vers.imagen selección []

Bloqueo de editor Verif.unicode activas

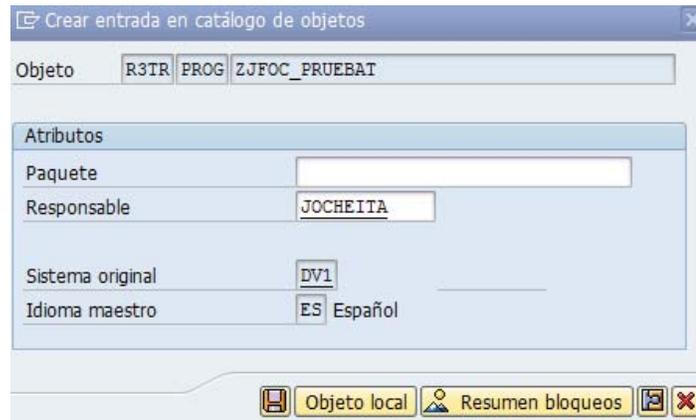
Cálculo de coma fija Iniciar mediante variante

Grabar

Fuente: elaboración propia, con base en SAP R/3.

Luego se da clic en el botón GRABAR.

Figura 49. **Pantalla para seleccionar el paquete al cual pertenecerá el programa o también puede ser objeto local**



Fuente: elaboración propia, con base en SAP R/3.

Al guardar como objeto local, el sistema muestra el editor de código. Por lo que se puede ingresar el siguiente código:

```
REPORT ZJFOC_PRUEBAT.
```

**así se piden los parámetros en la pantalla*

```
PARAMETERS:
```

```
valor TYPE n,
```

```
nombre Type String.
```

**estas son variables adentro del programa*

```
data:
```

```
real TYPE f,
```

```
msg TYPE String.
```

**ESTO NO TIENE CAMBIO DE LINEA*

```
WHILE real < 10.
```

```
  WRITE 'Hola Mundo!' .
```

```
  REAL = REAL + 1.
```

```
ENDWHILE.
```

**aquí van con fin de línea*

```
WRITE: /'FIN DE PROGRAMA 1'.
WRITE: / 'Saludos: ', Nombre.
*este es un "for (for i = 1 to 5)
DO 5 TIMES.
  WRITE: / 'MENSAJE EN CICLO CON CAMBIO DE LINEA: '.
ENDDO.
WRITE: / 'FIBONACCI'.
DATA:
  X TYPE I,
  Y TYPE I,
  Z TYPE I.
X = 1.
Y = 1.
Z = 1.
DO 10 TIMES.
  Z = X + Y.
  X = Y.
  Y = Z.
  WRITE: Z.
ENDDO.
```

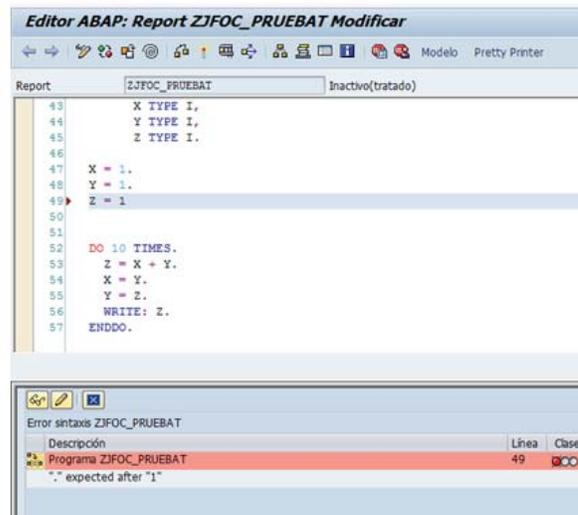
Figura 50. Editor de programas ABAP en SAP

```
1  *~-----
2  *~ Report  ZJFOC_PRUEBAT
3  *~
4  *~-----
5  *~
6  *~
7  *~-----
8
9  REPORT  ZJFOC_PRUEBAT.
10
11
12  *asi se piden los parametros en la pantalla
13  PARAMETERS:
14  valor TYPE n,
15  nombre TYPE String.
16
17
18  *estas son variables adentro del programa
19  data:
20  real TYPE f,
21  msg TYPE String.
22
23
24  *ESTO NO TIENE CAMBIO DE LINEA
25  WHILE real < 10.
26  WRITE 'Hola Mundo!' .
27  REAL = REAL + 1.
28  ENDWHILE.
```

Fuente: elaboración propia, con base en SAP R/3.

Con la combinación de teclas CTRL + F2 se verifica la sintaxis del programa.

Figura 51. **Verificación de Sintaxis utilizando Ctrl + F2**

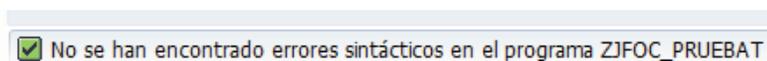


Fuente: elaboración propia, con base en SAP R/3.

Por lo visto hay un error, pues indica que hay que poner punto en la línea 49 para terminar la sentencia.

Al corregir el error y realizar la verificación de sintaxis el sistema muestra el siguiente mensaje en la barra de estado:

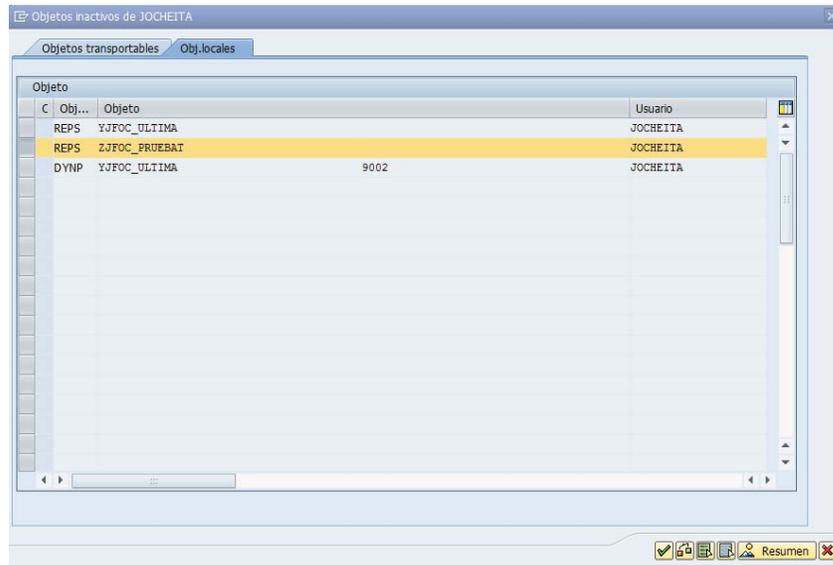
Figura 52. **Barra de estado, donde se despliegan los mensajes de SAP**



Fuente: elaboración propia, con base en SAP R/3.

Al presionar la combinación de teclas CTRL + F3, se activa el programa y el sistema muestra la siguiente ventana:

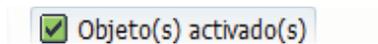
Figura 53. **Ventana de activación de programas utilizando Ctrl + F3**



Fuente: elaboración propia, con base en SAP R/3.

Se debe de seleccionar el programa y luego el sistema muestra el siguiente mensaje en la barra de estado.

Figura 54. **Mensaje en la barra de estado que los objetos han sido activados**



Fuente: elaboración propia, con base en SAP R/3.

Al presionar la tecla F8 → el programa se ejecuta y el sistema muestra la siguiente ventana.

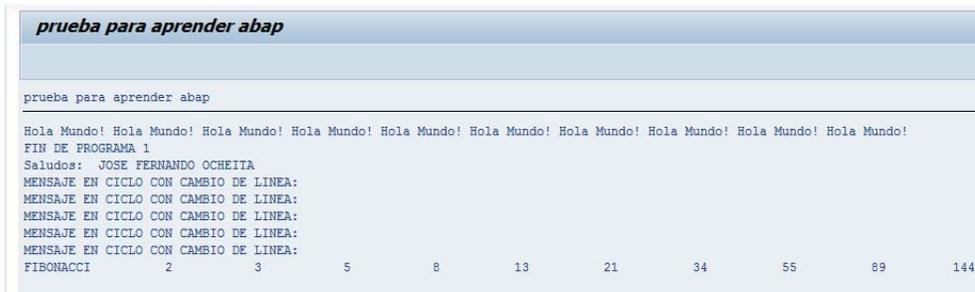
Figura 55. Ejecución del programa prueba e ingreso de parámetros



Fuente: elaboración propia, con base en SAP R/3.

Al presionar la tecla F8 → se ejecuta el programa y el sistema muestra el siguiente resultado.

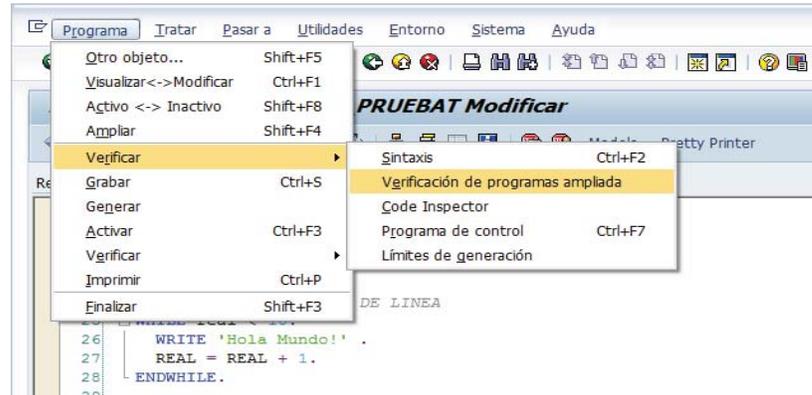
Figura 56. Resultado de la ejecución del programa de prueba



Fuente: elaboración propia, con base en SAP R/3.

Para realizar una verificación ampliada.

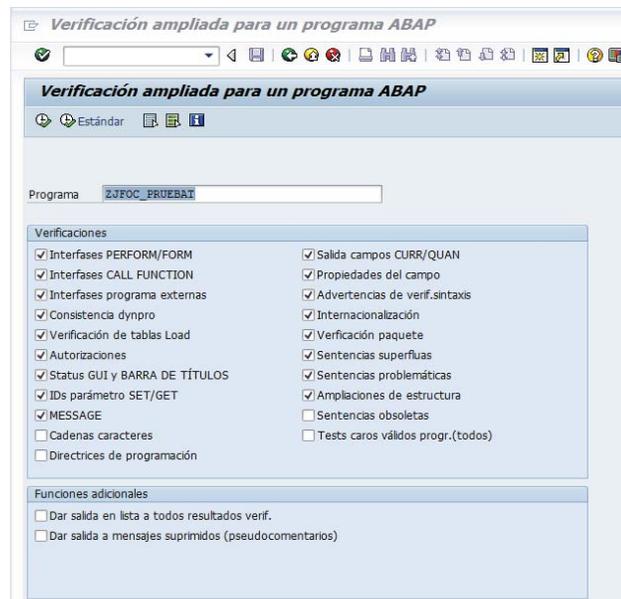
Figura 57. **Menú de acceso a la opción verificación de programas ampliada**



Fuente: elaboración propia, con base en SAP R/3.

El sistema muestra la siguiente ventana

Figura 58. **Ingreso de tipos de verificaciones a realizar**



Fuente: elaboración propia, con base en SAP R/3.

Al dar clic en el botón del reloj, el sistema realiza la verificación mostrando la siguiente ventana.

Figura 59. **Resultado del análisis de verificación de programa ampliada**

Verificación programa ZJFOC_PRUEBAT	Error	Advert.	Mensajes
Entorno de test	0	0	0
Interfases PERFORM/FORM	0	0	0
Interfases CALL FUNCTION	0	0	0
Interfases programa externas	0	0	0
Consistencia dynpro	0	0	0
Autorizaciones	0	0	0
Status GUI y BARRA DE TÍTULOS	0	0	0
IDs parámetro SET/GET	0	0	0
MENSAJE	0	0	0
Salida campos CURR/QUAN	0	0	0
Propiedades del campo	0	1	1
Sentencias superfluas	0	0	0
Advertencias de verif.sintaxis	0	0	0
Verificación del tamaño de carga	0	0	0
Internacionalización	0	0	0
Sentencias problemáticas	0	0	0
Ampliaciones de estructura	0	0	0
Verificación paquete	0	0	0
Errores y advertencias suprimidos	0	0	0

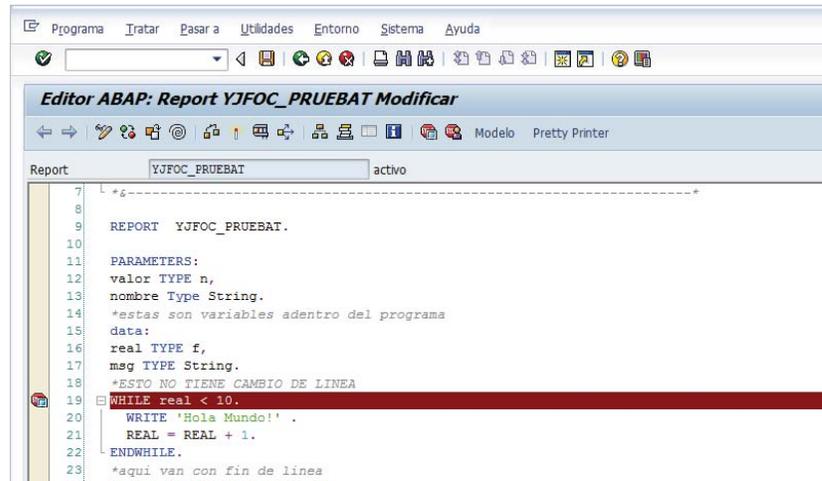
Fuente: elaboración propia, con base en SAP R/3.

Este sirve para verificar errores de manera más exhaustiva.

1.6. Utilización del Debugger

Lo que se debe de hacer es buscar en qué lugar del código se debe de poner el punto de interrupción como por ejemplo, en este caso se colocó en la línea 10 del código:

Figura 60. Ingreso de un break point



Fuente: elaboración propia, con base en SAP R/3.

Al ejecutar el programa presionando la tecla F8 y luego ingresar los parámetros de entrada.

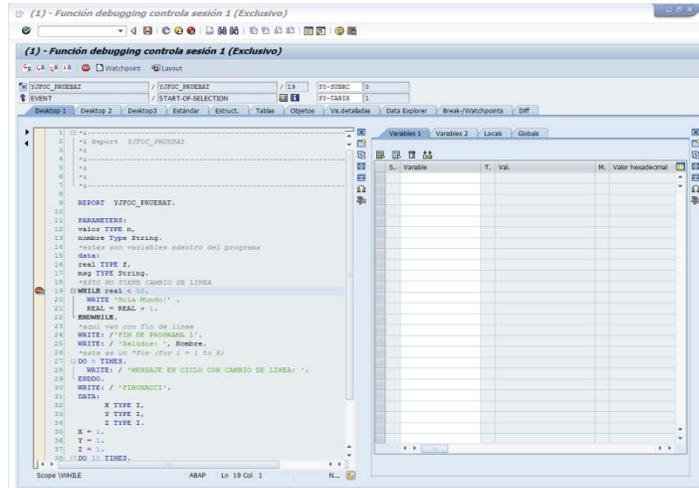
Figura 61. Ejecución de programa para realizar una sesión DEBUG



Fuente: elaboración propia, con base en SAP R/3.

El sistema muestra la siguiente pantalla de verificación paso a paso desde el punto de ruptura ingresado.

Figura 62. Entorno para DEBUG, del lado izquierdo el código ejecutándose y del lado derecho Inspección de variables



Fuente: elaboración propia, con base en SAP R/3.

Del lado izquierdo de la pantalla está la ejecución del código desde el punto de interrupción, del lado derecho se pueden realizar las inspecciones de las variables, en la parte superior se encuentran los valores de las variables del sistema SY-SUBRC, SY-TABIX. También al darle doble clic a cada variable, del lado izquierdo se muestran sus valores como se muestra a continuación:

Figura 63. Inspección de variables

S..	Variable	T.	Val.	M.	Valor hexadecimal
	REAL		0.0000000000000000E+00		0000000000000000
	MSG				
	VALOR		2		32
	NOMBRE		JOSE FERNANDO OCHEITA		4A4F5345204645524

Fuente: elaboración propia, con base en SAP R/3.

Para avanzar en la ejecución del código se puede presionar la tecla F5 la cual realiza la función de avanzar paso a paso, la tecla F7 sirve para ejecutar el código completamente.

A continuación se muestra la salida del programa.

Figura 64. **Resultado del programa luego de ejecutarlo con F7**



Fuente: elaboración propia, con base en SAP R/3.

1.7. Como crear una orden de transporte

Para realizar el transporte del desarrollo del ambiente DEV (desarrollo) a QAS (Quality Assurance), vamos a la transacción se10.

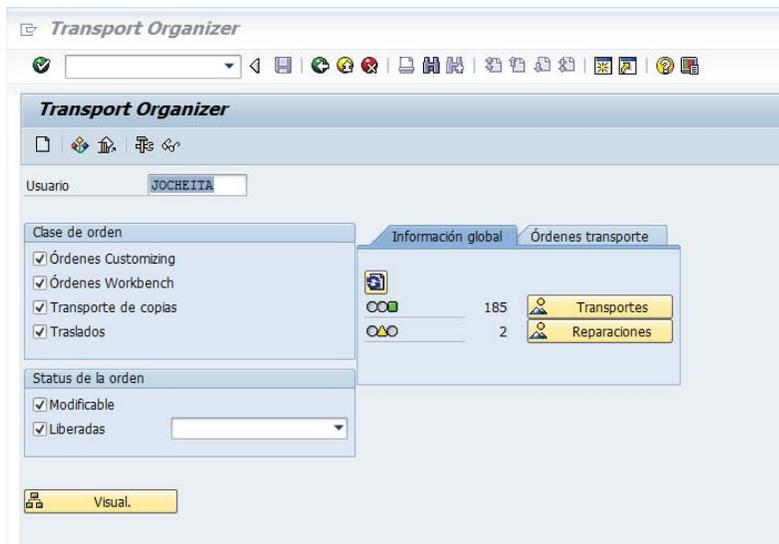
Figura 65. **Ingreso de la transacción Se10**



Fuente: elaboración propia, con base en SAP R/3.

Muestra el organizador de transportes.

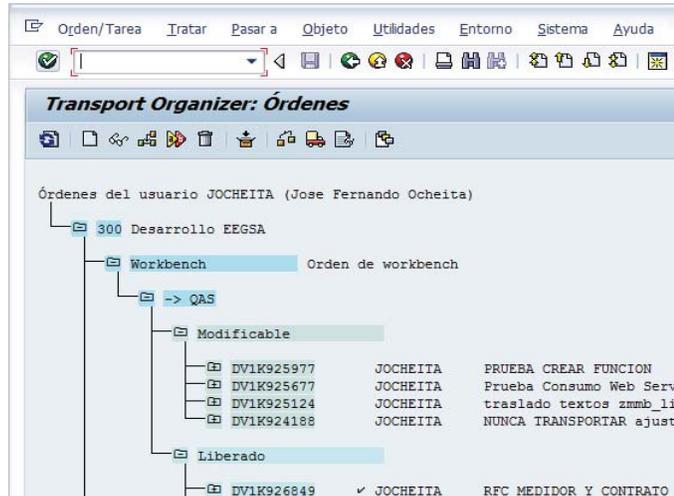
Figura 66. **Pantalla principal de la transacción Se10**



Fuente: elaboración propia, con base en SAP R/3.

Al presionar la tecla ENTER, el sistema muestra el listado de desarrollos que se han realizado y que están disponibles para transportarlos al siguiente ambiente de pruebas (QAS).

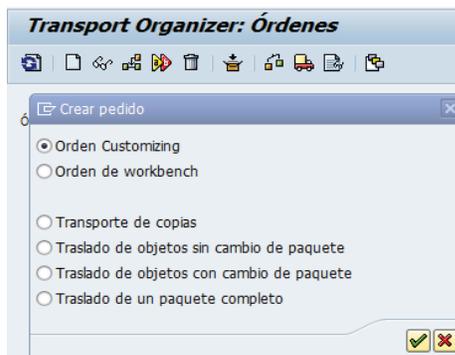
Figura 67. **Listado de órdenes de transporte, procesadas y no procesadas**



Fuente: elaboración propia, con base en SAP R/3.

Para ello hay que crear una nueva orden de transporte, entonces se debe de presionar la tecla F6 y el sistema muestra la siguiente ventana:

Figura 68. **Creación de una nueva orden de transporte, para un programa nuevo**



Fuente: elaboración propia, con base en SAP R/3.

Se selecciona la opción Orden Customizing y luego se da clic en el botón de cheque verde para que luego el sistema muestre la siguiente ventana:

Figura 69. **Ingreso de información de una nueva orden de transporte, para un programa nuevo**

The screenshot shows a SAP window titled 'Crear pedido' (Create Request) for 'Orden Customizing'. The window contains the following fields and data:

Orden		Orden Customizing	
Descripción breve	Creacion de Orden		
Proyecto			
Titular	JOCHEITA	Mandante fuente	300
Status	Nuevo	Destino	QAS
Última modificación	06.02.2013	16:38:08	

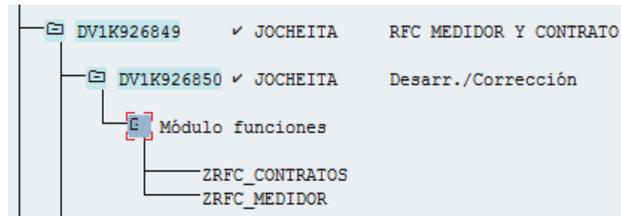
Below the fields, there is a 'Tareas' (Tasks) section with a list box containing 'Empleados' and 'JOCHEITA'. At the bottom right, there are icons for Save, Print, and Close.

Fuente: elaboración propia, con base en SAP R/3.

Se ingresa la información y luego se da clic en el botón de grabar, que en este caso es el disco rojo.

Ya que se creó la orden de transporte, el sistema muestra el listado de órdenes de transporte por lo que se debe de seleccionar la orden creada.

Figura 70. **Actualización del listado de órdenes de transporte**



Fuente: elaboración propia, con base en SAP R/3.

Luego se presiona la tecla F9 para liberar la orden seleccionada como se muestra a continuación:

Figura 71. **Liberación de orden de transporte seleccionada**



Fuente: elaboración propia, con base en SAP R/3.

Al tener la orden de transporte liberada se debe de ir a la transacción STMS.

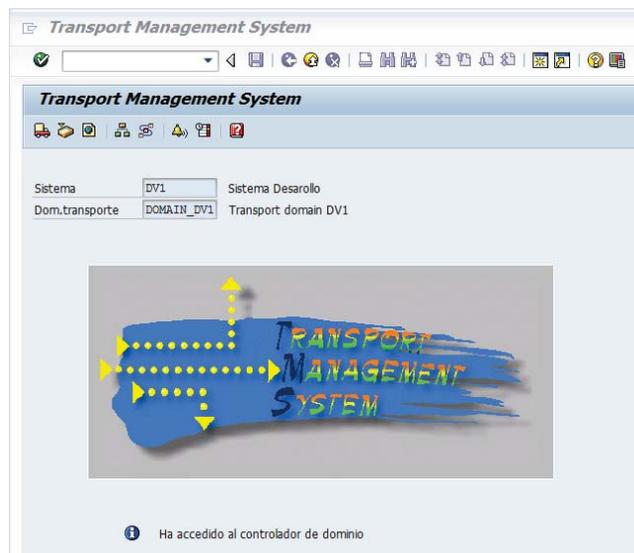
Figura 72. Ingreso de la transacción STMS



Fuente: elaboración propia, con base en SAP R/3.

Por lo que el sistema muestra la siguiente ventana:

Figura 73. Pantalla principal de la transacción STMS



Fuente: elaboración propia, con base en SAP R/3.

Se da clic en el botón del camión rojo y luego aparece la siguiente ventana:

Figura 74. Selección de mandante (DEV, PRD, QAS) en la transacción STMS

Cola	Descripción	Órdenes	Status
DV1	Sistema Desarrollo	0	
PRD	System PRD	1,862	
QAS	Sistema QAS	46	
		1,908	

Fuente: elaboración propia, con base en SAP R/3.

Se selecciona el mandante Sistema PRD con el doble clic, luego el sistema muestra las órdenes de transporte por lo que se debe de escoger la que tiene Estatus No Liberada, eso significa el cuadrado verde.

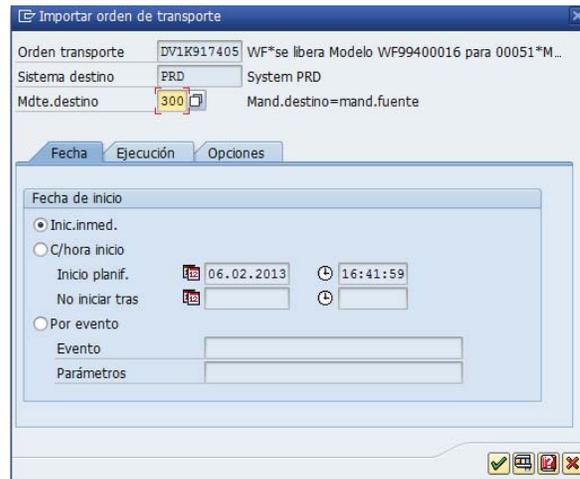
Figura 75. Orden de transporte no importada para el mandante PRD

Número	Orden	Titular	Txt.breve	St
12	[LV]K917405	MACEITUNO	WF*se libera Modelo WF99400016 para 00051*MEAB051108	St

Fuente: elaboración propia, con base en SAP R/3.

Luego se debe de dar clic en el camioncito gris, muestra la siguiente pantalla en la cual se debe de ingresar el mandante y listo. La orden de transporte con el respectivo programa está en el siguiente ambiente.

Figura 76. **Ingreso de Fecha, Hora de la importación de la orden de transporte**



Importar orden de transporte

Orden transporte DV1K917405 WF*se libera Modelo WF99400016 para 00051*M...

Sistema destino PRD System PRD

Mdte.destino 300 Mand.destino=mand.fuente

Fecha Ejecución Opciones

Fecha de inicio

Inic.inmed.

C/hora inicio

Inicio planif. 06.02.2013 16:41:59

No iniciar tras

Por evento

Evento

Parámetros

Fuente: elaboración propia, con base en SAP R/3.

2. IMPORTACIÓN, EXPORTACIÓN Y CONSULTA DE INFORMACIÓN A LOS MÓDULOS DE SAP

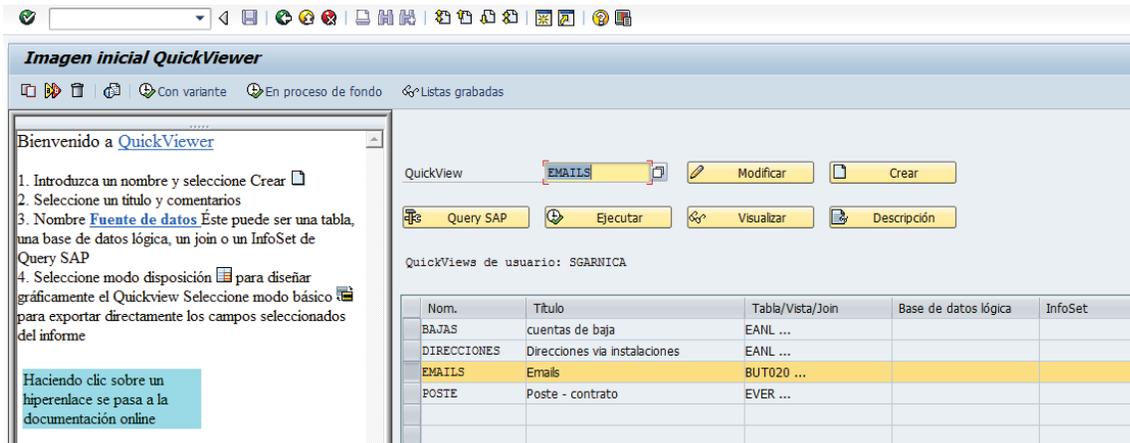
2.1. Transacciones para consultas

A continuación se presenta la utilización de la transacción SQVI para la extracción de información de la base de datos de SAP R/3, se debe de tomar en cuenta que para la utilización de la misma se debe de tener conceptos de lenguaje SQL.

2.1.1. Transacción SQVI

Esta transacción está diseñada para que el usuario pueda cubrir sus propias necesidades de reportes o de información muy puntual creando sus propias consultas otorgando cierta independencia evitando tener que solicitar al equipo de desarrollo de la organización la creación de un reporte; adicionalmente el usuario puede utilizar tablas, vistas, join o unión de tablas, base de datos lógicas.

Figura 77. Pantalla principal de la transacción SQVI



Fuente: elaboración propia, con base en SAP R/3.

Dar en Crear:

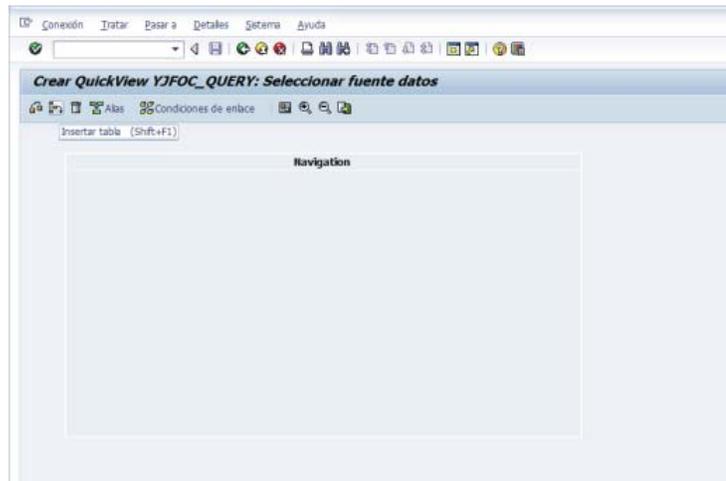
Figura 78. Pantalla de creación de una nueva consulta, para luego guardarla



Fuente: elaboración propia, con base en SAP R/3.

Shift + F1 Insertar tabla

Figura 79. **Área de trabajo para ingreso de tablas para consulta en la transacción SQVI**



Fuente: elaboración propia, con base en SAP R/3.

Ingresar el nombre de la tabla:

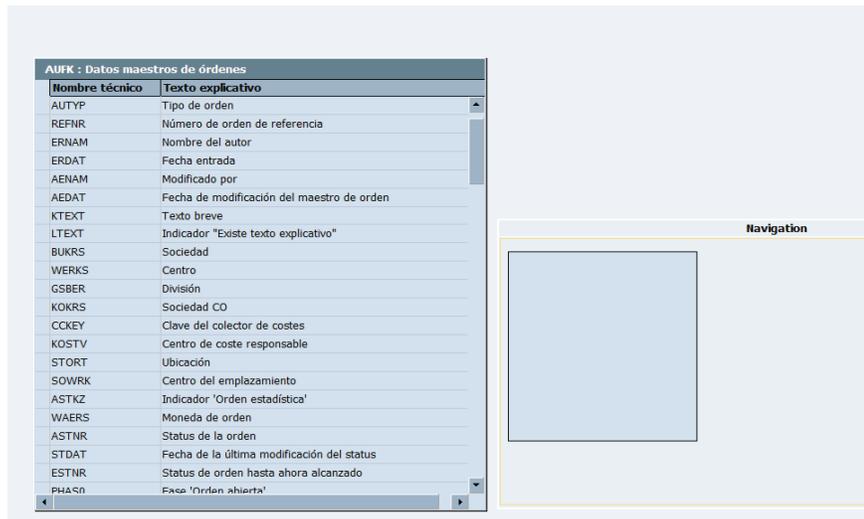
Figura 80. **Ingreso de la tabla a incluir en la consulta**



Fuente: elaboración propia, con base en SAP R/3.

Muestra la tabla Maestro:

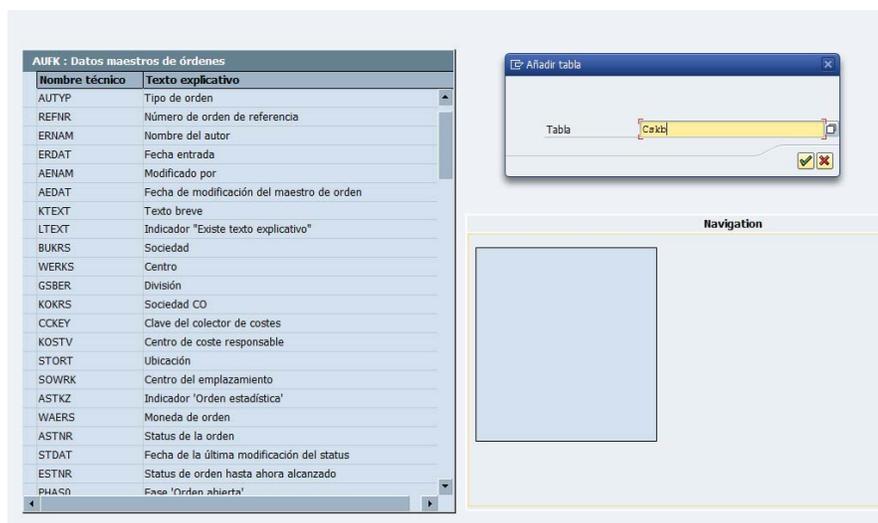
Figura 81. Vista de la tabla en el área de trabajo, en la transacción SQVI



Fuente: elaboración propia, con base en SAP R/3.

Al insertar otra tabla con la combinación de teclas Shift + F1

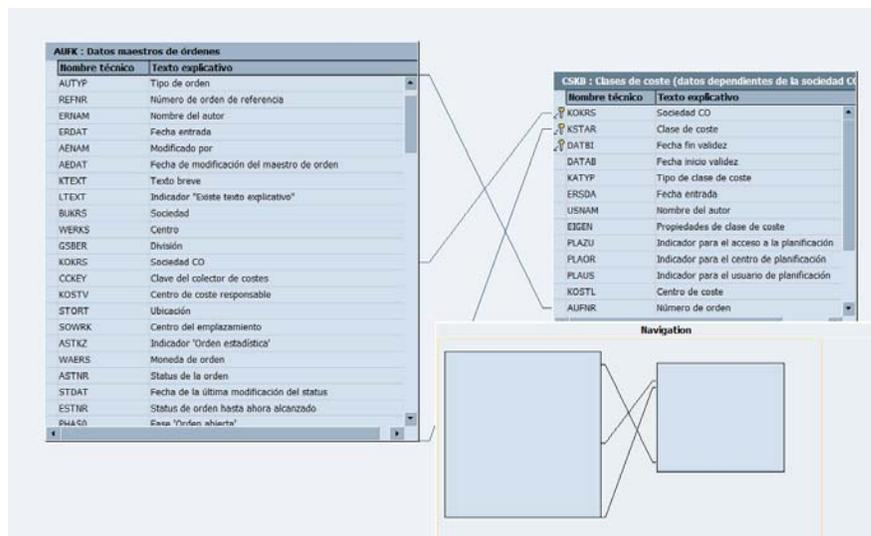
Figura 82. Ingreso de una tabla relacionada



Fuente: elaboración propia, con base en SAP R/3.

Entonces muestra la tabla CSKB y con sus relaciones correspondientes con la tabla AUFK

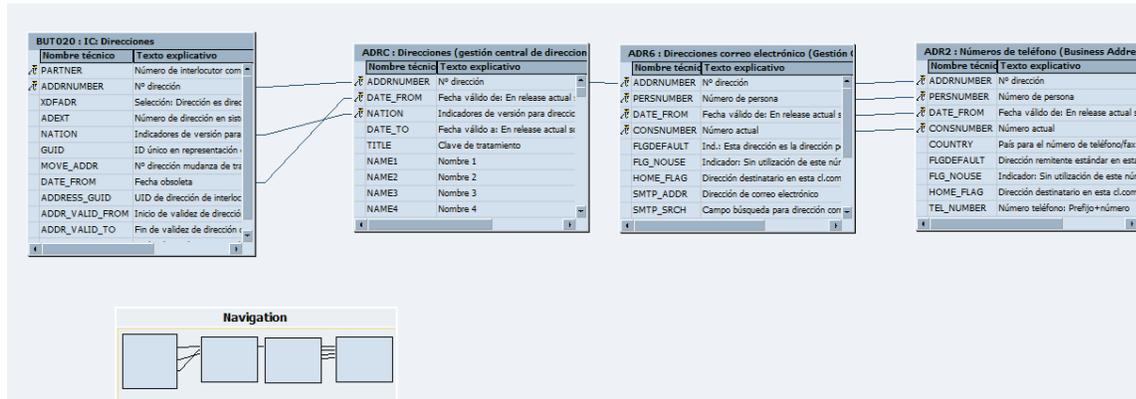
Figura 83. Diagrama entidad relación de tablas ingresadas en la transacción SQVI



Fuente: elaboración propia, con base en SAP R/3.

Se repite el proceso de ingreso de tablas para las tablas BUT020, ADRC, ADR6, ADR2. Y luego el sistema muestra el diagrama entidad relación:

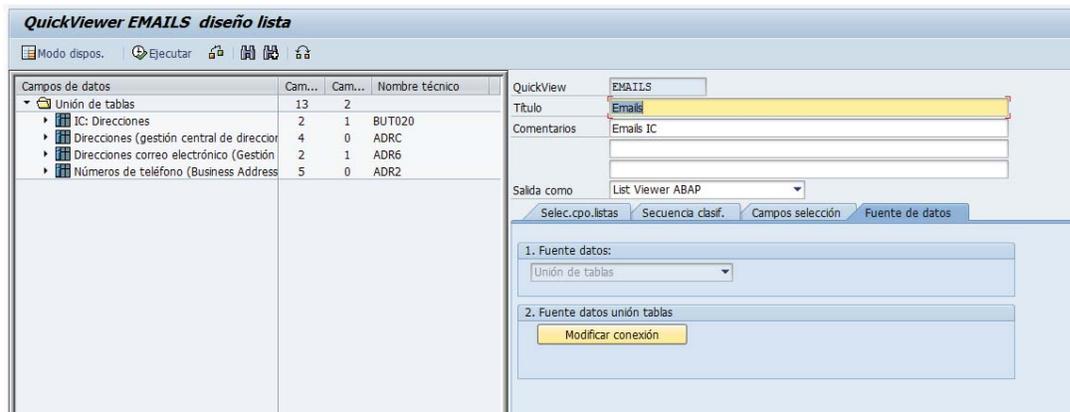
Figura 84. Navegación y vista del diagrama entidad relación de tablas ingresadas



Fuente: elaboración propia, con base en SAP R/3.

Al presionar la tecla F3 (Atrás), el sistema muestra la información de las tablas, como se muestra la siguiente pantalla:

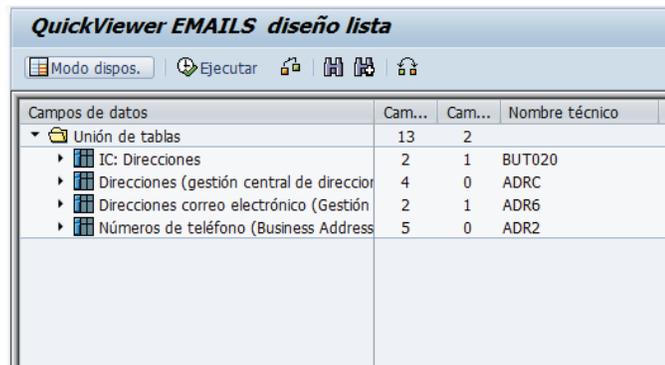
Figura 85. Pantalla de selección de campos de selección y de vista en la consulta a crear



Fuente: elaboración propia, con base en SAP R/3.

El siguiente paso es seleccionar los campos que se desean mostrar en la consulta. Por lo que se debe de dar clic en el listado de tablas para que el sistema muestre las tablas.

Figura 86. Campos a seleccionar



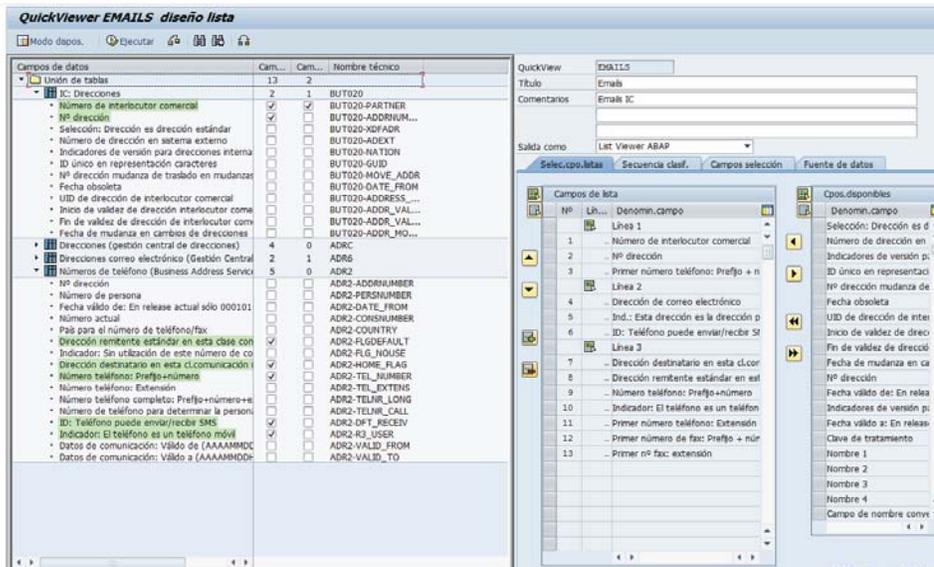
The screenshot shows the SAP QuickViewer interface for 'EMAILS diseño lista'. The window title is 'QuickViewer EMAILS diseño lista'. Below the title bar, there are buttons for 'Modo dispos.', 'Ejecutar', and several icons. The main area displays a table with the following data:

Campos de datos		Cam...	Cam...	Nombre técnico
▼ Unión de tablas		13	2	
▶ IC: Direcciones		2	1	BUT020
▶ Direcciones (gestión central de direcciones)		4	0	ADRC
▶ Direcciones correo electrónico (Gestión)		2	1	ADR6
▶ Números de teléfono (Business Address)		5	0	ADR2

Fuente: elaboración propia, con base en SAP R/3.

Luego se debe seleccionar de cada una de las tablas los campos que se desean que se muestren en la consulta.

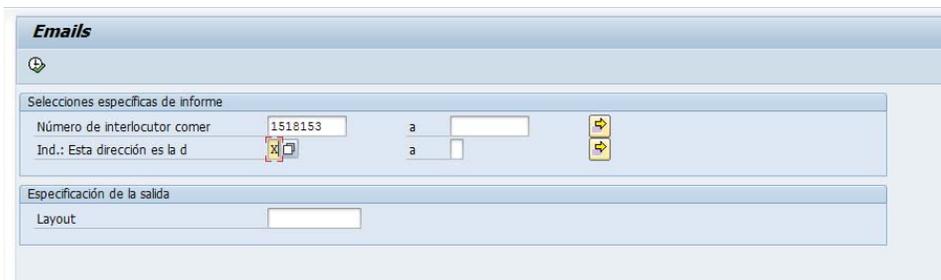
Figura 87. Vista de selección de campos por tabla del lado izquierdo interfaz para escoger campos y del lado derecho los campos seleccionados



Fuente: elaboración propia, con base en SAP R/3.

Luego se da clic en el botón del reloj (Ejecutar) o se puede presionar la tecla F8 por lo que el sistema después de realizar esta operación, muestra los parámetros para poder generar el resultado.

Figura 88. Ingreso de datos para consulta



Fuente: elaboración propia, con base en SAP R/3.

En el parámetro de interlocutor comercial se puede ingresar el siguiente valor 1518153 luego se puede ejecutar con la tecla F8 o dándole clic al botón del reloj.

Figura 89. **Resultado de consulta**

Interl.comercial	Nº direc.	Teléfono	Dirección de correo electrónico	#	SMS act.	PO	Número estándar	Teléfono	Tel.móvil	Extensión	Fax	Extensión
1518182	5370616		eguevara@olmea.com.gt	X	X	X	X	5249-7870	3			
1518197	5370663		alexander_giron@yahoo.com	X	X	X	X	5411-9977	3			

Fuente: elaboración propia, con base en SAP R/3.

Luego para poder realizar la exportación a Excel se debe de ir a modo de exportación y seleccionar Salida Lista como se muestra a continuación:

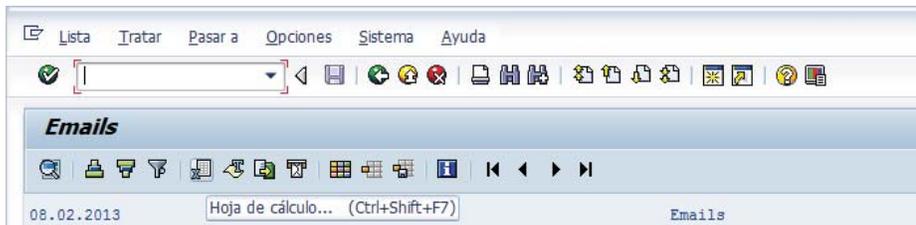
Figura 90. **Selección de resultado en estilo de lista para luego hacer una exportación a Excel**

Interl.comercial	Nº direc.	Teléfono	Dirección de c	S act.	PO	Número estándar	Teléfono	Tel.móvil	Extensión	Fax	Extensión
1518182	5370616		eguevara@olm	X	X	X	5249-7870	3			
1518197	5370663		alexander_giro	X	X	X	5411-9977	3			

Fuente: elaboración propia, con base en SAP R/3.

Luego le se da clic en el botón de Hoja de Cálculo Excel o se puede presionar la combinación de teclas Ctrl + Shift + F7.

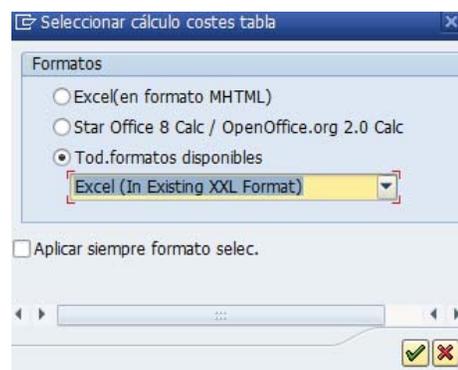
Figura 91. **Exportar en Excel**



Fuente: elaboración propia, con base en SAP R/3.

El sistema muestra una ventana para seleccionar el formato de salida del archivo, luego se selecciona la opción Tod. Formatos disponibles, como se muestra a continuación:

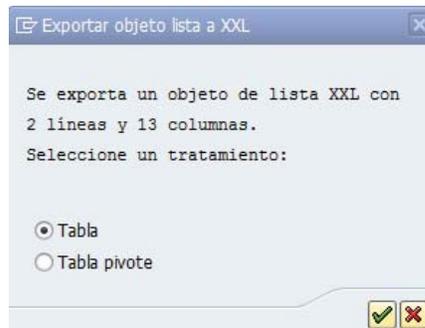
Figura 92. **Selección de formato de salida, del resultado de consulta**



Fuente: elaboración propia, con base en SAP R/3.

Luego se debe de seleccionar que sea de tipo tabla, como se muestra a continuación:

Figura 93. Tipo de salida si en tabla o tabla pivote



Fuente: elaboración propia, con base en SAP R/3.

Y muestra el archivo de Excel, con la información deseada.

Figura 94. Resultado en formato de Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Interl.comercial	N° direc.	Teléfono	Dirección de correo electrónico	# SMS act.	PO	Número estándar	Teléfono	Tel.móvil	Extensión	Fax	Extensi	
2	1518182	5370616	eguevara@olmecca.com.gt		X	X	X	X	5249-7870	3			
3	1518197	5370663	alexander_giron@yahoo.com		X	X	X	X	5411-9977	3			
4													
5													
6													
7													
8													

Fuente: elaboración propia, Excel.

2.2. Procesos batch input

Se puede simular la introducción de datos en una transacción tal y como se haría de forma manual. La diferencia con respecto al proceso de batch input se puede realizar el ingreso de datos de forma repetitiva y automática. Por lo que si se realiza el proceso total de forma manual, esto se convertiría algo muy tedioso.

Típicamente la programación de un batch input se utiliza para la carga masiva de datos en SAP a partir de un archivo plano secuencial. Sin embargo, el resultado de un programa de batch input no es la transferencia de datos directamente a SAP, sino la generación de un juego de datos. En este juego se guarda toda la información de la transacción relevante para la carga (Transacción SHDB):

- Pantallas por las que se pasa
- Datos que se deben poner en cada campo
- Las teclas que se pulsan
- Almacenamiento de datos en memoria

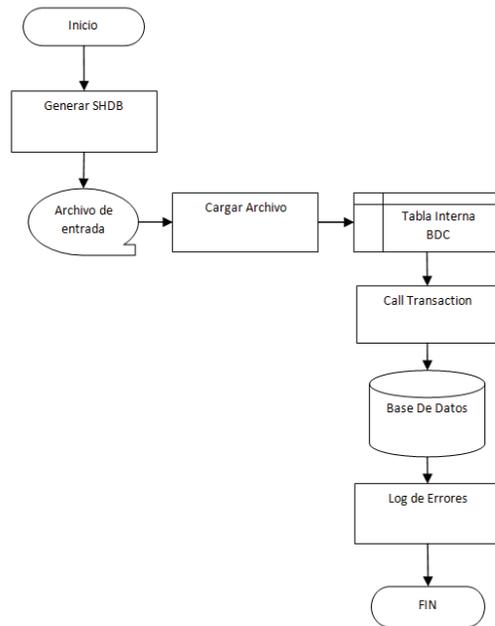
Se realiza una verificación de datos (proceso batch input):

- Carga de datos desde un archivo plano secuencial
- Verificación de formatos
- Verificación automática de rangos de valores
- Valores predeterminados de los campos
- Carga de datos a tablas internas de batch input BDC (Batch Data Communication).

Para procesar los datos inmediatamente y cargarlos inmediatamente se debe de realizar la llamada a la transacción a la cual se está cargando los datos:

- Modo de despliegue
 - Modo A: mostrar todos.
 - Modo E: mostrar solamente los errores.
 - Modo N: no mostrar nada.
- Modo de actualización
 - Modo síncrono: no se continúa con la carga hasta que se haya finalizado la carga anterior.
 - Modo asíncrono: se continúa procesando inmediatamente o de forma paralela.
 - Commit: confirmación de ingreso de datos.
- CALL TRANSACTION
 - CALL TRANSACTION código transacción USING tabla bdc MODE modo de despliegue UPDATE modo de actualización MESSAGES tabla interna de logs
 - Almacenamiento de datos a base de datos de SAP.
- Log de carga
 - Despliegue de errores y confirmaciones de carga en pantalla.

Figura 95. **Diagrama de procesos general de batch input**



Fuente: elaboración propia.

2.2.1. Transacción SHDB

La transacción SHDB es utilizada para grabar una serie de transacciones con sus pantallas. Esta se utiliza para:

- Para transferir la serie de pasos que se utilizan en una transacción a código ABAP.
- Para sesiones de batch input.
- Para probar datos.

Pasos para crear una grabación de transacción:

- A. Ingresar la transacción SHDB
- B. Ingresar el nombre de la grabación
- C. Presionar el botón Crear
- D. Ingresar el código de transacción de la que se desea grabar
- E. Ejecutar la transacción
- F. Escoger obtener transacción si no han ocurrido errores
- G. Escoger siguiente transacción si se quiere grabar otras transacciones y regresar al paso D.
- H. Salvar

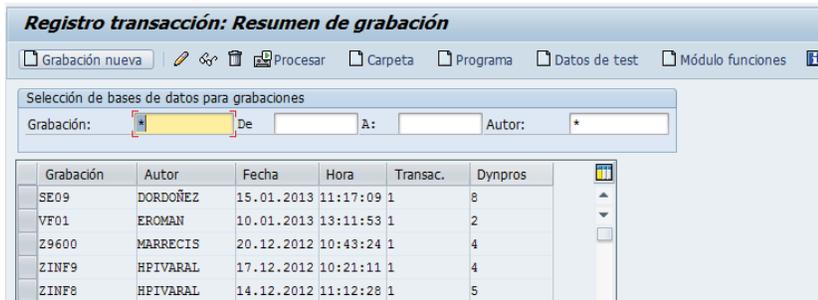
Pasos para procesar la transacción grabada:

- A. Ingresar la transacción SHDB
- B. Clic en la grabación correspondiente
- C. Clic en ejecutar
- D. Ingresar el modo de procesamiento (A, E, N)
- E. Ingresar el modo de actualización
- F. Presionar ENTER

Pasos para generar la sesión de batch input desde la grabación:

- A. Ingresar la transacción SHDB
- B. Clic en la grabación correspondiente
- C. Escoger Generar Sesión
- D. Ingresar el nombre de la sesión
- E. Clic en continuar

Figura 96. **Pantalla principal de la transacción SHDB**



Fuente: elaboración propia, con base en SAP R/3.

Al darle grabación Nueva muestra la siguiente pantalla:

Figura 97. **Cuadro de diálogo de nueva grabación de SHDB**



Fuente: elaboración propia, con base en SAP R/3.

Figura 98. Información del contrato 475837 a modificar



Fuente: elaboración propia, con base en SAP R/3.

Al Iniciar grabación el sistema se dirige a la transacción especificada:

Figura 99. Ingreso de datos en la transacción ES21



Fuente: elaboración propia, con base en SAP R/3.

Figura 100. **Pantalla de datos por contrato en la transacción ES21**

Modificar contrato: 475837

Contrato: 475837 Borrar:

Sector: 01 Energía Eléctrica

Sociedad: 1000 EEGSA

Página 1 Página 2 Página 3

Datos contractuales generales

Txt.p.contrato	ALONSO AGUIRRE,JOSE F.		
Cuenta contrato	511774 ALONSO AGUIRRE,JOSE F.		
ContrSistExist	15976729	FechaAltaSistEx	15.11.1989
Nº personal		Grupo autoriz.	
AutCons/ConsEmp		Gr.estad.contr.	R Cliente de EEGSA Resid..
Status tratam.	03 Contrato Firmado	Facturado	<input type="checkbox"/>
Gr.estruct.reg.			

Fuente: elaboración propia, con base en SAP R/3.

Figura 101. **Modificación de dato, marcado en amarillo**

Contrato: 475837 Borrar:

Sector: 01 Energía Eléctrica

Sociedad: 1000 EEGSA

Página 1 Página 2 Página 3

Datos contractuales generales

Txt.p.contrato	ALONSO AGUIRRE,JOSE F.		
Cuenta contrato	511774 ALONSO AGUIRRE,JOSE F.		
ContrSistExist	15976729	FechaAltaSistEx	15.11.1989
Nº personal		Grupo autoriz.	
AutCons/ConsEmp		Gr.estad.contr.	C Cliente de EEGSA Come...
Status tratam.	03 Contrato Firmado	Facturado	<input type="checkbox"/>
Gr.estruct.reg.			

Fuente: elaboración propia, con base en SAP R/3.

Al presionar la tecla ENTER y luego dar clic en el botón de Guardar. El sistema mostrará el código de secuencia de Dynpros e ingreso de valores en la transacción ES21

Figura 102. Código generado por la transacción SHDB después de
 parar la grabación

Registro transacción: Modificar grabación PRUEBA					
Grabar Procesar					
	Programa	Dyn...	I.	Nom.cpo.	Valor de campo
1			T	ES21	
2	SAPLES20	0100	X		
3				BDC_CURSOR	EVERD-VERTRAG
4				BDC_OKCODE	/00
5				EVERD-VERTRAG	475837
6	SAPLES20	0200	X		
7				BDC_OKCODE	/00
8				BDC_SUBSCR	SAPLATAB 0100TABSTRIP
9				BDC_SUBSCR	SAPLATAB 0200SUBSC
10				BDC_SUBSCR	SAPLES20 0311AREA1
11				BDC_CURSOR	EVERD-STAGRUVER
12				EVERD-VBEZ	ALONSO AGUIRRE,JOSE F.
13				EVERD-VREFER	15976729
14				EVERD-EINZDAT_ALT	15.11.1989
15				EVERD-STAGRUVER	C
16				EVERD-BSTATUS	03
17				BDC_SUBSCR	SAPLES06 1010ACCOUNT
18				BDC_SUBSCR	SAPLATAB 0300AREA2
19				BDC_SUBSCR	SAPLES20 0313AREA3
20				BDC_SUBSCR	SAPLATAB 0300AREA4
21				BDC_SUBSCR	SAPLATAB 0300AREA5
22				BDC_SUBSCR	SAPLATAB 0300AREA6
23				BDC_SUBSCR	SAPLATAB 0300AREA7
24	SAPLES20	0200	X		
25				BDC_OKCODE	=SAVE
26				BDC_SUBSCR	SAPLATAB 0100TABSTRIP
27				BDC_SUBSCR	SAPLATAB 0200SUBSC
28				BDC_SUBSCR	SAPLES20 0311AREA1
29				BDC_CURSOR	EVERD-VBEZ
30				EVERD-VBEZ	ALONSO AGUIRRE,JOSE F.
31				EVERD-VREFER	15976729
32				EVERD-EINZDAT_ALT	15.11.1989
33				EVERD-STAGRUVER	C

Fuente: elaboración propia, con base en SAP R/3.

Este se utilizará en la programación de batch input, ya que es la secuencia que se debe de llevar para el ingreso de datos.

2.2.2. Transacción SM35

Es una transacción estándar de SAP que sirve para monitoreo de batch inputs. Por lo que muestra el listado de batch inputs que tiene el sistema clasificados por nuevos, procesados, erróneos, en tratamiento, en proceso de fondo. Como se muestra a continuación:

Figura 103. Pantalla principal de la transacción SM35



Juego datos	S...	Autor	Fecha	Tiempo	Programa c...	Fe.bloqueo	Autorización	Trans.	Dyn...	IC
ZCENT2	3	HPIVARAL	21.12.2012	10:53:02	ZHR_CENTROS...		HPIVARAL	226 8	218 904	12
ZCENT1	3	HPIVARAL	21.12.2012	10:49:28	ZHR_CENTROS...		HPIVARAL	570 0	2,280	12
CENTR01	3	MARRECIS	20.12.2012	14:09:29	ZHR_CENTROS...		MARRECIS	0 0	0 0	12
INFO92	3	HPIVARAL	17.12.2012	10:46:27	ZHR_CARGAMA...		HPIVARAL	2 1	1 8	12
INFO91	3	HPIVARAL	17.12.2012	10:37:19	ZHR_CARGAMA...		HPIVARAL	1 1	0 4	12
INFO94	3	HPIVARAL	14.12.2012	12:02:53	ZHR_CARGAMA...		HPIVARAL	0 0	0 0	12

Fuente: elaboración propia, con base en SAP R/3.

En la pestaña de nuevo muestra todos los batch input que se han creado recientemente, en erróneos los batch input que no fueron ejecutados con éxito, procesado todos los batch input que se han ejecutado con éxito, en proceso de fondo son los que se están ejecutando en ese mismo momento, en creación son los que se están creando en ese mismo momento.

Al presionar el botón de estadísticas que está en la parte superior muestra una estadística general de todos los batch input que están en el sistema.

Figura 104. Estadística general de batch input

	Transacciones	Dynpros
erróneos	0	0
Procesado	19	38
Borrado	0	0
aún por procesar	0	0
Total	19	38

Fuente: elaboración propia, con base en SAP R/3.

En la parte superior al presionar el botón de LOG muestra mensajes de acuerdo de la ejecución de cada batch input.

Figura 105. Log general del estado de todos los batch input del sistema

Fecha	Tmpo	Juego dat.	Status jueg.dat.	Usuario	ID cola	Serv.aplicac.
21.12.2012	10:53:13	ZCENT2	erróneos	HPIVARAL	12122110530219664659	sapdev
21.12.2012	10:50:33	ZCENT1	En tratamiento	HPIVARAL	12122110492875041902	sapdev
17.12.2012	10:49:08	INFO93	Log sin jueg.dat	HPIVARAL	12121710485315842663	sapdev
17.12.2012	10:46:44	INFO92	erróneos	HPIVARAL	12121710462719664656	sapdev
17.12.2012	10:37:30	INFO91	erróneos	HPIVARAL	12121710371975041901	sapdev

Fuente: elaboración propia, con base en SAP R/3.

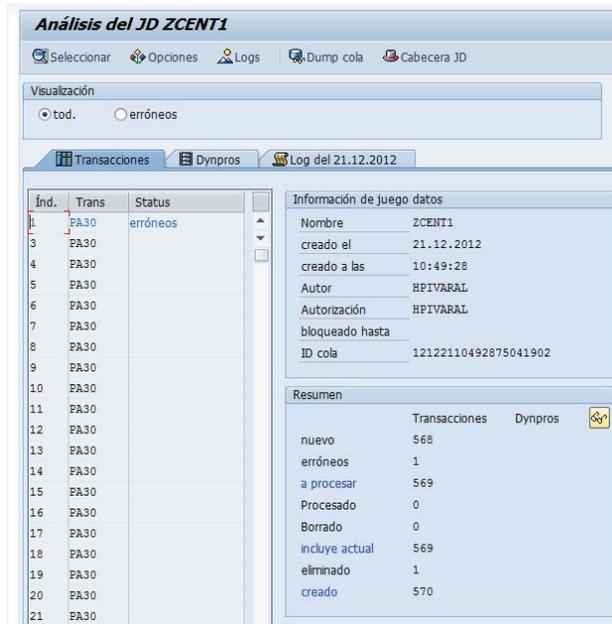
Figura 106. Selección de batch input a monitorear



Fuente: elaboración propia, con base en SAP R/3.

Al darle doble clic sale la información específica del batch input escogido así como la información de creación del mismo, el resumen si funcionó de forma exitosa, etc.

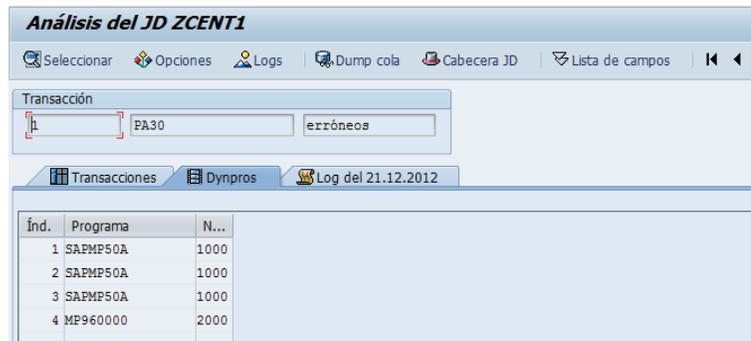
Figura 107. Información de batch input escogido



Fuente: elaboración propia, con base en SAP R/3.

En la pestaña DYNPROS muestra los cuadros de diálogo que se utilizaron para la carga de datos para luego realizar el call transaction.

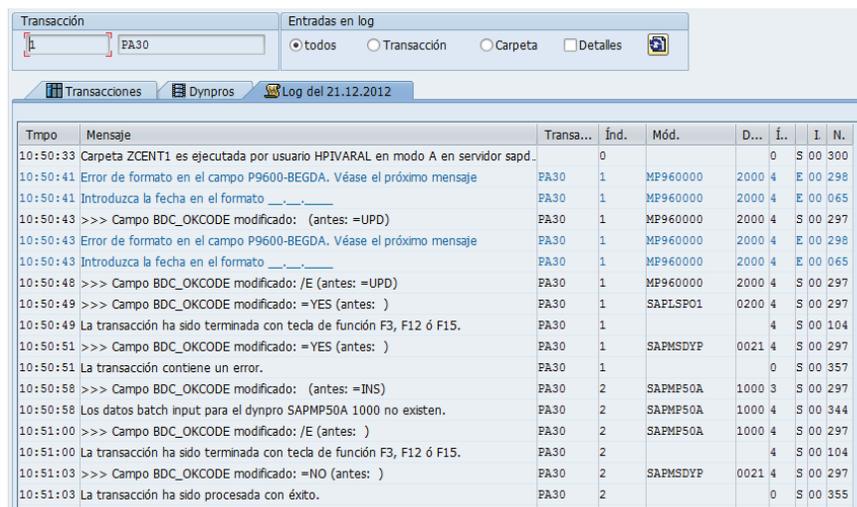
Figura 108. Dynpros ejecutados



Fuente: elaboración propia, con base en SAP R/3.

Y en la pestaña LOG muestra la información de ejecución del batch input si fue exitoso o hubo algunos errores.

Figura 109. Log de salida luego de la ejecución del batch input

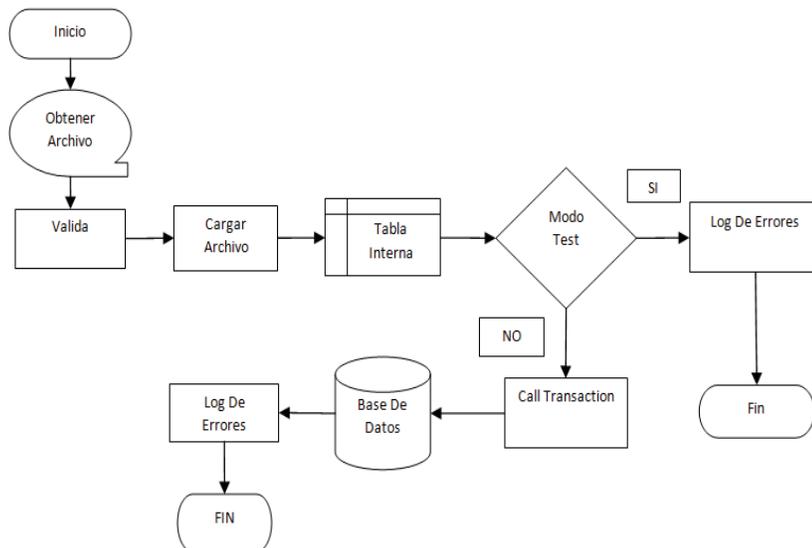


Fuente: elaboración propia, con base en SAP R/3.

2.2.3. Ejemplo de creación de batch input

Se desea realizar la carga de datos para la creación de unidades funcionales con su definición y características utilizando la Transacción IL01. Para ello se tiene que realizar en Modo Test la verificación de la existencia del campo TPLNR de la tabla IFLOT para poder realizar la carga.

Figura 110. Diagrama de procesos del programa



Fuente: elaboración propia.

Figura 111. Grabación SHDB de la transacción IL01

	Programa	Dynpro	In...	Nom.cpo.	Valor de campo
1			T	IL01	BS AA XX F
2	SAPMI00	1110	X		
3				BDC_CURSOR	IFLOS-STRNO
4				BDC_OKCODE	/00
5				IFLOS-STRNO	CENTGT1-CG169D-CG1692-P600496
6				RIL00-TPLKZ	LINTR
7				IFLO-FLTYP	Z
8	SAPMI00	2100	X		
9				BDC_OKCODE	=T03
10				IFLO-PLTXT	POSTE DE CONCRETO SECCIONADO 27MTS C3...
11				BDC_SUBSCR	SAPLIT00 0102SUB_DATA
12				BDC_SUBSCR	SAPLIT00 1020SUB_0102A
13				BDC_SUBSCR	SAPLIT00 1025SUB_1020A
14				ITOB-EQART	PCONCRETO
15				BDC_SUBSCR	SAPLIT00 1021SUB_0102B
16				BDC_SUBSCR	SAPLIT00 1000SUB_1021A
17				BDC_SUBSCR	SAPLIT00 1022SUB_0102C
18				BDC_CURSOR	ITOB-SERGE
19				ITOB-HERST	D/R, Post Type & SUSP, 79.84
20				ITOB-SERGE	D/R, TIR 5/16 , 79.84°
21				BDC_SUBSCR	SAPLIT00 1000SUB_0102D
22	SAPMI00	2100	X		
23				BDC_CURSOR	IFLO-PLTXT
24				BDC_OKCODE	=T04
25				IFLO-PLTXT	POSTE DE CONCRETO SECCIONADO 27MTS C3...
26				BDC_SUBSCR	SAPLIT00 0102SUB_DATA
27				BDC_SUBSCR	SAPLIT00 1052SUB_0102A
28				ITOB-GSBER	02
29				ITOB-KOSTL	30107
30				BDC_SUBSCR	SAPLIT00 1062SUB_0102B
31				ITOB-IWERK	3100
32				BDC_SUBSCR	SAPLIT00 1000SUB_0102C
33				BDC_SUBSCR	SAPLIT00 1000SUB_0102D

Fuente: elaboración propia, con base en SAP R/3.

Figura 112. Tabla de verificación de existencia de información para el batch input

Dict: Visualizar tabla

Tabla transparente: IFLOT tivo

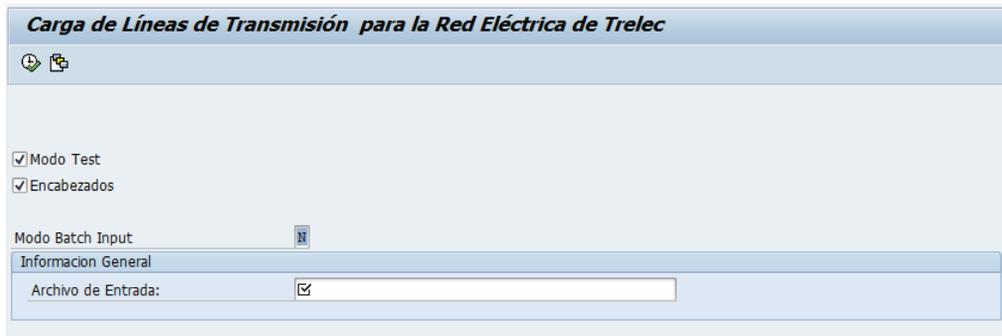
Descripción breve: Ubicación técnica (tabla)

Atributos Entrega y actualización Campos Ayuda p./Verif.entr. Campos de moneda/cantidad

Campo	C.	V...	Elem.datos	Tipo ...	Long.	Dec...	Descripción breve
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Mandante
TPLNR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	TPLNR	CHAR	30	0	Ubicación técnica
.INCLUDE	<input type="checkbox"/>	<input type="checkbox"/>	IFLOT_INC	STRU	0	0	Sección datos ubicación técnica
MLANG	<input type="checkbox"/>	<input type="checkbox"/>	MLANG	LANG	1	0	Idioma principal para textos del objeto
TPLKZ	<input type="checkbox"/>	<input type="checkbox"/>	TPLKZ	CHAR	5	0	Indicador de estructura de ubicación técnica
FLTYP	<input type="checkbox"/>	<input type="checkbox"/>	FLTYP	CHAR	1	0	Tipo de ubicación técnica
TPLMA	<input type="checkbox"/>	<input type="checkbox"/>	TPLMA	CHAR	30	0	Ubicación técnica superior

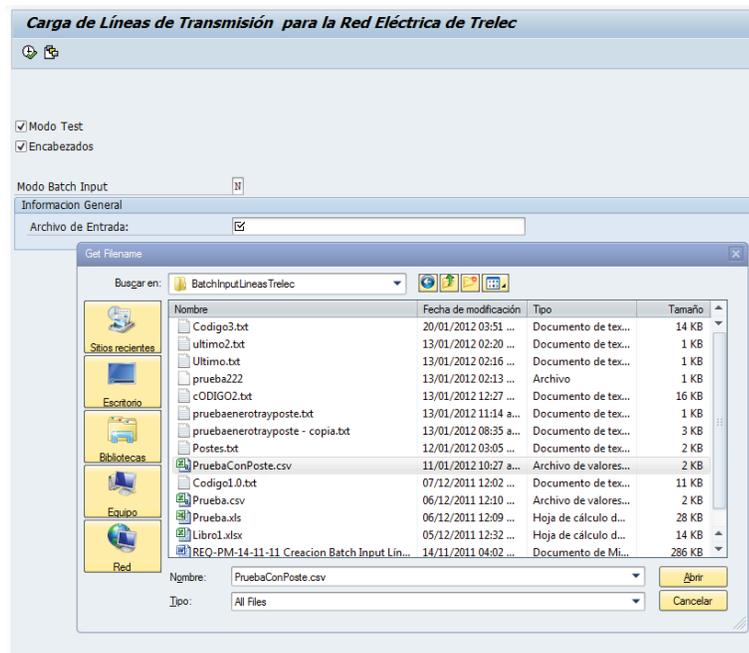
Fuente: elaboración propia, con base en SAP R/3.

Figura 113. Ejecución del programa



Fuente: elaboración propia, con base en SAP R/3.

Figura 114. Escoger el archivo a cargar



Fuente: elaboración propia, con base en SAP R/3.

Figura 115. **Formato del archivo de entrada a cargar a la base de datos**

	A	B	C	D	E	F
1	LINEA 5-ESC691-LIR691-P602908	Poste de concreto seccionado 21mts C3000	PCONCRETO	TANG, Post Type, 0.38°	TANG, OPGW, 0.38°	30-0612
2	LINEA 5-ESC691-LIR691-P602909	Poste de concreto seccionado 21mts C3000	PCONCRETO	TANG, Post Type, 0.90°	TANG, OPGW, 0.90°	30-0612
3	LINEA 5-ESC691-LIR691-P602907	Poste de concreto seccionado 21mts C3000	PCONCRETO	TANG, Post Type, 0.62°	TANG, OPGW, 0.62°	30-0612
4	LINEA 5-ESC691-LIR691-P602906	Poste de concreto seccionado 21mts C3000	PCONCRETO	TANG, Post Type, 0.61°	TANG, OPGW, 0.61°	30-0612
5	LINEA 5-ESC691-LIR691-P605726	Poste de concreto seccionado 21mts C3000	PCONCRETO	TANG, Post Type, 0.82°	TANG, OPGW, 0.82°	30-0612
6	LINEA 5-ESC691-LIR691-P602905	Poste de concreto seccionado 21mts C3000	PCONCRETO	TANG, Post Type, 0.06°	TANG, OPGW, 0.06°	30-0612

Fuente: elaboración propia, con base en SAP R/3.

Ejecución en modo de procesamiento:

Figura 116. **Salida del Log de errores I**

```

VERIFICACION DE ARCHIVO

Datos Correctos
Ubicacion Tecnica: LINEA05-ESC691-LIR691-P602902  Descripcion: Ingresado Exitosamente
Ubicacion Tecnica: LINEA05-ESC691-LIR691-P605727  Descripcion: Ingresado Exitosamente
Ubicacion Tecnica: LINEA05-ESC691-LIR691-P602901  Descripcion: Ingresado Exitosamente
Ubicacion Tecnica: LINEA05-ESC691-LIR691-P602900  Descripcion: Ingresado Exitosamente

Datos Incorrectos
    
```

Fuente: elaboración propia, con base en SAP R/3.

Ejecución en Modo Test

Cuando ya están registrados se ingresa un archivo con datos ya existentes.

Figura 117. Salida del Log de errores II

```
VERIFICACION DE ARCHIVO

Datos Correctos

Datos Incorrectos

Ubicacion Tecnica: LINEA05-ESC691-LIR691-P602902  Descripcion: El Poste y Linea De Transmision Ya Existen!
Ubicacion Tecnica: LINEA05-ESC691-LIR691-P605727  Descripcion: El Poste y Linea De Transmision Ya Existen!
Ubicacion Tecnica: LINEA05-ESC691-LIR691-P602901  Descripcion: El Poste y Linea De Transmision Ya Existen!
Ubicacion Tecnica: LINEA05-ESC691-LIR691-P602900  Descripcion: El Poste y Linea De Transmision Ya Existen!
```

Fuente: elaboración propia, con base en SAP R/3.

2.2.4. Código del programa

A continuación se presenta el código del programa del batch input que se mostró en la sección anterior. Con el objetivo de tener un mejor entendimiento y análisis de su funcionamiento.

Figura 118. Declaración de tablas internas

```
INCLUDE z_batch.

DATA: BEGIN OF ttab,
      rec(1000) TYPE c,
      END OF ttab.

TYPES: BEGIN OF tdat,
        fld1      TYPE string,
        fld2      TYPE string,
        fld3      TYPE string,
        fld4      TYPE string,
        fld5      TYPE string,
        fld6      TYPE string,
        error_p   TYPE i,
        error_des TYPE string,
      END OF tdat.

DATA: itab      LIKE TABLE OF ttab,
      tablita   TYPE TABLE OF tdat,
      idat      LIKE LINE OF tablita,
      lv_lineatxt TYPE c LENGTH 150.
      lv_lineatxt TYPE string.
```

Fuente: elaboración propia, con base en SAP R/3.

Figura 119. **Inicio del Programa, llamando a las subrutinas
Obtener_Archivo, Modo_Test, Procesa,
End_Selection**

```
START-OF-SELECTION.  
    PERFORM obtener_archivo.  
    IF cb_test = 'X'.  
        PERFORM modo_test.  
    ELSE.  
        PERFORM procesa.  
    ENDIF.  
END-OF-SELECTION.  
    PERFORM end_selection.
```

Fuente: elaboración propia, con base en SAP R/3.

Figura 120. **Subrutina Obtener_Archivo**

```
FORM obtener_archivo.  
    DATA: file_str TYPE string.  
    file_str = p_file.  
  
    CALL FUNCTION 'GUI_UPLOAD'  
    EXPORTING  
        filename = file_str  
    TABLES  
        data_tab = itab.  
ENDFORM.
```

Fuente: elaboración propia, con base en SAP R/3.

La cual hace el llamado de la función estándar de SAP GUI_UPLOAD, la cual lee el archivo de entrada y lo carga a una tabla interna ITAB.

Figura 121. La subrutina Modo_Test

```
FORM validar.

"REFRESH: tablilla, tablita.
REFRESH: tablita.
CLEAR: ttab.

DATA: gv_tab(1) TYPE c VALUE c1_abap_char_utilities=>horizontal_tab,
      inicio TYPE i VALUE 0,
      existe_p TYPE i VALUE 0,
      existe_l TYPE i VALUE 0.

IF encabe EQ 'X'.
  inicio = 2.
ELSE.
  inicio = 0.
ENDIF.

LOOP AT itab INTO ttab FROM inicio.

  CLEAR: idat, existe_p, existe_l.
  "SPLIT ttab-rec AT ',' INTO idat-fld1 idat-fld2 idat-fld3 idat-fld4 idat-fld5 idat-fld6.
  SPLIT ttab-rec AT gv_tab INTO idat-fld1 idat-fld2 idat-fld3 idat-fld4 idat-fld5 idat-fld6.

  PERFORM verifica_ut USING idat-fld1 CHANGING existe_l.
  PERFORM verifica_poste USING idat-fld1 CHANGING existe_p.

  IF existe_p = 1 AND existe_l = 0.
    idat-error_p = 0.
  ELSEIF existe_p = 1 AND existe_l = 1.
    idat-error_p = 1.
    idat-error_des = 'El Poste y Linea De Transmision Ya Existen!'.
  ELSEIF existe_p = 0 AND existe_l = 0.
    idat-error_p = 1.
    idat-error_des = 'El Poste y Linea De Transmision No Existen!'.
  ELSEIF existe_p = 0 AND existe_l = 1.
    idat-error_p = 1.
    idat-error_des = 'El Poste No Existe y Linea De Transmision Si Existe!'.
  ENDIF.

  APPEND idat TO tablita.
ENDLOOP.
```

Fuente: elaboración propia, con base en SAP R/3.

Realiza únicamente el llamado al procedimiento Validar, el cual de la tabla interna que tiene los datos realiza un recorrido secuencial de la misma para la verificación de la existencia de los campos utilizando dos funciones Verifica_UT y Verifica_Poste. De acuerdo a la existencia crea el Log de errores a desplegar al final del programa.

Figura 122. Procedimiento Verifica_UT

```
FORM verifica_ut USING ut TYPE string CHANGING
                                existe_1 TYPE i.

DATA: valor      TYPE iflot-tplnr.
" Si existe la linea completa
SELECT SINGLE tplnr INTO valor FROM iflot WHERE tplnr = ut.

IF sy-subrc <> 0.
    existe_1 = 0.
ELSE.
    existe_1 = 1.
ENDIF.

ENDFORM.                                "verifica_ut
```

Fuente: elaboración propia, con base en SAP R/3.

Este recibe como parámetro una cadena y su valor de retorno es un entero que indica la existencia del campo en la base de datos. Para la verificación del campo realiza una consulta a la base de datos a la tabla Iflot (Ubicación Técnica) del campo tplnr (Ubicación Técnica).

Figura 123. Procedimiento Procesa

```
FORM procesa.

    PERFORM validar.
    PERFORM parametros_batch USING modbat.
    PERFORM batch_input.

ENDFORM.                                "procesa
```

Fuente: elaboración propia, con base en SAP R/3.

Si no se selecciona el modo_test, entonces se hace la ejecución de la subrutina Procesa. Al ejecutar esta subrutina ejecuta la subrutina Validar que fue explicada anteriormente, parámetros_batch y el batch_input.

Figura 124. **Procedimiento parámetros_batch**

```
FORM parámetros_batch USING modo TYPE c.  
  gs_ctu_params-dismode = modo.  
  gs_ctu_params-updmode = 'S'.  
  gs_ctu_params-raccommit = 'X'.  
ENDFORM.                                "parámetros_batch
```

Fuente: elaboración propia, con base en SAP R/3.

Establece el modo de actualización a Sincrónico, Commit luego de realizar una carga, el modo batch que es N por default.

Figura 125. **Procedimiento batch input**

```
FORM batch_input.  
  CLEAR: idat.  
  DATA: comilla TYPE string VALUE '',  
        serge TYPE string,  
        herst TYPE string.  
  REFRESH: gt_bdcmsgcoll.  
  CHECK cb_test IS INITIAL.  
  SORT tablita BY error_p ASCENDING.  
  LOOP AT tablita INTO idat WHERE error_p = 0.  
    REFRESH: gt_bdcdata, gt_bdcmsgcoll, gt_items.  
    CLEAR lv_lineatxt.  
    IF idat-error_p = 0.  
      serge = idat-fld4.  
      herst = idat-fld5.  
      IF ( serge CS comilla ).  
        REPLACE ALL OCCURRENCES OF comilla IN serge WITH ' '.  
      ENDIF.  
      IF ( herst CS comilla ).  
        REPLACE ALL OCCURRENCES OF comilla IN herst WITH ' '.  
      ENDIF.  
    ENDIF.  
  ENDLOOP.  
ENDFORM.
```

Fuente: elaboración propia, con base en SAP R/3.

Inicializa la tabla BDC, la cual es la que controla el batch input, realiza un recorrido secuencial de la tabla interna que tiene los datos ya verificados y le da formato a los datos a insertar a la base de datos.

Figura 126. **Código SHDB implementado en ABAP**

```
PERFORM dynpro USING:
  'X'  'SAPMILOO'      '1110',
  ' '  'IFLOS-STRNO'   idat-fld1,    " ubicacion tecnica
  ' '  'RILOO-TPKZ'   'LINTR',      " Indicador de estructura de ubicacion tecnica
  ' '  'IFLO-FLTYP'   'Z'.          " Tipo de ubicacion tecnica

PERFORM dynpro USING:
  'X'  'SAPMILOO'      '2100',
  ' '  'BDC_OKCODE'   '=T\03',
  ' '  'IFLO-PLTXT'   idat-fld2,    " Denominación de la ubicación técnica
  ' '  'ITOB-EQART'   idat-fld3,    " Clase de objeto técnico
  ' '  'ITOB-HERST'   herst,        " Fabricante del activo fijo
  ' '  'ITOB-SERGE'   serge.        " Número de serie según el fabricante

PERFORM dynpro USING:
  'X'  'SAPMILOO'      '2100',
  ' '  'BDC_OKCODE'   '=T\04',
  ' '  'IFLO-PLTXT'   idat-fld2.    " Denominación de la ubicación técnica
```

Fuente: elaboración propia, con base en SAP R/3.

Del código SHDB que se obtuvo al principio para poderlo ejecutar se tiene que llamar a la subrutina DynPro.

Figura 127. Procedimiento DynPro

```
FORM dynpro USING dynbegin TYPE any
  name TYPE any
  value TYPE any.
CLEAR: gs_bdcdata.
IF dynbegin EQ 'X'.
  MOVE: name TO gs_bdcdata-program, "programa
        value TO gs_bdcdata-dynpro, "dynamic process
        'X' TO gs_bdcdata-dynbegin. "indicador dynpro
ELSE.
  MOVE: name TO gs_bdcdata-fnam, "campo
        value TO gs_bdcdata-fval. "valor
ENDIF.
APPEND gs_bdcdata TO gt_bdcdata.
ENDFORM. "dynpro
```

Fuente: elaboración propia, con base en SAP R/3.

En la ejecución de esta subrutina Dynpro, lo que hace es guardar la información del Dynpro o Pantalla grabada por el SHDB, con sus respectivos datos que se ingresaron manualmente en la transacción IL01.

Figura 128. Llamada a la función Llamar transacción

```
CONCATENATE idat-fld1 idat-fld2 idat-fld3 idat-fld6 INTO lv_lineatxt SEPARATED BY ' - '.
PERFORM llamar_transaccion USING lv_lineatxt 'IL01'.
CLEAR: idat.
ENDIF.
ENDLOOP.
ENDFORM. "batch input
```

Fuente: elaboración propia, con base en SAP R/3.

Luego de almacenar la información de los Dynpros se ejecuta el Call Transaction, el cual especifica que es la transacción IL01

Figura 129. **Procedimiento Llamar transacción**

```
FORM llamar_transaccion USING mensaje TYPE any
                           transaccion TYPE any.

"CALL TRANSACTION 'IRO2' USING gt_bdcdata OPTIONS FROM gs_ctu_params MESSAGES INTO gt_bdcmsgcoll.
CALL TRANSACTION transaccion USING gt_bdcdata OPTIONS FROM gs_ctu_params MESSAGES INTO gt_bdcmsgcoll.

IF gt_bdcmsgcoll[] IS NOT INITIAL.

    PERFORM logs_batchinput USING mensaje.
ENDIF.

ENDFORM.                "LLAMAR TRANSACCION
```

Fuente: elaboración propia, con base en SAP R/3.

Al llamar esta subrutina, obtiene los valores que se ingresaron en los dynpros y el número de transacción para realizar la carga de los datos ingresados por el usuario, y de salida ingresa el log de errores a una tabla interna, para que luego al final del programa se realice una lectura de esta tabla para mostrar los errores.

Figura 130. Función de generación de log de batch input

```
FORM logs_batchinput USING x_item TYPE any.

DATA: ls_t100 TYPE t100,
      lv_mstring TYPE c LENGTH 500.

LOOP AT gt_bdcmsgcoll INTO gs_bdcmsgcoll.

  SELECT SINGLE * INTO ls_t100 FROM t100 WHERE sprsl = gs_bdcmsgcoll-msgspra
                                         AND arqgb = gs_bdcmsgcoll-msgid
                                         AND msgnr = gs_bdcmsgcoll-msgnr.

  IF sy-subrc = 0.

    lv_mstring = ls_t100-text.

    IF lv_mstring CS '&1'.
      REPLACE '&1' WITH gs_bdcmsgcoll-msgv1 INTO lv_mstring.
      REPLACE '&2' WITH gs_bdcmsgcoll-msgv2 INTO lv_mstring.
      REPLACE '&3' WITH gs_bdcmsgcoll-msgv3 INTO lv_mstring.
      REPLACE '&4' WITH gs_bdcmsgcoll-msgv4 INTO lv_mstring.
    ELSE.
      REPLACE '&' WITH gs_bdcmsgcoll-msgv1 INTO lv_mstring.
      REPLACE '&' WITH gs_bdcmsgcoll-msgv2 INTO lv_mstring.
      REPLACE '&' WITH gs_bdcmsgcoll-msgv3 INTO lv_mstring.
      REPLACE '&' WITH gs_bdcmsgcoll-msgv4 INTO lv_mstring.
    ENDIF.

    CONDENSE lv_mstring.

    CLEAR: gs_items.
    MOVE-CORRESPONDING gs_bdcmsgcoll TO gs_items.
    gs_items-msgtext = lv_mstring.
    gs_items-item = x_item.
    APPEND gs_items TO gt_items.
  ENDIF.

ENDLOOP.

ENDFORM. " LOGS_BATCHINPUT
```

Fuente: elaboración propia, con base en SAP R/3.

Preparación del log de errores de ingreso de datos, junto con los errores generados por el sistema.

Figura 131. Despliegue de log de batch input

```
FORM end_selection.
DATA: "contador TYPE i VALUE 1,
      item TYPE c LENGTH 85,
      mensaje TYPE c LENGTH 100.

mensaje = 'MENSAJE'.

WRITE:/   item                COLOR COL_KEY INTENSIFIED ON,
          'Tipo De Mensaje'   COLOR COL_KEY INTENSIFIED ON,
          'ID Del Mensaje'    COLOR COL_KEY INTENSIFIED ON,
          'Numero De Mensaje' COLOR COL_KEY INTENSIFIED ON,
          mensaje             COLOR COL_KEY INTENSIFIED ON.

SKIP 1.
LOOP AT gt_items INTO gs_items.

  IF ( gs_items-msgtyp EQ 'W' ).

    WRITE:/ gs_items-item(85) COLOR COL_NORMAL INTENSIFIED ON,
            gs_items-msgtyp  COLOR COL_NORMAL INTENSIFIED OFF,
            gs_items-msgid   COLOR COL_NORMAL INTENSIFIED OFF,
            gs_items-msgnr   COLOR COL_NORMAL INTENSIFIED OFF,
            gs_items-msgtext(95) COLOR COL_NORMAL INTENSIFIED OFF.

  ELSE.

    WRITE:/ gs_items-item(85) COLOR COL_NORMAL INTENSIFIED ON,
            gs_items-msgtyp  COLOR COL_NEGATIVE INTENSIFIED OFF,
            gs_items-msgid   COLOR COL_NEGATIVE INTENSIFIED OFF,
            gs_items-msgnr   COLOR COL_NEGATIVE INTENSIFIED OFF,
            gs_items-msgtext(95) COLOR COL_NEGATIVE INTENSIFIED OFF.

  ENDIF.

ENDLOOP.
ENDIF.

ENDFORM.                    " END SELECTION
```

Fuente: elaboración propia, con base en SAP R/3.

Despliegue del Log de Errores en pantalla, recorriendo la tabla que los contiene, en el write le da el formato indicado dependiendo del tipo de error, con sus colores respectivos.

2.3. Transacciones para monitoreo

Las transacciones para monitoreo son importantes para poder ver el desempeño de los servidores con respecto a la carga que se les están dando, por medio las diferentes transacciones que se están ejecutando en tiempo real. Esto con el objetivo de que los diferentes servidores estén funcionando en óptimas condiciones.

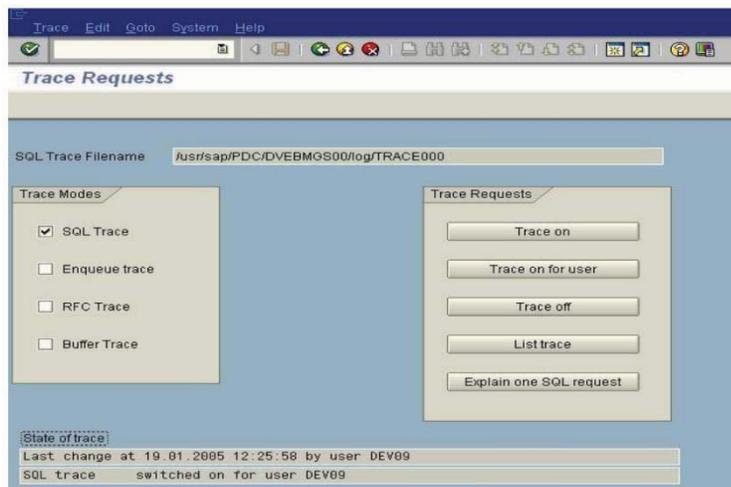
2.3.1. Transacción ST05, Análisis de Rastreo SQL (SQL Trace)

Es una herramienta de análisis de rendimiento de SQL, esta muestra que sentencias de SQL que se ejecutan al utilizar una cierta transacción estándar o desarrollada. Con el motivo de monitorear si las consultas de la transacción ejecutada son óptimas.

Para utilizar esta herramienta se utiliza la transacción ST05, los pasos a seguir para utilizarla son los siguientes:

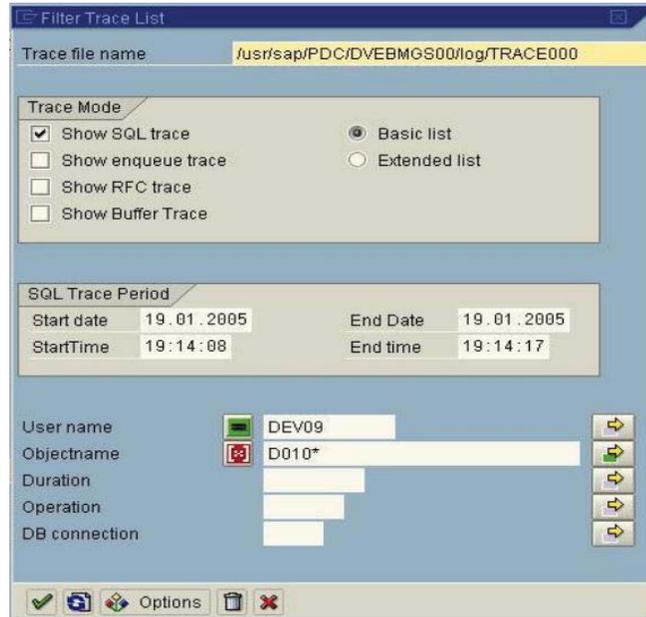
- A. Accesar a la transacción ST05. Desde la pantalla inicial, darle clic en el check box para SQL Trace, luego clic en Trace On.
- B. Ejecutar el programa en la transacción se38, que contiene la sentencia SQL que se desea analizar.
- C. Regresar a la transacción ST05, Clic en Trace Off.
- D. Clic en List Trace, para visualizar el resultado del análisis.

Figura 132. Pantalla inicial de la transacción



Fuente: elaboración propia, con base en SAP R/3.

Figura 133. Pantalla Filter Trace List



Fuente: elaboración propia, con base en SAP R/3.

Pantalla luego de dar clic en List Trace.

Figura 134. Pantalla de resultados de las sentencias SQL

The screenshot shows the SAP Basic Trace List interface. It displays two sections of data for different transactions. The first section is for Transaction = SE38, and the second is for Transaction = ST05. Each section contains a table with columns for Duration, ObjectName, Op., Rec., RC, and Statement. The rows are color-coded: yellow for REOPEN and FETCH operations, and pink for OPEN and FETCH operations.

Transaction = SE38					
		PID =	896	P type	DIA
		Client =	210	User = DEV09	
Duration	ObjectName	Op.	Rec.	RC	Statement
11,767	PROGDIR	REOPEN	0	0	SELECT WHERE "NAME" = 'ZPERFORMANCE_SELECT_SORT1' AND "STATE" = 'A'
802	PROGDIR	FETCH	1	0	
21	DWINACTIV	REOPEN	0	0	SELECT WHERE "OBJECT" = 'REPS' AND "OBJ_NAME" = 'ZPERFORMANCE_SELECT_SORT1'
450	DWINACTIV	FETCH	0	1403	
570	MARA	PREPARE	0	0	SELECT WHERE (T_01 . "MANDT" = :A0 AND T_00 . "MATNR" = T_01 . "MATNR") AND T_00 . "MANDT" = :A1 A
20	MARA	OPEN	0	0	SELECT WHERE (T_01 . "MANDT" = '210' AND T_00 . "MATNR" = T_01 . "MATNR") AND T_00 . "MANDT" = '21
591,367	MARA	FETCH	371	1403	

Transaction = ST05					
		PID =	8438	P type	DIA
		Client =	210	User = DEV09	
Duration	ObjectName	Op.	Rec.	RC	Statement
583,376	TFDIR	REOPEN	0	0	SELECT WHERE "FUNCNAME" = 'PERFORMANCE_TRACE_OFF'
769	TFDIR	FETCH	1	0	
21	TFDIR	REOPEN	0	0	SELECT WHERE "FUNCNAME" = 'SQLT_TRACE_OFF'
340	TFDIR	FETCH	1	0	

Fuente: elaboración propia, con base en SAP R/3.

Luego de la pantalla de List Trace, muestra el resultado de las sentencias SQL ejecutadas en la transacción. Esta brinda el tiempo en duración de cada operación SQL cuando se abre la consulta, cuando obtiene los datos, cuando se reabre la consulta y cuando se ejecuta. El tiempo de duración está clasificado por colores.

Figura 135. Pantalla de sumarización

PID	Typ	Cl	Time	Tcode/prog	Table	SQL_op	Recs.	Time
896	DIA	210	19:14:12.142	SE38	PROGDIR	SEL	1	12,569
896	DIA	210	19:14:12.155	SE38	DWINACTIV	SEL	0	471
896	DIA	210	19:14:12.446	SE38	MARA	SEL	371	591,387
8438	DIA	210	19:14:16.479	ST05	TFDIR	SEL	1	584,145
8438	DIA	210	19:14:17.064	ST05	TFDIR	SEL	1	361

Fuente: elaboración propia, con base en SAP R/3.

Se pueden sumarizar los tiempos de ejecución, para obtener la sumarización se debe de ir al menú GOTO->Summary. Esta opción brinda información los tiempos de utilización de cada tabla que está presente en la consulta, así como también a que proceso pertenece la consulta esto se ve en la columna PID (Process ID), SQL Op Recs significa que tipo de sentencia SQL se está analizando (en este caso SEL = Select).

Figura 136. Pantalla de monitoreo de una transacción en específico

Tcode/prog	Table	SQL op	Accesses	Recs.	Time Percent
SE38	DWINACTIV	SEL	1	0	471 0.1
SE38	MARA	SEL	1	371	591,387 97.8
SE38	PROGDIR	SEL	1	1	12,569 2.1
Total			3	372	604,427 100.0

Fuente: elaboración propia, con base en SAP R/3.

Sumarización de una consulta en específico, Solo con dar doble clic en un PID en específico como se mostró en la imagen anterior entonces muestra esta pantalla.

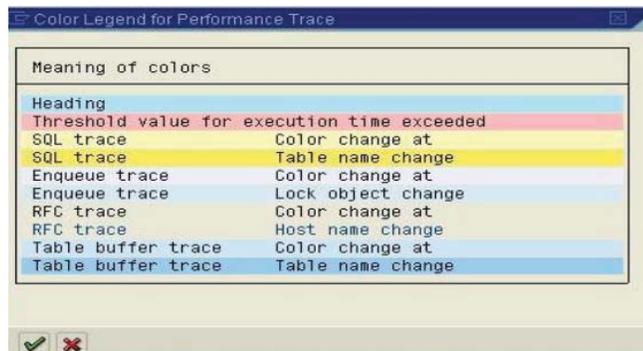
Figura 137. Pantalla de leyenda del reporte

Duration	Obj	SQL op	Accesses	Recs.	Time Percent
11,767	PRO	FETCH	0	1403	
802	PRO	PREPARE	0	0	
21	DWI	OPEN	0	0	
450	DWINACTIV	FETCH	0	1403	
578	MARA	PREPARE	0	0	
20	MARA	OPEN	0	0	
591,367	MARA	FETCH	371	1403	

Fuente: elaboración propia, con base en SAP R/3.

Para ver el significado de los colores o leyenda desplegados en la pantalla se debe de ir al menú GOTO-> Show Color Legend.

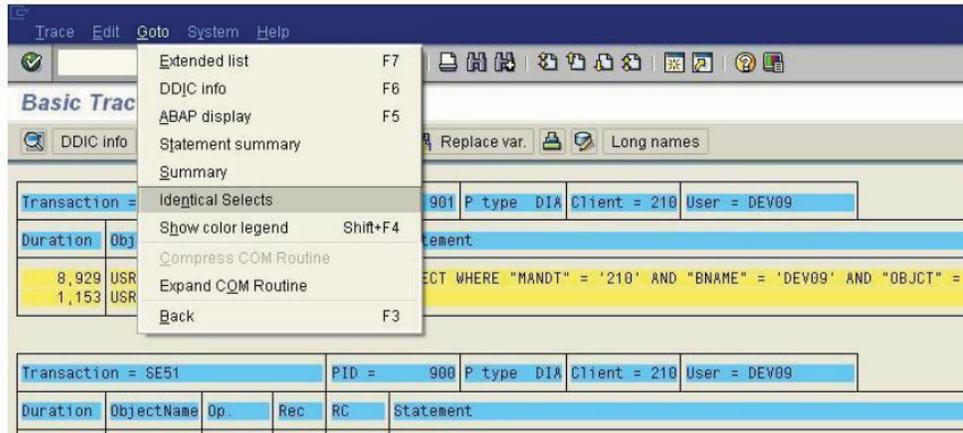
Figura 138. **Leyenda del reporte**



Fuente: elaboración propia, con base en SAP R/3.

Leyenda del análisis Trace SQL, el color rosado significa que el tiempo fue demasiado y esta excedido, el amarillo significa el cambio de sentencia SQL, el azul el cambio del buffer de la tabla, el blanco el cambio de cola, y el blanco el cambio de servidor o host (esto es en caso de una función remota accedida desde otro sistema).

Figura 139. Verificación de sentencias sql idénticas



Fuente: elaboración propia, con base en SAP R/3.

Para ver si hay sentencias SQL idénticas, esto sirve para poder reutilizar los resultados de las sentencias SQL y así minimizar el número de consultas en el programa. Por lo tanto ayuda al rendimiento del servidor.

Figura 140. Pantalla de resultados de sentencias de sql idénticas

The screenshot shows the 'Identical Selects' results table in SAP Basic Trace. The table has three columns: Executions, ObjectName, and Where Clause. The data is as follows:

Executions	ObjectName	Where Clause
5	DWINACTIV	"OBJECT" = 'REPS' AND "OBJ_NAME" = 'LZBJ\$01' AND ROWNUM <= 1
3	DWASYNCR	"UNAME" = 'DEV09'
6	E070	"TRKORR" = 'P0CK902454'
6	E070	"TRKORR" = 'P0CK902348'
2	EUDB	"RELID" = 'CU' AND "NAME" = 'SAPMSEUS' AND "SPRSL" = 'D' AND "SRTF2" = 0
2	EUDB	"RELID" = 'CU' AND "NAME" = 'SAPMSEUS' AND "SPRSL" = 'D' AND "SRTF2" >= 0 ORDER BY 1, 2, 3, 4
26	LICENSE	
2	MARA	(T_01 . "MANDT" = '210' AND T_00 . "MATNR" = T_01 . "MATNR") AND T_00 . "MANDT" = '210' AND T_01 . "NAME" = 'Z00ALVF14' AND "STATE" = 'I'
6	PROGDIR	"NAME" = 'Z00ALVF14' AND "STATE" = 'I'
3	PROGDIR	"NAME" = 'YWRUTALV0BJ' AND "STATE" = 'A'
6	PROGDIR	"NAME" = 'YWRUTALV0BJ' AND "STATE" = 'I'
3	RSEUINC	"INCLUDE" = 'YWRUTALV0BJ'
2	RSMPTXTS	"PROGRAM" = 'SAPMSEUS' AND "SPRSL" = 'D'
2	RSMPTXTS	"PROGRAM" = 'SAPMSEUS' AND "SPRSL" = 'E' AND "TEXT" <> ' '
2	RSMNORMB	"RELID" = 'ST' AND "NAME" = 'SCREENS' AND "O_TYPE" = ' ' AND "SRTF2" >= 0 ORDER BY 1, 2, 3, 4
2	SXC_ATTR	(T_00 . "IMP_NAME" = T_01 . "IMP_NAME") AND T_00 . "ACTIVE" = 'X' AND T_01 . "GUI_PROG" = 'SAPMSEUS'
2	TLOCK	"OBJECT" = 'PROG' AND "HIKEY" >= 'Z00ALVF14' REPOREPS' AND "HIKEY" LIKE
4	TLOCK	"OBJECT" = 'PROG' AND "HIKEY" >= 'YWRUTALV0BJ' REPOREPS' AND "HIKEY" LIKE

Fuente: elaboración propia, con base en SAP R/3.

Ejecución de las sentencias SQL idénticas.

2.3.2. Operaciones de base de datos medidas por el SQL Trace

El análisis SQL Trace ayuda a medir el tiempo de ejecución de cada operación de base de datos que son ejecutadas dentro de una sentencia SQL.

- DECLARE: esta operación crea un nuevo cursor y este es asignado a una sentencia SQL.
- PREPARE: esta operación convierte la sentencia OPEN SQL ABAP a SQL Nativo.
- OPEN: esta operación abre el cursor declarado y envía los parámetros a la base de datos.
- FETCH: esta operación envía uno o más registros obtenidos de la base de datos hacia la interface R/3.
- REOPEN: esta operación reabre un cursor para enviarle uno o más parámetros adicionales para que estos sean enviados a la base de datos.
- EXEC: esta operación envía los parámetros a la base de datos, y ejecuta las sentencias (Insert, Update, Delete) que cambian los datos en la base de datos.

2.4. Winshuttle

Es un sistema que provee varios módulos que le brindan al usuario final de SAP una mejor manejabilidad, flexibilidad, extracción, importación de datos, creación de reportes desde SAP hacia formato de Excel, etc. Sin la necesidad de realizar desarrollos en ABAP, sin tener los conocimientos técnicos de

programación de SQL. Por lo que este sistema brinda mayor productividad, ahorro de tiempo eliminando los tiempos de desarrollo y eliminando los riesgos de mal manejo de información ya sea cargando información erróneamente al sistema o la realización de reportes con información poco relevante.

En este capítulo solo se presentaran los módulos de Winshuttle más sencillos como también para realizar la comparación entre Winshuttle y los desarrollos en ABAP.

2.4.1. Winshuttle Transaction

Registra los pasos que da un usuario hasta que termina una transacción de SAP y asigna los campos de SAP relevantes a una hoja de cálculo de Excel, al mismo tiempo que crea sin complicaciones un archivo SHUTTLE que puede funcionar como una plantilla de transacción. Esta plantilla se puede ejecutar para transferir datos entre Excel y SAP y para automatizar cualquier proceso a la vez que se utiliza la seguridad SAP nativa y las autorizaciones para garantizar el cumplimiento de las regulaciones.

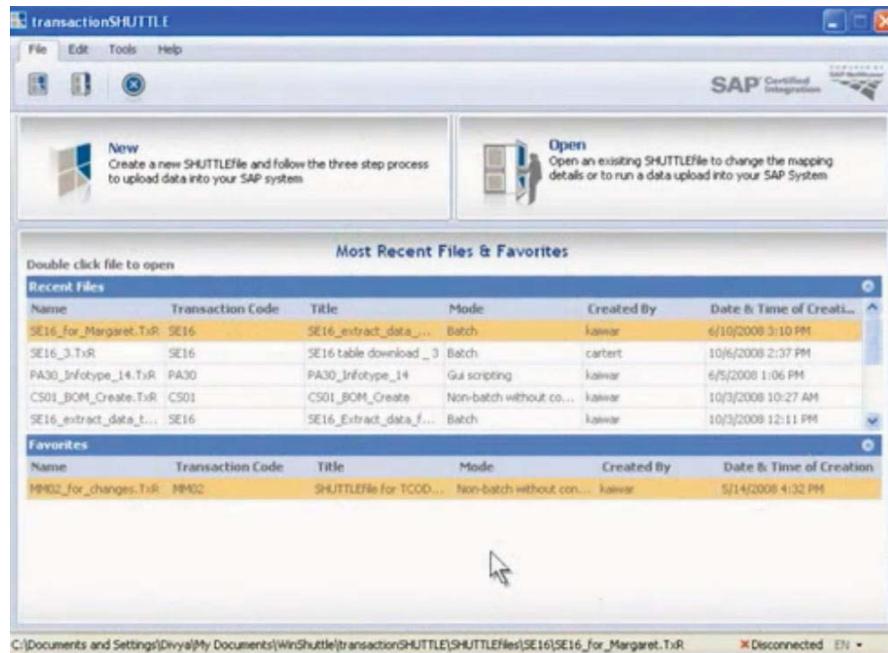
Se puede decir que Winshuttle tiene la funcionalidad de la transacción SHDB descrita anteriormente, ya que esta guarda los pasos que se realizan en las transacciones.

Figura 141. Se tienen los datos empresariales en Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		Material Number	Gross weight	Net weight	MRP Controller	Purchasing Group							
2	937		37	27	003	000							
3	938		38	28	002	012							
4	939		38	28	003	002							
5	940		39	29	002	000							
6	941		39	29	003	012							
7	942		40	30	002	002							
8	943		40	30	003	000							
9	Material 7896		41	31	002	002							

Fuente: elaboración propia, con programa de Excel.

Figura 142. Pantalla principal de Transaction Shuttle o WinShuttle Transaction



Fuente: elaboración propia, con programa de Winshuttle Transaction.

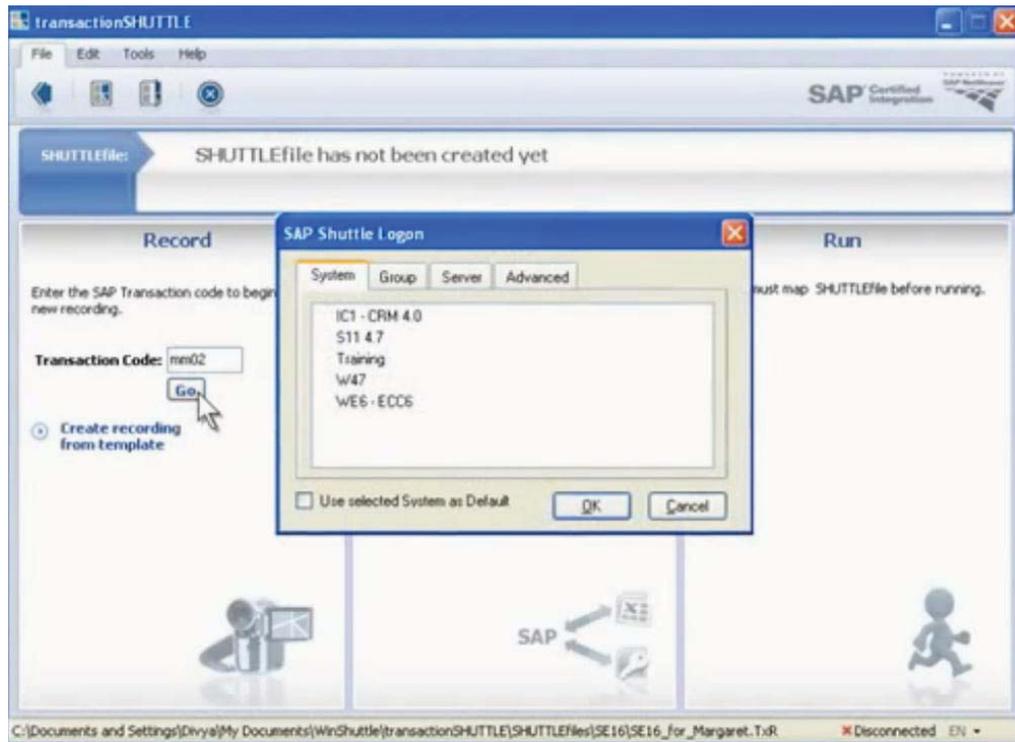
Se selecciona la Opción New o Nuevo y muestra la siguiente ventana:

Figura 143. **Proceso de creación de transacción en Transaction Shuttle**



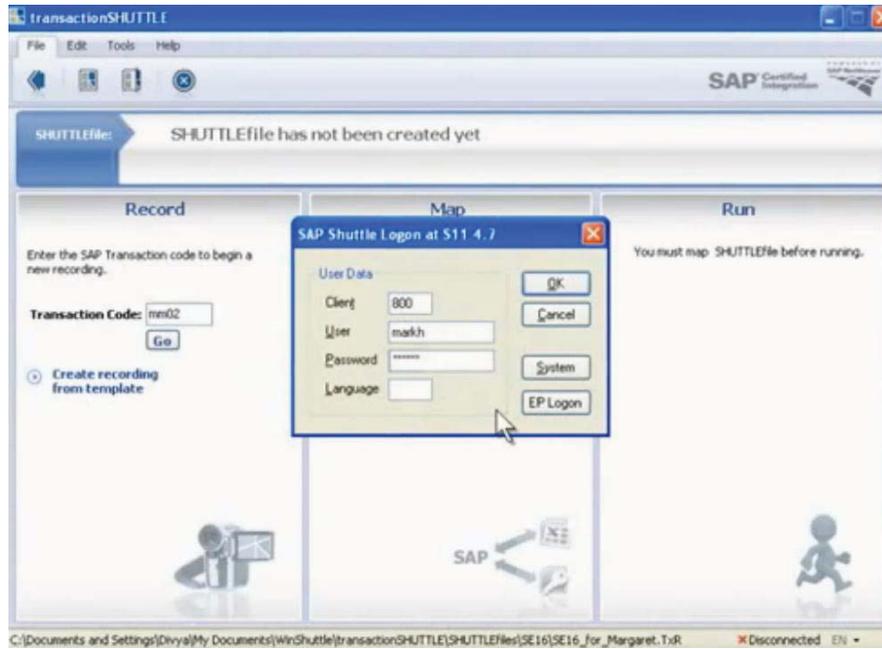
Fuente: elaboración propia, con programa de Winshuttle Transaction.

Figura 144. Se ingresa el código de transacción y luego se selecciona el módulo de SAP a acceder



Fuente: elaboración propia, con programa de Winshuttle Transaction.

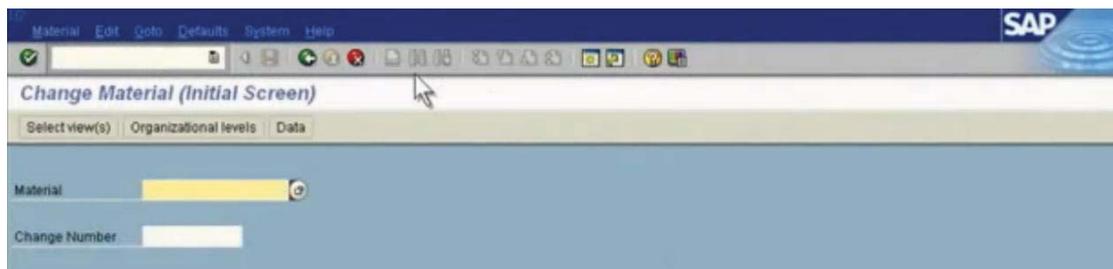
Figura 145. Login a SAP en Transaction Shuttle



Fuente: elaboración propia, con programa de Winshuttle Transaction.

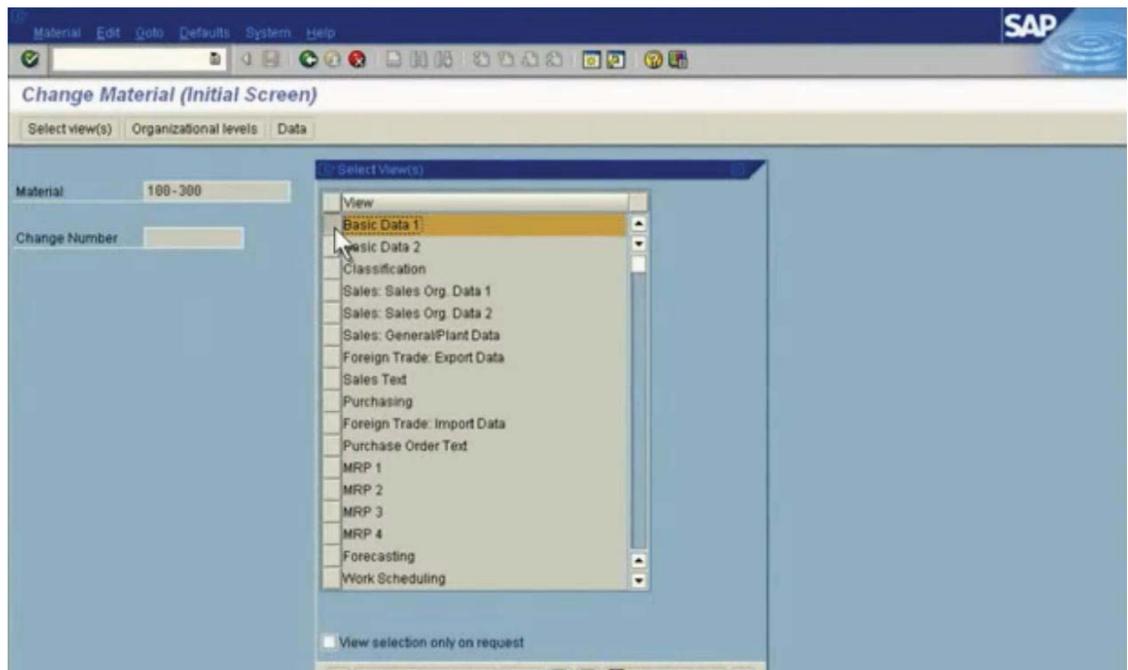
Lleva directamente a SAP y a la transacción ingresada como se muestra a continuación:

Figura 146. Transacción de cambio de material de SAP



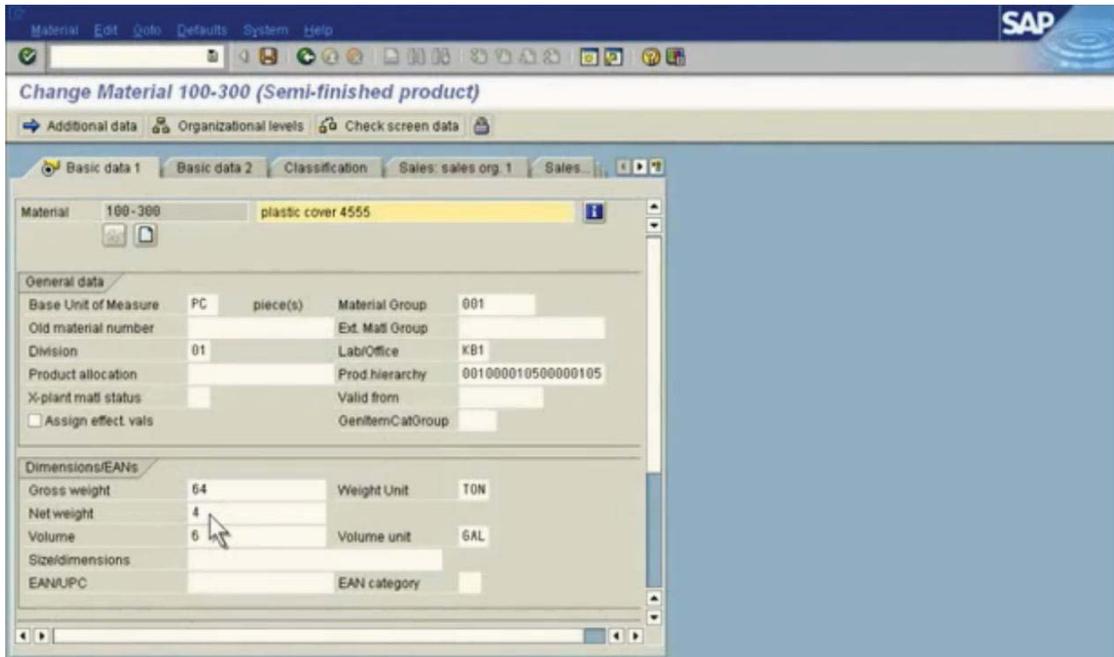
Fuente: elaboración propia, con base en SAP R/3.

Figura 147. Selección de datos básicos para la grabación de la transacción



Fuente: elaboración propia, con base en SAP R/3.

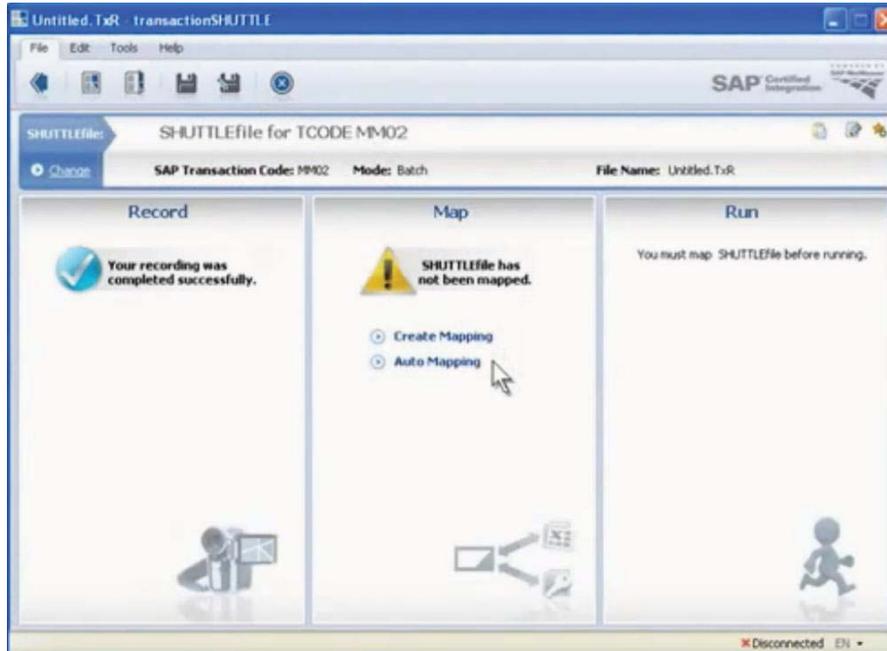
Figura 148. Inicio de grabación de ingreso de datos en la transacción



Fuente: elaboración propia, con base en SAP R/3.

Ya que se finalizó la grabación entonces se comienza a realizar el mapeo de datos:

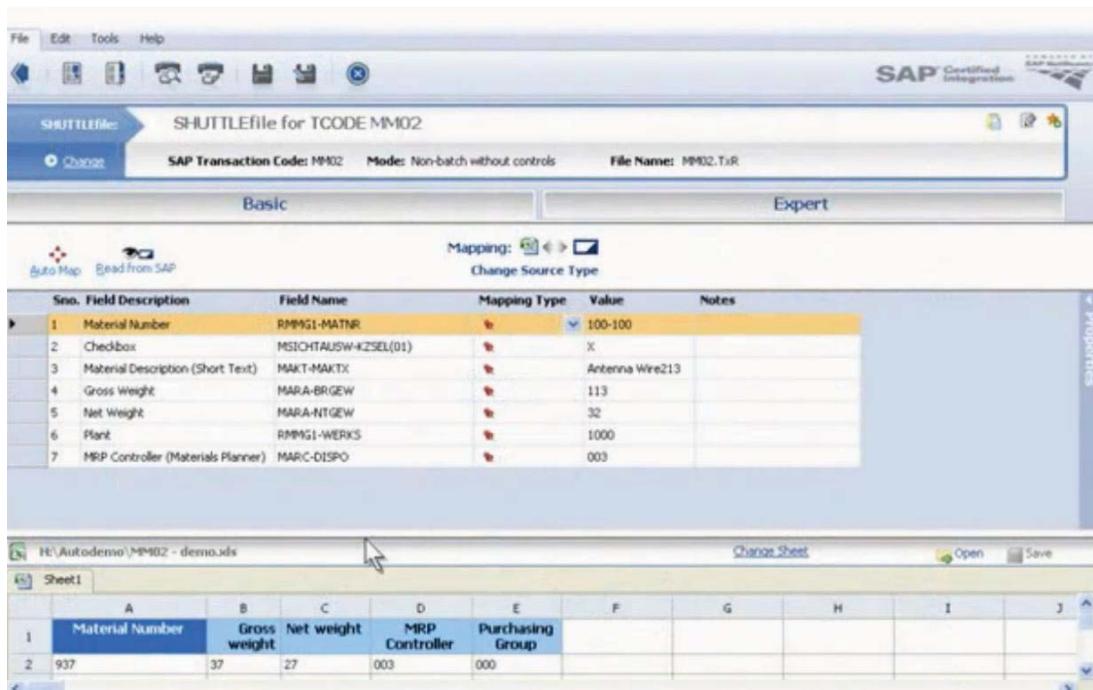
Figura 149. Mapeo de datos



Fuente: elaboración propia, con programa de Winshuttle Transaction.

Luego se tienen los pasos realizados en la transacción anteriormente y el archivo de Excel:

Figura 150. Código de la transacción y archivo de Excel juntos

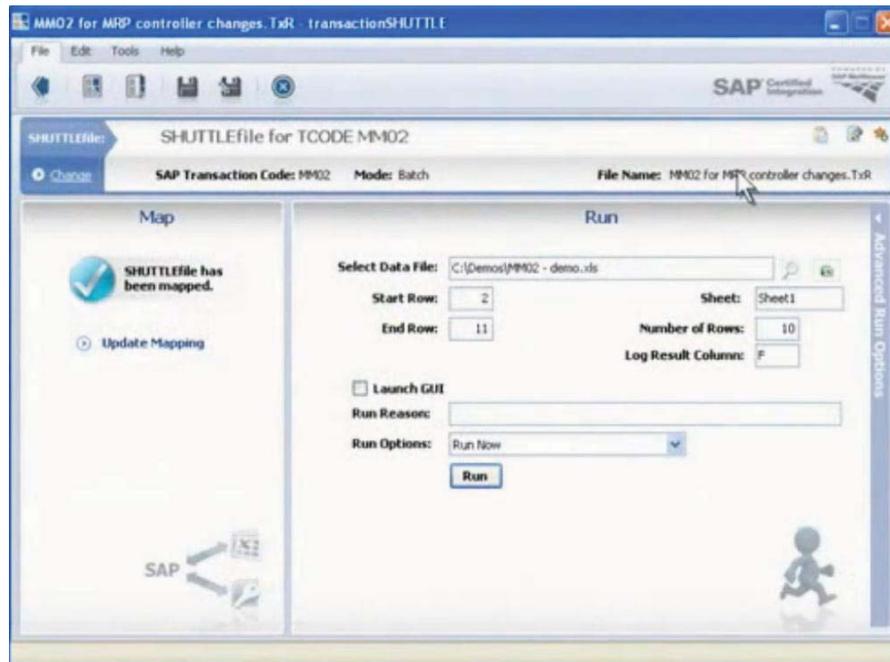


Fuente: elaboración propia, con programa de Winshuttle Transaction.

Solo consiste en arrastrar los campos de Excel hacia el código de la transacción o viceversa.

Ya que se realizó todo este proceso ya se comienza a realizar la transposición de información entre Excel y SAP.

Figura 151. **Proceso de transposición de datos entre Excel y SAP o viceversa**



Fuente: elaboración propia, con programa de Winshuttle Transaction.

Al ejecutar la carga, nuevamente Transaction Shuttle pide los datos para loguearse en SAP y luego se puede visualizar en tiempo real el proceso de carga de datos desde Excel hacia SAP, marcando en rojo las filas que no se han cargado exitosamente.

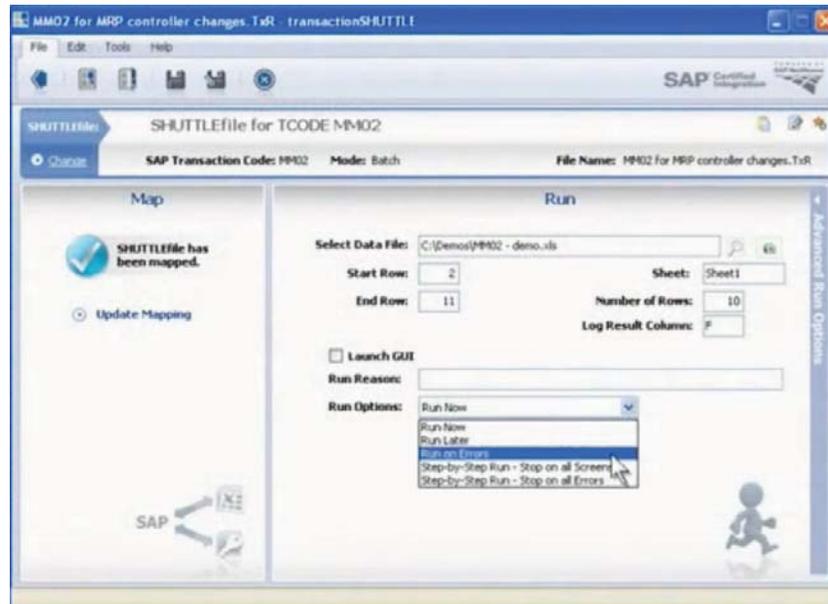
Figura 152. Carga de datos en tiempo real de Excel hacia SAP

	A	B	C	D	E	F	G	H	I	J
	Material Number	Gross weight	Net weight	MRP Controller	Purchasing Group	Material Description (Short Text)	Log runSHUTTLE addin 0.0 SP 1 Update 3 WE6-800, belindam Mode: Non-Batch without Controls 6/2/2008 9:16:12 AM MM02.TxR			
1										
2	100-100	37	27	003	000	Antenna W	No changes made			
3	100-101	38	28	002	012	100-101	No changes made			
4	100-110	38	28	003	002	template7	No changes made			
5	100-120	39	29	002	000	100-120	No changes made			
6	100-130	39	29	003	012	100-130	No changes made			
7	100-200	40	30	002	002	kgh	No changes made			
8	Material 7886	40	30	003	000		This material is not active in SAP and cannot be loaded. not activated			
9	100-250	41	31	002	002	100-250	No changes made			
10	100-251	41	31	003	002	100-251	No changes made			
11	100-260	42	32	002	000	100-260	No changes made			
12	100-261	42	32	003	012	100-261	Material 100-261 changed			
13	100-300	43	33	002	002	Antenna W	Material 100-300 changed			
14	100-301	43	33	003	000	100-301	Material 100-301 changed			
15	100-302	44	34	002	012	Antenna W	Material 100-302 changed			
16	100-310	44	34	003	002	100-310	Material 100-310 changed			
17	100-400	45	35	002	000	Cable111	Material 100-400 changed			
18	100-401	38	28	002	012					
19	100-101	39	29	003	000					

Fuente: elaboración propia, con programa de Excel.

Entonces se pueden realizar las correcciones necesarias en el archivo de Excel y luego se puede realizar la carga de nuevo, solo que utilizando una nuevo tipo de ejecución la cual se llama Run on Errors, como se muestra a continuación:

Figura 153. Ejecución de los errores del archivo de Excel



Fuente: elaboración propia, con programa de Winshuttle Transaction.

Y al darle ejecutar, de nuevo se hace la carga de forma segura utilizando el login de SAP y luego se realiza la ejecución de errores de carga.

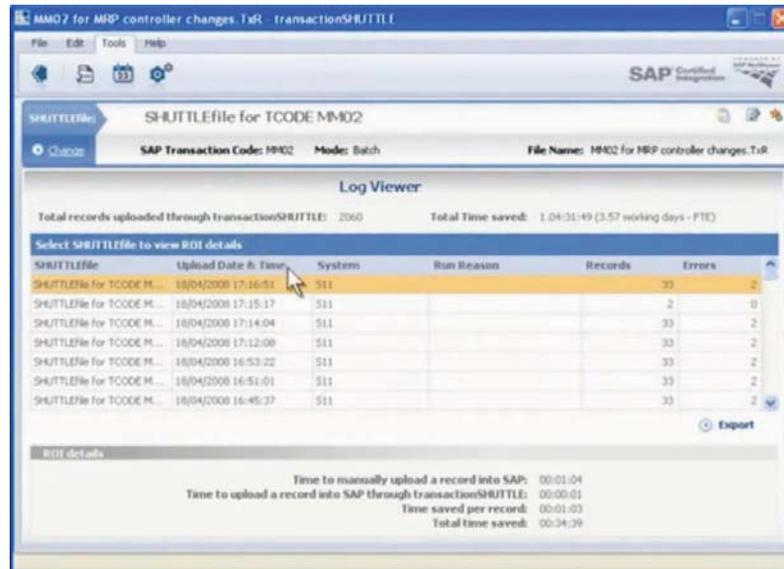
Figura 154. Realización de carga de datos erróneos

	A	B	C	D	E	F	G	H	I	J
	Material Number	Gross weight	Net weight	MRP Controller	Purchasing Group	Material Description (Short Text)	Log runSHUTTLE addin 8.0 SP 1 Update 3 WE6-800, belindam Mode: Non-Batch without Controls 6/2/2008 9:16:12 AM MMD2 TxR			
1						MAKT-MAKT				
2	100-100	37	27	003	000	Antenna W	No changes made			
3	100-101	38	28	002	012	100-101	Material 100-101 changed			
4	100-110	38	28	003	002	template7	Material 100-110 changed			
5	100-120	39	29	002	000	100-120	Material 100-120 changed			
6	100-130	39	29	003	012	100-130	Material 100-130 changed			
7	100-200	40	30	002	002	kgh	Material 100-200 changed			
8	100-300	40	30	003	000	Antenna W	No changes made			
9	100-250	41	31	002	002	100-250	Material 100-250 changed			
10	100-251	41	31	003	002	100-251	Material 100-251 changed			
11	100-260	42	32	002	000	100-260	Material 100-260 changed			
12	100-261	42	32	003	012	100-261	Material 100-261 changed			
13	100-300	43	33	002	002	Antenna W	Material 100-300 changed			
14	100-301	43	33	003	000	100-301	Material 100-301 changed			
15	100-302	44	34	002	012	Antenna W	No changes made			
16	100-310	44	34	003	002	100-310	No changes made			
17	100-400	45	35	002	000	Cable111	No changes made			
18	100-401	38	28	002	012	100-401	No changes made			
19	100-101	39	29	004	000	100-101	Material 100-101 changed			
20	100-110	38	28	003	002	template7	Material 100-110 changed			

Fuente: elaboración propia, con programa de Excel.

Se tiene un LOG Viewer para ver la información se ha cargado, las transacciones de SAP utilizadas y el rendimiento de carga.

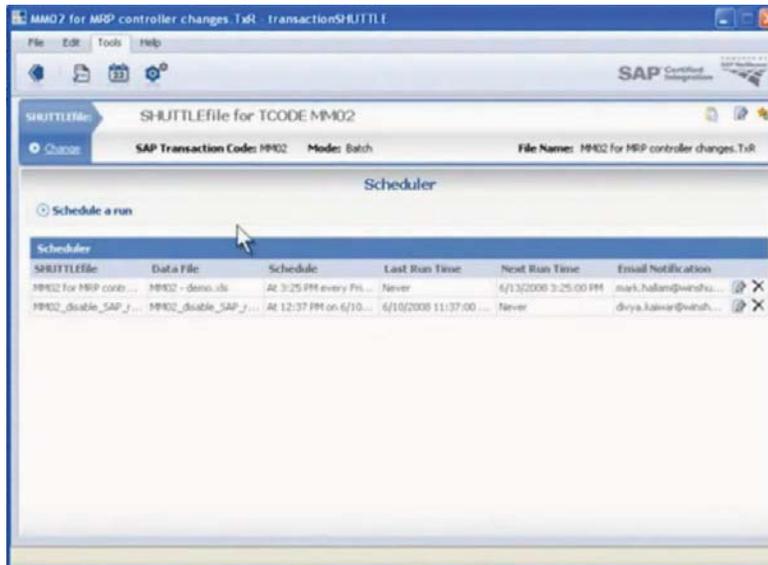
Figura 155. Log Viewer de Transaction Shuttle



Fuente: elaboración propia, con programa de Winshuttle Transaction.

Se puede tener un planificador de carga, para tener un horario automático:

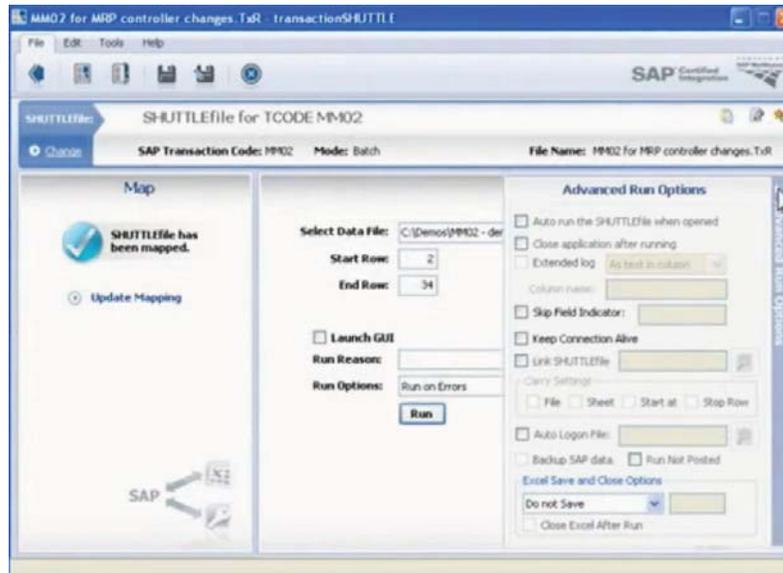
Figura 156. **Planificador de carga automático Transaction Shuttle**



Fuente: elaboración propia, con programa de Winshuttle Transaction.

También se pueden realizar plantillas de carga para que estos se puedan utilizar todas las veces que se desean, estos se pueden considerar como un backup o copia de seguridad para SAP.

Figura 157. Utilización de plantillas de carga

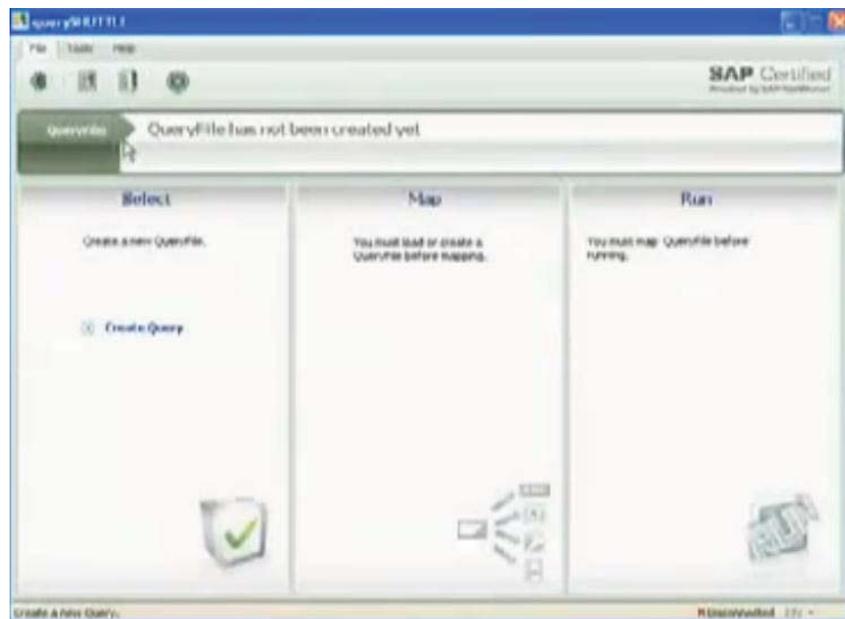


Fuente: elaboración propia, con programa de Winshuttle Transaction.

Lo que permite es evitar enviar datos que ya fueron procesados como se muestra en la siguiente figura, los que están en gris son datos que ya fueron procesados y no se volverán a cargar.

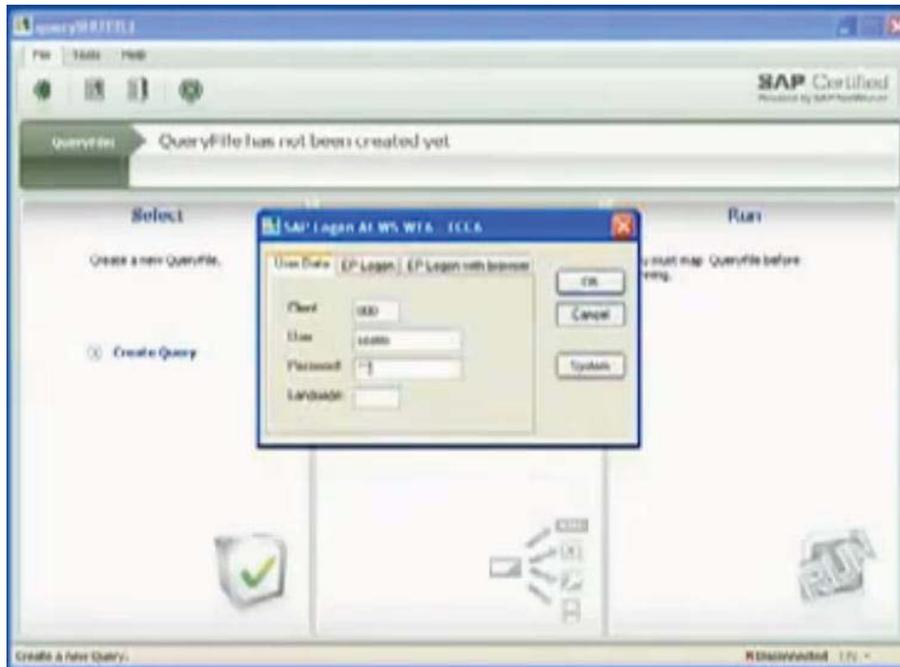
- Selección (Select): consiste en seleccionar las tablas de SAP y campos.
- Mapeo (Map): selección de campos de salida en el archivo de Excel para el reporte.
- Ejecutar (RUN): ejecución del reporte y almacenamiento del mismo.

Figura 159. Selección de nueva consulta



Fuente: elaboración propia, con programa de Winshuttle Query.

Figura 160. Pantalla de Login



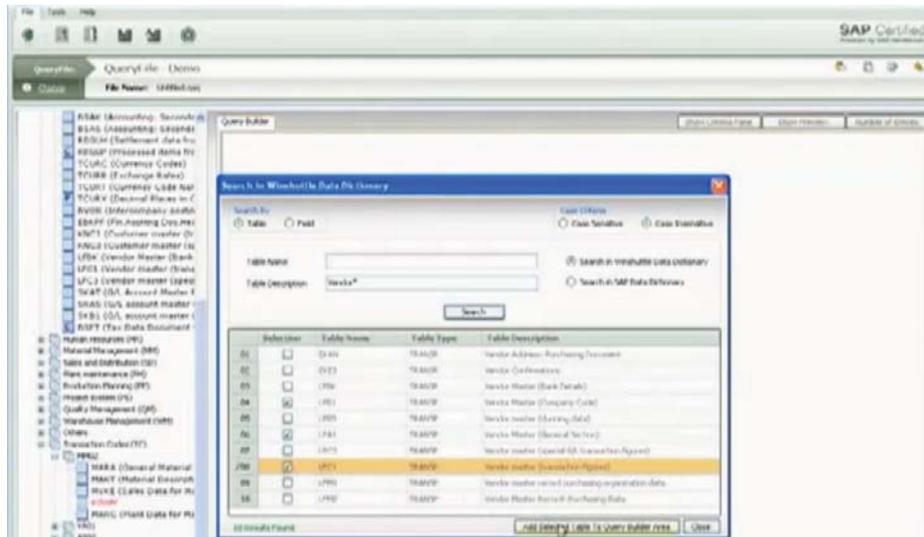
Fuente: elaboración propia, con programa de Winshuttle Query.

Figura 161. Selección de tablas de SAP



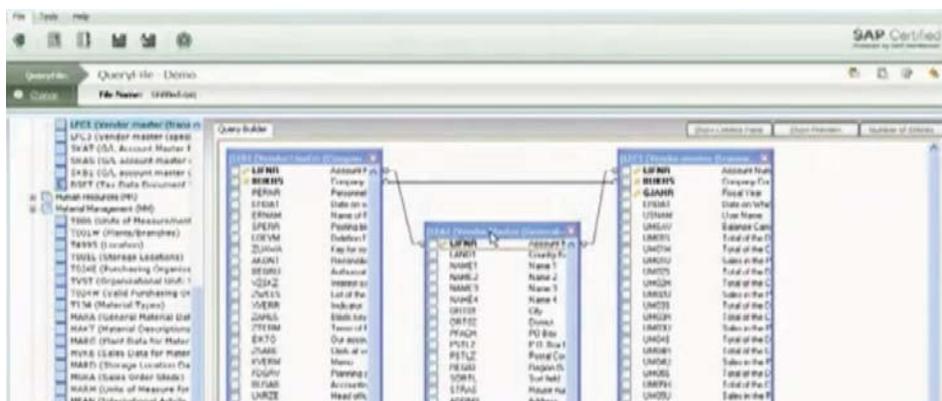
Fuente: elaboración propia, con programa de Winshuttle Query.

Figura 162. Selección de campos de la tabla seleccionada



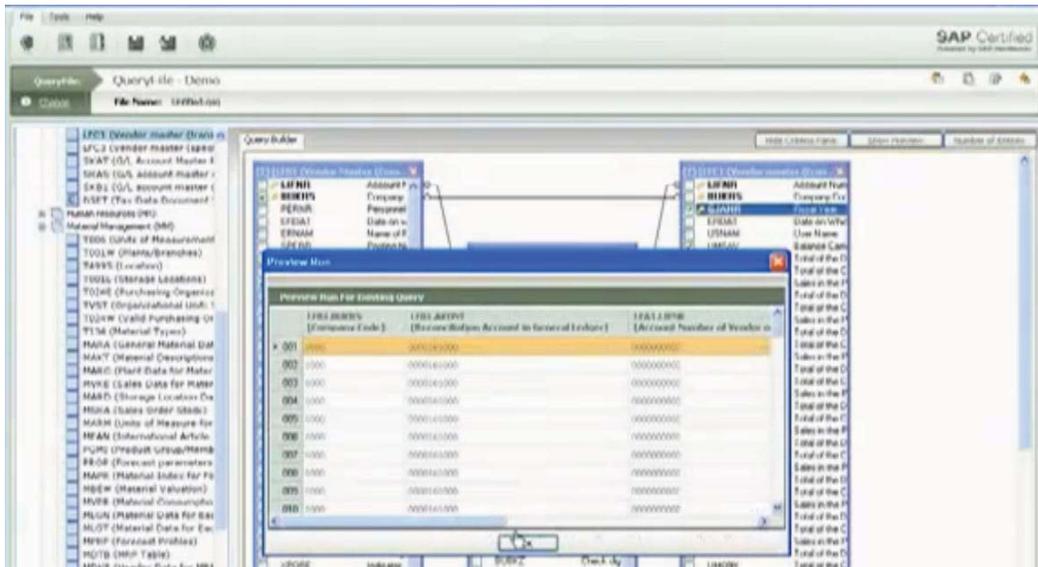
Fuente: elaboración propia, con programa de Winshuttle Query.

Figura 163. Tablas seleccionadas para realizar la consulta



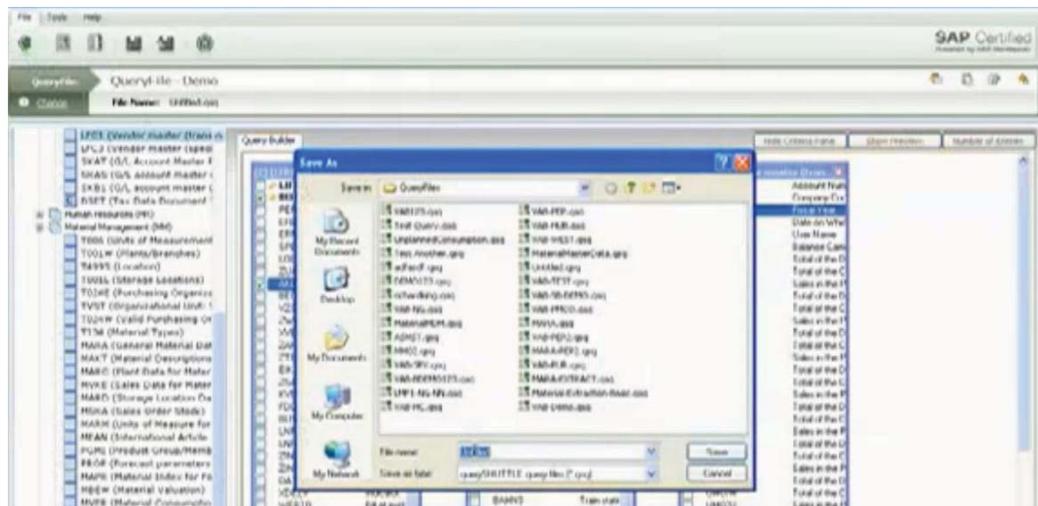
Fuente: elaboración propia, con programa de Winshuttle Query.

Figura 164. Vista preliminar de la consulta



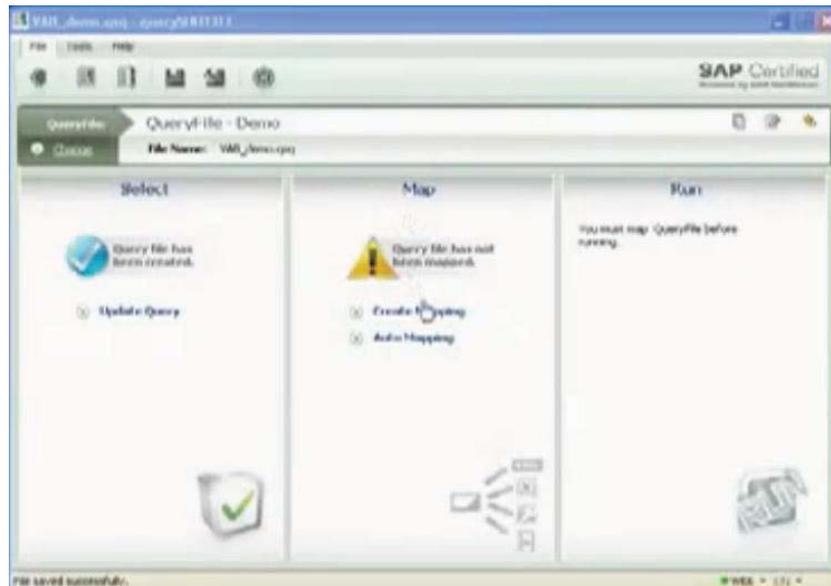
Fuente: elaboración propia, con programa de Winshuttle Query.

Figura 165. Guardar la plantilla de consulta



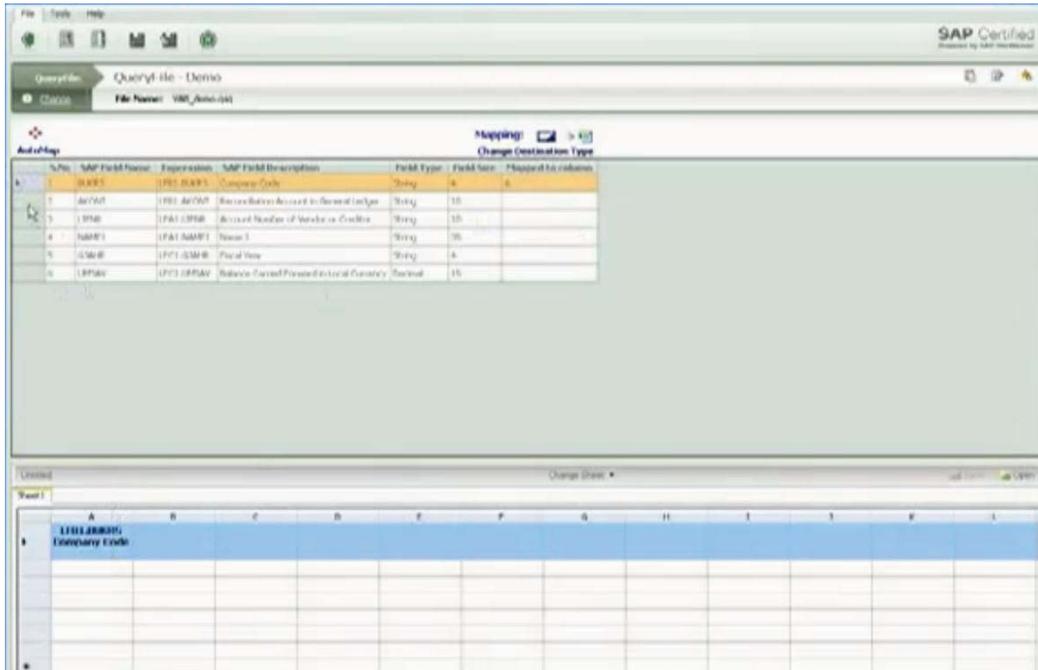
Fuente: elaboración propia, con programa de Winshuttle Query.

Figura 166. **Asignación de los campos seleccionados al archivo de salida**



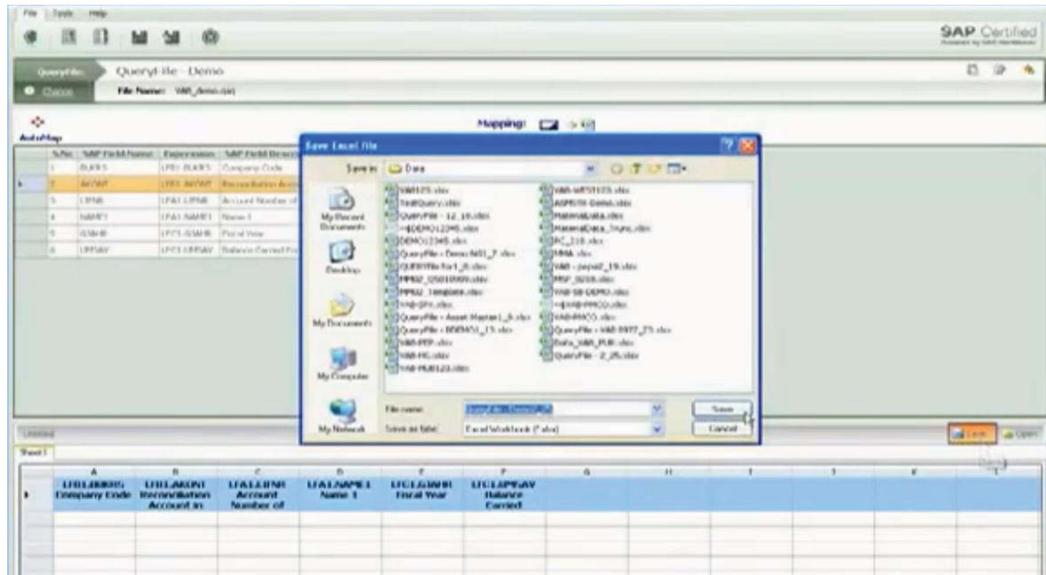
Fuente: elaboración propia, con programa de Winshuttle Query.

Figura 167. Selección de campos de la consulta al archivo de salida de Excel



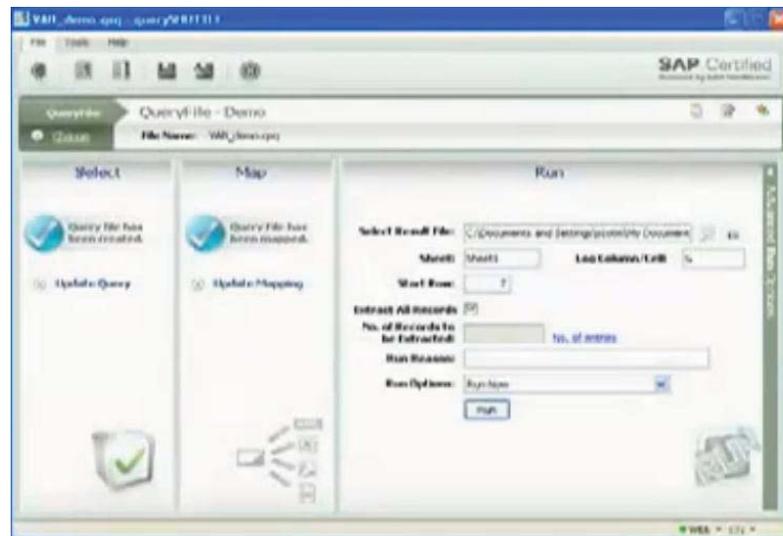
Fuente: elaboración propia, con programa de Winshuttle Query.

Figura 168. Exportación del reporte a un archivo de Excel



Fuente: elaboración propia, con programa de Winshuttle Query.

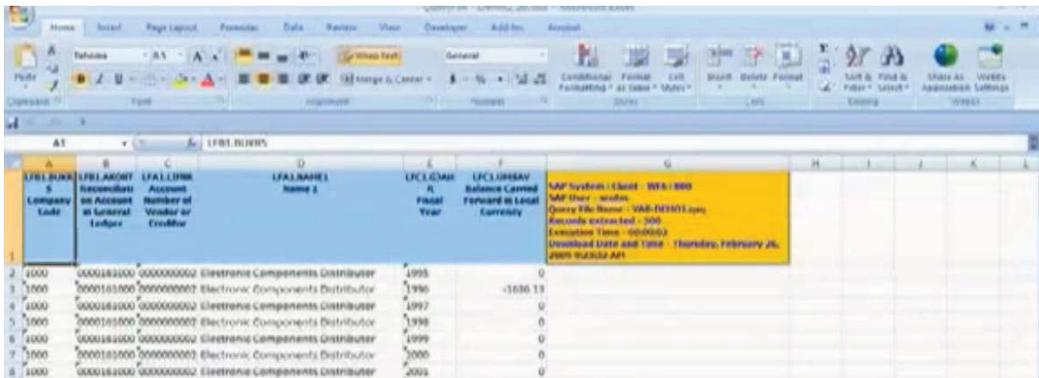
Figura 169. Grabación de archivo de salida Query Shuttle



Fuente: elaboración propia, con programa de Winshuttle Query.

Grabación del archivo de salida, junto con sus propiedades y número de registros a exportar.

Figura 170. Reporte en archivo de Excel



LFB1.BUERS	LFB1.AKONT	LFB1.LINK	LFB1.NAME	LFB1.GRUP	LFB1.SAV	
Company Code	Account	Number of Vendor or Creditor	Name 1	Fiscal Year	Balance Carried Forward in Local Currency	
1000	0000182000	0000000002	Electronic Components Distributor	1995	0	
1000	0000182000	0000000002	Electronic Components Distributor	1996	-1636.13	
1000	0000182000	0000000002	Electronic Components Distributor	1997	0	
1000	0000182000	0000000002	Electronic Components Distributor	1998	0	
1000	0000182000	0000000002	Electronic Components Distributor	1999	0	
1000	0000182000	0000000002	Electronic Components Distributor	2000	0	
1000	0000182000	0000000002	Electronic Components Distributor	2001	0	

SAP System: 1 Client: 001: 800
SAP User: wshuttle
Query File Name: VAR-DE1013.sps
Execution Date: 20090226
Download Date and Time: Thursday, February 26, 2009 10:03:22 AM

Fuente: elaboración propia, con programa de Excel.

Este módulo permite realizar uno de los mayores retos detectado por los usuarios finales de SAP es la falta de un fácil acceso en tiempo real a los datos de SAP en el sistema transaccional.

2.5. Análisis y comparación de ambos métodos Abap Vrs WinShuttle

En esta sección es muy importante tomar en cuenta las diferencias que hay entre ABAP y Winshuttle, ya que se puede tomar mejores decisiones para poder satisfacer las diferentes necesidades que se tengan.

2.5.1. Transaction Shuttle y Abap

A continuación se presenta una tabla de las características y ventajas de utilización de cada método. Para poder realizar un mejor análisis y así poder escoger que método utilizar.

Tabla III. **Características y Ventajas de Transaction Shuttle Vrs Abap**

CARACTERÍSTICAS	VENTAJAS
Asistente para grabar, mapear y ejecutar.	Los usuarios empresariales pueden cargar sus datos en SAP con una formación mínima.
Interface intuitiva de usuario.	Los usuarios que no sean técnicos pueden configurar fácilmente ficheros de TRANSICIÓN.
Modos de ejecución en proceso de fondo, modo de no ejecución en proceso de fondo (Enjoy) o modo GUI.	Diversos modos permiten registrar cualquier transacción de SAP.
Respaldo de seguridad de datos automático.	Antes de realizar modificaciones de datos, automáticamente se realiza un respaldo de seguridad de los datos de SAP existentes que le permite anular las modificaciones, si es necesario.
Bifurcaciones lógicas.	El ajuste a registros de variable permite a un script administrar varios tipos de datos.
Loop.	Crea un proceso repetible que se puede automatizar.
Carga en tiempo real.	Permite a los usuarios tener abierto el fichero de Excel durante el proceso de carga de forma que los resultados de SAP se pueden ver en tiempo real.
Conservación de la seguridad SAP.	Protege las transacciones y los datos del acceso y uso no deseados.
Cadenas.	Si los procesos empresariales requieren la ejecución de varias transacciones en un determinado orden, se podrán enlazar varias cargas.
Gestión de errores.	Los errores de datos se marcan inmediatamente en el fichero de Excel y se pueden corregir y volver a procesar rápidamente.

Fuente: elaboración propia.

2.5.2. Query Shuttle y Abap

A continuación se presenta una tabla de las características y ventajas de utilización de cada método. Para poder realizar un mejor análisis y así poder escoger que método utilizar.

Tabla IV. **Características y Ventajas de Query Shuttle Vrs Abap**

CARACTERÍSTICAS	VENTAJAS
Seguridad conforme a la ley SOX para el acceso a los datos de los usuarios aprovechando los perfiles de seguridad estándar de SAP sin necesidad del acceso SE16/SQ01/SQVI	No solo mantiene la seguridad de SAP en lo que se refiere a las tablas, sino que también mantiene la seguridad en lo que se refiere a la organización en un perfil de seguridad de SAP estándar. VEA EL PROCEDIMIENTO EN ESTE VÍDEO.
Optimización dinámica de los procesos para ejecutar las consultas.	Minimiza el impacto que las consultas de rendimiento puedan tener en el servidor de SAP.
Software fácil de utilizar con funcionalidad de búsqueda exhaustiva en tablas y campos, y con un generador de cláusulas where intuitivo	Capacita a los usuarios empresariales para crear sus propias consultas.
Extracciones en una o varias tablas con combinaciones automáticas. Compatibilidad con tablas transparentes, cluster, combinadas y vistas.	Ofrece flexibilidad y potencia para muchos fines diferentes.
Programador integrado.	Permite la ejecución de consultas sin conexión y la automatización de las extracciones rutinarias.
Mapeación y extracción de datos a Excel, Access, archivos de texto o XML.	Ofrece flexibilidad completa en el formato resultante.

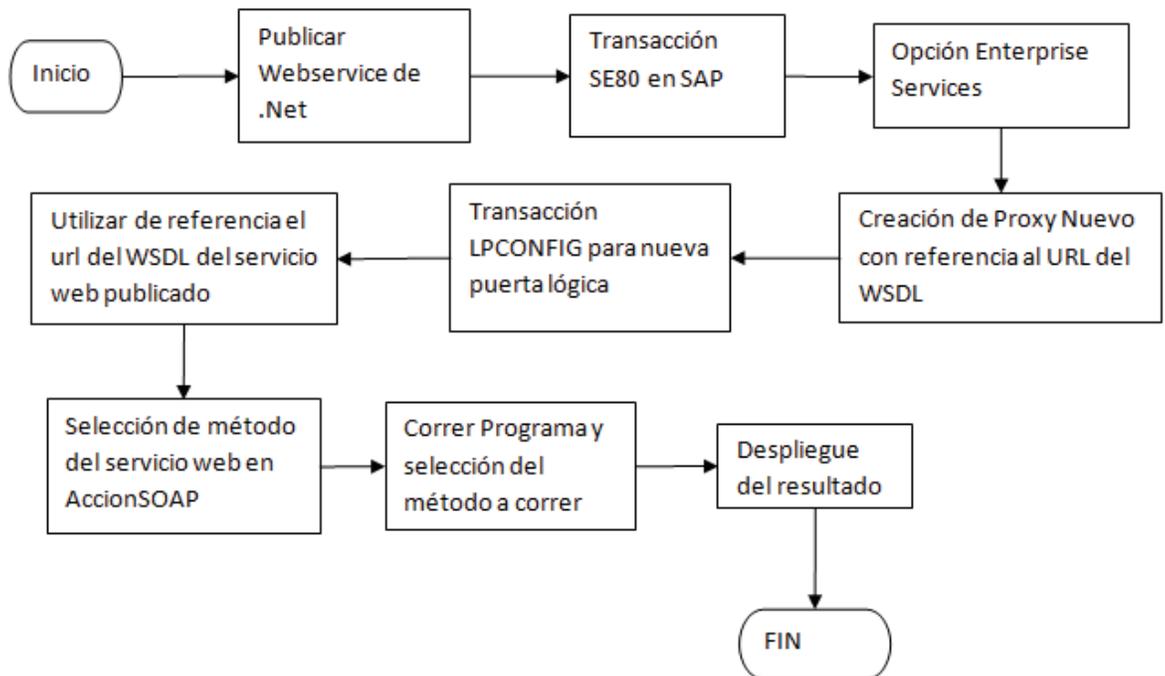
Fuente: elaboración propia.

3. INTERCOMUNICACIÓN ENTRE SISTEMAS EXTERNOS HACIA SAP

3.1. Consumo de Webservice de .Net desde SAP

A continuación se muestra una serie de pasos a seguir para poder consumir un servicio web de cualquier plataforma desde SAP.

Figura 171. Diagrama de flujo de consumo de web service con SAP



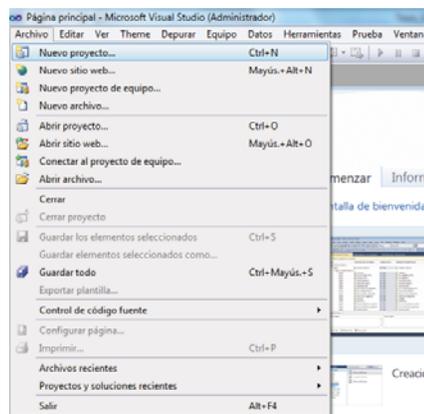
Fuente: elaboración propia.

3.1.1. Ejemplo de consumo de Webservice

En este ejemplo se muestra como consumir un servicio web publicado por medio de la plataforma de Visual Studio C# desde SAP.

Para ello debemos de ingresar a Visual Studio y se escoge nuevo proyecto como se muestra a continuación:

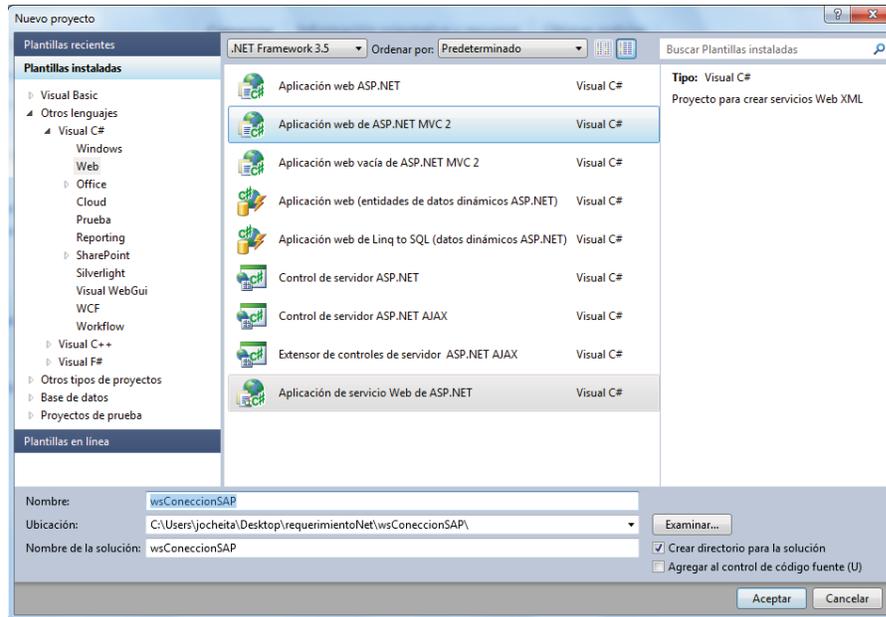
Figura 172. Creación de nuevo proyecto



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Se debe de seleccionar la opción WEB, Aplicación de servicio Web de Asp.Net y luego dar clic en el botón Aceptar.

Figura 173. Creación de Servicio Web



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Por lo que el sistema muestra la ventana de ingreso de código por lo que se puede ingresar el siguiente código:

Figura 174. Código servicio Web AspNet

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

namespace wsConeccionSAP
{
    /// <summary>
    /// Descripción breve de Service1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // Para permitir que se llame a este servicio Web desde un script, usando ASP.NET AJAX, quite
    // [System.Web.Script.Services.ScriptService]
    public class Service1 : System.Web.Services.WebService
    {
        [WebMethod]
        public string Prueba_Comunicacion_A_SAP()
        {
            return "Hola Es una prueba de consumo de servicio WEB desde SAP, Att. Jose Ocheita";
        }
        [WebMethod]
        public string Resta(int a, int b)
        {
            int resultado = 0;
            if (a >= b)
            {
                resultado = a - b;
            }
            else
            {
                resultado = 0;
            }
            return resultado.ToString();
        }
        [WebMethod]
        public string Multiplicacion(int a, int b)
        {
            int resultado = 0;
            resultado = a * b;
            return resultado.ToString();
        }
    }
}
```

Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Eliminar el siguiente código en el web. config, si es que este se genera:

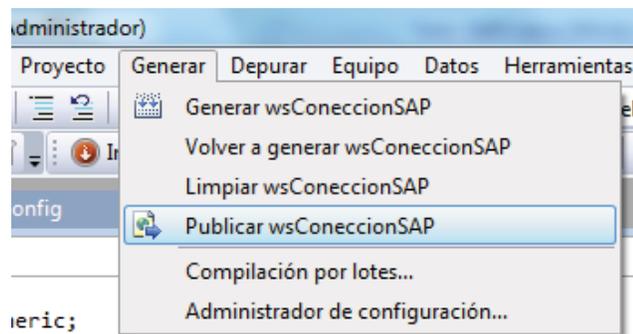
Figura 175. **Deshabilitación de los métodos HttpGet y HttpPost al protocolo Http**

```
...  
<location path="Service1.aspx">  
  <system.web>  
    <webServices>  
      <protocols>  
        <add name="HttpGet"/>  
        <add name="HttpPost"/>  
      </protocols>  
    </webServices>  
  </system.web>  
</location>
```

Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Al publicar el servicio Web:

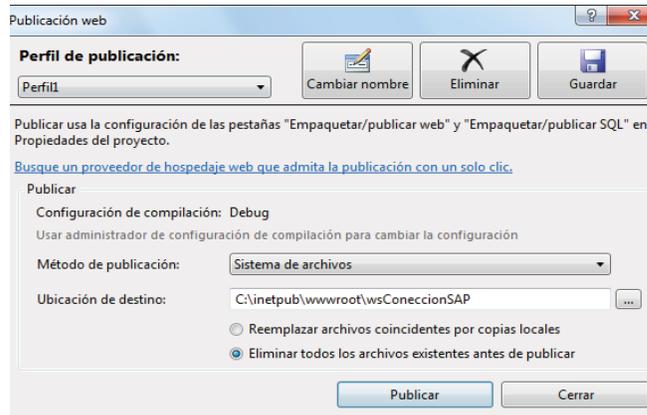
Figura 176. **Publicación del servicio Web**



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

En la opción método de publicación se debe de escoger la opción de publicación en sistema de archivos para publicar el servicio en el localhost:

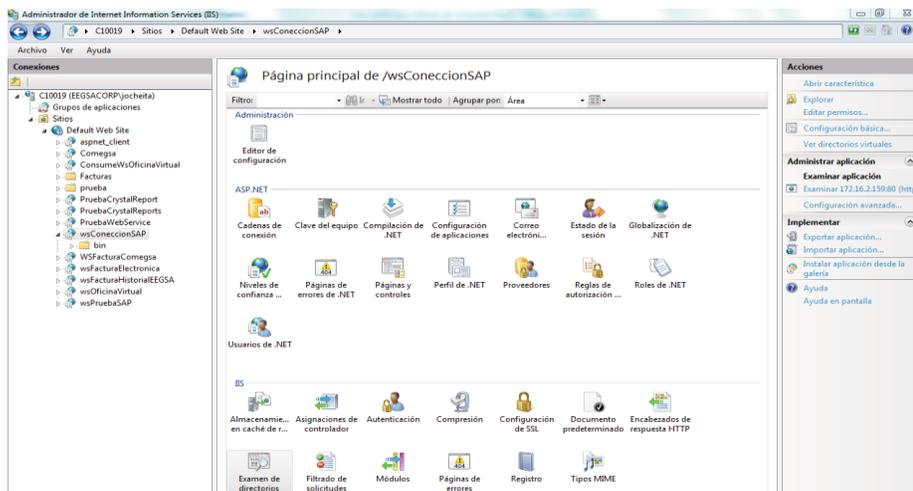
Figura 177. **Publicación del servicio web en sistema de archivos**



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Al ir a Inicio -> Panel de control -> Herramientas Administrativas -> Administrador de Internet Information Server (IIS). Encontramos el servidor Web IIS, como se muestra a continuación:

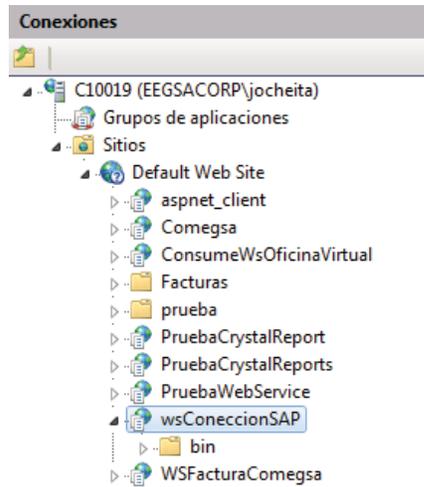
Figura 178. **Publicación del servicio web, utilizando IIS**



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Hay que buscar el servicio web publicado:

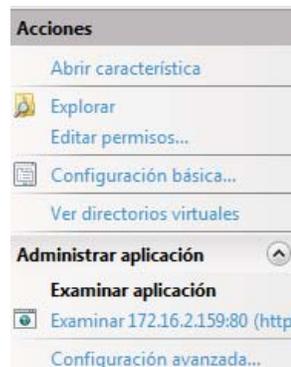
Figura 179. **Servicio web publicado wsConeccionSAP**



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Se debe de dar clic en la pestaña Administrar Aplicación la cual muestra la opción Examinar Aplicación:

Figura 180. **Opción de Examinar Aplicación**



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Por lo que en el navegador se muestra el servicio web publicado como se muestra a continuación:

Figura 181. **Servicio web publicado**



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Al dar clic en [Prueba_Comunicacion_A_SAP](#) el navegador muestra la siguiente página:

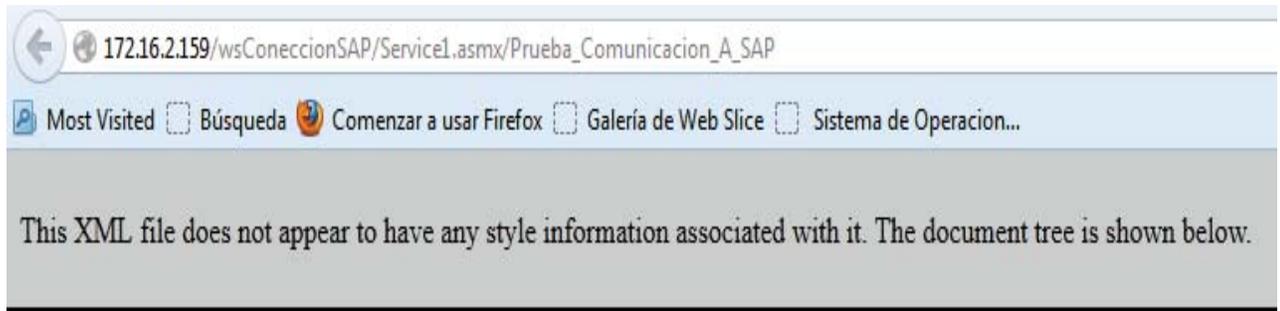
Figura 182. **Método Prueba_Comunicacion_A_SAP**



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Al dar clic en Invocar se muestra el resultado en formato de XML:

Figura 183. **Resultado del método Prueba_Comunicacion_A_SAP**



```
- <string>  
  Hola Es una prueba de consumo de servicio WEB desde SAP, Att. Jose Ocheita  
</string>
```

Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

No hay que olvidar que se debe de ingresar en el servicio web el Namespace correcto (La dirección del servicio web) para que SAP pueda tener referencia directa del mismo.

Figura 184. **Cambio del namespace en el servicio web**

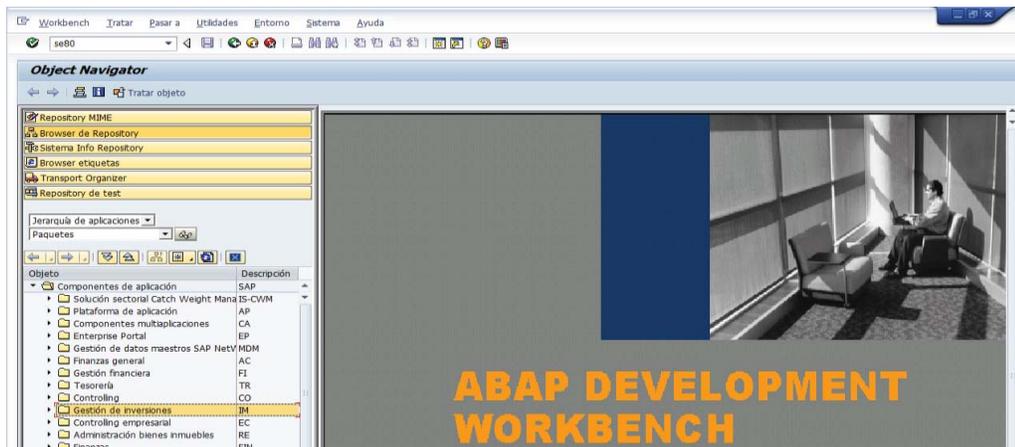
```
/// <summary>
/// Descripción breve de Service1
/// </summary>
[WebService(Namespace = "http://172.16.2.159/wsConeccionSAP/Service1.asmx")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
[System.ComponentModel.ToolboxItem(false)]
// Para permitir que se llame a este servicio Web desde un script, usando ASP.NET AJAX
// [System.Web.Script.Services.ScriptService]
```

Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Luego que está publicado el servicio web debidamente, ahora se va a SAP.

Para ello se debe de ir a la Transacción SE80.

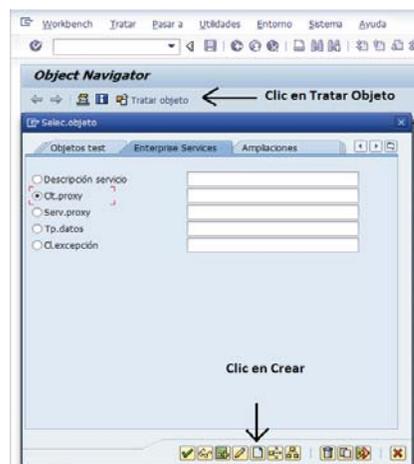
Figura 185. Transacción SE80



Fuente: elaboración propia, con base en SAP R/3.

Se da clic en tratar objeto->Enterprise Services-> Clt. Proxy y luego clic en Crear como se muestra a continuación:

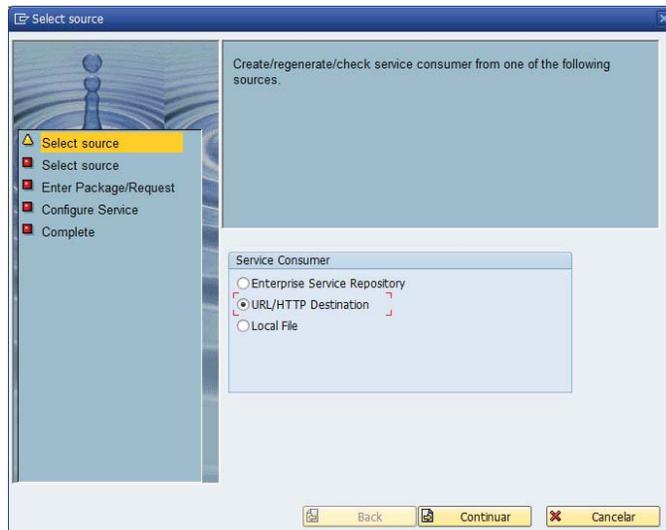
Figura 186. Creación de objeto de cliente proxy para comunicación con un servicio web



Fuente: elaboración propia, con base en SAP R/3.

Muestra la siguiente pantalla, la cual se debe de escoger la opción URL/HTTP Destination. Ya que se tiene la dirección del servicio web publicado.

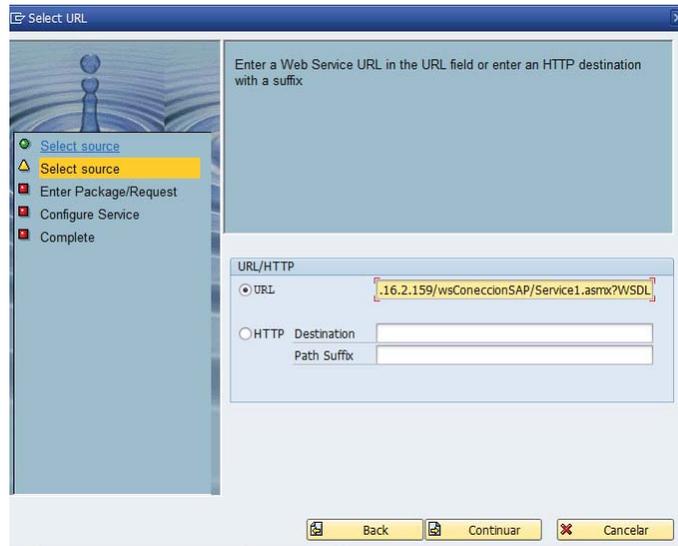
Figura 187. Selección de fuente de Servicio



Fuente: elaboración propia, con base en SAP R/3.

Al darle continuar se debe de ingresar el Link del servicio web junto con la dirección del archivo WSDL (<http://172.16.2.159/wsConeccionSAP/Service1.asmx?WSDL>) y luego darle continuar como se presenta a continuación:

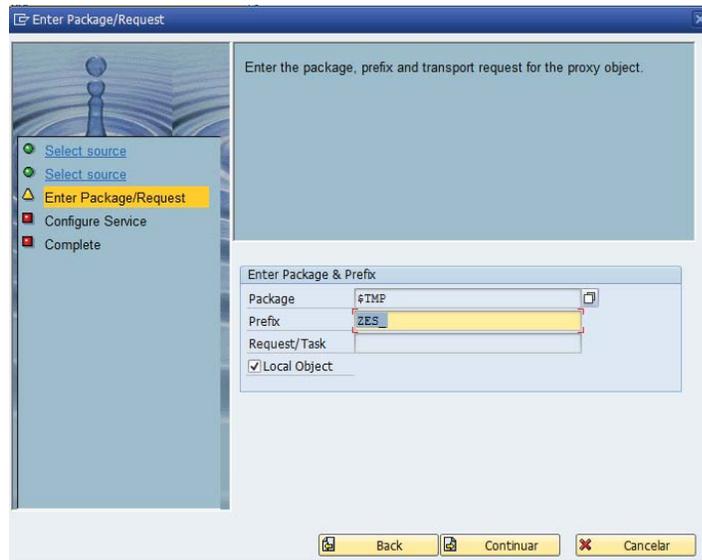
Figura 188. Ingreso del link del WSDL del servicio Web en SAP



Fuente: elaboración propia, con base en SAP R/3.

Luego se debe de introducir los detalles del paquete o simplemente al darle clic en la casilla de objeto local, se debe de introducir un prefijo de "ZES_". El objetivo de ingresar este prefijo es debido a que no debe de haber una duplicación de nombres de objetos cliente proxy en SAP. A continuación, al darle clic en Continuar aparece la siguiente pantalla

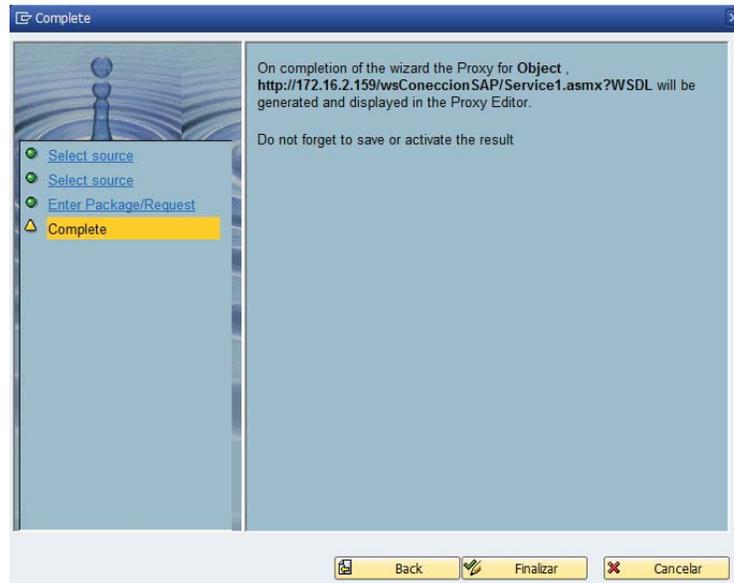
Figura 189. **Pantalla de ingreso de paquete para el cliente proxy**



Fuente: elaboración propia, con base en SAP R/3.

Luego darle Finalizar.

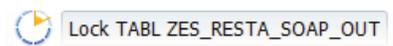
Figura 190. **Pantalla de finalización de creación de proxy**



Fuente: elaboración propia, con base en SAP R/3.

Al darle finalizar SAP crea todas las tablas internas según el archivo WSDL.

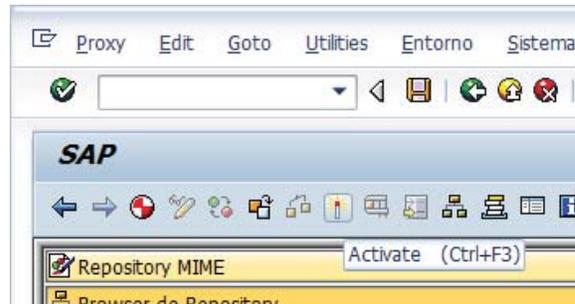
Figura 191. **Creación de tablas internas de SAP, luego de reconocer el servicio web**



Fuente: elaboración propia, con base en SAP R/3.

Al finalizar muestra las propiedades del servicio web en SAP.

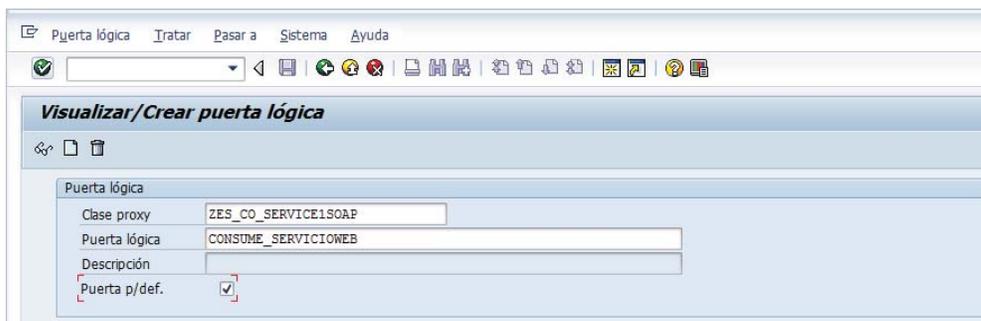
Figura 192. **Guardando y activando el servicio web**



Fuente: elaboración propia, con base en SAP R/3.

Se debe de crear el puerto lógico, utilizando la transacción LPCONFIG. Luego se debe de ingresar el nombre del proxy creado anteriormente en la transacción se80, en este caso es ZE_CO_SERVICE1SOAP, luego se debe de ingresar un nombre adecuado en el campo Puerta Lógica, por lo que le pondremos el nombre de CONSUME_SERVICIOWEB y por último poner cheque en el campo Puerta p/def que significa puerta por defecto. Como se muestra a continuación:

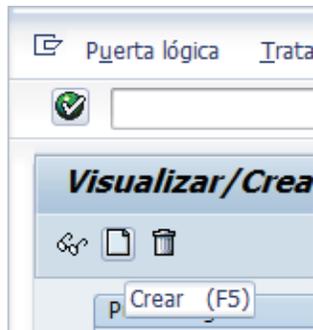
Figura 193. **Creación de puerto lógico en SAP con la transacción LPCONFIG, para consumo de servicio web**



Fuente: elaboración propia, con base en SAP R/3.

Se debe de dar clic en el icono de una hoja de papel o presionar la tecla F5 para crear el puerto lógico.

Figura 194. **Creación del puerto lógico**



Fuente: elaboración propia, con base en SAP R/3.

Luego al crear el puerto lógico muestra la siguiente pantalla. Por lo que hay que ingresar a la pestaña "Parámetros de llamada" seleccionar URL e introduzca la URL del WSDL de nuevo. (<http://172.16.2.159/wsConeccionSAP/Service1.asmx?WSDL>)

Figura 195. Ingreso de parámetros de puerto lógico en SAP

Fuente: elaboración propia, con base en SAP R/3.

Luego hay que dirigirse a la pestaña operaciones. Esta pestaña lo que presenta son los métodos que brinda el Servicio Web.

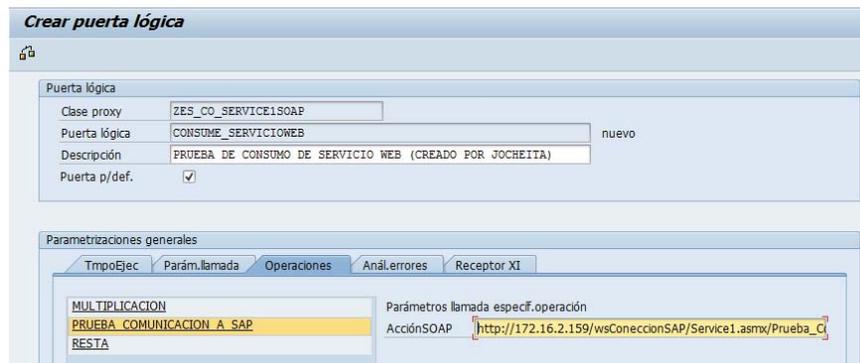
Justo aquí es donde se pone un poco complicado ya que ahora se debe de añadir la referencia del actual código WSDL en cada operación. Básicamente en el WSDL hay una sección que detalla cada operación y especifica una URL soapAction. Simplemente se debe de encontrar esta url soapAction y luego se debe de insertar en la pestaña de operaciones para la operación apropiada.

Figura 196. **SoapAction del método Prueba_Comunicacion_A_SAP**

```
<wsdl:operation name="Prueba_Comunicacion_A_SAP">  
<soap:operation soapAction="http://172.16.2.159/wsConeccionSAP/Service1.asmx/Prueba_Comunicacion_A_SAP" style="document"/>
```

Fuente: elaboración propia, con base en WSDL.

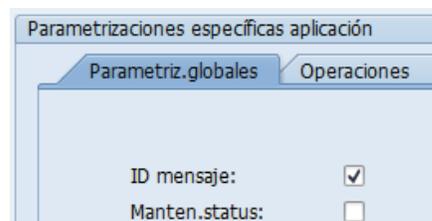
Figura 197. **Definición del método Prueba_Comunicacion_A_SAP en la puerta lógica**



Fuente: elaboración propia, con base en SAP R/3.

En parametrizaciones globales si no se le da cheque a mantenimiento de status como se muestra a continuación:

Figura 198. **Parametrizaciones globales, mantenimiento de status**



Fuente: elaboración propia, con base en SAP R/3.

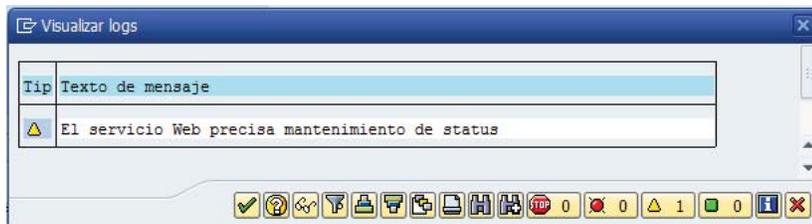
Al grabar y activar muestra los siguientes mensajes:

Figura 199. **Mensaje de información de parametrización global en Ipconfig**



Fuente: elaboración propia, con base en SAP R/3.

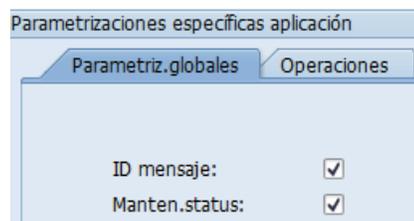
Figura 200. **Mensaje de mantenimiento de status en Ipconfig**



Fuente: elaboración propia, con base en SAP R/3.

Por lo que es necesario darle clic y checar la opción:

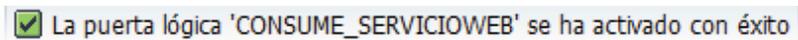
Figura 201. **Cheque de mantenimiento de status en Ipconfig**



Fuente: elaboración propia, con base en SAP R/3.

Al guardar y ejecutar muestra el siguiente mensaje en la barra de estado.

Figura 202. **Mensaje de éxito de grabación y activación del puerto lógico**



La puerta lógica 'CONSUME_SERVICIOWEB' se ha activado con éxito

Fuente: elaboración propia, con base en SAP R/3.

Ya que se activó y se guardó el puerto lógico hay que regresar a la transacción SE80, luego hay que presionar el botón de prueba y escribir el nombre del puerto lógico el cual es CONSUME_SERVICIOWEB. Como se muestra a continuación:

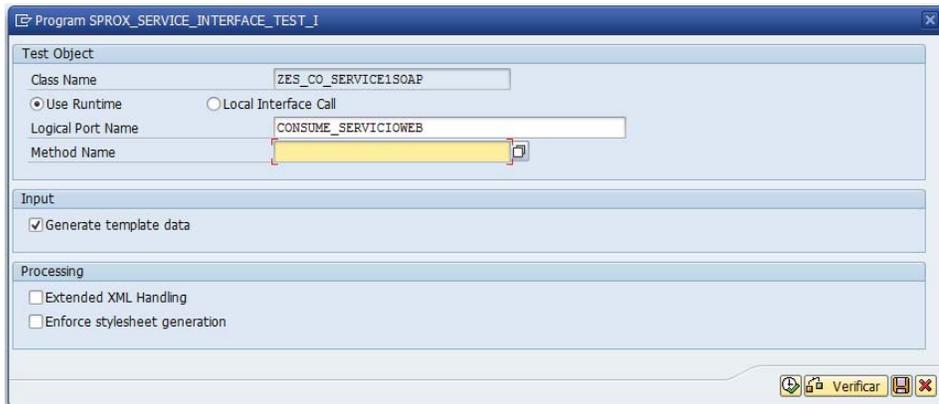
Figura 203. **Prueba del servicio Web, presionando el botón F8**



Fuente: elaboración propia, con base en SAP R/3.

Se ingresa el nombre del puerto lógico y el nombre del método, como se muestra a continuación:

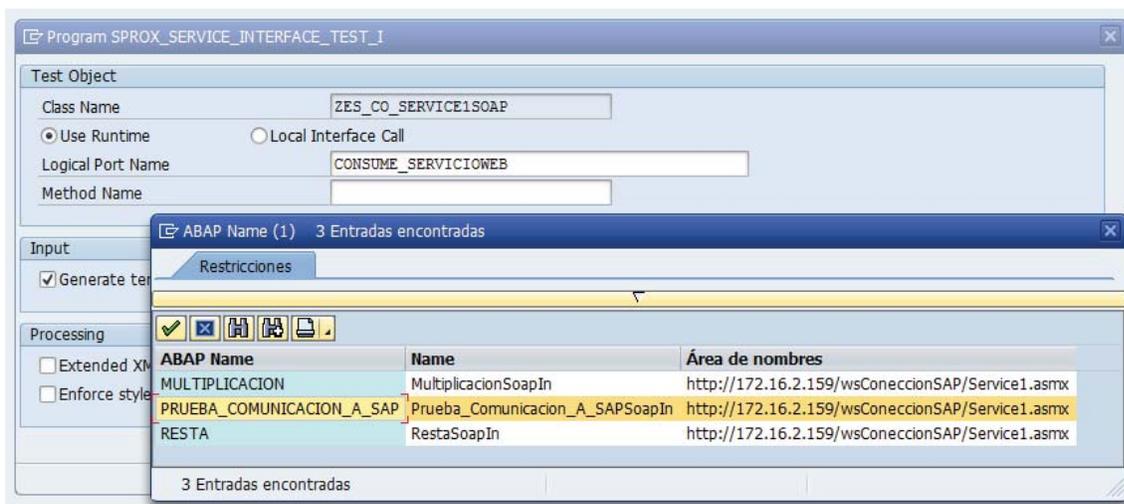
Figura 204. Ingreso del nombre del puerto lógico



Fuente: elaboración propia, con base en SAP R/3.

Se debe de buscar el nombre del método posicionándonos en Nombre del método (Method Name) y presionando F4 como se muestra a continuación:

Figura 205. Búsqueda del método a ejecutar



Fuente: elaboración propia, con base en SAP R/3.

Luego de seleccionarlo hay que presionar la tecla F8 para ejecutar por lo que muestra la siguiente salida:

Figura 206. **Salida de la ejecución del método del Servicio Web, ejecución del SOAP ACTION**



Fuente: elaboración propia, con base en SAP R/3.

Se vuelve a ejecutar presionando la tecla F8 y muestra el resultado del método, en este caso como el método no pide parámetros entonces sale el resultado de una vez.

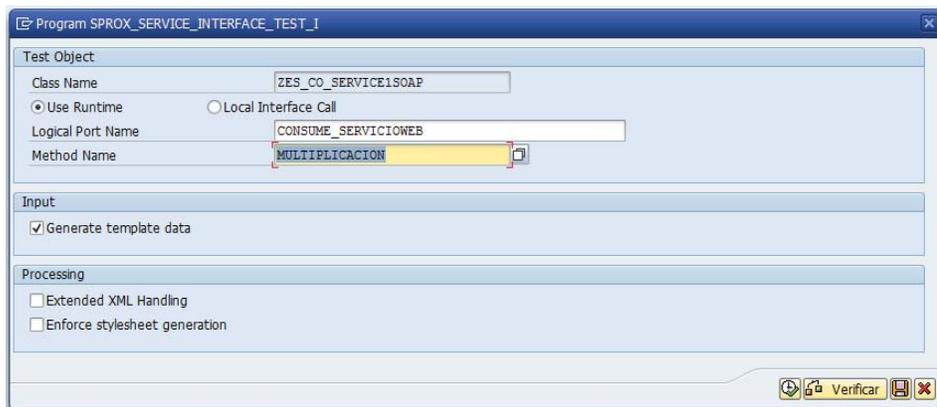
Figura 207. **Resultado de ejecución del método del servicio web**



Fuente: elaboración propia, con base en SAP R/3.

Ahora si en la ejecución se manda a llamar un método del servicio web el cual pide parámetros entonces solo hay que utilizar el editor XML de SAP y modificar los parámetros. Por ejemplo llamando al método multiplicación.

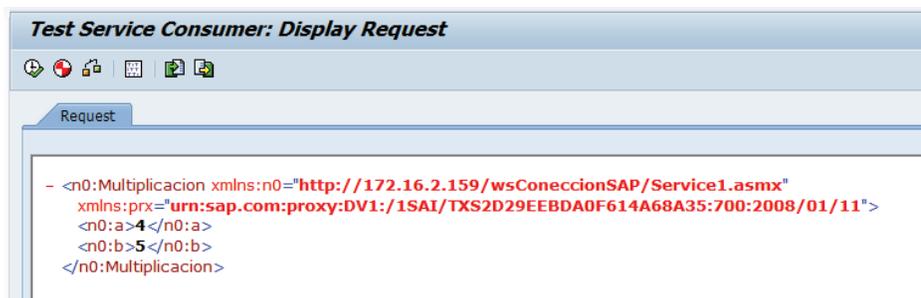
Figura 208. **Método de multiplicación, pidiendo parámetros de entrada**



Fuente: elaboración propia, con base en SAP R/3.

Al ejecutar con la tecla F8 y el sistema muestra el siguiente resultado:

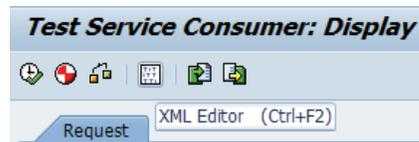
Figura 209. **Resultado de ejecución del método de multiplicación**



Fuente: elaboración propia, con base en SAP R/3.

Como se ve en la figura anterior están las etiquetas `<n0:a>4</n0:a>` y `<n0:b>5</n0:b>` , estas significan los parámetros a enviar al método, por lo que para modificarlos hay que presionar la combinación de teclas CTRL + F2.

Figura 210. **Editor XML para cambiar parámetros del archivo a enviar al servicio web**



Fuente: elaboración propia, con base en SAP R/3.

Se tienen que cambiar los parámetros:

Figura 211. **Cambio de parámetros para la llamada del método del servicio web**



Fuente: elaboración propia, con base en SAP R/3.

Luego hay que presionar la tecla F8 para ejecutar, entonces el sistema muestra el siguiente resultado de la multiplicación de $10 * 15$:

Figura 212. Resultado de la llamada del método con parámetros

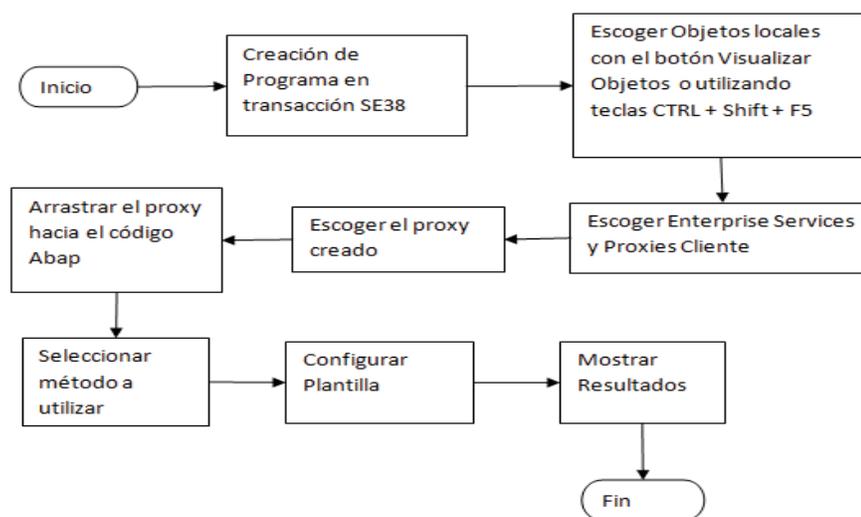


Fuente: elaboración propia, con base en SAP R/3.

3.1.2. Consumo de webservice desde un programa ABAP

En esta sección se presenta como consumir un servicio web creado en Visual Studio .Net C# desde SAP R/3 utilizando ABAP. Se debe tomar en cuenta exactamente los pasos que se describen a continuación para que el consumo sea óptimo.

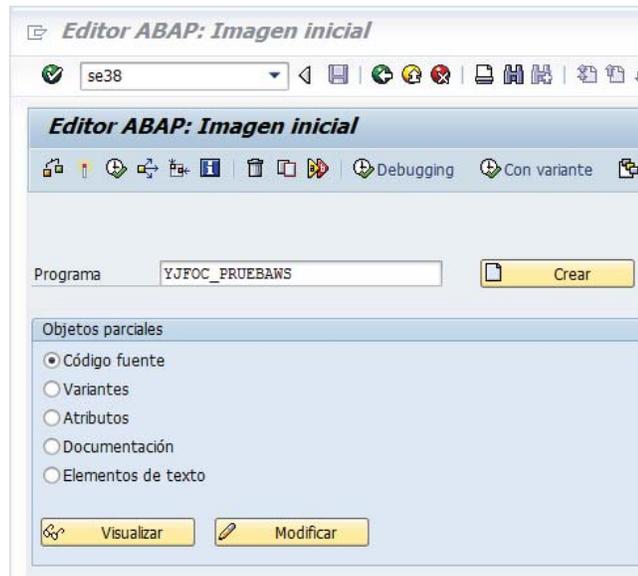
Figura 213. Diagrama de flujo de consumo de webservice con ABAP



Fuente: elaboración propia.

De primero hay que crear el programa utilizando la transacción SE38 y luego se le ingresa el nombre de yjfoc_pruebaws

Figura 214. **Creación de programa para consumo de servicio web**



Fuente: elaboración propia, con base en SAP R/3.

Luego al dar clic en el botón de Crear y se despliega la siguiente pantalla, por lo que hay que ingresarle la siguiente información:

Figura 215. **Parámetros para creación del programa de consumo de servicio web**

ABAP: Propiedades de programa YJFOC_PRUEBAWS modificar

Título Prueba de Consumo de Web Service (Att. Jose Ocheita)

Idioma maestro ES Español

Creado por 19.02.2013 JOCHEITA

Última modif. []

Status Nuevo(revisado)

Atributos

Tipo Programa ejecutable

Status Programa de test

Aplicación []

Grupo autorizaciones []

Base de datos lógica []

Vers.imagen selección []

Bloqueo de editor Cálculo de coma fija

Verif.unicode activas Iniciar mediante variante

Grabar

Fuente: elaboración propia, con base en SAP R/3.

Hay que darle clic en el botón grabar y luego se presenta la siguiente ventana y hay que darle clic en objeto local:

Figura 216. **Grabación del paquete y objeto local**

Crear entrada en catálogo de objetos

Objeto R3TR PROG YJFOC_PRUEBAWS

Atributos

Paquete \$TMP

Responsable JOCHEITA

Sistema original DVI

Idioma maestro ES Español

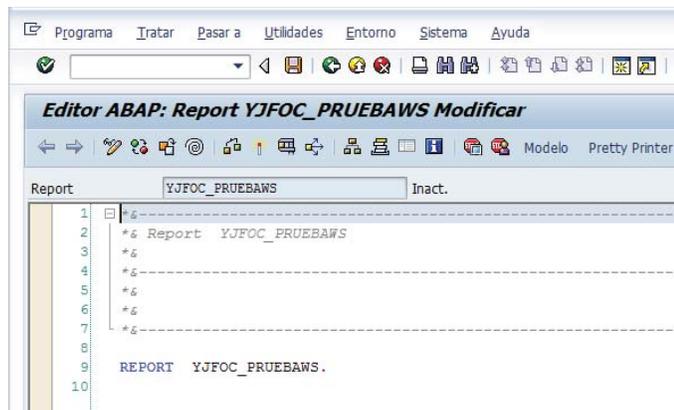
Objeto local Resumen bloqueos

Fuente: elaboración propia, con base en SAP R/3.

Ingreso de paquete y grabación como objeto local, del programa de consumo de servicio web.

Luego muestra el editor de abap.

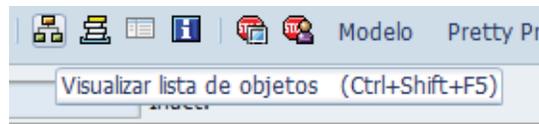
Figura 217. **Editor de Abap, para la programación del consumo del servicio web**



Fuente: elaboración propia, con base en SAP R/3.

Luego presionar la combinación de teclas CTRL + Shift + F5 o al dar clic en el siguiente icono:

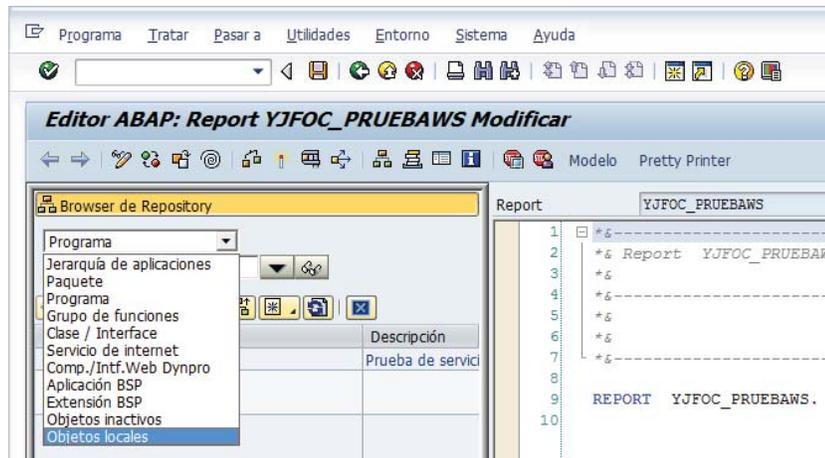
Figura 218. **Icono para ver el listado de objetos locales**



Fuente: elaboración propia, con base en SAP R/3.

Y presenta el siguiente menú, debemos de escoger Objetos Locales como se muestra a continuación:

Figura 219. Listado de objetos locales



Fuente: elaboración propia, con base en SAP R/3.

Y luego a dar clic en el icono de unos lentes el cual significa Visualizar:

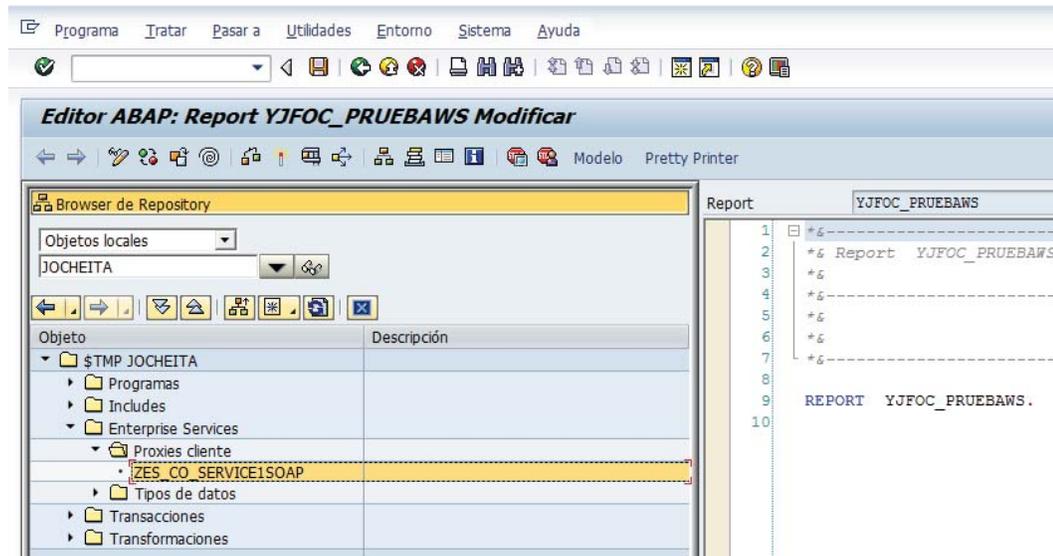
Figura 220. Botón para visualizar el listado de objetos que pertenecen al paquete



Fuente: elaboración propia, con base en SAP R/3.

Y muestra el siguiente menú, del cual escogemos Enterprise Services -> Proxies cliente.

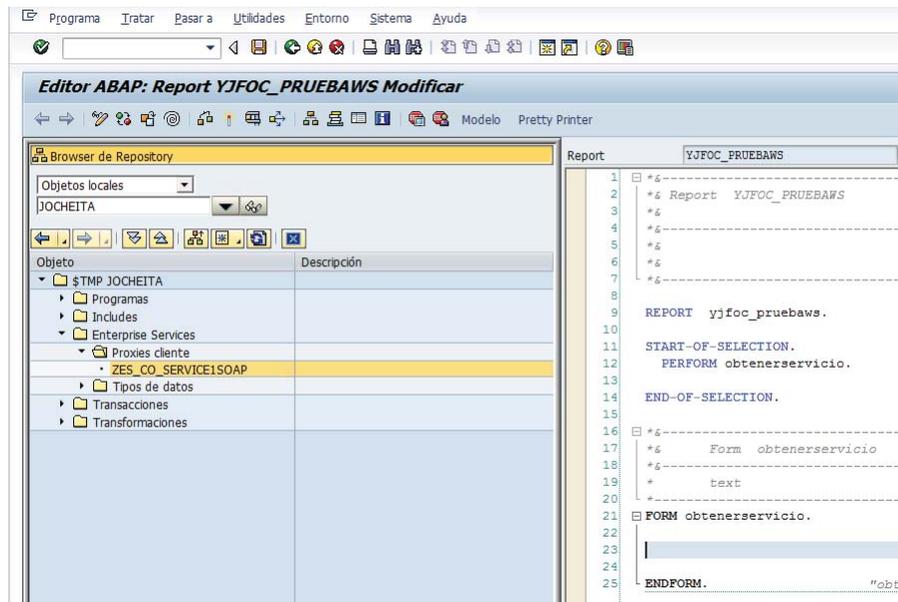
Figura 221. Selección del Enterprise Services en SAP



Fuente: elaboración propia, con base en SAP R/3.

Se ingresa el siguiente código el cual llama a un procedimiento llamado obtenerservicio.

Figura 222. Código para la llamada del servicio web

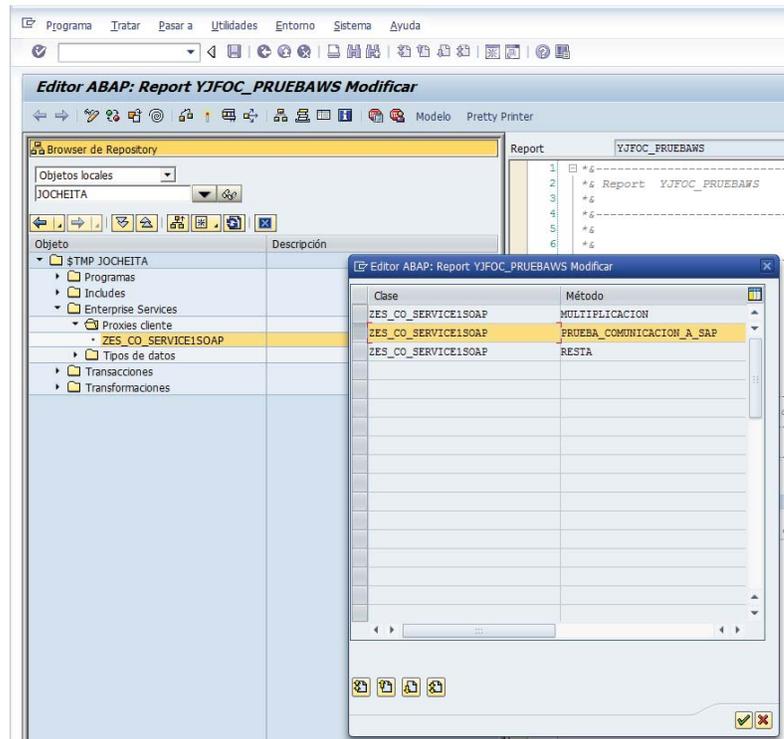


Fuente: elaboración propia, con base en SAP R/3.

Dentro del procedimiento obtener servicio se debe de arrastrar el proxy client ZES_CO_SERVICE1SOAP desde el listado de objetos locales hacia el código abap. El cual generara una plantilla para poder realizar la llamada al método deseado del servicio web.

Al soltar el botón del ratón, SAP mostrará los métodos que dispone el servicio web para luego realizar la llamada.

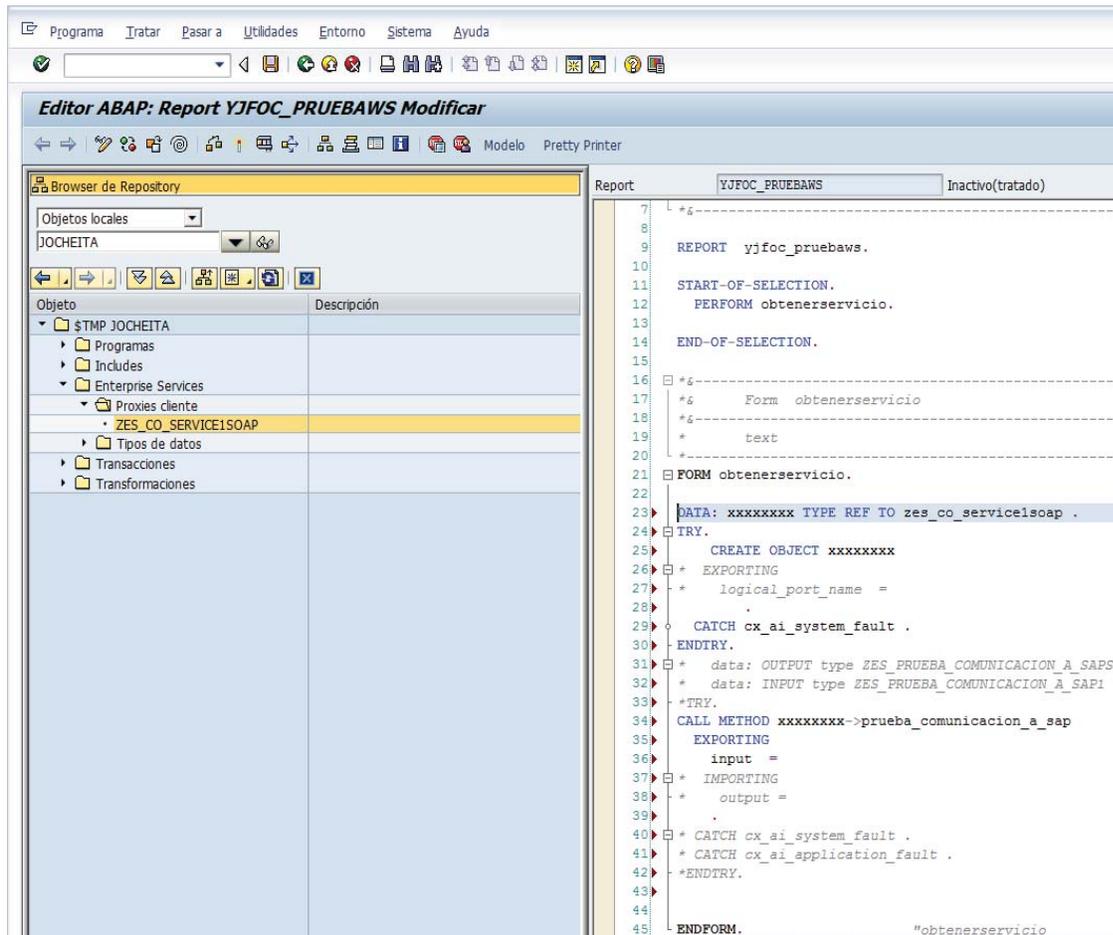
Figura 223. **Llamada al método PRUEBA_COMINICACION_A_SAP del servicio web**



Fuente: elaboración propia, con base en SAP R/3.

Al dar clic en el botón del cheque verde muestra la plantilla de programación, como se muestra a continuación:

Figura 224. Plantilla de programación ABAP, de la llamada del método del servicio web



Fuente: elaboración propia, con base en SAP R/3.

Entonces solo hay que eliminar los comentarios y se pueden realizar los cambios necesarios en el procedimiento obtiene servicio como se muestra a continuación:

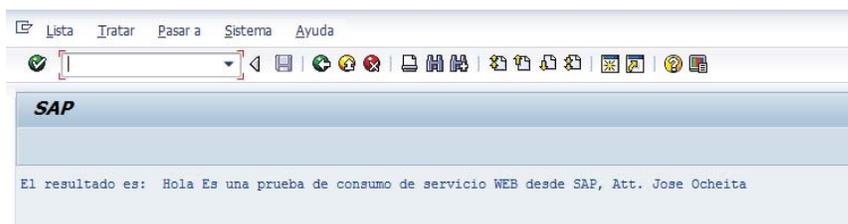
Figura 225. **Código de la llamada del método PRUEBA_COMUNICACION_A_SAP**

```
FORM obtenerservicio.  
  
DATA: proxy TYPE REF TO zes_co_servicelsoap .  
TRY.  
    CREATE OBJECT proxy  
    EXPORTING  
        logical_port_name = 'CONSUME_SERVICIOWEB'.  
    CATCH cx_ai_system_fault .  
ENDTRY.  
  
DATA: output TYPE zes_prueba_comunicacion_a_saps .  
DATA: input TYPE zes_prueba_comunicacion_a_sapl .  
  
TRY.  
    CALL METHOD proxy->prueba_comunicacion_a_sap  
    EXPORTING  
        input = input  
    IMPORTING  
        output = output.  
    CATCH cx_ai_system_fault .  
    CATCH cx_ai_application_fault .  
ENDTRY.  
  
WRITE: / 'El resultado es: ', output-prueba_comunicacion_a_sapresul.  
  
ENDFORM.                "Obtienservicio
```

Fuente: elaboración propia, con base en SAP R/3.

Luego al Salvar, Activar y Correr el programa muestra la siguiente salida.

Figura 226. **Resultado de la llamada del método PRUEBA_COMUNICACION_A_SAP**



Fuente: elaboración propia, con base en SAP R/3.

Ahora para a implementar el procedimiento de multiplicación, el cual recibe como parámetros 2 valores y devuelve el resultado. Se debe de realizar el mismo procedimiento para implementar el método Multiplicación del servicio web, pero esta vez tendrá parámetros para que el usuario pueda ingresar los valores.

Se debe de crear el procedimiento multiplicación como se muestra a continuación:

Figura 227. **Código de llamada del método MULTIPLICACION del servicio web**

```
FORM multiplicacion USING valor1 TYPE string
                        valor2 type string.

DATA: proxy TYPE REF TO zes_co_servicisoap .
TRY.
  CREATE OBJECT proxy
  EXPORTING
    logical_port_name = 'CONSUME_SERVICIOWEB'.
  CATCH cx_ai_system_fault .
ENDTRY.
DATA: output TYPE zes_multiplicacion_soap_out .
DATA: input TYPE zes_multiplicacion_soap_in .

"Ingresos de parametros para multiplicacion
input-a = valor1. |
input-b = valor2.

TRY.
  CALL METHOD proxy->multiplicacion
  EXPORTING
    input = input
  IMPORTING
    output = output
  .
  CATCH cx_ai_system_fault .
  CATCH cx_ai_application_fault .
ENDTRY.
WRITE: / 'El resultado es de la multiplicacion es: ', output-Multiplicacion_Result.

ENDFORM.                    "multiplicacion
```

Fuente: elaboración propia, con base en SAP R/3.

Luego se ingresa el siguiente código de inicio de la selección:

Figura 228. **Código de inicio de selección del programa**

```
START-OF-SELECTION.
  perform multiplicacion using valor1 valor2.
END-OF-SELECTION.
```

Fuente: elaboración propia, con base en SAP R/3.

Finalmente hay que salvar, compilar, activar y ejecutar el programa. Por lo que el sistema muestra la siguiente pantalla a continuación:

Figura 229. **Ejecución del programa**

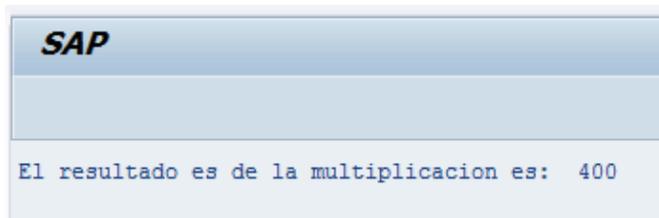


The screenshot shows a SAP input form with a title bar containing a refresh icon. Below the title bar is a header area with the text "Ingrese los valores a operar". Underneath, there are two input fields: "VALOR1" with the value "200" and "VALOR2" with the value "2". The "VALOR2" field is highlighted in yellow.

Fuente: elaboración propia, con base en SAP R/3.

Luego se presiona la tecla F8 o simplemente dar clic en el botón del reloj, por lo que el sistema realiza la operación y muestra el resultado de la multiplicación.

Figura 230. **Resultado de la multiplicación, utilizando servicio web**

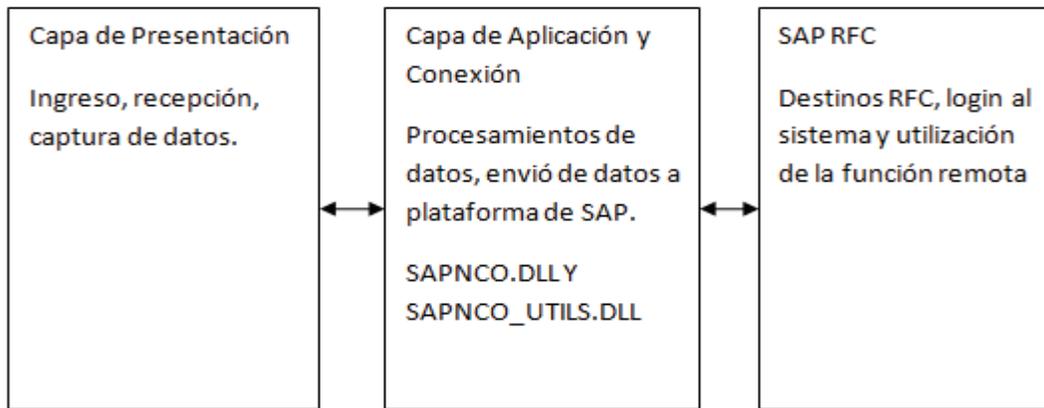


Fuente: elaboración propia, con base en SAP R/3.

3.2. **Remote Function Call (RFC)**

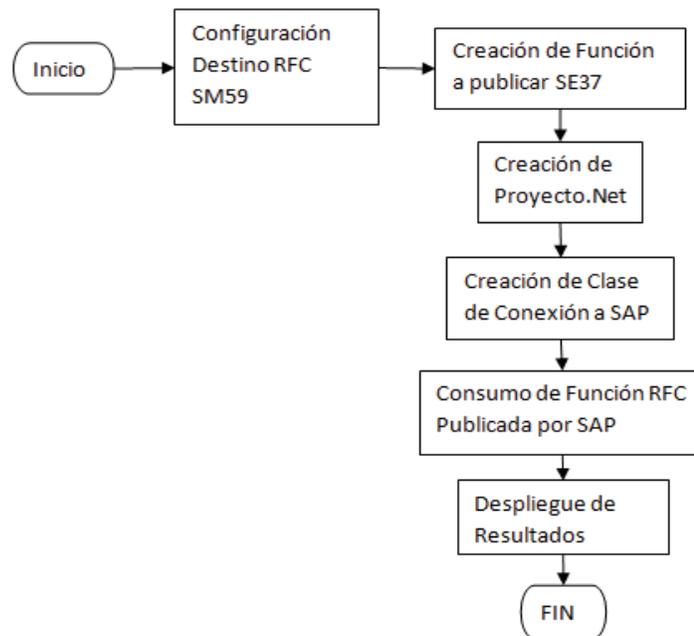
La comunicación entre aplicaciones en sistemas diferentes en el entorno SAP incluye las conexiones entre los sistemas SAP, así como entre los sistemas de sistemas SAP y no-SAP. Remote Function Call (RFC) es la interfaz estándar de SAP para la comunicación entre sistemas SAP. RFC llama a una función que se ejecuta en un sistema remoto.

Figura 231. Diagrama de procesos de funcionamiento de RFC



Fuente: elaboración propia.

Figura 232. Diagrama de flujo para consumir una RFC



Fuente: elaboración propia.

3.2.1. Configuración de destinos RFC despliegue y mantenimiento

En esta sección se presenta como configurar un destino RFC por medio de la transacción SM59, para poder dar acceso seguro desde cualquier programa realizado en Visual Studio .Net C#. Se debe de tomar en cuenta exactamente los pasos que se describen a continuación para que el funcionamiento sea óptimo.

3.2.1.1. Transacción SM59

Los destinos RFC son creados para definir una comunicación física a un destino remoto. Para poder configurar un destino RFC se deben de seguir los siguientes pasos:

Ingresar la transacción SM59.

Figura 233. Pantalla de configuración de conexiones RFC



Fuente: elaboración propia, con base en SAP R/3.

Dar clic en Conexiones ABAP y presionar el botón crear:

Figura 234. Creación de Conexión ABAP



Fuente: elaboración propia, con base en SAP R/3.

Presenta la siguiente pantalla se debe de ingresar Destino RFC, Descripción. En la ficha Opciones Técnicas en distribución de carga se debe dar clic en Distribución de Carga en NO, en máquina Destino se debe ingresar el nombre del Host a conectar y en almacenamiento de base de datos se debe de dar clic en Nom.Host y luego escribir el nombre del mismo como se presenta a continuación:

Figura 235. Pantalla de ingreso de Opciones Técnicas del RFC Destino

The screenshot displays the SAP R/3 configuration interface for a remote system destination. The window title is 'Destino RFC TesisOcheita'. The main area is divided into several sections:

- Header:** 'Destino RFC TesisOcheita' with sub-tabs: 'Entrada al sistema remota', 'Test de conexión', and 'Test unicode'.
- Basic Information:** 'Destino RFC' (TesisOcheita), 'Tipo conexión' (3), 'Conexión ABAP', and 'Descripción'.
- Descriptions:** Three text boxes for 'Descripción 1' (Prueba Conexion .Net a SAP y Viceversa), 'Descripción 2', and 'Descripción 3'.
- Navigation Tabs:** 'Info gestión', 'Opciones técnicas' (selected), 'Entr.sist./Seguridad', 'MDMP y unicode', and 'Opc.especiales'.
- Parametrizaciones sistema destino:**
 - 'Status última distribución' with 'Distrib.carga' set to 'No'.
 - 'Máquina destino' (sapdesarrollo) and 'Nº sistema' (00).
 - 'Clase de almacenamiento en base de datos' with 'Grabar como' set to 'Nom.host' and the value 'sapdesarrollo'.
- Opciones gateway:** Fields for 'Host gateway' and 'Servicio TCP', with a 'Borrar' button.

Fuente: elaboración propia, con base en SAP R/3.

Luego dar clic en la ficha Entr.sist/Seguridad, en la cual se debe de ingresar en Sist.trusted NO y luego en la sección Entrada al sistema se ingresa el idioma, mandante, el nombre del usuario a conectarse y clave de acceso, como se muestra a continuación:

Figura 236. Pantalla de ingreso de seguridad del destino RFC

Destino RFC TesisOcheita

Entrada al sistema remota Test de conexión Test unicode

Destino RFC: TesisOcheita

Tipo conexión: 3 Conexión ABAP Descripción

Descripción

Descripción 1: Prueba Conexion .Net a SAP y Viceversa

Descripción 2:

Descripción 3:

Info gestión Opciones técnicas **Entr.sist./Seguridad** MDMP y unicode Opc.especiales

Opciones de seguridad

Sistema trusted/Status imagen inicial

Sist.trusted: No Sí Imagen acceso

Status del log seguro

inact. activ.

Autorización p.destino:

Entrada al sistema

Idioma: ES

Mandante: 300

Usuario: PRUEBA Usuario actual

Status PW: Es inicial

Clv.acceso: *****

Fuente: elaboración propia, con base en SAP R/3.

En las demás fichas, son valores predeterminados que utiliza SAP para la conexión por lo que es muy recomendable dejarlo como lo tiene configurado. Por lo tanto solo hay que grabar el Destino RFC presionando la combinación de teclas CTRL + S o dándole clic en Guardar.

Luego dar clic en Test de Conexión:

Figura 237. **Prueba de conexión de Destino RFC**



Fuente: elaboración propia, con base en SAP R/3.

Luego el sistema muestra un log de la conexión como se muestra a continuación:

Figura 238. **Log de prueba de conexión de destino RFC**

The screenshot shows a log window titled "RFC - Prueba conexión". It contains a table with two columns: "Acción" and "Result.". The table lists several actions and their corresponding results.

Acción	Result.
EntrSist.	20 mseg
Transferencia 0 Kbytes	1 mseg
Transferencia 10 Kbytes	1 mseg
Transferencia 20 Kbytes	1 mseg
Transferencia 30 Kbytes	1 mseg

Fuente: elaboración propia, con base en SAP R/3.

3.2.1.2. Creación de función en Transacción SE37

Se debe de ir a la transacción Se37:

Figura 239. **Transacción SE37**



Fuente: elaboración propia, con base en SAP R/3.

En módulo de funciones se ingresa el nombre de la función en este caso YRFC_PRUEBA_CONEXION y luego dar clic en el botón Crear. Hay que ingresar el módulo de funciones, grupo de funciones existente y un texto breve:

Figura 240. **Creación de función**



Fuente: elaboración propia, con base en SAP R/3.

Luego muestra la plantilla para crear la función, solo que en la pestaña Atributos y en la sección Forma Ejec. Se debe dar clic en la opción Modulo de Acceso Remoto, como se muestra a continuación.

Figura 241. **Atributos de la función de acceso remoto**

The screenshot shows the SAP configuration interface for the function 'YRFC_PRUEBA_CONEXION'. The title bar reads 'Bibl.funciones: YRFC_PRUEBA_CONEXION modificar'. Below the title bar is a toolbar with various icons. The main area is divided into several sections:

- Módulo funciones:** YRFC_PRUEBA_CONEXION, Inactivo
- Navigation tabs:** Atributos (selected), Import, Export, Modif., Tablas, Excep., Cód.fte.
- Clasificación:**
 - Grupo de funciones: ZGRFC (Grupo de Funciones RFC)
 - Texto breve: Prueba conexion
- Forma ejec. (Execution Mode):**
 - Módulo de funciones normal
 - Módulo de acceso remoto
 - Módulo actualización
 - Inicio inmed.
 - Inic.inmed.-no actual.posteriorm.
 - Inicio retard.
 - Ejec.colectiva
- Datos generales (General Data):**
 - Responsable: JOCHEITA
 - Modificado por: JOCHEITA
 - Fecha modif.: 28.02.2013
 - Paquete: ZBC_002
 - Programa: SAPLZGRFC
 - Include: LZGRFCU03
 - Idioma maestro: ES
 - No liberado
 - Bloqueo editor
 - Global

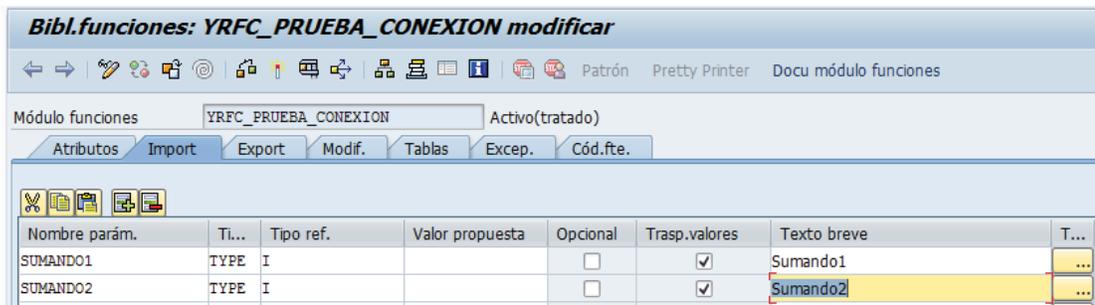
Fuente: elaboración propia, con base en SAP R/3.

La interface de creación de funciones de SAP está definida por:

- Parámetros de Entrada (Import Parameters): contiene datos a ser pasados desde el programa llamante a la función de SAP.
- Parámetros de Salida (Export Parameters): contiene los datos que la función pasa al programa que invoca.
- Parámetros de Entrada/Salida (Changing Parameters): contiene la referencia del dato del programa llamante de la función. (Solo es soportado por algunos conectores)
- Tablas de Entrada/Salida (Import/Export Table Parameters): contiene las tablas internas de la función.

Por lo que en la pestaña Import se puede ingresar los parámetros de la función como se muestra a continuación:

Figura 242. Ingreso de parámetros en la función



Fuente: elaboración propia, con base en SAP R/3.

En la pestaña de Export se ingresa los parámetros de salida de la función como se muestra a continuación:

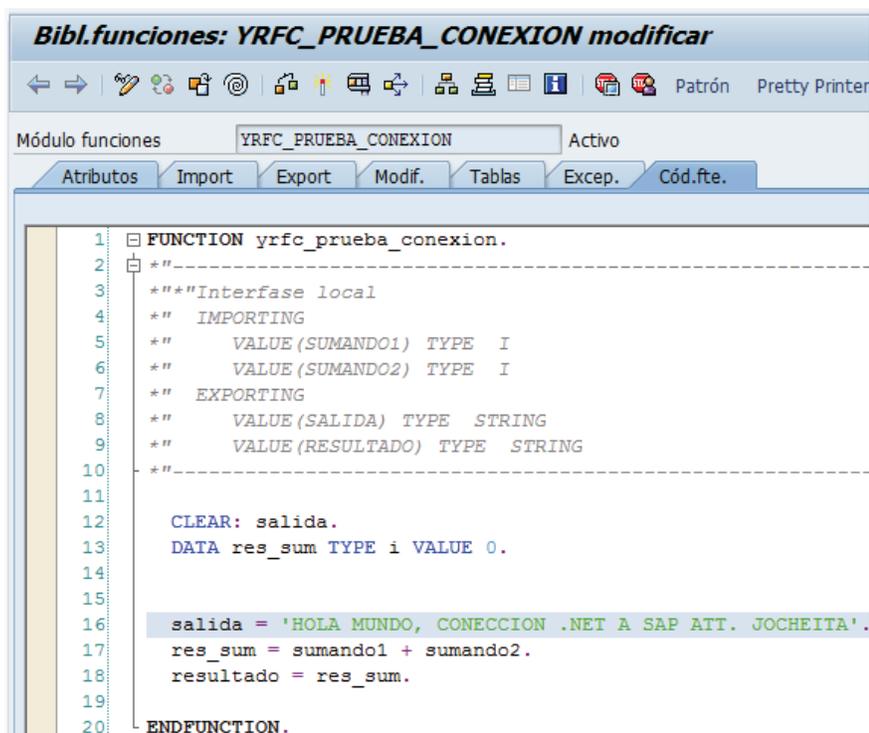
Figura 243. Ingreso de parámetro de salida de la función



Fuente: elaboración propia, con base en SAP R/3.

En la pestaña Cod.Fte, para ingresar el código de la función, el cual recibe como parámetros dos números enteros por lo que el resultado va ser la sumatoria de ambos y también la función va retornar el siguiente texto: HOLA MUNDO, CONECCION .NET A SAP ATT. JOCHEITA.

Figura 244. Código fuente de función RFC



```
1 FUNCTION yrfc_prueba_conexion.
2  *-----
3  ***"Interfase local
4  ** IMPORTING
5  **   VALUE(SUMANDO1) TYPE I
6  **   VALUE(SUMANDO2) TYPE I
7  ** EXPORTING
8  **   VALUE(SALIDA) TYPE STRING
9  **   VALUE(RESULTADO) TYPE STRING
10 *-----
11
12 CLEAR: salida.
13 DATA res_sum TYPE i VALUE 0.
14
15
16 salida = 'HOLA MUNDO, CONECCION .NET A SAP ATT. JOCHEITA'.
17 res_sum = sumando1 + sumando2.
18 resultado = res_sum.
19
20 ENDFUNCTION.
```

Fuente: elaboración propia, con base en SAP R/3.

Luego hay que compilar, activar y ejecutar la función .

Hay que ingresar los valores de los sumandos.

Figura 245. Ingreso de parámetros de la función

Entorno de test para módulos de funciones

Debugging Dir.datos test

Test para grupo funciones ZGRFC
Módulo funciones YRFC_PRUEBA_CONEXION
Mayúsculas/Minúsculas

Sist.dest.RFC:

Parámetros p.import	Valor
SUMANDO1	4
SUMANDO2	5

Fuente: elaboración propia, con base en SAP R/3.

Ejecutamos la función con la tecla F8 o dar clic en el botón del reloj, por lo que se mostrará el resultado de la función.

Figura 246. Resultado de la función

Test del módulo de funciones: Resultados

Test para grupo funciones ZGRFC
Módulo funciones YRFC_PRUEBA_CONEXION
Mayúsculas/Minúsculas

Tiempo ejec.: 104 Microsegundos

Sist.dest.RFC:

Parámetros p.import	Valor
SUMANDO1	4
SUMANDO2	5

Parámetros export	Valor
SALIDA	HOLA MUNDO, CONECCION .NET A SAP ATI. JOCHEITA
RESULTADO	9

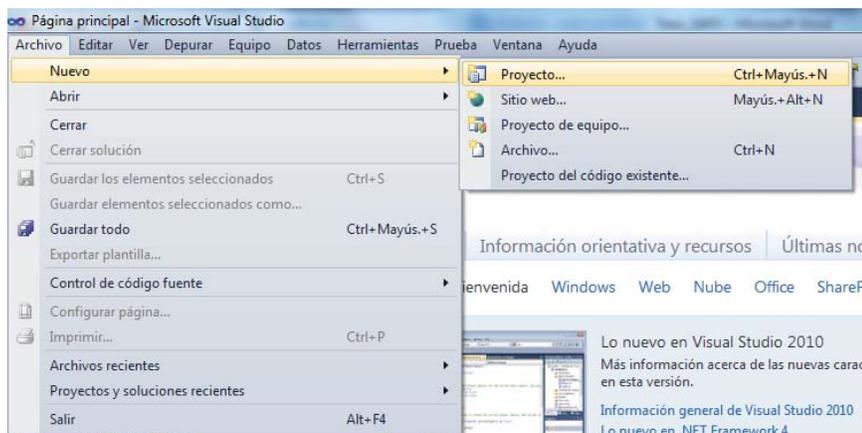
Fuente: elaboración propia, con base en SAP R/3.

Ya que se tiene la función RFC creada, junto con el destino RFC entonces lo que hace falta es realizar el programa .Net que utilice esta función.

Para poder acceder a la función RFC creada en SAP, hay que utilizar la plataforma de Visual Studio .Net C# con las librerías sapnco y sapnco_utils. Por lo que se realizara una clase para la conexión a SAP la cual se llamara ConexiónSAP y en la interfaz gráfica se realizara el ingreso de parámetros y recepción de resultados como se presenta a continuación:

Hay que crear un nuevo proyecto en Visual Studio.

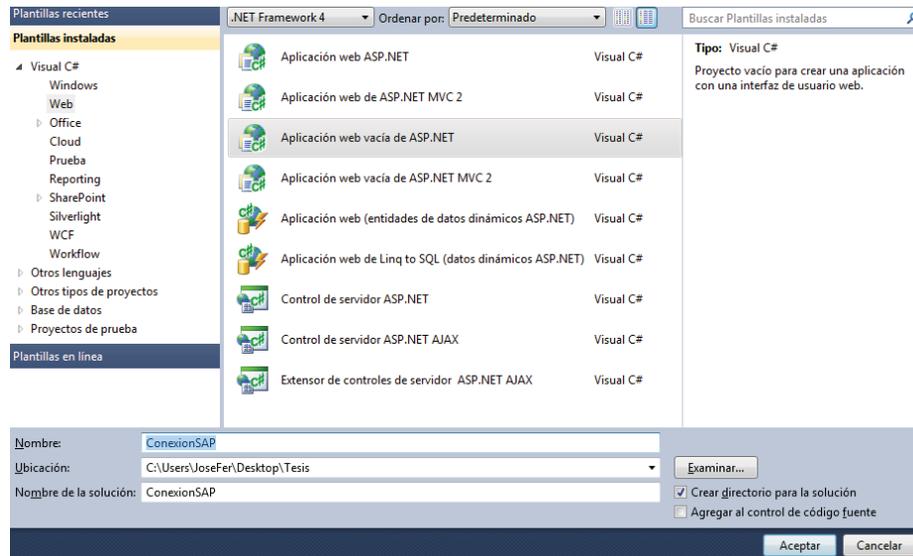
Figura 247. **Creación de nuevo proyecto en Visual Studio .Net**



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Luego hay que seleccionar Visual C# -> Web -> Aplicación web vacía de Asp.Net, le dar clic en el botón de Aceptar.

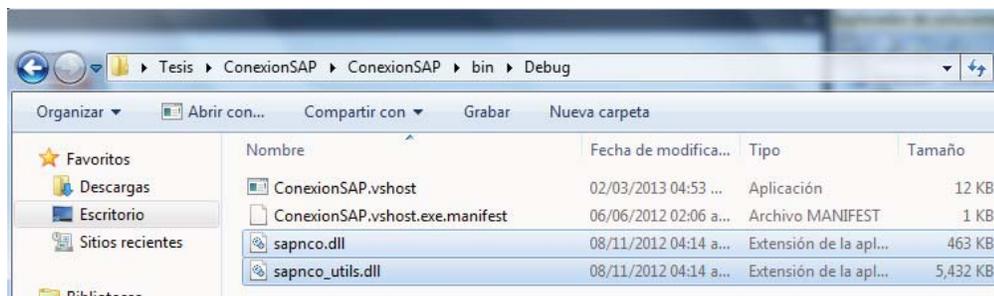
Figura 248. **Aplicación WEB vacías de ASP.NET**



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Se incluyen los archivos `sapnco.dll` y `sapnco_utils.dll` en la carpeta DEBUG del proyecto.

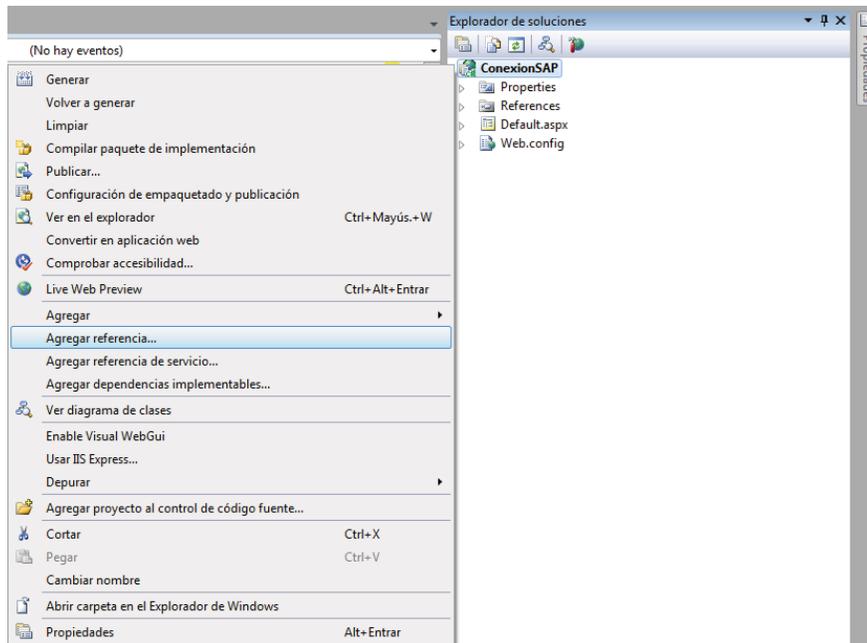
Figura 249. **Ingreso de archivos DLL en la carpeta del proyecto**



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Se incluyen las referencias con clic derecho en el proyecto y luego Agregar Referencia.

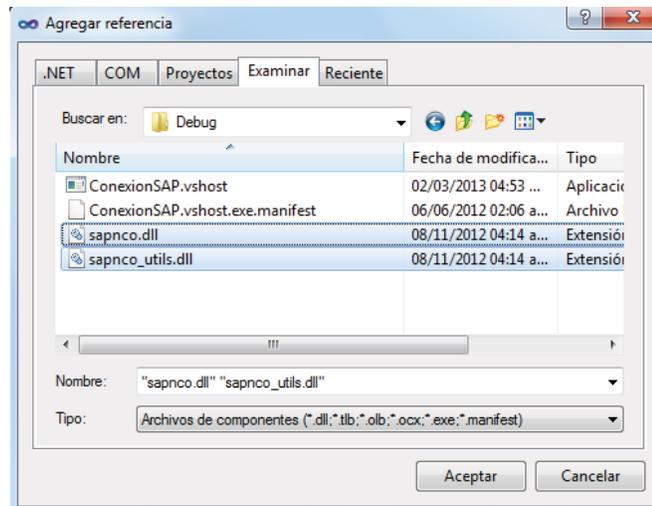
Figura 250. **Agregar referencia de archivos DLL**



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Hay que ir la pestaña de Examinar, luego hay que buscar la carpeta DEBUG, seleccionar los archivos DLL y luego dar clic en el botón de Aceptar.

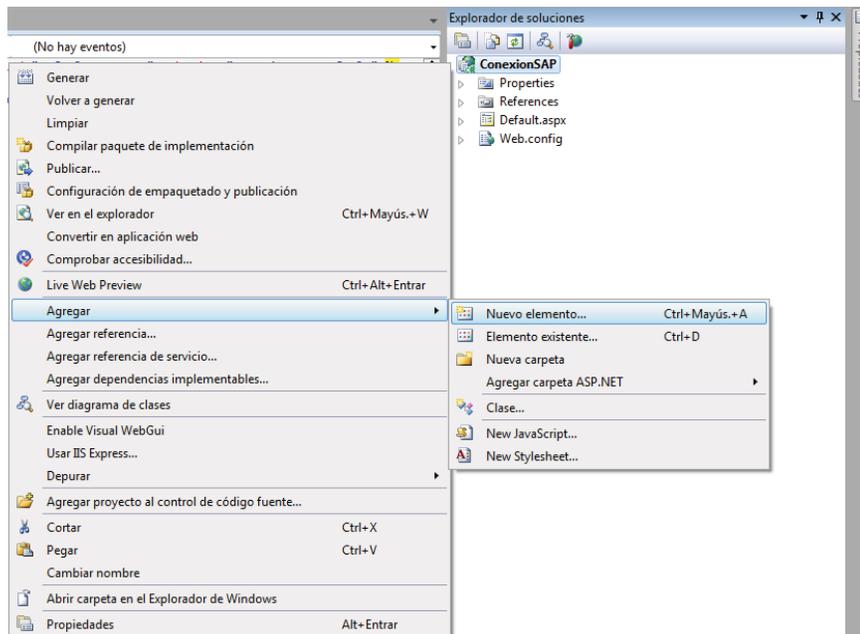
Figura 251. Selección de archivos DLL del proyecto



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Se debe de crear una nueva clase en el proyecto, dando clic derecho en el proyecto, agregar y luego Nuevo Elemento.

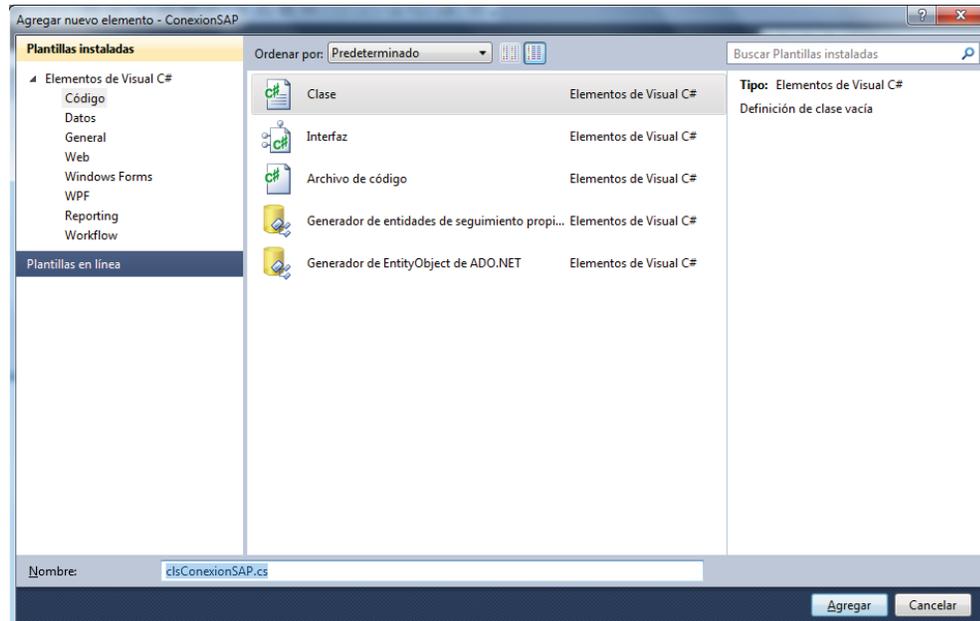
Figura 252. **Agregando nuevo elemento en el proyecto**



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Hay que seleccionar la sección Código, luego Clase, luego ingresar el nombre de ella de la clase cual es clsConexionSAP.cs y luego dar clic en el botón de agregar.

Figura 253. Creación de clase en Visual Studio .Net



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

En la sección de declaración de librerías hay que ingresar la siguiente instrucción: `using SAP.Middleware.Connector;`

Hay que crear el constructor de la clase y sus métodos respectivos.

Figura 254. Clase de conexión a SAP y consumo de funciones RFC

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using SAP.Middleware.Connector;
using System.Data;

public class clsConexionSAP
{
    private RfcDestination ManejoSAP;
    private RfcConfigParameters Coneccion;

    public clsConexionSAP(String Nombre, String Host, String Numero, String Usuario, String Contrasenia,
        String Cliente, String Idioma, String Pool, String TimeOut)
    {
        this.Coneccion = new RfcConfigParameters();

        this.Coneccion.Add(RfcConfigParameters.Name, Nombre);
        this.Coneccion.Add(RfcConfigParameters.AppServerHost, Host);
        this.Coneccion.Add(RfcConfigParameters.SystemNumber, Numero);
        this.Coneccion.Add(RfcConfigParameters.User, Usuario);
        this.Coneccion.Add(RfcConfigParameters.Password, Contrasenia);
        this.Coneccion.Add(RfcConfigParameters.Client, Cliente);
        this.Coneccion.Add(RfcConfigParameters.Language, Idioma);
        this.Coneccion.Add(RfcConfigParameters.PoolSize, Pool);
        this.Coneccion.Add(RfcConfigParameters.IdleTimeout, TimeOut);

        this.ManejoSAP = RfcDestinationManager.GetDestination(Coneccion);
    }

    public string PruebaConexion(int sumando1, int sumando2)
    {
        try
        {
            string salida = "";

            IRfcFunction funcion = this.ManejoSAP.Repository.CreateFunction("YRFC_PRUEBA_CONEXION");

            //Ingreso de parametros de la funcion
            funcion.SetValue("SUMANDO1", sumando1);
            funcion.SetValue("SUMANDO2", sumando2);

            //Ejecutando la funcion
            funcion.Invoke(this.ManejoSAP);

            //Obteniendo los valores de retorno de la funcion
            salida = funcion.GetValue("RESULTADO").ToString() + "_" + funcion.GetValue("SALIDA").ToString();

            return salida.Trim();
        }
    }
}
```

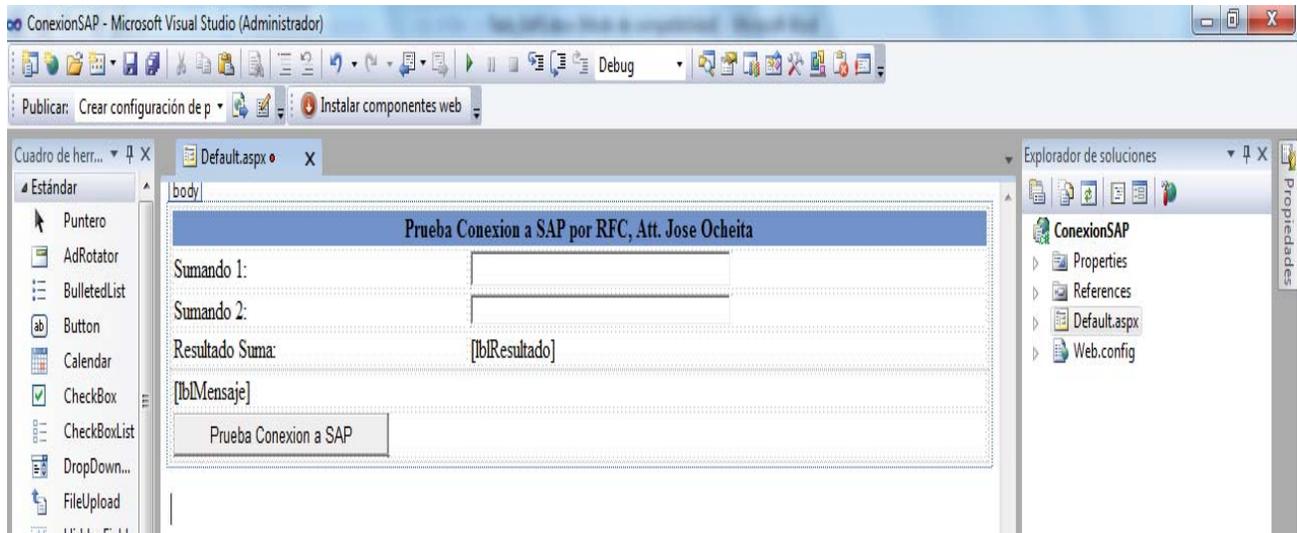
Continuación de la figura 254.

```
catch (RfcCommunicationException e)
{
    return e.Message.Trim();
    // network problem...
}
catch (RfcLogonException e)
{
    return e.Message.Trim();
    // user could not logon...
}
catch (RfcAbapRuntimeException e)
{
    return e.Message.Trim();
    // serious problem on ABAP system side...
}
catch (RfcAbapBaseException e)
{
    return e.Message.Trim();
    // The function module returned an ABAP exception, an ABAP message
    // or an ABAP class-based exception...
}
}
```

Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Luego hay que realizar el diseño de la página web la cual debe de contener el ingreso de parámetros de los sumandos, la recepción de la respuesta del servicio por medio la etiqueta Resultado Suma y un botón para realizar la prueba de conexión, como se muestra a continuación:

Figura 255. **Diseño de página web de utilización de RFC**



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

En el método Page_Load se hace la instancia de la clase y en el método Button1_Click se realiza la llamada a la función.

Figura 256. Creación de la instancia y ejecución de la función RFC

```
Default.aspx | Default.aspx.cs | clsConexionSAP.cs | conexion
ConexionSAP.Default
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace ConexionSAP
{
    public partial class Default : System.Web.UI.Page
    {
        private clsConexionSAP conexion;
        private String Nombre;
        private String Host;
        private String Numero;
        private String Usuario;
        private String Contrasenia;
        private String Cliente;
        private String Idioma;
        private String Pool;
        private String Timeout;

        protected void Page_Load(object sender, EventArgs e)
        {
            this.Nombre = "PRD"; //--> Variable
            this.Host = "sapdesarrollo"; //--> Host Name
            this.Numero = "00"; //--> Instancia
            this.Usuario = "prueba"; // --> Usuario RFC
            this.Contrasenia = "tesis_josefer"; // --> Clave Usuario
            this.Cliente = "300";
            this.Idioma = "ES";
            this.Pool = "50";
            this.Timeout = "8";

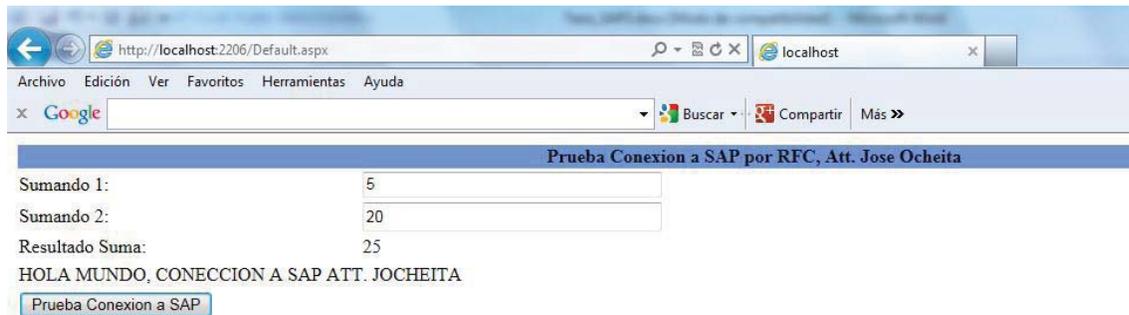
            this.conexion = new clsConexionSAP(this.Nombre, this.Host, this.Numero, this.Usuario, this.Contrasenia, this.Cliente, this.Idioma, this.Pool, this.Timeout);
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            string [] valores = this.conexion.PruebaConexion(5, 20).Split('_');
            this.lblResultado.Text = valores[0].Trim();
            this.lblMensaje.Text = valores[1].Trim();
        }
    }
}
```

Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

Luego se prueba la conexión ingresando los parámetros y presionando el botón de Prueba Conexión a SAP, como se muestra a continuación:

Figura 257. **Resultado de la llamada a la función RFC de SAP**



Fuente: elaboración propia, con programa de Visual Studio .Net C# 2010.

3.3. Análisis de utilización de Webservice y RFC (Remote Function Call)

En esta sección es muy importante tomar en cuenta las diferencias que hay entre la utilización de Webservice y RFC, ya que se puede tomar mejores decisiones para poder satisfacer las diferentes necesidades que se tengan.

3.3.1. Análisis de WebServices en SAP

A continuación se presenta una tabla de las características y ventajas de utilización de Webservices en SAP R/3. Esto ayuda a tener un mejor conocimiento analítico del funcionamiento del método.

Tabla V. **Características y ventajas de WebServices en SAP**

CARACTERÍSTICAS	VENTAJAS
Abstracción - ocultar detalles técnicos que puede ser confuso	Seguridad - Ambas arquitecturas tienen consideraciones de seguridad. Estos también deben sopesarse con la arquitectura de seguridad y estándares actuales
Modularidad - La descomposición de la complejidad y la creación de "piezas reutilizables"	Implementación - independientemente de la opción de punto de integración con SAP, el diseño de la aplicación debe ser flexible y reutilizable. Esto proporcionará flexibilidad para migrar entre estas opciones u otras opciones que se consideren necesarias en el futuro.
Estandarizado conectividad - permitiendo composición flexible de servicios para formar procesos	Costos - Presupuesto y costos son siempre una consideración y vienen en muchos sabores (coste de las licencias, el costo de mejoras de infraestructura necesarias, conjunto de habilidades de los recursos, el costo de los recursos, los costos de implementación, etc.)
Diseño Incremental - que permite cambios en la composición y la configuración sin afectar a las partes internas de los componentes, y viceversa	Sigue un modelo de Capas

Fuente: elaboración propia.

3.3.2. Análisis de RFC (Remote Function Call)

A continuación se presenta una tabla de las características y ventajas de utilización de RFC en SAP R/3. Esto ayuda a tener un mejor conocimiento analítico del funcionamiento del método.

Tabla VI. **Características y ventajas de RFC**

CARACTERÍSTICAS	VENTAJAS
Llamando interfaz para programas ABAP	Sigue un modelo de capas
Cada programa ABAP puede llamar a un módulo de función remota mediante la función de llamada de comando... DESTINO. El destino parámetro indica al sistema SAP que el módulo de función llamada se ejecuta en un sistema distinto al sistema de llamadas. RFC comunicación con el sistema remoto se lleva a cabo como parte de la función de comando CALL	Es un Estándar de Negocio.
RFC módulos de función en un sistema SAP debe ser módulos de función apropiados y deben estar registrados en el sistema SAP como a distancia.	Garantiza estabilidad y compatibilidad futura.
Si el programa que llama y el llamado programa son programas ABAP, la interfaz RFC ofrece ambos interlocutores. El programa de llamada puede ser cualquier programa ABAP, y el programa llamado debe ser un componente funcional que se registra como remoto.	Garantiza orientación a objetos.
Interfaces para llamar no-ABAP programas	Independencia de entorno (apertura).
Llamar funciones estandar en el repositorio de SAP	Permite la eficiencia en las operaciones en SAP

Fuente: elaboración propia.

CONCLUSIONES

1. Las transacciones SE16 y SQVI se pueden utilizar para la realización de reportes, sin embargo, es una técnica menos efectiva con respecto a WinShuttle Query ya que este brinda una interfaz gráfica muy sencilla.
2. El proceso Batch Input es un procedimiento para realizar cargas de datos hacia SAP sin embargo este no da el 100 % de confiabilidad para las cargas masivas de datos, porque depende mucho de la transacción que se está utilizando, la programación en ABAP, si esta es una carga secuencial o paralela, si el código generado por la transacción SHDB no hay ningún error de ingreso de datos, etc. En muchos casos las transacciones utilizadas para la carga masiva de datos no tienen la capacidad de atención ya sea por rapidez de la llegada de la información o simplemente por razones de configuración del sistema de SAP, por lo que el sistema en si aborta las operaciones de carga o no realiza una carga de datos consistente. Por lo que en este trabajo de investigación se sugirió la utilización de la herramienta de Winshuttle Transaction, ya que esta soluciona los problemas de carga masiva de datos a SAP y a la vez no se requiere conocimiento técnico de programación ABAP y otros aspectos.
3. La utilización de servicios web con SAP es mucho más sencillo que utilizar funciones RFC, ya que para intercomunicar varios sistemas con diferentes plataformas hacia SAP y viceversa con funciones RFC, se tiene que adquirir las librerías necesarias de cada plataforma. Por lo que conlleva invertir a nivel monetario, a nivel de tiempo de desarrollo y

entendimiento de las librerías. Mientras que los servicios web se pueden publicar servicios desarrollados en diferentes plataformas que SAP de la misma manera los consume ya que genera automáticamente las estructuras y procedimientos para poder realizar el consumo de los mismos. Por lo que se tiene un ahorro a nivel de tiempo con respecto a utilizar funciones RFC, sin embargo a nivel monetario la inversión es mucho mayor ya que se tiene que invertir en módulos de arquitectura orientada a los servicios en SAP.

4. Ambos métodos de intercomunicación Servicios Web y RFC, son muy seguros en SAP ya que estos siguen un estándar de seguridad. En Servicios Web, SAP tiene que reconocer el archivo WSDL del servicio publicado, para que luego este genere todas las estructuras necesarias para el consumo y posteriormente se tiene que configurar un puerto lógico para que SAP pueda administrar y guardar todos los servicios, esto permite que SAP pueda llevar un mayor control (monitoreo de consumo de memoria, disco, etc.) y a la vez tener un fácil desarrollo de aplicaciones de consumo de servicios con ABAP. En cambio en las funciones RFC es mucho más sencillo porque se tiene que crear un Destino, que este se puede interpretar como un acceso a SAP por medio de uno de los métodos de autenticación que este provee y luego se realiza la llamada a una de las funciones creadas en la transacción se37 que tienen como requisito la conexión remota.
5. Winshuttle Transaction y Query ambas herramientas están dirigidas para ingenieros informáticos que tienen conocimiento en programación en ABAP o ingenieros en sistemas y a usuarios finales que no tienen conocimiento de programación pero si tienen conocimiento de utilización de SAP.

RECOMENDACIONES

1. Utilizar la herramienta de Winshuttle Transaction para realizar cargas masivas de datos hacia SAP, en los casos que se necesita rapidez, eficiencia y específicamente cuando no se tiene una persona especializada en programación en ABAP o un ingeniero en sistemas, cuando no se tiene un acceso directo a SAP y permisos para crear una calendarización y procesos de fondo (Programador de tareas de SAP) ya que Winshuttle Transaction tiene su propio calendarizador, cuando es necesario realizar la carga de datos urgentemente sin tener que realizar ordenes de transporte para pasar a productivo.
2. Utilizar la herramienta de Winshuttle Query para realizar reportes de SAP en tiempo real, por lo que puede facilitar crear reportes con gráficos interactivos o también para crear tableros interactivos (Dashboards) utilizando las herramientas de Excel, Flash, etc.
3. En el método de carga de datos con Batch Input, se recomienda realizar un set de pruebas del programa, del código generado por la transacción SHDB antes de realizar la carga de datos en productivo. Para mayor efectividad en la carga total esta se puede subdividir en pequeñas cargas de datos con el objetivo de no sobrecargar el sistema. También es recomendable realizar un archivo LOG de los registros que no se han cargado exitosamente para luego realizar un arreglo de los mismos y realizar una nueva carga.

4. Se recomienda el método de carga de datos con Batch Input si se tiene una persona especializada en programación en ABAP o un ingeniero en sistemas que tenga buen conocimiento de SAP y que este dedicada totalmente a programar procesos de carga.

5. La mejor forma de poder intercomunicar sistemas externos hacia SAP y viceversa es utilizando Servicios Web para que SAP los pueda consumir y para que los sistemas externos puedan consumir servicios de SAP es utilizando RFC (Remote Function Call). Como se mostró en el capítulo 3, SAP consume servicios web desarrollados en Visual Studio .Net y SAP publica una función remota para que luego esta sea consumida en Visual Studio .Net. Por lo que el capítulo 3 se considera un método de intercomunicación sin tener costos grandes de inversión ya que esto implicaría actualización de módulos (Módulo PI) de SAP, actualización de arquitectura, etc.

BIBLIOGRAFÍA

1. BENITEZ, Victor; WILLIAMS, Tom. *Webinar a better way to integrate SAP and Microsoft Excel*. [en línea]. <<http://www.winshuttle.com/Resources/Webinars>>. [Consulta: 24 de enero 2012.]
2. Blog de SAP, customizing, ABAP y noticias. *Breve historia de SAP*. [en línea]. <<http://www.blogdesap.com/2012/08/la-historia-de-sap-en-una-infografia.html>>. [Consulta: lunes, 13 de agosto de 2012].
3. _____. *Procesar juegos de datos de batch input*. [en línea]. <<http://www.blogdesap.com/2011/05/procesar-juegos-de-datos-de-batch-input.html>>. [Consulta: 15 de febrero del 2013]
4. GOMEZ ZEA, Juan Carlos. *Webinar. Cómo mejorar la calidad de vida de usuarios SAP, generando grandes ahorros a sus empresas*. [en línea] <<https://www1.gotomeeting.com/register/226612945>>. [Consulta: 25 de octubre de 2012].
5. HERREROS LUCAS, José Luis. *Programación en ABAP/4 para SAP R/3*. México D.F.: Osborne McGraw-Hill, 1999, 700 p.
6. KELLER, Horst. *The official ABAP Reference*. 2 vols. Alemania: Galileo Press, 2005. 1216 p.

7. MEINERS, Johanne; NÜSSER, Wilhelm. *SAP Interface Programming*. Alemania: Galileo Press. USA: Sap. PressAmerica, 2004. 379 p.
8. Pack for SAP R3: *Configuring Sap R3 system, creating RFC destination for outbound data (SM59)*. [en línea]. <http://publib.boulder.ibm.com/infocenter/wtxdoc/v8r2m0/index.jsp?topic=/com.ibm.websphere.dtx.packsapr3.doc/tasks/t_pack_sapr3_2_Creating_RFC_Destination_for_Outbound_Data_SM59.htm>. [Consulta: 2 de febrero de 2013].
9. REYES, Francisco. *SAP R/3, Remote Function Call (RFC)*. [en línea]. <<http://www.stgocar.com/freyes/>>. [Consulta: 5 de febrero de 2013].
10. SAP Net Connector 3.0 Overview. 3a ed. Derechos de Autor 2010-2011 SAP AG, todos los derechos reservados.
11. SAP Net Connector 3.0 Programming Guide. 3a ed. Derechos de Autor 2010-2011 SAP AG, todos los derechos reservados.
12. *Sap And Abap Tips and Facts. SAP ABAP Tutorial on Subroutines. PERFORM FORM ENDFORM*. [en línea]. [ref. 21 de marzo del 2008]. Disponible en Web: <<http://abaplovers.blogspot.com/2008/03/sap-abap-tutorial-on-subroutines.html>>.
13. SAP Library. *RFC Administration: introductory example. Definition of the Destination in SM59*. [en línea]. <http://help.sap.com/saphelp_erp2005/helpdata/en/78/67ee40755>

2742ae10000000a155106/frameset.htm>. [Consulta: 6 de febrero de 2013].

14. _____. *Setting RFC Type in Transaction SM59*. [en línea]. <http://help.sap.com/saphelp_erp2005/helpdata/en/42/e7a9753c303ee4e10000000a1553f6/frameset.htm>. [Consulta: febrero de abril de 2013].
15. Sap Netweaver 2004. *Abap Workbench Fundamentals, Participant Handbook*. Numero de material 50077029. Derechos de Autor 2005 SAP AG, todos los derechos reservados.
16. SAP Tutorials @ Sapbrainsonline. *ABAP Syntax & Keyword*. [en línea]. <<http://sapbrainsonline.com/abap-tutorial/syntax>>. [Consulta: 6 de enero de 2013].
17. SCN Sap Community Network. *Batch Input FAQ*. [en línea]. [ref. 2 de septiembre del 2011]. Disponible en Web: <http://search.sap.com/ui/scn#query=batch+input&startindex=1&filter=scm_a_site%28scm_v_Site11%29>.
18. Scribd. *Sentencias condicionales en ABAP*. [en línea]. <<http://es.scribd.com/doc/25289651/SAP-Instrucciones-de-Control-y-Bucles-en-ABAP>>. [Consulta: 5 de enero de 2013].
19. _____. *Tablas internas*. [en línea]. Disponible en Web: <<http://www.scribd.com/doc/26898636/SAP-Tablas-Internas-en-ABAP>>.[Consulta: 10 de enero de 2013].

20. Shoovng.com. *Sap, QuickViewer SQVI*. [en línea]. [ref. 22 de junio del 2011]. Disponible en Web: <<http://es.shvoong.com/internet-and-technologies/software/2177126-sap-quickviewer-sqvi/>>.
21. TcodeSearch. *SAP "SM59" Transaction codes*. [en línea]. <<http://www.tcodesearch.com/tcodes/search?q=SM59>>. [Consulta: 10 de enero de 2013].
22. Todo ABAP. *Datos y Tipos en ABAP*. [en línea]. <http://www.todoabap.com.ar/tipos_datos.html>. [Consulta: 15 de enero de 2013].
23. WEB de programación ABAP 4. *Tipos de datos y otras sentencias*. [en línea]. <http://www.abap.es/centro_Manual_03_tipos.htm>. [Consulta: 5 de enero de 2013].