

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA

**DESARROLLO EN SAP DE HERRAMIENTAS PARA EL
MÓDULO LABORATORIO**

**SAP DEVELOPMENT OF TOOLS FOR THE LABORATORY
MODULE**

Realizado por
Adrián Sánchez Fernández
Tutorizado por
José María Álvarez Palomo
Eusebio Hernández Villalobos
Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, Diciembre de 2016

Fecha defensa:
El Secretario del Tribunal

Resumen:

En la actualidad, los sistemas de planificación de recursos empresariales, también conocidos como sistemas ERP, son herramientas muy usadas en la mayoría de los negocios, facilitando el control del proceso empresarial. Dos de las grandes características de los ERP es que son modulares y configurables, permitiendo instalar aquellos módulos necesarios para la empresa y poder configurarlos conforme surjan necesidades de hacerlo.

SAP es una compañía alemana cuyo producto estrella es su ERP, compatible con la gran mayoría de sistemas operativos, bases de datos, aplicaciones y componentes de hardware de casi cualquier proveedor. El desarrollo de nuevas herramientas se realiza mediante código ABAP IV, también propiedad de SAP.

En este trabajo, se va a desarrollar la creación de nuevas herramientas para dicho ERP y su implantación en el sistema final, gestionando todas las fases que existen en cualquier desarrollo software. Más concretamente, se crearán herramientas que permitirán guardar una serie de datos en el sistema, realizar cálculos y generar informe de los resultados obtenidos.

Palabras clave:

Planificación de recursos empresariales, ERP, Empresa, SAP, SAP NetWeaver, ABAP, Control de Materiales, Normas UNE.

Abstract:

Nowadays, Enterprise Resource Planning Systems, more known as ERP Systems, are tools widely used in most business, making business process control easier. The two most important characteristics of ERP Systems are they are modular and configurable, allowing companies to install only those modules required by the company and they can be set when they are needed.

SAP is a German company whose best-known product is its ERP, which is compatible with most of operating systems, databases, applications and hardware from any supplier. The development of new tools within SAP System is done using ABAP code, also owned by SAP.

In this project, new tools for this ERP are going to be developed and implemented in it, managing all phases which exist in any software development. More specifically, we are going to develop tools which will save data in the system, perform calculations and generate reports about the results obtained.

Keywords:

Enterprise Resource Planning, ERP, Company, SAP, SAP NetWeaver, ABAP, Materials Control, UNE Standards.

Índice

1. Introducción	9
1.1 Entorno del Trabajo Fin de Grado	9
1.1.1 Finalidad.....	9
1.1.2 Ámbito del Trabajo.....	10
1.2 Metodología	13
1.3 Entorno de desarrollo	14
1.3.1 SAP.....	14
1.3.2 SAP Business Suite.....	14
1.3.3 SAP NetWeaver	15
1.3.4 SAP ERP.....	16
1.3.5 ABAP IV	18
1.3.6 Otras herramientas.....	22
1.4 Organización del documento	23
2. Preparación del trabajo	25
2.1 Elección de la temática del TFG	25
2.2 Recopilación de documentación	25
3. Análisis y Diseño	27
3.1 Análisis de los Requisitos	28
3.2 Casos de Uso	30
3.3 Diseño	36
3.3.1 Entrada de Muestras	36
3.3.2 Entrada de Resultados.....	37
3.3.3 Generación de Informes.....	48
3.4 Modelo de Datos	49
4. Implementación	63
4.1 Diccionario de Datos	65
4.2 Entrada de Muestras	66
4.3 Entrada de Resultados	70
4.4 Creación de Informes	77
5. Pruebas e Implantación	87
5.1 Pruebas	89
5.2 Implantación	91
6. Cambios	93

7. Conclusiones	99
Bibliografía	101
Tabla de Ilustraciones	103

1.Introducción

1.1 Entorno del Trabajo Fin de Grado

1.1.1 Finalidad

La idea de este Trabajo Fin de Grado es aprovechar el funcionamiento de un Sistema de Planificación de Recursos Empresariales (más conocido como ERP: Enterprise Resource Planning) para unir 2 departamentos totalmente diferentes de una empresa, de forma que puedan colaborar entre ellos para obtener un beneficio común.

Un ERP es un sistema de gestión de información que automatiza mucha de las prácticas de negocio asociadas con los aspectos operativos o productivos de una empresa, tales como producción, ventas, compras, logística, gestión de proyectos...

Los objetivos principales de los sistemas ERP son:

- Optimización de los procesos empresariales.
- Acceso a la información
- Posibilidad de compartir información entre todos los componentes de la organización
- Eliminación de datos y operaciones innecesarias de reingeniería.



Ilustración 1 - Elementos de un ERP

El propósito fundamental de un ERP es otorgar apoyo a los clientes del negocio, tiempos rápidos de respuesta a sus problemas, así como un eficiente manejo de información que permita la toma oportuna de decisiones y disminución de los costos totales de la operación.

Algunas de las características destacables de los ERP son:

- Deben ser modulares; es decir, su funcionalidad debe encontrarse dividida en módulos, que puedan ser instalados de acuerdo a los requerimientos del cliente, de forma que el conjunto de departamentos pueda aprovechar toda la información que comparten entre ellos y que se genera a partir de sus procesos.
- Deben ser configurables; es decir, pueden ser configurados mediante desarrollos de código software.
- Deben disponer de una Base de datos centralizada.
- Los componentes del ERP deben interactuar entre sí, consolidando las operaciones.
- Etc...

1.1.2 Ámbito del Trabajo

El principal objetivo de este Trabajo Fin de Grado es el de configurar un ERP para implantar una nueva funcionalidad en su entorno, de forma que de soporte a nuevas actividades de la empresa.

En este caso, se va a desarrollar e implantar una serie de herramientas transversales al módulo de Laboratorio de una empresa de Control de Calidad de Materiales y, en concreto, para el material Ladrillo.

La función de este departamento es de ofertar una serie de servicios, en los cuales se realizan una o más pruebas o ensayos con materiales, de forma que puedan comprobar que cumplen la normativa específica para dicho material.

Dichas normas, denominadas Normas UNE (acrónimo de Una Norma Española), son un conjunto de normas tecnológicas creadas por los comités técnicos de normalización (CTN), de los que forman parte todas las entidades y agentes implicados e interesados en los trabajos del comité. Se tratan de unos documentos técnicos de aplicación voluntaria, fruto del consenso, basados en los resultados de la experiencia y del desarrollo tecnológico y aprobado por un organismo de normalización reconocido.



Ilustración 2 - Normas UNE

Estas normas garantizan unos niveles de calidad y seguridad que permiten a cualquier empresa posicionarse mejor en el mercado y constituyen una importante fuente de información para los profesionales de cualquier actividad económica.

Desde laboratorio se realiza una oferta dentro del ERP, creando el expediente de un cliente que desea contratar una serie de servicios para uno o varios materiales. Tras la contratación, a partir de una ventana genérica en el módulo laboratorio del ERP de la empresa, se crea una muestra¹ a la que se le asigna los servicios que se han contratado.

Ilustración 3 - Entrada de Muestras previa

El proceso ya se encuentra creado para otros materiales (como suelos, zahorras, mezclas bituminosas, etc.), pero se pretende crear un proceso similar para Ladrillos, de forma que se le de soporte a todas las operaciones de Laboratorio con este material.

Antes de la elaboración de la de este trabajo, el proceso continuaba de la siguiente forma: los técnicos de laboratorio realizaban los ensayos y anotaban los datos de las mediciones en unos borradores². Después, introducían los datos en una plantilla Excel, configurada con las tablas y fórmulas necesarias para calcular los resultados de los ensayos de acuerdo a la norma UNE que le correspondiera. Posteriormente, se insertan las tablas o campos de resultados en una plantilla Word, que hará las funciones de Informe. Este documento debe ser manipulado para añadir los datos

¹ Una muestra es una cierta cantidad de un material, suficiente para realizar los controles de calidad que se soliciten, que se considera representativa del total del conjunto del que ha sido extraída.

² Los borradores son plantillas en papel dónde los técnicos anotan las mediciones de un ensayo y sus resultados.

relativos de la muestra y del cliente que lo solicita, así como otros datos que varían de un informe a otro.

1.1.- MEDICIÓN DE LAS DIMENSIONES Y COMPROBACIÓN DE LA FORMA (UNE 67030:85 - UNE 67030:86 Ent)

Fecha inicio de ensayo: _____ Equipos: Calibre cod. MA0417
Galgas cod. MA0575
Piega de precisión cod. MA0477

MECIDA DE LAS DIMENSIONES

FIL	SOGA (mm)	TIGÓN (mm)			DRIBESO (mm)
		1ª medida	2ª medida	valor medio	
-1				0,0	
-2				0,0	
-3				0,0	
-4				0,0	
-5				0,0	
-6				0,0	

COMPROBACIÓN DE LA FORMA (MÉTODO DE FECHAS)

Ladrillo tipo V (medir flecha en 1 tabla, 1 canto y 2 testas)
 Ladrillo tipo NV (medir flecha en 1 tabla y 1 canto)

FIL	TABLA (mm)	CANTO (mm)	TESTA 1 (mm)	TESTA 2 (mm)
-1				
-2				
-3				
-4				
-5				
-6				

DIAGONAL: Es la medida de la flecha en el diagonal, su valor se calcula como la raíz cuadrada de la suma de los cuadrados de las medidas de la tabla y del canto.

cemosa
Ingeniería y Control

C/ Suroeste 9 29004 MÁLAGA
TEL: 952220842
FAX: 952221214
URL: www.cemosa.es
E-MAIL: cesaga@cemosa.es

Coordenador: Q11142814 Asesor: 02847201
Teléfono: 952220842

a) Rellenar solo en caso de medir por el método g.

(INSERTAR TABLA 10)

(INSERTAR TABLA 11)

(INSERTAR TABLA 12)

3.1.c. ESPESOR DE PAREDES (UNE-EN 772-16:2011)

(INSERTAR TABLA 16)

3.1.d. PROFUNDIDAD DE LAS CAVIDADES (UNE-EN 772-16:2011)

(INSERTAR TABLA 17)

3.1.e. PARALELISMO DE LAS TABLAS (UNE-EN 772-16:2011)

(INSERTAR TABLA 19)

Página 4 de 25

Ilustración 4 - Plantilla e Informe originales

Tras su finalización, se guarda el documento en el Sistema y se proporciona una copia del mismo al cliente.

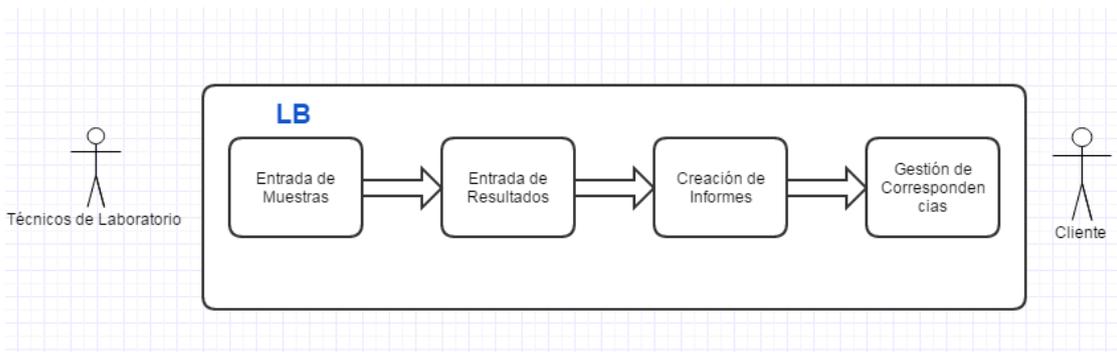


Ilustración 5 - Flujo del Módulo de Laboratorio

Tal y como se puede apreciar, todo el proceso posterior a la creación de la muestra se ejecuta fuera del entorno SAP de la empresa, no existiendo ninguna forma de guardar los datos de los ensayos que no sea la del informe en sí; esto implica que cualquier consulta o modificación deba realizarse sobre el propio informe o los borradores, con el riesgo de que si se perdiesen o estropeasen los borradores, se perdería la información de estos.

Por lo tanto, se pretende informatizar todo este proceso de forma que el sistema se encargue de realizar todos estos pasos, liberando a los técnicos de gran parte de la carga de trabajo y mejorando la eficiencia y rapidez del proceso; además, los datos de los ensayos estarán guardados en la Base de Datos del Sistema, pudiendo ser

consultados en cualquier momento y desde cualquier lugar desde el que se tenga acceso al mismo, sin depender de la localización en la que se encuentren los borradores de dichos ensayos. También existe otro beneficio a tener en cuenta: Cada laboratorio cuenta con su propia plantilla, por lo que la existencia de un modo de entrada de resultados común homogenizará el proceso entre todos los laboratorios. Por lo tanto, funcionalmente hablando, se pretende dar soporte para almacenar los datos de una muestra, operar sobre las mediciones realizadas durante los ensayos y guardar los datos de estos, crear los informes a partir de estos datos y el envío de los resultados al cliente que ha realizado la contratación.

1.2 Metodología

Este trabajo se ha dividido en una serie de módulos, de forma que se facilite la integración con los módulos ya existentes en el ERP y los futuros módulos que se añadan. Debido a que este trabajo formará parte de un proyecto mayor, deberá cumplir una serie de especificaciones en cuanto al estilo de desarrollo, de forma que se ajuste a los módulos ya implementados y a módulos futuros.

Tras una reunión con el responsable de desarrollo, se decidió que la planificación de dichos módulos estuviese formada por las siguientes fases:

1. Preparación: esta fase es común a todo ERP e incluye la adquisición de los recursos necesarios para su elaboración. Los técnicos de laboratorio proporcionaron una recopilación de normas, borradores e informes sobre Ladrillos. Adicionalmente, el responsable de desarrollo proporcionó una serie de manuales sobre el ERP en cuestión y el lenguaje de programación.
2. Análisis y Diseño: estudio y análisis de los requisitos y del mejor diseño de la solución. Esta fase se realizó a partir de los documentos anteriormente proporcionados, consultas a los técnicos de laboratorio (quienes simularon el rol de cliente), reuniones con el responsable de desarrollo para consultar opciones de diseño y funcionalidad, entre otras. Además, se creó un foro que sirviese de canal de comunicación directa con los técnicos de laboratorio, de forma que pudiesen ver el resultado del trabajo realizado conforme se fuese desarrollando y dar su visto bueno o proponer modificaciones.
3. Implementación: durante esta fase, se produce el diseño de la Base de Datos que dará soporte a este Trabajo (tablas, vistas, relaciones...) y desarrollo del código fuente.
4. Preparación final: una vez desarrollados las aplicaciones, se realizan operaciones de testeo y verificación para comprobar su correcto funcionamiento y se documenta el trabajo realizado.
5. Implantación y aceptación: Integración del módulo con el resto del Sistema.

1.3 Entorno de desarrollo

Este Trabajo Fin de Grado gira en torno a la tecnología SAP, abarcando el entorno y lenguaje de programación, Bases de Datos, entorno de pruebas, etc.

1.3.1 SAP

SAP SE, más conocida como SAP, es una empresa multinacional alemana dedicada al diseño de productos informáticos de gestión empresarial.



Ilustración 6 - Logo de SAP

Fundada en Alemania en 1972 por cinco ingenieros que trabajaron en IBM, SAP es considerada la mayor compañía mundial del negocio del Software en las áreas de CRM, ERP y SCM, y el tercero más grande como proveedor de software independiente.

Además de la fabricación de software de aplicaciones, SAP construye orientación a servicios directamente en sus soluciones y proporciona una plataforma de tecnología SAP NetWeaver y orientación, proporcionando apoyo a las empresas en el desarrollo de su propia arquitectura orientada a servicios, que abarcan tanto las soluciones SAP y no SAP.

La Arquitectura orientada a servicios (SOA) mueve el panorama ERP hacia el software basado en la web y servicios basados en la actividad empresarial, aumentando la adaptabilidad, la flexibilidad, la apertura y la eficiencia. Este movimiento ayuda a las empresas a reutilizar los componentes de software y no depender tanto dentro de la empresa de las tecnologías de hardware ERP, lo que hace que la adopción de ERP sea más atractivo para las pequeñas y medianas empresas.

Los productos de SAP se centran en ERP, por lo que el producto principal de la compañía es SAP ERP, que forma parte de SAP Business Suite.

1.3.2 SAP Business Suite

SAP Business Suite es un conjunto de programas que permiten a las empresas ejecutar y optimizar distintos aspectos como los sistemas de ventas, finanzas, operaciones bancarias, compras, fabricación, inventarios y relaciones con los clientes. Ofrece la posibilidad de realizar procesos específicos de la empresa o crear módulos

independientes para funcionar con otro software de SAP o de otros proveedores. SAP está basado en una plataforma de tecnología integrada llamada NetWeaver. La suite puede soportar sistemas operativos, bases de datos, aplicaciones y componentes de hardware de casi cualquier proveedor.

Algunas de las áreas empresariales a las que SAP Business Suite proporciona soporte son: finanzas, fabricación, marketing, ventas, Recursos Humanos, etc.

Además de SAP ERP, SAP Business Suite se divide en otros 4 módulos:

- SAP CRM (Customer Relationship Management): se encarga de interactuar con todos los temas relacionados con el cliente, ya sean ventas, marketing o servicios.
- SAP PLM (Product Lifecycle Management): Las funciones más importantes de este módulo son crear y suministrar productos y optimizar los procesos de desarrollo de los productos y sistemas para acelerar su introducción en el mercado.
- SAP SCM (Supply Chain Management): Permite diseñar, construir y poner en marcha la cadena de suministro.
- SAP SRM (Supplier Relationship Management): Ofrece funciones tales como el análisis de gastos, abastecimiento, contratos operativos, pedidos, facturas y gestión de proveedores.

1.3.3 SAP NetWeaver

SAP NetWeaver es una plataforma de tecnología integrada para todas las aplicaciones SAP en el plano técnico. Es conocida como una aplicación orientada a servicios y a la integración, provee al usuario de un vínculo entre lenguajes y aplicaciones y está construido usando estándares abiertos de la industria, por lo que es sencillo negociar transacciones de información con desarrollos de Microsoft .NET, Sun Java EE, e IBM WebSphere.

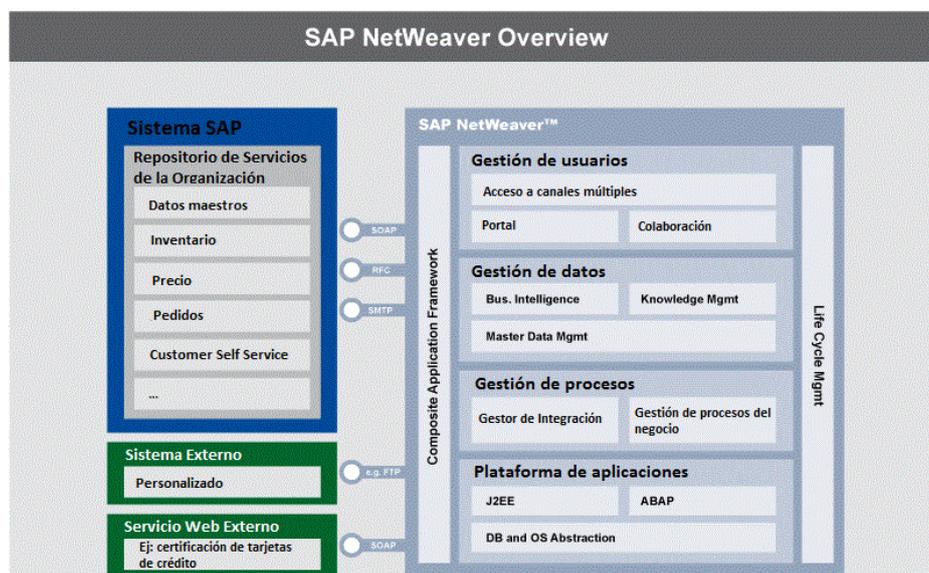


Ilustración 7 - Plataforma SAP NetWeaver

Fue lanzado como un movimiento estratégico de SAP que plantea a las empresas ejecutar todas sus aplicaciones empresariales en una única plataforma integrada con la más firme infraestructura. Esta solución incorpora una gran flexibilidad, una mejor integración con las aplicaciones y construcción en estándares para asegurar la futura interoperación.

Este lanzamiento en suma es una parte del plan de SAP de transformarse en una herramienta más abierta y orientada a servicios adecuados a las necesidades del mercado.

1.3.4 SAP ERP

En la actualidad, y con el fin de afrontar las necesidades que surgen en el seno de las empresas para lograr una productividad más eficaz, existen una gran cantidad de alternativas de software que se aplica a toda la cadena que recorre un producto. Una de las más utilizadas, y en las que se basan otros sistemas, es el producto ERP de SAP, denominado SAP R/3.

El SAP R/3 es un sistema ERP que ha sido diseñado en base a una plataforma que ofrece una gran versatilidad de programación, facilidad de uso, y precisión en el manejo total de los datos recolectados.

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

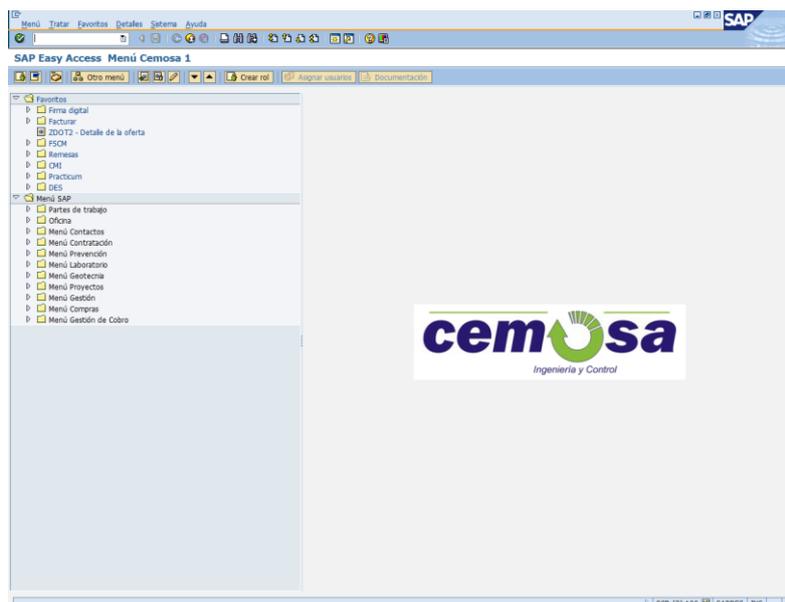


Ilustración 8 - SAP R/3

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

El sistema R/3 está diseñado en módulos claramente diferenciados, que responden a las necesidades propias de cada una de las áreas de una empresa. A su vez, cada uno de los módulos que conforman este sistema posee diferentes submódulos que interactúan entre sí, ofreciendo de esta manera una colección precisa y diferenciada de la información referente a los productos fabricados por la compañía, desde la recepción de las materias primas, pasando por la manufactura y empaquetado de los artículos, hasta la entrega del producto terminado en el punto de venta.



Ilustración 9 - Módulos SAP R/3

Este sistema no sólo es útil para los casos en que se requiera una solución estándar, sino también para aquellas empresas que necesiten un sistema a medida; esto es posible gracias a las poderosas herramientas de desarrollo que se proveen, mediante el lenguaje de programación ABAP/4, que ha sido creado en base a las necesidades comerciales que existen en la actualidad. Por lo tanto, la empresa puede crear o modificar los módulos existentes para ajustarlos a sus necesidades.

El sistema R/3 permite una total personalización, incluyendo la posibilidad de construir interfaces propias, creadas en base a los requerimientos de cada sector, ofreciendo la posibilidad de trabajar con un sistema del tipo abierto, en el que la información almacenada se encuentra disponible en cualquier momento y para todas las áreas de la empresa.

De esta manera, se reducen los costos, se mejora la productividad, y se optimizan los procesos de transacciones, entre otras grandes ventajas.

En base al principio de cliente/servidor, el sistema R/3 funciona a través de un software que trabaja a varios niveles, permitiendo una interacción constante entre los clientes y los servidores, gracias a la posibilidad de crear la cantidad de módulos que sean necesarios. Por otra parte, al tratarse de un sistema de tipo abierto, no sólo facilita los procesos de transacciones y el manejo de la información, sino que además permite un eficaz control de los datos y las comunicaciones entre los clientes y los servidores. SAP R/3 funciona con el protocolo TCP/IP para comunicaciones en red, soporta redes del tipo SQL, utiliza interfaz de comunicación CPI-C para las comunicaciones realizadas entre diferentes programas por intermedio de sistemas múltiples, trabaja bajo las normas ODBC para la conectividad de la base de datos abierta, incluye la arquitectura RPC, dentro del lenguaje ABAP/4 como interfaz de programación abierta, funciona a través del estándar denominado OLE/DDE para llevar a cabo las integraciones de las aplicaciones de las computadoras con el sistema, y responde a las normas de comunicaciones externas X.400/X.500, MAPI.

En la actualidad, el sistema SAP R/3 es totalmente compatible con los sistemas operativos Windows 95, Windows 98, Windows NT, Macintosh, Red Hat de Linux, IBM OS/400, IBM S/390, OS/2, Citrix, HP-UX, AIX, MPE/iX y Open VMS. Además, posee soporte para bases de datos del tipo Microsoft SQL Server, Oracle, IBM DB/2, Adabas, Informix y Sybase ASE.

Otra característica es que la implementación de programas y funciones se puede realizar en un lenguaje orientado a objetos (por ejemplo: Java) o, siendo más habitual este modo, realizarlo con el propio lenguaje de cuarta generación del sistema, ABAP, a través del entorno de programación que incluye el Sistema: ABAP Workbench.

1.3.5 ABAP IV

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...).

ABAP fue desarrollado por SAP como lenguaje de informes para SAP R/2, en los años 80, una plataforma que permitía a las grandes corporaciones construir aplicaciones de negocios para gestión de materiales y finanzas.

En sus inicios, ABAP estuvo basado en COBOL, ya que este era el paradigma de lenguaje orientado a negocios; además incluía el concepto de Bases de datos lógicas, que suministraba un alto nivel de abstracción para el acceso a bases de datos.

ABAP fue pensado como un lenguaje de programación para que los usuarios finales pudieran manipular la información, pero el 4GL (Lenguaje de cuarta Generación) se fue volviendo demasiado complicado para usuarios normales, por lo que se requieren programadores con una cierta experiencia en esta tecnología para realizar desarrollos.

ABAP se mantuvo como el lenguaje de desarrollo para la siguiente versión cliente-servidor de SAP R/3, que fue lanzada al mercado en 1992, en el que casi todo el

sistema, menos las llamadas al sistema básicas estaban escritas en ABAP. En 1999, con el lanzamiento de la versión 4.6 de R/3, SAP lanzó una extensión orientada a objetos denominada ABAP Objects, lo que permite la compatibilidad con lenguajes de programación orientados a objetos como Java.

La comunicación con bases de datos se realiza a través de una interfaz que transforma las sentencias escritas en Open SQL en el SQL Nativo del correspondiente DBMS. Además, esta interfaz proporciona servicios muy útiles como el buffering en memoria local del servidor de tablas y datos recientes o frecuentemente solicitados.

Sintaxis

Aunque las sentencias de este lenguaje de programación son similares a las de COBOL, estas cuentan con cierta variedad de opciones, característica que hace que el conjunto total de sentencias sea bastante amplio, pudiendo dividirse en los siguientes seis grupos:

- Sentencias Declarativas, usadas para definir tipos y variables.
- Sentencias de Modularización, usadas para definir los bloques³ de proceso.
- Sentencias de Control, usadas para establecer el flujo de ejecución del programa en un evento.
- Sentencias de Llamadas, usadas para realizar la llamada a otros bloques.
- Sentencias Operativas: usadas para modificar el contenido de las variables, incluyendo las sentencias Open SQL y las destinadas para manejo de tablas internas.
- Sentencias de Formato, usadas para dar formato a la salida de datos del programa.

Anteriormente, se ha hecho mención a las tablas internas; se trata de una de las funcionalidades más prácticas del lenguaje ABAP, y que ofrece mayor potencia a la hora de trabajar con datos. Las tablas internas son variables que permiten almacenar registros en memoria, pudiendo considerar componente en una línea como una columna en una tabla interna.

Se tratan de estructuras de datos declaradas durante la implementación del programa para adaptarse a las necesidades del programador, y que permiten el almacenamiento de datos temporalmente en la memoria de trabajo. Su importancia radica en que este almacenamiento permite una mayor velocidad de computación y tratamiento de datos, además de ser unas herramientas muy prácticas para incluir estructuras complejas en un programa. Otra aplicación que se le puede dar a las tablas internas es su utilización como tipos de datos.

³ Un bloque es un código llamado (subrutinas, funciones, métodos, etc.) o código ejecutable ante un evento.

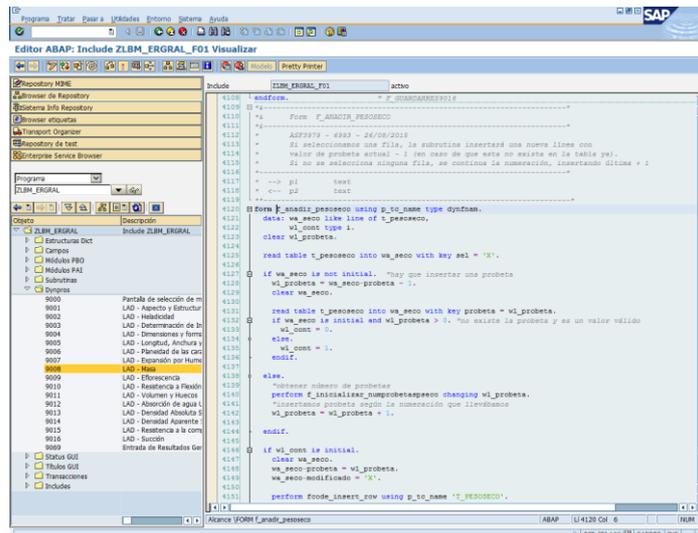


Ilustración 10 - Entorno de programación ABAP

Tipos de programas

Como en otros lenguajes de programación, un programa ABAP no es simplemente una unidad ejecutable o una biblioteca, sino que proporciona código ejecutable para otros programas no ejecutado independientemente. Por lo tanto, podemos dividir estas dos clases siguiendo criterios de funcionalidad.

- Programas Ejecutables, mayoritariamente asociados a transacciones⁴.
 - Informes (Reports).
Los informes siguen un modelo de programación relativamente simple, dónde el usuario introduce una serie de parámetros y el programa los usa para producir un informe en forma de lista interactiva.
 - Module Pool.
Los module pools definen unos patrones más complejos de interacción con el usuario a través de una colección de pantallas, cada una con su propia lógica.
- Programas no ejecutables, que funcionan como apoyo o soporte del resto.
 - INCLUDEs: Usados para modularizar programas muy largos.
 - Subrutinas: Contiene subrutinas ABAP, trozos de código que se suelen repetir con frecuencia.
 - Grupos de Funciones: Son librerías de módulos de funciones, visibles desde cualquier programa.
 - Clases de Objetos: Representan la orientación a objetos junto a las interfaces.
 - Interfaces: Definiciones de métodos para los cuales las clases deben ofrecer el código.
 - Tipos: Colección de datos, tipos y constantes.

⁴ En terminología SAP, se denomina transacción a la ejecución de un programa.

En ABAP, las pantallas (mencionadas anteriormente) son la interfaz gráfica que comunica el Sistema con el usuario. Cada pantalla cuenta con un “flujo lógico”, el cual se refiere al código implícito de ABAP invocado por las pantallas, formando lo que se denomina ‘DynPro’ (Dynamic Program, Programa Dinámico). Cada pantalla tiene su propio flujo lógico, el cual se divide en:

- PBO (Process Before Output), eventos que se producen antes de mostrar la pantalla;
- y PAI (Process After Input), eventos que se producen tras una acción del usuario.

Un mismo programa puede contar con varios DynPros, permitiéndonos navegar entre las distintas ventanas o recargar los datos de la que nos encontremos, dependiendo de la acción que queramos llevar a cabo.

Una transacción en terminología SAP es la ejecución de un programa, ya sea estándar del Sistema o desarrollados con posterioridad. La forma normal de ejecutar código ABAP en el sistema SAP es ingresando un código de transacción. Por ejemplo, SE11 es el código de transacción que llama al Diccionario de Datos.

Las Transacciones pueden ser llamadas a través de menús definidos por el sistema, especificados por el usuario o basados en roles. También pueden ser invocados mediante el ingreso del código de transacción directamente en el campo de comandos, el cual está presente en todas las pantallas SAP. Las transacciones también pueden ser invocadas por código mediante las sentencias ABAP “CALL TRANSACTION” y “LEAVE TO TRANSACTION”.

ABAP Workench

Es el espacio de trabajo para el desarrollo de programas en SAP. No se trata únicamente de un editor de código fuente, sino que cuenta con varias herramientas que complementan la edición de código, como el ‘Screen Painter’ (utilizado para diseñar las pantallas gráficamente), ‘Function Builder’ (utilizado para crear grupos de funciones) o el ‘ABAP Dictionary’ (desde el cual gestionar las tablas, tipos de datos, dominios, ayudas de búsqueda, etc.), entre otras.

El diccionario ABAP contiene todos los metadatos acerca del sistema SAP. Está muy ligado con la plataforma de trabajo de ABAP en el que cualquier referencia a los datos se obtiene del diccionario (por ejemplo: tablas, vistas, tipos de datos,...).

Los tipos de datos más importantes del diccionario son los siguientes:

- Tablas: son contenedores de los datos que existen en la base de datos relacional subyacente.
- Índices: proporcionan accesos rápidos a los datos de las tablas para aquellas selecciones usadas con mayor frecuencia.
- Vistas: se tratan de tablas virtuales que no contienen ningún dato físicamente, sino que definen un subconjunto de columnas de una o más tablas usando condiciones ‘join’.
- Estructuras: son tipos de datos que contienen varios campos.

- Elementos de datos: proporcionan contenido semántico a una tabla o campo estructurado.
- Dominios: definen las características estructurales de los elementos anteriormente mencionados (número de decimales de un tipo numérico, cantidad de caracteres de una cadena,...). Los dominios también pueden proporcionar contenido semántico, proporcionando una posible lista de valores.
- Ayudas de búsqueda: proporciona una estrategia de búsqueda avanzada cuando un usuario quiere ver el posible valor que tiene un campo.

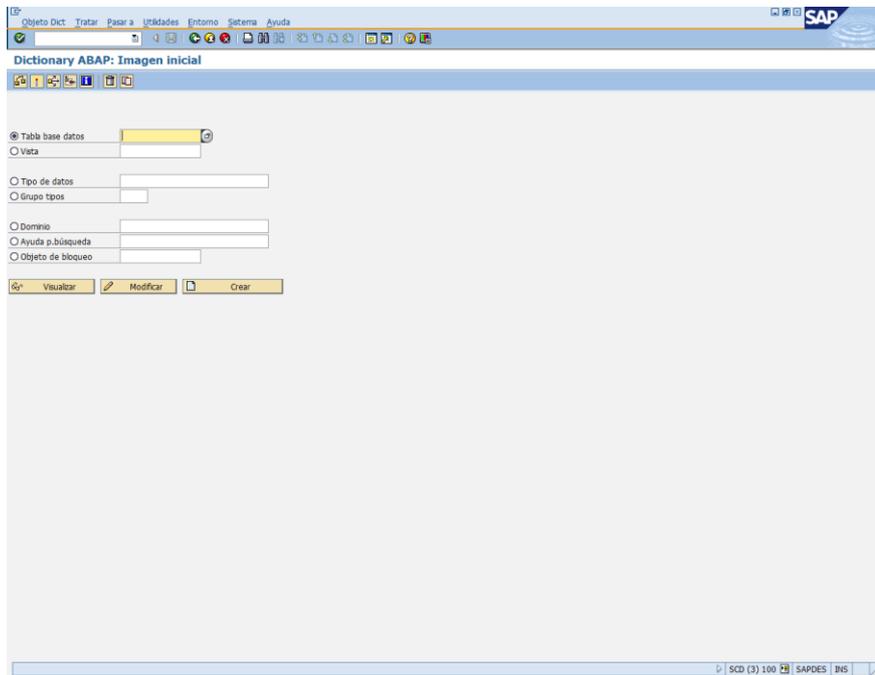


Ilustración 11 - Diccionario ABAP

1.3.6 Otras herramientas

Además de los elementos Software descritos anteriormente, se han usado las siguientes herramientas durante el desarrollo de este Trabajo:

- Microsoft Office: Word, Excel y Visio. Estas herramientas han sido usadas para realizar la documentación del Trabajo, elaboración de informes y diagramas y de apoyo a la hora de realizar y comprobar los resultados de los ensayos.
- Snagit. Se trata de un programa de captura de pantalla y editor de imágenes, que permite también capturas de video.
- Citrix: El acceso a SAP NetWeaver se ha realizado a través de un servidor con el sistema operativo Citrix, lo que permitía un acceso remoto a la herramienta sin necesidad de tenerla instalada en el equipo.
- Gliffy: Herramienta on-line para la creación de diagramas de flujo, UML, etc.
- Draw.io: otra herramienta on-line para creación de diagramas.

1.4 Organización del documento

Esta memoria se ha elaborado siguiendo el esquema de desarrollo anteriormente explicado; de esta forma, cada capítulo representará una fase del proceso, diferenciando claramente las distintas etapas y simplificando la comprensión de las tareas y procesos realizados en cada una de ellas.

El modelo de planificación descrito se asemeja al proceso de desarrollo en cascada, debido a que no se debía comenzar ninguna fase nueva del proyecto hasta que se hubiese finalizado la anterior. Esta decisión fue tomada para minimizar el impacto del nuevo desarrollo en el Sistema. Aunque este modelo pueda parecer bastante rígido y poco favorable a cambios, se ha tenido que adaptar a estos ya que, a lo largo del desarrollo, han surgido nuevos requisitos o mejoras de funcionalidad, llegándose a mezclar tareas de distintas fases en algunos momentos del desarrollo.

1. Introducción: en este capítulo se expone el contexto del trabajo realizado, los objetivos y necesidades que lo han hecho posible y una introducción a las tecnologías usadas para este fin.
2. Preparación del trabajo: en este capítulo se analizan las tareas realizadas al comienzo del proyecto, como la elección de la temática del Trabajo Fin de Grado o el estudio de la documentación necesaria antes de iniciar la elaboración del proyecto.
3. Análisis y Diseño: este capítulo está formado por fases como la recopilación de requisitos, el estudio de casos de usos, una visión global de la estructura del proyecto y una primera propuesta del modelo de datos.
4. Implementación: en este capítulo se explicará el desarrollo realizado y se mostrarán ejemplos del código desarrollado.
5. Pruebas e Implantación: en este capítulo se describen los métodos de testado y validación que se han utilizado para comprobar el correcto funcionamiento de las aplicaciones, así como los pasos a seguir para implantar finalmente los módulos en el sistema y puedan ser usados por los usuarios.
6. Cambios: en este capítulo se detallan los cambios que han surgido durante el desarrollo y la solución adoptada.
7. Conclusiones: este capítulo, el último de la memoria, servirá de reflexión sobre los resultados obtenidos, tanto desde un punto de vista profesional como personal.

2.Preparación del trabajo

2.1 Elección de la temática del TFG

La elección de este Trabajo Fin de Grado se debió en gran parte a un motivo fundamental: durante la época en la que debía empezar a realizarlo, me encontraba trabajando para una empresa de Control de Calidad de Materiales en calidad de becario. Desde la empresa, surgió la oportunidad de utilizar el trabajo allí desarrollado para la realización del TFG y propusieron este proyecto. El hecho de que esta empresa utilizara las tecnologías SAP descritas en el capítulo anterior fue otra de las razones que influyeron en la decisión, ya que serviría para continuar aprendiendo y profundizando en esta tecnología de SAP, muy demandada en la actualidad.

Tras valorar esta opción, el siguiente paso fue encontrar un tutor académico que accediera a la labor de dirigir el Trabajo y confirmar con la Escuela que no existiría ningún problema en cuanto a la normativa del Trabajo. Además, el tutor contaba con experiencia en este tema ya que había dirigido hace unos años el Proyecto Fin de Carrera de un alumno, realizado en la misma empresa y con una temática similar.

Tras la obtención de la aprobación de todas las partes, se iniciaron los trámites académicos previos al desarrollo del Trabajo.

2.2 Recopilación de documentación

Como se ha explicado en el capítulo anterior, la certificación de calidad de los materiales viene dada a partir de una serie de normas, las normas UNE.

Esta empresa, al igual que cualquier otra empresa del sector de Calidad que deseen aplicar esta garantía, deben comprar las normas a una entidad perteneciente al CTN. En este caso, desde Laboratorio se proporcionaron un conjunto de normas distribuidas por AENOR, una de las diez certificadoras más importantes del mundo. Estas normas son la base de las operaciones que se realizan en la Entrada de Resultados, ya que deben ser capaces de calificar la muestra según los datos que se introduzcan en la pantalla; es por ello por lo que su estudio es obligatorio y que fueron usadas continuamente durante las fases de desarrollo y testado.

Además de estas normas, fueron proporcionadas las plantillas de cálculos e informes, así como borradores y ejemplos de informes realizados.

Aparte, se debía analizar también los documentos elaborados en proyectos anteriores, de forma que este proyecto se ajustase a los anteriores, tanto en el desarrollo como en la documentación.

Con todos estos documentos y los requerimientos impuestos tanto desde el Departamento de Sistemas de Información como Laboratorio, se inició el proceso de análisis y definición previo al desarrollo.

3.Análisis y Diseño

En este capítulo se explicará los pasos que se han seguido y las tareas que se han realizado desde la primera visión del proyecto que se tuvo tras las conversaciones con los técnicos de laboratorio y el responsable del departamento de Sistemas de Información. Estas tareas se estructuran en una serie de fases, comunes a cualquier proyecto de Ingeniería Software, proporcionará la información necesaria para un correcto desarrollo del producto.

Este capítulo es de vital importancia, ya que serán los cimientos sobre los que se construirá el producto final y delimitará el alcance real del proyecto. Un correcto estudio de requisitos y casos de uso proporcionará una amplia visión del producto, pudiendo diseñar el modelo de datos antes de comenzar el desarrollo software, facilitando esta fase.

Aun así, durante el desarrollo de este proyecto surgieron diversas dudas, funcionales y de diseño. Para ello, desde el departamento de Sistemas de Información se creó un foro que permitiese el contacto directo con los técnicos de desarrollo, para aquellas cuestiones técnicas que debiesen resolver desde allí. De esta forma, formarían parte del desarrollo bajo el rol de cliente y darían su visión del sistema, tanto en requisitos como su opinión a las soluciones diseñadas.

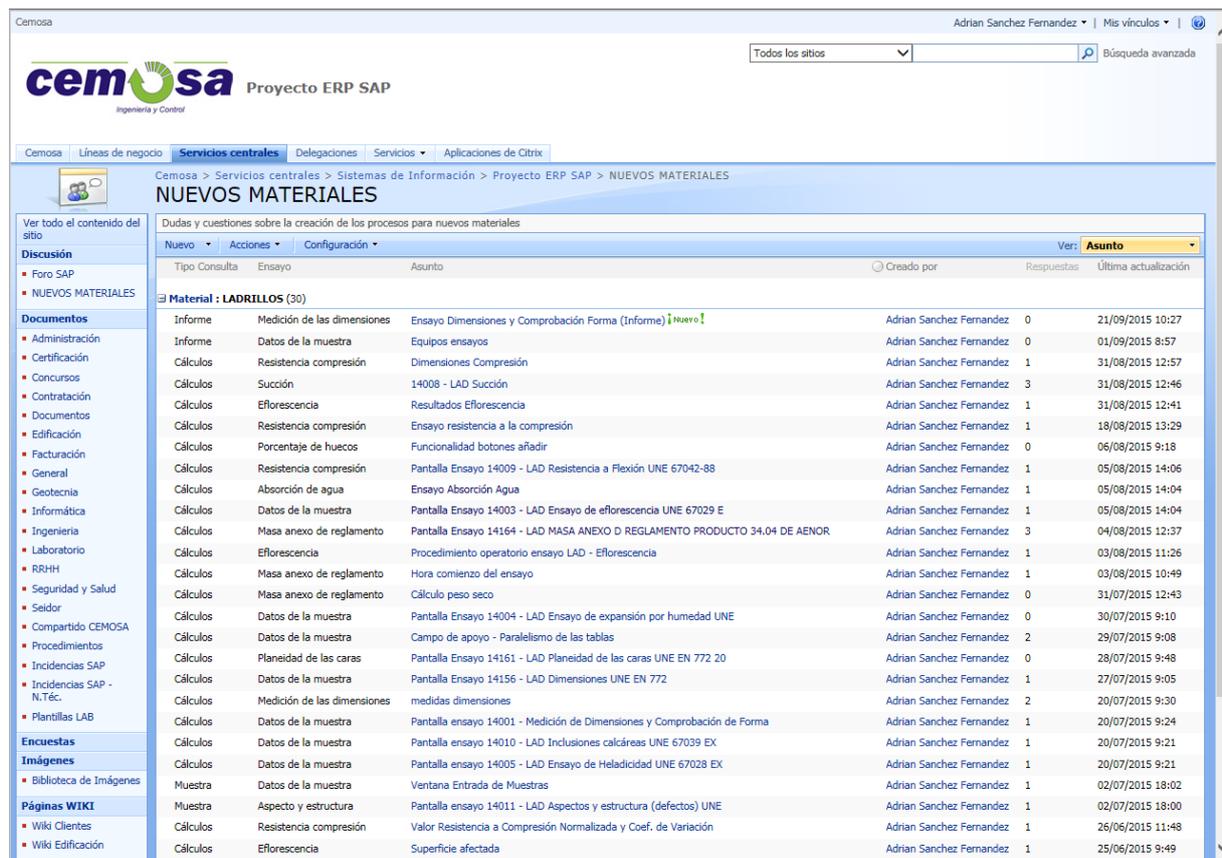


Ilustración 12 - Foro de dudas

3.1 Análisis de los Requisitos

En esta fase, a través de las entrevistas realizadas con los clientes (técnicos de laboratorio) y el responsable de desarrollo, se realizó una lista de los requisitos que el Sistema debía proporcionar.

Estos requisitos se han clasificado en dos tipos: Requisitos funcionales y Requisitos no funcionales.

Los requisitos funcionales son aquellos establecen los diferentes comportamientos que puede tener el Sistema; es decir, describen un conjunto de entradas, su comportamiento y su salida (por ejemplo, cálculos, detalles técnicos, manipulación de datos u otras funcionalidades específicas que se supone que el Sistema debe cumplir).

Por otro lado, deben existir una serie de requisitos que, aunque no describan la información a guardar ni las funciones a realizar, puedan especificar diferentes criterios que puedan usarse para juzgar la operación del Sistema. Estos son los requisitos no funcionales; algunos ejemplos de este tipo de requisitos son el rendimiento, disponibilidad, mantenibilidad, etc.

Se han clasificado estos requisitos según el módulo al que pertenezcan: Entrada de Muestras, Entrada de Resultados o Informes.

- Entrada de Muestras
 - RF-EM1: Gestionar los datos de una muestra.
 - RF-EM1.1: Almacenar datos de una nueva muestra
Este requisito cuenta con una serie de especificaciones:
 - Se deben añadir nuevos campos propios del material Ladrillo.
En este caso, se debe añadir el campo Designación, con sus distintas opciones, y dar soporte a otros campos como Fabricantes o Métodos de Muestreo, operativos para otros materiales pero no para Ladrillos.
 - Se debe identificar unívocamente una muestra a través de un Código de Muestra (tal y como ocurre con otros materiales).
 - Se debe asociar una muestra con su expediente y los servicios asignados.
 - RF-EM2: Asignar servicios a cada muestra.
 - RF-EM3: Asignar documentos a la muestra.
Ya sean borradores o informes.
- Entrada de Resultados
 - RF-ER1: Gestionar los datos de la muestra.
 - RF-ER1.1: Almacenar los datos de un ensayo.
 - RF-ER1.2: Modificar los datos de un ensayo.

- RF-ER1.3: Llevar un control de auditoría, que indique cuando se creó y/o modificó un registro en la tabla donde se guardan los resultados, además del usuario que ha realizado este cambio.
 - RF-ER2: Generar Resultados de los Ensayos.
 - RF-ER2.1: El sistema debe indicar si los resultados son correctos o se ha producido un error.
 - RF-ER2.2: Controlar excepciones en los cálculos (por ejemplo: división por cero) e indicar mediante un mensaje el motivo de la excepción.
- Informes
 - RF-IN1: Gestionar Informes
 - RF-IN1.1: Creación de informe a partir de los datos de la muestra y los resultados.
 - RF-IN1.2: Almacenar Informe en el Sistema.
 - RF-IN2: Compatibilidad con herramientas de Microsoft Office.
El formato numérico de SAP utiliza el carácter coma para la separación de millares, y el carácter punto para los valores decimales; si en un futuro se requiriese obtener un informe en formato Excel en vez de Word, existiría una incompatibilidad de datos, ya que Excel utiliza la coma para separar los valores enteros y decimales.

En cuanto a los requisitos no funcionales, se han extraído un conjunto de ellos que, al ser comunes a todos los módulos, no existe necesidad de clasificarlos; son los siguientes:

- RNF-1: Varios usuarios pueden acceder a la vez al Sistema.
- RNF-2: El expediente debe adjuntarse a la muestra, siendo posible su visualización cuando se requiera.
- RNF-3: El informe debe contener la información establecida por la norma de cada ensayo, y sus resultados deben mostrarse según las indicaciones de dicha norma.
- RNF-4: Generar copias del informe para entregar a los diferentes interlocutores⁵.
- RNF-5: Las pantallas de Entradas de Muestras deben ser lo más parecidas posibles a los borradores de los Técnicos de Laboratorio.
- RNF-6: Evitar la creación y uso de macros en la tarea de generación de informes.
- RNF-7: La Entrada de Muestras de Ladrillos debe crearse a partir de otra ya existente en el Sistema, denominada Entrada de Muestras General.

⁵ Un interlocutor es cualquiera de las partes interesadas en una contratación (empresa, cliente, contratista, supervisores, fabricantes, etc.).

Además, existe un requisito común a todos y es el más importante de ellos de cara a la empresa, y por eso se indica aparte: El Sistema debe aprovechar todas las funcionalidades ya existentes, tanto los elementos ya existentes en el Diccionario de Datos como módulos o funciones existentes en el Sistema para otro tipo de materiales, así como imitar la filosofía de programación de estos programas en la medida de lo posible.

3.2 Casos de Uso

Se denomina como Casos de Uso a las secuencias de interacciones que se desarrollan entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Se pueden considerar una herramienta simple para describir el comportamiento del software, describiendo las maneras en las que los actores pueden trabajar con el sistema sin describir la funcionalidad interna del mismo ni su implementación, por lo que nos proporcionan una visión global y sencilla del trabajo a realizar.

Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

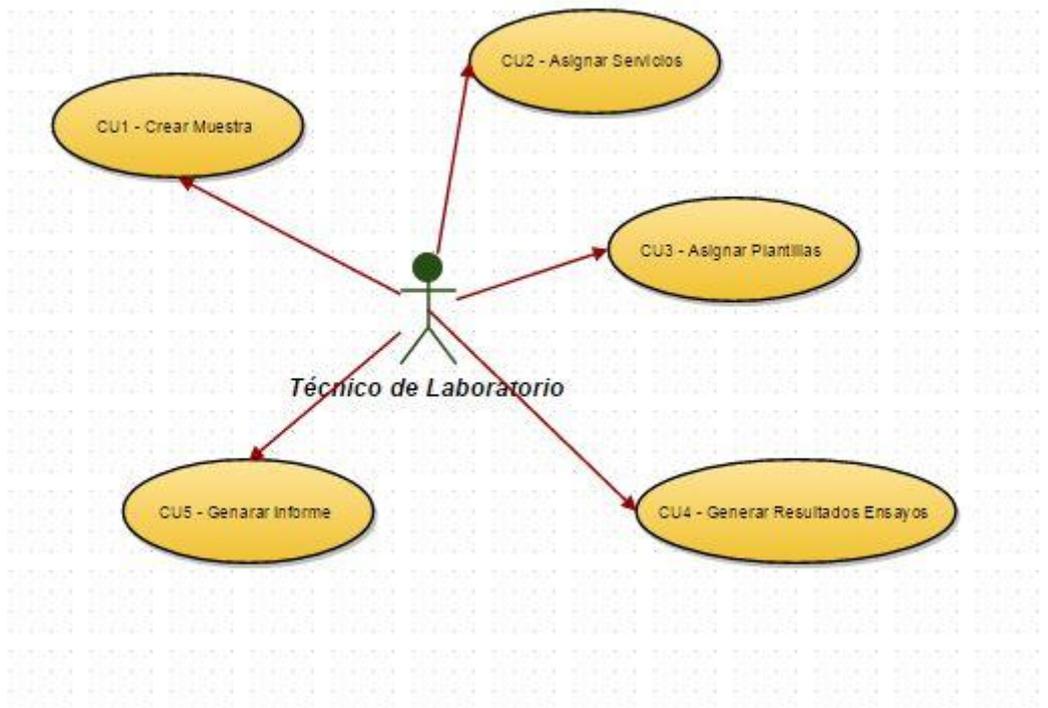


Ilustración 13 - Diagrama de Casos de Uso

A continuación se representan los diferentes diagramas de secuencia de los cada uno de los casos de uso que se han indicado anteriormente, indicando como interaccionan cada uno de los objetos del Sistema en cada Caso de Uso.

- Entrada de Muestras
 - CU1 – Crear Muestra

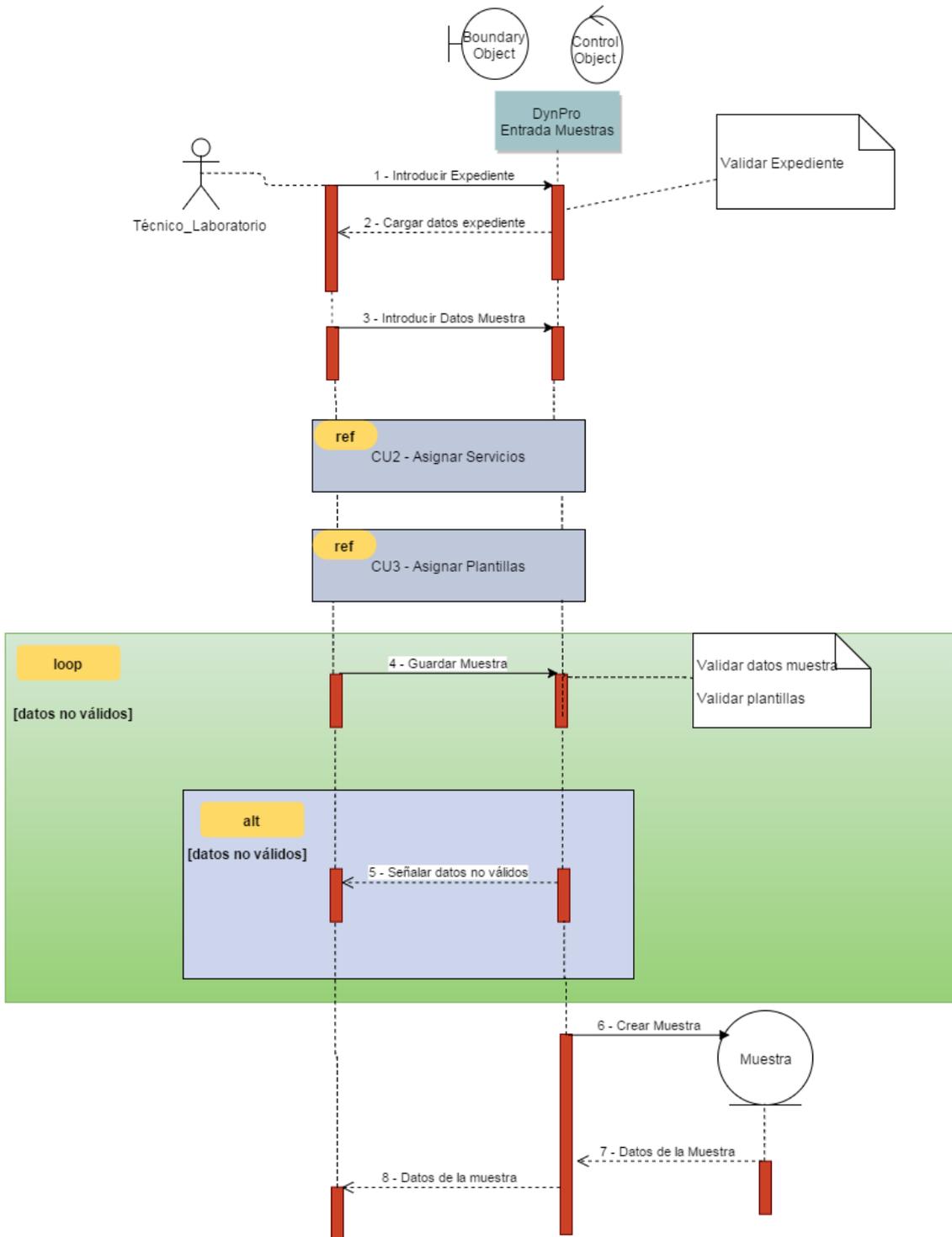


Ilustración 14 - Diagrama de Secuencia CU1

○ CU2 – Asignar Servicios

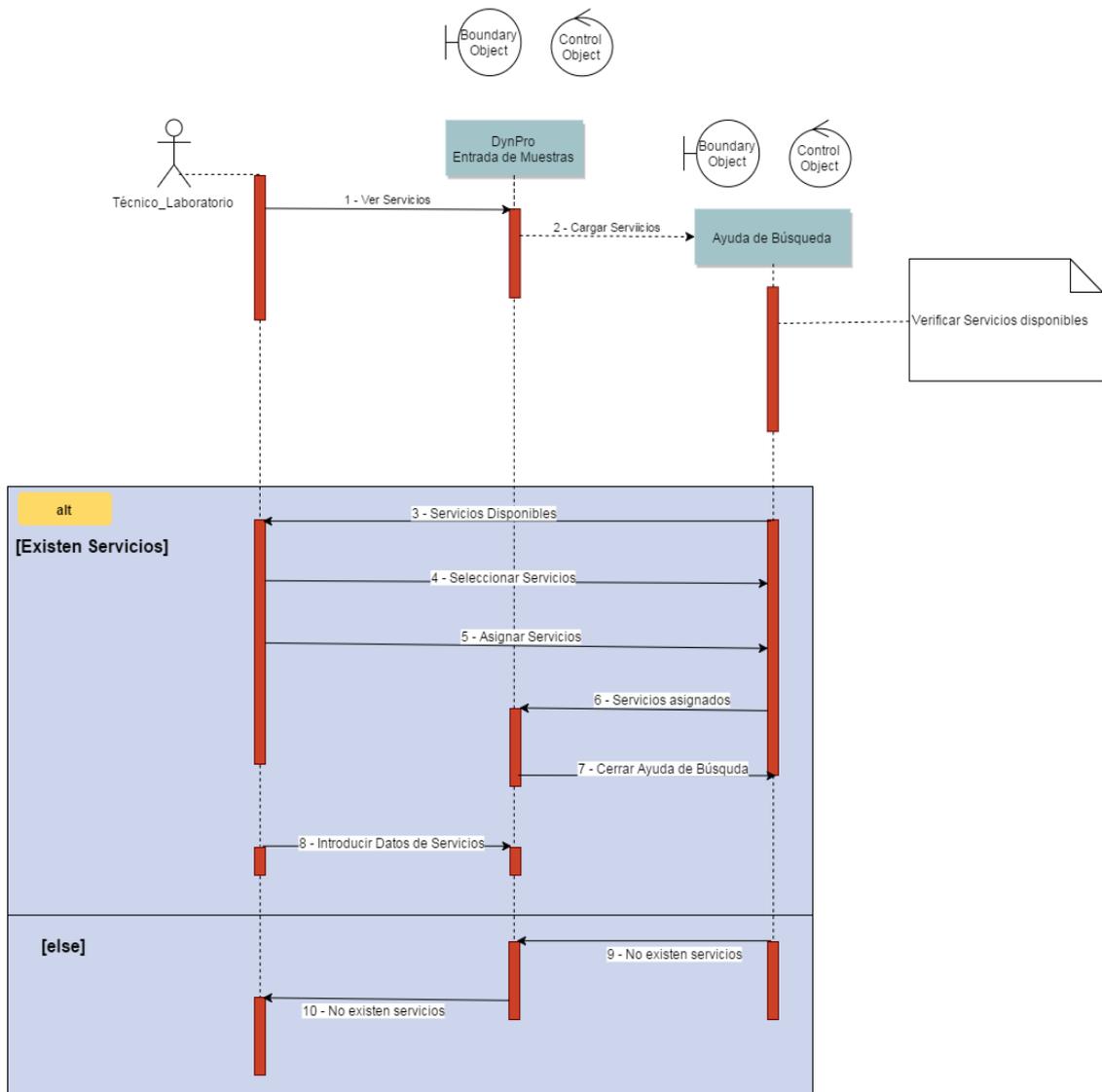


Ilustración 15 - Diagrama de Secuencia CU2

○ CU3 – Asignar Plantillas

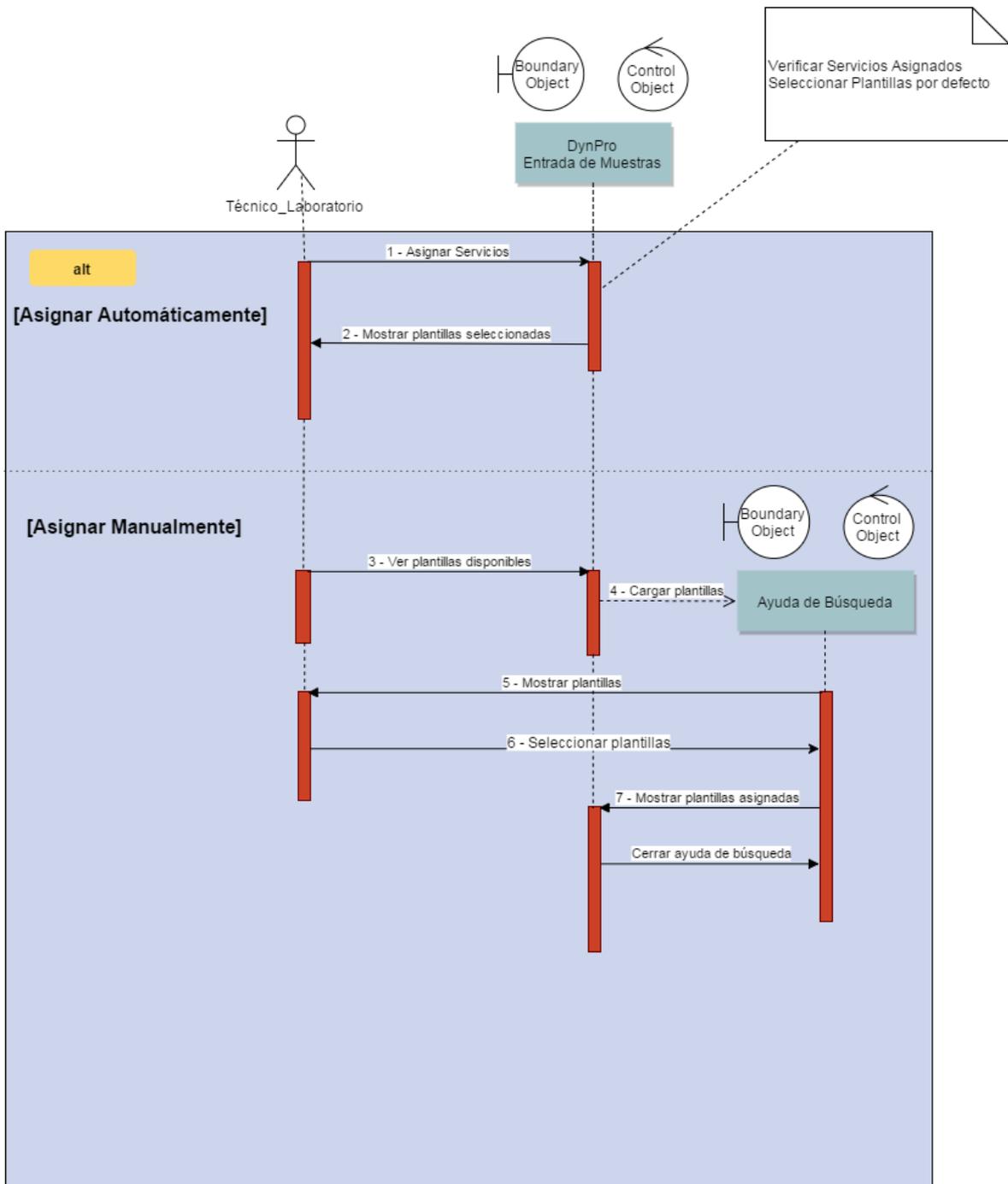


Ilustración 16 - Diagrama de Secuencia CU3

- Entrada de Resultados

- CU4 – Generar resultados Ensayos

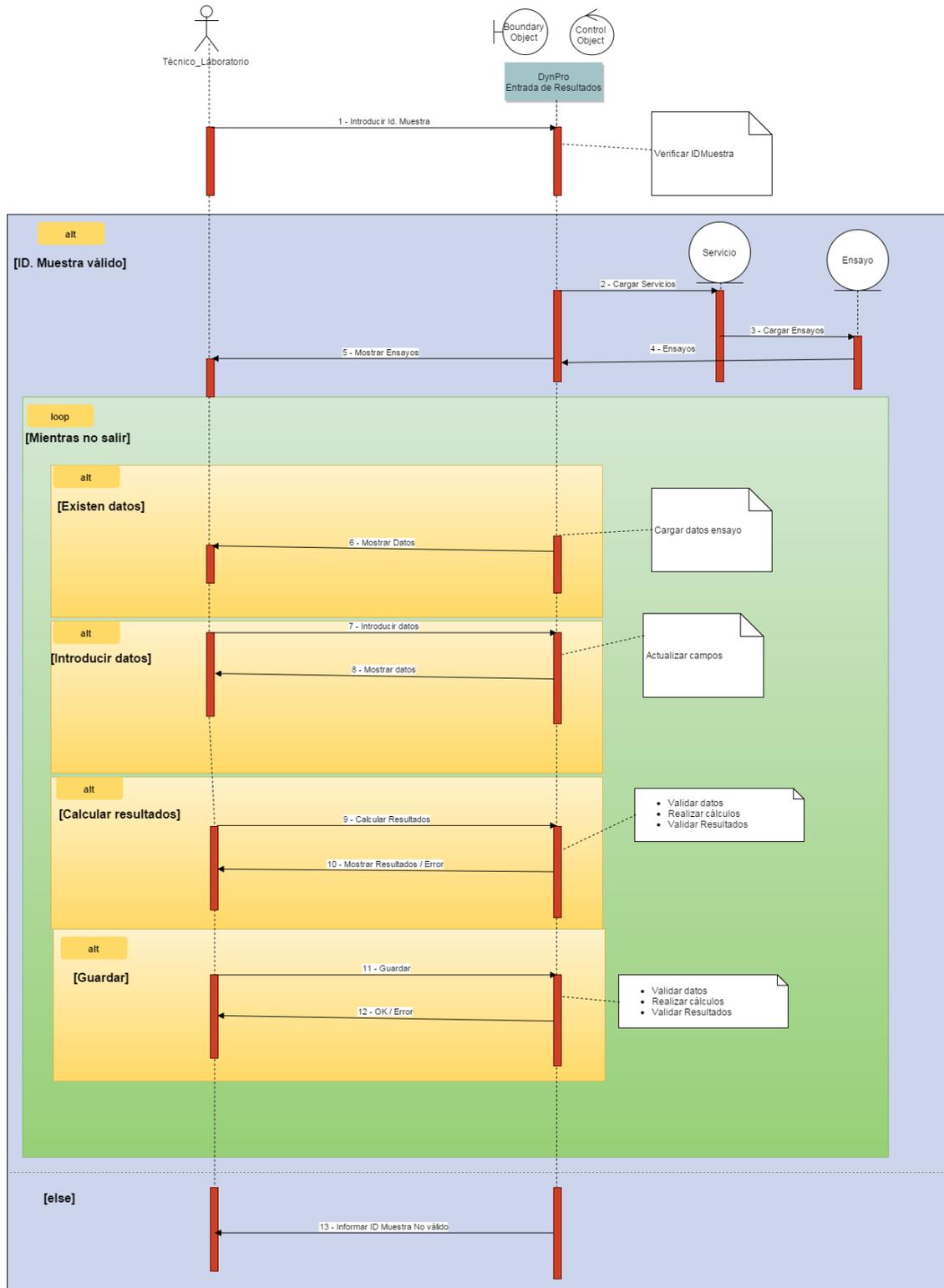


Ilustración 17 - Diagrama de Secuencia CU4

- Generación de Informes
 - CU5 – Generar Informe

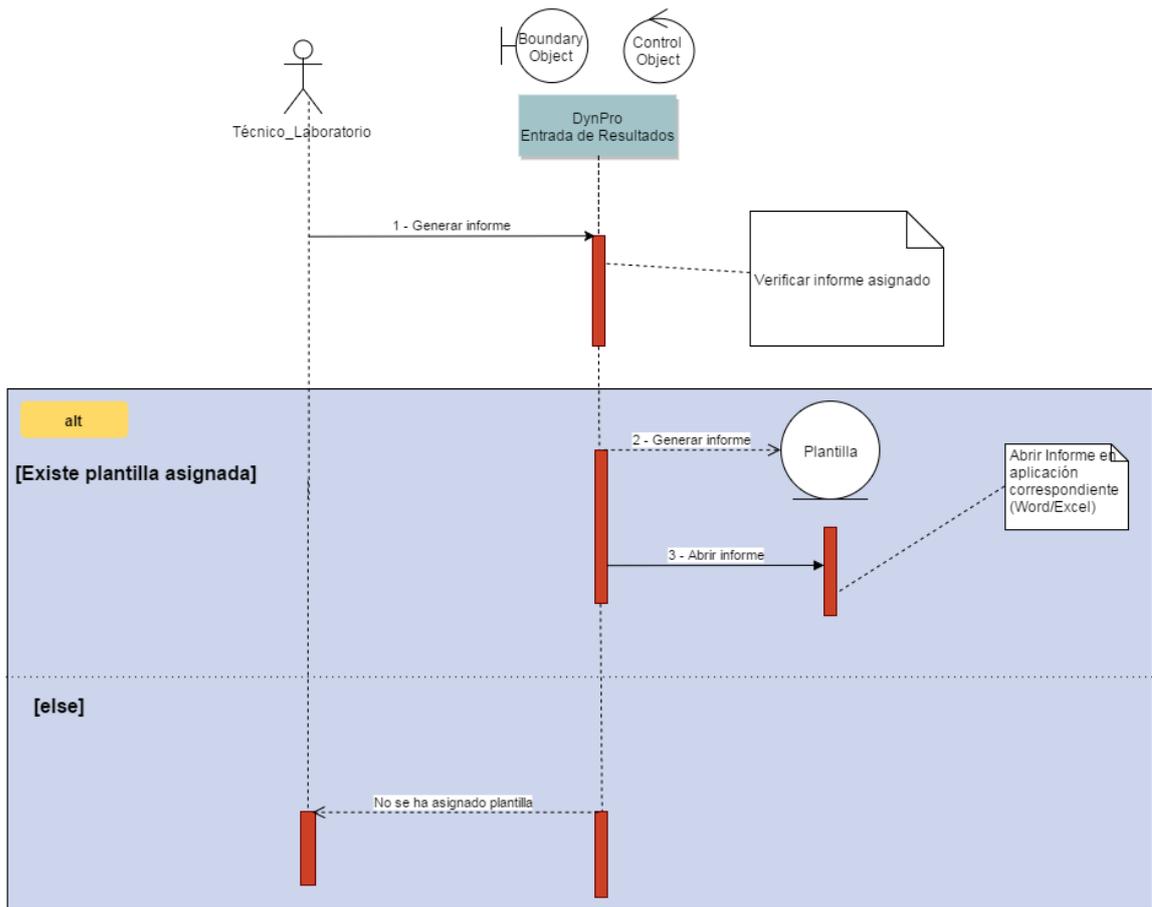


Ilustración 18 - Diagrama de Secuencia CU5

3.3 Diseño

Una vez estudiados los requisitos y casos de uso, la siguiente tarea es encontrar una solución visual que de soporte a las especificaciones dadas y que nos permita diseñar el modelo de datos adecuadamente y de forma sencilla.

Todas las pantallas de ensayos han sido aprobadas por los técnicos de laboratorio (en calidad de cliente) antes de proceder a la fase de desarrollo, sugiriendo en ocasiones algunos cambios o modificaciones.

3.3.1 Entrada de Muestras

A partir de la Entrada de Muestras General, la nueva Entrada de Muestras para Ladrillos debe dar soporte a las nuevas características de este material.

En este caso, bastará con añadir un nuevo campo para la Designación del Ladrillo, ya que el resto de campos se encuentran creados en las tablas y tipos existentes en el Diccionario ABAP, y crear o modificar su comportamiento para adaptarse a este nuevo material será una de las tareas de la fase de desarrollo.

The screenshot shows the SAP 'Entrada Hojas Servicio 0 (General)' interface. It features a top navigation bar with 'Sistema' and 'Ayuda' options. Below this is a toolbar with various icons. The main content area is divided into several sections:

- Header Section:** Contains fields for 'Expediente', 'Obra', 'Cliente', 'Muestra' (value: 1), 'Muestra Origen', 'Aviso N°', and 'F. Pedido'.
- Datos de Toma:** Includes 'N° Albarán', 'Muestras Albarán' (1/1), 'Fecha-hora toma', 'Duración' (MIN), 'Operador', 'Realizada por', 'F. Entrada' (22.09.2015), 'Tipo Informe' (MUEI0SSCA), and '¿Incluye plano?'.
- Datos de Muestra:** Contains 'Tipo material' (35 Ladrillo), 'Descripción', 'Estado muestra' (Programado), 'Designación', 'Lote CEMOSA', 'Lugar toma', 'Norma Muestreo', 'Método', 'Fabricante', and 'Cntd / Unds'.
- Servicios:** A table with columns: Código, Descripción, Operador, Técnico, Fecha Prevista, Lab Destino, Mues Destino, Inicio Real, and Obj.

The bottom of the screen shows a status bar with 'Asinnar Plantillas', 'Plantilla Borrador', 'Plantilla Informe', and system information: 'SCD (3) 100', 'SAPDES', 'INS'.

Ilustración 19 - Nueva Entrada de Muestras

3.3.2 Entrada de Resultados

Al igual que en la Entrada de Muestras, se nos proporciona una pantalla base sobre la que se creará la nueva funcionalidad; en este caso, la Entrada de Resultados sólo cuenta con los campos que identifican la muestra, datos generales del ensayo (fecha, técnico que lo ha realizado, observaciones, ...) y datos de auditoría. Uno de los objetivos principales a la hora de diseñar estas nuevas pantallas fue la de que fuesen lo más similares a las plantillas de Laboratorio, para así facilitar la tarea a los técnicos, ya acostumbrados al diseño de las mismas.

Sistema Ayuda

Entrada de Resultados 0 (General)

Datos de cabecera

Expediente	Muestra	R.L.
Obra	Muestra O	R.L.
Cliente	Estado muestra	Téc. Rev.
Contratista	Tipo Material	
Observación		
Servicio		
F.Inicio Servicio	F.Fin Servicio	Fecha Revisión
		Fecha de Pedido
		F.Fin Mues.

Resultado

Resultado de ensayo

Datos Auditoria

Creado el	Creado por	Última modif.	Modificado por
-----------	------------	---------------	----------------

Ilustración 20 - Entrada de Resultados General

Se realizará una pantalla de Entrada de Resultados por cada Ensayo que se realice para Ladrillos. Según las acciones a realizar para llevar a cabo el ensayo, la entrada variará y tendrá un tipo distinto de elementos para la entrada o salida de datos; en algunos bastará con una serie de campos mientras que otros necesitarán de tablas, casillas de verificación o listas despegables. Además, se deberán incluir campos para seleccionar los equipos empleados en las mediciones.

A continuación se citan los servicios que se ofertan y los ensayos que se van a realizar junto a una descripción y su diseño correspondiente; en los primeros se indica también como se realiza el ensayo para se pueda apreciar la idea del diseño. La nomenclatura que se ha elegido ha sido la del código de servicio asociado al ensayo, nombre del ensayo y norma o normas asociadas, por lo que se aconseja la lectura de dicha norma en caso de querer ampliar o consultar más información.

1. 14001 – Medición de las dimensiones y comprobación de la forma – UNE 67030:85, UNE 67030:86.

Se toman una serie de piezas del material cuya muestra se va a evaluar, de ahora en adelante denominadas como probetas, y se miden la longitud de sus aristas: sogá (arista mayor), tizón (arista media) y grueso (arista menor).

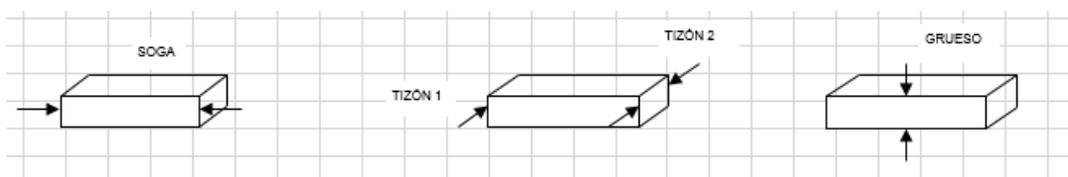


Ilustración 21 - Aristas de un Ladrillo

Como se puede apreciar en la imagen, para la arista denominada Tizón se deben realizar dos mediciones, calculando la media de las mismas; esto se realiza para conocer si existen diferencias en alguna de las dos dimensiones para una misma probeta.

Con estas mediciones, y por cada arista, se calcula la media de las mediciones (Valor Medio) y junto al Valor Nominal de la arista en cuestión, se calcula la tolerancia o desviación sobre el valor nominal (valor absoluto de la diferencia entre el valor medio y el nominal de la arista) y la dispersión (valor absoluto máximo de la diferencia entre el valor medio de una dimensión de la probeta y cada valor aislado de la misma).

Posteriormente, dependiendo del tipo de ladrillo que se tenga, se deben realizar mediciones sobre cada una de las caras del ladrillo: Tabla (cara mayor), Canto (cara intermedia) y Testa (cara menor).

- Ladrillo tipo V (Visto⁶): se debe medir las flechas en las diagonales de una tabla, un canto y dos testas.
- Ladrillo tipo NV (No Visto⁷): se debe medir una flecha en las diagonales de la tabla, canto y testa.

A partir de estas mediciones, se debe clasificar la longitud de cada cara en 3 categorías:

- $L \leq 25$
- $30 \geq L > 25$
- $L > 30$

Por lo tanto, se ha decidido crear una tabla donde se introducirán las mediciones para cada probeta y obtener los resultados en campos externos. Mediante un listado despegable se indicará la forma del ladrillo y se incluirán una serie de botones que permitirán añadir o eliminar filas en la tabla.

⁶ Ladrillo Visto es aquel que no se recubre con ningún acabado para su uso.

⁷ Ladrillo No Visto es aquel que va a quedar oculto a la hora de usarlo.

En la parte superior, deberán existir botones para realizar los cálculos, guardar los datos del ensayo en la base de datos, cambiar a otro ensayo, crear y adjuntar informe a la muestra y poder consultar los informes pertenecientes al expediente.

Ilustración 22 - E.R. del ensayo Dimensión y Comprobación de Forma

2. 14002 – Determinación de la absorción de agua – UNE-EN 771-2003.

Antes de iniciar el ensayo, se debe sumergir cada probeta (ladrillo) en agua durante 24h; de esta forma obtendremos el valor de Ladrillo Húmedo.

Posteriormente, se debe calcular el peso seco de cada probeta; para ello, se secan los ladrillos húmedos en una estufa a $105 \pm 5^\circ\text{C}$ hasta masa constante (se considera que la masa es constante si la diferencia entre dos pesadas a menos de veinticuatro horas no supera el 0.2%).

Con ambos valores, se calcula el porcentaje de absorción de agua de cada probeta de la siguiente forma:

$$\% \text{ abs} = \frac{L. \text{Húmedo} - L. \text{Seco}}{L. \text{Seco}} * 100$$

3. 14003 – Ensayo de eflorescencia – UNE 67029:95 EX.

Entrada de Resultados 0 (General)

Datos de cabecera

Expediente: 0/1500035/1 Oferta Ladrillos Muestra: 11304 R.L.: 0201 1 2012 684
 Obra: 019002 CONTROL DE PLANTAS Muestra O: R.L.: 0 0
 Cliente: 9 CEMOSA Estado muestra: EN Entrada Realza Téc. Rev.:
 Contratista: Tipo Material: 35 Ladrillo
 Observación:
 Servicio: 14003 LAD Ensayo de eflorescencia UNE 67029 E
 F.Inicio Servicio: 30.09.2015 F.Fin Servicio: Fecha Revisión: 30.09.2015 Fecha de Pedido: F.Fin Mues.:

Probeta	Intensidad	Superficie Cara (cm2)	%Sup. Afectada	Calificación

Calificación de la muestra

Recipiente Ensayo Eflorescencia:

Registrador de Tª y Humedad:

Datos Auditoria

Creado el: 30.09.2015 Creado por: ABAPCEMOSA Última modif.: 30.09.2015 Modificado por: ABAPCEMOSA

Ilustración 24 - E.R. Eflorescencia

4. 14004 – Expansión por humedad – UNE 67036.

Entrada de Resultados 0 (General)

Datos de cabecera

Expediente: 0/1500035/1 Oferta Ladrillos Muestra: 11304 R.L.: 0201 1 2012 684
 Obra: 019002 CONTROL DE PLANTAS Muestra O: R.L.: 0 0
 Cliente: 9 CEMOSA Estado muestra: EN Entrada Realza Téc. Rev.:
 Contratista: Tipo Material: 35 Ladrillo
 Observación:
 Servicio: 14004 LAD Ensayo de expansión por humedad UNE
 F.Inicio Servicio: 30.09.2015 F.Fin Servicio: Fecha Revisión: 30.09.2015 Fecha de Pedido: F.Fin Mues.:

Lecturas Comparadores

Probeta	Calibre 24h/60°		6h/60°		24h/100° agua		Calibre Lecturas Comparadores			Exp.Humedad	Exp.Potencial	
	L0-1	L0-2	L1-1	L1-2	L2-1	L2-2	L3-1	L3-2	L0			L1

Media: 0.000 0.000

Calibre:

Regla:

Datos Auditoria

Creado el: 30.09.2015 Creado por: ABAPCEMOSA Última modif.: 30.09.2015 Modificado por: ABAPCEMOSA

Ilustración 25- E.R. Exp. por Humedad

7. 14008 – Ensayo de tasa de absorción inicial (succión) – UNE-EN 772-11:2001.

Sistema Ayuda

Entrada de Resultados 0 (General)

Datos de cabecera

Expediente 0/1500035/1 Oferta Ladrillos Muestra 11304 R.L. 0201 1 2012 684
 Obra 019002 CONTROL DE PLANTAS Muestra O R.L. 0 0
 Cliente 9 CEMOSA Estado muestra EN Entrada Realiza Téc. Rev.
 Contratista Tipo Material 35 Ladrillo
 Observación
 Servicio 14008 LAD Succión UNE EN 772 11 Mar. 2001
 F.Inicio Servicio 30.09.2015 F.Fin Servicio Fecha Revisión 30.09.2015 Fecha de Pedido F.Fin Mues.

Probeta	Soga	Tizon	Grueso

Calibre
 Bandeja
 Cronóm.
 Superficie sumergida

Probeta	Área As (mm2)	Peso Seco	Peso Húmedo	Tasa A.I. (kg/m2*min)	Tasa A.I. (g/cm2*min)

Tasa A.I. (kg/m2*min) 0.00 Tasa A.I. (g/cm2*min) 0.00

Datos Auditoría

Creado el 30.09.2015 Creado por ABAPCEMOSA Última modif. 30.09.2015 Modificado por ABAPCEMOSA

Ilustración 28 - E.R. Succión

16.14161 – Planeidad de las caras – UNE-EN 772-20.

The screenshot shows a software application window titled 'Sistema Ayuda' with a menu bar and a toolbar. The main content area is titled 'Entrada de Resultados 0 (General)'. It contains a 'Datos de cabecera' section with various input fields for 'Expediente', 'Obra', 'Cliente', 'Contratista', 'Observación', 'Servicio' (14161), 'LAD Planeidad de las caras UNE EN 772 20', and dates for 'F.Inicio Servicio', 'F.Fin Servicio', 'Fecha Revisión', 'Fecha de Pedido', and 'F.Fin Mues.'. Below this is a section for 'Caras especificadas como planas' with radio buttons for 'Tabla (Cara mayor)', 'Canto (Cara media)', and 'Testa (Cara menor)'. A table with 10 columns (Probeta, Cara, Medida, Diagonal, Flecha, Diagonal, Flecha, Diagonal, Flecha) and multiple rows is provided for data entry. Below the table are summary fields for 'Planeidad de las caras especificadas como planas' (Diagonal Media, Flecha Media Máxima) and 'Planeidad Tablas' (Flecha Media Máxima, Flecha Máxima). There are also fields for 'Tabla', 'Canto', 'Testa', 'Calibre', and 'Galgas'. At the bottom, a 'Datos Auditoria' section includes fields for 'Creado el', 'Creado por', 'Última modif.', and 'Modificado por'.

Ilustración 37 - Planeidad de las caras

3.3.3 Generación de Informes

El diseño de la generación de informes será prácticamente igual que el de la plantilla proporcionada por los técnicos de laboratorio, eliminando aquellos ensayos que no se realizan y representando los datos de dos formas: mediante tablas y campos.

Las tablas se crearán con las columnas que hagan falta por cada ensayo y dos filas: una para la cabecera, con un formato distinto, y otra a partir de la cual se insertarán los datos de los ensayos. Si aplicamos formato a las celdas de la segunda fila, se repetirá durante todas las filas que se añadan para los registros que se inserten en la tabla.

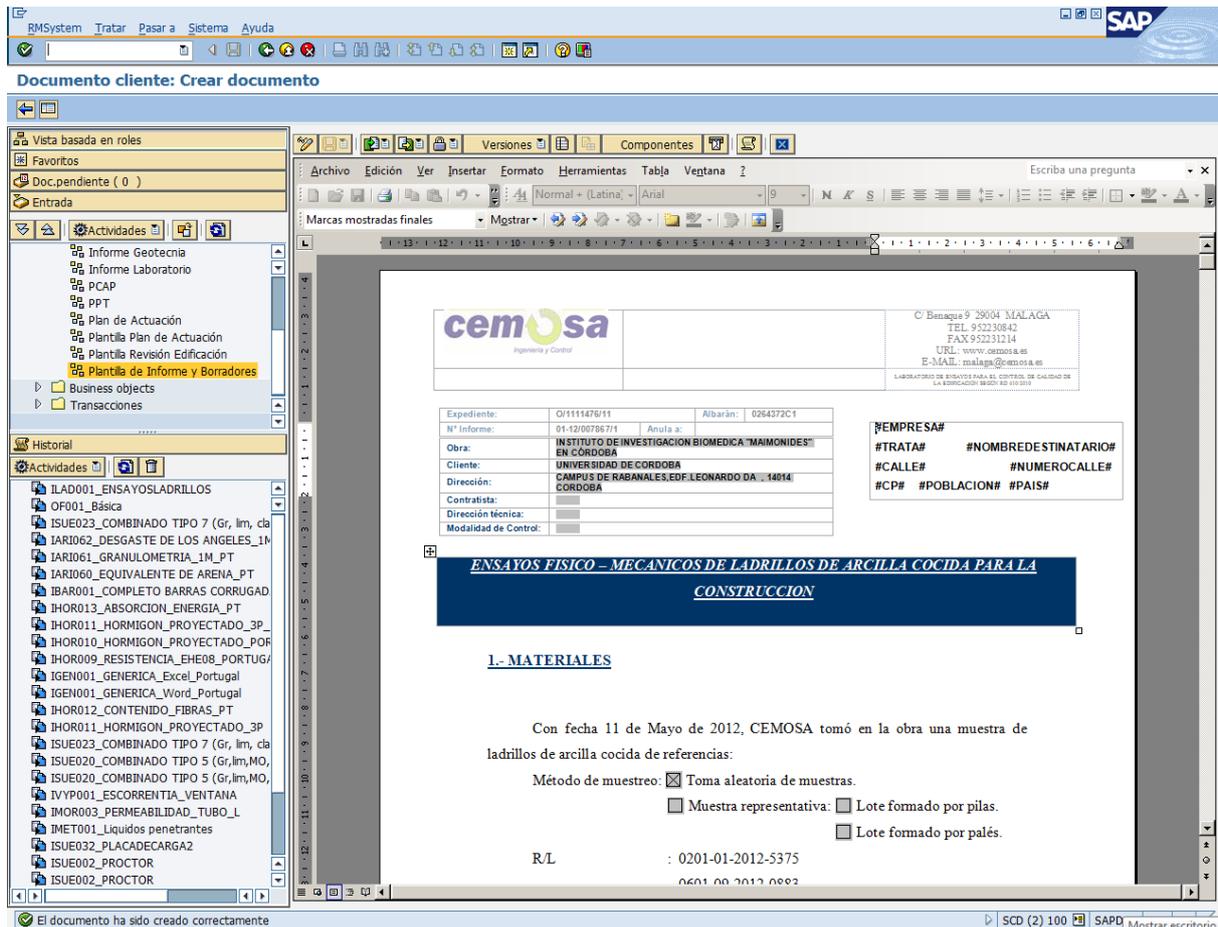


Ilustración 38 - Diseño del Informe de Ladrillos

3.4 Modelo de Datos

Por último, para finalizar esta fase del proyecto, y a partir de la información obtenida mediante las tareas anteriores, se diseñará el modelo de datos que dará soporte al nuevo módulo del Sistema.

El modelo de datos proporciona una visión global de las entidades que forman el sistema y las relaciones que existen entre ellas. En este apartado se presentarán dos modelos diferentes que ayudarán en la tarea de decidir cómo se almacenarán los datos y cómo se realizará el acceso a los mismos, facilitando la tarea del desarrollo posterior.

- Modelo del Dominio, un modelo conceptual simple y de alta abstracción que indica cómo funciona el Sistema físicamente.
- Modelo de Datos, un modelo más complejo y grande en el que se indicarán las entidades que participarán en la base de datos; es decir, las tablas que se usarán y las relaciones entre ellas.

El modelo del dominio que se ha obtenido del sistema es el siguiente:

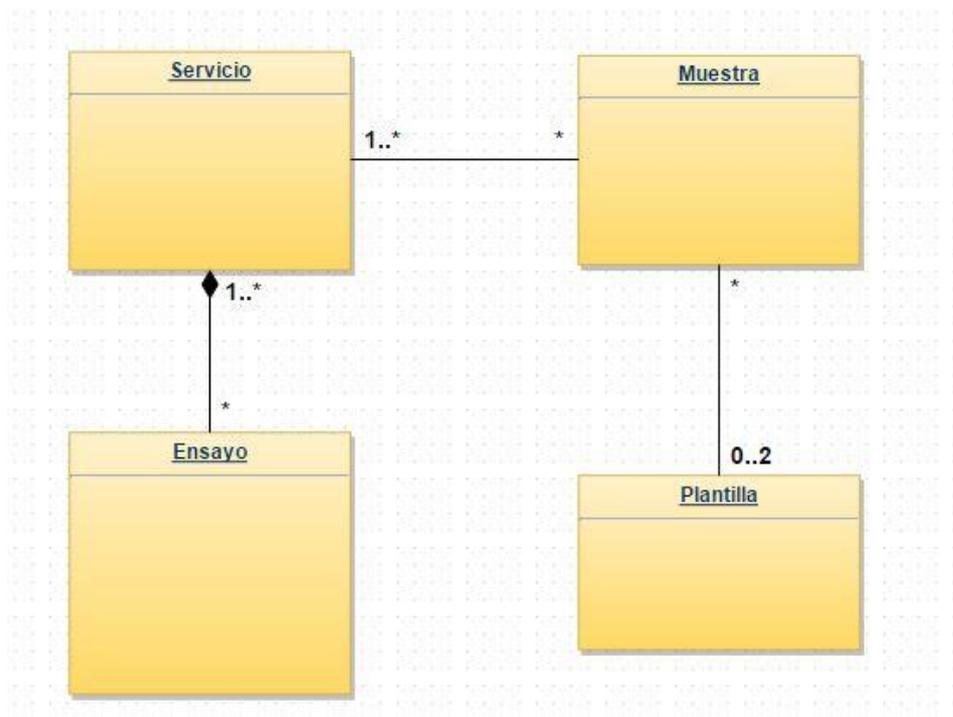


Ilustración 39 - Modelo del Dominio

Como se puede apreciar, este modelo es lo suficientemente básico para darnos una idea de cómo funciona el Sistema: A cada muestra se le asigna una serie de Servicios y una o dos plantillas (según se requiera que tenga informe, borrador, ambas o ninguna). Cada servicio está compuesto por varios ensayos, y pueden ser asignados a multitud de muestras. Cada ensayo puede estar darse en multitud de servicios y cada plantilla puede aparecer en multitud de muestras.

Para simplificar la comprensión del modelo de datos que se ha diseñado, se ha dividido según el módulo al que pertenecen.

1. Entrada de Muestras

El diseño del modelo de datos será simple, ya que la mayor parte del modelo existe en el Sistema. La tabla `ztlb_muestras` es la encargada de almacenar toda la información sobre las muestras creadas, por lo que bastará con añadir un campo para la designación del material.

Como pueden existir varios tipos de designaciones y, de cara al futuro, ampliar este dato a otro tipos de materiales, se decidió crear la tabla `ztlb_desmat` donde, a partir del código del material y código que identifique a la designación, obtener la descripción de esta.

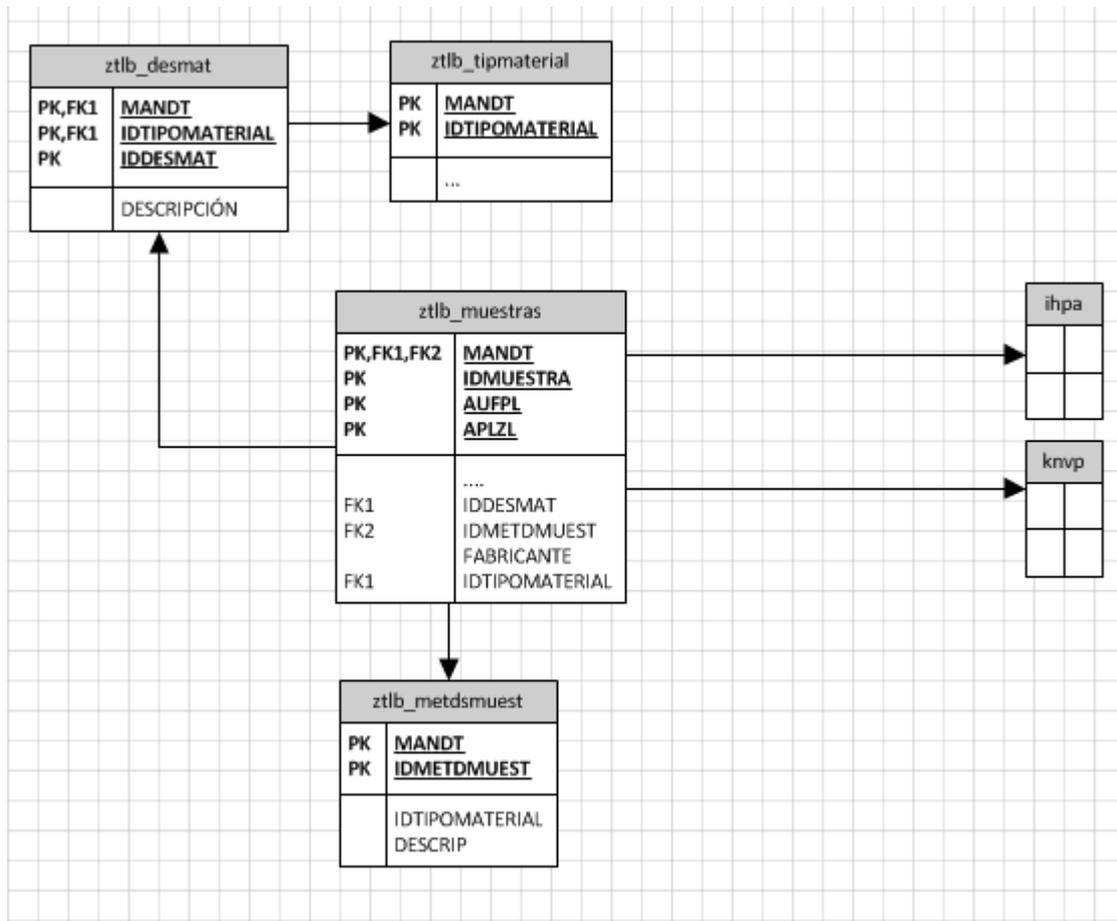


Ilustración 40 - Modelo de Datos: Entrada de Muestras

Observar que el conjunto de campos que forman las claves primarias de una tabla usan un campo denominado 'MANDT'; este campo MANDT es una característica de SAP que hace referencia al cliente que se encuentra usando el sistema en ese momento, provocando que los registros que se almacenen o se quieran observar dependan de la persona que está accediendo.

En cuanto a los campos de que identifican el método de muestreo y el fabricante, servirán en este diseño, modificando una serie de cosas durante la fase de desarrollo.

2. Entrada de Resultados

Ya que existen muchas entradas de resultados, y a cada una se le ha designado una o más tablas, se va a mostrar por separado el modelo de datos de cada una de ellas. Se mostrará el modelo de cada uno de los servicios en el orden en el que se han realizado.

Existe una consideración a tener en cuenta: todas las tablas donde se almacenen datos relativos a los ensayos deben hacer referencia a una tabla existente en el Sistema, denominada ztlb_murelser. Se trata de una tabla

maestra (principal) que relaciona las muestras con los servicios, de ahí los tres campos que forman su clave primaria (junto al ya conocido MANDT):

- IDMUESTRA: identificado único de una muestra.
- AUFPL: Número de hoja de ruta de operaciones
- APLZL: Contador general de la orden

IDMUESTRA identifica a la muestra sobre la que se realiza el ensayo, mientras que la combinación de AUFPL y APLZL identifica al servicio.

2.1. Aspecto y Estructura. Defectos.

En este tipo de ensayos bastará con una tabla en la que guardar cada uno de los datos que califican al ensayo. Además se incluyen los campos pertenecientes a la estructura zscA_auditoria, que serán los encargados de guardar la auditoria de la creación y/o modificación de cada registro.

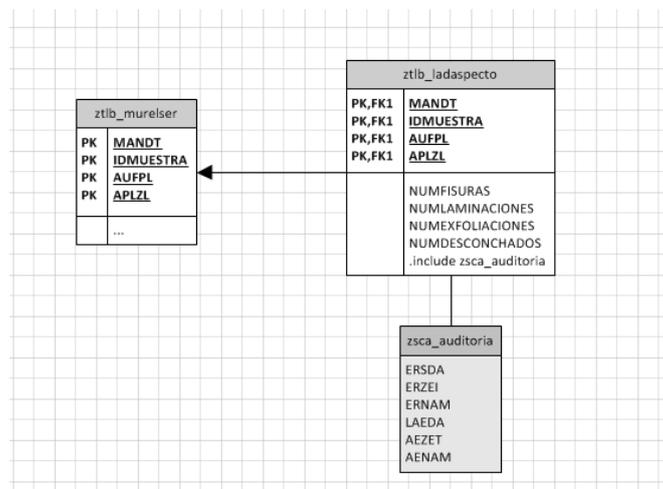


Ilustración 41 - Modelo de datos: E.R-Defectos

Componente	TpR	Tp.componente	Tipo de dat.	Long.	Decim.	Descripción breve
ERSDA	<input type="checkbox"/>	ERSDA	DATS	8		0 Fecha de creación
ERZEI	<input type="checkbox"/>	ANUZI	TIMS	6		0 Hora en la que fue creado el registro
ERNAM	<input type="checkbox"/>	ERNAM	CHAR	12		0 Nombre del responsable que ha añadido el objeto
LAEDA	<input type="checkbox"/>	LAEDA	DATS	8		0 Fecha última modificación
AEZET	<input type="checkbox"/>	AEZET	TIMS	6		0 Hora de última modificación
AENAM	<input type="checkbox"/>	AENAM	CHAR	12		0 Nombre del responsable que ha modificado el objeto

Ilustración 42 - Campos auditoría

2.2. Heladicidad

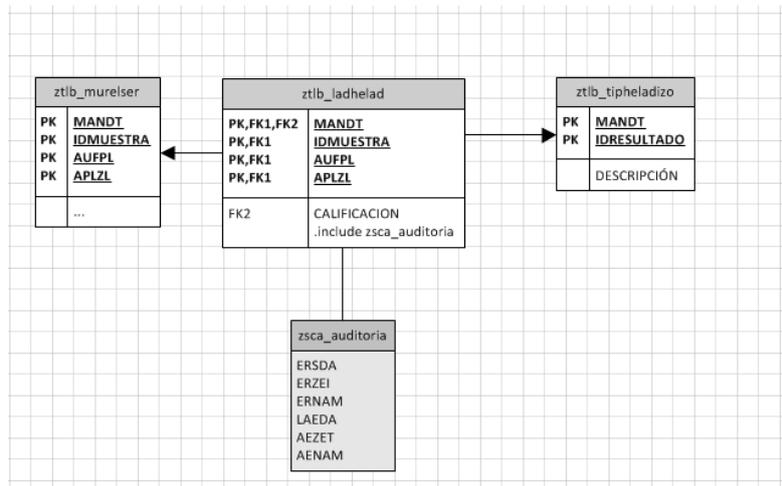


Ilustración 43 - Modelo de Datos: ER-Heladicidad

2.3. Inclusiones Calcáreas

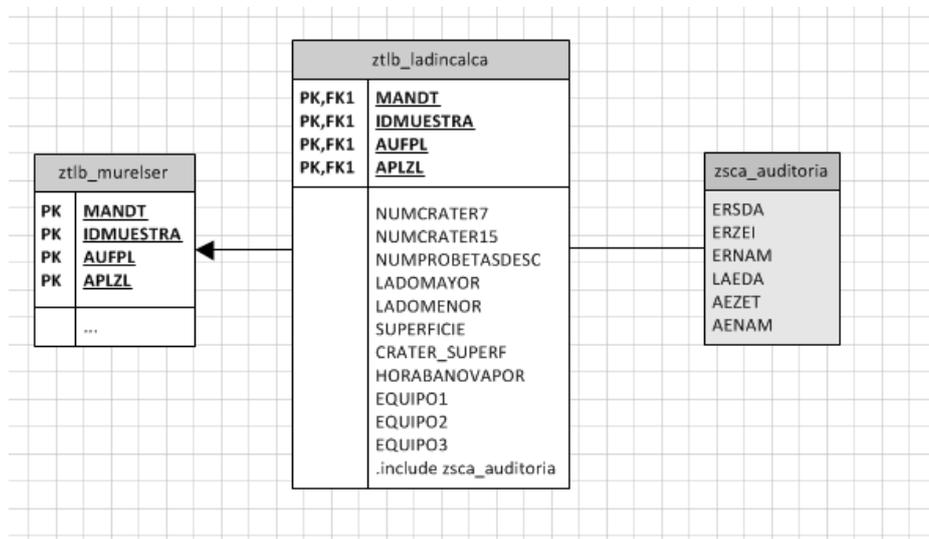


Ilustración 44 - Modelo de Datos: ER-Inclusiones calcáreas

2.4. Dimensión y forma

En este tipo de servicios, se crearán dos tablas: una para los resultados, similar a la de los ensayos anteriores; y otra para las mediciones, a la que se le añade un campo más como combinación de clave primaria: el campo probeta (ya que en un mismo ensayo, las mediciones se realizarán sobre varias probetas). Al igual que en los servicios anteriores, ambas tablas deben tener los campos de auditoría.

Además, como se debe diferenciar el ladrillo según la forma que tiene, el campo forma será una clave foránea a otra tabla (ztlb_Idtipoforma), donde se almacenarán los diferentes valores de forma que puedan existir (en este caso, Ladrillo Visto o Ladrillo No Visto).

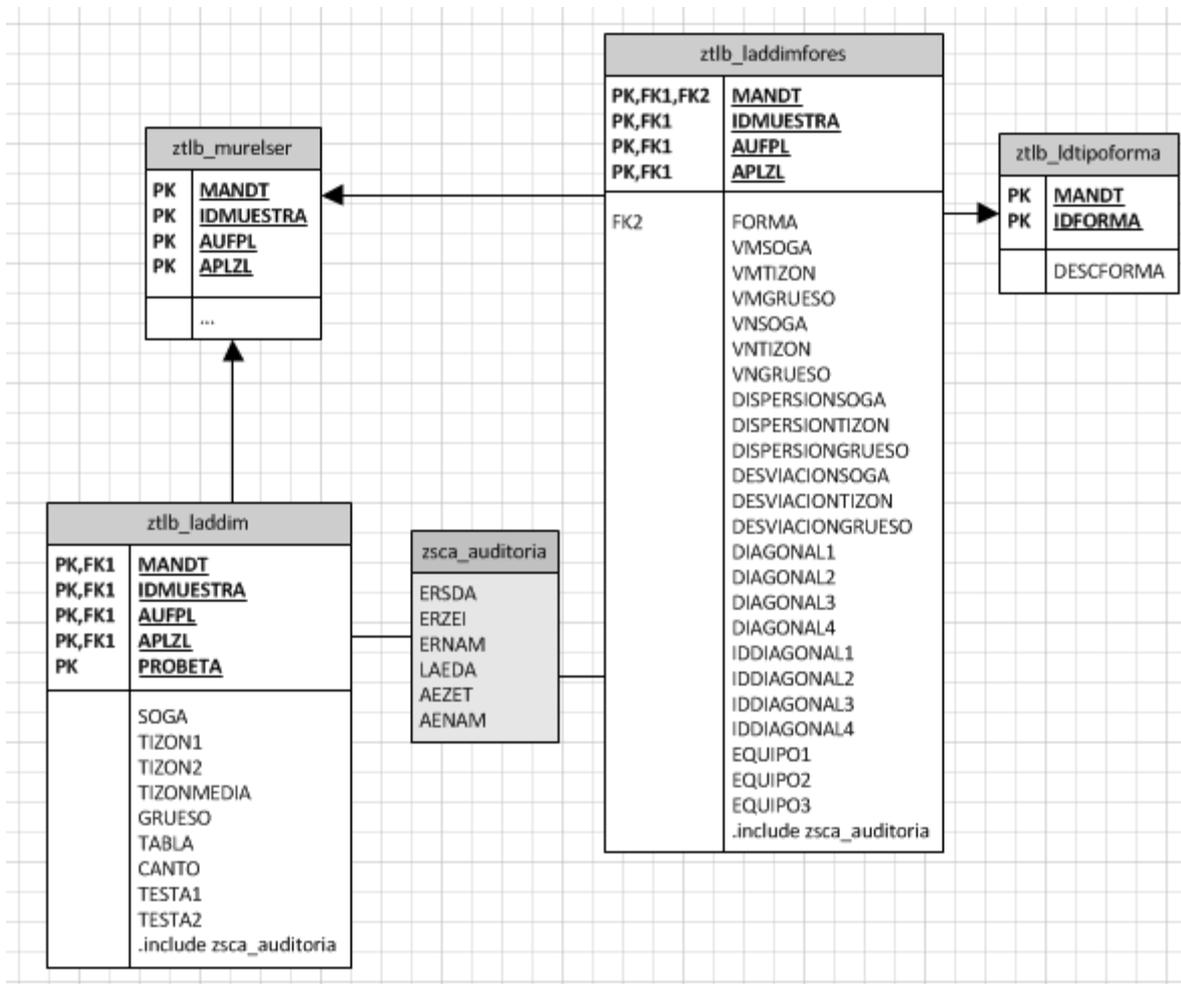


Ilustración 45 - Modelo de Datos: ER-Dimensión y forma

2.5. Longitud, Anchura y altura

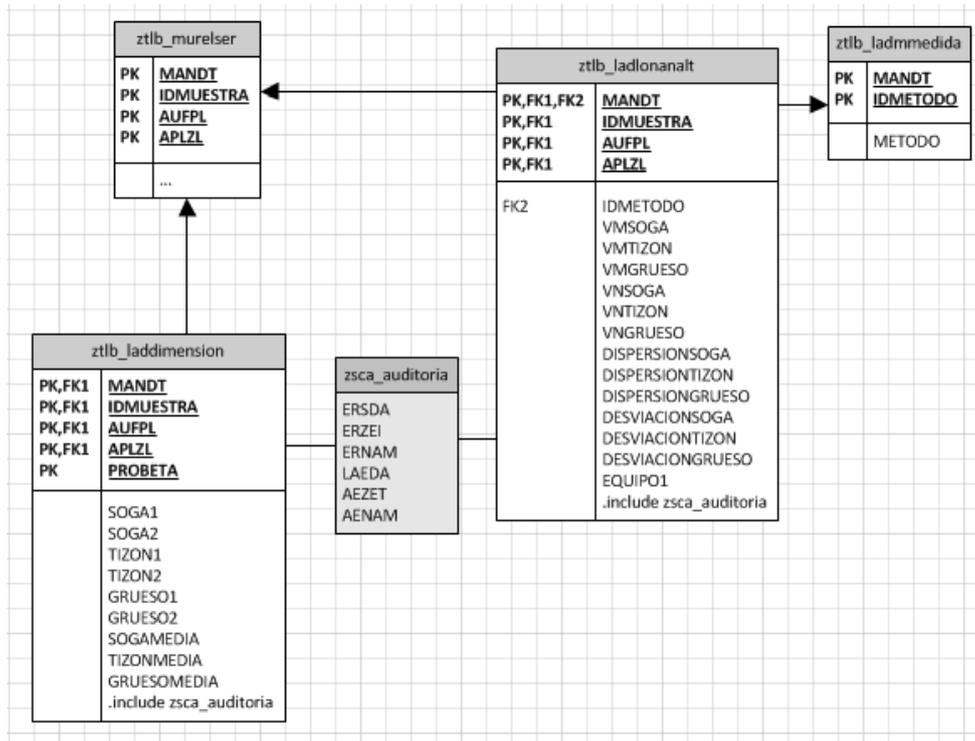


Ilustración 46 - Modelo de Datos: ER-Long., Altura y Anchura

2.6. Planeidad

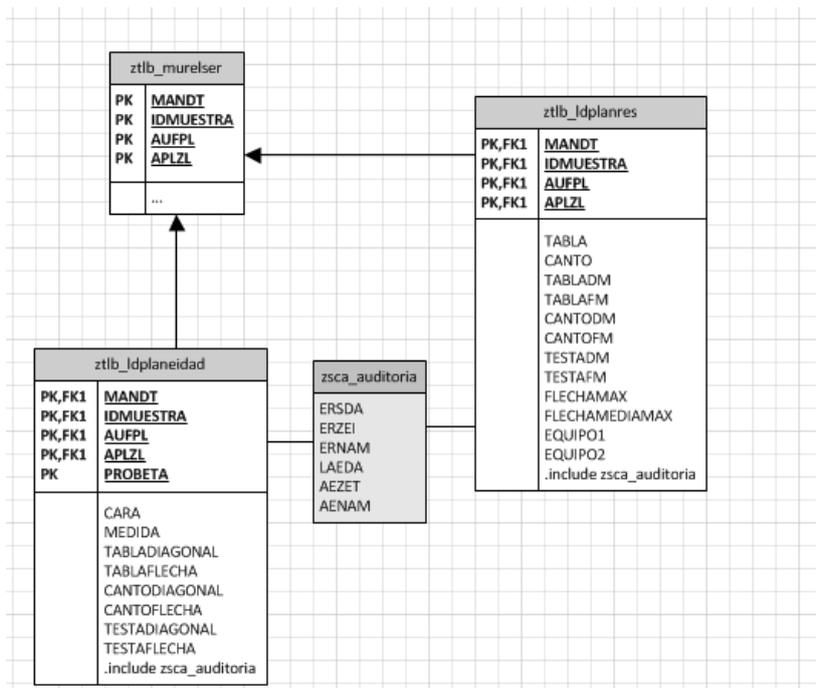


Ilustración 47 - Modelo de Datos: ER-Planeidad

En este modelo se ha realizado un pequeño cambio. El campo cara puede tener 2 valores, A o B, según la cara que se esté midiendo. En los ensayos anteriores, se ha resuelto mediante un identificador que referencia a otra tabla, donde se almacenan las diferentes caras; en este caso se ha optado por escoger la solución que ofrece SAP en los dominios de tipo: Al crear el dominio de datos, se especifica cuáles son los valores posibles que puede tomar este campo:

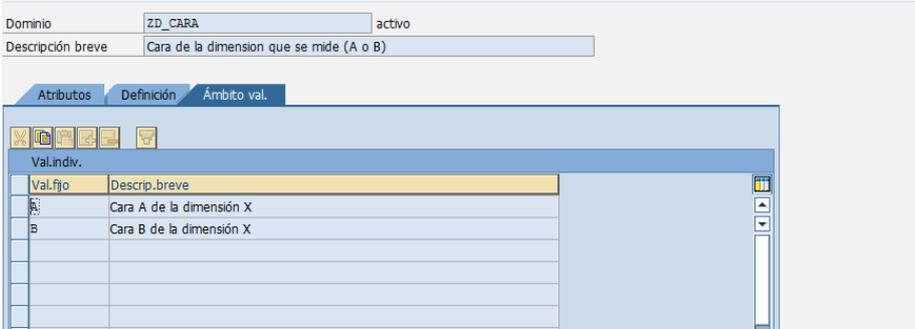


Ilustración 48 - Modelo de Datos: Ámbito de valores de un Dominio

2.7. Expansión por humedad

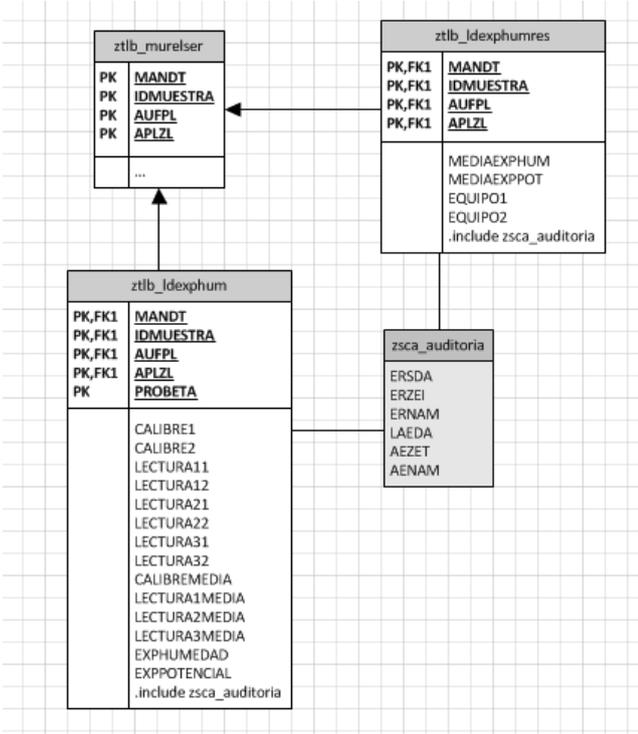


Ilustración 49 - Modelo de Datos: ER-Exp por Humedad

2.8. Masa Anexo D

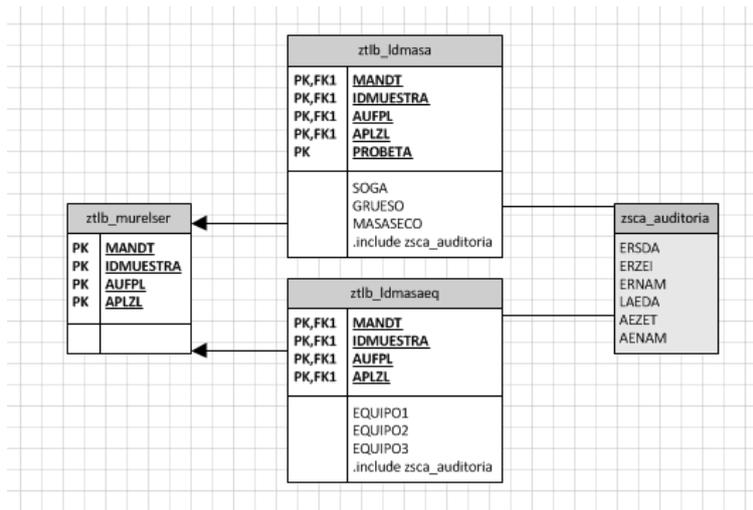


Ilustración 50 - Modelo de Datos: ER-Masa

2.9. Eflorescencia

Este modelo de datos es similar a los anteriores: una tabla para mediciones por probeta y otra para resultados finales. Además se incluyen tablas para listar los distintos valores de intensidad, superficie afectada y calificación (esta última se obtiene a partir de los identificadores de intensidad y superficie). El campo calificación de la tabla ztlb_idefloresc será el valor que más se repita de las calificaciones obtenidas.

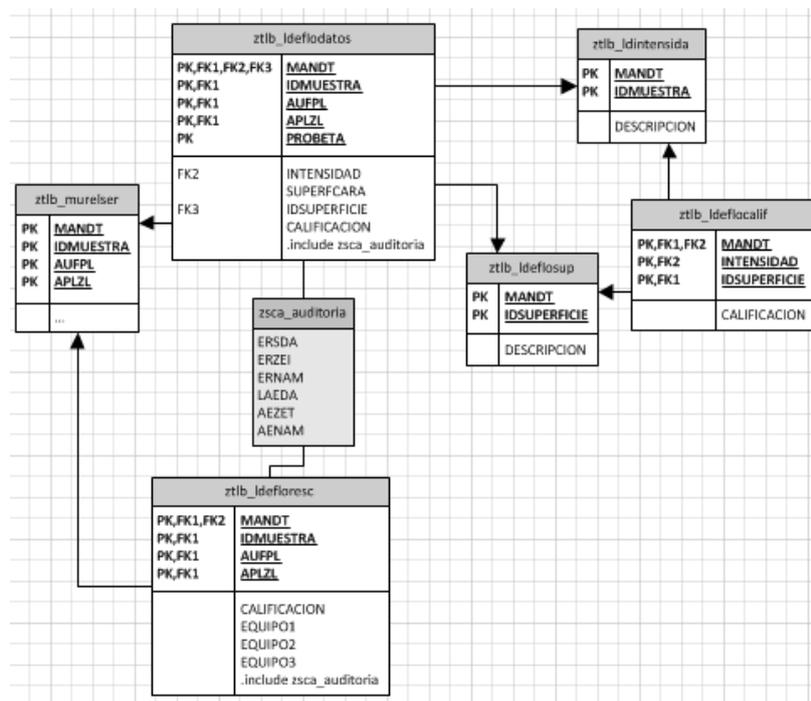


Ilustración 51 - Modelo de Datos: ER-Eflorescencia

2.10. Flexión

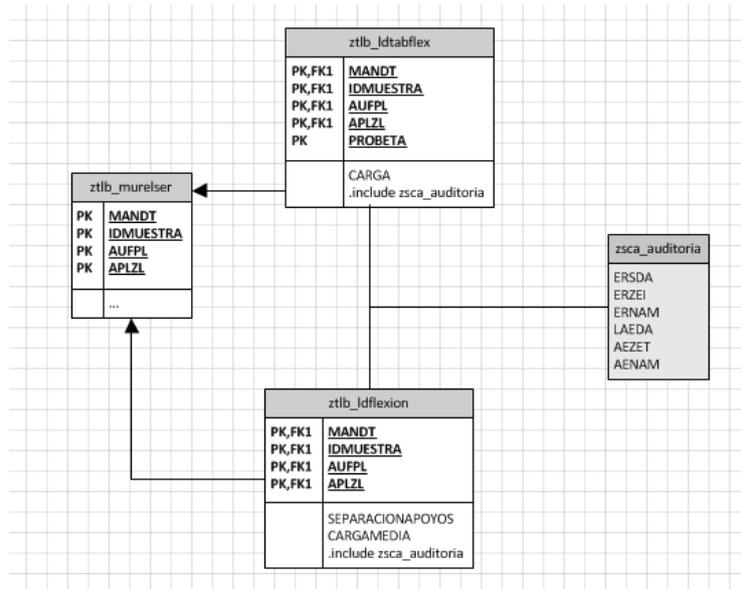


Ilustración 52 - Modelo de Datos: ER-Flexión

2.11. Porcentaje Huecos

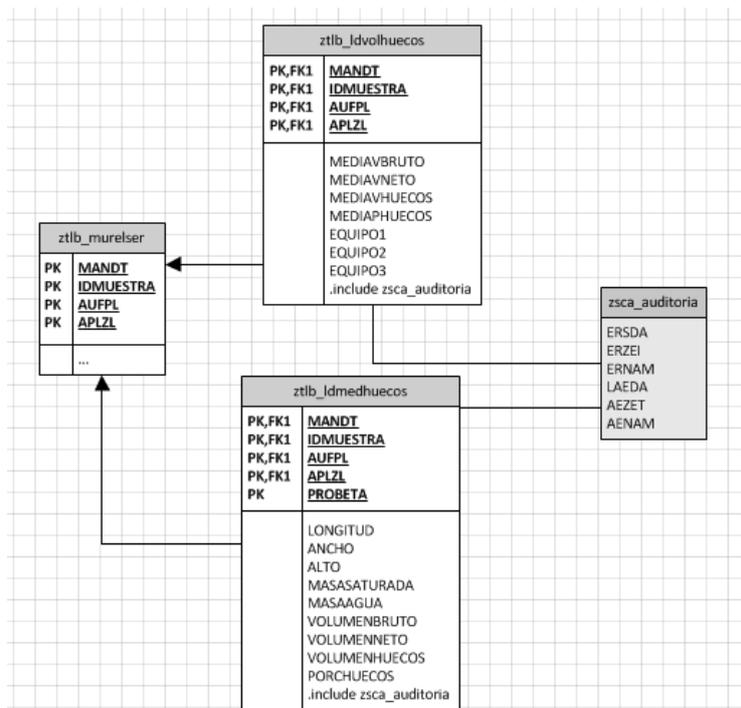


Ilustración 53 - Modelo de Datos: ER-Huecos

2.12. Absorción Agua

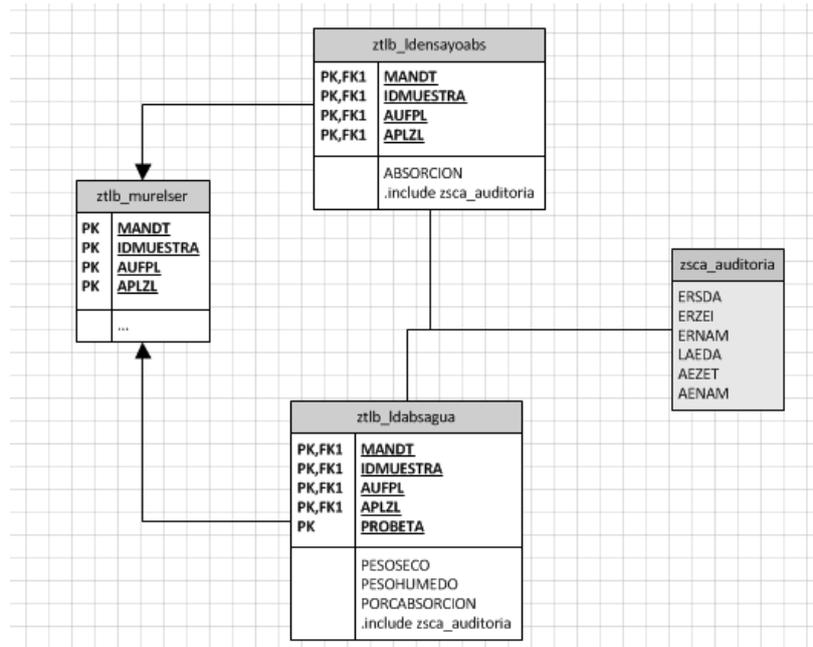


Ilustración 54 - Modelo de Datos: ER-Absorción agua

2.13. Densidad Absoluta

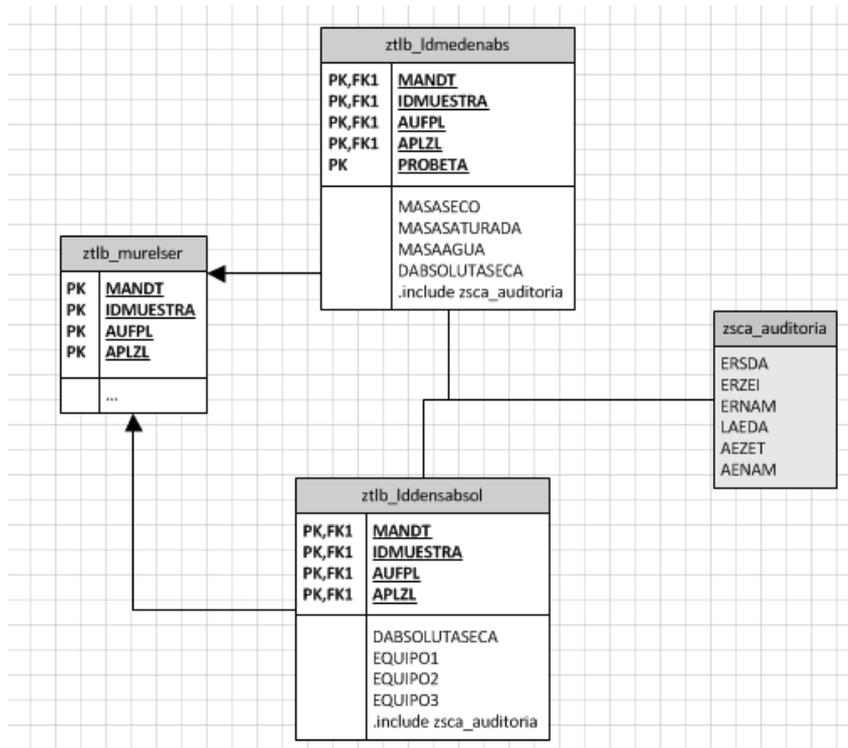


Ilustración 55 - Modelo de Datos: ER-Densidad absoluta

2.14. Densidad Aparente

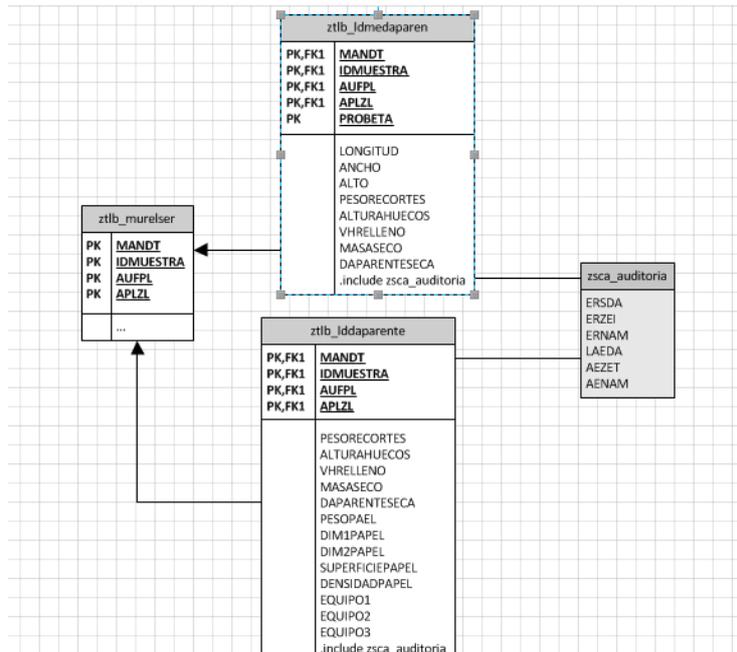


Ilustración 56 - Modelo de Datos: ER-Densidad Aparente

2.15. Compresión

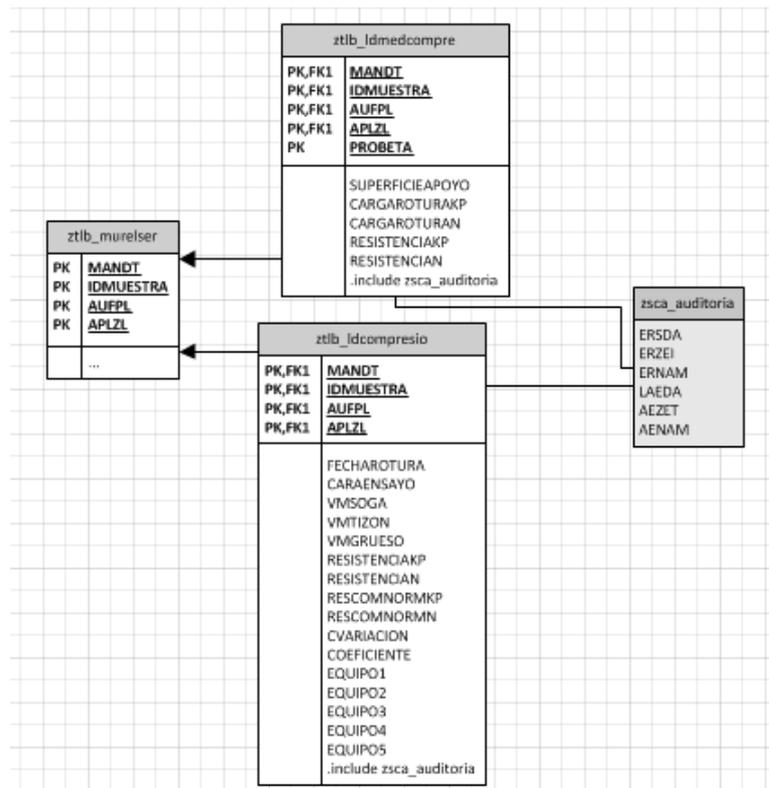


Ilustración 57 - Modelo de Datos: ER-Compresión

2.16. Absorción inicial (succión)

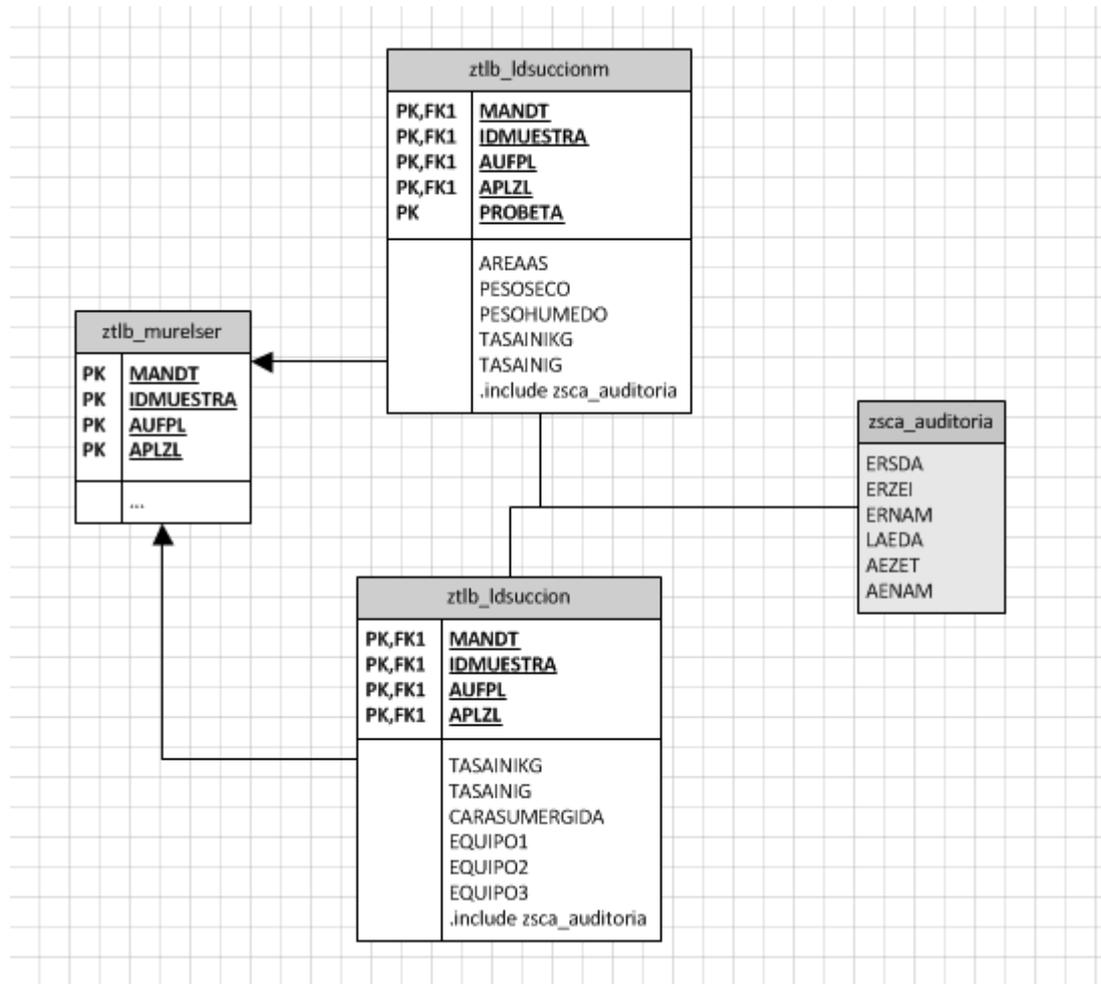


Ilustración 58 - Modelo de Datos: ER-Succión

3. Generación de Informes

Para la generación de informes no será necesario la creación de ninguna tabla en la base de datos ya que tan sólo serán necesarias una serie de consultas sobre las tablas existentes, tanto de muestras como de ensayos.

4. Implementación

En este capítulo se abordará la fase del desarrollo del software mediante las herramientas que proporciona SAP para este fin.

Con la finalidad de simplificar todo el proceso, de forma que se comprenda de una manera fácil la labor realizada, se ha vuelto a dividir el capítulo en tres secciones, en la que se indicarán los puntos clave de este desarrollo y se mostrará trozos de código para ilustrar estas ideas.

En general, se podría decir que el desarrollo de la Entrada de Muestras y la Entrada de Resultados ha consistido en la creación de un programa, para cada una de las dos entradas, del tipo ModulPool.

Los ModulPool son programas que únicamente se pueden ejecutar a través de una transacción y que nos permiten mantener los DynPros de la misma (recordar que el término DynPro hace referencia al recurso tecnológico que ofrece SAP para manejar la capa de presentación de las aplicaciones o, dicho de otra forma, la interfaz sobre la que trabajará el usuario final). Los Dynpros proveen un conjunto de herramientas para modelar y diseñar las interfaces de usuario basándose en el paradigma MVC (Modelo Vista-Controlador).

Los DynPros están compuestos, en primer lugar, de una pantalla y una lógica de procesos de esa pantalla, donde se definen los atributos de la misma y los atributos de cada campo. En la Lógica de proceso se definen los módulos que se ejecutarán antes de que el sistema presente la pantalla (PBO, Process Before Output) y los procesos que se ejecutarán después de una interacción del usuario con la pantalla (PAI, Process After Input). Por lo tanto, los objetos parciales de un DynPro serían los siguientes:

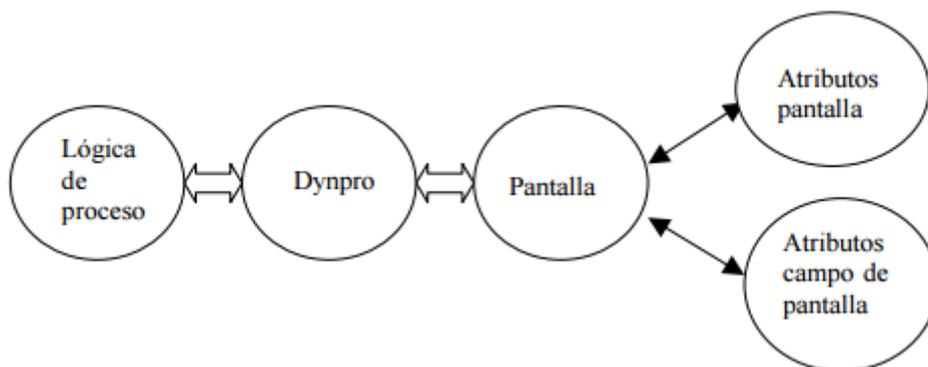


Ilustración 59 - Objetos DynPro

Todos los programas ModulPool desarrollados cuentan con los siguientes elementos:

- DynPros (como mínimo uno)
- Top: Espacio donde se definirán las variables globales, tablas y demás elementos que se usarán durante la programación del DynPro.

- PBO: existirá un módulo PBO por cada DynPro realizado. En él se llamarán a las subrutinas que cargarán los datos a mostrar o definirán el comportamiento de la ventana antes de mostrarse o al recargarse.
- PAI: al igual que ocurre con los PBO, existirá un módulo PAI por cada DynPro diseñado. En él se definirá el comportamiento del Sistema tras realizar una acción; algunos ejemplos serían la introducción de datos en la pantalla, pulsar el botón de guardar, etc.
- F01: módulo donde se diseñarán las subrutinas o se llamarán a los Módulos de Funciones.

Por lo tanto, el programa principal deberá contener una sentencia 'include' por cada módulo que se realice.

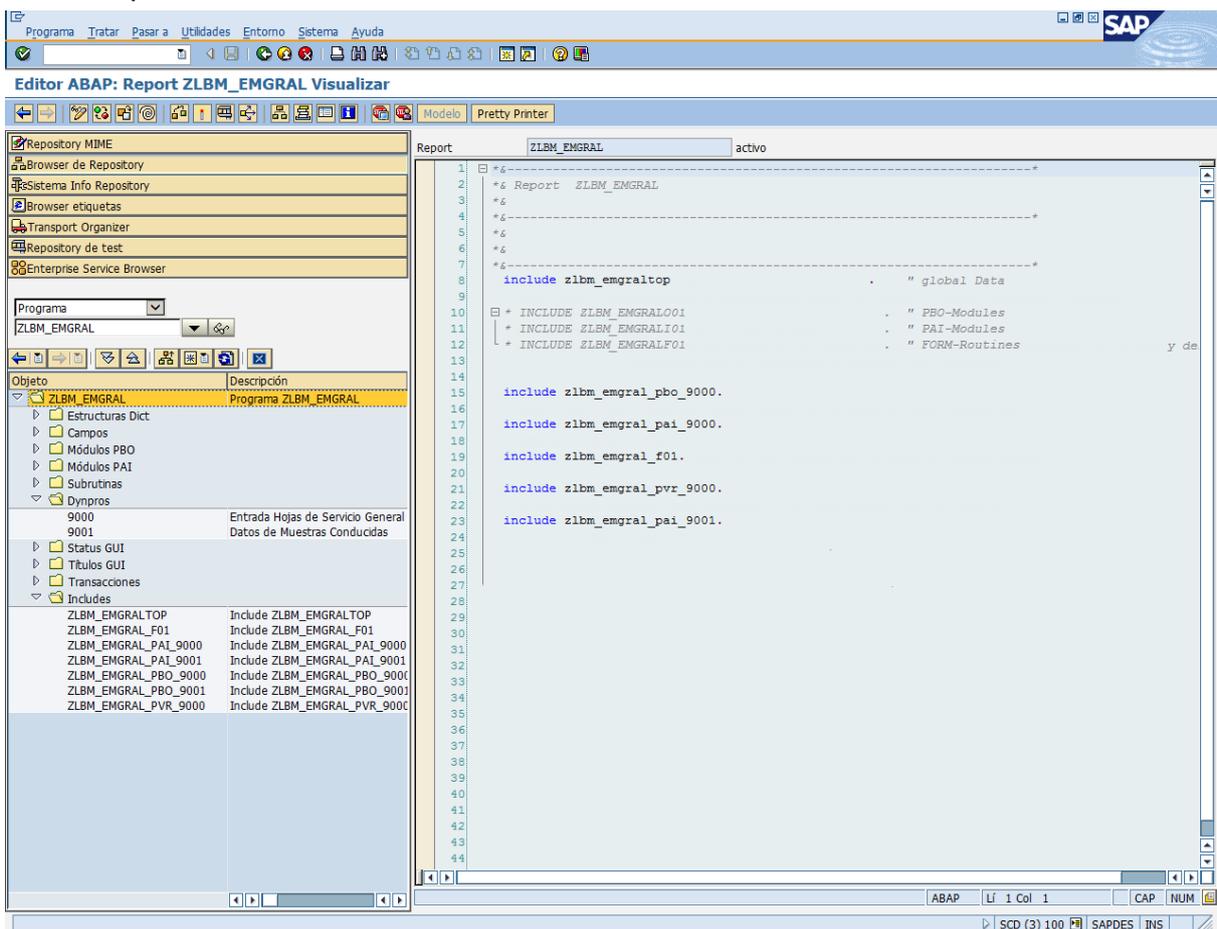


Ilustración 60 - Programa Principal - Sentencias Include

La creación de las distintas interfaces se realizará a través de la herramienta 'Screen Painter'. Esta herramienta permite la creación de las interfaces y sus componentes de una fácil e intuitiva, ya sean campos de texto, botones, tablas o cualquier otro elemento de los disponibles.

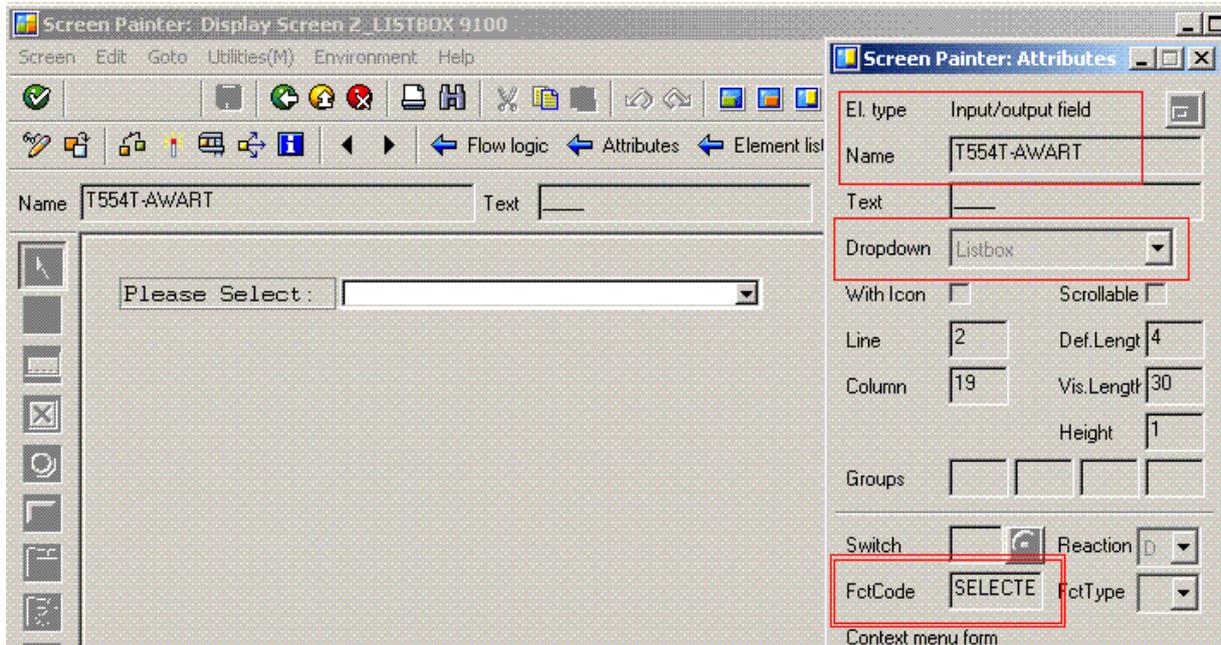


Ilustración 61 - Screen Painter

La generación de informes se realizará de una forma distinta, ya que SAP provee de una herramienta con la cual cargar un documento que funcionará como plantilla y, de forma gráfica y programación de funciones, realizar las tareas que se requieran para mostrar los datos.

4.1 Diccionario de Datos

El diccionario de datos de ABAP tendrá una labor importante durante la tarea de desarrollar el software, ya que la utilización de Estructuras, Vistas, Ayudas de Búsqueda y demás elementos que proporciona el diccionario facilitarán la tarea de manejar los datos extraídos de Base de Datos y mostrarlos en el DynPro.

En este caso se utilizaron los siguientes elementos:

- Dominios y Elementos de Datos: utilizados para la creación de los campos de las tablas.
- Estructuras: tanto Estructuras de una tabla (funcionan como un registro donde se almacenan los campos de una o más tablas, junto con otros campos que se quieran añadir) y Tipos Tablas, que funcionan como una tabla en la que cada registro es una estructura.
- Vistas: funcionan como el operador JOIN de una consulta a Base de Datos, obteniendo los datos de varias tablas cuando cumplen una condición.

- Ayudas de Búsqueda: útiles para cuando se quieren mostrar por pantallas las distintas opciones que el usuario pueda seleccionar para rellenar un campo con un valor; realiza una consulta a una tabla o vista con una serie de requisitos, devolviendo los campos que hayamos definido para los registros que cumplan dicha condición; el usuario no tendrá más que pulsar sobre el registro que le interese para que este se cargue en la ventana.

4.2 Entrada de Muestras

Tal y como se ha especificado anteriormente, se requiere que se modifique la Entrada de Muestras General, ya existente en el Sistema, para que dé soporte al nuevo material (Ladrillo). Por lo tanto, bastará con realizar unas pequeñas modificaciones al programa ZLB_EMGRAL, adaptándolo a los nuevos campos o valores de Ladrillo.

Se deberá añadir este nuevo campo a la tabla de muestras, se creará un campo en el dynpro que funcione como referencia a él mismo en la tabla. También se creará una ayuda de búsqueda en el diccionario de datos para facilitar la tarea de selección a los técnicos.

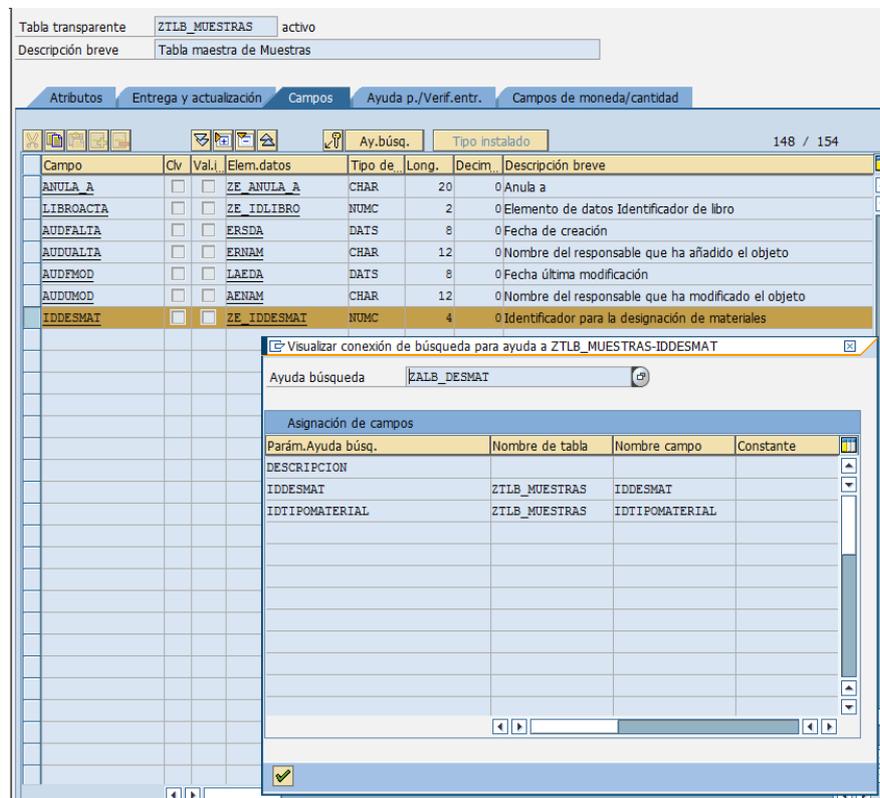


Ilustración 62 - A. Búsqueda Designación Material

A la hora de crear la Ayuda de Búsqueda, se indica la tabla a la que se le realiza la consulta y se indican los campos que se importan (muestran al usuario) y exportan (valores que se muestran en la ventana tras su selección).

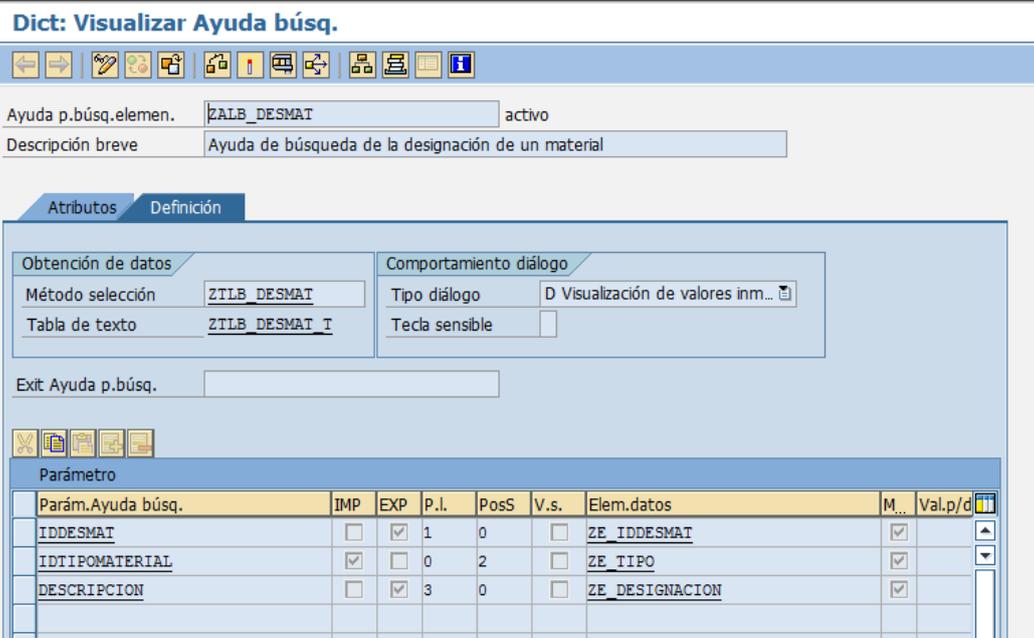


Ilustración 63 - Ay. Búsqueda Designación Material

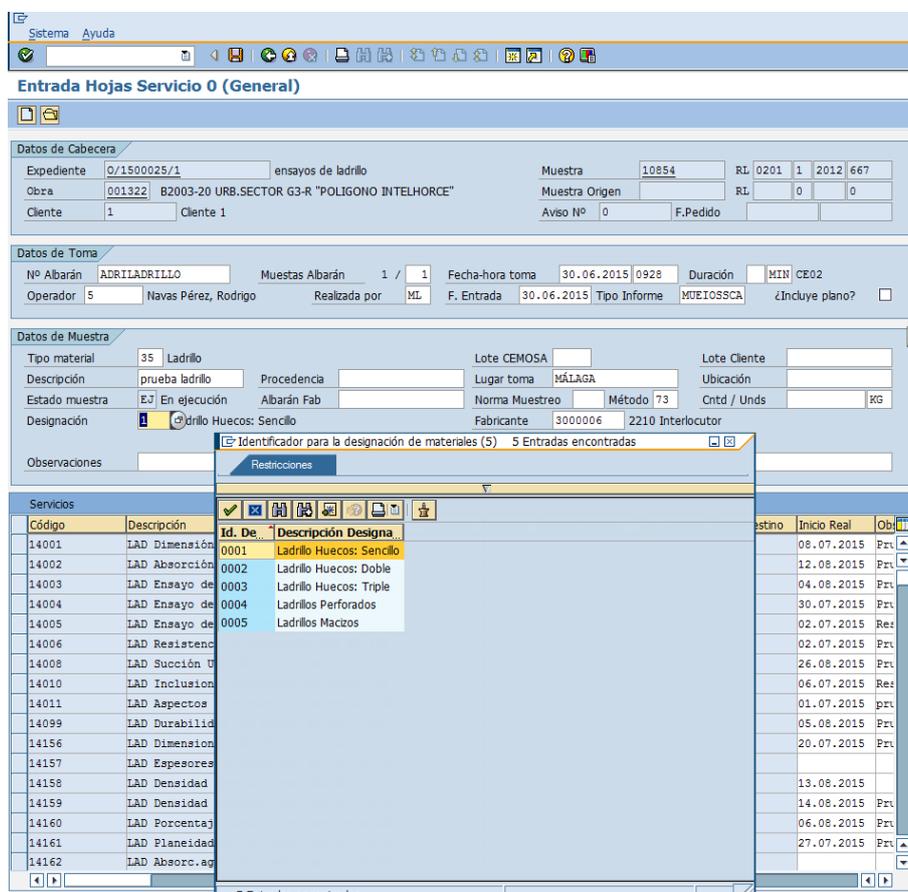


Ilustración 64 - Ejemplo Ayuda de Búsqueda

Una vez realizado este proceso, se creará la Lógica que dará soporte a esta modificación. Tal y como se ha indicado anteriormente, se separa la funcionalidad de mostrar por pantalla (PBO) de las acciones a realizar tras una interacción con la

ventana (PAI). Para el manejo de los datos introducidos por pantalla, se utilizará una variable tabla, del mismo tipo que la tabla de muestras, donde se almacenarán temporalmente los valores introducidos desde la pantalla.

El PBO se encuentra dividido en diferentes módulos, donde se indican las acciones a realizar. Existen módulos para indicar los valores predefinidos de la pantalla, otros para indicar el estado⁸ (se ha dejado el que tiene SAP por defecto), inicializar los datos, etc.

A continuación, nos vamos a centrar en este último módulo, que será donde vaya parte de la nueva modificación. En este caso, tendremos que mostrar por pantalla la designación del material en caso de que se haya seleccionado y el fabricante, por lo que se creará una subrutina dentro del PBO para cada una de las acciones, donde se realiza la consulta a la base de datos o se actualiza el campo de la pantalla con el nuevo valor introducido, almacenado en la variable tabla descrita anteriormente.

Posteriormente, se indica la forma por la cual, a partir del código de designación, se muestra la descripción de la misma, utilizando el código de material y el código de identificación de la designación:

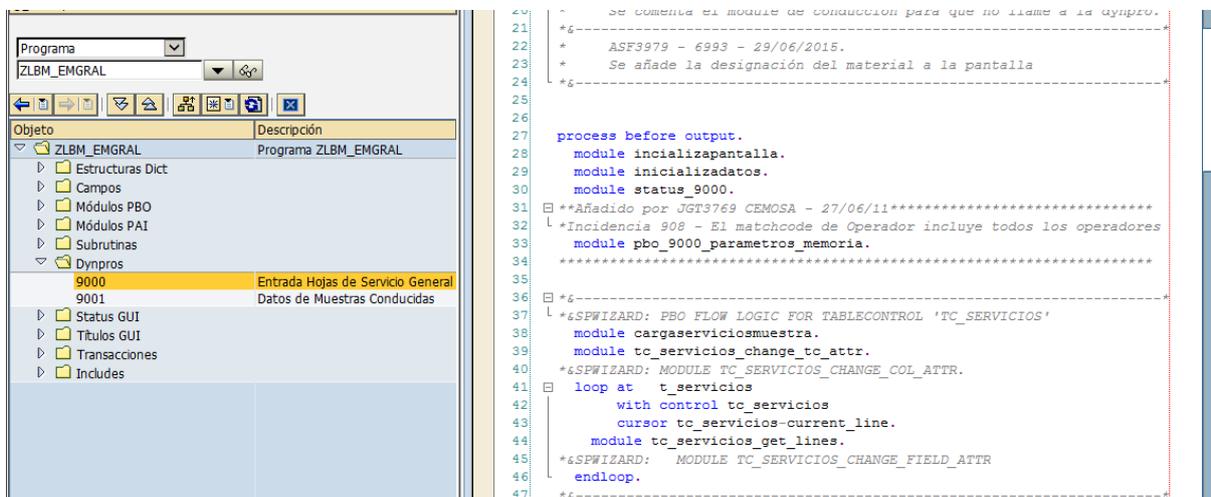


Ilustración 65 - DynPro-PBO

⁸ Los iconos de estado son elementos de la pantalla que se pueden usar para representar el estado del programa gráficamente.

```

Include ZLEM_EMGRAL_PBO_9000 activo
212 WHERE planti EQ ztlb_muestras-idplaninforme.
213 ENDIF.
214
215 *---> 6993
216 "Inicializa descripción de la designación
217 if ztlb_muestras-iddesmat is not initial.
218 perform f_cargar_descripcion_desig.
219 endif.
220
221 "Inicializa fabricante
222 if ztlb_muestras-fabricante is not initial.
223 perform f_carga_fabricante.
224 endif.
225
226
227 "inicializar función interlocutor
228 if wg_parvw is initial.
229 perform f_carga_funcion_interlocutor.
230 endif.
231
232 *<--6993
...

Include ZLEM_EMGRAL_F01 activo
3487 *% Form F_CARGAR_DESCRIPCION_DESIG
3488 *% -----*
3489 * ASF3979 - 6993 - 30/06/2015
3490 * Carga la descripción de la designación (si existe) de esta muestra
3491 * -----*
3492 * ASF3979 - 6993 - 30/06/2015
3493 * Se modifica la carga para que se haga desde una Vista, incluyendo así
3494 * la tabla de texto (seleccionar valor según idioma de SY-LANGU)
3495 * -----*
3496 * --> p1 text
3497 * <-- p2 text
3498 * -----*
3499 FORM F_CARGAR_DESCRIPCION_DESIG .
3500 data: w1_descripcion type ZE_DESIGNACION.
3501
3502 if t_muestras-iddesmat <> ztlb_muestras-iddesmat. "Si se ha producido cambio,
3503 "se entra en la ejecución.
3504
3505 select single descripcion from ZVLS_DESMAT
3506 into w1_descripcion
3507 where idtipomaterial = ztlb_muestras-idtipomaterial
3508 and iddesmat = ztlb_muestras-iddesmat
3509 and spras = SY-LANGU.
3510
3511 if sy-subrc <> 0.
3512 clear w1_descripcion.
3513 clear ztlb_muestras-iddesmat.
3514 endif.
3515
3516 * Al guardar, el programa lee la tabla interna t_muestras (del tipo ztlb_muestras)
3517 * y guarda los datos que están en ella, por lo que para aprovechar esta funcionalidad
3518 * tan sólo hay que incluir los 2 nuevos valores de designación en la tabla.
3519
3520 t_muestras-descripcion = w1_descripcion.
3521 t_muestras-iddesmat = ztlb_muestras-iddesmat.
3522 endif.
3523 ENDFORM. " F_CARGAR_DESCRIPCION_DESIG

```

Ilustración 66 - Cargar designación

Todos los procesos de carga de datos en el PBO se realizan de la misma forma: consulta a la tabla de la base de datos que corresponda a través de los parámetros necesarios para la obtención de un registro o campo.

Ahora supongamos que queremos introducir algún valor en uno de los campos de la pantalla; la ejecución de esta tarea será realizada a través del PAI, y se utilizará el bloque de procesos CHAIN....ENDCHAIN. Este bloque comprueba si se ha producido alguna modificación en uno o más campos y, en caso afirmativo, llama al módulo donde se indica la acción a realizar. En una misma pantalla se pueden tener tantos bloques CHAIN como se requieran. En el ejemplo que se muestra debajo, el módulo invocado llamará a la subrutina que carga la descripción de la designación, mostrada anteriormente.

```

136      " 6993 - comprobar designación material y fabricante
137      chain.
138          field ztlb_muestras-iddesmat.
139          field ztlb_muestras-fabricante.
140          module pai_9000_nuevo_ladrillo on chain-request.
141      endchain.
142
143
144

```

Incluye **ZLBM_ERGRAL_PAI_9000** activo

```

614      ENDMODULE.
615      " PAI_9000_NUMMUESALBARAN INPUT
616
617      *-----*
618      * Module PAI_9000_NUEVO_LADRILLO INPUT
619      *-----*
620      * ASF3979 - 6993 - 30/06/2015
621      * Controla el cambio de designación y fabricante
622      *-----*
623      MODULE PAI_9000_NUEVO_LADRILLO INPUT.
624          perform f_cargar_descripcion_desig.
625          perform f_cargar_fabricante.
626      ENDMODULE.
        " PAI_9000_NUEVO_LADRILLO INPUT

```

Ilustración 67 - Chain..Endchain

4.3 Entrada de Resultados

Esta fase es la más extensa del proyecto y la que llevó más tiempo implementar y comprobar, ya que se deben realizar dieciséis DynPros, uno para cada ensayo, con sus correspondientes modules, subrutinas y módulos de funciones.

Para realizar esta tarea, se ha utilizado el programa ZLB_ERGRAL, compuesto por una única DynPro con un campo para escribir el resultado de un ensayo; esta entrada se utiliza para todos aquellos ensayos de materiales que aún no han sido implementados.

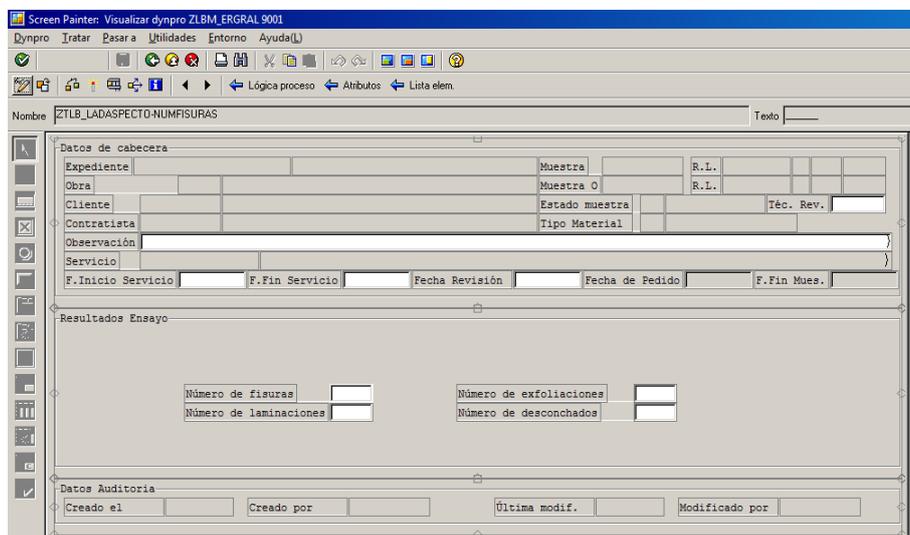


Ilustración 68 - Ejemplo DynPro E.R.

Estas entradas se han desarrollado de dos formas: los ensayos más sencillos (aquellos que solo deben mostrar unos pocos datos, como se muestra en la siguiente figura) se han realizado de forma similar a la Entrada de Muestras:

1. PBO: además de asignar los valores básicos de la ventana (como por ejemplo, el estado o nombre de la misma), debe llamar a un módulo que inicialice los campos de la ventana; este módulo llamará a la subrutina pertinente donde, tras comprobar que la tabla interna del ensayo⁹ está vacía (al entrar en el ensayo, esta tabla debe estar vacía ya que cuando se ejecuta el módulo aún no se ha mostrado la pantalla y, por lo tanto, no se ha podido escribir en ella), llamará a la función que realice esta carga de datos, guardándolos en la tabla interna, y asignar estos valores a los campos de la pantalla.

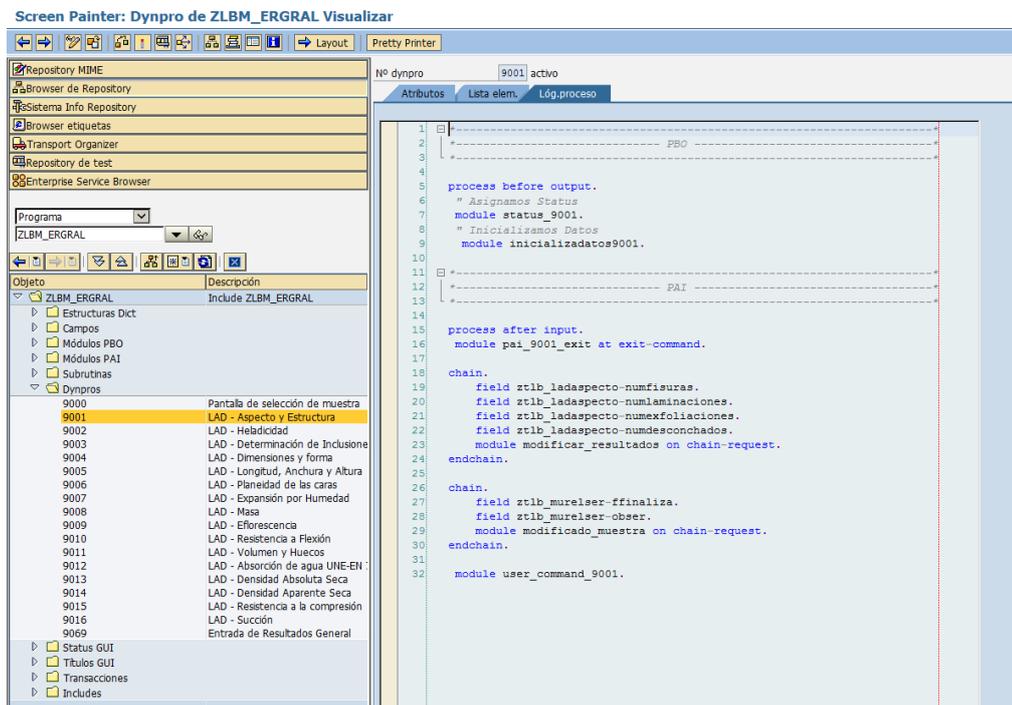


Ilustración 69 - Ejemplo código DynPro

⁹ Se ha creado una tabla interna en el TOP del programa para almacenar temporalmente los datos que se inserten o modifiquen por pantalla. De esta forma se opera solamente sobre esta tabla interna y, en caso de querer guardar los datos, se llama a una función que realice este proceso sobre la tabla de base de datos.

```

Include      ZLBM_ERGRAL_PBO_9001      activo
1  *-----*
2  ***INCLUDE ZLBM_ERGRAL_PBO_9001 .
3  *-----*
4  *&-----*
5  *&      Module STATUS_9001 OUTPUT
6  *&-----*
7  *      text
8  *-----*
9  module status_9001 output.
10  "Establecer Título de pantalla
11  set titlebar '9001'.
12  "Establecer estatus
13  perform setstatus using '9001'.
14
15  endmodule.                " STATUS 9001 OUTPUT
16  *-----*
17  *&-----*
18  *&      Module INICIALIZADATOS9001 OUTPUT
19  *&-----*
20  *      ASF3979 - 6993 - 30/06/2015
21  *      Module que inicializa los datos de la pantalla
22
23  module inicializadatos9001 output.
24  perform f_inicializadatos9001.
25  endmodule.                " INICIALIZADATOS9001 OUTPUT

```

Ilustración 70 - PBO Entrada Resultados

<pre> Include ZLBM_ERGRAL_F01 activo 1397 *&-----* 1398 *& Form F_INICIALIZADATOS9069 1399 *&-----* 1400 * ASF3979 - 6993 - 01/07/2015 1401 * Form que inicializa datos en la dynpro 1402 *-----* 1403 form f_inicializadatos9001 . 1404 1405 "técnico, f. realización y f. finalización 1406 perform f_cargar_datos_comunes. 1407 1408 "inicializar resultados (si los hubiera) 1409 perform f_cargar_ladaspecto. 1410 1411 endform. " F INICIALIZADATOS9001 </pre>		<pre> Include ZLBM_ERGRAL_F01 activo 2626 perform f_cargar_ladplaneidad. 2627 endform. " F INICIALIZADATOS9006 2628 *-----* 2629 *&-----* 2630 *& Form F_CARGAR_DATOS_COMUNES 2631 *&-----* 2632 * ASF3979 - 6993 - 23/07/2015 2633 * Carga datos de técnico, fecha realización y fecha finalización 2634 *-----* 2635 * ASF3979 - 6993 - 01/09/2015 2636 * Se incluye la inicialización del parámetro ZLAB para filtrar por laboratorio 2637 * al seleccionar equipos que se usan en los ensayos. 2638 *-----* 2639 * --> p1 text 2640 * <-- p2 text 2641 2642 form k_cargar_datos_comunes . 2643 if ztlb_murelser-idtecn_revis is initial. 2644 ztlb_murelser-idtecn_revis = sy-uname. 2645 endif. 2646 " Si no hay fecha de realización la inicializamos a la actual 2647 if ztlb_murelser-freal is initial. 2648 ztlb_murelser-freal = sy-datum. 2649 endif. 2650 " Si no tenemos fecha de revisión, la inicializamos a la actual 2651 if ztlb_murelser-frevisión is initial. 2652 ztlb_murelser-frevisión = sy-datum. 2653 endif. 2654 "Cargar parámetro idlab para la ayuda de búsqueda de los equipos utilizados 2655 if ztlb_muestras-idlab is not initial. 2656 set parameter id 'ZLAB' field ztlb_muestras-idlab. 2657 endif. 2658 endform. " F CARGAR DATOS COMUNES </pre>
<pre> de ZLBM_ERGRAL_F01 activo 1492 *&-----* 1493 *& Form F_CARGAR_LADASPECTO 1494 *&-----* 1495 * ASF3979 - 6993 - 01/07/2015 1496 * Inicializar la dynpro con los datos de ztlb_ladaspecto (si existen) 1497 *-----* 1498 * --> p1 text 1499 * <-- p2 text 1500 *-----* 1501 form k_cargar_ladaspecto . 1502 if t_ladaspecto is initial. "no se ha escrito en la pantalla anteriormente, cargamos de la 1503 call function 'Z_LB_LAD_CARGARASPECTO' 1504 exporting 1505 pi_m = wq_muestra 1506 pi_su = wa_navegacion-aufpl 1507 pi_ap = wa_navegacion-aplzl 1508 importing 1509 pe_aspecto = t_ladaspecto 1510 pe_error = x_error. 1511 *-----* 1512 if x_error is not initial. 1513 perform f_msg_error. 1514 endif. 1515 *-----* 1516 ztlb_ladaspecto-numfisuras = t_ladaspecto-numfisuras. 1517 ztlb_ladaspecto-numlaminaciones = t_ladaspecto-numlaminaciones. 1518 ztlb_ladaspecto-numexfoliaciones = t_ladaspecto-numexfoliaciones. 1519 ztlb_ladaspecto-numdesconchados = t_ladaspecto-numdesconchados. 1520 *-----* 1521 endform. " F_CARGAR_LADASPECTO </pre>		

Ilustración 71 - Inicializar datos ensayo

Como se aprecia en las imágenes superiores, se ha utilizado un diseño en capas por el cuál separar claramente la interfaz (capa de presentación) de los programas (capa de negocio) y datos (capa de negocio);

Los datos se extraerán mediante llamadas a módulos de funciones, pasando como parámetros de entrada las claves que identifican la muestra y el ensayo, y como parámetro de salida se obtendrá la tabla interna donde se almacenan temporalmente los datos. Los valores de esta tabla de salida serán asignados a los campos correspondientes del DynPro, para que muestren el contenido por pantalla.

Además, se ha incluido una estructura de control de errores, propia de SAP, por la cual se almacenará un valor determinado dependiendo de si la consulta a base de datos ha tenido éxito o no; dependiendo de este valor, se le asignará un tipo de mensaje, identificador, biblioteca donde buscarlos y, opcionalmente, parámetros que se usarán para formar el mensaje, que será lanzado para que el usuario pueda comprobar si la ejecución se ha realizado correctamente.

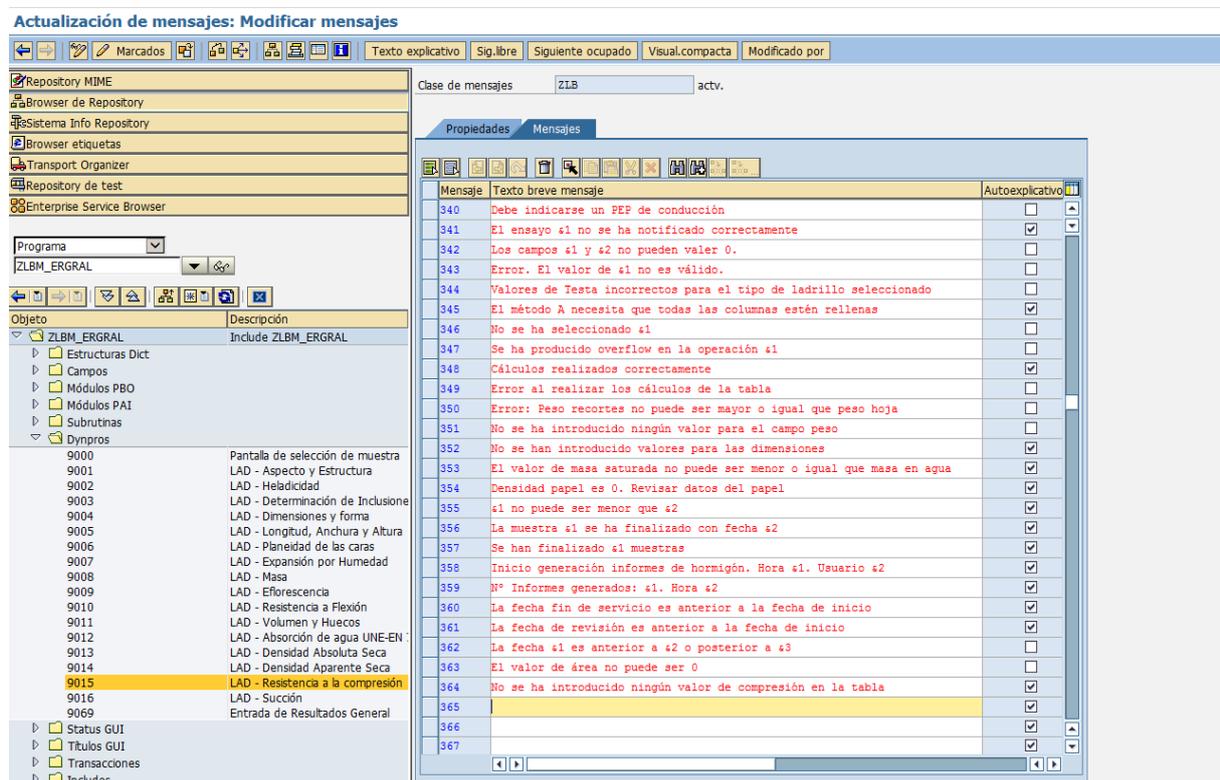


Ilustración 72 - Mensajes ABAP

En este caso, para controlar la ejecución de estos mensajes, se ha optado por comprobar que la variable propia del sistema 'sy-subrc' si la ejecución de una instrucción (asignación, consulta, etc.) ha sido correcta o, si por el contrario, ha habido algún fallo en ella. Por ejemplo, si tras realizar una consulta a la base de datos para hallar un registro, SY-SUBRC contiene el valor cero, la búsqueda

ha devuelto algún valor; si por el contrario, cuenta con algún otro valor, se puede comprobar que ha habido algún problema al ejecutarla.

Esta variable es muy útil de cara a encontrar errores y excepciones, por lo que se recomienda encarecidamente su uso cuando se realiza desarrollo software.

```
1 function z_lb_lad_cargaraspecto.
2  *-----
3  *""Interfase local
4  *  IMPORTING
5  *    REFERENCE(PI_M) TYPE ZE_IDMUESTRA
6  *    REFERENCE(PI_AU) TYPE CO_AUFPL
7  *    REFERENCE(PI_AP) TYPE CO_APLZL
8  *  EXPORTING
9  *    REFERENCE(PE_ASPECTO) TYPE ZTLB_LADASPECTO
10 *    REFERENCE(PE_ERROR) TYPE /BEV3/CHMSG
11 *-----
12 clear pe_error.
13 select single * from ztlb_ladaspecto
14   into pe_aspecto
15   where idmuestra = pi_m
16   and aufpl = pi_au
17   and aplzl = pi_ap.
18
19 if sy-subrc <> 0.
20   "error al cargar
21   pe_error-msgid = 'ZLB'.
22   pe_error-msgty = 'W'.
23   pe_error-msgno = '100'.
24 endif.
25
26 endfunction.
```

Ilustración 73 - Módulo de Función y comprobación de errores

2. PAI: está formado por aquellos módulos que requieran la intervención del usuario para realizar una acción, tales como que hacer tras pulsar el botón de cerrar programa, insertar un nuevo dato o pulsar los botones de avanzar a una nueva pantalla o guardar datos. Para ello, tal y como se ha indicado en la Entrada de Muestras, se usarán nuevos módulos y bloques CHAIN...ENDCHAIN para comprobar que se ha producido una interacción con la pantalla y la acción a realizar. Para esta pantalla se han creado 4 módulos, que servirán para salir, actualizar datos o indicar las acciones de los botones del menú superior.

```

Include      ZLBM_ERGRAL_PAI_9001      activo

4  *-----*
5  *      Module PAI_9001_EXIT INPUT
6  *-----*
7  *      ASF3979 - 6993 - 01/07/2015
8  *      Module para el exit
9  *-----*
10 module pai_9001_exit input.
11     perform f_salir_programa.
12 endmodule.          " PAI 9001 EXIT INPUT
13 *-----*
14 *      Module USER_COMMAND_9001 INPUT
15 *-----*
16 *      ASF3979 - 6993 - 01/07/2015
17 *      Module para las acciones de usuario
18 *-----*
19 module user_command_9001 input.
20     "data: p_msg type char1.
21     clear p_msg.
22     "wl_okcode = sy-ucomm.
23     wl_okcode = okcode01.
24     clear okcode01.
25
26     case wl_okcode.
27     when 'IR'.
28         perform f_navigarexpediente.
29     when 'SAVE'.
30         perform f_guardar_datos_9001.
31     when 'P+'.
32         perform f_avanzar.
33     when 'P-'.
34         perform f_retroceder.
35     when 'ADJUNTA'.
36         perform f_adjunta_doc_expediente
37             using ztlb_muestras-idplaninforme p_msg .
38     when 'EXIT'.
39         leave program.
40     when 'PICK'.
41         perform f_navigaciones.
42     when others.
43
44     endcase.
45     sy-ucomm = okcode01.
46
47 endmodule.          " USER COMMAND 9001 INPUT

```

Ilustración 74 - Módulos PBO

Al igual que se ha explicado anteriormente, cada subrutina llamará a otra o a un módulo de funciones, dependiendo de si quiere operar sobre los datos o realizar operaciones sobre las estructuras, variables o tipos del programa.

A continuación se mostrará el ejemplo de la función que guarda los datos en la tabla correspondiente de la base de datos; a esta función se le pasarán como parámetros de entrada los datos que identifican a la muestra y ensayo y, como parámetro de entrada-salida, la tabla auxiliar. El motivo de que esta tabla se pueda usar tanto de entrada como salida es por cuestiones de auditoría, ya que antes de guardar se actualizarían estos datos y la tabla contendría toda la información de los resultados actualizada la siguiente vez que accediese al PBO, por lo que no necesitaría cargar ninguna información nueva.

También se utilizará la variable SY-SUBRC para controlar si la operación de guardado se ha realizado con éxito o no, informando al usuario mediante un mensaje.

Módulo funciones Z_LB_GUARDAR_LAD_ASPECTO Activo

Atributos Import Export Modif. Tablas Excep. Cód.fte.

```

10  *" CHANGING
11  *" REFERENCE(PC_RESULT) TYPE ZTLB_LADASPECTO
12  *"-----
13  * ASF3979 - 6993 - 01/07/2015
14  *Guarda en la tabla del ensayo el resultado del mismo e información de auditoria
15  *"-----
16  * ASF3979 - 6993 - 20/07/2015
17  * Modificación de la función para utilizar la función de auditoría general creada
18  * y devolviendo la estructura de errores para lanzarlos desde la dynpro
19  *"-----
20  clear pe_error.
21  if p_muestras is not initial and p_aufpl is not initial and p_aplzl is not initial.
22      pc_result-idmuestra = p_muestras.
23      pc_result-aufpl = p_aufpl.
24      pc_result-aplzl = p_aplzl.
25
26      call function 'Z_LB_LAD_GUARDARAUDITORIA'
27      exporting
28          pi_ersda = pc_result-ersda
29          pi_erzei = pc_result-erzei
30          pi_ernam = pc_result-ernam
31      importing
32          pe_ersda = pc_result-ersda
33          pe_erzei = pc_result-erzei
34          pe_ernam = pc_result-ernam
35          pe_laeda = pc_result-laeda
36          pe_aezet = pc_result-aezet
37          pe_aenam = pc_result-aenam
38      .
39
40      modify ztlb_ladaspecto from pc_result.
41
42  if sy-subrc <> 0. "control de errores
43      pe_error-msgid = 'ZLB'.
44      pe_error-msgty = 'W'.
45      pe_error-msgno = '033'.
46  else. "se ha guardado correctamente
47      pe_error-msgid = 'ZLB'.
48      pe_error-msgty = 'S'.
49      pe_error-msgno = '003'.
50  endif.

```

Ilustración 75 - Guardar datos ensayo

La función que guarda la auditoría comprueba si los campos que indican la creación del registro (fecha, hora y usuario) están vacíos; si es así, se les asigna estos valores de las variables del sistema; en caso contrario, se actualizan los otros tres valores de salida (fecha, hora de la modificación y usuario que la realiza).

4.4 Creación de Informes

El desarrollo de la función que permite crear Informes a partir de una muestra del material Ladrillo y sus correspondientes servicios se desarrollará en dos partes: visual y desarrollo de código.

La primera, la parte visual, se elaborará a través de la transacción de SAP 'ZGC_PLANTILLAS', que permite la inserción y tratamiento de un documento, previamente guardado en el sistema, que será la plantilla de nuestro futuro informe.

Primero, desde la transacción 'SCASE', se carga el documento que funcionará como plantilla (documento Word o Excel) y se guarda con el nombre que queremos asignarle.



Ilustración 76 - Creación de plantilla

Posteriormente, desde la transacción 'ZGC_PLANTILLAS', se incluye y se clasifica según el módulo donde queremos incluirlo.

Plantillas

Plantilla | Gestionar plantillas | Incluir plantillas | Editar cat. campos | Fórmulas

Act	Tipo	Grupo	Subg	Tipo	Descripción	Plantilla
	WORD_DOC	EDI			prueba_ehv4	SRM_DOC04 DFCBDFBD08B186F19E24005056AC7B91
	WORD_DOC	EDI			prueba_ehv5	SRM_DOC04 DFCBE056610F84F19E24005056AC7B91
	WORD_DOC	LAB	SUE	INF	Plantilla ejemplo	SRM_DOC04 DFC038FC2C8434F19E24005056AC7B91
	EXCEL_XLS	LAB	SUE	BORR	Borrador ejemplo	SRM_DOC04 DFCFA0BFADB954F19E24005056AC7B91

Ilustración 77 - Selección de Plantilla

Ya sólo tenemos que acceder a la plantilla en cuestión para realizar las modificaciones.

Esta pantalla se encontrará dividida en dos: a la izquierda veremos una serie de tablas, donde se insertarán las fórmulas que devolverán los datos correspondientes a los campos, tablas o imágenes que queramos incluir en el documento; esta parte será explicada más adelante, una vez que se haya hablado sobre el código a desarrollar. A la derecha, aparecerá el documento en cuestión, para que lo modifiquemos incluyendo las tablas y campos que necesitamos que aparezcan en él.

La inclusión de campos se hace de la siguiente manera: desde las propiedades del documento se crea un campo con el nombre de la fórmula que devuelve este valor; posteriormente, se debe incluir en el documento a través de la opción Insertar -> Campo.

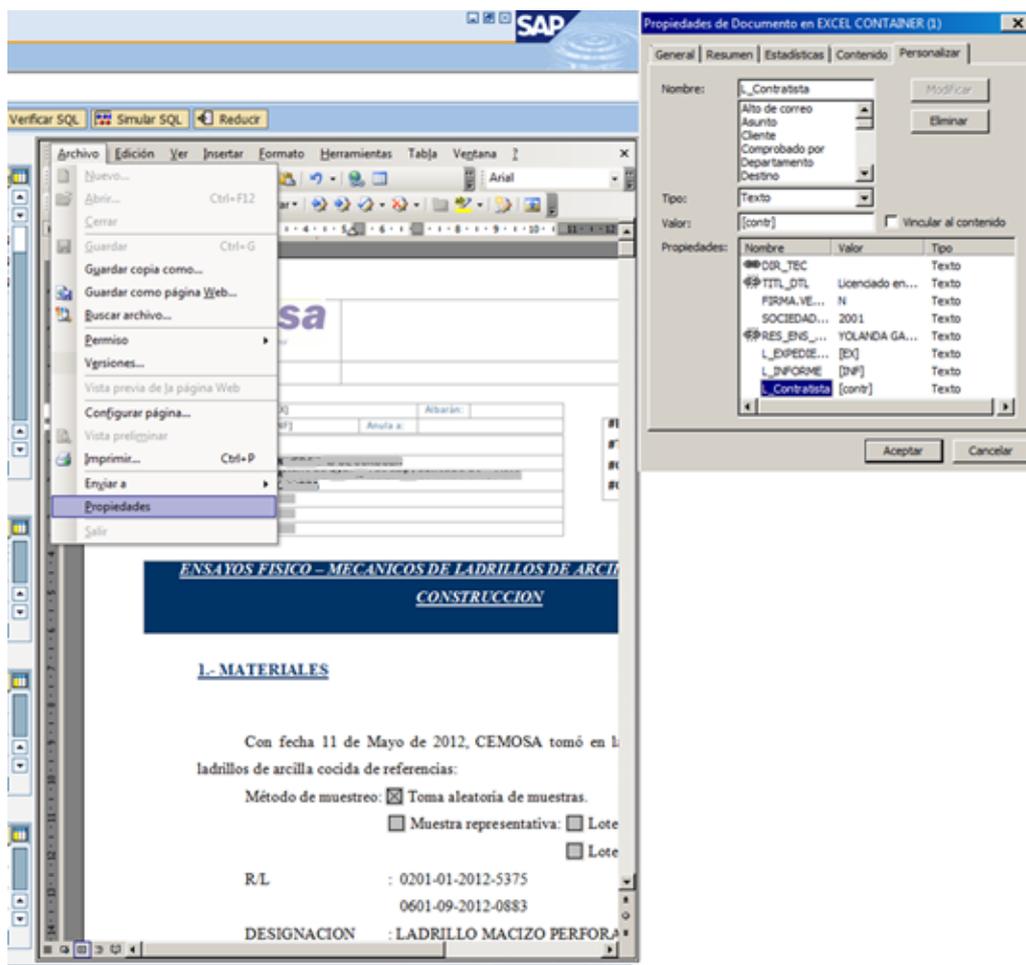


Ilustración 78 - Crear campo plantilla

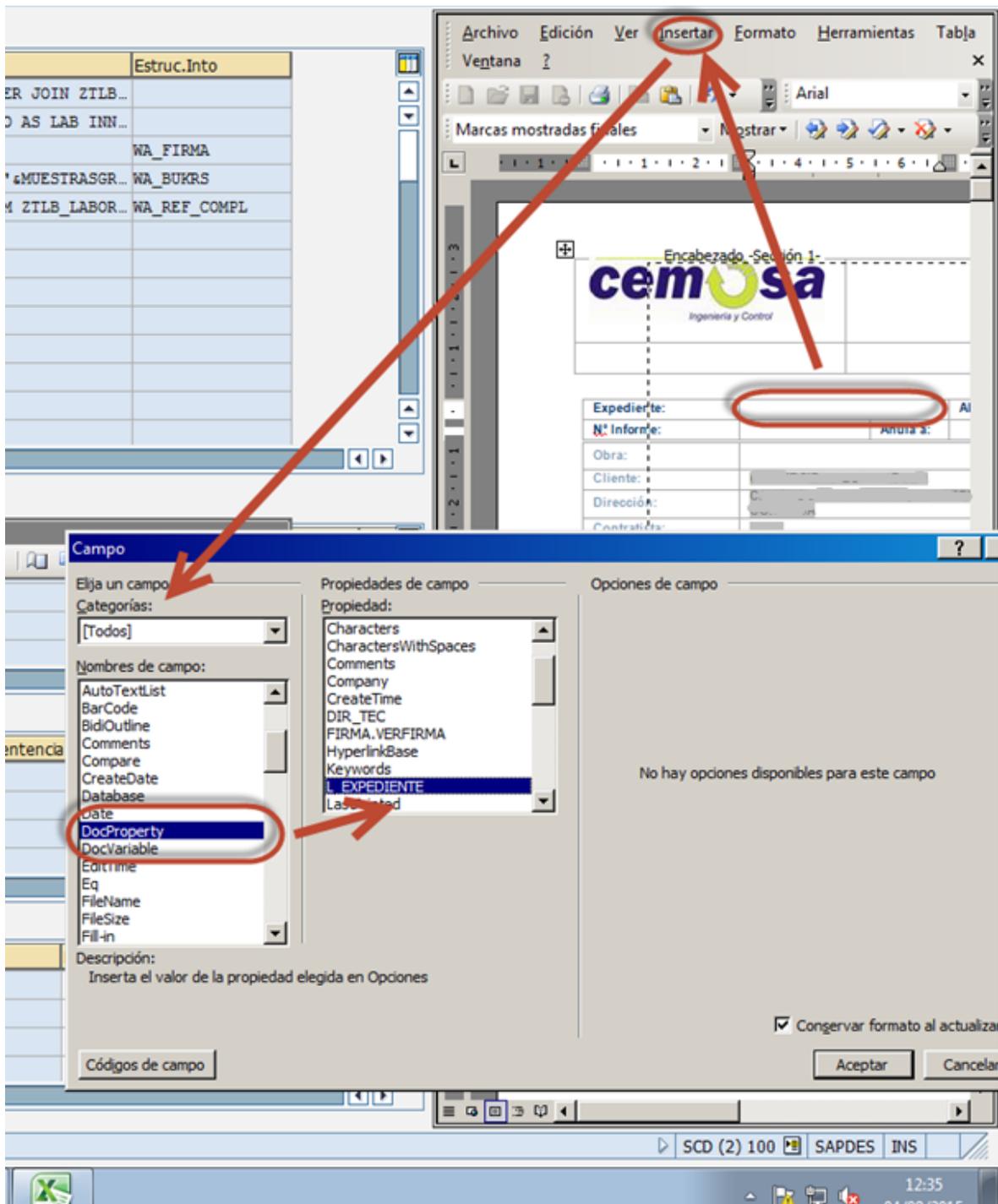


Ilustración 79 - Insertar campo

Para insertar datos en forma de tabla, se deberá crear una tabla, mediante la herramienta incluida en Word, con tantas columnas como campos del registro se quieran mostrar, y dos filas: una funcionará como encabezado, que tendrá un formato especial y el título de cada campo; la otra fila tendrá el formato que queramos que se repita por cada registro y será en la cual se comenzarán a incluir los registros de una tabla a través de la fórmula diseñada para este fin.

Posteriormente se volverá a esta parte y se explicará el funcionamiento de las fórmulas, necesarias para mostrar los datos. Pero antes, es necesario comentar cómo funciona el desarrollo del código fuente que extraerá los valores a mostrar de la base de datos y su apropiado cambio de formato, tanto para mostrar estos datos con la precisión que pide cada norma como adaptarlos a una posible plantilla en Excel.

Para ello, y al igual que en las plantillas ya implementadas en el sistema, se creará un módulo de funciones al que se le pasan como parámetros de entrada los valores que identificarán el informe y algunos campos a insertar (como la firma) y devolverá tres estructuras de datos: una para los campos, otra para las tablas y otra para parámetros necesarios a la hora de generar el informe.

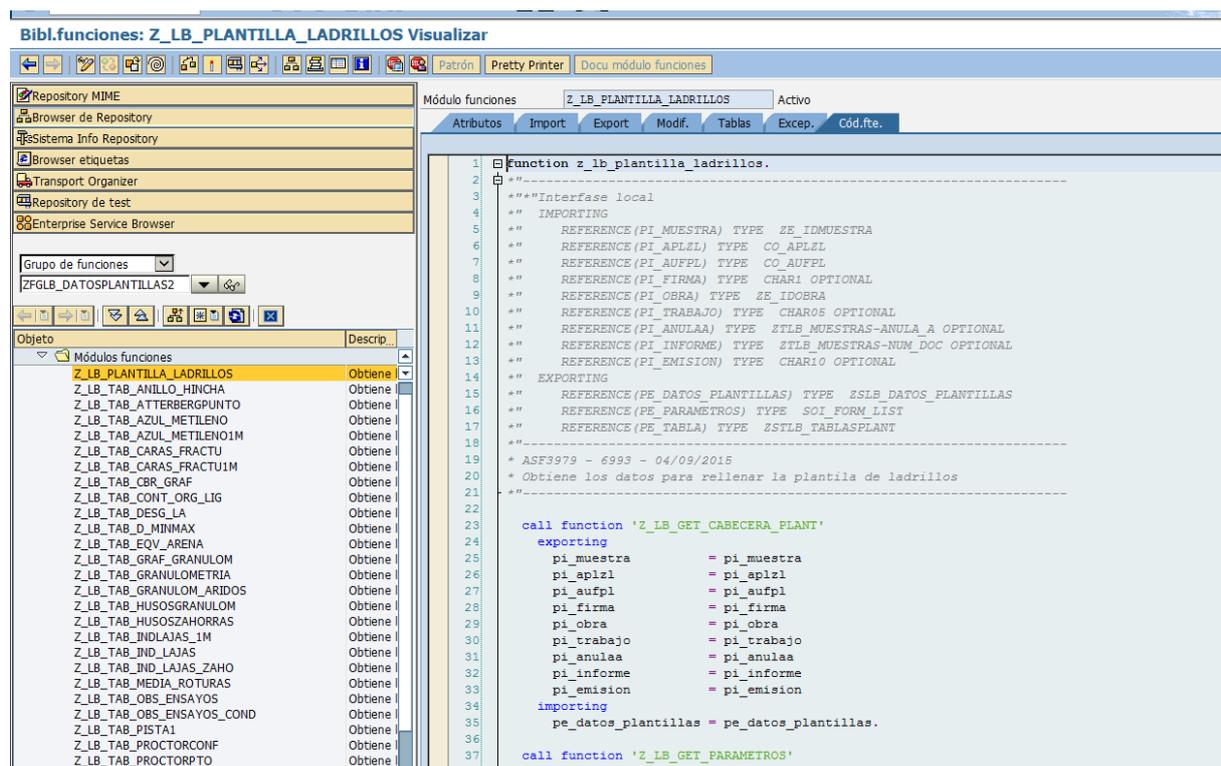


Ilustración 80 - FM plantilla ladrillo

De las dos estructuras de salida que se han mencionado anteriormente, se usarán dos en la creación del nuevo código:

- 'zslb_datos_plantillas': Esta estructura se aprovecha de la definición de una estructura ABAP por la cual puede contener campos de una tabla u otras estructuras que referencien a tablas. Estará formada por varias estructuras, cada una de las cuales fue creada para separar los valores a devolver según al material al que pertenezcan o la función que tendrán en la creación del informe. Mencionar que al ser un tipo del diccionario ABAP, no guarda ningún dato en la base de datos, sino que los guarda en memoria durante su ejecución y posteriormente se eliminan.

Cada uno de los campos de la estructura no tendrá el formato de sus homólogos en base de datos, sino que contará con el formato final con el que se mostrarán en el informe; esto es debido a que durante una asignación entre tipos distintos compatibles (por ejemplo, un número con tres decimales y otro con uno), ABAP realiza la conversión redondeando el valor o añadiendo los ceros necesarios.

The image displays two screenshots of the SAP 'Dict: Actualizar estructura' (Structure Maintenance) interface. The top screenshot shows the structure 'ZSLB_DATOS_PLANTILLAS' with a description 'Datos para las plantillas de laboratorio'. The bottom screenshot shows the structure 'ZSLB_INFORMELADRILLO' with a description 'Estructura para los datos del informe de ladrillos'.

Top Screenshot: ZSLB_DATOS_PLANTILLAS

Componente	TipR	Tp.componente	Tipo de dat.	Long.	Decim.	Descripción breve	Grupo
.INCLUDE		ZSLB_RESISTADH7	NUM	0	0	0 Datos de puntos para la resistencia del hormigón para plant.	
.INCLUDE		ZSLB_RESISTADH8	NUM	0	0	0 Datos de puntos para la resistencia del hormigón para plant.	
.INCLUDE		ZSLB_RESISTADH9	NUM	0	0	0 Datos de puntos para la resistencia del hormigón para plant.	
.INCLUDE		ZSLB_RESISTADH10	NUM	0	0	0 Datos de puntos para la resistencia del hormigón para plant.	
.INCLUDE		ZSLB_DESG_LA_SM	NUM	0	0	0 Tabla desgaste para las plantillas	
.INCLUDE		ZSLB_AZULMETIENO	NUM	0	0	0 Datos azul de metileno para la plantilla	
.INCLUDE		ZSLB_CARFRACIAR	NUM	0	0	0 Datos cara fractura para las plantillas	
.INCLUDE		ZSLB_LIMPIAR	NUM	0	0	0 Datos zahorra para las plantillas	
.INCLUDE		ZSLB_CONTLIG	NUM	0	0	0 Datos contenido en ligante para las plantillas	
.INCLUDE		ZSLB_NUECHARRIT	NUM	0	0	0 Datos NLT para las plantillas	
.INCLUDE		ZSLB_PRESA_HOR	NUM	0	0	0 Datos prensa de hormigón para las plantillas	
.INCLUDE		ZSLB_MARSHALLMT	NUM	0	0	0 Datos Marshall para las plantillas	
.INCLUDE		ZSLB_DATOS_PLAN	NUM	0	0	0 Datos plantillas de laboratorio	
.INCLUDE		INFORMELADRILLO	NUM	0	0	0 Estructura para los datos del informe de ladrillos	

Bottom Screenshot: ZSLB_INFORMELADRILLO

Componente	TipR	Tp.componente	Tipo de dat.	Long.	Decim.	Descripción breve	Grupo
SOCIEDADNOMBRE		BUTXT	CHAR	25		0 Denominación de la sociedad o empresa	
DESCDESIGNACION		NAME1_GP	CHAR	35		0 Nombre 1	
FABRICANTE LADR.		NAME1_GP	CHAR	35		0 Nombre 1	
RLORIGEN		ZE_RLORIGEN	CHAR	20		0 Concatenación de los 4 campos que forma la R/L	
METODO1LAA		XFELD	CHAR	1		0 Casilla de selección	
METODO2LAA		XFELD	CHAR	1		0 Casilla de selección	
METODO3LAA		XFELD	CHAR	1		0 Casilla de selección	
VMSOGA		ZE_VMLD	DEC	7		0 Cálculo del valor medio de la dimensión del ladrillo (mm)	
VMTIZON		ZE_VMLD	DEC	7		0 Cálculo del valor medio de la dimensión del ladrillo (mm)	
VNGRUESO		ZE_VMLD	DEC	7		0 Cálculo del valor medio de la dimensión del ladrillo (mm)	
VMSOGA		ZE_VNLDINF	CHAR	5		0 Muestra valor o '' si está vacío	
VMTIZON		ZE_VNLDINF	CHAR	5		0 Muestra valor o '' si está vacío	
VNGRUESO		ZE_VNLDINF	CHAR	5		0 Muestra valor o '' si está vacío	
RECORRIDOSOGA		ZE_DATOSINFORME	CHAR	20		0 tipo caracter para mostrar valores en informe	
RECORRIDOTIZON		ZE_DATOSINFORME	CHAR	20		0 tipo caracter para mostrar valores en informe	
RECORRIDOGRUESO		ZE_DATOSINFORME	CHAR	20		0 tipo caracter para mostrar valores en informe	
DESVIACIONESOGA		ZE_DESVIACIONLDR	CHAR	5		0 Muestra valor o '' si está vacío	
DESVIACIONTIZON		ZE_DESVIACIONLDR	CHAR	5		0 Muestra valor o '' si está vacío	
DESVIACIONGRUESO		ZE_DESVIACIONLDR	CHAR	5		0 Muestra valor o '' si está vacío	
TABLADM		ZE_PARALELISMOI	CHAR	10		0 Tipo char10, almacena valor, '' o 'CNEP'	
TABLAFM		ZE_PARALELISMOI	CHAR	10		0 Tipo char10, almacena valor, '' o 'CNEP'	

Ilustración 81 - Datos para plantilla

- La otra estructura, 'ztlb_tablasplant' fue creada por un miembro del equipo de desarrollo hace un tiempo. Esta estructura, de tipo tabla (por lo que puede tener varios registros almacenados temporalmente) está formada por unos campos que identifican a la muestra junto a otro, de tipo cadena, en el que se almacena el nombre de la tabla que queremos almacenar; es decir, este tipo tendrá en su interior registros de varias tablas, siendo posible acceder a ellas a través de su nombre. El resto de campos, de tipo cadena, permite homogenizar todas las salidas, ya que tendrán el mismo tipo de datos. La conversión entre tipos se realiza de forma automática también, pero se debe realizar una tarea adicional: Los valores numéricos en SAP tienen un formato incompatible con los valores numéricos de Excel¹⁰ (ya que la coma funciona como separador de millares y el punto como separador de números enteros y decimales), por lo que se debe llamar a una función que realice este cambio. A continuación se muestra una imagen desde el debugger de ABAP con esta estructura rellena:

The screenshot shows the SAP ABAP debugger interface. The main window displays a table structure named 'PE_TABLA' with the following columns: LÍNEA, MANDT(C(3)), NOMBRE(C(80)), IDMUJESTRA(C(12)), CONT(I(4)), CAMPO1(C(200)), CAMPO2(C(200)), CAMPO3(C(200)), and CAMPO4(C(200)). The table contains 24 rows of data, including entries for 'SERV_NORMA_MUE' and 'LONGANALT'. A callout box on the right contains the following text:

Se almacenan todos los datos que se rellenarán en el informe clasificados por el campo nombre, donde especificaremos a que tabla corresponden.

A la hora de crear la tabla en la plantilla, hacer select con idmuestra y nombre

Ilustración 82 - Estructura para tablas

¹⁰ En el formato SAP se utiliza el carácter coma (,) para separar los millares y el carácter punto (.) para separar los números enteros de los decimales. En Excel, este formato es justamente el contrario: coma para separar decimales y punto para separar millares.

Adicionalmente, desde el Diccionario de Datos de ABAP y para aquellos ensayos en los que se necesite mostrar los datos con una precisión distinta a la utilizada en los cálculos, se creará una estructura en la que cada campo tendrá su tipo de datos final. Así, con una única sentencia 'move-corresponding', todos los datos se copiarán en esta nueva estructura con la precisión adecuada, no teniendo que realizar cambios sobre cada uno de ellos.

Una vez introducidos los tipos a usar, se va a proceder a comentar el funcionamiento de esta función. En el interior de este módulo, se crearán distintas llamadas a módulos de funciones, de cara a separar las diferentes obtenciones de datos según su procedencia, de forma modular: existirá un módulo que cargue la designación, otro que cargue la lista de ensayos que se realicen, otros que carguen los datos de cada uno de los ensayos, etc.

Para obtener los datos resultados o datos que se insertarán como campos en el informe, el procedimiento a seguir es el siguiente: se obtiene el registro a través de una consulta a base de datos, almacenándolo en una estructura del mismo tipo creada durante la ejecución de la función (también es posible crear una estructura que difiera en el número de campos, obteniendo solamente aquellos que son necesarios).

Tras ajustar la precisión de estos datos tal y como se ha indicado anteriormente (en caso necesario), se llama a la función que ajusta este formato al mismo que Excel y se guarda el nuevo valor en su campo correspondiente de la estructura final ('zslb_datos_plantillas').

```

Módulo funciones      Z_IB_GET_LDVALORESIAA      Activo
-----
Atributos  Import  Export  Modif.  Tablas  Excep.  Cód.fte.
-----
13  * ASF3979 - 6993 - 16/09/2015
14  * Uso de una estructura del Diccionario de Datos, donde los campos ya tienen
15  * el formato de salida del informe (nº de decimales ya cambiados).
16  * Tras el select, pasamos los datos a la estructura y realizamos las operaciones posteriores.
17  *-----*
18  data: wa_res type zslb_ldinformelaa.
19  data: wa_laa type zslb_ladlonanalt.
20
21  select single idmetodo vmsoga vmtizon vmgrueso
22                vnsoga vntizon vngrueso
23                recorridosoga recorridotizon recorridogrueso
24                desviacionsoga desviaciontizon desviaciongrueso
25
26  from ztlb_ladlonanalt
27  into corresponding fields of wa_laa
28  where idmuestra = pi_muestra.
29
30  move-corresponding wa_laa to wa_res. "ya tenemos los valores con decimales correctos
31
32  "Método
33  if wa_res-idmetodo = 1.
34    pe_datos_plantillas-metodo1laa = 'X'.
35  elseif wa_res-idmetodo = 2.
36    pe_datos_plantillas-metodo2laa = 'X'.
37  elseif wa_res-idmetodo = 3.
38    pe_datos_plantillas-metodo3laa = 'X'.
39  endif.
40
41  "VM
42  pe_datos_plantillas-vmsoga = wa_res-vmsoga.
43  pe_datos_plantillas-vmtizon = wa_res-vmtizon.
44  pe_datos_plantillas-vmgrueso = wa_res-vmgrueso.
45
46  "VN
47  if wa_res-vnsoga is initial.
48    pe_datos_plantillas-vnsoga = '-'.
49  else.
50    pe_datos_plantillas-vnsoga = wa_res-vnsoga.

```

Ilustración 83 - Cargar campos del informe

En cuanto a los módulos de funciones que tablas completas, el funcionamiento es similar pero con ciertas diferencias.

En primer lugar, tras obtener los registros a través de una consulta a la base de datos, se almacenan en una variable de tipo tabla, ya sea igual a la tabla a la que hacemos la consulta o, como se ha dicho antes, creándola en la función con los campos necesarios.

Se deberá recorrer la tabla creada para modificar los valores (en caso necesario) y almacenarlos en la estructura que devolverá estos valores; para ello se utiliza un bucle 'for', que recorrerá la tabla devolviendo su posición actual, y se almacenará el registro completo en una variable de tipo 'field-symbol'¹¹. A partir de ella, se podrá acceder a cada campo del registro y hacer las tareas pertinentes. Por último, al estar trabajando sobre cadenas (recordar que la estructura muestra los datos en forma de cadena de caracteres), se realiza la operación 'condense' sobre cada una (elimina los espacios en blanco que pueda haber a la izquierda del valor) y 'append', que añade el registro completo a la estructura de tipo tabla. Se debe observar también como al inicio del bucle, se actualizan los valores a través de los que accederemos a estos registros para insertarlos en las tablas del informe, como el identificador de muestra, el nombre de la tabla que añadimos y la posición del registro.

```

Módulo funciones  Z_LB_GET_LD_INFHEUCOS  Activo
Atributos  Import  Export  Modif.  Tablas  Excep.  Cód.fte.
25  * Modificación para usar la estructura de informe, de forma que los valores ya estén redondos
26  **-----**
27  data: wa_res type ztlb_ldinformehuecos.
28
29  data: t1_huecos type table of ztlb_ldmedhuecos,
30        w1_tabla like line of pe_tabla,
31        wa_huecos type ztlb_idvolhuecos.
32
33  field-symbols: <fs_huecos> like line of t1_huecos.
34
35  select * into corresponding fields of table t1_huecos
36        from ztlb_ldmedhuecos
37        where idmuestra = pi_muestra.
38
39  loop at t1_huecos assigning <fs_huecos>.
40    clear w1_tabla.
41    move-corresponding <fs_huecos> to wa_res.
42
43
44    w1_tabla-idmuestra = pi_muestra.
45    w1_tabla-nombre = 'MEDHUECOS'.
46    w1_tabla-cont = sy-tabix.
47
48    w1_tabla-campo1 = wa_res-probeta.
49
50  *dimensiones
51  call function 'Z_LB_CAMBIAR_FORMAT_DEC'
52    exporting
53      pi_formato_sap = wa_res-longitud
54    importing
55      pe_formato_excel = w1_tabla-campo2.
56
57  call function 'Z_LB_CAMBIAR_FORMAT_DEC'
58    exporting
59      pi_formato_sap = wa_res-ancho
60    importing
61      pe_formato_excel = w1_tabla-campo3.
62
63  call function 'Z_LB_CAMBIAR_FORMAT_DEC'
64    exporting
65      pi_formato_sap = wa_res-alto
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88  condense w1_tabla-campo7. "Comprimir para eliminar espacios y poder concatenar correctamente
89  concatenate w1_tabla-campo7 'x10^4' into w1_tabla-campo7.
90
91  call function 'Z_LB_CAMBIAR_FORMAT_DEC'
92    exporting
93      pi_formato_sap = wa_res-volumenneto
94    importing
95      pe_formato_excel = w1_tabla-campo8.
96
97  condense w1_tabla-campo8. "Comprimir para eliminar espacios y poder concatenar correctamente
98  concatenate w1_tabla-campo8 'x10^4' into w1_tabla-campo8.
99
100  call function 'Z_LB_CAMBIAR_FORMAT_DEC'
101    exporting
102      pi_formato_sap = wa_res-volumenhuecos
103    importing
104      pe_formato_excel = w1_tabla-campo9.
105
106  condense w1_tabla-campo9. "Comprimir para eliminar espacios y poder concatenar correctamente
107  concatenate w1_tabla-campo9 'x10^4' into w1_tabla-campo9.
108
109  call function 'Z_LB_CAMBIAR_FORMAT_DEC'
110    exporting
111      pi_formato_sap = wa_res-porchuecos
112    importing
113      pe_formato_excel = w1_tabla-campo10.
114
115  condense w1_tabla-campo1.
116  condense w1_tabla-campo2.
117  condense w1_tabla-campo3.
118  condense w1_tabla-campo4.
119  condense w1_tabla-campo5.
120  condense w1_tabla-campo6.
121  condense w1_tabla-campo7.
122  condense w1_tabla-campo8.
123  condense w1_tabla-campo9.
124  condense w1_tabla-campo10.
125
126  append w1_tabla to pe_tabla.
127  endloop.

```

Ilustración 84 - Carga de tablas para informe

Por último, en este módulo de funciones, existirá un Módulo adicional que se encargará de comunicar los datos de estas estructuras con el programa que realiza la

¹¹ Una variable de tipo field-symbol es un puntero cuyo contenido es la dirección de un objeto. Al asignar un field-symbol a un registro de una tabla, la variable apuntará al registro en cuestión, permitiendo el acceso o modificación de los valores almacenados.

creación de informes, por lo que tan sólo tendremos que crear las fórmulas que seleccionen los datos apropiados para su realización.

En la transacción 'zgc_plantillas', se ha indicado que la pantalla aparece dividida en dos partes: una donde se introducen las fórmulas de selección y otra donde se realizan las modificaciones visuales y se insertan campos y tablas.

Volviendo a la primera zona, esta se encuentra dividida en cuatro tablas: una donde declarar variables de tipo campo, otra para declarar variables de tipo tabla, otra para declarar variables de tipo gráficas y la última para declarar imágenes.

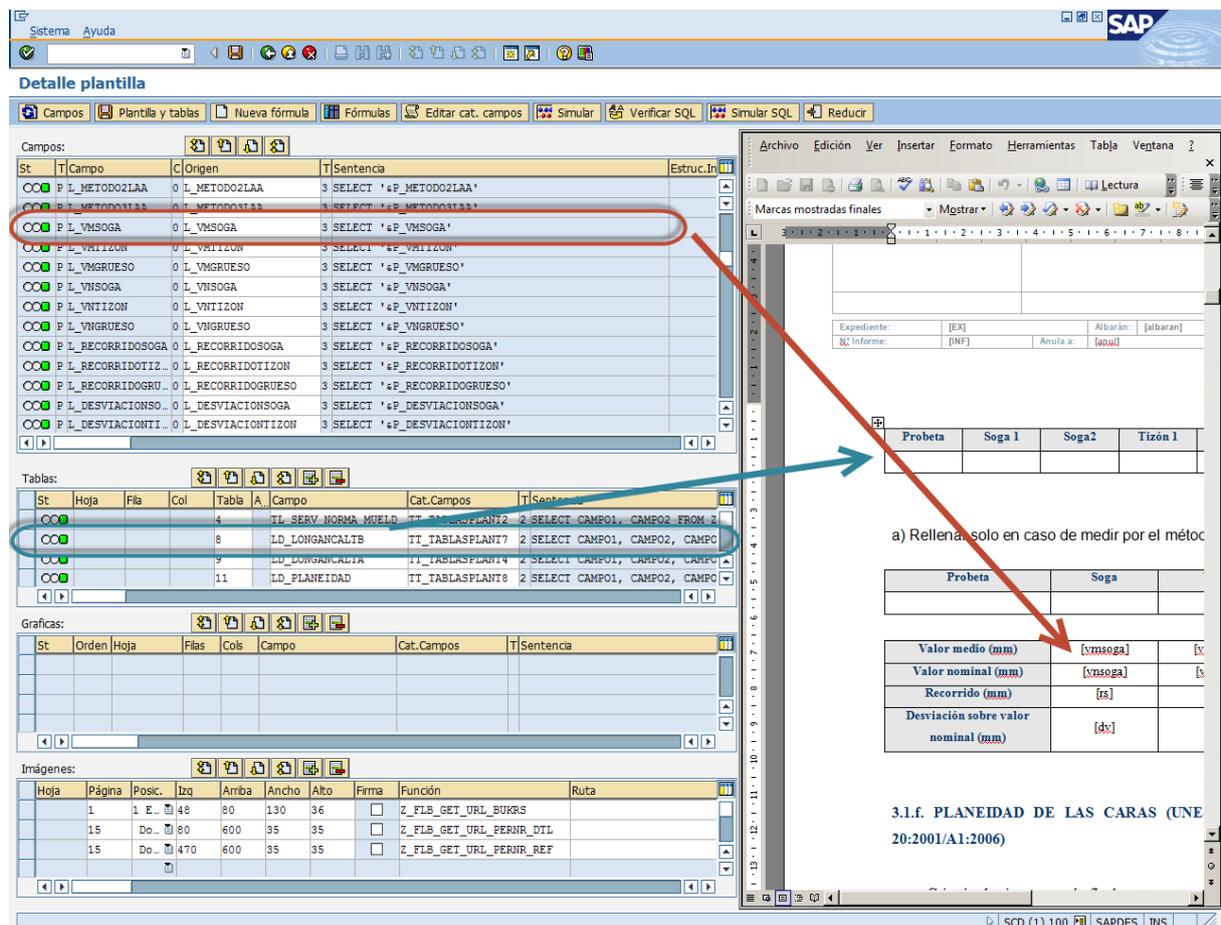


Ilustración 85 - Plantilla y fórmulas

Tal y como se aprecia en la tabla superior, se inserta una fórmula para cada elemento que se quiera mostrar en el informe. Para ello se debe pulsar sobre el botón 'Nueva Fórmula', que se encuentra en el menú superior, y crear la fórmula haciendo referencia al origen (campo de la estructura), y nombre de destino (que será el que se utilice para la creación del campo en el documento Word).

Las tablas funcionan de forma similar, pero se deberán añadir dos valores más: uno indica la posición de la tabla en el documento (es decir, si es la primera tabla que aparece, la segunda, la décima, etc.) y la otra será la variable de tipo tabla donde se almacenarán. Esta variable se crea a través del botón 'Editar cat. campos', indicando

el número de campos con los que contará cada registro (de ahí el número que se añade tras el nombre; por ejemplo, 'TT_TABLASPLANT7' contará siete campos en cada registro). La razón de que se deban crear varias tablas similares pero con distintas columnas es que cada ensayo cuenta con datos diferentes, por lo que si se crease una común, sobrarían o faltarían columnas.

En cuanto a las imágenes, se les indica el nombre del campo donde se encuentran almacenado su valor y la posición en el documento (si es parte de la cabecera o del cuerpo, hoja en la que se debe insertar y coordenadas de la posición).

Una vez definida la fórmula, se debe crear la consulta que servirá para devolver los valores almacenados en memoria interna al informe.

El campo 'TIPOSELECT' indica si se va a devolver un valor único (tipo 3), una tabla (tipo 2) o una imagen (tipo 1). Se introduce el nombre del campo, que será el que se añadirá al documento, y la consulta que devolverá este valor y lo almacenará en la variable anteriormente indicada.

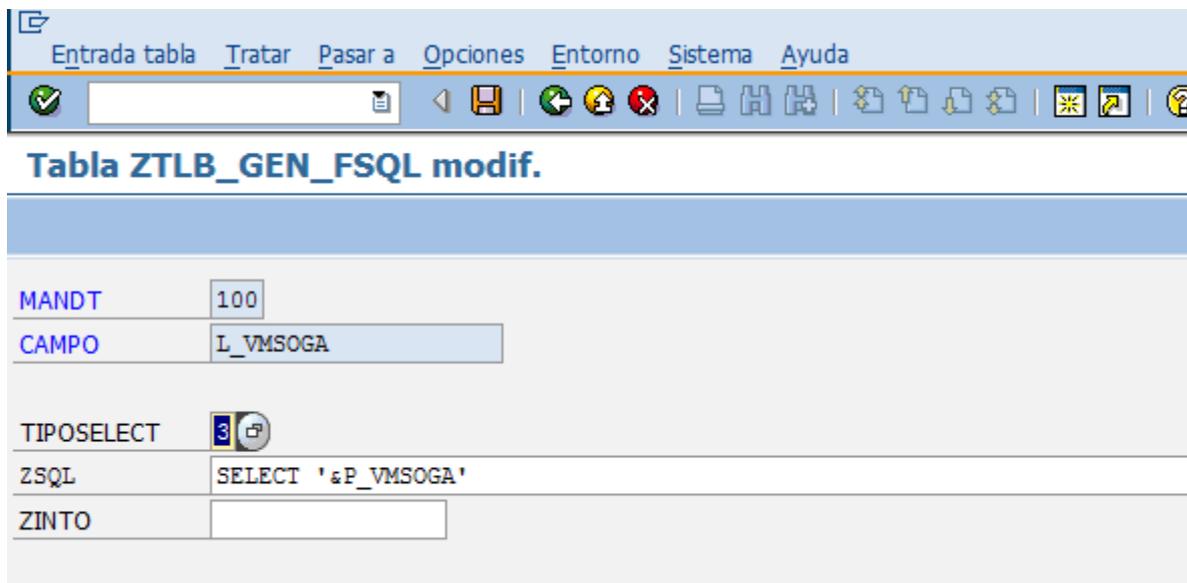


Ilustración 86 - Obtención de un campo

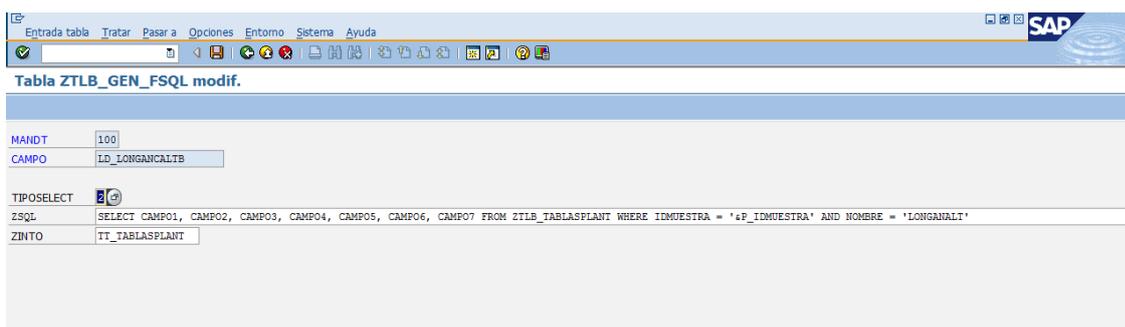


Ilustración 87 - Obtención de una tabla

5.Pruebas e Implantación

Tras desarrollar los nuevos módulos, comienza una fase de vital importancia en todo proceso de desarrollo software: el testeo y validación de los módulos, de forma que se puedan encontrar errores producidos en las fases anteriores o corroborar su correcto funcionamiento.

Esta fase es de vital importancia porque, hasta ahora, estos módulos solamente han sido usados por el equipo de desarrollo y con datos de prueba. Al ser parte de un sistema ERP de una Empresa, debemos asegurar que no se produzcan errores cuando el usuario final sea el que maneje el Sistema, ya que podría suponer pérdidas, tanto de tiempo como, en el peor de los casos, de los datos almacenados en el Sistema.

SAP, como cualquier entorno de software de gestión empresarial, presenta la necesidad de tener sistemas complejos (hardware y software) separados dedicados a funciones específicas. Algunos ejemplos de estas funciones pueden ser el desarrollo de software, las pruebas del mismo, formación a los usuarios finales y, la más importante de todas, la puesta en producción del software.

El Sistema SAP R/3 utilizado ha contado con tres de estos escenarios, permitiendo realizar tareas específicas en cada uno de ellos: desarrollo, integración y producción.

- Sistema de desarrollo: es el sistema inicial donde se origina el software. Todos los desarrollos y parametrizaciones se llevan a cabo aquí y, una vez realizadas las pruebas unitarias de los programas, se transportan al Sistema de Integración. El Sistema de Desarrollo suele contar con pocos datos, ya que se van creando como pruebas y, en ocasiones, son inconsistentes.
- Sistema de Integración: en este sistema se realizan pruebas definitivas del software: pruebas integradas, pruebas de rendimiento, pruebas de usuario y pruebas de transporte.
- Sistema de producción: este sistema tiene una única función: la explotación real del software. Es donde se almacenan los datos reales de la empresa y donde se ejecutan los procesos del negocio. Antes de transportar los programas o parametrizaciones a este sistema, se debe garantizar que estos no afecten ni al trabajo productivo ni a los datos reales.

Por lo tanto, las pruebas básicas se realizan en Desarrollo. Una vez superadas estas pruebas, se transportan los programas, tablas y demás componentes del proyecto a Integración. Si en este sistema se detectan fallos, toda modificación de datos o código debe realizarse en Desarrollo, volviendo a transportar a Integración después de que se hallan vuelto a superar las pruebas básicas realizadas en este entorno. Una vez que se han superado todas las pruebas en el sistema de Integración, se transporta a Producción para que esté operativo de cara a la empresa.

La principal ventaja de la existencia de este esquema de sistemas es que evita interferir en el trabajo que realizan los usuarios día a día, a la vez que los desarrolladores pueden seguir modificando el sistema introduciendo nuevas mejoras. Para realizar este transporte entre sistemas, se usa lo que se denomina 'órdenes de transporte'.

En SAP, existen dos formas de guardar los cambios de desarrollo (elementos del Diccionario de Datos, código fuente, diseños, etc.):

- Local: todas las creaciones o modificaciones se guardarán en el entorno en el que nos encontremos, no afectando a otros entornos (no podrán ser transportadas a los otros sistemas).
- Órdenes de Transporte: se usan para guardar los contenidos y transportarlos entre diferentes entornos, con el fin de que el desarrollo realizado no afecte al funcionamiento del sistema.



Ilustración 88 - Creación de una orden de trabajo

En estas órdenes de trabajo se incluirán programas, funciones, tablas, estructuras del Diccionario, documentos,... en definitiva, todo aquello que interfiera en el funcionamiento del nuevo software.

Existen varias tablas de parametrización o datos que se han debido crear para dar soporte al funcionamiento de los programas; por ejemplo, en la tabla 'ZTLB_LDINTENSIDA' se han introducido los diferentes tipos de intensidad que pueden darse en un ladrillo al realizarse el ensayo de Eflorescencia. Estos datos deben incluirse en la orden de transporte ya que, de lo contrario, al ejecutar este ensayo en alguno de los otros sistemas, la tabla transportada no tendrá ningún registro y no funcionará correctamente.

5.1 Pruebas

Para comprobar el correcto funcionamiento de los programas, se han realizado diversos tipos de prueba, que se pueden categorizar de la siguiente forma:

- Pruebas de caja negra¹²: a partir de los borradores de ensayos facilitados desde laboratorio (con datos reales) y la plantilla Excel que utilizan para realizar los cálculos, se ha comprobado que, en cada ensayo y para los mismos valores de los borradores, los resultados de la plantilla Excel y el programa son las mismas. Además, a través de transacciones como la SE16N (transacción que permite consultar los registros existentes en una determinada tabla), se comprueba que los datos son guardados o cargados correctamente en el programa. También se han introducido valores incorrectos para comprobar que el programa avisa al usuario de estas incorrecciones a través de los mensajes creados.
- Pruebas de caja blanca¹³: Este tipo de pruebas se han realizado a través de la herramienta Debugger ABAP, permitiendo observar paso a paso la ejecución del programa, comprobar el contenido de las tablas o registros, cambiar el contenido de las variables para forzar ejecuciones distintas, etc.

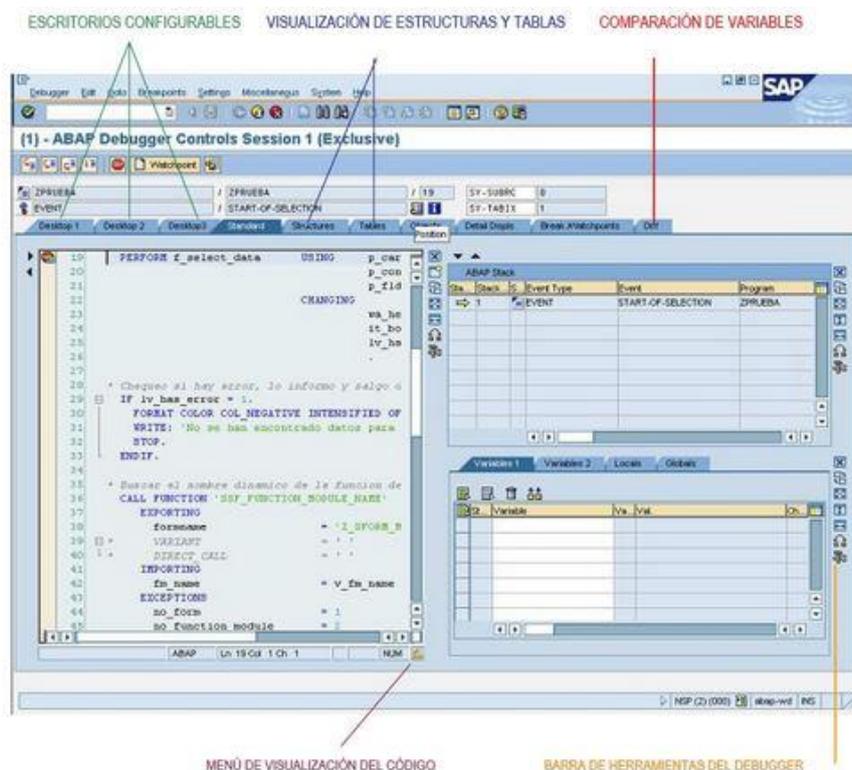


Ilustración 89 - Debugger ABAP

¹² Las pruebas de caja negra se realizan sobre la interfaz del software y consiste en proporcionar una serie de entradas para comprobar que las salidas concuerdan con las esperadas.

¹³ Las pruebas de caja blanca, por el contrario, se centran más en el software en sí. Se escogen distintos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y asegurar que se devuelven los valores de salida adecuados.

Una de las formas de ejecutar el debugger es añadir un punto de parada en una línea del código fuente y ejecutar el programa. Al llegar al punto de parada, la ejecución se pausará y la pantalla de la herramienta emergerá para que podamos realizar las acciones oportunas.

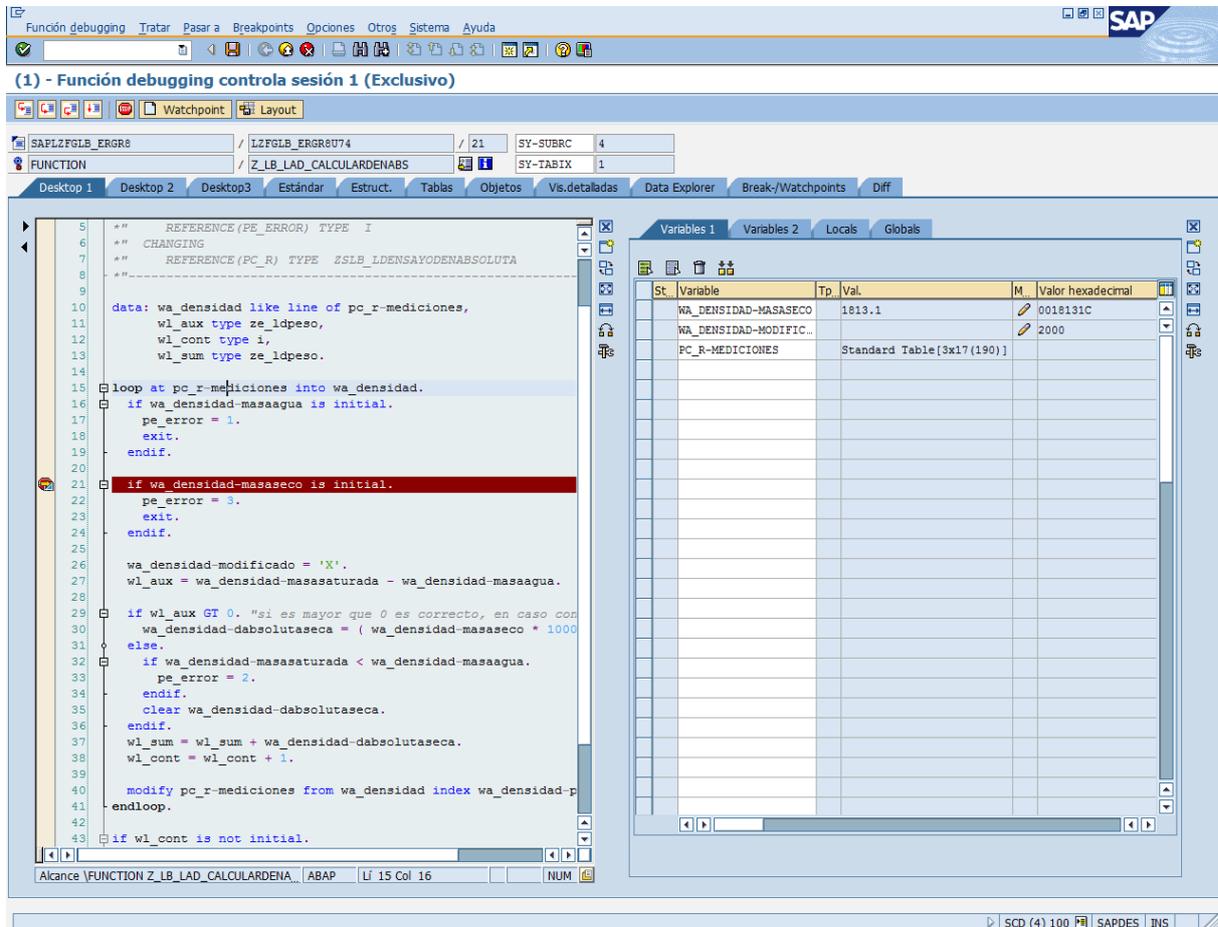


Ilustración 90 - Ejemplo de ejecución del debugger

A la derecha podemos ver una tabla donde agregar las variables que queremos comprobar (haciendo doble click sobre el nombre de la variable en el código), pudiendo añadir cualquier elemento que se utilice en el programa (tablas, variables, etc.).

Para visualizar el contenido de estos elementos, tan solo bastará con pulsar sobre ellos en la tabla y se abrirá una ventana con su contenido, tal y como se muestra a continuación:

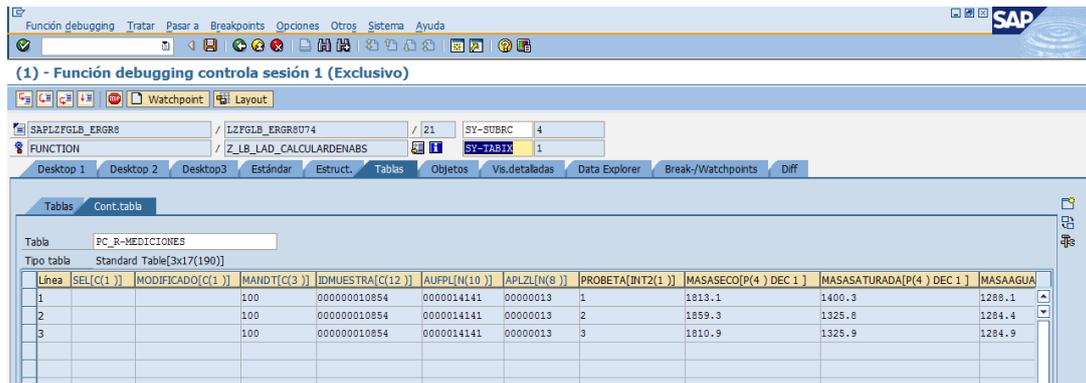


Ilustración 91 - Visualizar contenido de tabla en el debugger

La herramienta también permite cambiar el contenido de una variable o campo, de forma que se fuerce al programa a seguir un camino distinto al inicial:

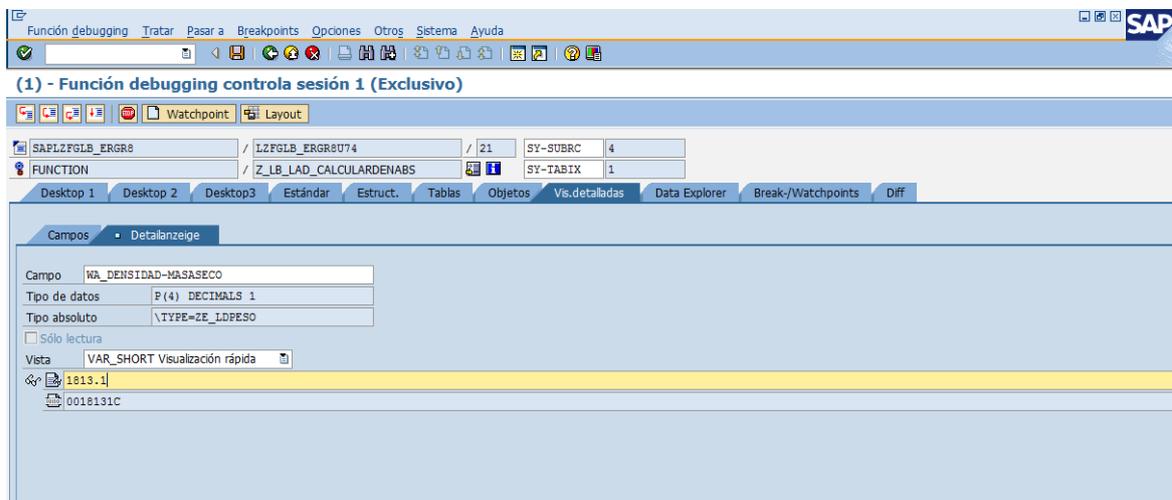


Ilustración 92 - Cambiar contenido de una variable en el debugger

5.2 Implantación

Tal y como se ha explicado anteriormente, las órdenes de transporte deben pasar de un sistema a otro para comprobar el producto y, finalmente, ser implantado.

Para ello, cuando creamos que las tareas realizadas en Desarrollo han sido finalizadas, se debe liberar la orden de transporte. Este proceso de liberar consiste básicamente en preparar la orden para transporte, bloqueándola para que no pueda seguir siendo modificada. Cualquier otra modificación que queramos realizar deberá guardarse en una orden de transporte nueva.

Este proceso de liberación se realiza desde la transacción SE10 de SAP. En ella se selecciona la orden que queremos y pulsamos sobre el botón de liberar.

El siguiente paso será entrar en la transacción STMS; desde esta transacción se elige el sistema al que queremos transportar una o varias órdenes y, tras seleccionarla, procedemos a realizar esta operación. Una vez el transporte acabe, tendremos nuestros nuevos programas en el sistema seleccionado.

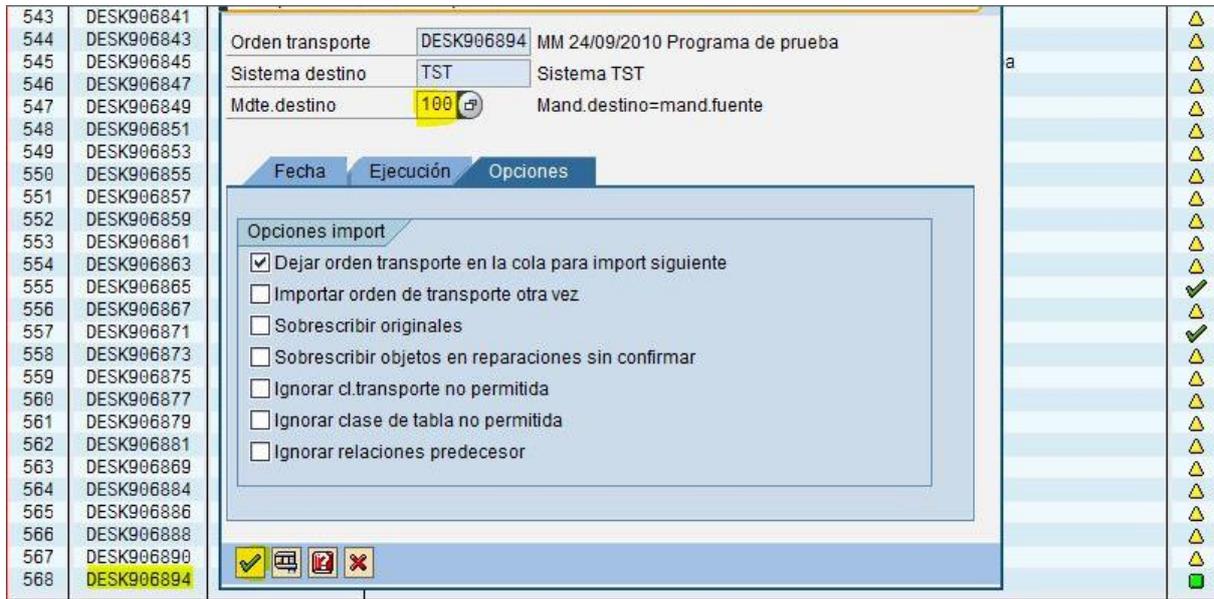


Ilustración 93 - Transport Management System

Tras realizar todas las pruebas posibles en Desarrollo, se transporta la orden a Integración, donde se harán pruebas más exhaustivas. Si en Integración se detectan fallos, deben resolverse en Desarrollo y volver a transportar; esto conllevará la creación de una serie de órdenes de transporte distintas, cada una con sus incidencias resueltas.

Es conveniente anotar y guardar todas estas órdenes por un motivo muy simple: una vez que el producto está listo para transportarse a Producción y ser usado por los usuarios, no se pueden transportar estas órdenes de cualquier forma, sino que deben ser transportadas al Sistema de Producción en el mismo orden en el que fueron transportadas a Integración. Esto se debe a que cada orden se basa en los programas o elementos guardados en la orden anterior, pero con una serie de modificaciones o elementos nuevos. Al transportar una orden que se ha creado posterior a otra aun no transportada, puede crear inconsistencias y errores, provocando la pérdida de información o la caída del Sistema; es por ello por lo que hay que tener muy presente qué órdenes se han creado para un trabajo y en qué orden han sido transportadas.

Una vez que todas las órdenes de transporte han sido transportadas a Producción, el producto estará listo para su uso. A partir de entonces, entraría la fase de mantenimiento del software, en la que, a partir de peticiones o incidencias encontradas por los usuarios, se deberán resolver comenzando el ciclo: Desarrollo, Integración, Producción.

6. Cambios

Durante el desarrollo de este trabajo, desde el departamento de Sistemas de Información surgió una cuestión que merece la pena estudiar. Por el momento, todo el desarrollo realizado está enfocado para empresas en las que se trabaje con el idioma castellano; ¿pero qué ocurriría si la empresa decidiera expandirse internacionalmente? ¿Se deberían traducir todas las tablas y/o pantallas en las que se mostrase texto o demás información relevante de cara al usuario a otro idioma? Y si continuase la expansión a otros países, ¿traducir todo lo anterior a otro idioma distinto? El volumen de información existente en el sistema sería enorme y muy ineficiente su mantenimiento. Para proporcionar solución a este problema, SAP cuenta con una herramienta denominada Tabla de Texto.

Una Tabla de Texto es una estructura de tipo tabla, que se asocia con la tabla que queremos traducir a través de su clave primaria y de un campo adicional, la clave idioma. De esta manera, en la Tabla de Texto se podrá registrar texto explicativo en varios idiomas por cada entrada de la tabla con la que se asocia. A continuación, de forma básica, se muestra un esquema de esta explicación:

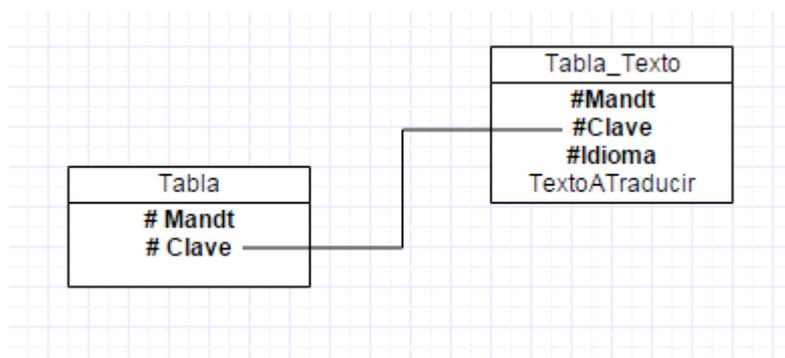


Ilustración 94 - Tablas de Texto

Existe una restricción, cada tabla de la base de datos solo podrá asociarse con una Tabla de Texto. Esto ocurre debido a que para crear esta asociación, ambas tablas se conectan a través de una clave externa; por lo que, al activar la Tabla de Texto, se verifica si alguna otra tabla ya tiene una clave externa de texto para la tabla que se especifica.

La creación de tablas de texto es muy sencilla, y no necesita de código fuente para ello; A través del Diccionario de Datos de ABAP, se realiza una copia de la tabla a la que queremos asignar una Tabla de Texto (con un nombre diferente). Tras crear la nueva tabla, se le añaden los campos de idioma y descripción, y se incluye el campo idioma como clave o parte de la clave primaria de la tabla.

Este campo idioma se creará utilizando el Elemento de Datos 'SPRAS', cuyo dominio es de tipo 'LANG'. Este tipo es el mismo que el de idioma dentro de las variables del

sistema, que se actualiza durante el inicio de sesión en SAP NetWeaver, utilizando el idioma que se selecciona en la pantalla.

The screenshot shows the SAP login interface. At the top, there are menu items: 'Usuarios', 'Sistema', and 'Ayuda'. Below the menu is a search bar and a toolbar with icons for back, save, refresh, home, and print. The main area is titled 'SAP' and contains a 'Clave acceso nueva' button. Below this, there are input fields for 'Mandante' (value: 100), 'Usuarios', 'Clv.acc.' (masked with asterisks), and 'Idioma' (value: EN).

Ilustración 95 - Inicio de sesión con el idioma Inglés

Como ejemplo, se va a mostrar la creación de la tabla de texto para las designaciones. Antes se ha eliminado el campo descripción de la tabla, ya que sería en la tabla de texto donde se debería insertar.

The screenshot shows the SAP 'Dict: Visualizar tabla' screen. The table name is 'IDDESMAI' and the description is 'Tabla con las designaciones de cada material'. The table structure is as follows:

Campo	Clv	Vali.	Elem.datos	Tipo de...	Long.	Decim.	Descripción breve	Grupo
MANDI	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDI	CLNT	3	0	Mandante	
IDDIPOATERIAL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZE_TIPO	NUMC	3	0	Tipos de material	
IDDESMAI	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZE_IDDESMAI	NUMC	4	0	Identificador para la designación de materiales	

Ilustración 96 - Tabla de designaciones

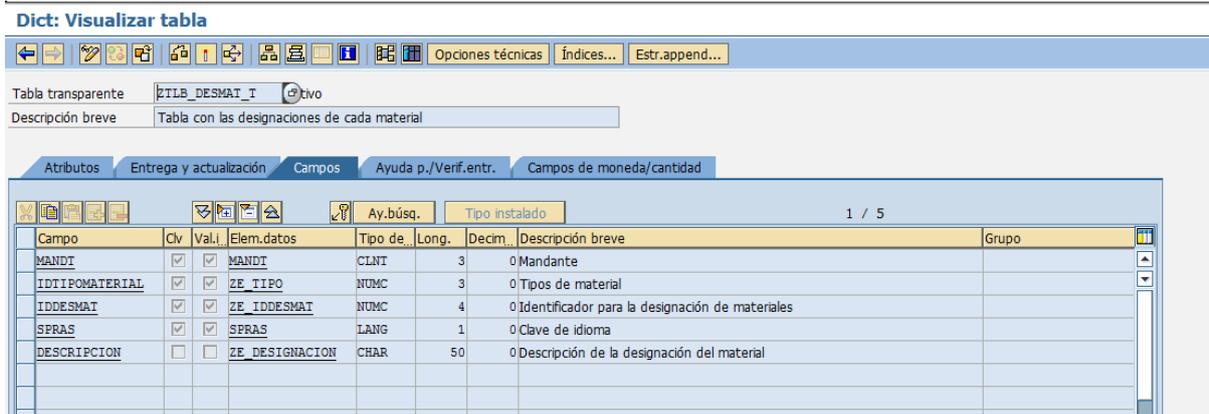


Ilustración 97 - Tabla de Texto de las designaciones

Una vez realizada esta tarea, se debe indicar a SAP que esta tabla es una Tabla de Texto de otra, por lo que se elige el campo que va a ser la referencia (en este caso, será el campo Identificador de la Designación) y se crea una clave externa que referencie al mismo campo de la tabla padre:

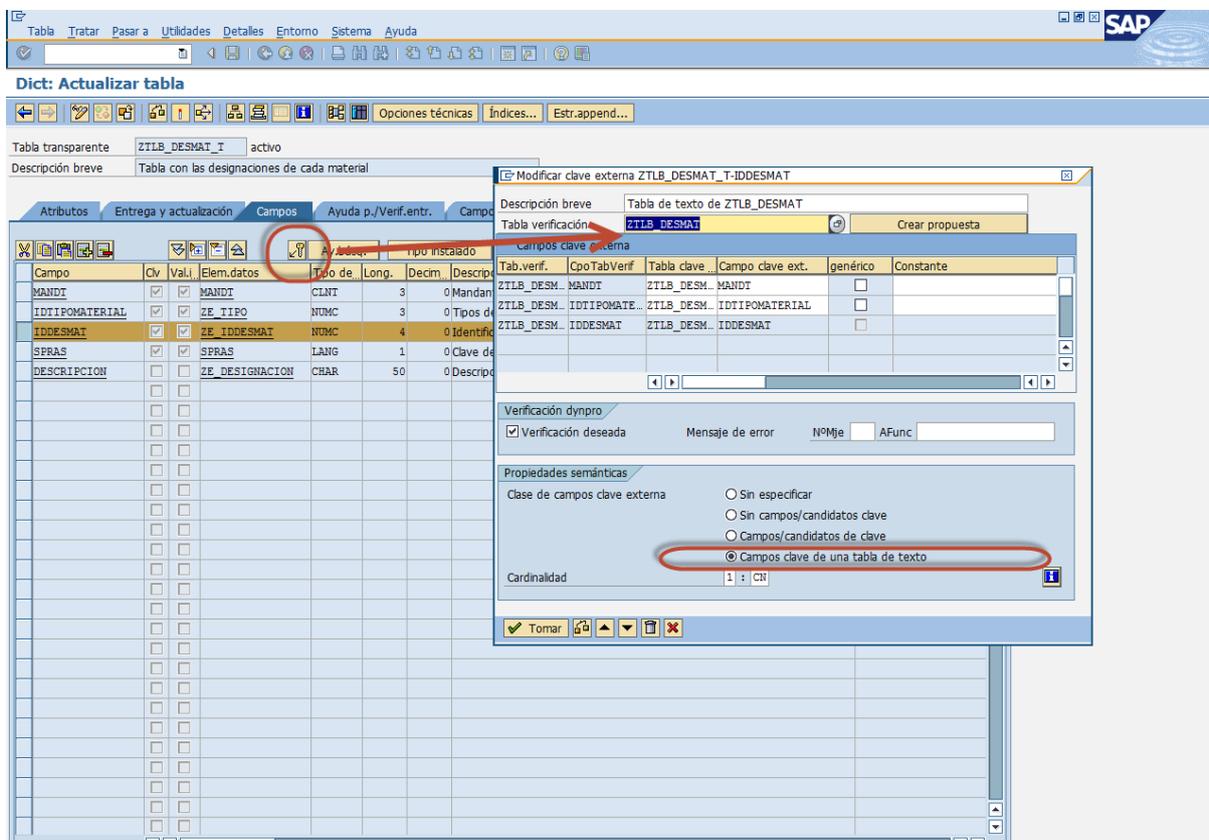


Ilustración 98 - Crear referencia Tabla de Texto

Tan solo quedará rellenar la tabla texto con los diferentes registros según su identificador e idioma, tal y como se muestra en la siguiente ilustración:

MAN...	IDTIPO MATERIAL	IDDESM...	SPR...	DESCRIPCION
100	35	1	E	Simple Hollow Bricks
100	35	2		Double Hollow Bricks
100	35	3		Triple Hollow Bricks
100	35	4		Perforated Bricks
100	35	5		Solid Bricks
100	35	1	S	Ladrillo Huecos: Sencillo
100	35	2		Ladrillo Huecos: Doble
100	35	3		Ladrillo Huecos: Triple
100	35	4		Ladrillos Perforados
100	35	5		Ladrillos Macizos

Ilustración 99 - Registros de la Tabla de Texto - Designación Materiales

Y el resultado de un programa que haga uso de este tipo de tablas será el que se muestra a continuación:

Ilustración 100 - Transacción desde sesiones e idiomas distintos

A la hora de programar, se tendrán que cambiar ligeramente las consultas ya que, si queremos obtener el texto de un registro, se deberá realizar la consulta sobre la tabla de texto, en vez de sobre la tabla padre, añadiendo una nueva restricción: el idioma

del registro debe ser igual al idioma del Sistema durante la sesión; con esta restricción, además de las que tuviéramos anteriormente, obtendremos el registro correcto.

Se ha mostrado una pequeña modificación, pero la idea se puede extrapolar a todo el Sistema de la misma forma, guardando todo los textos en tablas y creando sus Tablas de Texto; de esta forma, durante la creación de pantallas bastaría con realizar consultas a bases de datos para extraer los textos en los idiomas correspondientes y mostrarlos en la pantalla en su posición correspondiente.

7. Conclusiones

En este capítulo, el último de la memoria del Trabajo Fin de Grado, se hará un análisis del trabajo realizado, haciendo hincapié en aquellas partes que no resultaron tan satisfactorias o en las que, por un motivo u otro, han causado inconvenientes en la elaboración del trabajo. Además, se dará una valoración personal sobre lo aprendido y lo que significará este trabajo de cara al futuro.

En este sentido, hay que decir que durante la elaboración de este Trabajo Fin de Grado se han producido una serie de retrasos que han puesto en peligro la finalización del mismo dentro de los plazos marcados. Gran parte de este inconveniente tiene que ver con la inexperiencia, ya que la fase de estudio del Sistema con el que se debía trabajar llevó un tiempo que, en caso de haber trabajado antes con él, no se hubiese llevado.

Durante la toma de requisitos y diseño del modelo de datos se produjeron también una serie de inconvenientes que, de cara al futuro, sirvieron para aprender que es una fase primordial de todo desarrollo. El primer error fue utilizar únicamente las normas proporcionadas para el diseño de las pantallas de los ensayos; algunas de estas normas se encuentran desactualizadas o la normativa referente a ellas ha cambiado, por lo que algunas de las operaciones que los técnicos de laboratorio realizaban para obtener los datos de los ensayos dejaron de hacerse. Esto conllevó que una vez que el trabajo estaba en su fase final, tener que volver a diseñar varias pantallas de ensayos para poder adaptarlas al estilo de trabajo de los técnicos.

Otro de los errores fue el de estudiar cada uno de los módulos o pantallas por separado, sin englobar una visión general. Esto provocó que el primer modelo de datos obtenido fuese ineficiente, con demasiadas tablas e, incluso, algunas con los mismos campos y, por lo tanto, iguales. Esto provocó tener que empezar esta fase desde el principio de nuevo, con el retraso que conllevó.

El resto de inconvenientes vinieron por parte de la adaptación a la metodología de programación existente e inexperiencia, además de la inseguridad del programador que debe trabajar y hacer modificaciones en partes del sistema que son vitales para su correcto funcionamiento y que se comparten entre distintos módulos o programas, por lo que cada cambio debía ser estudiado y probado minuciosamente.

Hay que destacar que en todo momento se contó con el apoyo del equipo de desarrolladores que, aunque no intervinieron en el proceso, sus consejos, advertencias e ideas fueron útiles a la hora de encontrar una solución a todos los inconvenientes encontrados.

Otro inconveniente encontrado, pero de naturaleza menos técnica, fue la mala administración del tiempo a la hora de realizar esta memoria. Esto provocó que su

finalización se produjese muy cerca de la fecha objetivo, con la presión e inconvenientes que pueden sucederse en dichas circunstancias.

Tras la finalización de este Trabajo Fin de Grado, se puede decir que el nivel de satisfacción personal y profesional es muy alto, y que el aprendizaje adquirido durante su elaboración será de gran ayuda frente a proyectos futuros y, el poder enfrentarme a los problemas anteriormente citados, conllevará que no cometa los mismo errores y que pueda prever dichas situaciones.

Bibliografía

- ABAP IV; <http://www.abap.es/>
 - SAP NetWeaver; <http://scn.sap.com/community/netweaver>
 - ModulPool; <http://www.abap.es/Descargas/Manual%20Modul-Pool%20.pdf>
 - Tablas de texto; <http://solutionssap.com/como-asociar-una-tabla-de-texto-a-una-tabla-z/>
 - Normas UNE; <http://www.aenor.es/aenor/inicio/home/home.asp>
-
- Seidor (proporcionados por la empresa); **Manuales de programación e introducción a SAP**
 - Horst Keller, Sascha Krüger (2007); **ABAP Objects: ABAP Programming in SAP NetWeaver**; Galileo Press
 - Karl-Heinz Kühnhauser (2008); **Discover ABAP**; Galileo Press
 - Jesús Gómez Tenedor (2011); **Desarrollo en SAP de aplicaciones para laboratorio de control de calidad en materiales**; Proyecto fin de carrera, Universidad de Málaga
 - AENOR (proporcionadas por la empresa); **Normas UNE**

Tabla de Ilustraciones

Ilustración 1 - Elementos de un ERP	9
Ilustración 2 - Normas UNE	10
Ilustración 3 - Entrada de Muestras previa	11
Ilustración 4 - Plantilla e Informe originales	12
Ilustración 5 - Flujo del Módulo de Laboratorio	12
Ilustración 6 - Logo de SAP	14
Ilustración 7 - Plataforma SAP NetWeaver	15
Ilustración 8 - SAP R/3	16
Ilustración 9 - Módulos SAP R/3	17
Ilustración 10 - Entorno de programación ABAP	20
Ilustración 11 - Diccionario ABAP	22
Ilustración 12 - Foro de dudas	27
Ilustración 13 - Diagrama de Casos de Uso	30
Ilustración 14 - Diagrama de Secuencia CU1	31
Ilustración 15 - Diagrama de Secuencia CU2	32
Ilustración 16 - Diagrama de Secuencia CU3	33
Ilustración 17 - Diagrama de Secuencia CU4	34
Ilustración 18 - Diagrama de Secuencia CU5	35
Ilustración 19 - Nueva Entrada de Muestras	36
Ilustración 20 - Entrada de Resultados General	37
Ilustración 21 - Aristas de un Ladrillo	38
Ilustración 22 - E.R. del ensayo Dimensión y Comprobación de Forma	39
Ilustración 23 - E.R. Absorción de agua	40
Ilustración 24 - E.R. Eflorescencia	41
Ilustración 25 - E.R. Exp. por Humedad	41
Ilustración 26 - E.R. Heladicidad	42
Ilustración 27 - E.R. Compresión	42
Ilustración 28 - E.R. Succión	43
Ilustración 29 - E.R. Flexión	44
Ilustración 30 - E.R. Inclusiones calcáreas	44
Ilustración 31 - Entrada E.R. Aspectos y Estructura	45
Ilustración 32 - Entrada E.R. Masa	45
Ilustración 33 - E.R. Dimensiones: Longitud, Altura y Anchura	46
Ilustración 34 - E.R. Densidad absoluta	46
Ilustración 35 - E.R. Densidad aparente	47
Ilustración 36 - Porcentaje Huecos	47
Ilustración 37 - Planeidad de las caras	48
Ilustración 38 - Diseño del Informe de Ladrillos	49
Ilustración 39 - Modelo del Dominio	50
Ilustración 40 - Modelo de Datos: Entrada de Muestras	51
Ilustración 41 - Modelo de datos: E.R-Defectos	52
Ilustración 42 - Campos auditoría	52
Ilustración 43 - Modelo de Datos: ER-Heladicidad	53

Ilustración 44 - Modelo de Datos: ER-Inclusiones calcáreas.....	53
Ilustración 45 - Modelo de Datos: ER-Dimensión y forma.....	54
Ilustración 46 - Modelo de Datos: ER-Long., Altura y Anchura.....	55
Ilustración 47 - Modelo de Datos: ER-Planeidad	55
Ilustración 48 - Modelo de Datos: Ámbito de valores de un Dominio.....	56
Ilustración 49 - Modelo de Datos: ER-Exp por Humedad.....	56
Ilustración 50 - Modelo de Datos: ER-Masa	57
Ilustración 51 - Modelo de Datos: ER-Eflorescencia	57
Ilustración 52 - Modelo de Datos: ER-Flexión.....	58
Ilustración 53 - Modelo de Datos: ER-Huecos	58
Ilustración 54 - Modelo de Datos: ER-Absorción agua	59
Ilustración 55 - Modelo de Datos: ER-Densidad absoluta.....	59
Ilustración 56 - Modelo de Datos: ER-Densidad Aparente	60
Ilustración 57 - Modelo de Datos: ER-Compresión.....	60
Ilustración 58 - Modelo de Datos: ER-Succión.....	61
Ilustración 59 - Objetos DynPro.....	63
Ilustración 60 - Programa Principal - Sentencias Include.....	64
Ilustración 61 - Screen Painter	65
Ilustración 62 - A. Búsqueda Designación Material	66
Ilustración 63 - Ay. Búsqueda Designación Material	67
Ilustración 64 - Ejemplo Ayuda de Búsqueda	67
Ilustración 65 - DynPro-PBO	68
Ilustración 66 - Cargar designación	69
Ilustración 67 - Chain..Endchain	70
Ilustración 68 - Ejemplo DynPro E.R.	70
Ilustración 69 - Ejemplo código DynPro.....	71
Ilustración 70 - PBO Entrada Resultados	72
Ilustración 71 - Inicializar datos ensayo.....	72
Ilustración 72 - Mensajes ABAP	73
Ilustración 73 - Módulo de Función y comprobación de errores.....	74
Ilustración 74 - Módulos PBO	75
Ilustración 75 - Guardar datos ensayo.....	76
Ilustración 76 - Creación de plantilla	77
Ilustración 77 - Selección de Plantilla	77
Ilustración 78 - Crear campo plantilla.....	78
Ilustración 79 - Insertar campo	79
Ilustración 80 - FM plantilla ladrillo.....	80
Ilustración 81 - Datos para plantilla.....	81
Ilustración 82 - Estructura para tablas.....	82
Ilustración 83 - Cargar campos del informe.....	83
Ilustración 84 - Carga de tablas para informe.....	84
Ilustración 85 - Plantilla y fórmulas	85
Ilustración 86 - Obtención de un campo	86
Ilustración 87 - Obtención de una tabla	86
Ilustración 88 - Creación de una orden de trabajo	88

Ilustración 89 - Debugger ABAP.....	89
Ilustración 90 - Ejemplo de ejecución del debugger.....	90
Ilustración 91 - Visualizar contenido de tabla en el debugger.....	91
Ilustración 92 - Cambiar contenido de una variable en el debugger.....	91
Ilustración 93 - Transport Management System.....	92
Ilustración 94 - Tablas de Texto.....	93
Ilustración 95 - Inicio de sesión con el idioma Inglés.....	94
Ilustración 96 - Tabla de designaciones.....	94
Ilustración 97 - Tabla de Texto de las designaciones.....	95
Ilustración 98 - Crear referencia Tabla de Texto.....	95
Ilustración 99 - Registros de la Tabla de Texto - Designación Materiales.....	96
Ilustración 100 - Transacción desde sesiones e idiomas distintos.....	96