



UNIVERSIDAD  
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA  
GRADUADO EN INGENIERÍA INFORMÁTICA

**ENSAYOS NO DESTRUCTIVOS EN SOLDADURAS Y  
MATERIALES METÁLICOS. INTEGRACIÓN DE LA  
INFORMACIÓN EN EL FLUJO DE PROCESOS BIM EN EL  
ÁMBITO DE CONTROL DE CALIDAD**

**NON-DESTRUCTIVE TEST ON WELD AND METALLIC  
MATERIALS. INTEGRATION OF INFORMATION IN THE BIM  
PROCESS FLOW IN THE FIELD OF QUALITY OF CONTROL**

Realizado por  
**TARUN DEVNANI DEVNANI**

Tutorizado por  
**ANTONIO MUÑOZ GALLEGO**  
**FRANCISCO JOSÉ JAIME RODRÍGUEZ**

Departamento  
**LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN**

UNIVERSIDAD DE MÁLAGA  
MÁLAGA, 2021



UNIVERSIDAD  
DE MÁLAGA



# Resumen

SAP o como sus siglas significan, Sistemas, Aplicaciones y Productos para Procesamiento de Datos, es un sistema informático utilizado en las empresas para administrar correctamente las acciones de la empresa.

Este sistema se relaciona con otros que nos sirven para la planificación de los recursos empresariales, a estos les llamamos ERP, que son sistemas íntegros compuestos por diferentes módulos para administrar los recursos de cada área en la empresa.

El lenguaje de programación de SAP es ABAP, Advanced Business Application Programming, y se encarga de desarrollar las nuevas y/o adecuar las funcionalidades en SAP para que cumplan los requerimientos de la empresa.

Lo que se va a tratar en este trabajo será la realización de una serie de herramientas para el laboratorio de CEMOSA (Centro de Estudios de Materiales y Control de Obra SA), siguiendo las normas UNE específicas para el material. Estas herramientas van a consistir en la posibilidad de tomar las muestras del tipo de material metálico, visualizarlo en la entrada de muestras y la generación de informes y albaranes referentes a dicho material.

**Palabras clave:**

SAP, ABAP, ERP, Servicio, Materiales Metálicos



# Abstract

SAP or as its acronym means, Systems, Applications and Products for Data Processing, is a computer system used in companies to correctly manage the actions of the company.

This system is related to others that serve us for the planning of the business resources, we call these ERP, which are integral systems composed of different modules to manage the resources of each area in the company.

The programming language of SAP is ABAP, Advanced Business Application Programming, and it is in charge of developing the new ones and/or adapting the functionalities in SAP so they can achieve the requirements of the company.

This work is about the realization of a series of tools for CEMOSA's (Center of the Studies of Materials and Work Control SA) laboratory, following the UNE standards specified for the material. These tools will consist on the possibility of taking samples of the type of metallic material, displaying it in the sample entry and generating reports and delivery notes relating to the material.

**Keywords:**

SAP, ABAP, ERP, Service, Metallic Materials



# Índice

<b>Resumen .....</b>	<b>1</b>
<b>Abstract.....</b>	<b>1</b>
<b>Índice .....</b>	<b>1</b>
<b>Índice de Ilustraciones .....</b>	<b>2</b>
<b>Introducción .....</b>	<b>5</b>
<b>1.1 Motivación .....</b>	<b>5</b>
<b>1.2 Objetivos.....</b>	<b>6</b>
<b>1.3 Tecnologías a usar .....</b>	<b>6</b>
<b>1.4 Metodología.....</b>	<b>7</b>
<b>1.5 Estructura de la memoria .....</b>	<b>9</b>
<b>Análisis y Requisitos .....</b>	<b>11</b>
<b>2.1 Análisis.....</b>	<b>11</b>
<b>2.2 Requisitos .....</b>	<b>13</b>
2.2.1 Requisitos Funcionales .....	13
2.2.2 Requisitos No Funcionales .....	15
2.2.3 Requisitos de Información .....	16
2.2.4 Casos de uso .....	17
<b>2.3 Diseño.....</b>	<b>18</b>
2.3.1 Modelo Conceptual .....	20
2.3.2 Diagrama de secuencia .....	21
2.3.3 Diagrama de clases.....	23
<b>2.4 Entrada de Datos del Servicio .....</b>	<b>25</b>
<b>Desarrollo.....</b>	<b>29</b>
<b>3.1 Creación de las Interfaces .....</b>	<b>29</b>
<b>3.2 Implementación de Código .....</b>	<b>37</b>
<b>Pruebas .....</b>	<b>71</b>
<b>Futuras Mejoras y Posibles Ampliaciones .....</b>	<b>79</b>
<b>Conclusiones.....</b>	<b>81</b>
<b>Referencias.....</b>	<b>83</b>
<b>Manual de Usuario .....</b>	<b>85</b>
<b>Acceso a SAP-Fiori:.....</b>	<b>85</b>
<b>Acceso en SAP.....</b>	<b>90</b>

# Índice de Ilustraciones

Ilustración 1: Metodología ágil.....	7
Ilustración 2: Normas UNE-EN ISO .....	12
Ilustración 3: Ensayo de Líquidos Penetrantes .....	12
Ilustración 4: Diagrama de Casos de Uso .....	18
Ilustración 5: Modelo conceptual .....	20
Ilustración 6: Flujo de datos 1 .....	22
Ilustración 7: Flujo de datos 2 .....	23
Ilustración 8: Diagrama de clases 1.....	24
Ilustración 9: Diagrama de clases 2.....	25
Ilustración 10: Objetos parciales de una dynpro .....	25
Ilustración 11: Diseño pantalla del ensayo .....	26
Ilustración 12: Entrada de Resultado del ensayo.....	26
Ilustración 13: Visualización del ensayo en Frontend.....	27
Ilustración 14: Transacción <i>SE11</i> .....	30
Ilustración 15: Crear un Dominio .....	30
Ilustración 16: Crear un Elemento de Dato.....	31
Ilustración 17: Crear una tabla .....	32
Ilustración 18: Clase de entrega de la tabla .....	32
Ilustración 19: Tabla <i>ZTLB_METAL</i> .....	33
Ilustración 20: Estructura <i>ZSLB_METAL</i> .....	33
Ilustración 21: Tipo tabla <i>ZSTLB_METAL</i> .....	34
Ilustración 22: Tabla <i>ZTLB_ELEMMETAL</i> .....	34
Ilustración 23: Estructura <i>ZSLB_ELEMMETAL</i> .....	35
Ilustración 24: Tipo tabla <i>ZSTLB_ELEMMETAL</i> .....	35
Ilustración 25: Proyecto <i>ZGW_MATMETAL</i> .....	36
Ilustración 26: Entidad de DatosEnsayo.....	36
Ilustración 27: Entidad de DatosElementos .....	37
Ilustración 28: Configuración del ensayo .....	37
Ilustración 29: Ensayo configurado .....	38
Ilustración 30: PBO .....	39
Ilustración 31: Inicializar datos 9012.....	39
Ilustración 32: Inicializar datos del ensayo .....	40
Ilustración 33: Función <i>Z_LB_CARGAR_METAL_ELEM</i> .....	41
Ilustración 34: Carga los datos de la tabla <i>ZTLB_METAL</i> .....	41
Ilustración 35: Carga los datos de la tabla <i>ZTLB_ELEMMETAL</i> .....	42
Ilustración 36: PAI .....	43
Ilustración 37: Función guardar datos 9012 .....	44
Ilustración 38: Guardar datos del ensayo .....	44
Ilustración 39: Función de guardar <i>Z_LB_GUARDAR_METAL_ELEM</i> .....	45
Ilustración 40: Guardamos los datos de la tabla <i>ZTLB_METAL</i> .....	45
Ilustración 41: Guardamos los datos en la tabla <i>ZTLB_ELEMMETAL</i> .....	46



Ilustración 42: Añadir más filas .....	47
Ilustración 43: Perform para añadir más filas .....	47
Ilustración 44: Métodos redefinidos del proyecto.....	48
Ilustración 45: GET de DatosEnsayo.....	49
Ilustración 46: Recoger datos por clave o por filtros .....	49
Ilustración 47: <i>GET_ENTITY</i> de DatosEnsayo .....	50
Ilustración 48: <i>GET_ENTITYSET</i> de DatosElementos.....	50
Ilustración 49: <i>GET_ENTITY</i> de DatosElementos.....	51
Ilustración 50: <i>CREATE</i> de DatosElementos .....	51
Ilustración 51: Función para crear filas en la tabla .....	52
Ilustración 52: <i>DELETE</i> de DatosElementos .....	53
Ilustración 53: Función para eliminar filas en la tabla .....	54
Ilustración 54: <i>UPDATE</i> de DatosEnsayo.....	55
Ilustración 55: Guarda datos en la tabla <i>ZTLB_METAL</i> .....	55
Ilustración 56: <i>UPDATE</i> de DatosElementos .....	56
Ilustración 57: Guarda datos en la tabla <i>ZTLB_ELEMMETAL</i> .....	57
Ilustración 58: Botón navega Entrada de Resultados .....	58
Ilustración 59: Función para navegar a la Entrada de Resultados .....	58
Ilustración 60: Proyecto para Fiori .....	59
Ilustración 61: Código para la vista de la pantalla .....	60
Ilustración 62: Función para cargar los datos en la pantalla .....	61
Ilustración 63: Función para añadir filas .....	62
Ilustración 64: Función para eliminar filas .....	63
Ilustración 65: Función para guardar .....	64
Ilustración 66: Transacción para crear plantilla Adobe .....	65
Ilustración 67: Interfaz para la plantilla Adobe.....	65
Ilustración 68: Formulario de la plantilla Adobe.....	66
Ilustración 69: Contexto de la plantilla Adobe.....	66
Ilustración 70: Vista de la plantilla Adobe.....	67
Ilustración 71: Función para la plantilla .....	68
Ilustración 72: Elección de la plantilla Adobe .....	69
Ilustración 73: Registros añadidos en las tablas para usar la plantilla Adobe .....	69
Ilustración 74: Transacción para la Entrada de Resultados .....	72
Ilustración 75: Vista de datos de la Entrada de Resultados.....	72
Ilustración 76: Mensaje de guardado.....	73
Ilustración 77: Mensaje de error.....	74
Ilustración 78: Transacción para usar el proyecto para Frontend .....	75
Ilustración 79: Pantalla de pruebas para el proyecto .....	75
Ilustración 80: Mensaje de confirmación para la carga .....	76
Ilustración 81: Mensaje de confirmación para la carga especificando .....	76
Ilustración 82: Crear filas de la tabla.....	77
Ilustración 83: Eliminar filas de la tabla .....	77
Ilustración 84: Guardar datos modificando en la tabla .....	78
Ilustración 85: Confirmando la modificación. ....	78
Ilustración 86: Login SAP-Fiori.....	85
Ilustración 87: Aplicaciones disponibles para el usuario .....	86
Ilustración 88: Expediente de prueba .....	86

Ilustración 89: Elección del Tipo de Material .....	87
Ilustración 90: Elección del Tipo de Ensayo .....	87
Ilustración 91: Elección de las unidades a tomar .....	88
Ilustración 92: Entrada de la muestra en Fiori .....	88
Ilustración 93: Mensaje de guardado de la muestra .....	89
Ilustración 94: Botón para navegar a la Entrada de Resultados .....	89
Ilustración 95: Pantalla de la Entrada de Resultados.....	90
Ilustración 96: Mensaje de guardado en la Entrada de Resultados .....	90
Ilustración 97: Transacción para hacer la Entrada de la Muestra.....	91
Ilustración 98: Completar campos obligatorios para ir a la Entrada de Resultados.....	91
Ilustración 99: Visualización de los datos añadidos en Fiori.....	92
Ilustración 100: Transacción para visualizar el informe Adobe .....	93
Ilustración 101: Botón para visualizar el informe Adobe.....	93
Ilustración 102: Elección del libro .....	94
Ilustración 103: Modo de visualización.....	95
Ilustración 104: Visualización del informe Adobe.....	96

# 1

## Introducción

### 1.1 Motivación

Esta y las sucesivas páginas que encontrará el lector cuentan no solo un trabajo de fin de grado, cuentan algo más, y es el esfuerzo que hay detrás de toda obra o producto.

Cuando la mayoría de las personas tienen delante de sí una creación que no es de su campo de conocimiento, no son capaces de imaginar lo arduo que puede ser el desarrollo completo desde una idea hasta el producto entregado al cliente. Pues bien, aquí puede hablar alguien que ha tenido la maravillosa oportunidad de estar a ambos lados, y solo se puede concluir que no se volverá a caer en la tentación de pensar “qué cosa tan simple y fácil de hacer”, sino se pensará “qué cosa tan compleja e interesante de diseñar”.

El trabajo que se realiza es una oportunidad concedida por la empresa Centro de Estudios de Materiales y Control de Obra SA (En lo referente en adelante, CEMOSA). El área de desarrollo de la actividad es en investigación, desarrollo e innovación (I+D+I), en el área de desarrollo de software de la empresa. La actividad realizada es un ensayo de pruebas de laboratorio, especificando un poco más en este amplio abanico de variedades, concretamente el ensayo de Soldaduras y Materiales Metálicos. Este ensayo elegido tenía las carencias del siglo pasado, o dicho de otra forma, no estaba preparado para este siglo. El operario que realiza las muestras anota y posteriormente transcribe lo anotado al ordenador, o tiene que volver a la

instalación para llevar a cabo la transcripción. El problema que se puede observar es que el usuario puede perder las muestras anotadas, o transcribirlas mal. Por no añadir problemas de tiempo en la agenda si se tienen que tomar diversas muestras, en distintos puntos de la zona.

En resumen, una empresa puntera de ingeniería y control de calidad no puede ser competente si no tiene actualizado puntos tan críticos como son las tomas de muestras para su posterior análisis y por ello se desarrolló una idea, que finalmente acabó siendo un producto.

## **1.2 Objetivos**

El objetivo u objetivos que se intentan conseguir es desarrollar una aplicación que resulte útil para la empresa. Que aporte los beneficios derivados de la automatización de los procesos y suprima los costes o perjuicios de la realización de manera tradicional de la toma de muestras.

Con esto se persigue entre otros objetivos la reducción del coste derivado de tener al personal visitando varias veces la misma obra por motivos erróneos en la toma de muestras, y contrariamente a esta pérdida, el beneficio de que el operario pueda realizar visitas a otras obras si fuera necesario. La sincronización y centralización de los datos evitará inconsistencia en los mismos si alguien realiza una modificación en los datos y posteriormente hay distintas versiones y no se sabe cuál es la correcta.

## **1.3 Tecnologías a usar**

CEMOSA está llevando a cabo un proceso de transformación digital para la mejoras y optimización de sus recursos en cuanto al proceso del control de calidad de los materiales.

Para ello, ha sido de gran utilidad contar con un ERP como SAP. Concretamente, se usará ABAP para la programación en Backend y SAPUI5 para la programación en Frontend o Fiori.

ABAP (Advanced Business Application Programming) es un lenguaje propio de SAP, que se utilizará para programar y que tiene similitud con distintos tipos de lenguajes de programación, como, por ejemplo, C o los conceptos definidos por la Programación Orientada a Objetos. Es un lenguaje utilizado para crear nuevos programas que permiten

crear transacciones que no existen en SAP. Además, se pueden añadir nuevas funcionalidades por si las predefinidas por SAP no son suficientes para la realización de algún proyecto.

SAPUI5 es una herramienta de desarrollo para las aplicaciones basadas en HTML5, las cuales se pueden usar en cualquier dispositivo. Además, proporcionan un modelo de programación ligero para las aplicaciones móviles y, también, de escritorio, usando el patrón de diseño MVC (Modelo - Vista - Controlador).

### 1.4 Metodología

La metodología que se va a emplear será la metodología ágil (Ilustración 1), donde el proceso de trabajo vendrá marcado por las características que destacan en esta metodología.



Ilustración 1: Metodología ágil

En CEMOSA los proyectos los van definiendo las necesidades que transmiten el resto de los departamentos. Lo primero que se establece es la planificación estimada y el personal asignado. Luego, se empieza con las distintas fases del desarrollo de Software.

**1. Definición de los Requisitos:** al principio de cada proyecto donde se emplea la metodología ágil, hay que realizar el proceso de la recogida de requisitos y necesidades que requiere el proyecto. Se establecen varias reuniones con las partes involucradas para una mayor recogida de datos.

**2. Análisis de los Requisitos:** una vez estudiado y definido posibles soluciones se vuelven a realizar más reuniones para plantear las soluciones propuestas a la espera de su validación.

**3. Diseño:** ya definida la solución que se va a realizar, comienza con el modelado de esta mediante distintos diagramas UML como, por ejemplo, Diagrama de Casos de Uso, Diagrama de Secuencia o Diagrama de Clases.

**4. Desarrollo:** en esta fase se hará la configuración de las tablas y las estructuras que se van a utilizar para posteriormente realizar el código en ABAP y SAPUI5 para la ejecución del programa.

**5. Implementación:** una vez llegada a esta fase ya disponemos de las tablas y las estructuras que necesitamos para realizar todo el proceso. Por ello, comenzamos con la programación tanto para Backend como para Frontend.

**6. Pruebas:** terminado todo el proceso de implementación, procederemos a realizar las distintas pruebas pertinentes para comprobar el correcto funcionamiento del programa. Si se detecta alguno se vuelve a la fase de implementación para corregir el error y luego se vuelven a hacer las pruebas.

**7. Mantenimiento:** esta fase se incluye el soporte a los usuarios que revisan el proyecto, puesto que sus valoraciones, comentarios y sugerencias es lo que nos ayuda a detectar fallos e incluso a realizar ampliaciones o mejoras para su facilidad.

## 1.5 Estructura de la memoria

La memoria está estructura de la siguiente manera:

1. **Introducción:** se establece la motivación y los objetivos a seguir para la realización de este TFG. Además, se incluye una descripción de la metodología que se ha utilizado durante el proceso de desarrollo del presente proyecto y las tecnologías usadas.

2. **Fases del Desarrollo:** incluimos aquí todos los análisis llevados a cabo y los requisitos que se van a establecer para la realización del trabajo. Así como, diagramas y casos de uso para un mayor conocimiento del funcionamiento del sistema. También se mostrará el diseño que se va a realizar, describiendo con detalle todo el proceso del desarrollo e implementación que se involucran en este proceso.

3. **Pruebas:** se muestran los tipos de pruebas que se realizan en cada fase del desarrollo e implementación.

4. **Mejoras y ampliación:** se describirán algunas mejoras que se pueden llevar a cabo para las futuras versiones. Incluimos una serie de ampliaciones que se pueden realizar para dar facilidad a los trabajadores que utilicen nuestra aplicación.

5. **Conclusiones:** describimos aquí si se han alcanzado los objetivos o no. Así como, las ventajas y mejoras que supone esta ampliación a la empresa.

6. **Referencias:** se incluye un listado con las fuentes consultadas para la realización de este TFG.





# 2

## Análisis y Requisitos

### 2.1 Análisis

Como en todo proyecto, antes de empezar a trabajar debemos conocer qué vamos a desarrollar, así que antes de empezar, concretamos una pequeña reunión donde se explicaron qué tenemos que realizar para ayudar a la empresa a mejorar su proceso en el control de calidad en las obras.

Para ello nos facilitaron documentos que se utilizan a mano y las normas españolas UNE (Ilustración 2) con las que trabajan día a día para que tengamos claro qué estructura y qué reglamento seguir para facilitarles el trabajo a los trabajadores que están en la obra. Las normas facilitadas fueron:

- UNE-EN ISO 3059-2013
- UNE-EN ISO 3452-3 1999
- UNE-EN ISO 3452-4 1999
- UNE-EN ISO 5817-2014
- UNE-EN ISO 9712-2012
- UNE-EN ISO 17365-2010
- UNE-EN ISO 23277-2015
- UNE-EN ISO 3452-2 2014.

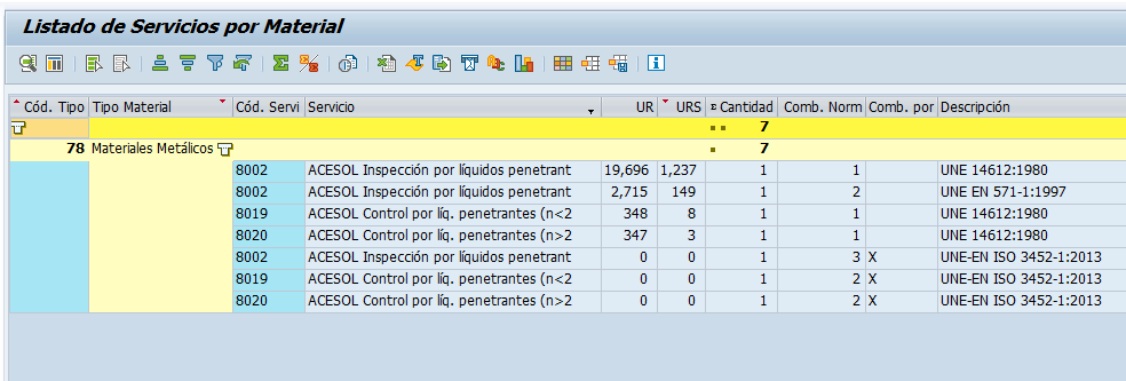
# norma española

UNE-EN ISO 3059

Febrero 2013

Ilustración 2: Normas UNE-EN ISO

Una vez que tuvimos los documentos supimos sobre qué tipo de material y que tipo de ensayo trabajaríamos. Este TFG se realizará sobre el tipo de material Materiales Metálicos que contiene el tipo de ensayo de Líquidos Penetrantes (Ilustración 3). Vemos aquí el tipo de material Materiales Metálicos que tiene como identificador el número 78. Filtramos luego por los servicios que contienen Líquidos Penetrantes y vemos el número de ensayos que se han llegado a realizar desde que existe dicho ensayo. Escogemos por ello el servicio 8002 para realizar el proyecto, puesto que es el que más se utiliza y más rápido será el proceso de la toma de datos en la obra.



Cód. Tipo	Tipo Material	Cód. Servi	Servicio	UR	URS	Cantidad	Comb. Norm	Comb. por	Descripción
78	Materiales Metálicos					7			
		8002	ACESOL Inspección por líquidos penetrant	19,696	1,237	1	1		UNE 14612:1980
		8002	ACESOL Inspección por líquidos penetrant	2,715	149	1	2		UNE EN 571-1:1997
		8019	ACESOL Control por líq. penetrantes (n<2	348	8	1	1		UNE 14612:1980
		8020	ACESOL Control por líq. penetrantes (n>2	347	3	1	1		UNE 14612:1980
		8002	ACESOL Inspección por líquidos penetrant	0	0	1	3	X	UNE-EN ISO 3452-1:2013
		8019	ACESOL Control por líq. penetrantes (n<2	0	0	1	2	X	UNE-EN ISO 3452-1:2013
		8020	ACESOL Control por líq. penetrantes (n>2	0	0	1	2	X	UNE-EN ISO 3452-1:2013

Ilustración 3: Ensayo de Líquidos Penetrantes

## 2.2 Requisitos

### 2.2.1 Requisitos Funcionales

A continuación, se enumeran los requisitos funcionales del software, que son aquellos servicios que deben cumplirse. Para ello intentamos definirlo de la manera más específica posible, procurando evitar ambigüedades en su descripción sin restar claridad.

#### 2.2.1.1 Requisitos Funcionales de la muestra

RF – Muestra		
Código	Título	Descripción
RF – CEMOSA - mu01	Datos de la muestra	<p>Se añaden los siguientes campos a la pantalla general de la entrada de muestras:</p> <ul style="list-style-type: none"><li>- Tiempo de Penetración</li><li>- Temperatura Ambiente Encima</li><li>- Temperatura Ambiente Debajo</li></ul> <p>Se controlará que los datos introducidos del usuario sean correctos</p>
RF – CEMOSA – mu02	CRUD Servicios	<p>Crear, Leer, Actualizar y Borrar servicios para la muestra.</p> <p>Podrá asignar el servicio a la muestra para luego rellenar los datos que procedan.</p>
RF – CEMOSA – mu03	Albarán	<p>Visualización del albarán por el cual mostrar la muestra.</p>

		Se incrusta un botón ya implementado para poder abrir este albarán.
--	--	---

### 2.2.1.2 Requisitos Funcionales del ensayo

RF – Ensayo		
Código	Titulo	Descripción
RF – CEMOSA - en01	Datos de ensayo	Añadir los resultados obtenidos por los técnicos. Asegurando que los datos sean correctos.
RF – CEMOSA – en02	Administrar datos	Guardar los datos. Asegurando que se guarden los datos en sus respectivas tablas.
RF – CEMOSA – en03	Informe	Se genera el informe con los resultados obtenidos en la entrada de resultados

### 2.2.1.3 Requisitos Funcionales del albarán

RF – Albarán		
Código	Titulo	Descripción
RF – CEMOSA – in01	Crear Albarán	Se genera el albarán para el tipo del material.  Un botón incrustado que se reutiliza para poder generar el albarán.

### 2.2.1.4 Requisitos Funcionales de Fiori

RF – Fiori		
Código	Titulo	Descripción
RF – CEMOSA – fi01	Obtención de la muestra	Se habilitará en la aplicación SAP – Fiori la posibilidad de introducir los datos de la muestra

RF – CEMOSA – fi02	Albarán	Generación y visualización del albarán de la muestra con los ensayos notificados que se desean visualizar
--------------------	---------	---

## 2.2.2 Requisitos No Funcionales

En el siguiente apartado se habla de los requisitos no funcionales, que son aquellos que indican restricciones que el sistema debe cumplir.

### 2.2.2.1 Requisitos No Funcionales del Código

RNF		
Código	Título	Descripción
RNF – CEMOSA – 01	Integridad de datos	Garantía de que la información es exacta, completa, homogénea, sólida y coherente
RNF – CEMOSA – 02	Ley de protección de datos	Es la protección de la información importante ante la corrupción y/o pérdida
RNF – CEMOSA – 03	Seguridad	Es la creación de métodos, procedimientos y normas que logren identificar y eliminar vulnerabilidades en la información
RNF – CEMOSA – 04	Control de volumen de datos	Es la gestión de la cantidad de datos que se maneja en el sistema
RNF – CEMOSA – 05	Tolerancia a fallos	Es la propiedad que permite al sistema seguir funcionando en caso de fallo en uno o varios componentes

RNF – CEMOSA – 06	Documentación	Nos ayudará a gestionar en el riesgo del mismo, entre otras muchas cosas como la de tomar decisiones acertadas, apoyadas sobre datos específicos
RNF – CEMOSA – 07	Acceso a web	Debe ser posible acceder desde cualquier dispositivo. Además del diseño, deberá ser responsive o adaptativo

### 2.2.3 Requisitos de Información

Para finalizar se define requisitos de información que engloban al software. Estos son aquellos que aportan datos valiosos que están contenidos en la aplicación.

#### 2.2.3.1 Requisitos de información de la muestra

RI - Muestra		
Código	Título	Descripción
RI – CEMOSA – mu01	Datos de la muestra	<ul style="list-style-type: none"> <li>- Tiempo de Penetración</li> <li>- Temperatura Ambiente Encima</li> <li>- Temperatura Ambiente Debajo</li> </ul>
RI – CEMOSA – mu02	Servicios	<ul style="list-style-type: none"> <li>- Código de Servicio</li> <li>- Nombre de servicio</li> <li>- Norma aplicada para el servicio</li> </ul>
RI – CEMOSA – mu03	Informe	<ul style="list-style-type: none"> <li>- Tipo de Informe</li> </ul>

#### 2.2.3.2 Requisitos de información del ensayo

RI – Ensayo
-------------

Código	Título	Descripción
RI – CEMOSA – en01	Datos del ensayo	<ul style="list-style-type: none"> <li>- Elemento</li> <li>- Localización</li> <li>- Nomenclatura</li> <li>- Defectos</li> <li>- Nivel de calidad</li> <li>- Calificación</li> <li>- Control Geométrico</li> <li>- Conclusiones</li> </ul>
RI – CEMOSA – en02	Informe	Se añadirán los datos al informe para su visualización

### 2.2.3.3 Requisitos de información de Fiori

RI – Fiori		
Código	Título	Descripción
RI – CEMOSA – fi01	Datos de la muestra	<ul style="list-style-type: none"> <li>- Número muestra</li> <li>- Datos de la muestra</li> <li>- Datos del ensayo</li> </ul>
RI – CEMOSA – fi02	Albarán	<ul style="list-style-type: none"> <li>- Generación del albarán</li> </ul>

### 2.2.4 Casos de uso

El Diagrama de Caso de Uso siguiente muestra la interacción entre los diversos usuarios y el sistema (Ilustración 4).

El técnico u operador realiza la toma de ensayos y estos se incluyen en la muestra. Si el usuario lo desea o lo requiere puede generar y visualizar el albarán.

El usuario que está en el laboratorio podrá visualizar los ensayos y servicios que ha creado el técnico, además de realizar las comprobaciones necesarias pertinentes. Por otro lado, podrá visualizar el albarán que generó el técnico. Además, podrá generar el informe y visualizarlo.

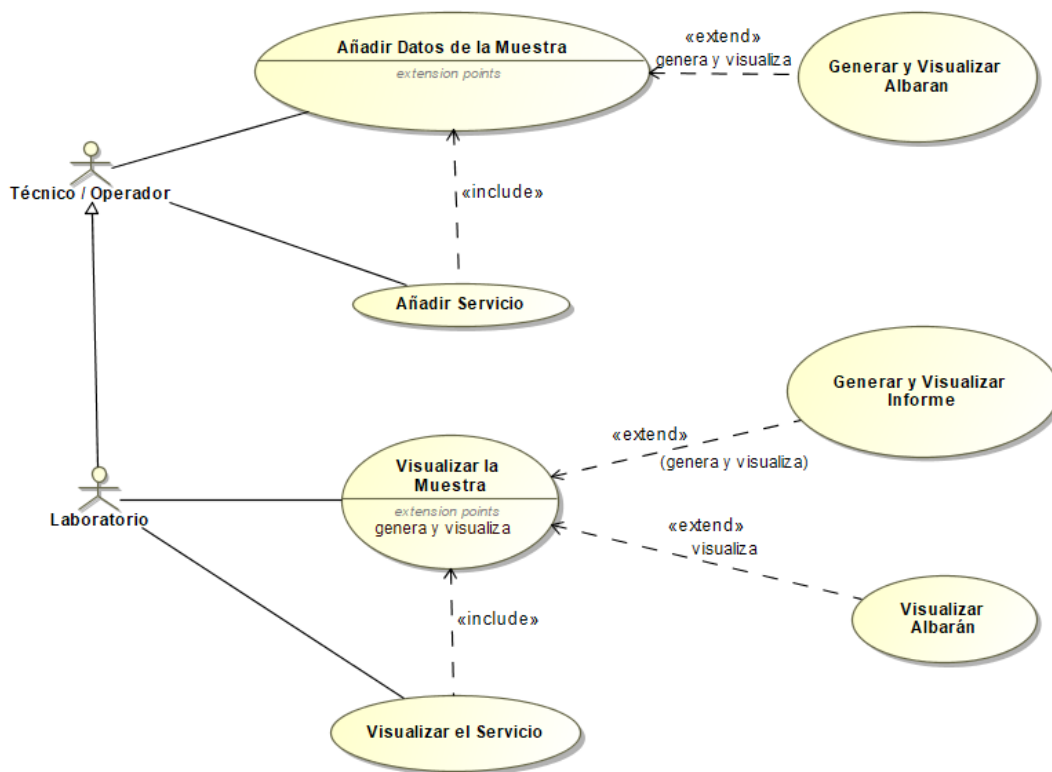


Ilustración 4: Diagrama de Casos de Uso

### 2.3 Diseño

Se realiza un estudio de lo que es necesario para poder desarrollar el proyecto. Se procede a crear los dominios y elementos que son necesarios. Estos forman la unidad básica que componen las tablas donde se almacenan los valores que los operarios insertan.

ZTLB_MUESTRAS	Tabla maestra de muestras		
IDMUESTRA	CHAR	12	Identificador de la muestra
RESTO DE CAMPOS	-	-	No relacionados con materiales metálicos.

Tabla 1. ZTLB\_MUESTRAS

ZTLB_MURELSER	Tabla maestra de relación Muestras – Servicios		
IDMUESTRA	CHAR	12	Identificador de la muestra
AUFPL	NUMC	10	N.º Hoja ruta de operaciones de orden



<b>APLZL</b>	NUMC	8	Contador general de la orden
RESTO DE CAMPOS	-	-	No relacionados con materiales metálicos

**Tabla 2. ZTLB\_MURELSER**

<b>ZTLB_METAL</b>	Tabla maestra de Materiales Metálicos		
<b>IDMUESTRA</b>	CHAR	12	Identificador de la muestra
<b>AUFPL</b>	NUMC	10	N.º Hoja ruta de operaciones de orden
<b>APLZL</b>	NUMC	8	Contador general de la orden
TPENETRACION	NUMC	2	Tiempo de penetración
TAMBIENTEENC	NUMC	2	Temperatura Ambiente por Encima
TAMBIENTEDEB	NUMC	2	Temperatura Ambiente por Debajo
LOTEPENETRANTE	CHAR	14	N.º Lote del material Penetrante
LOTEELIMINADOR	CHAR	14	N.º Lote del material Eliminador
LOTEREVELADOR	CHAR	14	N.º Lote del material Revelador

**Tabla 3. ZTLB\_METAL**

La tabla *ZTLB\_METAL* es la tabla maestra de materiales metálicos del ensayo. Para su identificación la tabla contiene una serie de elementos únicos (*MANDT*, *IDMUESTRA*, *AUFPL*, *APLZL*). Estos campos aseguran que no hay ningún ensayo repetido en la tabla.

<b>ZTLB_ELEMMETAL</b>	Tabla maestra para los Elementos de los Resultados		
<b>IDMUESTRA</b>	CHAR	12	Identificador de la muestra
<b>AUFPL</b>	NUMC	10	N.º Hoja ruta de operaciones de orden
<b>APLZL</b>	NUMC	8	Contador general de la orden
<b>IDELEMENTO</b>	NUMC	2	Identificador del elemento del resultado
ELEMENTO	CHAR	100	Elemento del resultado
LOCALIZACION	CHAR	70	Localización donde se ha realizado el elemento
NOMENCLATURA	CHAR	12	Nomenclatura del elemento
DEFECTOS	CHAR	35	Defectos obtenidos del elemento
NVLCALIDAD	CHAR	2	Nivel de Calidad del elemento
CALIFICACION	CHAR	7	Calificación del elemento
CTRLGEOM	CHAR	9	Control Geométrico del elemento
OBSERVACIONES	CHAR	255	Observaciones con respecto al elemento

**Tabla 4. ZTLB\_ELEMENTOSMETAL**

### 2.3.1 Modelo Conceptual

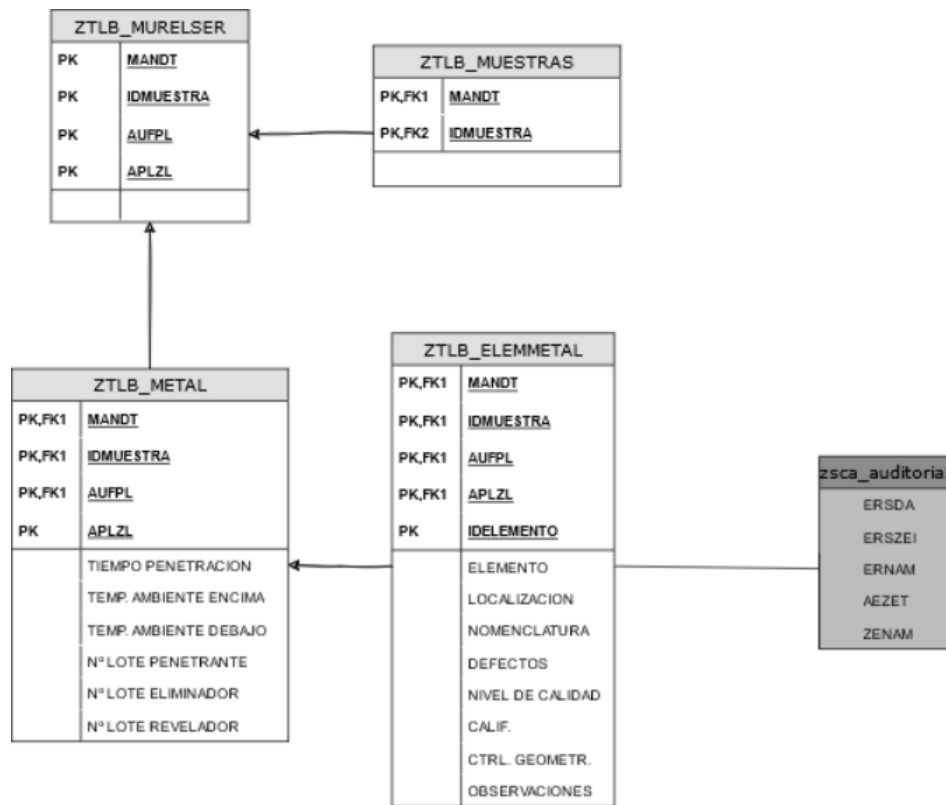


Ilustración 5: Modelo conceptual

En la imagen de arriba (Ilustración 7) se muestra la relación entre las muestras y los ensayos de Soldadura de Materiales Metálicos. La tabla *ZTLB\_MUESTRAS* contiene los campos generales de las muestras y es identificada mediante dos claves primarias, *MANDT*, que es el mandante (empresa) a la que se le asocia la muestra y el *IDMUESTRA*, que es un identificador único que se asigna a cada muestra. Los demás campos de esta tabla son usados por el laboratorio para muchos tipos de ensayos.

Disponemos también de la tabla *ZTLB\_MURELSER*, la cual se encarga de relacionar los ensayos o servicios con las muestras. Como claves en esta tabla tenemos el *MANDT*, *IDMUESTRA*, *AUFPL* y *APLZL*. Las dos primeras claves las conocemos de la tabla de *ZTLB\_MUESTRAS*, en cambio, las otras dos son claves para identificar los servicios que están añadidos en las muestras. El *AUFPL* es un código de servicio asociado al ensayo, el cual identifica el grafo del expediente al que pertenece el ensayo, y el *APLZL* es un registro numérico, un contador que sirve para identificar el orden del ensayo. Aparte de estas claves,

los otros campos que se encuentran en esta tabla son para identificar, por ejemplo, cuando se hicieron cada servicio y quién los realizó.

La tabla de *ZTLB\_METAL*, está identificada por cuatro claves primarias. Dos son heredades de la tabla *ZTLB\_MUESTRAS* y se añaden además otras dos, que son el *AUFPL* y *APLZL*. Estas claves tendrán el mismo valor que en la tabla *ZTLB\_MURELSER*. Se añade en esta tabla otros campos que formarán parte de los datos del ensayo, así como, el Tiempo de Penetración, Temperatura Ambiente por Encima y por Debajo, y los Números de Serie de los Lotes usados para el lote de Penetrante, el de Eliminador y el de Revelador.

La tabla que contendrá los datos de los resultados de los Elementos aplicados en el ensayo será la tabla *ZTLB\_ELEMMETAL*. Esta tendrá como claves primarias los mismos campos que la tabla *ZTLB\_METAL*, pero le vamos a añadir una clave primaria más para poder identificar los distintos elementos que se puedan añadir para un mismo servicio, es decir, en una muestra puede haber más de un elemento inspeccionado o tener el mismo elemento, pero con propiedades distintas. Propiedades como la Localización, la Nomenclatura, los Defectos, el Nivel de Calidad, el Control Geométrico y las Observaciones que se quieran hacer sobre el elemento inspeccionado.

### **2.3.2 Diagrama de secuencia**

A continuación, se presentan varios diagramas de flujo o de secuencia (Ilustración 5). Uno para cada usuario de la aplicación:

El técnico u operador, realiza la toma de la muestra y lo anota en la APP Fiori, le pedirá que introduzca los datos del servicio que ha realizado, así como fotografías y bocetos que aportarán rigidez a las tomas de datos realizadas. Una vez ha completado con éxito el paso anterior, el usuario podrá generar el albarán, y posteriormente dispondrá de la opción de visualizarlo si lo desea.

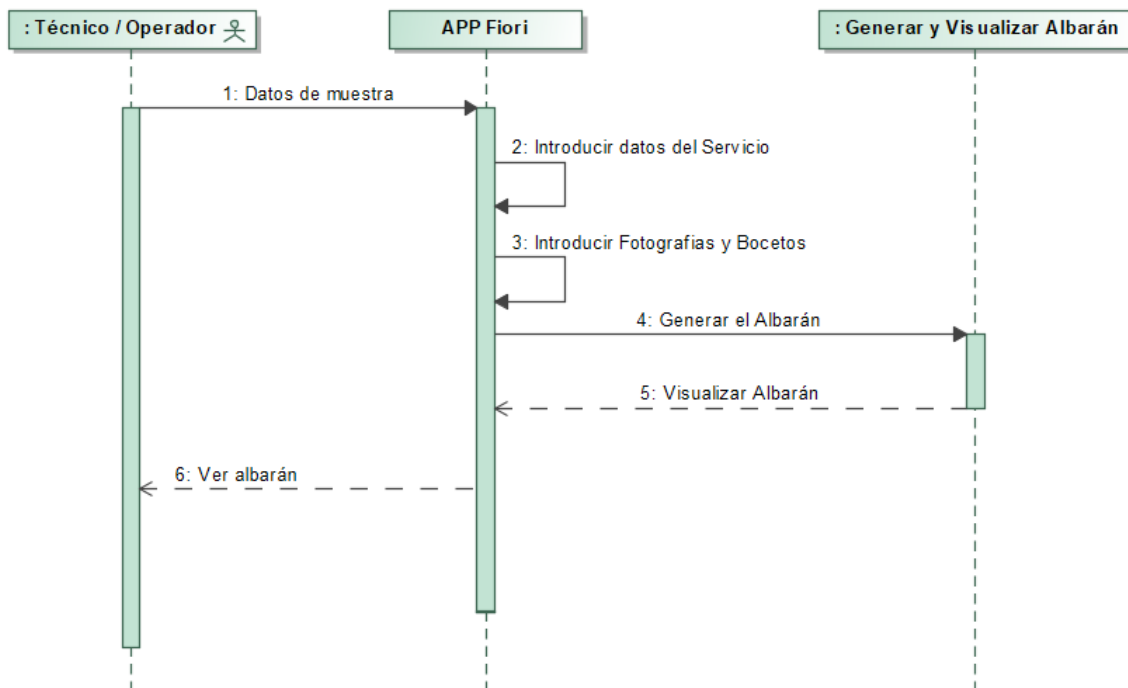


Ilustración 6: Flujo de datos 1

En la siguiente imagen (Ilustración 6) se puede observar cómo el usuario de laboratorio puede visualizar la muestra que ha sido previamente registrada por el usuario técnico. Si detecta algún tipo de error en la muestra, se lo reportará al usuario para que realice las comprobaciones necesarias o vuelva a realizar la toma de muestra. Si por el contrario no detecta ningún tipo de error, podrá realizar la asignación de la plantilla al informe y en último caso dispondrá de la opción de visualización, lo que le permitirá ver el informe.

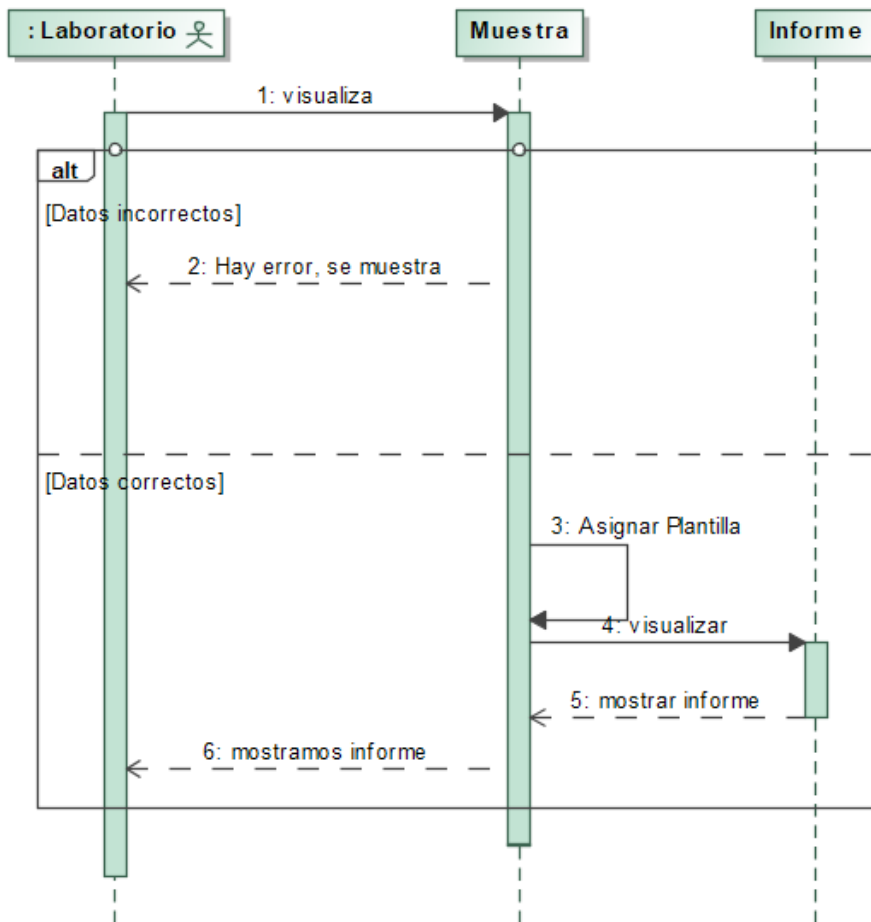


Ilustración 7: Flujo de datos 2

### 2.3.3 Diagrama de clases

La Muestra que está en el centro de la imagen es el pilar angular (Ilustración 8). Una muestra si es que existe, puede contener 1 o más servicios. Estos servicios son los ensayos o pruebas que se realizan y que posteriormente se analizan. Como se puede ver en la imagen, un servicio no puede existir si no existe una muestra.

El albarán se genera y está asociado a una única muestra, no puede haber un albarán que esté en varias muestras. Este caso es recíproco pues una muestra solo puede estar en un Albarán. Si existe otra muestra tendrá que generarse otro albarán para esa muestra.

Lo mismo ocurre con el informe. Una muestra se encuentra en un informe, que es similar al albarán, salvo que el informe aporta algunas cosas extra, entre ellas, la firma de validación. Un informe solo tiene una muestra. Si fuese necesario realizar un informe distinto, este solo puede contener una muestra distinta.

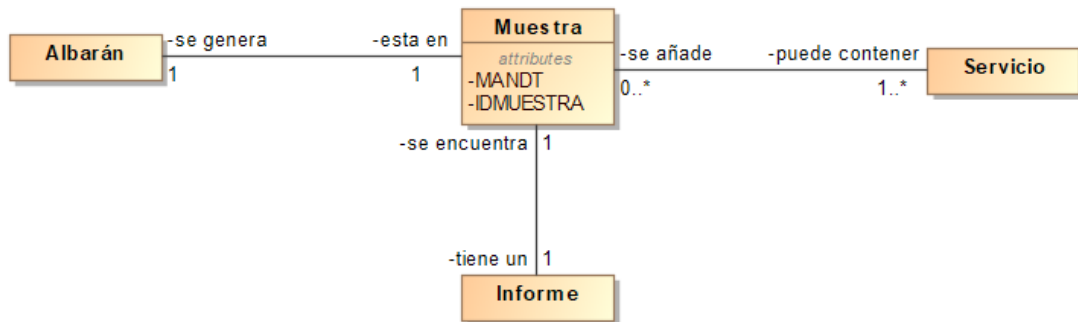


Ilustración 8: Diagrama de clases 1

Nuestro diagrama (Ilustración 9) está relacionado con las dos tablas que creamos que se unen con las dos tablas maestras que se encargan de registrar los datos de la muestra relacionando los ensayos que haya en esta muestra. Para ello, los datos de la muestra se guardan en *ZTLB\_MUESTRAS* que se relacionan con la tabla *ZTLB\_MURELSER* que guarda los ensayos que haya en esta muestra. Por lo que relacionamos nuestra tabla principal que es *ZTLB\_METAL* con la tabla de los ensayos y a esta tabla estará relacionado la tabla de los elementos, *ZTLB\_ELEMMETAL*.

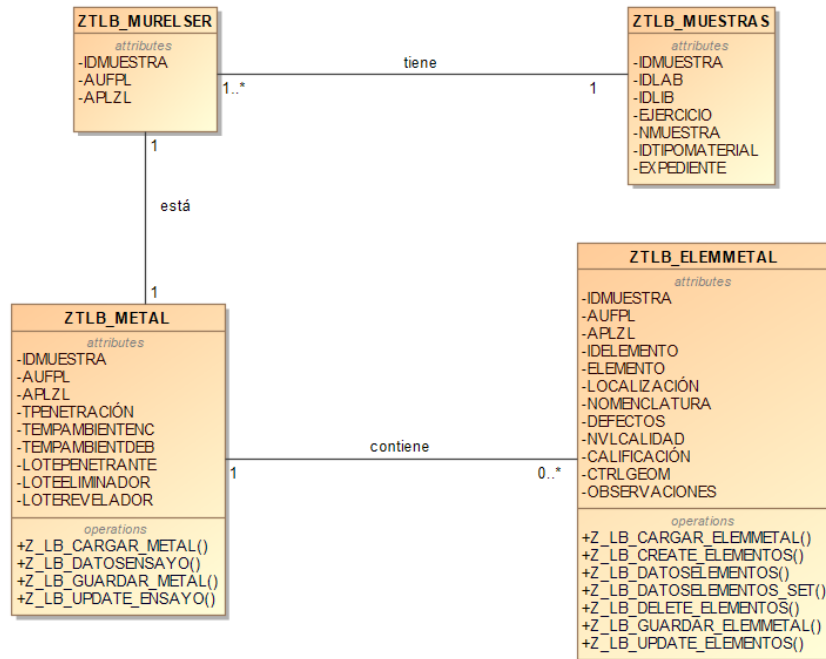


Ilustración 9: Diagrama de clases 2

## 2.4 Entrada de Datos del Servicio

Disponemos de un Programa creado como Modul-Pool que nos permite mantener los dynpros de una transacción. Un dynpro está compuesto por una pantalla y una lógica de procesos de la pantalla. En la pantalla se definen los atributos y en la lógica de procesos definimos los módulos que se ejecutarán antes de que el sistema visualice la pantalla y los procesos que se ejecutarán después (Ilustración 10).

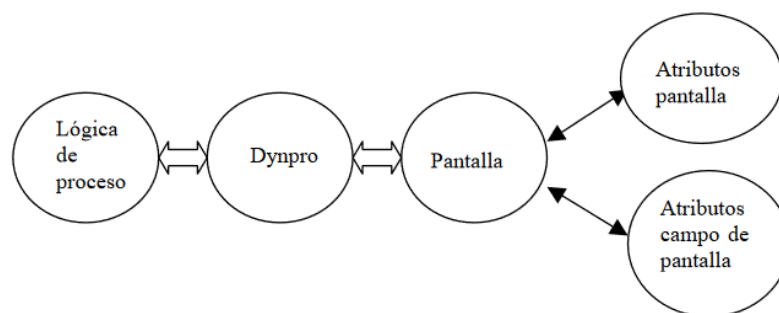


Ilustración 10: Objetos parciales de una dynpro

Este programa nos permite crear nuestra pantalla de resultados con la ayuda y las herramientas que nos proporciona pudiendo crear la siguiente pantalla (Ilustración 11). Está

pantalla solo será accesible para aquellas muestras que tengan el tipo de material de Materiales Metálicos y hagan el ensayo de Líquidos Penetrantes.

The screenshot shows a software interface for test design, organized into several sections:

- Datos de Cabecera:** Fields for Expediente, Obra, Cliente, Contratista, Observación, Fecha Inicio, Fecha Fin, Técnico Revisor, F.Revisión, Pedido, and a checkbox for 'Ensayo realizado'. It also includes fields for Muestra, Muestra O, Estado muestra, Tipo material, and F.Fin Muestra, along with R.L. (Rol) fields.
- Datos del Ensayo:** Fields for Tiempo Penetración, T° Ambiente Encima, T° Ambiente Debajo, Lote Penetrante, Lote Eliminador, and Lote Revelador.
- Datos de Elementos:** A table with columns: ID, Elemento, Localización, Nomenclatura, Defectos, Nivel de Calidad, Calif., Ctrl. Geom., and Observaciones.
- Datos Auditoria:** Fields for Creado el, Creado por, Última modif., and Modificado por.

Ilustración 11: Diseño pantalla del ensayo

Para un usuario común, a la hora de entrar y ver la pantalla de entrada de resultados se verá de la siguiente forma. Donde podrá ver los datos que previamente estén guardados para dicha muestra, añadir datos nuevos, modificar datos existentes, eliminar datos existentes, generar el informe de la muestra o visualizar el informe de dicha muestra (Ilustración 12).

The screenshot shows the 'Entrada de Resultados 7 (Materiales Metálicos)' interface, displaying the following data:

- Header:** 8002 - ACESOL INSPECCIÓN POR LÍQUIDOS PENETRANT
- Datos de Cabecera:** Expediente: 0/1107639/1/01; Obra: 074619; Cliente: 11538; Contratista: FCC CONST.,SA,CONST.VERA,SA, CONST.; Muestra: 2543; Muestra O: 0; Estado muestra: FI Finalizada; Tipo material: 78; F.Fin Muestra: 21.09.2011; Fecha Inicio: 21.09.2011; Pedido: 36453 / 21.09.2011; Ensayo realizado:
- Datos del Ensayo:** Tiempo Penetración, T° Ambiente Encima, T° Ambiente Debajo, Lote Penetrante, Lote Eliminador, Lote Revelador.
- Datos de Elementos:** A table with columns: ID, Elemento, Localización, Nomenclatura, Defectos, Nivel de Calidad, Calif., Ctrl. Geom., and Observaciones.
- Datos Auditoria:** Creado el: 26.04.2011; Creado por: 3451; Última modif.: 21.09.2011; Modificado por: 3012.

Ilustración 12: Entrada de Resultado del ensayo



Para el caso del Frontend, se programará dicho diseño en lenguaje SAPUI5 para nuestra aplicación en SAP Fiori. Tal que la pantalla queda como la siguiente (Ilustración 13).

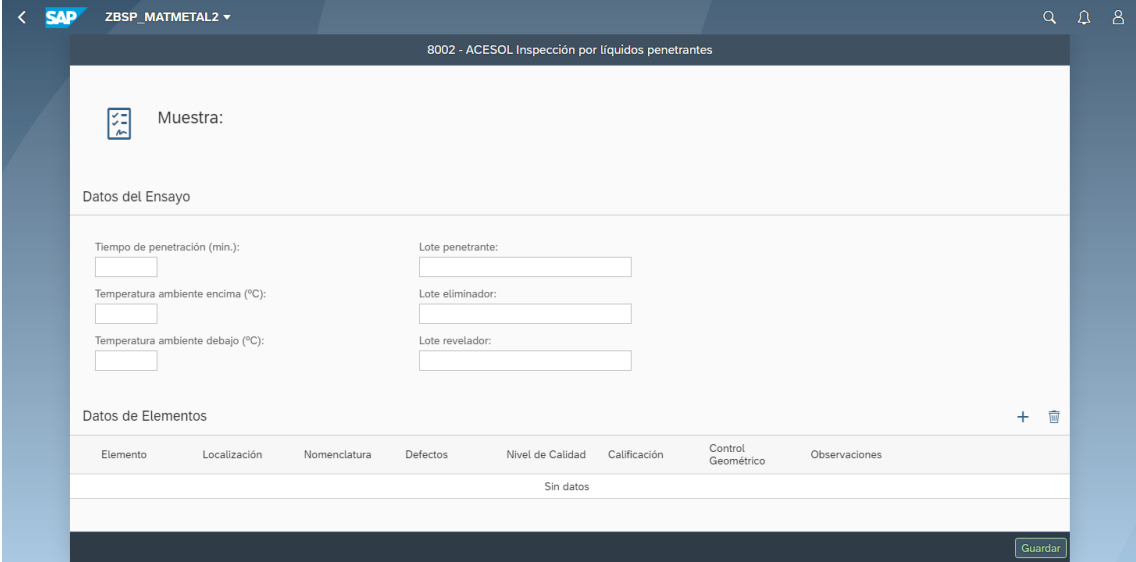


Ilustración 13: Visualización del ensayo en Frontend



# 3

## Desarrollo

### 3.1 Creación de las Interfaces

Para la realización del proyecto, debemos crear las tablas, estructuras, tipos de tablas y sobre todo los elementos de datos y dominios que no existen previamente en SAP. Sabemos que las tablas, estructuras y tipos de tablas serán nuevas puesto que no se ha llegado a realizar este trabajo nunca antes. En cuanto a los elementos de datos y dominios, estos son el tipo de dato de un campo de la tabla, es decir, en el dominio especificamos qué tipo de dato va a tener nuestro campo y la longitud que puede llegar a tener, y en el elemento de datos ponemos ese dominio creado más el nombre que tendrá este campo cuando se visualice en algún listado.

Para crear esto disponemos en SAP de la transacción *SE11* (Ilustración 14) que nos permite crear, modificar, visualizar, copiar o borrar las tablas, vistas, tipos de datos, etc.

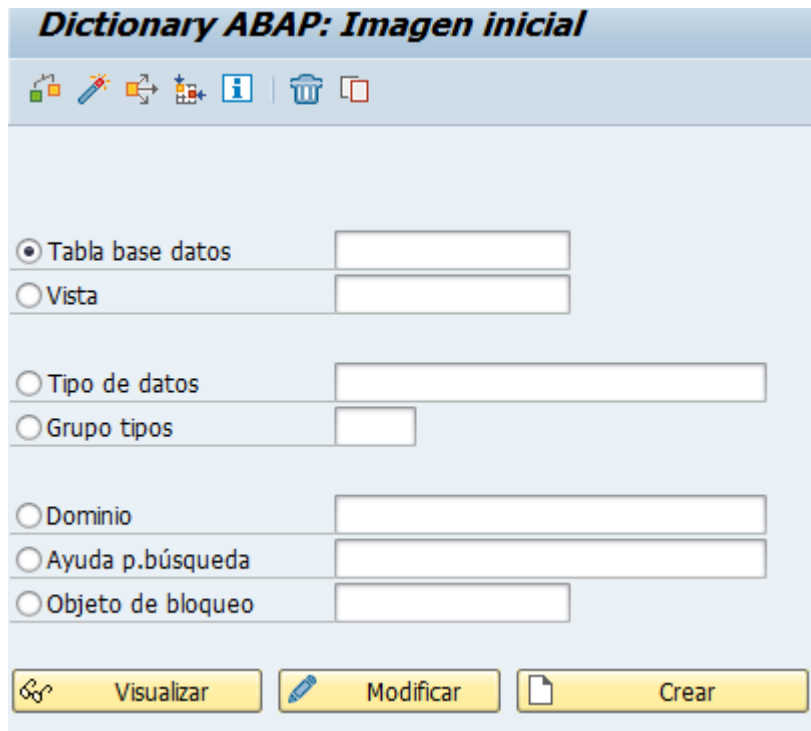


Ilustración 14: Transacción SE11

Antes de crear algo nuevo, siempre es importante buscar si ya existe un dominio o un elemento de datos en SAP y que podamos reutilizar. Para crear un dominio, elegimos la opción de Dominio e introducimos el dominio que queremos añadir y le damos a Crear (Ilustración 15). Una vez dentro, especificamos la descripción, el tipo de dato y la cantidad de posiciones.

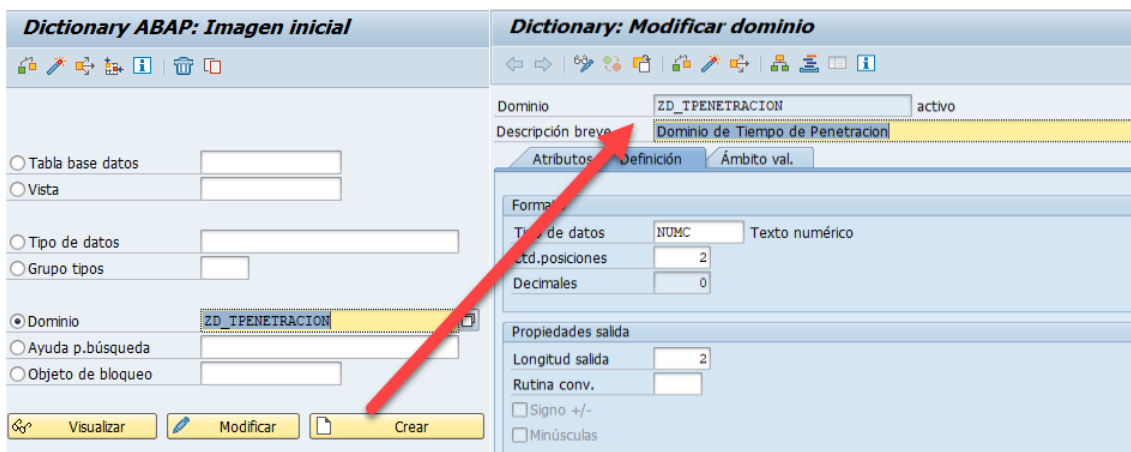


Ilustración 15: Crear un Dominio

Luego crearemos el elemento de dato que contendrá este dominio. Para ello, volvemos a la transacción y seleccionamos la opción de Tipo de Datos, introducimos el nombre que queremos asignar y le damos a crear (Ilustración 16). Nos sale una ventana para elegir qué queremos crear, ya que esta opción de Tipo de Datos permite crear un elemento de datos, una estructura o un tipo tabla. En este caso seleccionamos Elem. Datos y aceptamos. Luego, insertamos la descripción y el dominio previamente creado.

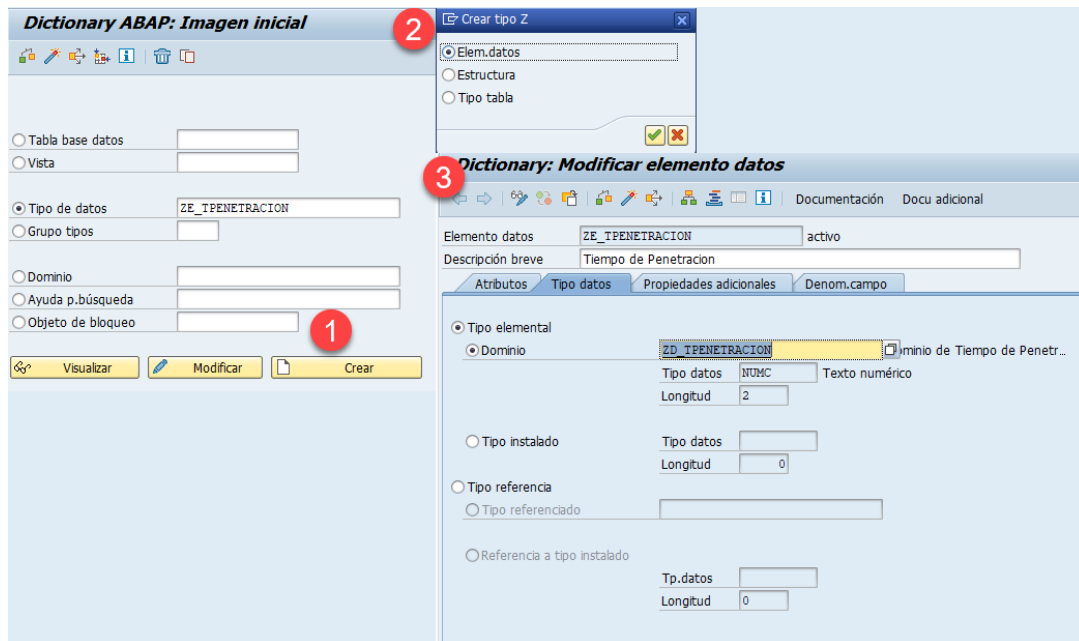


Ilustración 16: Crear un Elemento de Dato

Una vez tengamos ya todos los dominios creados e insertados en todos sus respectivos elementos de datos, crearemos la tabla donde asignaremos estos elementos de datos a los campos de la tabla. Para ello, de la transacción seleccionamos la opción de Tabla base datos e insertamos el nombre de nuestra tabla para luego darle a Crear (Ilustración 17).

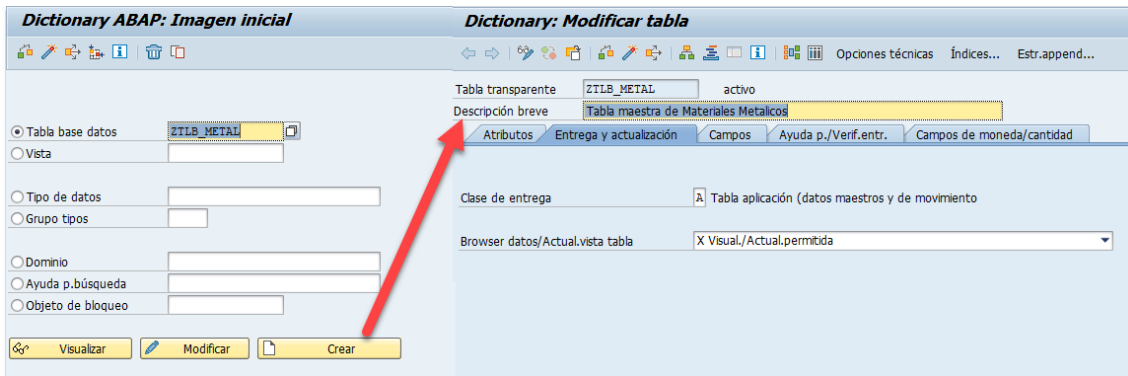


Ilustración 17: Crear una tabla

Insertamos la descripción, la clase de entrega y la opción de actualizar la tabla o no. La clase de entrega es para especificar cómo se va a comportar la tabla y para ello SAP te da a elegir entre las siguientes opciones (Ilustración 18).

Clase entrega	Descrip.breve
A	Tabla aplicación (datos maestros y de movimiento)
C	Tabla customizing, actual.sólo por cliente, sin import SAP
L	Tabla para almacenar datos temporales, entregada vacía
G	Tabla customizing, protegida contra UPD de SAP, sólo INS
E	Tabla control, SAP y cliente tienen ámbitos clave propios
S	Tabla sistema, actual.sólo por SAP, Cambio = Modificación
W	Tabla sistema, cont. transportable con objetos TR propios

Ilustración 18: Clase de entrega de la tabla

Una vez realizado esto, iremos a la pestaña de Campos para añadir los campos de la tabla y asignar sus respectivos elementos de tabla, tal que queda como lo siguiente en SAP (Ilustración 19):

Tabla transparente  ZTLB\_METAL  activo

Descripción breve

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

1 / 10

Campo	Clv	Val...	Elem.datos	Tipo de d...	Long.	Deci...	Descripción breve
MANDI	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDI	CLNT	3	0	Mandante
IDMUESTRA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZE_IDMUESTRA	CHAR	12	0	Identificador de Muestra
AUFPL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CO_AUFPL	NUMC	10	0	Nº hoja ruta de operaciones en orden
APLZL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CO_APLZL	NUMC	8	0	Contador general de la orden
TPENETRACION	<input type="checkbox"/>	<input type="checkbox"/>	ZE_TPENETRACION	NUMC	2	0	Tiempo de Penetracion
TAMBIENTEENC	<input type="checkbox"/>	<input type="checkbox"/>	ZE_TAMBIENTEENC	NUMC	2	0	Temperatura Ambiente por Encima
TAMBIENTEDEB	<input type="checkbox"/>	<input type="checkbox"/>	ZE_TAMBIENTEDEB	NUMC	2	0	Temperatura Ambiente por Debajo
LOTEPENETRANTE	<input type="checkbox"/>	<input type="checkbox"/>	ZE_LOTEPENETRAN...	CHAR	14	0	Nº Lote del material Penetrante
LOTEELIMINADOR	<input type="checkbox"/>	<input type="checkbox"/>	ZE_LOTEELIMINAD...	CHAR	14	0	Nº Lote del material Eliminator
LOTEREVELADOR	<input type="checkbox"/>	<input type="checkbox"/>	ZE_LOTEREVELADOR	CHAR	14	0	Nº Lote del material Revelador

Ilustración 19: Tabla ZTLB\_METAL

Para una fácil utilización de esta tabla, se crea una estructura llamada ZSLB\_METAL (Ilustración 20):

Estructura  ZSLB\_METAL  activo

Descripción breve

Atributos Componentes Ayuda p./Verif.entr. Campos de moneda/cantidad

1 / 11

Componente	Clase tipificación	Tp.componente	Tipo de d...	Long.	Deci...	Descripción breve
.INCLUDE	1 Type	ZTLB_METAL			0	Tabla maestra de Materiales Metalicos
MANDI	1 Type	MANDI	CLNT	3	0	Mandante
IDMUESTRA	1 Type	ZE_IDMUESTRA	CHAR	12	0	Identificador de Muestra
AUFPL	1 Type	CO_AUFPL	NUMC	10	0	Nº hoja ruta de operaciones en orden
APLZL	1 Type	CO_APLZL	NUMC	8	0	Contador general de la orden
TPENETRACION	1 Type	ZE_TPENETRACION	NUMC	2	0	Tiempo de Penetracion
TAMBIENTEENC	1 Type	ZE_TAMBIENTEENC	NUMC	2	0	Temperatura Ambiente por Encima
TAMBIENTEDEB	1 Type	ZE_TAMBIENTEDEB	NUMC	2	0	Temperatura Ambiente por Debajo
LOTEPENETRANTE	1 Type	ZE_LOTEPENETRAN...	CHAR	14	0	Nº Lote del material Penetrante
LOTEELIMINADOR	1 Type	ZE_LOTEELIMINAD...	CHAR	14	0	Nº Lote del material Eliminator
LOTEREVELADOR	1 Type	ZE_LOTEREVELADOR	CHAR	14	0	Nº Lote del material Revelador

Ilustración 20: Estructura ZSLB\_METAL

Y para poder usarla como una tabla, se crea el tipo tabla ZSTLB\_METAL (Ilustración 21):

Tipo tabla  activo

Descripción breve

Atributos   Tipo de línea   Inicialización y acceso   Clave primaria   Clave secundaria

Tipo de línea

Tipo instalado

    Tipo de datos

    Ctd.posiciones     Decimales

Tipo referencia

Tipo referenciado

Referencia a tipo instalado

        Tipo datos

        Longitud     Decimales

Ilustración 21: Tipo tabla ZSTLB\_METAL

La otra tabla que se ha creado es la tabla ZTLB\_ELEMMETAL (Ilustración 22).

Tabla transparente   activo

Descripción breve

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

Ay.búsq.   Tipo instalado   1 / 20

Campo	Cv	Val...	Elem.datos	Tipo de d...	Long.	Deci...	Descripción breve
MANDI	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDI	CLNT	3	0	Mandante
IDMUESTRA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZE_IDMUESTRA	CHAR	12	0	Identificador de Muestra
AUFPL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CO_AUFPL	NUMC	10	0	Nº hoja ruta de operaciones en orden
APLZL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CO_APLZL	NUMC	8	0	Contador general de la orden
IDELEMENTO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZE_IDELEMENTOME	NUMC	2	0	ID del Elemento de Resultados
ELEMENTO	<input type="checkbox"/>	<input type="checkbox"/>	ZE_ELEMENTO	CHAR	100	0	Elemento del resultado
LOCALIZACION	<input type="checkbox"/>	<input type="checkbox"/>	ZE_LOCALIZACION	CHAR	70	0	Localizacion del elemento
NOMENCLATURA	<input type="checkbox"/>	<input type="checkbox"/>	ZE_NOMENCLATURA	CHAR	12	0	Nomenclatura del elemento
DEFECTOS	<input type="checkbox"/>	<input type="checkbox"/>	ZE_DEFECTOS	CHAR	35	0	Defectos del elemento
NVLCALIDAD	<input type="checkbox"/>	<input type="checkbox"/>	ZE_NVLCALIDAD	CHAR	2	0	Nivel de Calidad del elemento
CALIFICACIONELEM	<input type="checkbox"/>	<input type="checkbox"/>	ZE_CALIFICACION..	CHAR	7	0	Calificacion del elemento
CTRLGEOM	<input type="checkbox"/>	<input type="checkbox"/>	ZE_CTRLGEOM	CHAR	9	0	Control Geometrico del elemento
OBSERVACION	<input type="checkbox"/>	<input type="checkbox"/>	ZE_TXT_OBSERV	CHAR	200	0	Texto Observaciones
.INCLUDE	<input type="checkbox"/>	<input type="checkbox"/>	ZSCA_AUDITORIA	STRU	0	0	Campos de auditoría de tablas
ERSDA	<input type="checkbox"/>	<input type="checkbox"/>	ERSDA	DATS	8	0	Fecha de creación
ERZEI	<input type="checkbox"/>	<input type="checkbox"/>	ANUZI	TIMS	6	0	Hora en la que fue creado el registro
ERNAM	<input type="checkbox"/>	<input type="checkbox"/>	ERNAM	CHAR	12	0	Nombre del responsable que ha añadido el objeto
LAEDA	<input type="checkbox"/>	<input type="checkbox"/>	LAEDA	DATS	8	0	Fecha última modificación
AEZET	<input type="checkbox"/>	<input type="checkbox"/>	AEZET	TIMS	6	0	Hora de última modificación
AENAM	<input type="checkbox"/>	<input type="checkbox"/>	AENAM	CHAR	12	0	Nombre del responsable que ha modificado el objeto

Ilustración 22: Tabla ZTLB\_ELEMMETAL

Para ello, se ha creado también la estructura ZSLB\_ELEMMETAL (Ilustración 23):



Componente	Clase tipificación	Tp.componente	Tipo de d...	Long.	Deci...	Descripción breve
<u>SEL</u>	1 Type	▼ ZE_SELECT	CHAR		1	0 Campo que indica si se ha seleccionado fila en la tabla
<u>.INCLUDE</u>	1 Type	▼ ZTLB_ELEMMETAL	☐		0	0 Tabla maestra para los Elementos de los Resultados
<u>MANDI</u>	1 Type	▼ MANDI	CLNI		3	0 Mandante
<u>IDMUESTRA</u>	1 Type	▼ ZE_IDMUESTRA	CHAR		12	0 Identificador de Muestra
<u>AUFPL</u>	1 Type	▼ CO_AUFPL	NUMC		10	0 N° hoja ruta de operaciones en orden
<u>APLZL</u>	1 Type	▼ CO_APLZL	NUMC		8	0 Contador general de la orden
<u>IDELEMENTO</u>	1 Type	▼ ZE_IDELEMENTOME	NUMC		2	0 ID del Elemento de Resultados
<u>ELEMENTO</u>	1 Type	▼ ZE_ELEMENTO	CHAR		100	0 Elemento del resultado
<u>LOCALIZACION</u>	1 Type	▼ ZE_LOCALIZACION	CHAR		70	0 Localizacion del elemento
<u>NOMENCLATURA</u>	1 Type	▼ ZE_NOMENCLATURA	CHAR		12	0 Nomenclatura del elemento
<u>DEFECTOS</u>	1 Type	▼ ZE_DEFECTOS	CHAR		35	0 Defectos del elemento
<u>NVLCALIDAD</u>	1 Type	▼ ZE_NVLCALIDAD	CHAR		2	0 Nivel de Calidad del elemento
<u>CALIFICACIONELEM</u>	1 Type	▼ ZE_CALIFICACION...	CHAR		7	0 Calificacion del elemento
<u>CTRLGEOM</u>	1 Type	▼ ZE_CTRLGEOM	CHAR		9	0 Control Geometrico del elemento
<u>OBSERVACION</u>	1 Type	▼ ZE_TXT_OBSERV	CHAR		200	0 Texto Observaciones
<u>.INCLUDE</u>	1 Type	▼ ZSCA_AUDITORIA	☐		0	0 Campos de auditoría de tablas
<u>ERSDA</u>	1 Type	▼ ERSDA	DATE		8	0 Fecha de creación
<u>ERZEI</u>	1 Type	▼ ANUZT	TIME		6	0 Hora en la que fue creado el registro
<u>ERNAM</u>	1 Type	▼ ERNAM	CHAR		12	0 Nombre del responsable que ha añadido el objeto
<u>LAEDA</u>	1 Type	▼ LAEDA	DATE		8	0 Fecha última modificación
<u>AEZEI</u>	1 Type	▼ AEZET	TIME		6	0 Hora de última modificación

Ilustración 23: Estructura ZSLB\_ELEMMETAL

Que a su vez, también disponemos del tipo tabla ZSTLB\_ELEMMETAL (Ilustración 24).

Tipo tabla		ZSTLB_ELEMMETAL		activo	
Descripción breve		Tipo tabla de ZSLB_ELEMMETAL			
Atributos		Tipo de línea		Inicialización y acceso	
Clave primaria		Clave secundaria			
<input checked="" type="radio"/> Tipo de línea		ZSLB_ELEMMETAL ☐			
<input type="radio"/> Tipo instalado		Tipo de datos <input type="text"/> Ctd.posiciones <input type="text" value="0"/> Decimales <input type="text" value="0"/>			
<input type="radio"/> Tipo referencia		<input type="radio"/> Tipo referenciado <input type="text"/>			
<input type="radio"/> Referencia a tipo instalado		Tipo datos <input type="text"/> Longitud <input type="text" value="0"/> Decimales <input type="text" value="0"/>			

Ilustración 24: Tipo tabla ZSTLB\_ELEMMETAL

Estas tablas, estructuras y tipo tablas formarán parte del servicio para poder realizar la entrada de resultados, pero falta por mostrar la creación del proyecto en el OData que se encargará de procesar los datos obtenidos de la aplicación de SAP-Fiori, cargando, tratando, guardando y mostrando los datos del servicio.

Para ello, nos creamos un proyecto llamado ZGW\_MATMETAL (Ilustración 25)

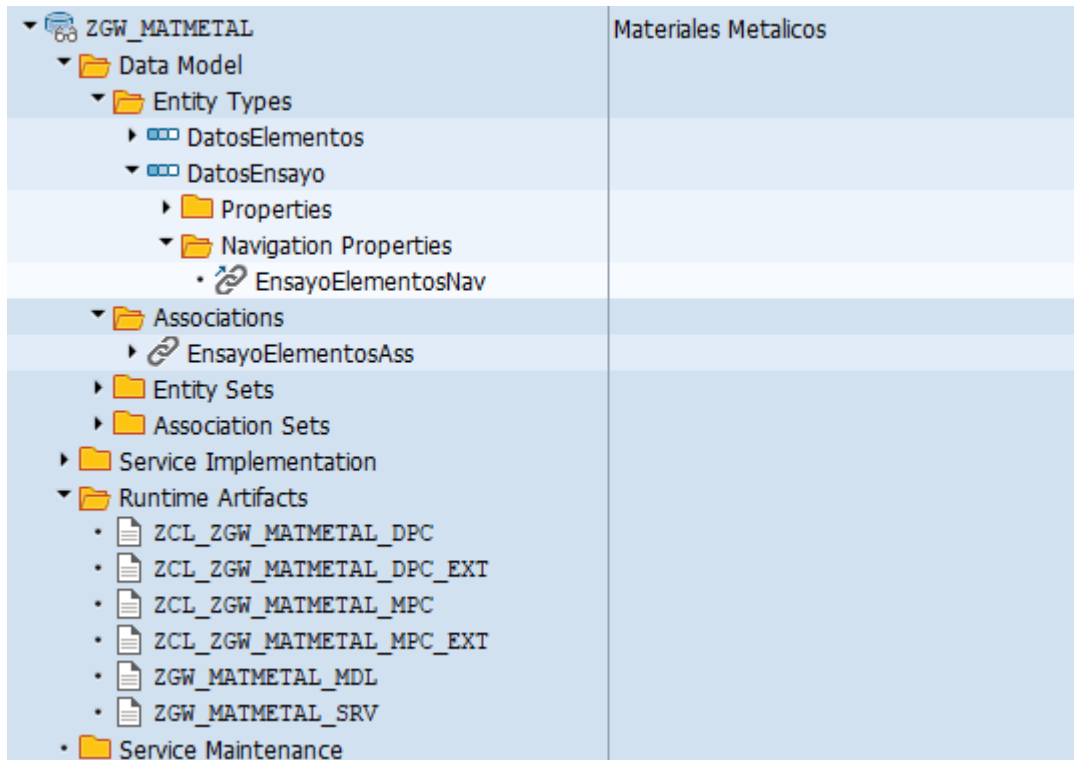


Ilustración 25: Proyecto ZGW\_MATMETAL

Donde tendremos una entidad que será DatosEnsayo, que hace referencia a la estructura ZSLB\_METAL (Ilustración 26).

Name	Key	Edm Core Type	Prec.	Scale	Max...	Unit Prop.	Cre...	Upd...	Sort...	Null...	Flt.	Label	Lab Comp. Type	ABAP Field Name	A...	Semantics
Idmuestra	<input checked="" type="checkbox"/>	Edm.String	0	0	12		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Identifir Muestra	<input type="checkbox"/>	IDMUESTRA	<input type="checkbox"/>	
Aufpl	<input checked="" type="checkbox"/>	Edm.String	0	0	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nº HRuta op.	<input type="checkbox"/>	AUFPL	<input type="checkbox"/>	
Aplzl	<input checked="" type="checkbox"/>	Edm.String	0	0	8		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Contador	<input type="checkbox"/>	APLZL	<input type="checkbox"/>	
Tpenetracion	<input type="checkbox"/>	Edm.String	0	0	2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tiem. Penetracion	<input type="checkbox"/>	TPENETRACION	<input type="checkbox"/>	
Tambienteenc	<input type="checkbox"/>	Edm.String	0	0	2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Temp. Amb. Enc...	<input type="checkbox"/>	TAMBIENTEENC	<input type="checkbox"/>	
Tambientedeb	<input type="checkbox"/>	Edm.String	0	0	2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Temp. Amb. De...	<input type="checkbox"/>	TAMBIENTEDEB	<input type="checkbox"/>	
Lotepenetrante	<input type="checkbox"/>	Edm.String	0	0	14		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Lote Penetrante	<input type="checkbox"/>	LOTEPENETRAN...	<input type="checkbox"/>	
Loteeliminador	<input type="checkbox"/>	Edm.String	0	0	14		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Lote Eliminador	<input type="checkbox"/>	LOTEELIMINADOR	<input type="checkbox"/>	
Loterevelador	<input type="checkbox"/>	Edm.String	0	0	14		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Lote Revelador	<input type="checkbox"/>	LOTEREVELADOR	<input type="checkbox"/>	

Ilustración 26: Entidad de DatosEnsayo

Y otra entidad, que se llamará DatosElementos, que hará referencia a la estructura ZSLB\_ELEMMETAL (Ilustración 27).

Name	Key	Edm Core	Type	Prec.	Scale	Max...	Unit	Prop.	Cre...	Upd...	Sort...	Null...	Flt.	Label	Lab Comp.	Type	ABAP Field Name	A... Semantics
Idmuestra	<input checked="" type="checkbox"/>	Edm.String		0	0	12			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Identifir Muestra	<input checked="" type="checkbox"/>		IDMUESTRA	<input type="checkbox"/>
Aufpl	<input checked="" type="checkbox"/>	Edm.String		0	0	10			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nº HRuta op.	<input checked="" type="checkbox"/>		AUFPL	<input type="checkbox"/>
Aplzl	<input checked="" type="checkbox"/>	Edm.String		0	0	8			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Contador	<input checked="" type="checkbox"/>		APLZL	<input type="checkbox"/>
Idelemento	<input checked="" type="checkbox"/>	Edm.String		0	0	2			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ID. Elemento	<input checked="" type="checkbox"/>		IDELEMENTO	<input type="checkbox"/>
Elemento	<input type="checkbox"/>	Edm.String		0	0	100			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Elemento Resul.	<input checked="" type="checkbox"/>		ELEMENTO	<input type="checkbox"/>
Localizacion	<input type="checkbox"/>	Edm.String		0	0	70			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Localizacion Elem.	<input checked="" type="checkbox"/>		LOCALIZACION	<input type="checkbox"/>
Nomenclatura	<input type="checkbox"/>	Edm.String		0	0	12			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nomenclatura	<input checked="" type="checkbox"/>		NOMENCLATURA	<input type="checkbox"/>
Defectos	<input type="checkbox"/>	Edm.String		0	0	35			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Defectos Elem.	<input checked="" type="checkbox"/>		DEFECTOS	<input type="checkbox"/>
Nvcalidad	<input type="checkbox"/>	Edm.String		0	0	2			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nvl. Calidad	<input checked="" type="checkbox"/>		INVLCALIDAD	<input type="checkbox"/>
Calificacionelem	<input type="checkbox"/>	Edm.String		0	0	7			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Calificacion Elem.	<input checked="" type="checkbox"/>		CALIFICACIONEL...	<input type="checkbox"/>
Ctrlgeom	<input type="checkbox"/>	Edm.String		0	0	9			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Ctrl. Geometrico	<input checked="" type="checkbox"/>		CTRLGEOM	<input type="checkbox"/>
Observacion	<input type="checkbox"/>	Edm.String		0	0	200			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Observaciones	<input checked="" type="checkbox"/>		OBSERVACION	<input type="checkbox"/>

Ilustración 27: Entidad de DatosElementos

Se mantendrá una relación de uno a muchos entre la entidad de DatosEnsayo y la de DatosElementos, para así poder obtener todas las filas de elementos que contenga el ensayo de la muestra.

### 3.2 Implementación de Código

A continuación, se muestra el progreso y resultado del esfuerzo y trabajo realizado durante el transcurso del proyecto.

Para ello, lo primero es configurar el ensayo que vamos a modificar para que cuando se utilice dicho ensayo nos navegue a la pantalla que crearemos, ya que ahora mismo está configurada de manera que navega a la pantalla de Entrada de Resultados General y esto se especifica desde la transacción ZML10 (Ilustración 28).

ID comb. norma	Dynpro	Descripción de la dynpro
1	9069	Entrada de Resultados General
2	9069	Entrada de Resultados General
3	9069	Entrada de Resultados General

Ilustración 28: Configuración del ensayo

Simplemente hay que modificar la dynpro que se va a utilizar, en este caso y para seguir el orden que hay de las dynpros ya existentes, es el 9012 (Ilustración 29).

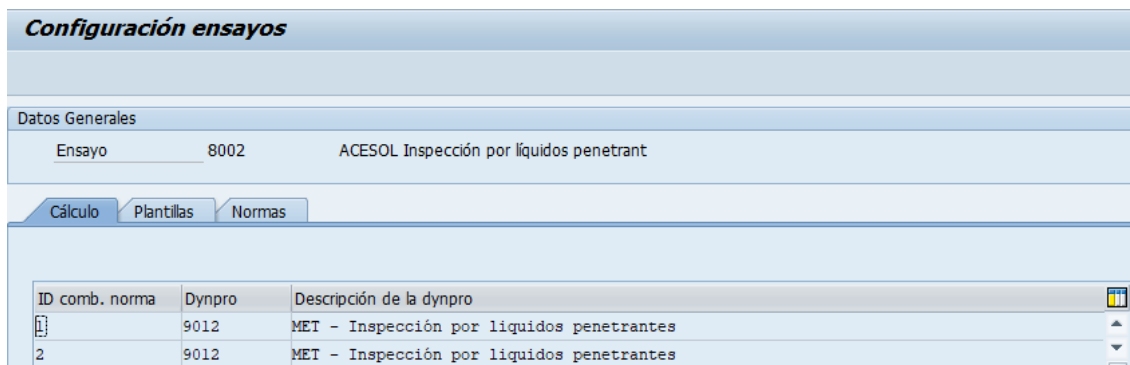


Ilustración 29: Ensayo configurado

Para hacerlo de manera más sencilla y de forma que se pueda reutilizar en cualquier momento, circunstancias, y sea escalable tanto como para este programa, como para futuros programas que precisen de obtener algún valor o valores. Se maximiza el uso del encapsulamiento en funciones, para reutilizar el código.

El programa se ejecuta en módulos de diálogo, ya sea para preparar la salida de la pantalla, eventos PBO, o para procesar datos introducidos por el usuario, eventos PAI. El PBO, del inglés (Process Before Output), es donde los procesos se realizan antes de mostrar la pantalla al usuario y el PAI, del inglés (Process After Input), es donde los procesos son ejecutados después de la interacción del usuario. Para su implementación ha sido necesario entender la lógica que siguen los programas en SAP. Y en concreto aquellos programas relacionados con las tareas de laboratorio y sus muestras.

Cuando se comprende el procedimiento que sigue al funcionamiento desde la recogida de muestras hasta su posterior exposición en la pantalla, se diseña el código que se integra en el sistema.

Para el correcto funcionamiento de la pantalla, insertamos los módulos que correspondan en el PBO (Ilustración 30). Los módulos *status\_9012* e *inicializardatos9012* son los módulos que se encargan de seleccionar el menú de la pantalla y de obtener los datos guardados que estén en las tablas, respectivamente. El módulo siguiente y el Loop se encargan de controlar la tabla de la pantalla a la hora de obtener los registros de la tabla y mostrarlos.

```

PROCESS BEFORE OUTPUT.
  " Asignamos Status
  MODULE status_9012.
  " Inicializamos datos
  MODULE inicializadatos9012.

*&SPWIZARD: PBO FLOW LOGIC FOR TABLECONTROL 'TC_ELEMMETAL'
  MODULE tc_elemmetal_change_tc_attr.
*&SPWIZARD: MODULE TC_ELEMMETAL_CHANGE_COL_ATTR.
  LOOP AT t_elemmetal
    INTO wa_telemmetal
    WITH CONTROL tc_elemmetal
    CURSOR tc_elemmetal-current_line.
    MODULE tc_elemmetal_get_lines.
*&SPWIZARD: MODULE TC_ELEMMETAL_CHANGE_FIELD_ATTR
  ENDLOOP.

```

Ilustración 30: PBO

El módulo *inicializadatos9012* contiene varios performs (subrutinas). Una de ellas carga los datos comunes de las tablas *ZTLB\_MUESTRAS* y *ZTLB\_MURELSE*. La otra carga los datos específicos del ensayo, es decir, de las tablas *ZTLB\_METAL* y *ZTLB\_ELEMMETAL* (Ilustración 31).

```

MODULE inicializadatos9012 OUTPUT.
  "Se cargan los datos de la tabla muestras y murelser.
  PERFORM f_inicialiar_datos_comunes.

  "Se cargan los datos del ensayo
  PERFORM f_inicializadatos9012.
ENDMODULE.

```

Ilustración 31: Inicializar datos 9012

La subrutina *f\_inicializadatos9012*, contiene una llamada a la función *z\_lb\_cargar\_metal\_elem* que carga los datos del ensayo de materiales metálicos. Para ello se utilizan las claves primarias, que son genéricas para todos los ensayos. Que son *IDMUESTRA*, *AUFPL* y *APLZL*. Estos datos se pasan a la función y ésta retorna *t\_elemmetal* que es declarado como tipo tabla de *zstlb\_elemmetal*, *t\_metal* que es declarado como tipo tabla de *zstlb\_metal* y *wa\_metal* que es declarado como una estructura interna para usar el tipo tabla *t\_metal* (Ilustración 32).

```

*   TDD4773 - 20.05.2021
*   Se inicializan los datos recuperando de las tablas
*-----*
FORM f_inicializardatos9012 .

" cargamos datos del ensayo
CALL FUNCTION 'Z_LB_CARGAR_METAL_ELEM'
EXPORTING
  pi_muestra      = wa_navegacion-idmuestra
  pi_aufpl        = wa_navegacion-aufpl
  pi_aplz1        = wa_navegacion-aplz1
IMPORTING
  pe_elemmetal    = t_elemmetal
  pe_metal        = t_metal
  pe_wa_metal     = wa_metal.

- ENDFORM.

```

Ilustración 32: Inicializar datos del ensayo

A su vez, en esta función se hace llamadas a las dos funciones que se encargan de obtener los datos de las tablas, una para la tabla *ZTLB\_METAL* que es la función *Z\_LB\_CARGAR\_METAL* y otra para la tabla *ZTLB\_ELEMMETAL* que es la función *Z\_LB\_CARGAR\_ELEMMETAL*. Ambas serán llamadas si es la primera vez que se entra a la pantalla del ensayo, puesto que, si no se controlase y se hiciese siempre, por cualquier cambio que se realice en dicha pantalla no se vería afectado, ya que el PBO se ejecutaría y por consecuente esta función volvería a cargar de las tablas y no se haría ningún cambio. Por ello, se controla que solo entran a las funciones de cargar si es la primera vez que se entra a la pantalla (Ilustración 33).

```

* TDD4773 - 20.05.2021
* Cargamos los datos si existen de la BBDD
* "-----
IF pe_metal IS INITIAL.
CALL FUNCTION 'Z_LB_CARGAR_METAL'
EXPORTING
    pi_muestra = pi_muestra
    pi_aufpl   = pi_aufpl
    pi_aplzl  = pi_aplzl
IMPORTING
    pe_metal  = pe_metal.

IF pe_metal IS NOT INITIAL.
    READ TABLE pe_metal INTO pe_wa_metal INDEX 1.
ENDIF.
ENDIF.

IF pe_elemmetal IS INITIAL.
CALL FUNCTION 'Z_LB_CARGAR_ELEMMETAL'
EXPORTING
    pi_muestra = pi_muestra
    pi_aufpl   = pi_aufpl
    pi_aplzl  = pi_aplzl
IMPORTING
    pe_elemmetal = pe_elemmetal.
ENDIF.

```

Ilustración 33: Función Z\_LB\_CARGAR\_METAL\_ELEM

Una vez dentro, simplemente carga de las tablas los registros almacenados, si no es la primera vez que entran (Ilustración 34 y 35).

Módulo funciones    Z\_LB\_CARGAR\_METAL    Activo

Atributos    Import    Export    Changing    Tablas    Excep.    Cód.fte.

```

10  * "-----
11  * TDD4773 - 20.05.2021
12  * Cargamos los datos de la tabla ztlb_metal
13  * "-----
14
15  SELECT *
16  FROM ztlb_metal
17  INTO TABLE pe_metal
18  WHERE idmuestra = pi_muestra
19  AND aufpl = pi_aufpl
20  AND aplzl = pi_aplzl.
21
22  ENDFUNCTION.

```

Ilustración 34: Carga los datos de la tabla ZTLB\_METAL

```
10  *"--
11  *   TDD4773 - 20.05.2021
12  *   Cargamos los datos de la tabla ztlb_elemmetal
13  *"--
14  SELECT * FROM ztlb_elemmetal
15         INTO CORRESPONDING FIELDS OF TABLE pe_elemmetal
16         WHERE idmuestra = pi_muestra
17         AND   aufpl = pi_aufpl
18         AND   aplzl = pi_aplzl
19         ORDER BY idelemento.
20
21  ENDFUNCTION.
```

Ilustración 35: Carga los datos de la tabla ZTLB\_ELEMMETAL

Para seguir con el flujo del programa, debemos crear los módulos que correspondan el PAI. El módulo *pai\_9001\_exit* es un módulo genérico que se usa en el programa donde se ha creado la pantalla y permite poder navegar para atrás. Luego, tenemos el Loop a la tabla de elementos que se encarga de obtener cualquier modificación realizada en las filas, así como, modificar, añadir o eliminar filas. Después, se obtienen las modificaciones realizadas en los campos de los datos del ensayo, ya que, si se modifica, se tendrá en cuenta para guardar estas modificaciones. Para acabar, tendremos el módulo que se encargará de actuar según la interacción del usuario, *user\_command\_9012*, como, por ejemplo, adjuntar el fichero o guardar los resultados del ensayo (Ilustración 36).



```

PROCESS AFTER INPUT.
  MODULE pai_9001_exit AT EXIT-COMMAND.

  *&SPWIZARD: PAI FLOW LOGIC FOR TABLECONTROL 'TC_ELEMMETAL'
  LOOP AT t_lemmetal.
    CHAIN.
      FIELD wa_telemmetal-idelemento.
      FIELD wa_telemmetal-elemento.
      FIELD wa_telemmetal-localizacion.
      FIELD wa_telemmetal-nomenclatura.
      FIELD wa_telemmetal-defectos.
      FIELD wa_telemmetal-nvlcalidad.
      FIELD wa_telemmetal-calificacionelem.
      FIELD wa_telemmetal-ctrlgeom.
      FIELD wa_telemmetal-observacion.
      MODULE tc_lemmetal_modify ON CHAIN-REQUEST.
    ENDCHAIN.
    FIELD wa_telemmetal-sel
      MODULE tc_lemmetal_mark ON REQUEST.
  ENDLOOP.
  MODULE tc_lemmetal_user_command.
  *&SPWIZARD: MODULE TC_ELEMMETAL_CHANGE_TC_ATTR.
  L *&SPWIZARD: MODULE TC_ELEMMETAL_CHANGE_COL_ATTR.

  CHAIN.
    FIELD wa_metal-tpenetracion.
    FIELD wa_metal-tambienteenc.
    FIELD wa_metal-tambientedeb.
    FIELD wa_metal-lotepenetrante.
    FIELD wa_metal-loteeliminador.
    FIELD wa_metal-loterevelador.
    FIELD ztlb_murelser-ffinaliza.
    FIELD ztlb_murelser-obser.
    FIELD ztlb_murelser-ensrealizado.
    MODULE modificado ON CHAIN-REQUEST.
  ENDCHAIN.

  MODULE user_command_9012.

```

Ilustración 36: PAI

La función encargada de guardar los datos de la pantalla está dividida a su vez por distintas subrutinas que se encargan de guardar los datos en distintas tablas. La primera subrutina se encarga de guardar los datos en las tablas *ZTLB\_METAL* y *ZTLB\_ELEMMETAL*. La segunda subrutina actualiza los datos en la tabla *ZTLB\_MUESTRAS* y la tercera subrutina actualiza los datos en la tabla *ZTLB\_MURELSER*. Estas dos últimas subrutinas son genéricas y se utilizan para todas las pantallas que están en este programa (Ilustración 37).

```

* TDD4773 - 20.05.2021
* Guardamos los datos en la tablas de BBDD
*-----*
FORM f_guardar_datos9012 USING p_finaliza.
*Grabamos datos

PERFORM f_guardar_res9012.

PERFORM f_actualiza_muestra.
PERFORM f_actualiza_servicios CHANGING p_finaliza.

CLEAR wg_modificado.
ENDFORM.

```

Ilustración 37: Función guardar datos 9012

La función que nos interesa es la que nos guarda los datos relacionados con nuestro ensayo. Se observa que solo se hará la modificación si ha habido alguna modificación en la pantalla, esto es controlado en el PAI. En caso de realizar alguna modificación entrará en la función `Z_LB_GUARDAR_METAL_ELEM` (Ilustración 38).

```

* TDD4773 - 20.05.2021
* Guardamos los datos en la tablas ZTLB_METAL y ZTLB_ELEMMETAL
*-----*
FORM f_guardar_res9012 .

IF wg_modificado IS NOT INITIAL.
wa_metal-idmuestra = wa_navegacion-idmuestra.
wa_metal-aufpl = wa_navegacion-aufpl.
wa_metal-aplz1 = wa_navegacion-aplz1.

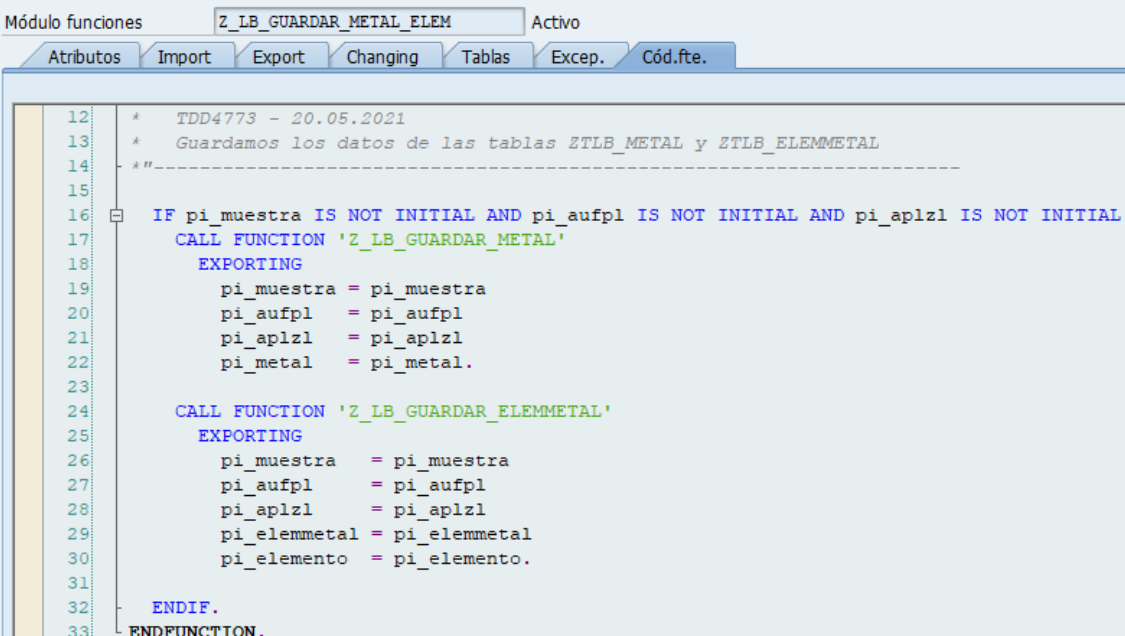
CALL FUNCTION 'Z_LB_GUARDAR_METAL_ELEM'
EXPORTING
pi_muestra = wa_navegacion-idmuestra
pi_aufpl = wa_navegacion-aufpl
pi_aplz1 = wa_navegacion-aplz1
pi_elemmetal = t_elemmetal
pi_elemento = wl_elemento
pi_metal = wa_metal.
ENDIF.
ENDFORM.

```

Ilustración 38: Guardar datos del ensayo

Una vez dentro, se llamará a dos funciones distintas para hacer el guardado de los datos insertados y la pantalla de resultados. Tenemos entonces la función `Z_LB_GUARDAR_METAL` que se encarga de guardar los datos en la tabla `ZTLB_METAL` y la función

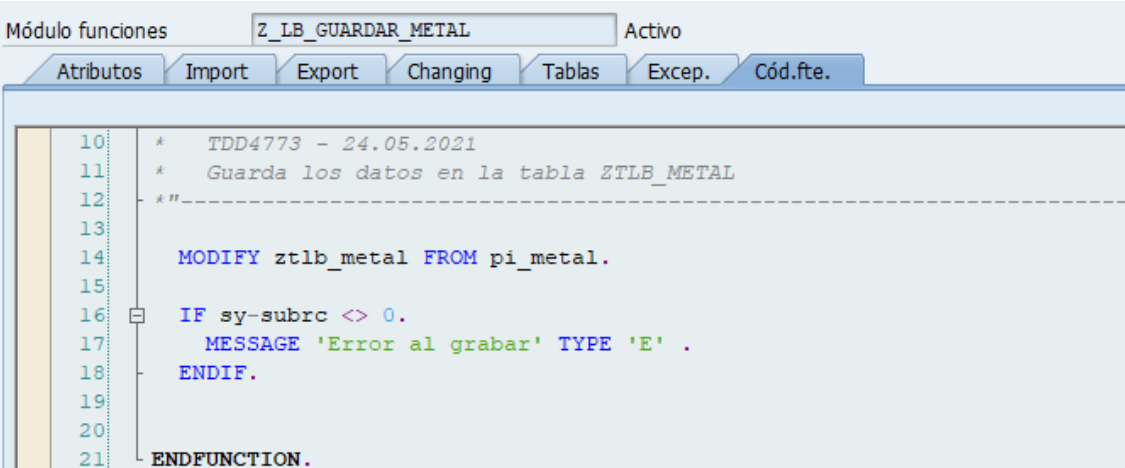
Z\_LB\_GUARDAR\_ELEMMETAL que se encarga de guardar los datos en la tabla ZTLB\_ELEMMETAL (Ilustración 39).



```
12 * TDD4773 - 20.05.2021
13 * Guardamos los datos de las tablas ZTLB_METAL y ZTLB_ELEMMETAL
14 *-----
15
16 IF pi_muestra IS NOT INITIAL AND pi_aufpl IS NOT INITIAL AND pi_aplzl IS NOT INITIAL.
17   CALL FUNCTION 'Z_LB_GUARDAR_METAL'
18     EXPORTING
19       pi_muestra = pi_muestra
20       pi_aufpl   = pi_aufpl
21       pi_aplzl  = pi_aplzl
22       pi_metal  = pi_metal.
23
24   CALL FUNCTION 'Z_LB_GUARDAR_ELEMMETAL'
25     EXPORTING
26       pi_muestra = pi_muestra
27       pi_aufpl   = pi_aufpl
28       pi_aplzl  = pi_aplzl
29       pi_elemmetal = pi_elemmetal
30       pi_elemento = pi_elemento.
31
32 ENDIF.
33 ENDFUNCTION.
```

Ilustración 39: Función de guardar Z\_LB\_GUARDAR\_METAL\_ELEM

Vemos la función Z\_LB\_GUARDAR METAL (Ilustración 40) que pasado el parámetro donde tenemos los datos del ensayo, lo guardamos en la tabla ZTLB\_METAL.



```
10 * TDD4773 - 24.05.2021
11 * Guarda los datos en la tabla ZTLB_METAL
12 *-----
13
14   MODIFY ztlb_metal FROM pi_metal.
15
16 IF sy-subrc <> 0.
17   MESSAGE 'Error al grabar' TYPE 'E' .
18 ENDIF.
19
20
21 ENDFUNCTION.
```

Ilustración 40: Guardamos los datos de la tabla ZTLB\_METAL

Para la función Z\_LB\_GUARDAR\_ELEMMETAL (Ilustración 41) es algo más complejo, ya que tenemos campos para controlar la auditoría de los elementos y, además, hay que controlar cuando se elimina alguna fila de la tabla de los Datos de Elementos.

```

1  FUNCTION z_lb_guardar_elemmetal.
2  ""
3  """"Interfase local
4  "" IMPORTING
5  "" REFERENCE(PI_MUESTRA) TYPE EE_IDMUESTRA
6  "" REFERENCE(PI_AUFPL) TYPE CO_AUFPL
7  "" REFERENCE(PI_APLEL) TYPE CO_APLEL
8  "" REFERENCE(PI_ELEMMETAL) TYPE ZTLB_ELEMMETAL
9  "" REFERENCE(PI_ELEMENTO) TYPE I
10 ""
11 * TDD4773 - 24.05.2021
12 * Guarda los datos en la tabla ZTLB_ELEMMETAL
13 ""
14 DATA: t1_elemmetal TYPE ztlb_elemmetal OCCURS 0 WITH HEADER LINE,
15        wa_telemmetal LIKE LINE OF pi_elemmetal.
16
17 CLEAR: wa_telemmetal.
18 "Guardaremos los datos de los elementos
19 LOOP AT pi_elemmetal INTO wa_telemmetal.
20 IF wa_telemmetal-ersda IS INITIAL.
21 "Creación.
22 wa_telemmetal-ersda = sy-datum.
23 wa_telemmetal-erzei = sy-uzeit.
24 wa_telemmetal-ernam = sy-uname.
25 ELSE.
26 "Modificación.
27 wa_telemmetal-laeda = sy-datum.
28 wa_telemmetal-aezet = sy-uzeit.
29 wa_telemmetal-aenam = sy-uname.
30 ENDIF.
31
32 MOVE-CORRESPONDING wa_telemmetal TO t1_elemmetal.
33 APPEND t1_elemmetal.
34 ENDLOOP.
35
36 IF pi_elemento IS NOT INITIAL.
37 DELETE FROM ztlb_elemmetal
38 WHERE idmuestra = pi_muestra
39 AND aufpl = pi_aufpl
40 AND aplzl = pi_aplzl.
41 ENDIF.
42
43 IF t1_elemmetal IS NOT INITIAL.
44 MODIFY ztlb_elemmetal FROM TABLE t1_elemmetal.
45
46 IF sy-subrc <> 0.
47 MESSAGE 'Error al grabar' TYPE 'E' .
48 ENDIF.
49 ENDIF.
50 ENDFUNCTION.

```

Ilustración 41: Guardamos los datos en la tabla ZTLB\_ELEMMETAL

Como algo adicional a añadir en la pantalla de la entrada de resultados, se ha añadido un botón por el cual se pueden añadir varias filas de Elementos del tirón, sin necesidad de estar añadiendo filas de uno en uno (Ilustración 42).

**Entrada de Resultados 7 (Materiales Metálicos)**

8002 - ACESOL INSPECCIÓN POR LÍQUIDOS PENETRANT

Datos de Cabecera

Expediente 0/1107639/1/01 (L1 PCS) LINEA 1 METRO MALAGA. TRAMO R Muestra 2543 R.L. 0201 1 2011 696  
 Obra 074619 (L1 PCS) LINEA 1 METRO MALAGA. TRAMO R Muestra O R.L. 0 0  
 Cliente 11538 FCC CONST.,SA,CONST.VERA,SA, CONST. Estado muestra FI Finalizada 1 de 1 TI ALB  
 Contratista Tipo material 78 F.Fin Muestra 21.09.2011  
 Observación  
 Fecha Inicio Fecha Fin 21.09.2011 Técnico Revisor F.Revisión Pedido 36453 / 21.09.2011  Ensayo realizado

Datos del Ensayo

Tiempo Penetración Lote Penetrante  
 T° Ambiente Encima °C Lote Eliminator  
 T° Ambiente Debajo °C Lote Revelador

Datos de Elementos

ID	Elemento	Localización	Nomenclatura	Defectos	Nivel de Calidad	Calif.	Ctrl. Geom.	Observaciones

Datos Auditoria

Creado el 26.04.2011 Creado por 3451 Última modif. 21.09.2011 Modificado por 3012

Ilustración 42: Añadir más filas

La implementación de este botón se realiza en el *User\_Command* del PAI, donde al pulsar el botón entrará en este caso y llamará a la siguiente subrutina para añadir el número de elementos insertados o, por su defecto, añadir 10 filas (Ilustración 43).

```

WHEN 'ELEMENTO'.
  PERFORM f_anadir_elementos_general USING '10' '2TLB_ELEMMETAL' 'tc_elemmetal' 't_elemmetal'.
WHEN OTHERS.
- ENDCASE.

```

Ilustración 43: Perform para añadir más filas

Esto ha sido la parte del Backend para la Implementación de nuestro TFG, pero la parte que realmente importa y es la que va a causar el avance para la empresa es la parte del Frontend, es decir, la aplicación de Fiori. Hemos señalado antes el proyecto en el OData y sus entidades, por lo que ahora veremos la funcionalidad del código en cuanto a recuperación de los datos y guardados de ellos.

Para la utilización de las entidades, hay que redefinir una serie de métodos que serán utilizados para la obtención (*GET*), crear (*CREATE*), eliminar (*DELETE*), actualización y guardado (*UPDATE*) (Ilustración 44).

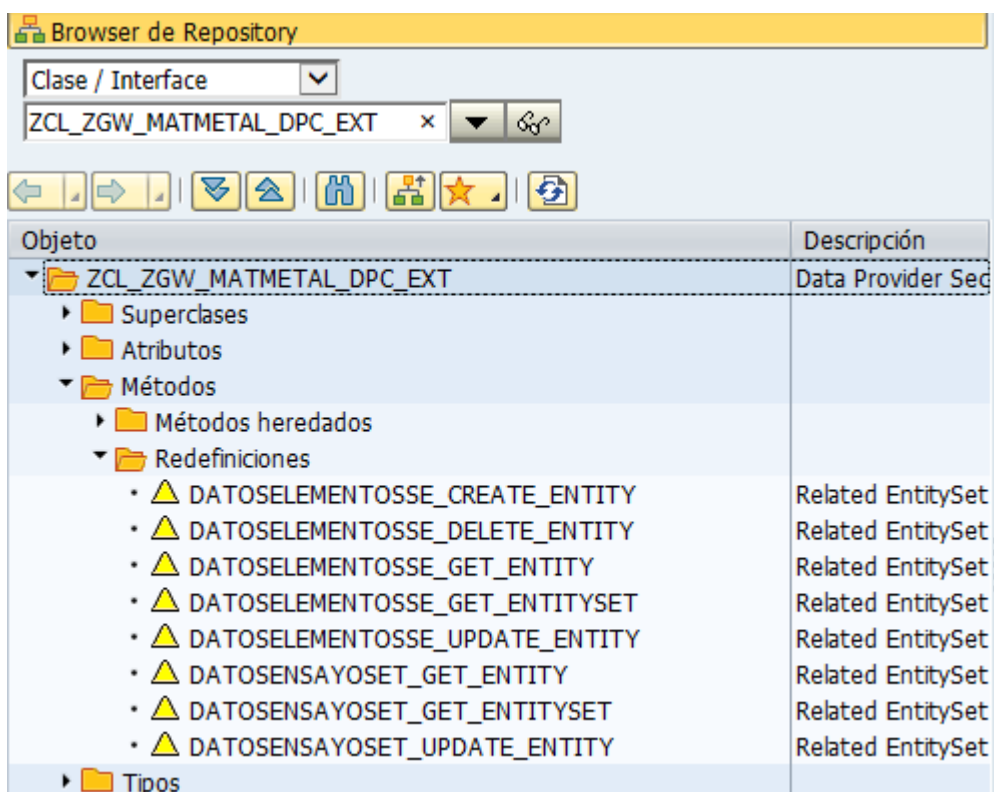


Ilustración 44: Métodos redefinidos del proyecto

A la hora de crear las entidades y que no de error para la llamada a esta entidad, hay que redefinir sobre todo el método *GET\_ENTITYSET*, pero para obtener solo un registro de la BBDD, es decir, para obtener los datos de una muestra que esté en la estructura de la entidad, habrá que redefinir el método *GET\_ENTITY*.

Vemos primero el *GET\_ENTITYSET* de la entidad de DatosEnsayos que contienen la estructura de *ZSLB\_METAL* (Ilustración 45). Para este tipo de métodos redefinidos hay dos maneras de pasar los parámetros, por clave o por filtros, para luego su extracción en variables que usaremos puestos que vienen en tablas y nos interesa que estén sueltos. Para ellos, al principio del método hay dos *performs* en el que uno se encarga de leer de la tabla de claves y otro de la tabla de filtros (Ilustración 46). Una vez obtenidos los valores en las variables para la obtención de los datos que haya en la BBDD, procederemos a llamar a las funciones ya usadas anteriormente en el Backend. En este caso, llamaremos a la función que carga el registro que haya en la tabla de *ZTLB\_METAL* para dicha muestra. Si existe, se mostrarán los datos que contenga y si no existe, es decir, es una muestra nueva, entonces

insertamos el identificador de la muestra en la variable de retorno para que se vea dicho valor en la pantalla de Fiori.

```

Método DATOSENSAYOSET_GET_ENTITYSET Actv.
1  METHOD datosensayoset_get_entityset.
2  CALL FUNCTION 'Z_LB_DATOSENSAYO'
3  EXPORTING
4      pi_key          = it_key_tab
5      pi_filtro       = it_filter_select_options
6  IMPORTING
7      pe_entity_set  = et_entityset.
8  ENDMETHOD.

Módulo funciones Z_LB_DATOSENSAYO Activo
Atributos Import Export Changing Tablas Excep. Cód.fte.

14  "-----
15  CLEAR: wg_idmuestra, wg_aufpl, wg_aplzl.
16
17  PERFORM f_carga_filtros_key USING pi_key.
18  PERFORM f_carga_filtros USING pi_filtro.
19  " IF wg_idmuestra IS NOT INITIAL AND wg_aufpl IS NOT INITIAL AND wg_aplzl IS NOT INITIAL.
20  CALL FUNCTION 'Z_LB_CARGAR_METAL'
21  EXPORTING
22      pi_muestra     = wg_idmuestra
23      pi_aufpl       = wg_aufpl
24      pi_aplzl       = wg_aplzl
25  IMPORTING
26      pe_metal       = pe_entity_set.
27
28  IF pe_entity_set IS NOT INITIAL.
29      READ TABLE pe_entity_set INTO pe_entity
30      WITH KEY idmuestra = wg_idmuestra
31             aufpl      = wg_aufpl
32             aplzl      = wg_aplzl.
33
34  ELSE. "Si es una muestra nueva, agregaremos un registro nuevo en pe_entity_set para que se muestre el idmuestra
35  Data: wa_metal type ZSLB_METAL.
36  wa_metal-idmuestra = wg_idmuestra.
37  wa_metal-aufpl     = wg_aufpl.
38  wa_metal-aplzl     = wg_aplzl.
39  APPEND wa_metal to pe_entity_set.
40
41  pe_entity = wa_metal.
42  ENDF.
43  ENDFUNCTION.

```

Ilustración 45: GET de DatosEnsayo

```

Include LZFGSLB_FRGR11F01 activo
72  FORM f_carga_filtros_key USING pi_key TYPE /iwbp/t_mgw_name_value_pair.
73
74  FIELD-SYMBOLS: <fs_key> LIKE LINE OF pi_key.
75  "Filtro Idmuestra.
76  READ TABLE pi_key
77  ASSIGNING <fs_key>
78  WITH KEY name = 'Idmuestra'.
79  IF sy-subrc = 0.
80      wg_idmuestra = <fs_key>-value.
81
82  CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
83  EXPORTING
84      input = wg_idmuestra
85  IMPORTING
86      output = wg_idmuestra.
87  ENDF.
88
89  "Filtro Aufpl.
90  READ TABLE pi_key
91  ASSIGNING <fs_key>
92  WITH KEY name = 'Aufpl'.
93  IF sy-subrc = 0.
94      wg_aufpl = <fs_key>-value.
95  ENDF.
96
97  "Filtro Aplzl.
98  READ TABLE pi_key
99  ASSIGNING <fs_key>
100 WITH KEY name = 'Aplzl'.
101 IF sy-subrc = 0.
102     wg_aplzl = <fs_key>-value.
103 ENDF.

Include LZFGSLB_FRGR11F01 activo
11  FORM f_carga_filtros USING pi_filtro TYPE /iwbp/t_mgw_select_option.
12
13  FIELD-SYMBOLS: <fs_filtro> LIKE LINE OF pi_filtro,
14                  <fs_select_options> TYPE /iwbp/s_cod_select_option.
15
16  "Filtro Idmuestra.
17  READ TABLE pi_filtro
18  ASSIGNING <fs_filtro>
19  WITH KEY property = 'Idmuestra'.
20  IF sy-subrc = 0.
21      LOOP AT <fs_filtro>-select_options ASSIGNING <fs_select_options>.
22          wg_idmuestra = <fs_select_options>-low.
23
24  CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
25  EXPORTING
26      input = wg_idmuestra
27  IMPORTING
28      output = wg_idmuestra.
29
30  ENDF.
31  ENDF.
32
33  "Filtro Aufpl.
34  READ TABLE pi_filtro
35  ASSIGNING <fs_filtro>
36  WITH KEY property = 'Aufpl'.
37  IF sy-subrc = 0.
38      LOOP AT <fs_filtro>-select_options ASSIGNING <fs_select_options>.
39          wg_aufpl = <fs_select_options>-low.
40
41  ENDF.
42  ENDF.

```

Ilustración 46: Recoger datos por clave o por filtros

Para esta entidad, tanto en el *GET\_ENTITYSET* como en el *GET\_ENTITY* se llama a la misma función ya que debe hacer lo mismo (Ilustración 47), pero para el caso del método de *GET\_ENTITY* no tiene como parámetro de entrada la opción de pasarles filtros por lo que solo es posible usarla pasando claves.

```

Método DATOSENSAYOSET_GET_ENTITY Actv.
1  METHOD datosensayoset_get_entity.
2      CALL FUNCTION 'Z_LB_DATOSENSAYO'
3      EXPORTING
4          pi_key      = it_key_tab
5      IMPORTING
6          pe_entity  = er_entity.
7  ENDMETHOD.

```

Ilustración 47: *GET\_ENTITY* de DatosEnsayo

Para el *GET\_ENTITYSET* de DatosElementos que contienen la estructura de *ZSLB\_ELEMMETAL* (Ilustración 48) se hacen los mismos pasos. Creamos la función a la que le pasamos las claves y los filtros para luego obtener como resultado un registro o varios registros que se recuperen de la tabla de BBDD. La función obtendrá los registros que existen en la tabla y se mostrarán en la pantalla de Fiori.

```

Método DATOSELEMENTOSSE_GET_ENTITYSET Actv.
1  METHOD datoselementosse_get_entityset.
2
3  DATA: wa_error          TYPE /iwbp/s_message_container,
4         go_message_container TYPE REF TO /iwbp/if_message_container.
5
6  CALL FUNCTION 'Z_LB_DATOSELEMENTOS_SET'
7  EXPORTING
8      pi_key      = it_key_tab
9      pi_filtro   = it_filter_select_options
10 IMPORTING
11     pe_entity_set = et_entityset
12     pe_error      = wa_error.
13
14
15 IF wa_error IS NOT INITIAL.
16     go_message_container = mo_context->get_message_container( )
17     go_message_container->add_message(
18         EXPORTING
19             iv_msg_type      = wa_error-type
20             iv_msg_id        = wa_error-id
21             iv_msg_number    = wa_error-number
22             iv_add_to_response_header = abap_true
23             iv_is_leading_message = abap_true
24         ).
25     RAISE EXCEPTION TYPE /iwbp/cx_mgw_busi_exception
26     EXPORTING
27         message_container = go_message_container.
28 ENDIF.
29
30 ENDMETHOD.

```

tódulo funciones

Z\_LB\_DATOSELEMENTOS\_SET Activo

Atributos Import Export Changing Tablas Excep. Cód.fte.

```

16 CLEAR: wg_idmuestra, wg_aufpl, wg_aplzl.
17
18 PERFORM f_carga_filtros_key USING pi_key.
19 PERFORM f_carga_filtros USING pi_filtro.
20
21 CALL FUNCTION 'Z_LB_CARGAR_ELEMMETAL'
22 EXPORTING
23     pi_muestra      = wg_idmuestra
24     pi_aufpl        = wg_aufpl
25     pi_aplzl        = wg_aplzl
26 IMPORTING
27     pe_elemmetal   = pe_entity_set.
28
29 IF pe_entity_set IS INITIAL.
30     pe_error-type = 'E'.
31     pe_error-id = 'ZLB'.
32     pe_error-number = '110'.
33 ENDIF.
34 ENDFUNCTION.

```

Ilustración 48: *GET\_ENTITYSET* de DatosElementos



A diferencia de la otra entidad, en esta entidad los métodos *GET\_ENTITYSET* y *GET\_ENTITY* son distintos, puesto que para esta entidad podemos especificar qué registro queremos visualizar (Ilustración 49).

```

Método DATOSELEMENTOSSE_GET_ENTITY Actv.
1  METHOD datoselementosse_get_entity.
2
3  DATA: wa_error          TYPE /iwbep/s_message_container,
4         go_message_container TYPE REF TO /iwbep/if_message_co
5
6  CALL FUNCTION 'Z_LB_DATOSELEMENTOS'
7  EXPORTING
8     pi_key      = it_key_tab
9  IMPORTING
10     pe_entity  = er_entity
11     pe_error   = wa_error.
12
13
14  IF wa_error IS NOT INITIAL.
15     go_message_container = mo_context->get_message_container(
16     go_message_container->add_message(
17         EXPORTING
18             iv_msg_type      = wa_error-type
19             iv_msg_id        = wa_error-id
20             iv_msg_number    = wa_error-number
21             iv_add_to_response_header = abap_true
22             iv_is_leading_message = abap_true
23         ).
24     RAISE EXCEPTION TYPE /iwbep/cx_mgw_busi_exception
25     EXPORTING
26         message_container = go_message_container.
27  ENDIF.
28
29  ENDMETHOD.

```

```

Módulo.funciones Z_LB_DATOSELEMENTOS Activo
Atributos Import Export Changing Tablas Excep. Cód.fte.
16  CLEAR: wg_idmuestra, wg_aufpl, wg_aplzl.
17
18
19  PERFORM f_carga_filtros_key USING pi_key.
20
21  CALL FUNCTION 'Z_LB_CARGAR_ELEMETAL'
22  EXPORTING
23     pi_muestra      = wg_idmuestra
24     pi_aufpl        = wg_aufpl
25     pi_aplzl        = wg_aplzl
26  IMPORTING
27     pe_elemmetal    = pe_entity_set.
28
29  IF pe_entity_set IS NOT INITIAL.
30     READ TABLE pe_entity_set INTO pe_entity
31     WITH KEY idmuestra = wg_idmuestra
32            aufpl      = wg_aufpl
33            aplzl      = wg_aplzl
34     idelemento = wg_idelemento.
35
36     IF sy-subrc <> 0.
37         pe_error-type = 'E'.
38         pe_error-id   = 'ZLB'.
39         pe_error-number = '110'.
40     ENDIF.
41
42  ENDFUNCTION.

```

Ilustración 49: *GET\_ENTITY* de DatosElementos

Una vez visto cómo obtenemos los datos, veremos cómo se pueden crear nuevas filas en la tabla de Elementos. Para ello, redefiniremos el método *CREATE* de la entidad de DatosElementos (Ilustración 50).

```

Método DATOSELEMENTOSSE_CREATE_ENTITY Actv.
1  METHOD datoselementosse_create_entity.
2
3  DATA: wa_input_data      TYPE zcl_zgw_matmetal_mpc=>ts_datoselementos.
4
5  CLEAR: wa_input_data.
6  "Obtiene los datos de la incidencia
7  io_data_provider->read_entry_data(
8  IMPORTING es_data = wa_input_data ).
9
10 CALL FUNCTION 'Z_LB_CREATE_ELEMENTOS'
11 EXPORTING
12     pi_input_data = wa_input_data
13 IMPORTING
14     pe_entity     = er_entity.
15
16 ENDMETHOD.

```

Ilustración 50: *CREATE* de DatosElementos

Vemos que en *wa\_input\_data* llegará una fila vacía con el tipo de la tabla de la entidad y se llamará a la función de crear. Una vez dentro, buscaremos el identificador del último elemento que esté guardado en la tabla de los elementos. En caso de obtener un identificador distinto a 0, sumaremos en uno el nuevo registro para añadirlo en la siguiente posiciones y en caso de ser igual a 0 pues le asignaremos el valor 1 para que se sepa que es la primera fila de elementos de esa muestra. Luego añadiremos los datos de auditoría para tener controlado cuando se crearon y cuando se modificaron por última vez. Para acabar, los añadimos a la tabla de los elementos (Ilustración 51).

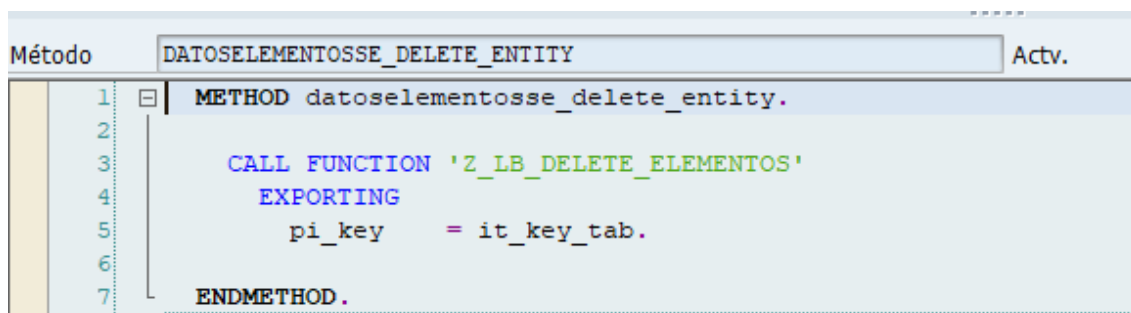
```

Módulo funciones      Z_LB_CREATE_ELEMENTOS      Activo
┌──────────┬──────────┬──────────┬──────────┬──────────┬──────────┬──────────┐
│ Atributos │ Import   │ Export   │ Changing │ Tablas   │ Excep.   │ Cód.fte.  │
└──────────┴──────────┴──────────┴──────────┴──────────┴──────────┴──────────┘
14  DATA: t1_elemento  TYPE ztlb_elemmetal OCCURS 0 WITH HEADER LINE,
15         wa_elemento  LIKE LINE OF t1_elemento,
16         wa_idelemento TYPE ze_idelementometal.
17
18  MOVE-CORRESPONDING pi_input_data TO wa_elemento.
19
20  SELECT MAX( idelemento ) FROM ztlb_elemmetal
21         INTO wa_idelemento
22         WHERE idmuestra = wa_elemento-idmuestra
23               AND aufpl = wa_elemento-aufpl
24               AND aplzl = wa_elemento-aplzl.
25
26  IF wa_idelemento >= 1.
27     wa_idelemento = wa_idelemento + 1.
28
29     SELECT * FROM ztlb_elemmetal
30            INTO TABLE t1_elemento
31            WHERE idmuestra = wa_elemento-idmuestra
32                  AND aufpl = wa_elemento-aufpl
33                  AND aplzl = wa_elemento-aplzl
34            ORDER BY idelemento.
35
36  ELSEIF wa_idelemento = 0. "Si es el primer registro
37     wa_idelemento = 1.
38  ENDIF.
39
40  wa_elemento-idelemento = wa_idelemento.
41  wa_elemento-ersda = sy-datum.
42  wa_elemento-erzei = sy-uzeit.
43  wa_elemento-ernam = sy-uname.
44  MOVE-CORRESPONDING wa_elemento TO t1_elemento.
45  APPEND t1_elemento.
46
47  MODIFY ztlb_elemmetal FROM TABLE t1_elemento.
48  COMMIT WORK.
49
50  MOVE-CORRESPONDING t1_elemento TO pe_entity.
51  ENDFUNCTION.

```

Ilustración 51: Función para crear filas en la tabla

Continuamos con la tabla de elementos y vemos como es el código para eliminar una fila de la tabla de elementos. Para ello se redefine el método *DELETE* de la misma entidad (Ilustración 52).



```
Método DATOSELEMENTOSSE_DELETE_ENTITY Actv.
1  METHOD datoselementosse_delete_entity.
2
3      CALL FUNCTION 'Z_LB_DELETE_ELEMENTOS'
4          EXPORTING
5              pi_key = it_key_tab.
6
7      ENDMETHOD.
```

Ilustración 52: *DELETE* de DatosElementos

Primero leemos las claves para saber que fila de la tabla de elementos se está queriendo eliminar. Luego, guardaremos en una variable local aquellas filas que no se vayan a borrar, modificando los identificadores de los elementos de las filas posteriores a la fila que se desea modificar para mantener el orden de las filas. Finalmente, se eliminan los registros de la tabla y se guardan las filas que se desean conservar (Ilustración 53).

```

Módulo funciones      Z_LB_DELETE_ELEMENTOS      Activo
Atributos  Import  Export  Changing  Tablas  Excep.  Cód.fte.
10      DATA: t1_elemmetal TYPE  ztlb_elemmetal OCCURS 0 WITH HEADER LINE,
11              cont      TYPE  i,
12              wa_elemmetal LIKE LINE OF t1_elemmetal,
13              t_elemmetal TYPE  ztlb_elemmetal OCCURS 0 WITH HEADER LINE.
14
15      PERFORM f_carga_filtros_key USING pi_key.
16
17      SELECT * FROM ztlb_elemmetal
18      INTO TABLE t1_elemmetal
19      WHERE idmuestra = wg_idmuestra
20      AND aufpl = wg_aufpl
21      AND aplz1 = wg_aplz1
22      ORDER BY idelemento.
23
24      LOOP AT t1_elemmetal INTO wa_elemmetal.
25      IF wa_elemmetal-idelemento <> wg_idelemento.
26      cont = cont + 1.
27      wa_elemmetal-idelemento = cont.
28      MOVE-CORRESPONDING wa_elemmetal TO t_elemmetal.
29      APPEND t_elemmetal.
30      ENDIF.
31      ENDLLOOP.
32
33      DELETE FROM ztlb_elemmetal
34      WHERE idmuestra = wg_idmuestra
35      AND aufpl = wg_aufpl
36      AND aplz1 = wg_aplz1.
37
38      MODIFY ztlb_elemmetal FROM TABLE t_elemmetal.
39      COMMIT WORK.
40

```

Ilustración 53: Función para eliminar filas en la tabla

Para acabar la parte de OData, nos queda por ver cómo guardar los datos que se inserten en la pantalla de Fiori y, para ello, se redefinirá el método *UPDATE* de ambas entidades.

Para el guardado de la entidad de DatosEnsayo, se pasará a nuestra función de guardado el registro de la tabla en *wa\_input\_data* (Ilustración 54).

```

Método DATOSENSAYOSET_UPDATE_ENTITY Actv.
1  METHOD datosensayoset_update_entity.
2
3  DATA: wa_input_data      TYPE zcl_zgw_matmetal_mpc=>ts_datosensayo,
4         wa_error          TYPE /iwbep/s_message_container,
5         go_message_container TYPE REF TO /iwbep/if_message_container.
6
7  CLEAR: wa_input_data.
8  "Obtiene los datos de la incidencia
9  io_data_provider->read_entry_data(
10     IMPORTING es_data = wa_input_data ).
11
12  CALL FUNCTION 'Z_LB_UPDATE_ENSAYO'
13     EXPORTING
14     pi_input_data = wa_input_data
15     IMPORTING
16     pe_entity    = er_entity
17     pe_error     = wa_error.
18
19  IF wa_error IS NOT INITIAL.
20     go_message_container = mo_context->get_message_container( ).
21     go_message_container->add_message(
22         EXPORTING
23         iv_msg_type      = wa_error-type
24         iv_msg_id        = wa_error-id
25         iv_msg_number    = wa_error-number
26         iv_add_to_response_header = abap_true           " Flag is set to show msgs on header
27         iv_is_leading_message = abap_true
28         ).
29     RAISE EXCEPTION TYPE /iwbep/cx_mgw_busi_exception
30     EXPORTING
31     message_container = go_message_container.
32  ENDIF.

```

Ilustración 54: UPDATE de DatosEnsayo

Una vez dentro, simplemente se modifica o se guarda el registro en la tabla (Ilustración 55).

```

Módulo funciones Z_LB_UPDATE_ENSAYO Activo
Atributos Import Export Changing Tablas Excep. Cód.fte.
13  * "-----
14
15  DATA: wa_metal TYPE ztlb_metal.
16
17  MOVE-CORRESPONDING pi_input_data TO wa_metal.
18
19  MODIFY ztlb_metal FROM wa_metal.
20
21  IF sy-subrc <> 0.
22     pe_error-type = 'E'.
23     pe_error-id = 'ZLB'.
24     pe_error-number = '253'.
25     EXIT.
26  ENDIF.

```

Ilustración 55: Guarda datos en la tabla ZTLB\_METAL

En cuanto al guardar de la entidad de DatosElementos, es igual a la entidad de DatosEnsayo con la diferencia del método al que se llama (Ilustración 56).

```
Método DATOSELEMENTOSSE_UPDATE_ENTITY Actv.
1 | METHOD datoselementosse_update_entity.
2 |
3 | DATA: wa_input_data      TYPE zcl_zgw_matmetal_mpc=>ts_datoselementos,
4 |       wa_error           TYPE /iwbep/s_message_container,
5 |       go_message_container TYPE REF TO /iwbep/if_message_container.
6 |
7 | CLEAR: wa_input_data.
8 | "Obtiene los datos de la incidencia
9 | io_data_provider->read_entry_data(
10 | IMPORTING es_data = wa_input_data ).
11 |
12 | CALL FUNCTION 'Z_LB_UPDATE_ELEMENTOS'
13 | EXPORTING
14 |   pi_input_data = wa_input_data
15 | IMPORTING
16 |   pe_entity      = er_entity
17 |   pe_error       = wa_error.
18 |
19 | IF wa_error IS NOT INITIAL.
20 |   go_message_container = mo_context->get_message_container( ).
21 |   go_message_container->add_message(
22 |     EXPORTING
23 |       iv_msg_type      = wa_error-type
24 |       iv_msg_id        = wa_error-id
25 |       iv_msg_number    = wa_error-number
26 |       iv_add_to_response_header = abap_true           " Flag is set to show msgs on header
27 |       iv_is_leading_message = abap_true
28 |     ).
29 |   RAISE EXCEPTION TYPE /iwbep/cx_mgw_busi_exception
30 |   EXPORTING
31 |     message_container = go_message_container.
32 | ENDIF.
```

Ilustración 56: UPDATE de DatosElementos

A este método se entrará por cada fila que tenga la tabla y se recuperará dicha fila de la tabla de BBDD para luego modificar los datos de auditoría. Luego se actualizarán los datos en la tabla (Ilustración 57).

Módulo funciones      Z\_LB\_UPDATE\_ELEMENTOS      Activo

Atributos    Import    Export    Changing    Tablas    Excep.    Cód.fte.

```

16    DATA: wa_elemento      TYPE ztlb_elemmetal,
17           wa_elemento_ant TYPE ztlb_elemmetal.
18
19    MOVE-CORRESPONDING pi_input_data TO wa_elemento.
20
21    SELECT SINGLE * INTO wa_elemento_ant
22        FROM ztlb_elemmetal
23        WHERE idmuestra = wa_elemento-idmuestra
24              AND aufpl = wa_elemento-aufpl
25              AND aplz1 = wa_elemento-aplz1
26              AND idelemento = wa_elemento-idelemento.
27
28    wa_elemento-ersda = wa_elemento_ant-ersda.
29    wa_elemento-erzei = wa_elemento_ant-erzei.
30    wa_elemento-ernam = wa_elemento_ant-ernam.
31    wa_elemento-laeda = sy-datum.
32    wa_elemento-aezet = sy-uzzeit.
33    wa_elemento-aenam = sy-uname.
34
35    "MODIFY ztlb_elemmetal FROM wa_elemento.
36    UPDATE ztlb_elemmetal SET elemento        = wa_elemento-elemento
37                            localizacion      = wa_elemento-localizacion
38                            nomenclatura      = wa_elemento-nomenclatura
39                            defectos          = wa_elemento-defectos
40                            nvlocalidad       = wa_elemento-nvlocalidad
41                            calificacionelem = wa_elemento-calificacionelem
42                            ctrlgeom         = wa_elemento-ctrlgeom
43                            observacion     = wa_elemento-observacion
44                            ersda            = wa_elemento-ersda
45                            erzei           = wa_elemento-erzei
46                            ernam           = wa_elemento-ernam
47                            laeda           = wa_elemento-laeda
48                            aezet          = wa_elemento-aezet
49                            aenam          = wa_elemento-aenam
50                            WHERE idmuestra = wa_elemento-idmuestra
51                            AND aufpl       = wa_elemento-aufpl
52                            AND aplz1       = wa_elemento-aplz1
53                            AND idelemento = wa_elemento-idelemento.
54
55    IF sy-subrc <> 0.
56        pe_error-type = 'E'.
57        pe_error-id = 'ZLB'.
58        pe_error-number = '253'.
59        EXIT.
60    ENDIF.
61

```

Ilustración 57: Guarda datos en la tabla ZTLB\_ELEMMETAL

Vemos ahora la parte de la implementación que corresponde a Fiori, esto se hace mediante el lenguaje SAPUI5.

Para poder navegar a la pantalla creada, primero se añadirá un botón de navegación en la aplicación ya existe. Este botón aparecerá para todas las muestras, pero solo se podrá

navegar dependiendo del tipo de material y del tipo de ensayo que se haya elegido (Ilustración 58).

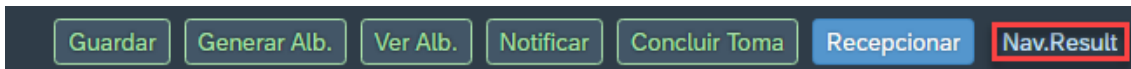


Ilustración 58: Botón navega Entrada de Resultados

La implementación de dicho botón se hace en el controlador de esta vista y para que funcione este botón se ha creado la siguiente función que navega dependiendo de los datos indicados antes (Ilustración 59).

```
navToRes: function (oEvent) {
    var oView = this.getView();
    var oData = oView.byId("DatosMuestrasOtros").getModel("Tomamu2").getData(oEvent.getSource().getParent().getBindingContextPath())[0];
    var oCrossAppNavigator = sap.usHELL.Container.getService("CrossApplicationNavigation");
    var hash;
    if (oData.Idtipomaterial === "078" && oData.Idtipoensayo === "8002") {
        hash = (oCrossAppNavigator && oCrossAppNavigator.hrefForExternal({
            target: {
                semanticObject: "ZSQMM",
                action: "display"
            },
            params: {
                "Idmuestra": oData.Idmuestra,
                "Aufpl": oData.Aufpl,
                "Ap1z1": oData.Ap1z1
            }
        ))) || "";
        oCrossAppNavigator.toExternal({
            target: {
                shellHash: hash
            }
        });
    }
    } else if (oData.Idtipomaterial === "001" && oData.Idtipoensayo === "9066") {
        hash = (oCrossAppNavigator && oCrossAppNavigator.hrefForExternal({
            target: {
                semanticObject: "ZSOPC",
                action: "display"
            },
            params: {
                "Idmuestra": oData.Idmuestra,
                "Aufpl": oData.Aufpl,
                "Ap1z1": oData.Ap1z1
            }
        ))) || "";
        oCrossAppNavigator.toExternal({
            target: {
                shellHash: hash
            }
        });
    }
}
}
```

Ilustración 59: Función para navegar a la Entrada de Resultados

Para nuestra aplicación se crea un proyecto (Ilustración 60) que contendrá la vista y el controlador. El modelo ya lo hemos definido antes como DatosEnsayo y DatosElementos.



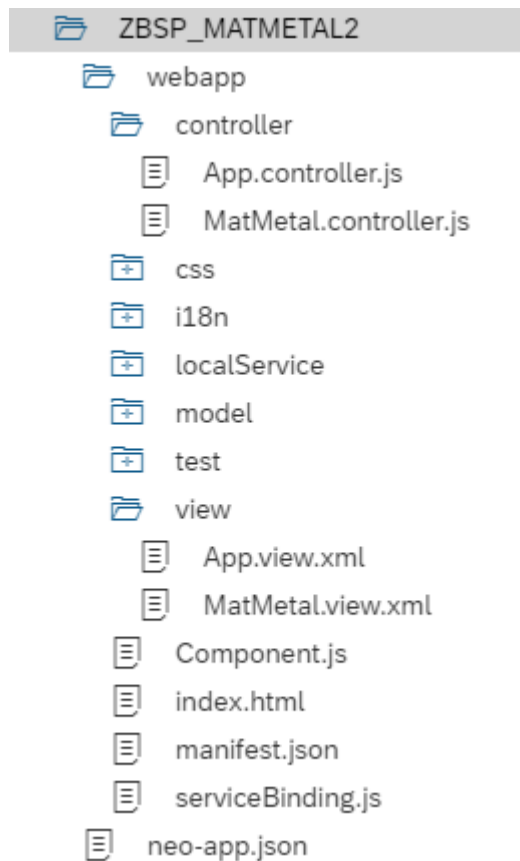


Ilustración 60: Proyecto para Fiori

En la vista, programaremos el diseño que tendrá nuestra aplicación en lenguaje *XML*. Destacamos la creación de la tabla en la que por una parte hay que especificar las columnas y por otra parte hay que especificar qué valores insertar en cada columna (Ilustración 61).

```

<Table id="tableElementos" items="{EnsayoElementosNav}" mode="SingleSelectLeft">
  <columns>
    <Column minScreenWidth="Tablet" demandPopin="true">
      <Text text="Elemento" class="customLetraMenos"/>
    </Column>
    <Column minScreenWidth="Tablet" demandPopin="true">
      <Text text="Localización" class="customLetraMenos"/>
    </Column>
    <Column minScreenWidth="Tablet" demandPopin="true">
      <Text text="Nomenclatura" class="customLetraMenos"/>
    </Column>
    <Column minScreenWidth="Tablet" demandPopin="true">
      <Text text="Defectos" class="customLetraMenos"/>
    </Column>
    <Column minScreenWidth="Tablet" demandPopin="true">
      <Text text="Nivel de Calidad" class="customLetraMenos"/>
    </Column>
    <Column minScreenWidth="Tablet" demandPopin="true">
      <Text text="Calificación" class="customLetraMenos"/>
    </Column>
    <Column minScreenWidth="Tablet" demandPopin="true">
      <Text text="Control Geométrico" class="customLetraMenos"/>
    </Column>
    <Column minScreenWidth="Tablet" demandPopin="true" width="15em">
      <Text text="Observaciones" class="customLetraMenos"/>
    </Column>
    <Column minScreenWidth="Phone" demandPopin="false"></Column>
  </columns>
  <items>
    <ColumnListItem>
      <cells>
        <Input value="{Elemento}" type="{Text}" maxLength="100"/>
        <Input value="{Localizacion}" type="{Text}" maxLength="70"/>
        <Input value="{Nomenclatura}" type="{Text}" maxLength="12"/>
        <Input value="{Defectos}" type="{Text}" maxLength="35"/>
        <Input id="NvlCalidad" value="{Nvlcalidad}" type="{Text}" maxLength="2"/>
        <Input id="Calif" value="{Calificacionelem}" type="{Text}" maxLength="7"/>
        <Input value="{Ctrlgeom}" type="{Text}" maxLength="9"/>
        <Input value="{Observacion}" type="{Text}" maxLength="200"/>
      </cells>
    </ColumnListItem>
  </items>
</Table>

```

Ilustración 61: Código para la vista de la pantalla

Y luego en el controlador de la vista, definiremos los métodos que se encargarán de recuperar la interacción que mantenga el usuario en la aplicación para posteriormente llamar al Backend y tratar los datos.

Cuando se navega a nuestra pantalla, se llama a la función *onInit* que se encarga de llamar a los *GET* de nuestras dos entidades para obtener los datos de la muestra. Si es una muestra nueva solo obtendrá el identificador de la muestra, pero si es una muestra que ya estuviese guardada, obtendrá todos los datos que estén en las tablas y se mostrarán en la pantalla de Fiori en sus respectivos sitios (Ilustración 62).

```
onInit: function () {
    window.deleteElemento = 0;
    var oRouter = this.getOwnerComponent().getRouter();
    oRouter.getRoute("MatMetal").attachPatternMatched(this._onMatMetalMatched, this);
},

_onMatMetalMatched: function () {
    var startupParams = this.getOwnerComponent().getComponentData().startupParameters;

    if (startupParams.Idmuestra && startupParams.Aufpl && startupParams.Aplzl) {
        this.Idmuestra = startupParams.Idmuestra[0];
        this.Aufpl = startupParams.Aufpl[0];
        this.Aplzl = startupParams.Aplzl[0];

        var sPath = "/DatosEnsayoSet(Idmuestra=" + encodeURIComponent(this.Idmuestra) + ",Aufpl=" + encodeURIComponent(this.Aufpl) +
            ",Aplzl=" + encodeURIComponent(this.Aplzl) + ")";
        this.getView().bindElement({
            path: sPath
        });
    }
},
```

Ilustración 62: Función para cargar los datos en la pantalla

Para controlar los botones que el usuario puede pulsar en la pantalla, se crea una función que actúa cuando se acciona dicho botón.

En primer lugar, vemos el botón de añadir elementos. Este botón está en la parte superior derecha de la tabla y representado con el símbolo de la suma (+). Cuando se acciona, se llama a esta función (Ilustración 63) que crea un registro vacío de la estructura de la tabla de elementos con las claves de la entidad para que después se llame a la función redefinida *CREATE*, donde hay que especificar el método a usar, en este caso *POST*.

```

addElemento: function (oEvent) {
    var oDataEnsayo = this.getView().getModel().getData(this.getView().getBindingContext().getPath());
    var oElemento = {
        "Idmuestra": oDataEnsayo.Idmuestra,
        "Aufpl": oDataEnsayo.Aufpl,
        "Aplz1": oDataEnsayo.Aplz1,
        "Idelemento": "",
        "Elemento": "",
        "Localizacion": "",
        "Nomenclatura": "",
        "Defectos": "",
        "Nvlcalidad": "",
        "Calificacionelem": "",
        "Ctrlgeom": "",
        "Observacion": ""
    };

    this.getView().getModel().create("/DatosElementosSet", oElemento, {
        method: "POST",
        success: function (oData, oResponse) {},
        error: function (oError) {}
    });
},

```

Ilustración 63: Función para añadir filas

En segundo lugar, tenemos el botón de eliminar una fila que estará a la derecha del botón de añadir una fila y se representará con el símbolo de la papelera. Esta función (Ilustración 64) la controlaremos más, ya que obligaremos al usuario a que guarde primero y luego poder darle la opción de borrar la fila que haya seleccionado. Este control lo realizamos como una medida preventiva por si acaso el usuario le da a borrar sin tener intención de borrar. Cuando haya guardado y le da a borrar, se llamará a la función redefinida *DELETE*, donde especificamos el método que usamos, en este caso *DELETE*.

```

deleteElemento: function (oEvent) {
    var oTabla = this.getView().byId("tableElementos");
    var aIndices = oTabla.getSelectedContexts();

    if (window.deleteElemento === 1) {
        var sPath = aIndices[0].sPath;

        this.getView().getModel().remove(sPath, {
            method: "DELETE",
            success: function (oData, oResponse) {},
            error: function (oError) {}
        });
        window.deleteElemento = 0;
        oTabla.removeSelections();
    } else {
        MessageToast.show("Guarde antes de borrar");
    }
},

```

Ilustración 64: Función para eliminar filas

Por último, vemos la función de guardar (Ilustración 65) que será llamada cuando accionamos el botón de Guardar que aparece en la parte inferior derecha de la pantalla. Primero se llamará, uno por uno tantas veces como filas haya en la tabla de elementos, a la función redefinida *UPDATE* de la entidad de DatosElementos usando el método *PUT*. Luego se llamará a la función redefinida *UPDATE* de la entidad de DatosEnsayo usando el mismo método *PUT* para guardar los datos que correspondan a esta entidad.

```

guardarDatos: function (oEvent) {
    var oView = this.getView();
    var oModel = this.getView().getModel();
    //Se guardan los Datos de la tabla de Elementos
    var oTabla = this.getView().byId("tableElementos");
    var aIndices = oTabla.getItems();

    for (var i = 0; i < aIndices.length; i++) {
        var sPath = oTabla.getItems()[i].getBindingContextPath();
        var oDataElementos = oModel.getData(oTabla.getBinding("items").getContexts()[i].sPath);

        oModel.update(sPath, oDataElementos, {
            method: "PUT",
            success: function (oData, oResponse) {},
            error: function (oError) {
                MessageToast.show("Error al guardar");
            }
        });
    }

    //Se guardan los Datos del Ensayo
    var oDataEnsayo = oModel.getData(oView.getBindingContext().getPath());
    var urlEnsayo = "/DatosEnsayoSet(Idmuestra=" + encodeURIComponent(oDataEnsayo.Idmuestra) + ",Aufpl=" +
        encodeURIComponent(oDataEnsayo.Aufpl) + ",Aplz1=" + encodeURIComponent(oDataEnsayo.Aplz1) + ")";

    oModel.update(urlEnsayo, oDataEnsayo, {
        method: "PUT",
        success: function (oData, oResponse) {
            MessageToast.show("Datos guardados correctamente");
            window.deleteElemento = 1;
        },
        error: function (oError) {
            MessageToast.show("Error al guardar");
        }
    });
},
},

```

Ilustración 65: Función para guardar

Para acabar con la fase de Implementación, se ha realizado un informe en formato PDF para visualizar el acta de la muestra de la que se ha realizado el ensayo. Para crear este informe se hace uso de los Adobe Forms. Esta es la tecnología más reciente para la creación de formularios en SAP. Nos permite crear formularios interactivos basados en Adobe PDF que optimizan los procesos de impresión de las aplicaciones.

Un formulario de Adobe Forms está compuesto por tres partes, la interfaz, el formulario y el código implementado para obtener los datos. La interfaz y el formulario se pueden crear y modificar desde la transacción *SFP* (Ilustración 66).

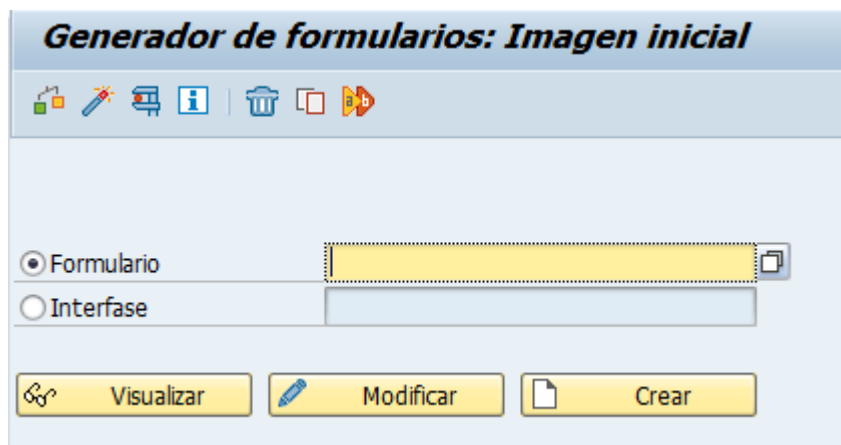


Ilustración 66: Transacción para crear plantilla Adobe

En la interfaz (Ilustración 67) especificaremos la estructura que se utilizará en el formulario.

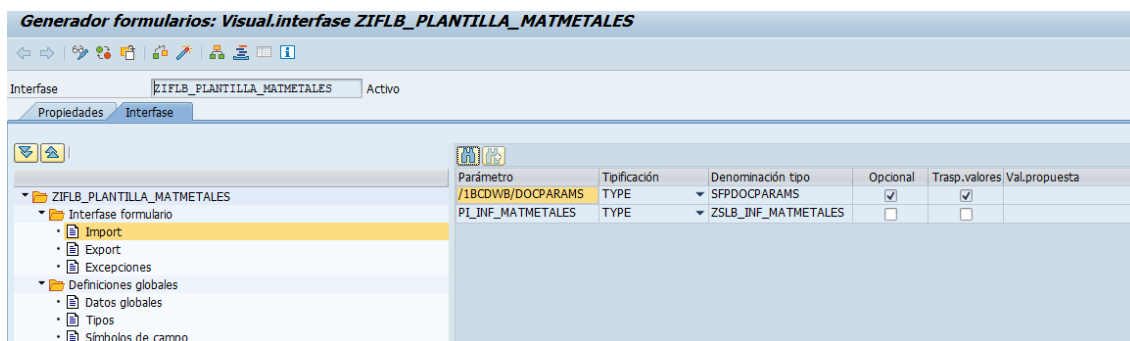


Ilustración 67: Interfaz para la plantilla Adobe

En el formulario (Ilustración 68) incluiremos esta interfaz para usar los campos que componen la estructura.

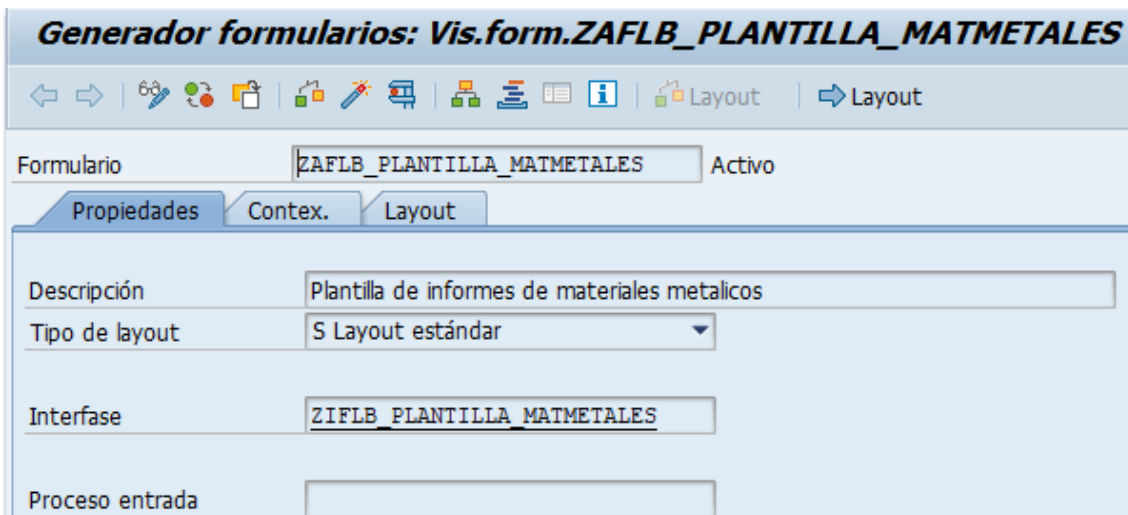


Ilustración 68: Formulario de la plantilla Adobe

Una vez incluida la interfaz, insertamos la estructura que usará la plantilla (Ilustración 69).

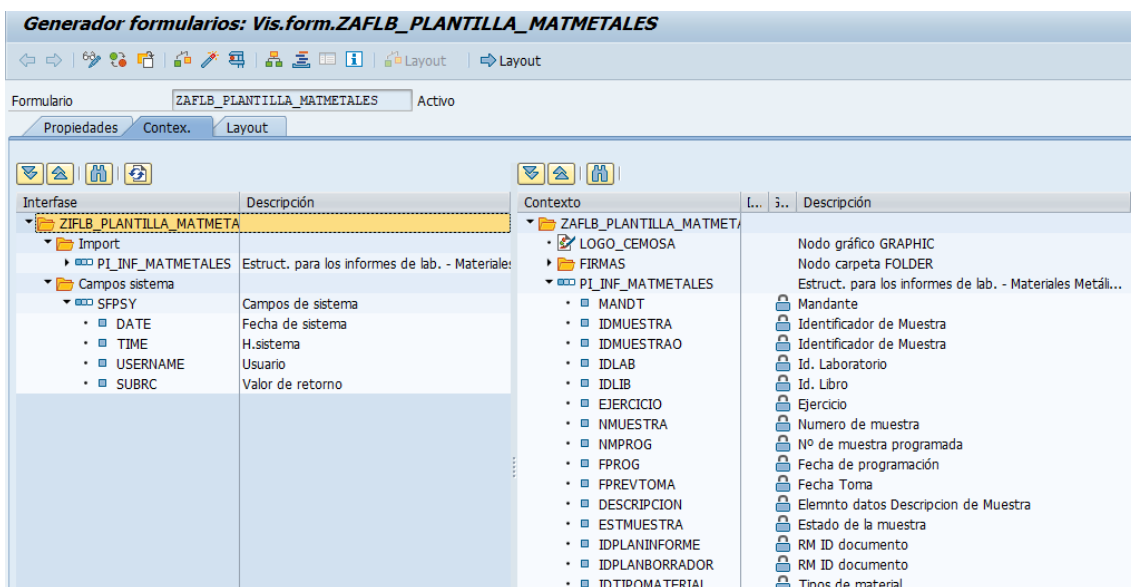


Ilustración 69: Contexto de la plantilla Adobe

Luego, para hacer el diseño del informe, hay que ir a la pestaña de Layout. Aquí es donde se añaden los campos, las tablas y sus enlaces para que cuando la función de la plantilla llame a este formulario, los campos que haya en él tengan los enlaces correctos para que se vea el resultado (Ilustración 70).



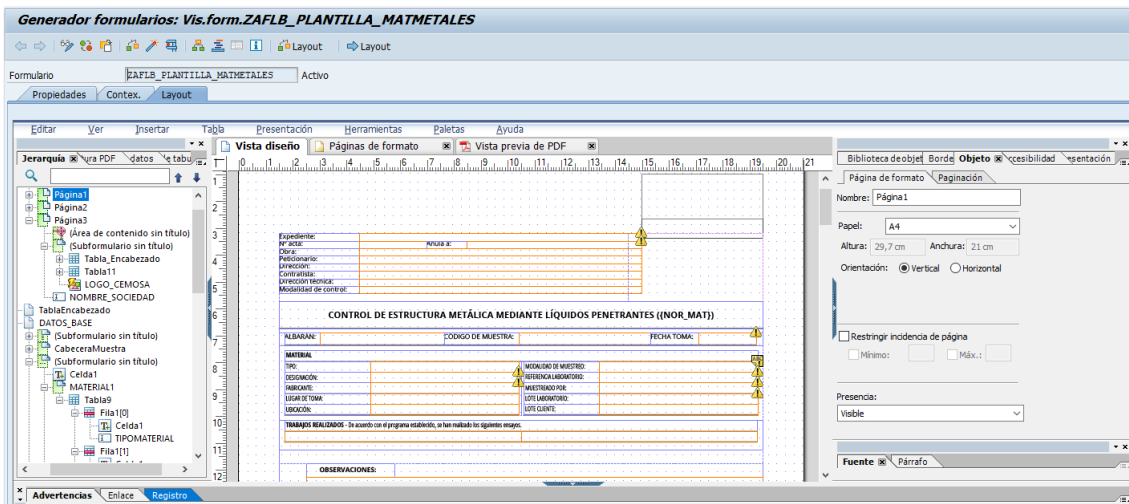


Ilustración 70: Vista de la plantilla Adobe

En cuanto a la implementación del código, se crea una función llamada `Z_LB_PLANTILLA_MATMETALES` que se encarga de pasar todos los datos que sean necesarios a la plantilla. Esta función está ubicada en el grupo de funciones `ZFGLB_DATOSPLANTILLAS2` en el que ya hay contenida otras funciones que recuperan los datos para sus respectivas plantillas.

Lo importante de esta función es la obtención de los datos del ensayo (Ilustración 71). Este tipo de funciones no reciben todas las claves primarias, por lo que primero hay que recuperar las claves primarias y posteriormente se pueden usar las funciones ya creadas anteriormente que nos sirven para cargar los datos del ensayo.

```
Módulo funciones      Z_LB_PLANTILLA_MATMETALES      Activo
Atributos  Import  Export  Changing  Tablas  Excep.  Cód.fte.
67      MOVE-CORRESPONDING pe_datos_plantillas TO pe_inf_matmetales.
68
69      SELECT SINGLE  aufpl  aplzl  FROM ztlb_murelser
70      INTO ( wl_aufpl , wl_aplzl )
71      WHERE idmuestra = pi_muestra.
72
73
74      CALL FUNCTION 'Z_LB_CARGAR_METAL'
75      EXPORTING
76          pi_muestra = pi_muestra
77          pi_aufpl   = wl_aufpl
78          pi_aplzl   = wl_aplzl
79      IMPORTING
80          pe_metal   = pe_inf_matmetales-metal.
81
82      CALL FUNCTION 'Z_LB_CARGAR_ELEMMETAL'
83      EXPORTING
84          pi_muestra = pi_muestra
85          pi_aufpl   = wl_aufpl
86          pi_aplzl   = wl_aplzl
87      IMPORTING
88          pe_elemmetal = pe_inf_matmetales-elemmetal.
89
90
```

Ilustración 71: Función para la plantilla

Una vez realizada la plantilla que usará el informe, hay que configurar las tablas que recogen dicha plantilla para que puedan ser asignadas a la muestra y poder visualizar dicho informe. Estos informes se asignan en la pantalla de la entrada de la muestra (Ilustración 72).

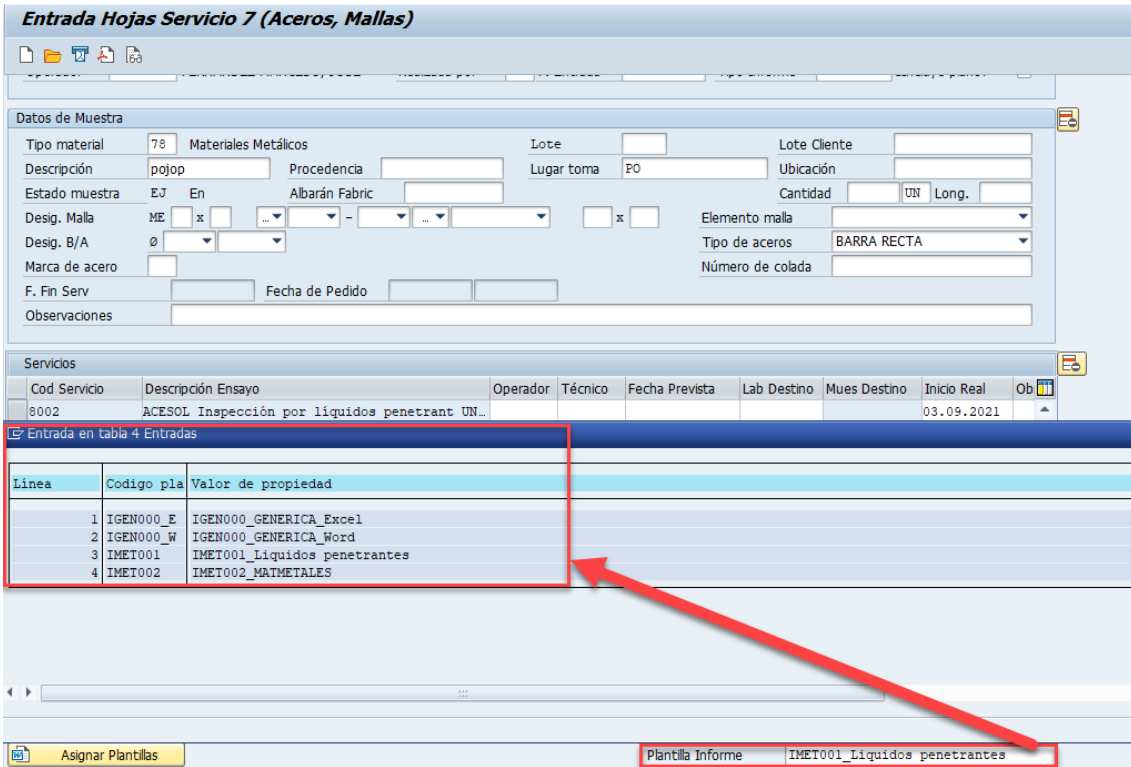


Ilustración 72: Elección de la plantilla Adobe

Para que un informe aparezca en esta ayuda de búsqueda para asignar la plantilla, debemos configurar las tablas *ZTLB\_PLANTILLAS*, *ZTLB\_SERVNORMADY* y *ZTLB\_SOCPLANT* (Ilustración 73).

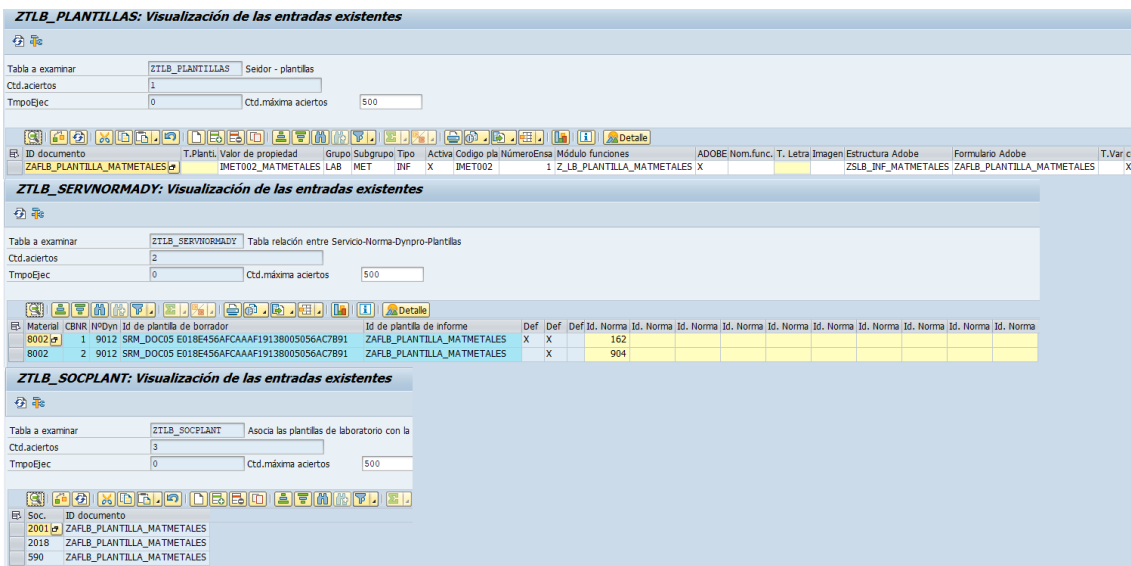


Ilustración 73: Registros añadidos en las tablas para usar la plantilla Adobe

Configurado esto, ya estaría completa la parte de la implementación del proyecto.



# 4

## Pruebas

Especificaremos ahora las distintas pruebas que se han ido realizando durante todo el TFG. El código, tanto del Backend como del Frontend, ha sido realizado totalmente en el entorno de Desarrollo, creando muestras falsas para probar las pantallas y su correcto funcionamiento.

Veamos primero las pruebas que realizamos en el Backend, ya que empezamos desde aquí el proyecto. Una vez que tenemos ya la pantalla creada, obtenemos una muestra que contenga el ensayo de Líquidos Penetrantes y desde la transacción *ZLER* podemos navegar a dicho ensayo introduciendo el número de la muestra (Ilustración 74) y pulsando el botón o pulsando la techa *f8*.

### Entrada de Resultados (Materiales).

**Idmuestra**

Id. Muestra

Servicio

**Referencia de laboratorio**

Laboratorio

Id. Libro

Ejercicio

N.Muestra

Ilustración 74: Transacción para la Entrada de Resultados

Una vez dentro en la Entrada de Resultados podemos rellenar los campos con datos reales o, en este caso, pueden ser ficticios y, además, añadir filas de elementos (Ilustración 75).

**Entrada de Resultados 7 (Materiales Metálicos)**

---

8002 - ACESOL Inspección por líquidos penetrant UNE 14612:1980

**Datos de Cabecera**

Expediente	0/2000025/1/01	CONTROL DE CALIDAD	Muestra	755916	R.L.	0201	1	2021	46
Obra	099993	CONTROL DE CALIDAD VIVID. UNIF. (CASA	Muestra O		R.L.		0		0
Cliente	6	CENTRO DE ESTUDIOS DE MATERIALES Y	Estado muestra	EJ	En ejecución	1	de	1	TI MUE
Contratista			Tipo material	78	F.Fin Muestra				

Observación

Fecha Inicio 03.09.2021 Fecha Fin  Técnico Revisor  F.Revisión  Pedido  /   Ensayo realizado

**Datos del Ensayo**

Tiempo Penetración	20	Lote Penetrante	68C/1044A
Tº Ambiente Encima	15 °C	Lote Eliminador	85/1041A
Tº Ambiente Debajo	40 °C	Lote Revelador	70/1168A

**Datos de Elementos**

ID Elemento	Localización	Nomenclatura	Defectos	Nivel de Calidad	Calif.	Ctrl. Geom.	Observaciones
1	UNION PERFIL	VIVIENDA 1, ESCAL... PT1	S.D.N.	1	A	220/-	NINGUNA
2	UNION PERFIL	VIVIENDA 2, ESCAL... PT1	S.D.N.	1	A	220/-	NINGUNA
3	UNION PERFIL	VIVIENDA 3, ESCAL... PT1	S.D.N.	1	A	220/-	NINGUNA
4	PLACA DE ANCLAJE	PILAR 1 PT1	S.D.N.	1	A	200/7.0	NINGUNA
5	PLACA DE ANCLAJE	PILAR 2 PT1	S.D.N.	1	A	240/7.0	NINGUNA

**Datos Auditoría**

Creado el 03.09.2021 Creado por ABAPCEMOSA Última modif. 09.09.2021 Modificado por ABAPCEMOSA

Ilustración 75: Vista de datos de la Entrada de Resultados

Para guardar estos datos, seleccionamos el botón de guardar y nos mostrará una ventana emergente de que se han guardado los datos correctamente (Ilustración 76).

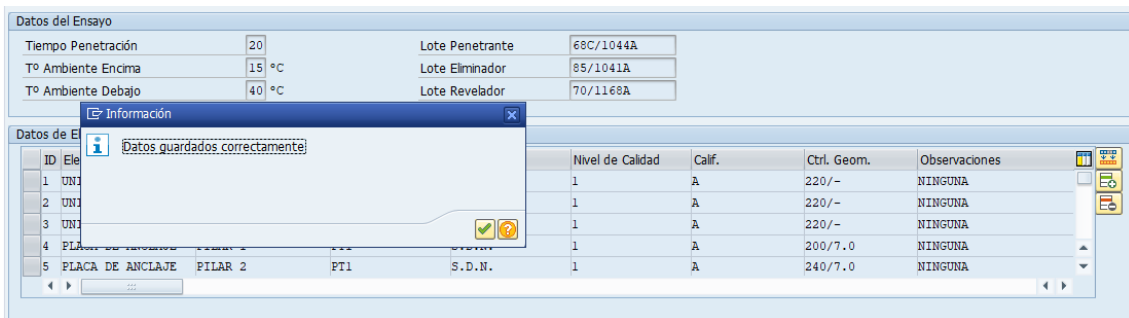


Ilustración 76: Mensaje de guardado

Los fallos en esta pantalla están bien asegurados ya que los campos pertenecientes al ensayo en su total mayoría son de tipo *CHAR*, es decir, que permiten cualquier tipo de caracteres o *String*. Además, cada campo tiene un límite de longitud para añadir datos, como, por ejemplo, el Nivel de Calidad solo permite como máximo dos caracteres. En cambio, los campos que no son iguales son los de Tiempo de Penetración y los campos de Temperatura, puesto que ahí van números y por ello son de tipo *NUMC*, que viene a ser que solo permiten valores de tipo numérico y no permiten poner letras. Si se intenta poner algún carácter y se intenta guardar, el programa nos mostrará un mensaje predefinido en SAP y se muestra en estas ocasiones este tipo de error (Ilustración 77).

Datos del Ensayo					
Tiempo Penetración	<input type="text" value=""/>	Lote Penetrante	<input type="text" value="85/1041A"/>		
Tº Ambiente Encima	<input type="text" value="4"/> °C	Lote Eliminador	<input type="text" value="85/1041A"/>		
Tº Ambiente Debajo	<input type="text" value="40"/> °C	Lote Revelador	<input type="text" value="70/1168A"/>		

Datos de Elementos					
ID	Elemento	Localización	Nomenclatura	Defectos	Nivel de Calidad
1	UNION PERFIL	VIVIENDA 1, ESCAL...	PT1424242424	S.D.N.	44
2	UNION PERFIL	VIVIENDA 2, ESCAL...	PT1	S.D.N.	1
3	UNION PERFIL	VIVIENDA 3, ESCAL...	PT1	S.D.N.	1
4	PLACA DE ANCLAJE	PILAR 1	PT1	S.D.N.	1
5	PLACA DE ANCLAJE	PILAR 2	PT1	S.D.N.	1

Datos Auditoria			
Creado el	<input type="text" value="03.09.2021"/>	Creado por	<input type="text" value="ABAPCEMOSA"/>
		Última modif.	<input type="text" value=""/>

 Entrada sólo en la forma \_

Ilustración 77: Mensaje de error

Una vez realizada las pruebas que se pueden hacer en el Backend, mostraremos cómo serán las pruebas en el Frontend, pero antes mostraremos el paso intermedio entre estas dos, ya que hacer las pruebas directamente desde Frontend habría sido muy pesado por el hecho de que si hubiese fallos no sabríamos dónde estaría exactamente cuándo se hacen las llamadas a los métodos redefinidos (*GET*, *UPDATE*, *CREATE* y *DELETE*), por ello es necesario hacer las pruebas de estos métodos antes de pasar a la implementación de la aplicación en SAPUI5.

Para realizar estas pruebas disponemos de una herramienta en el entorno de Desarrollo Gateway en SAP que se accede desde la transacción */IWFND/MAINT\_SERVICE* (Ilustración 78). Desde aquí seleccionamos nuestro proyecto y pasamos a la pantalla de cliente para poder realizar las llamadas a los métodos redefinidos.



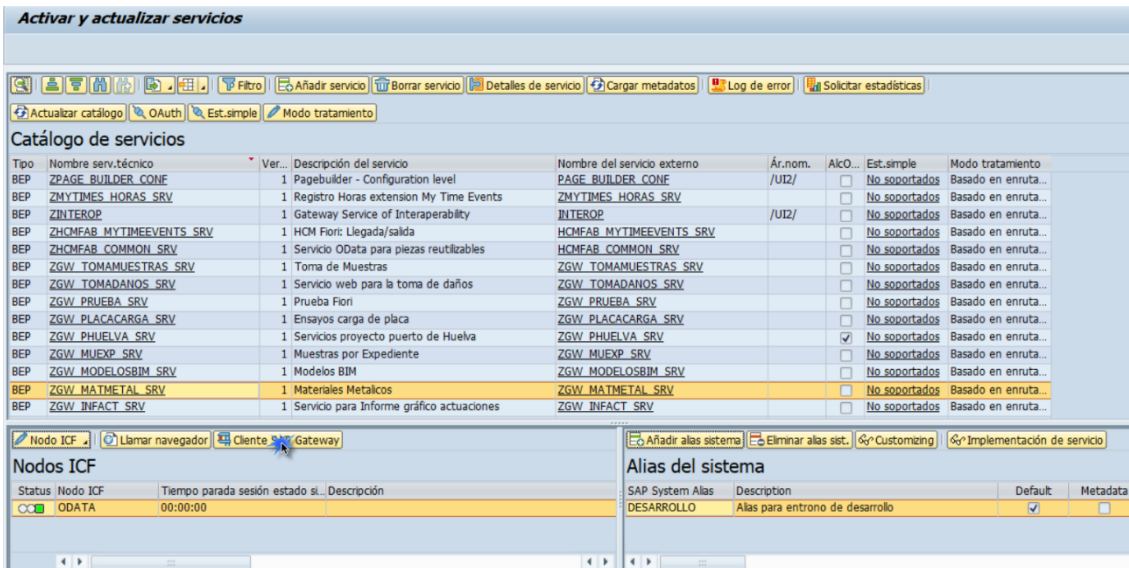


Ilustración 78: Transacción para usar el proyecto para Frontend

La pantalla que nos aparece nos sirve para simular las peticiones que realizaremos en la aplicación introduciendo la *URI* para ver los resultados de dicha petición (Ilustración 79).

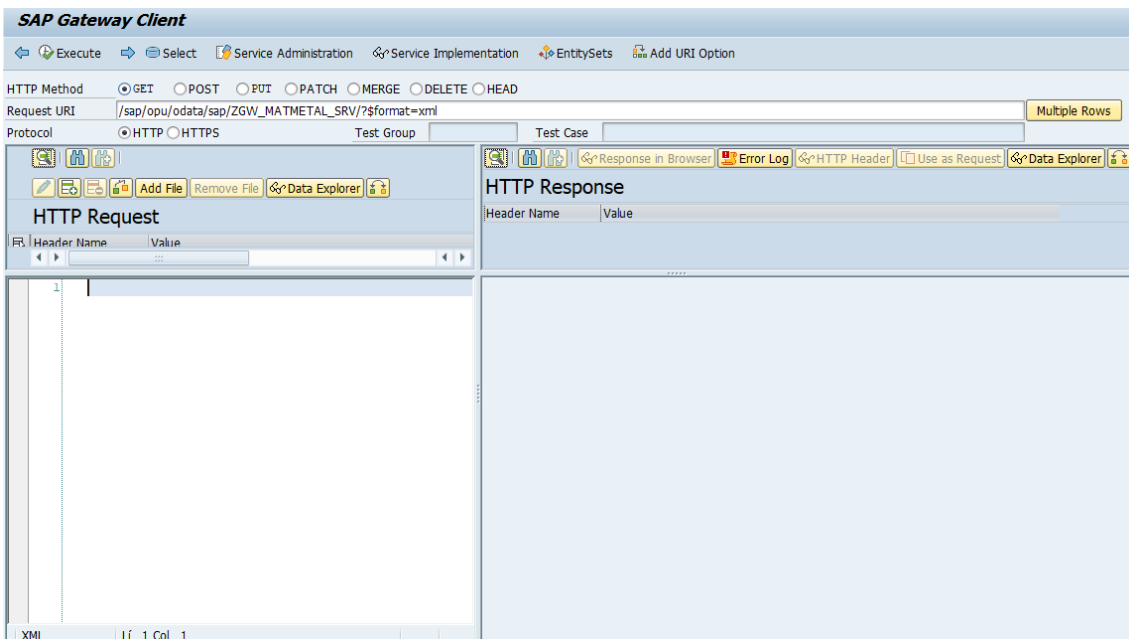


Ilustración 79: Pantalla de pruebas para el proyecto

Probaremos primero con el método *GET*. Para este método existen dos redefiniciones, el *ENTITY* y el *ENTITYSET*. Tenemos que especificar correctamente y tener en cuenta la *URI* que pongamos para que entre en un método o en otro, puesto el *ENTITY* es una estructura y solo

devolverá un registro, pero el *ENTITYSET* es una tabla que puede devolver más de un registro. Para usar el *ENTITYSET* nos vale con poner la siguiente *URI* (Ilustración 80), con esto obtendríamos uno o más filas dependiendo de lo que esté guardado en la tabla de la BBDD.

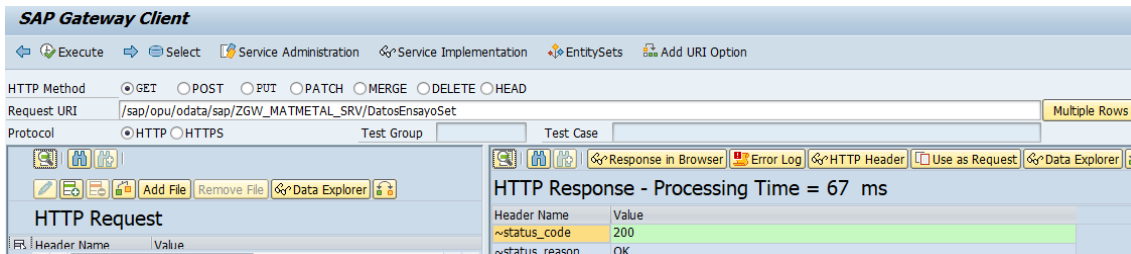


Ilustración 80: Mensaje de confirmación para la carga

Si lo que queremos es obtener un solo registro, debemos especificar en la *URI* insertando las claves que asignamos a las entidades (Ilustración 81). Cuando especificamos claves, entrará al método redefinido *ENTITY* y obtendrá o no el registro que esté guardado en la tabla de la BBDD.

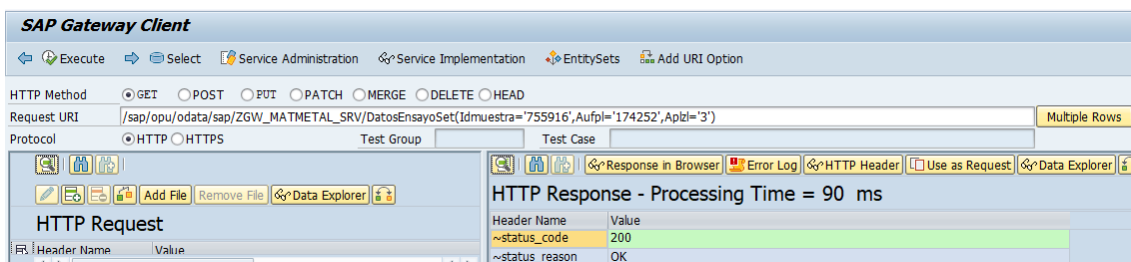


Ilustración 81: Mensaje de confirmación para la carga especificando

Luego tenemos el método *POST*, que sirve para llamar a la función redefinida *CREATE*. En nuestro caso solo hemos redefinido esta función para la entidad de DatosElementos, puesto que es la que tiene la tabla de elementos y en la que se pueden añadir filas. Para probarlo debemos poner en la pestaña de *HTTP Request* los campos de la tabla de elementos con los valores que queramos que se guarden, especificando sobre todo las claves (Ilustración 82).

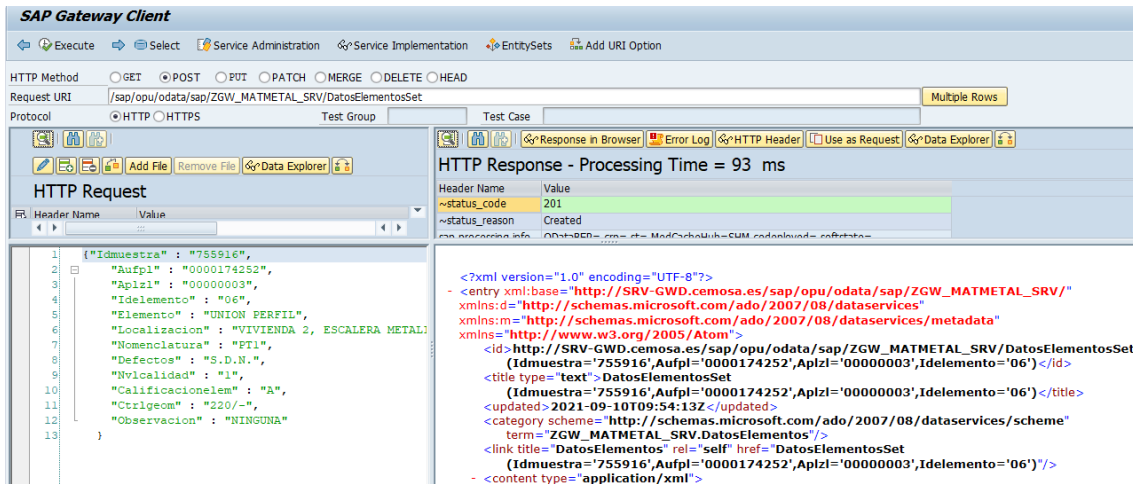


Ilustración 82: Crear filas de la tabla

Tal y como vemos que se pueden añadir filas, se puede también eliminar con el método *DELETE* que llama a la función redefinida *DELETE* de la entidad *DatosElementos*. Aquí debemos especificar exactamente qué elemento queremos eliminar y esto se hace añadiendo las claves en la *URI* (Ilustración 83).

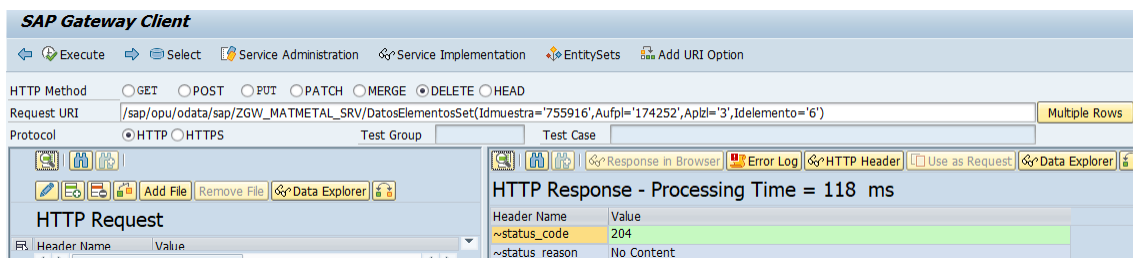


Ilustración 83: Eliminar filas de la tabla

Para acabar con las pruebas aquí en el Gateway, vemos el método *PUT* que llama a la función redefinida *UPDATE* para actualizar un dato de un registro que ya existe en la tabla de la BBDD (Ilustración 84).

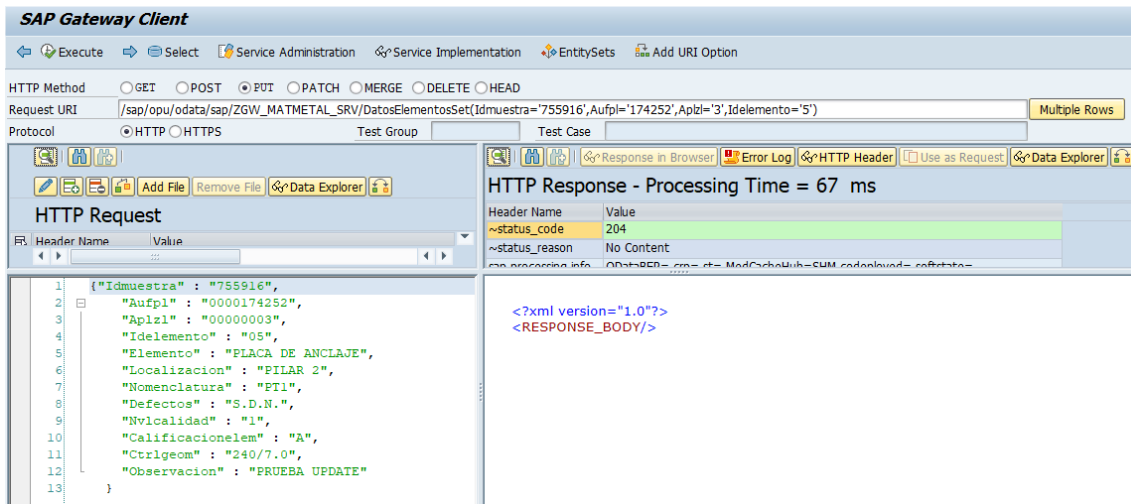


Ilustración 84: Guardar datos modificando en la tabla

Podemos visualizar si todas las peticiones se han hecho correctamente mirando en las tablas o mirando en la Entrada de Resultados en el servicio (Ilustración 85).

ID	Elemento	Localización	Nomenclatura	Defectos	Nivel de Calidad	Calif.	Ctrl. Geom.	Observaciones
1	UNION PERFIL	VIVIENDA 1, ESCAL.	PT1	S.D.N.	1	A	220/-	NINGUNA
2	UNION PERFIL	VIVIENDA 2, ESCAL.	PT1	S.D.N.	1	A	220/-	NINGUNA
3	UNION PERFIL	VIVIENDA 3, ESCAL.	PT1	S.D.N.	1	A	220/-	NINGUNA
4	PLACA DE ANCLAJE	PILAR 1	PT1	S.D.N.	1	A	200/7.0	NINGUNA
5	PLACA DE ANCLAJE	PILAR 2	PT1	S.D.N.	1	A	240/7.0	PRUEBA UPDATE

Ilustración 85: Confirmando la modificación.

Ahora bien, realizaremos las pruebas en la aplicación de Fiori. Todas las pruebas que realicemos aquí son las mismas que se han realizado en Gateway, ya que se llama usando la misma URI y los mismos métodos.

# 5

## Futuras Mejoras y Posibles Ampliaciones

Una vez transportado el proyecto al entorno de Producción y donde puede ser usado por todos los usuarios de la empresa y sobre todo por los obreros que son los principales usuarios que usan este ensayo, se espera que los primeros meses de incidencias se seguirá dando soporte y atendiendo a los problemas que pueda haber y las solicitudes de cambios o inclusiones de nuevas funcionalidades por parte de los jefes de Laboratorio.

Continuando el flujo de procesos del área de control de calidad de los materiales se enfocará en las siguientes ampliaciones:

1. Listado ALV: posibilidad de visualizar las muestras de tipo de material Materiales Metálicos con la posibilidad de filtrar por distintos campos para una fácil visualización de los datos.
2. Implementación de IA: opción de hacer que el sistema aprenda en función de los datos previos insertados por los usuarios y completar sin necesidad de que los usuarios introduzcan los datos. Además, leer los datos y ver si son adecuados y que tengan sentido.



# 6

## Conclusiones

Finalizado el proceso de Desarrollo e Implementación de la aplicación en Backend y Frontend se puede dar por conseguido el objetivo principal de este TFG. Se ha aportado a la empresa un valor añadido aumentando la productividad automatizando el proceso de la toma de la entrada de resultados de una muestra. Este proceso anteriormente no existía y se hacía todo a bolígrafo y papel, lo que ocasiona fallos durante la transferencia de los datos al ordenador.

A pesar de haber logrado el objetivo, han surgido varios inconvenientes durante la realización de dicho trabajo.

El primer contratiempo fue la falta de comunicación que hubo con los técnicos de laboratorio, puesto que están muy ocupados con toda la carga de trabajo y es muy difícil poder contactar con ellos para realizar reuniones para solventar las dudas que iban surgiendo.

El segundo inconveniente fue la falta de experiencia que tenía con la programación en SAPUI5, puesto que cuando hice la formación al comienzo de empezar a trabajar en CEMOSA, en esta parte de Fiori tuve una formación un tanto escasa ya que las pruebas no

las podía realizar en un entorno más realista sino en el entorno de Formación donde no se disponen de datos y apenas hay tablas.

Como último inconveniente destacaría que se ha tardado más de la cuenta en realizar este proyecto debido a la carga de trabajo que he tenido estos meses, puesto que no todo el tiempo era dedicado al proyecto de fin de grado.

A pesar de todo, agradezco mucho la ayuda de mis compañeros de desarrollo por ayudarme con las dudas que me han ido surgiendo sobre la implementación para Fiori, ya que en cuanto al Backend ya llevaba un año trabajando en ello y he ganado mucha experiencia en saber cómo funciona realmente todo.

Debo destacar también que emplear la metodología ágil para realizar este proyecto ha sido una decisión más que correcta, puesto que sabiendo los requisitos previos al desarrollo del proyecto fue más fácil realizar todo el proceso de desarrollo e implementación para luego poder realizar las pruebas y por si fuesen fallos poder corregirlo cuanto antes posible.

El proyecto ha supuesto una oportunidad que valoro muy positivamente en cuanto a aprendizaje y desarrollo personal y profesional. Ha requerido un esfuerzo muy grande asimilar y entender todos los conceptos nuevos tanto por las herramientas usadas como los conceptos que eran desconocidos en la empresa hasta ahora que requerían entenderlo para un correcto funcionamiento del proyecto.



# 7

## Referencias

SAP, ERP: [Qué es SAP y para qué sirve - NETCheck](#)

ABAP: "¿Que es ABAP? Curso ABAP Inicial Gratuito | CVOSOFT."

[http://www.cvosoft.com/sistemas\\_sap\\_abap/recursos\\_tecnicos\\_abap/que-es-abap.php](http://www.cvosoft.com/sistemas_sap_abap/recursos_tecnicos_abap/que-es-abap.php).

Ian H. Witten, Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques*,  
Second Edition, pags. 4-5

Metodología Gestión de requerimientos: [Capítulo 3 DEFINICIÓN REQUERIMIENTOS - Google Sites: Sign-in](#)

SAPUI5 SDK: [SAPUI5 SDK - Demo Kit](#)



# Apéndice A

## Manual de Usuario

### Acceso a SAP-Fiori:

1. Una vez dado de alta por parte del Departamento de Informática podrá acceder a SAP-Fiori (Ilustración 86).



Ilustración 86: Login SAP-Fiori

2. Una vez accedido podrá visualizar las aplicaciones que le hayan sido asignadas. Estas aplicaciones son publicadas por el Departamento de Informática y solicitadas por el mismo usuario que puede pedirlo y que son autorizadas por su responsable o directamente el responsable solicita que se le den esas aplicaciones (Ilustración 87).



Ilustración 87: Aplicaciones disponibles para el usuario

3. En la pestaña de Control de Calidad se encuentran los expedientes para realizar la toma de las muestras. Para realizar una muestra in situ hay que pulsar el botón de Act. In Situ (Ilustración 88).

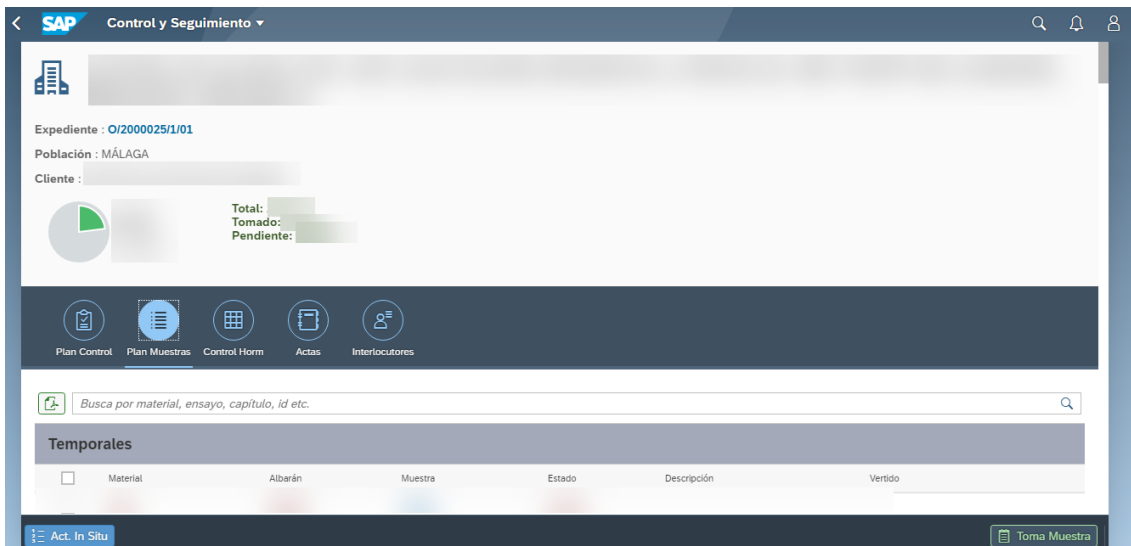


Ilustración 88: Expediente de prueba

4. Luego, elegir el tipo de material del que queremos realizar la muestra (Ilustración 89).

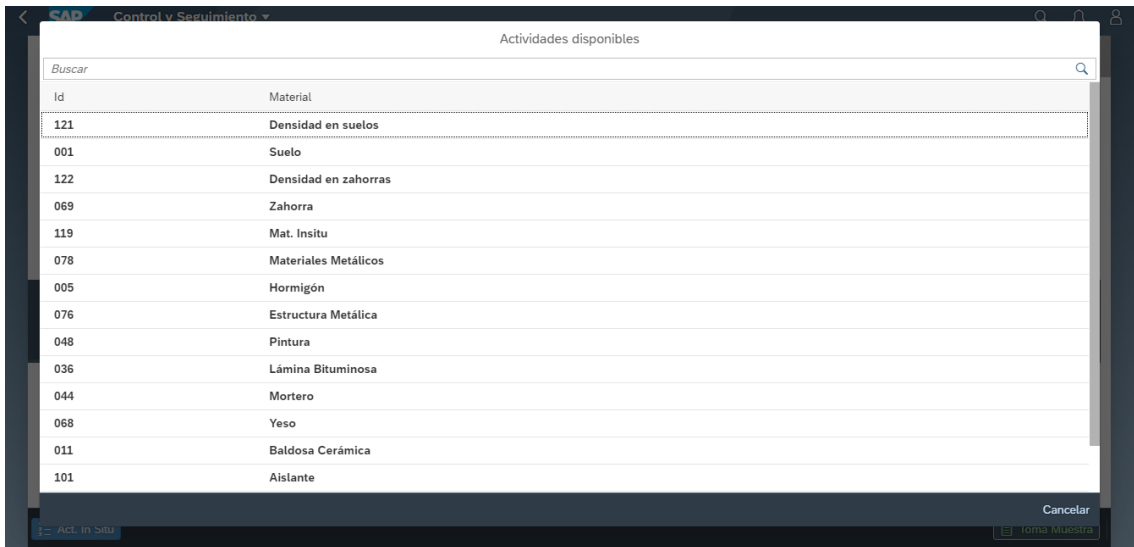


Ilustración 89: Elección del Tipo de Material

- Para finalizar elegiremos los ensayos que estén en el expediente para dicho material (Ilustración 90).

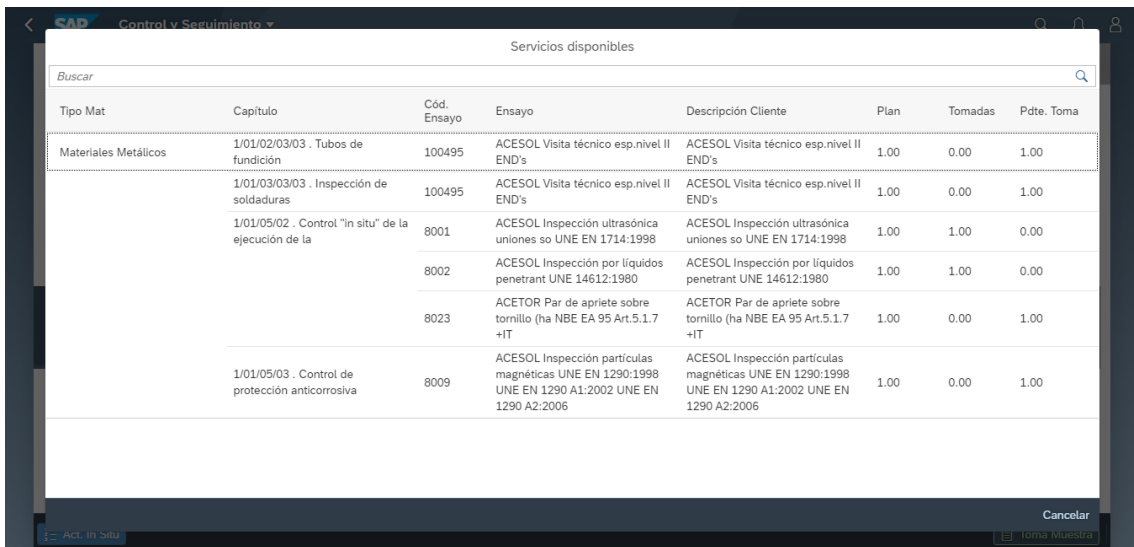


Ilustración 90: Elección del Tipo de Ensayo

- Después, nos pregunta el número de unidades que queremos tomar y le damos a Toma (Ilustración 91).

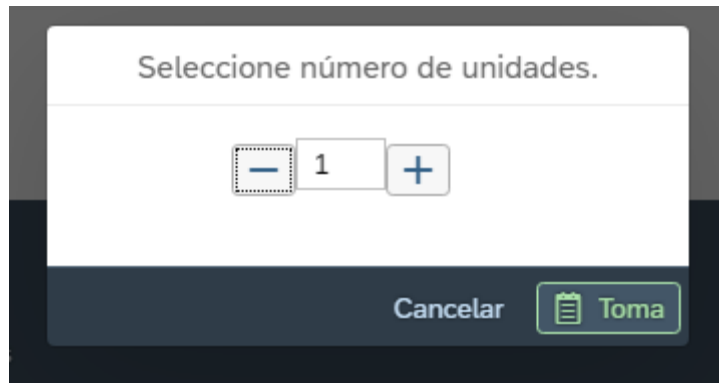


Ilustración 91: Elección de las unidades a tomar

7. La pantalla que vemos es la Entrada de la Muestra para realizar la toma de la muestra (Ilustración 92).

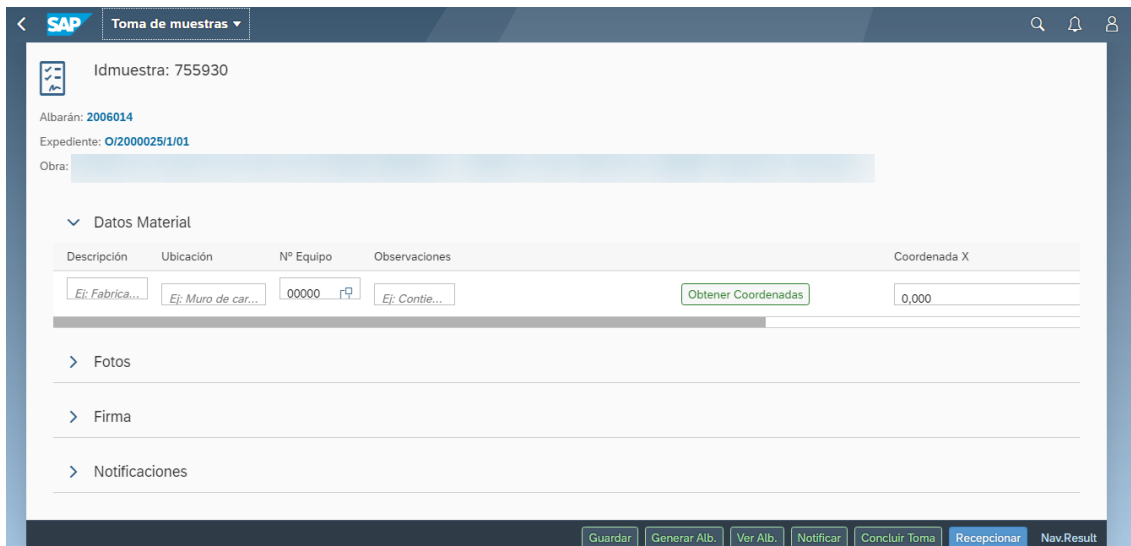


Ilustración 92: Entrada de la muestra en Fiori

8. Rellenamos los datos de la toma de la muestra y guardamos (Ilustración 93).

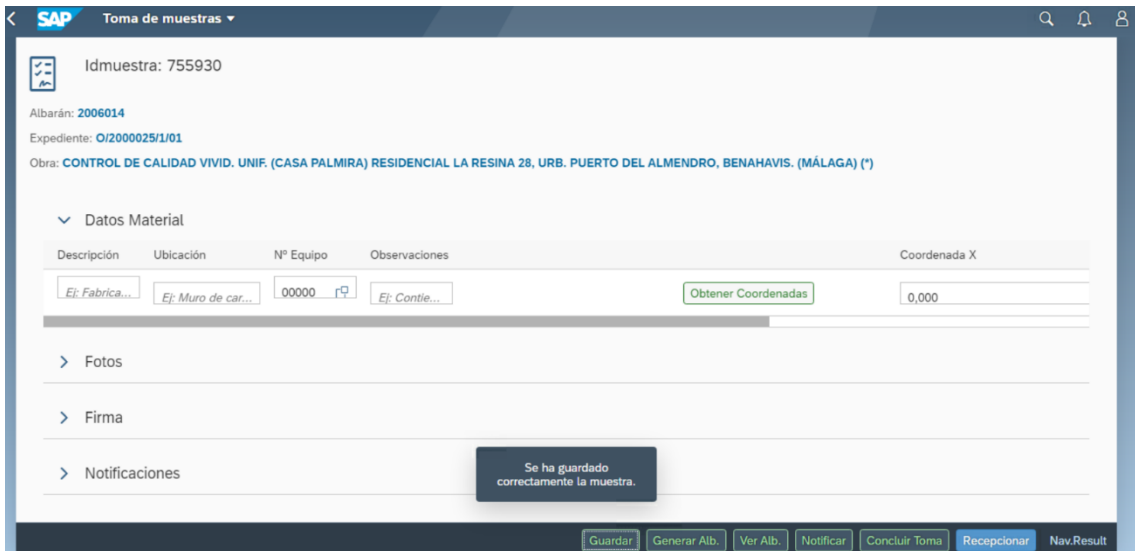


Ilustración 93: Mensaje de guardado de la muestra

9. Una vez guardado navegaremos a la Entrada de Resultados del ensayo (Ilustración 94).

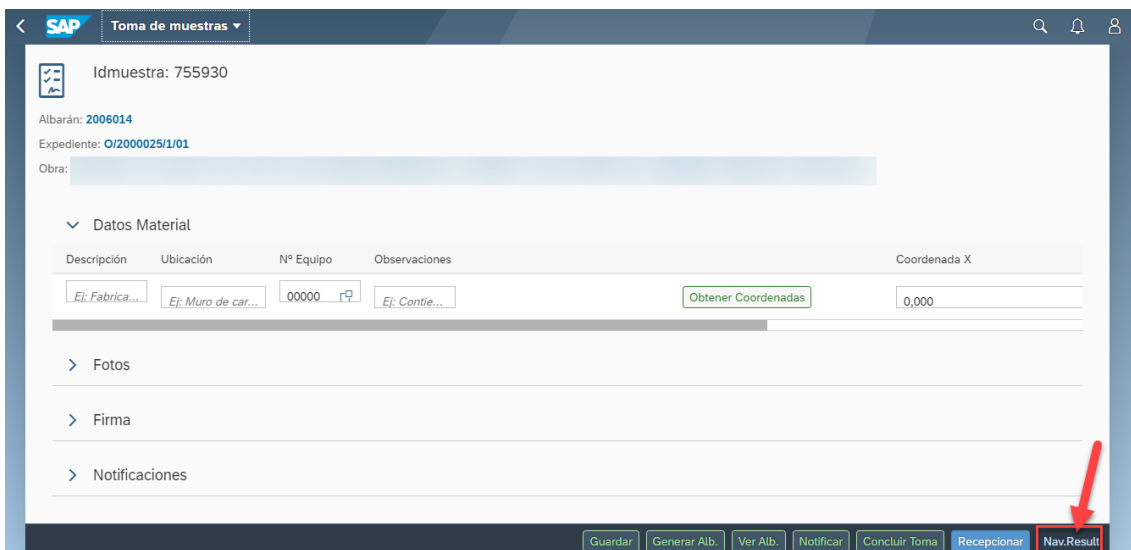


Ilustración 94: Botón para navegar a la Entrada de Resultados

10. En la pantalla vemos el ensayo creado e insertamos los datos (Ilustración 95).

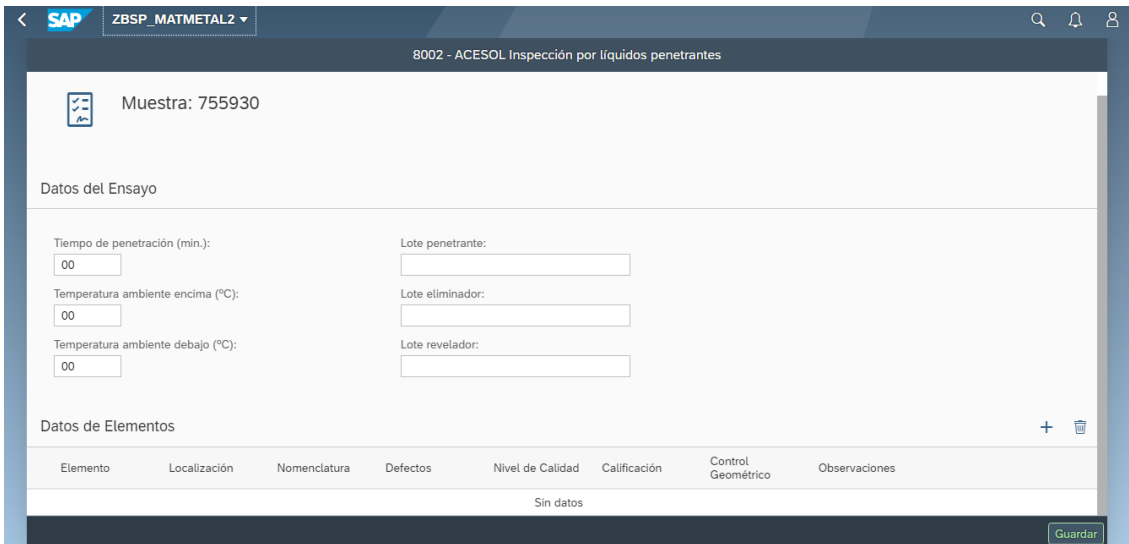


Ilustración 95: Pantalla de la Entrada de Resultados

11. Le damos a guardar una vez rellenados los datos (Ilustración 96).

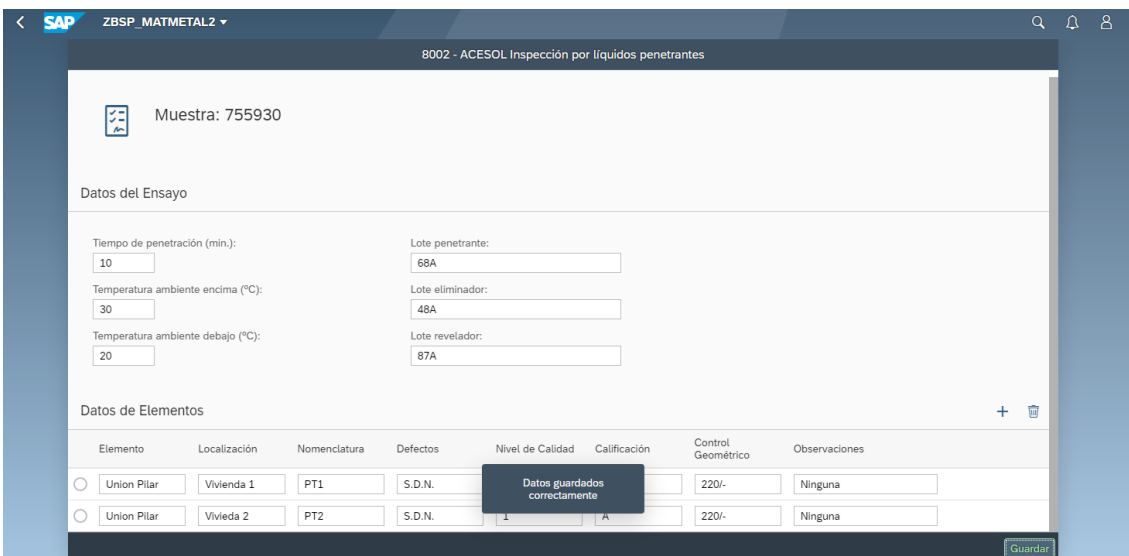


Ilustración 96: Mensaje de guardado en la Entrada de Resultados

## Acceso en SAP

12. Para completar la toma nos vamos a Backend en SAP y falta por hacer la Entrada de la Muestra. Desde la transacción ZLER ponemos la muestra y ejecutamos (Ilustración 97).



**Entrada Común de Hojas de Servicio.**

Id. Muestra: 755930

Referencia de laboratorio:  
Laboratorio: 0201  
Ejercicio: 2021  
N.Muestra:

**Entrada Hojas Servicio 7 (Aceros, Mallas)**

**Cabecera**  
Expediente: 0/2000025/1/01 CONTROL DE CALIDAD Muestra: 755930 RL: 0201 1 0  
Obra: 099993 CONTROL DE CALIDAD VIVID. UNIF. (CASA... Muestra Origen: RL: 0 0  
Cliente: 6 CENTRO DE ESTUDIOS DE MATERIALES Y Aviso Nº: 0

**Datos de Toma**  
Nº Albarán: 2006014 Muestras Albarán: 1 / 1 Fecha-hora toma: 10.09.2021 1441 Duración: MIN  
Operador: Realizada por: ML F. Entrada 10.09.2021 Tipo Informe: MUEIOSSCA ¿Incluye plano?

**Datos de Muestra**  
Tipo material: 78 Materiales Metálicos Lote: Lote Cliente:  
Descripción: Procedencia: Lugar toma: Ubicación:  
Estado muestra: PR Pr Albarán Fabric: Cantidad: UN Long.  
Desig. Malla: ME x Elemento malla:  
Desig. B/A: Tipo de aceros: BARRA RECTA  
Marca de acero: Número de colada:  
F. Fin Serv: Fecha de Pedido:  
Observaciones:

**Servicios**

Cod Servicio	Descripción Ensayo	Operador	Técnico	Fecha Prevista	Lab Destino	Mues Destino	Inicio Real	Ob
8002	ACESOL Inspección por líquidos penetrant UN...						10.09.2021	

Ilustración 97: Transacción para hacer la Entrada de la Muestra

13. Insertamos los datos obligatorios como Descripción, Lugar Toma, Operador y la plantilla, y grabamos (Ilustración 98).

**Entrada Hojas Servicio 7 (Aceros, Mallas)**

**Cabecera**  
Expediente: 0/2000025/1/01 CONTROL DE CALIDAD Muestra: 755930 RL: 0201 1 0  
Obra: 099993 CONTROL DE CALIDAD VIVID. UNIF. (CASA... Muestra Origen: RL: 0 0  
Cliente: 6 CENTRO DE ESTUDIOS DE MATERIALES Y Aviso Nº: 0

**Datos de Toma**  
Nº Albarán: 2006014 Muestras Albarán: 1 / 1 Fecha-hora toma: 10.09.2021 1441 Duración: MIN  
Operador: 631 FERNANDEZ BAEZ, JUAN FRANC Realizada por: ML F. Entrada 10.09.2021 Tipo Informe: MUEIOSSCA ¿Incluye plano?

**Datos de Muestra**  
Tipo material: 78 Materiales Metálicos Lote: Lote Cliente:  
Descripción: Material Procedencia: Lugar toma: PLANTA Ubicación:  
Estado muestra: PR Pr Albarán Fabric: Cantidad: UN Long.  
Desig. Malla: ME x Elemento malla:  
Desig. B/A: Tipo de aceros: BARRA RECTA  
Marca de acero: Número de colada:  
F. Fin Serv: Fecha de Pedido:  
Observaciones:

**Servicios**

Cod Servicio	Descripción Ensayo	Operador	Técnico	Fecha Prevista	Lab Destino	Mues Destino	Inicio Real	Ob
8002	ACESOL Inspección por líquidos penetrant UN...						10.09.2021	

Asignar Plantillas Plantilla Informe: IMET002\_MATMETALES

Creado por: ABAPCEMOSA 10.09.2021 Modificado por: ABAPCEMOSA 10.09.2021

Ilustración 98: Completar campos obligatorios para ir a la Entrada de Resultados

14. Pulsamos doble clic en el número del ensayo y navegamos a la Entrada de Resultados y vemos que los campos están correctamente guardados (Ilustración 99).

**Entrada de Resultados 7 (Materiales Metálicos)**

8002 - ACESOL Inspección por líquidos penetrant UNE 14612:1980

**Datos de Cabecera**

Expediente: 0/2000025/1/01 CONTROL DE CALIDAD Muestra: 755930 R.L.: 0201 1 2021 49  
 Obra: 099993 CONTROL DE CALIDAD VIVID. UNIF. (CASA) Muestra O: R.L.: 0 0  
 Cliente: 6 CENTRO DE ESTUDIOS DE MATERIALES Y Estado muestra: EN Entrada Realizada: 1 de 1 TI: MUE  
 Contratista: Tipo material: 78 F.Fin Muestra:  
 Observación:  
 Fecha Inicio: 10.09.2021 Fecha Fin: Técnico Revisor: F.Revisión: Pedido: /  Ensayo realizado

**Datos del Ensayo**

Tiempo Penetración: 10 Lote Penetrante: 68A  
 Tº Ambiente Encima: 30 °C Lote Eliminador: 48A  
 Tº Ambiente Debajo: 20 °C Lote Revelador: 87A

**Datos de Elementos**

ID	Elemento	Localización	Nomenclatura	Defectos	Nivel de Calidad	Calif.	Ctrl. Geom.	Observaciones
1	Union Pilar	Vivienda 1	PT1	S.D.N.	1	A	220/-	Ninguna
2	Union Pilar	Vivienda 2	PT2	S.D.N.	1	A	220/-	Ninguna

**Datos Auditoria**

Creado el: 10.09.2021 Creado por: ABAPCEMOSA Última modif.: 10.09.2021 Modificado por: ABAPCEMOSA

Ilustración 99: Visualización de los datos añadidos en Fiori

15. Lo último que queda por ver es el informe. Desde la transacción ZUCE se puede visualizar el informe para revisarlo (Ilustración 100).

**Gestión de muestras**

Referencia de Laboratorio

Laboratorio	0201	a		
Id. Libro		a		
Ejercicio		a		
N.Muestra		a		

Datos de la muestra.

Fecha Toma		a		
Fecha Entrada		a		
Fecha de Finalización Servicio		a		
Fecha fin de albarán		a		
Expediente		a		
Tipo de material		a		
Nº Albarán		a		
Id. Muestra	755930	a		
Número de Documento		a		
Proyecto		a		
Código de Obra		a		
Cliente		a		
Operador		a		

Revisar informes

Seleccionar Layout

Layout

Ilustración 100: Transacción para visualizar el informe Adobe

16. Seleccionamos la fila y visualizamos el documento de la muestra (Ilustración 101).

**Gestión de muestras**

Fin informe Correspondencias

Ep.	Fecha Toma	Nombre 1	Expediente	Denominación	Nº Albarán	Id.Muestra	Descripción	Cl. Homi. Desc.	Uds.	Id. Lab	Id. Libro	Año	N.Muestra	Cantidad
	10.09.2021	CENTRO DE ESTUDIOS DE MATERIAL	O/2000025/1/01	CONTROL DE MATERIALES COMPLETA	2006014	755930	Materiales Metálicos	Material	0,0	0201	1	2021	49	1

Ilustración 101: Botón para visualizar el informe Adobe

17. Elegimos el libro al que queremos asignarle (Ilustración 102).

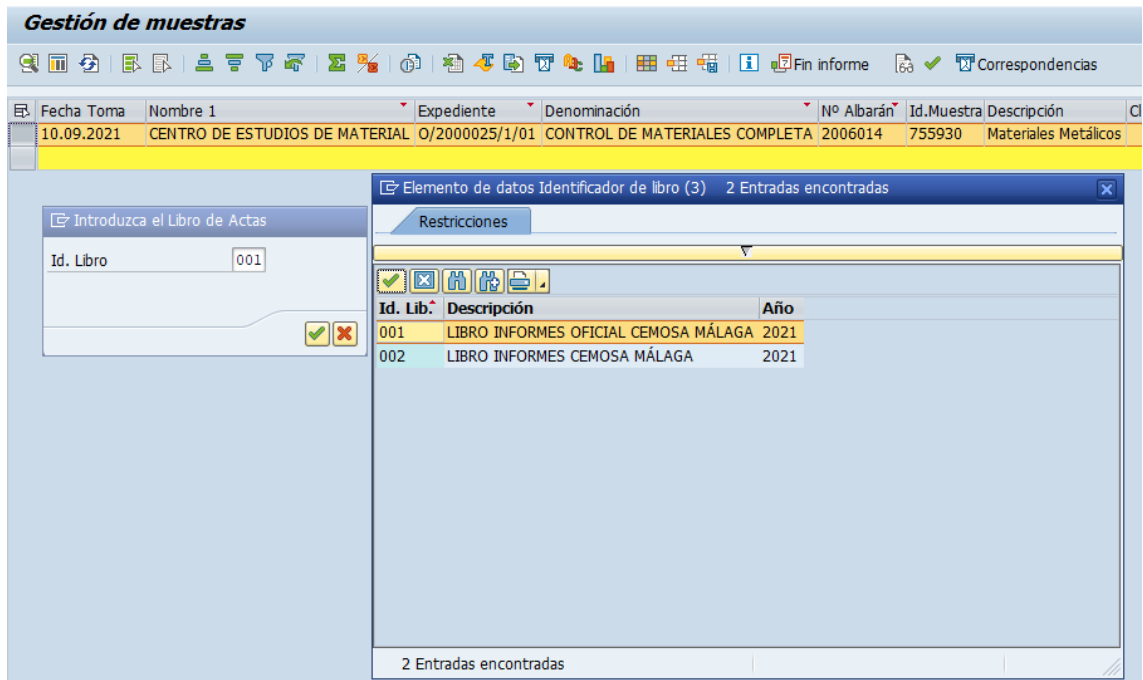


Ilustración 102: Elección del libro

18. Luego en la ventana emergente que sale, elegimos la opción de Visualización de impresión (Ilustración 103).

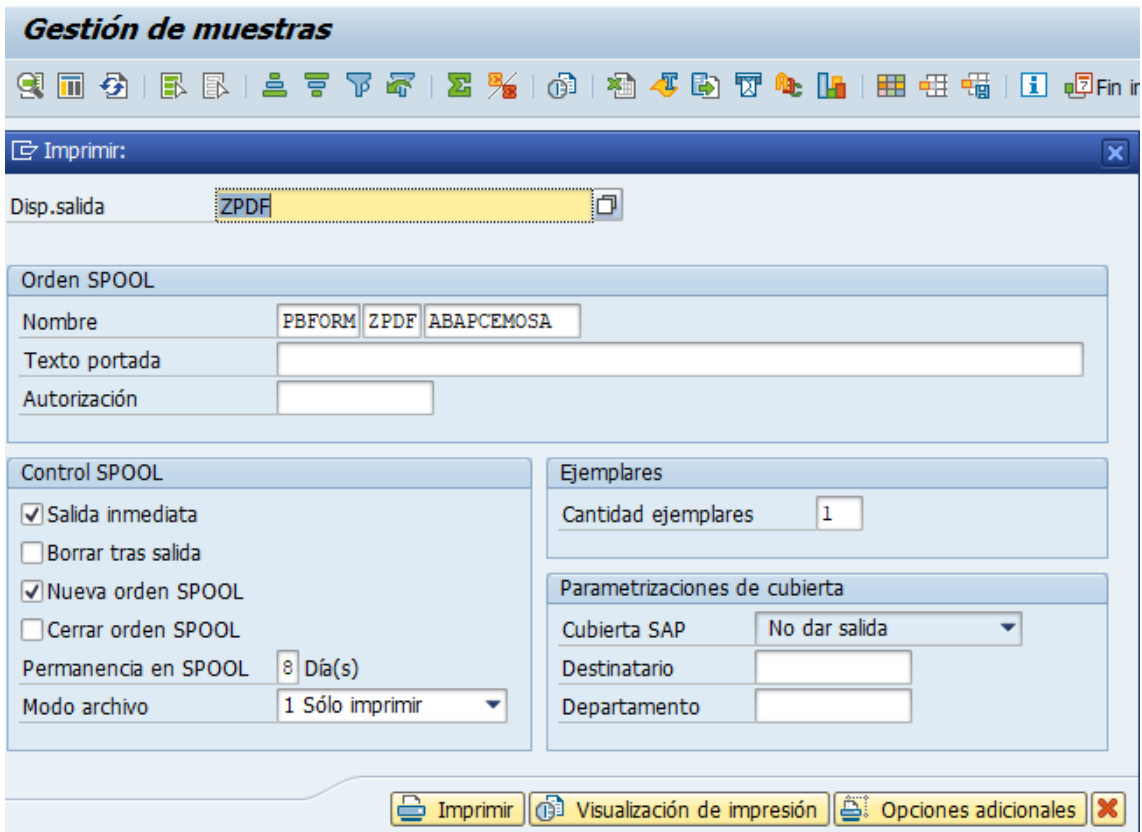


Ilustración 103: Modo de visualización

19. Visualizamos el informe final del acta (Ilustración 104).

Expediente: O/2000025/1/01  
Nº acta: Anula a:  
Obra: CONTROL DE CALIDAD VIVID. UNIF. (CASA PALMIRA) RESIDENCIAL LA RESINA 28, URB. PUERTO DEL ALMENDRO, BENAHAIVIS. (MÁLAGA) (\*)  
Peticionario: CENTRO DE ESTUDIOS DE MATERIALES Y  
Dirección: C/ BENAQUE, 9 29004 MALAGA  
Contratista:  
Dirección técnica:  
Modalidad de control:

### CONTROL DE ESTRUCTURA METÁLICA MEDIANTE LÍQUIDOS PENETRANTES ( )

ALBARÁN: 2006014	CÓDIGO DE MUESTRA: 0201/001/2021/000049(755930)	FECHA TOMA: 10/09/2021-14:41
<b>MATERIAL</b>		
TIPO: Materiales Metálicos	MODALIDAD DE MUESTREO: ML - Muestreo por laboratorio	
DESIGNACIÓN:	REFERENCIA LABORATORIO: Málaga	
FABRICANTE:	MUESTREO POR: JUAN FRANC FERNANDEZ BAEZ	
LUGAR DE TOMA: PLANTA	LOTE LABORATORIO:	
UBICACIÓN:	LOTE CLIENTE:	
<b>TRABAJO REALIZADOS</b> - De acuerdo con el programa establecido, se han realizado los siguientes ensayos.		
ACESOL Inspección por líquidos penetrant	UNE 14612:1980	

#### RESULTADOS

Los resultados de los ensayos se presentan en las siguientes páginas.

#### EJECUCIÓN Y MÉTODO OPERATIVO

Este ensayo ha sido realizado siguiendo las recomendaciones reflejadas en las normas:

UNE EN ISO 9712:2012, UNE EN ISO 17635:2017, UNE EN ISO 5817:2014, UNE EN ISO 3452-1:2013, UNE EN ISO 3452-2:2013 y UNE EN ISO 3059:2013 y los niveles de aceptación de la norma UNE EN 23277:2015.

Los líquidos penetrantes son muy adecuados para la detección de defectos superficiales, es decir, defectos que afloran a la superficie.

Se basan estos líquidos en la propiedad que poseen de introducirse por capilaridad en el interior de discontinuidades que tengan una abertura hacia la superficie examinada.

Eliminando después el exceso de penetrante, la aplicación de un revelador con alto poder de absorción, nos mostrará aquellos defectos no apreciables a simple vista.

Los materiales a utilizar son:

		Lote nº
PENETRANTE:	CRACK MARKER MR 68C (rojo)	68A
ELIMINADOR:	CRACK MARKER MR 85	48A
REVELADOR:	CRACK MARKER MR 70 (blanca)	87A

#### - Preparación de la superficie:

Se limpia toda la superficie de proyecciones, grasa, escoria, etc., y dejando un tiempo de diez minutos antes de aplicar el penetrante.

#### - Aplicación del penetrante:

El penetrante se aplica siguiendo las recomendaciones del fabricante y en una capa uniforme.

El tiempo de penetración ha sido de 10 minutos.

La temperatura de la pieza es ambiental, estando por encima de los 30°C y por debajo de los 20°C.

#### - Aplicación del revelador:

Se extendió en capa uniforme, en toda la superficie de los cordones.

#### - Examen:

El examen del cordón se realizó a los quince minutos.

ELEMENTO	LOCALIZACIÓN	NOMENCLATURA	DEFECTOS	NIVEL DE CALIDAD	CALIF.	CONTROL GEOMÉTRICO LONGITUD (mm)	OBSERVACIONES
Union Pilar	Vivienda 1	PT1	S.D.N.	1	A	220/-	Ninguna
Union Pilar	Vivienda 2	PT2	S.D.N.	1	A	220/-	Ninguna

Fdo. ELENA FRADE VIANO  
Director Técnico de Laboratorio  
Licenciado en Ciencias Químicas

Málaga a //

Fdo. YOLANDA GARRIDO CAMACHO  
Responsable de Ensayos Físicos  
Ldo. en Ciencias Ambientales

--	--

Ilustración 104: Visualización del informe Adobe



UNIVERSIDAD  
DE MÁLAGA | **uma.es**

E.T.S de Ingeniería Informática  
Bulevar Louis Pasteur, 35  
Campus de Teatinos  
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA