



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

— **TELECOM** ESCUELA  
TÉCNICA **VLC** SUPERIOR  
DE INGENIERÍA DE  
TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de  
Telecomunicación

Desarrollo de una aplicación SAP Fiori para el módulo SAP  
WM de un cliente empresarial

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías y Servicios de  
Telecomunicación

AUTOR/A: Pulgar Soriano, Daniel

Tutor/a: Guijarro Coloma, Luis Alejandro

CURSO ACADÉMICO: 2022/2023

## **Resumen**

En el presente trabajo de final de grado se ha llevado a cabo el desarrollo de una aplicación SAP Fiori para el módulo SAP WM de un cliente empresarial con ciertos requisitos pedidos por éste. En primer lugar, antes de llevar a cabo dicho desarrollo, se ha hablado del programa SAP, así como de la utilidad que tiene éste y sus distintos módulos. Asimismo, se ha hablado del entorno donde se va a desarrollar la aplicación mencionando sus características más destacables. Más tarde, se realiza una descripción de cómo va a ser la aplicación y que pasos seguir para el desarrollo de ésta. Cabe destacar, que los datos referentes al desarrollo de la aplicación se han extraído del documento funcional que se ha elaborado tras conocer los requisitos pertinentes y que es analizado en este trabajo. Finalmente, cuando la aplicación se encuentre terminada y operativa, se hace el despliegue de ésta en la plataforma SAP Fiori Launchpad del cliente.

Palabras clave: SAP, SAPUI5, back-end y front-end, ABAP y oData.

## **Abstract**

In this final degree project, the development of a SAP Fiori application for a specific client's SAP WM module with certain requested requirements has been carried out. Firstly, before proceeding with the development, an overview of the SAP program has been provided, including its utility and different modules. Additionally, the environment where the application will be developed has been described, highlighting its main features. Subsequently, a description of the application and the steps to be followed for its development is presented. It is worth noting that the data related to the application development has been extracted from the functional document that was created after understanding the relevant requirements, which is analyzed in this work. Finally, once the application is completed and operational, it is deployed on the client's SAP Fiori Launchpad platform.

**Keywords:** SAP, SAPUI5, back-end and front-end, ABAP and oData.

## **Agradecimientos**

Quisiera expresar mi más sincero agradecimiento a la empresa Sothis por brindarme una valiosa oportunidad de formación durante la realización de mi Trabajo de Fin de Grado (TFG). La experiencia adquirida durante mi estancia en la empresa ha sido invaluable para mi desarrollo académico y personal.

Agradezco especialmente a los profesionales de Sothis que me han guiado y compartido sus conocimientos a lo largo de mi formación en la empresa. Sobre todo, a personas como German Serrat Salvador y Jonathan Xavier Días Toapanta, las cuales me han sido de apoyo en el proyecto que he desarrollado en el presente trabajo.

Además, quiero expresar mi gratitud a la empresa en su conjunto por proporcionarme un ambiente de trabajo estimulante y colaborativo y por creer en mi potencial y brindarme la oportunidad de aplicar los conocimientos teóricos adquiridos durante mi formación académica en un entorno real.

Por último, agradecer a mis padres, por ser mi inspiración, por apoyarme, así como ayudarme durante toda mi vida y especialmente más en estos últimos años de carrera donde las cosas no han sido fáciles.

# ÍNDICE DE CONTENIDO

<b>Capítulo 1. Introducción y Objetivos</b> .....	5
1.1. Introducción .....	5
1.2. Objetivos .....	6
<b>Capítulo 2. Programa SAP</b> .....	6
2.1. ¿Qué es un ERP? .....	6
2.2. ¿Qué es SAP? .....	9
2.3. ¿Qué utilidad tiene el software de SAP? .....	9
2.4. Módulos del programa .....	10
2.5. SAP Fiori .....	12
2.5.1. Definición .....	12
2.5.2. Utilidad .....	13
2.5.3. Arquitectura .....	13
2.6. SAP WebIDE .....	14
2.7. SAPUI5 .....	15
2.7.1. Implementación de una aplicación SAPUI5 .....	15
2.8. Lenguaje de programación ABAP .....	17
<b>Capítulo 3. Desarrollo de la aplicación</b> .....	18
3.1. Descripción de la aplicación .....	18
3.2. Cloud Connector y creación del destino .....	19
3.3. Creación de la estructura y tipos de tabla del modelo de datos .....	20
3.4. Creación de los módulos de funciones .....	21
3.5. Creación del modelo de datos .....	23
3.6. Publicación del modelo de datos .....	26
3.6.1. Publicación del modelo de datos como un servicio OData .....	27
3.7. Front-end .....	28
3.7.1. Creación del proyecto .....	28
3.7.2. Creación de la vista principal ( <i>Worklist.view.xml</i> ) .....	31
3.7.3. Creación del controlador asociado a la vista principal ( <i>Worklist.controller</i> ) .....	33
3.7.4. Creación de las anotaciones locales ( <i>annotation0.xml</i> ) .....	36
3.7.5. Creación del formatter ( <i>formatter.js</i> ) .....	37

3.8. Vista en web de la aplicación .....	38
<b><i>Capítulo 4. Despliegue de la aplicación en el Launchpad del cliente</i></b> .....	42
4.1. Pasos a seguir desde el planteamiento hasta el despliegue de la aplicación .....	42
<b><i>Capítulo 5. Conclusiones</i></b> .....	44
5.1. Conclusiones .....	44
<b><i>Bibliografía</i></b> .....	45
<b><i>Anexo</i></b> .....	46

## ÍNDICE DE TABLAS

Tabla 1. <i>Creación de la estructura de almacén</i> .....	20
Tabla 2. <i>Creación de la estructura de línea de producción</i> .....	20
Tabla 3. <i>Creación de la estructura de órdenes</i> .....	21
Tabla 4. <i>Propiedades de la entidad de órdenes</i> .....	23
Tabla 5. <i>Propiedades de la entidad de almacén</i> .....	23
Tabla 6. <i>Propiedades de la entidad de línea</i> .....	24
Tabla 7. <i>Mapeo del “Entity Set” de órdenes</i> .....	25
Tabla 8. <i>Mapeo del “Entity Set” de línea de producción</i> .....	25

## ÍNDICE DE FIGURAS

Figura 1. <i>Procesos empresariales integrables por un ERP</i> .....	7
Figura 2. <i>ERPS utilizados por las 1000 empresas que más facturan</i> .....	8
Figura 3. <i>Características destacables de Fiori</i> .....	12
Figura 4. <i>Arquitectura de SAP Fiori</i> .....	13
Figura 5 <i>Publicación del modelo de datos en la transacción “/n/iwfn/maint_services”</i> .....	26
Figura 6. <i>Conectividad exitosa entre el cliente y el SAP Gateway</i> .....	27
Figura 7. <i>Creación del proyecto</i> .....	28
Figura 8. <i>Conexión al sistema y servicio publicado</i> .....	29
Figura 9. <i>Personalización del Worklist</i> .....	29
Figura 10. <i>Archivos que conforman el proyecto</i> .....	30
Figura 11. <i>Definición de la vista, filtros y tabla inteligente en la vista principal</i> .....	31
Figura 12. <i>Definición de columnas y campos en la vista principal</i> .....	33
Figura 13. <i>Función “_showObject”</i> .....	34
Figura 14. <i>Función “onUpdateFinished”</i> .....	34
Figura 15. <i>Función “oPress”</i> .....	34
Figura 16. <i>Función “onSearch”</i> .....	34
Figura 17. <i>Función “onRefresh”</i> .....	35
Figura 18. <i>Función “_applySearch”</i> .....	35
Figura 19. <i>Función “onLocatePress”</i> .....	36
Figura 20. <i>Anotaciones locales del proyecto</i> .....	37
Figura 21. <i>Formatter</i> .....	38
Figura 22. <i>Llamada desde la vista principal al formatter</i> .....	38
Figura 23. <i>Diseño final de la aplicación</i> .....	38
Figura 24. <i>Filtro por número de orden</i> .....	39
Figura 25. <i>Resultado del filtraje de la orden 80007828</i> .....	39
Figura 26. <i>Filtraje por línea a abastece</i> .....	39
Figura 27. <i>Ubicación actual del lote 0822003260</i> .....	40
Figura 28. <i>Reubicación del lote 0822003260</i> .....	40
Figura 29. <i>Nueva ubicación del lote 0822003260</i> .....	41
Figura 30. <i>Error al reubicar el lote en el SAP Gateway</i> .....	41
Figura 31. <i>Mensaje de error al reubicar el lote en la interfaz web</i> .....	42



## ***Capítulo 1. Introducción y Objetivos***

En este capítulo se va a introducir el tema a tratar en el presente trabajo, así como los objetivos a conseguir.

### ***1.1. Introducción***

En el entorno empresarial actual, la gestión eficiente de los procesos logísticos y el control de inventario son elementos fundamentales para el éxito y la competitividad de las organizaciones. En este contexto, las soluciones integradas de gestión empresarial, como el Sistema SAP (Systems, Applications, and Products in Data Processing), se han convertido en pilares esenciales para optimizar y agilizar las operaciones logísticas.

El módulo SAP WM (Warehouse Management) desempeña un papel crucial en la gestión de almacenes y la logística interna de una empresa. Permite supervisar y controlar de manera efectiva todas las actividades relacionadas con el almacenamiento, la ubicación de materiales, la gestión de stocks y la ejecución de operaciones de entrada y salida. Sin embargo, para maximizar la eficiencia y adaptarse a las necesidades específicas de cada organización, es fundamental desarrollar aplicaciones personalizadas que complementen las funcionalidades estándar proporcionadas por SAP WM.

En el presente trabajo, se abordará el desafío de desarrollar una aplicación SAP Fiori para el módulo SAP WM de un cliente empresarial. SAP Fiori es una plataforma moderna de diseño y desarrollo de aplicaciones que ofrece una experiencia de usuario intuitiva y centrada en el usuario, optimizada para dispositivos móviles y de escritorio.

Cabe destacar que el presente trabajo ha sido realizado durante el período de prácticas en el curso 2022/2023 en Sothis, una empresa tecnológica con sede en España que trata áreas como son la consultoría, el desarrollo software, infraestructuras tecnológicas y ciberseguridad. Ésta se enfoca en entender las necesidades y desafíos específicos de cada cliente para poder ofrecerle a éste soluciones personalizadas y adaptadas a sus requerimientos.

Por otro lado, se va a mencionar de manera breve las características del cliente con el que se está tratando de manera que se pueda contextualizar la aplicación pedida por éste. Dicho cliente se encarga de la exploración de los mercados y la obtención, así como la financiación y transporte de los productos de los más potentes productores de acero. Éste también se encarga de ofrecer soporte necesario para las operaciones de los clientes en los mercados internacionales.

## ***1.2. Objetivos***

El objetivo principal de este proyecto es diseñar y desarrollar una aplicación SAP Fiori que se integre con el módulo SAP WM existente en el cliente, brindando una interfaz amigable y eficiente para llevar a cabo tareas específicas de gestión de almacén. Esto permitirá a los usuarios acceder de manera rápida y sencilla a la información relevante, realizar consultas de inventario, ejecutar movimientos de mercancías, gestionar ubicaciones y llevar un seguimiento preciso de las operaciones logísticas. Para lograr dicho objetivo, se detallará cada uno de los pasos seguidos en profundidad.

Una vez se tenga la aplicación desarrollada se comprobará que cumple los requisitos que fueron establecidos inicialmente por el cliente y que funciona correctamente, mostrando de manera práctica el funcionamiento de ésta.

El resultado final de este proyecto no solo será la entrega de una aplicación SAP Fiori funcional y adaptada a las necesidades específicas del cliente, sino también la adquisición de conocimientos valiosos sobre el desarrollo de aplicaciones personalizadas en el entorno SAP WM. Este TFG representa una oportunidad para explorar y aplicar habilidades técnicas y conceptos empresariales en un contexto real, contribuyendo así al crecimiento y la mejora continua de la organización cliente.

## ***Capítulo 2. Programa SAP***

En este capítulo se va a explicar en qué consiste un ERP y de qué trata SAP, hablando en profundidad de la utilidad de éste, sus módulos, el lenguaje de programación que se usa en dicho ERP y la interfaz web a partir de la cual se llevará el proceso de creación de la aplicación pedida.

### ***2.1. ¿Qué es un ERP?***

Un sistema de planificación de recursos empresariales (ERP) viene a ser un conjunto de aplicaciones de software integradas, las cuales permiten automatizar la mayor parte de las prácticas de negocio que tienen relación con los aspectos tanto productivos como operativos de la empresa, mejorando así la eficiencia y la productividad de la empresa, así como la toma de decisiones. A continuación, se describe con más detalle cada una de las características de un ERP [1]:

- Integración de procesos: éste integra procesos empresariales como pueden ser de finanzas, recursos humanos y muchos más en un solo sistema. Los procesos empresariales integrables por un ERP son los mostrados en la Figura 1.



Figura 1. Procesos empresariales integrables por un ERP.

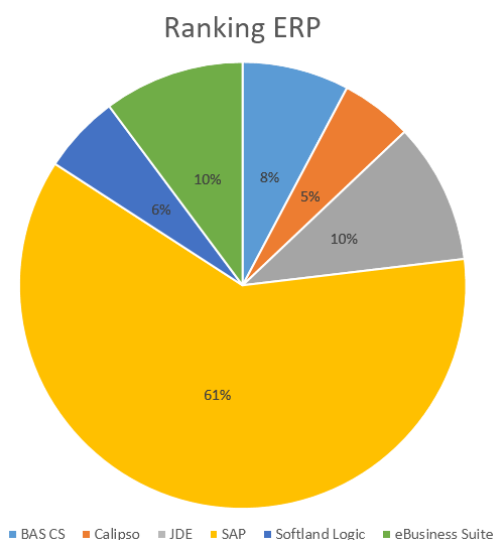
Fuente: [www.digitalicce.org/](http://www.digitalicce.org/)

- Base de datos centralizada: un ERP es capaz de diseñar, crear y gestionar una sola base de datos para toda la organización, evitando así la duplicación de datos, la generación de información innecesaria y la redundancia. De esta forma, al registrar un cambio en una base de datos centralizada, el cambio se produce una sola vez de forma definitiva y es registrado, quedando vigente para beneficio de todos los departamentos. Algunas de las ventajas destacables de tener una base de datos centralizada es la facilidad de traspasar información de un sitio a otro sin la necesidad de mecanizarla manualmente. Además, también ofrece la posibilidad de poder racionalizar el uso del tiempo y desarrollar determinadas tareas con mayor rapidez, por ejemplo, a la hora de procesar datos por parte del personal de contabilidad.
- Automatización de procesos: un ERP automatiza procesos manuales, reduciendo así la necesidad de intervención humana. Ello otorga una ventaja para la empresa, liberando así tiempo del personal para poder realizar otras labores estratégicas y rentables para el negocio.
- Acceso a la información en tiempo real: un ERP es capaz de proporcionar información actualizada en tiempo real mediante informes y paneles de control. El poder trabajar con datos en tiempo real ofrece los siguientes beneficios:
  - Ayuda a minimizar los costes en diferentes áreas de la organización.
  - Mejora la toma de decisiones, haciendo que ésta sea más rápida.
  - Ofrece un tiempo de respuesta rápido y soluciones más adecuadas en situaciones de difícil reacción.
  - Permite la adaptación de la empresa a los cambios evitando situaciones que pueden repercutir negativamente en el beneficio, facilitando así la observación de cambios en el mercado.
  - Permite acceso a la información a cualquier hora desde cualquier parte del mundo mediante sistemas de almacenamiento en la nube.
  - Disminuye la probabilidad de obtener errores en procesos de comunicación y operativos.
- Personalización: dicho ERP puede ser personalizado para conseguir adaptarse a las necesidades específicas de la empresa. [2] [3]

Por otro lado, no todo van a ser ventajas al utilizar un ERP. A continuación, se muestran algunas de las desventajas a tener en cuenta si se hace uso de éste:

- Requiere actualizaciones constantes que exige una nueva capacitación del personal traduciéndose así en un gasto adicional.
- Implica tener que rediseñar procesos de trabajo si se quiere llevar a cabo una transición hacia dicho software para no dejar de ser eficientes.
- Conlleva un proceso de adaptación por parte de los trabajadores traduciéndose así en tiempo a invertir, debido a las diversas funcionalidades que provee éste.
- Exige un equipo considerable de desarrollo para poder realizar el despliegue de la herramienta, así como la configuración de las funciones a utilizar por la empresa.
- Coste inicial alto.
- Los beneficios no son inmediatos.

Tras haberse expuesto toda la información acerca de lo que es un ERP y sus principales características, se mostrará un ranking de los ERP que más utilizan las 1000 empresas que más facturan.



**Figura 2. ERPS utilizados por las 1000 empresas que más facturan.**

Fuente: [www.mundoerp.com/](http://www.mundoerp.com/)

Como se puede observar en la Figura 2., el ERP más utilizado y, por tanto, más importante en la actualidad es SAP, siendo implementado más que todos los otros ERPS juntos y liderando así el mercado. A continuación, se pasará a hablar en profundidad de lo que es SAP, así de que prestaciones nos ofrece dicho software. [4]

## **2.2. ¿Qué es SAP?**

Dichas siglas responden a “Systems, Applications and Products”. Viene a ser una multinacional nacida en Alemania en el año 1972. Presentando su software original SAP R/2 y SAP R/3, se estableció el estándar global para el software de planificación de recursos empresariales. Actualmente, el nuevo software SAP S/4HANA tiene la finalidad de procesar cantidades abundantes de datos al igual que de dar soporte a tecnologías más avanzadas como son la inteligencia artificial y machine learning.

El principal objetivo de dicho software es el de mejorar y facilitar la gestión de todos los ámbitos de una empresa tales como la financiación, producción, entre otros. La idea es ser capaces de tener un sistema de administración de la empresa con la capacidad de unir toda la información de todos los departamentos. [5]

A continuación, se exponen algunos de los beneficios de usar SAP:

- **Visibilidad y accesibilidad:** como ya se ha comentado anteriormente en las características de un ERP, SAP ayuda a los empleados de su empresa a acceder a los datos del software desde cualquier localización y dispositivo al usarse una ubicación centralizada. Ello permite mejorar tanto la comunicación como la colaboración entre departamentos de la organización haciendo que los flujos de trabajo sean más coherentes ayudando a la finalización de las tareas. Como resultado se obtiene un mejor seguimiento de procesos.
- **Mejora de eficiencia:** SAP excluye las operaciones repetitivas haciendo así que no exista la necesidad de introducir manualmente los datos ayudando así a aumentar la productividad general y a reducir la posibilidad de que existan datos inexactos resultando en errores relevantes.
- **Sistema de información unificado:** SAP ofrece un sistema de informes unificado para cada proceso. El hecho de usar una fuente de datos precisos y actualizados permite llevar a cabo un análisis más exacto, sin la ayuda de un experto en el área. Con dicho sistema, se puede comparar y analizar de una mejor forma el trabajo de diferentes departamentos.
- **Seguridad de datos mejorada:** SAP pone al servicio de quien lo utiliza de sistemas de seguridad y firewalls integrados. Las funciones del software ofrecen un refuerzo automático para evitar fallos.
- **Personalización:** SAP ofrece flexibilidad y adaptabilidad para poder adaptarse a los requisitos de la empresa que lo use.

## **2.3. ¿Qué utilidad tiene el software de SAP?**

Se puede utilizar para la recopilación y procesado de datos de todas las funciones todo ello en una sola plataforma, no necesitando así varias soluciones de software para realizar diversas tareas. A continuación, se expone la utilidad del software según las necesidades que se quieran cubrir.

En el caso de ser un director de proyectos, puede utilizar las herramientas que ofrece SAP para ayudar a la gestión de sus proyectos. Se puede dirigir fondos y recursos donde se requieran al igual que controlar el tiempo de entrega y que dicho proyecto se encuentre dentro de los límites del presupuesto. Se puede definir tanto una fecha de inicio como de finalización del proyecto y estructurar cada etapa de éste.

Por otro lado, en el área de los recursos humanos hay varias funcionalidades que ofrece SAP. Para la gestión organizativa, se permite la creación de informes de la estructura organizativa, así como de realizar seguimientos de los puestos en diversos departamentos entre otros. También es útil para registrar datos de empleados, la administración de gastos de viajes y la contratación de éstos. Existe la opción de monitorear turnos, controlar la asistencia y horarios de trabajo. Además, ayuda a la hora de procesar pagos por el trabajo realizado por los empleados.

En el área de almacenaje, SAP permite el procesado de entrada y salida en el almacén, la gestión de operaciones y almacenamiento.

En cuanto a las relaciones con clientes, SAP ofrece ayuda al equipo de marketing y ventas de la empresa a llevar a cabo una optimización de los procesos automatizados.

Un último punto a destacar es que SAP ayuda a realizar un seguimiento de las finanzas incluyendo una planificación y análisis financiero, gestión contable y fiscal, gestión de tesorería y cuentas por pagar y cobrar, todo ello al obtener acceso a varias funcionalidades tanto de gestión financiera como de contabilidad. [6]

#### ***2.4. Módulos del programa***

El sistema está compuesto por diversos componentes básicos funcionales y algunos técnicos. A continuación, se nombrará cada uno de ellos, así como la descripción pertinente.

##### Módulo de gestión de almacenes (WM):

Proporciona funciones y procesos para administrar y controlar eficientemente las operaciones logísticas de un almacén. Éste se utiliza para automatizar y optimizar las actividades de almacenamiento, movimiento y seguimiento de materiales en un almacén. Las principales funcionalidades de éste son: gestión de ubicaciones, recepción y expedición de mercancías, gestión de inventario, control de stock, picking y embalaje, entre otras.

La aplicación a desarrollar en el presente trabajo tratará este módulo.

##### Módulo de finanzas (FI):

Éste sirve para gestionar cada una de las transacciones contables y operaciones financieras en la organización. Dentro de este módulo, la información financiera se encuentra disponible en tiempo real.

#### Módulo de gestión de costes (CO):

Organiza la información de los gastos de una empresa, haciendo uso de las clases y centros de costes, centro de beneficios, entre otros. La información de la que se dispone en este módulo junto con la del componente de finanzas otorga la capacidad de realizar análisis detallados de la optimización de costes. Gracias a este módulo, se simplifica la toma de decisiones.

#### Módulo de ventas (SD):

Donde se gestionan los principales procesos de distribución, así como de ventas tanto de productos como de servicios. Junto con el módulo de gestión de materiales forman el módulo de logística.

#### Módulo de gestión de materiales (MM):

Consiste en la realización del inventario de los materiales y servicios, así como del aprovisionamiento de la empresa. También incluye la gestión de stocks mediante la salida, entrada y traslado de mercancías que se puedan realizar entre diferentes centros y almacenes ubicados en localizaciones diferentes. Además, se incluye la verificación de facturas de proveedores.

#### Módulo de gestión de recursos humanos (HCM):

Este módulo ayuda con las gestiones que influyen en los empleados de la empresa, tales como el control de las nóminas, vacaciones, gestión del tiempo, planificación del trabajo, entre otras.

#### Módulo de planificación y producción (PP):

Aquí se vigila la producción de la empresa y su rendimiento. Apoyándose en los datos generales, se consigue mejorar la capacidad de fabricación. Este módulo interviene en procesos de creación de demanda y de la capacidad de fabricación, garantizando así la elaboración de productos, venta y distribución en los tiempos requeridos. También se ofrecen un centro de trabajo, sistemas de planificación de la fabricación y herramientas para la gestión de procesos industriales.

#### Módulo de gestión de calidad (QM):

Gestiona todos los procesos relacionados con el control de la calidad de los procesos, instalaciones o del producto. Permite la realización de auditorías internas o externas para asegurar la calidad de los productos. Actividades principales de este módulo pueden ser la gestión de lote de inspección, entrada de resultados y defectos y muchas más.

#### Módulo de mantenimiento (PM):

La finalidad de este módulo no es más que mantener en buenas condiciones tanto la maquinaria como las instalaciones para que éstas funciones correctamente. Además, éste también ayuda con el mantenimiento de procesos técnicos. Una actividad adicional que se realiza en este módulo es la de conocer la carga real de trabajo para conseguir una planificación de trabajos. Todo ello, permitiría tomar decisiones más y con fundamento acertadas por medio de los informes estadísticos pertinentes.

### Módulo de gestión de proyectos (PS):

Está orientado para proyectos grandes con procesos de coste elevado y complejos de gestionar. Tiene la función de controlar y garantizar que el trabajo se realice dentro del tiempo acordado planificando todas las fases de desarrollo de éste.

### Módulo de servicio al cliente (CS):

Tiene el objetivo de resolver eficientemente los problemas de los clientes y el cumplimiento de sus solicitudes, optimizando así la experiencia del usuario. Es capaz de gestionar de forma rápida y eficaz todas las etapas que componen la gestión de servicios.

[7]

## **2.5. SAP Fiori**

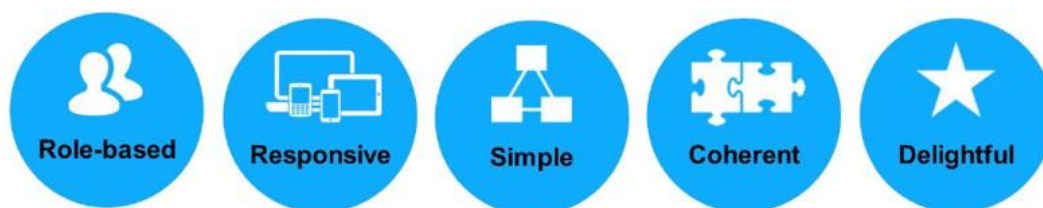
### *2.5.1. Definición*

Se trata de un sistema de diseño que permite la creación de aplicaciones de negocio con una experiencia de usuario a nivel de consumidor. Desde el punto de vista del usuario, consiste en una nueva interfaz web para interactuar con el sistema SAP ya conocido.

Cabe destacar que éste surgió de la necesidad de mejorar la experiencia del usuario haciendo que ésta fuera más intuitiva, accesible e interactiva.

Fiori se compone de unos rasgos muy destacables, estos son:

- Está basado en roles
- Es simple al centrarse en lo importante.
- Adaptativo, pues se adapta según tu forma de trabajar y dónde lo hagas en cualquier momento.
- Coherente, ya que provee de una experiencia fluida e intuitiva.
- Encantador al crear una conexión emocional.



**Figura 3. Características destacables de Fiori.**

Fuente: <https://blogs.sap.com/>



### 2.5.2. Utilidad

SAP Fiori se utiliza para optimizar los procesos de las empresas otorgándoles así una experiencia de usuario cómoda, sirviéndose de las conexiones en línea para trabajar en distintos dispositivos. Ésta viene a ser una herramienta de gestión de procesos complementaria ideal para conseguir mejorar el rendimiento de la empresa.

Fiori funciona haciendo uso de un solo dispositivo electrónico, así como de una conexión a internet estable, ya que se ingresa mediante vía web con usuario y contraseña. Aquí es donde entra en juego el Launchpad de Fiori, en el cual se encontrarán las aplicaciones que tienen relación con el rol de trabajo definido. Se puede acceder a la información del estado operativo de la tarea y los detalles de ésta.

### 2.5.3. Arquitectura

SAP Fiori está compuesto por diferentes capas. A continuación, detallaremos cuales son y que utilidad tienen.

Cliente/navegador: es la capa donde se ejecutan las aplicaciones SAP Fiori, siendo el Launchpad de SAP Fiori el punto de entrada para todas las aplicaciones de SAP Fiori en diferentes dispositivos.

Proxy inverso: se sitúa detrás del firewall en una red privada generalmente. Tiene la obligación de redireccionar las peticiones de los clientes al servidor correcto con la protección adicional pertinente para garantizar el flujo fluido del tráfico de red entre clientes y servidores. SAP recomienda el uso de SAP Web Dispatcher como servidor proxy inverso, éste solo se requiere como proxy inverso para aplicaciones analíticas y de hojas informativas.

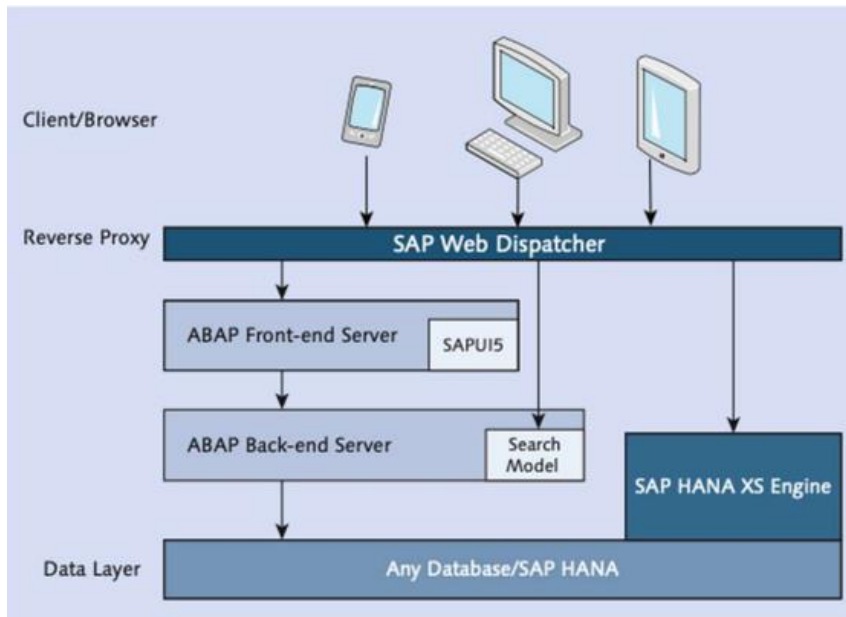
Servidor back-end: puede funcionar en cualquier base de datos mientras se trate de aplicaciones de tipo transaccional. Contiene lógica empresarial back-end y modelos de búsqueda de aplicaciones de hoja informativa.

Servidor front-end: recomendado para todo tipo de aplicaciones. Contiene los componentes de infraestructura, capa de interfaz de usuario y la puerta de enlace SAP con la habilitación del servicio OData.

SAP HANA XS: únicamente obligatorio para aplicaciones analíticas. Tiene la función de replicar datos del servidor back-end y de remodelarlos usando modelos de datos virtuales.

Base de datos back-end: puede ser cualquier base de datos para aplicaciones transaccionales o una base de datos SAP HANA para hoja informativa y aplicaciones analíticas.

A continuación, en la Figura 4. se muestra más detalladamente la arquitectura definida en los puntos anteriores.



**Figura 4. Arquitectura de SAP Fiori.**

**Fuente: SAP Fiori Implementation and Development**

## **2.6. SAP Web IDE**

Cabe destacar que, para poder desarrollar la aplicación pedida por el cliente, en cuanto al front-end, se desarrollará en SAP Web IDE. Éste es uno de los servicios que ofrece SAP Cloud Platform y es conocida como una potente herramienta de desarrollo de aplicaciones de extremo a extremo basada en el navegador que permite crear y desarrollar las aplicaciones SAP Fiori. Las características más destacables de SAP Web IDE son las siguientes:

- Es un editor gráfico con funciones que permiten arrastrar y soltar para crear la interfaz de usuario.
- Proporciona diversas plantillas predefinidas, así como aplicaciones de referencia para desarrolladores acerca de cómo implementar las mejores prácticas.
- Ofrece capacidad de vista previa de la aplicación a incorporar con una función para elegir como entornos de destino diferentes dispositivos como son ordenadores, teléfonos móviles, entre otros.

## 2.7. SAPUI5

Para la creación de aplicaciones Fiori se utilizará SAPUI5, siendo éste un framework conformado por un conjunto de librerías de Javascript para el desarrollo de interfaces de usuario en aplicaciones de negocios modernas soportadas sobre entornos web. Éste permite crear aplicaciones una sola vez y ejecutarlas en cualquier plataforma o dispositivo. Las ventajas que ofrece SAPUI5 sobre otros frameworks son las siguientes:

- Simplifica el manejo y coherencia de los datos a través de “2 way data binding”, la cual es una técnica que permite mantener sincronizada la información entre una interfaz de usuario y una fuente de datos.
- Facilita el aprendizaje y su uso debido a que se basa en estándares Open Source como HTML5, Javascript, XML, entre otros.
- Mantiene una única base de código, no haciendo falta codificación adicional en las aplicaciones HTML5 para funcionar en diferentes idiomas y tipos de dispositivos.
- Mejora el rendimiento de las aplicaciones mediante la carga asíncrona de recursos.

### 2.7.1. Implementación de una aplicación SAPUI5

Tres tipos de archivos básicos en una aplicación SAPUI5:

#### Descriptor (manifest.json)

Es un archivo de configuración importante utilizado para describir y definir las propiedades de una aplicación SAPUI5. Proporciona metadatos y configuraciones que controlan el comportamiento y apariencia de la aplicación. Se utiliza en el marco de la arquitectura de modelo-vista-controlador de SAPUI5 y se ubica en la raíz de la estructura de carpetas de una aplicación SAPUI5. Algunas de las utilidades de este archivo son:

- Permite la configuración de aspectos como son el título, la descripción, el icono, el autor y la versión de la aplicación.
- Define rutas de navegación y vistas asociadas, lo que permite la navegación basada en URL dentro de la aplicación y la carga dinámica de las vistas correspondientes.
- Permite la declaración de modelos de datos, facilitando así en gran medida la gestión de datos en la aplicación. Cabe destacar que los modelos declarados en este archivo se pueden utilizar en las vistas y controladores de la aplicación sin tener la necesidad de crearlos de forma manual.
- Se pueden especificar las bibliotecas y recursos adicionales necesarios para la aplicación.
- En cuanto al inicio de la aplicación, este archivo permite la definición de la configuración del lanzamiento de la aplicación, así como de la clase inicial y otros parámetros de gran importancia.

### Componente (Component.js)

Contiene la configuración de la aplicación. Algunas de sus principales utilidades son las siguientes:

- Es donde se definen y configuran aspectos como el modelo de datos, ruta de la aplicación, bibliotecas utilizadas y otros parámetros relevantes.
- Define los métodos y eventos que gestionan el ciclo de vida de la aplicación, incluyendo eventos como “init”, “exit”, entre otros. Éstos son utilizados para inicializar y configurar la aplicación, crear y eliminar componentes, así como para llevar a cabo tareas específicas en según qué etapa del ciclo de vida de la aplicación.
- Mediante este archivo, se establece la configuración inicial de los modelos de datos y se realiza el enlace de datos en la aplicación, permitiendo así una buena propagación de los datos hacia las vistas y los controladores de la aplicación.
- Es utilizado para la configuración del enrutamiento y la navegación en la aplicación SAPUI5.
- Permite la personalización y extensibilidad, siendo posible la adaptación de la aplicación a las necesidades específicas del proyecto y la agregación de funcionalidades adicionales.

### Vista raíz (App.view.xml)

Aquí se define la estructura y diseño de la vista principal de la aplicación. Proporciona una representación visual de la interfaz de usuario definiendo los controles y componentes que componen la pantalla principal de la aplicación. Algunas de las utilidades y funciones del archivo son las siguientes:

- Se pueden agregar controles, componentes como botones, listas, formularios, tablas y muchos más elementos visuales para llegar a tener la disposición deseada.
- Permite el establecimiento de datos entre controles y modelos de datos de la aplicación, por tanto, las propiedades de los modelos pueden estar vinculadas a los valores de los controladores, permitiendo así la visualización y manipulación de los datos en tiempo real.
- Se pueden especificar qué acciones deben ocurrir cuando un usuario interactúa con los controles como, por ejemplo, al seleccionar un elemento de una lista o presionar un botón.
- Permite definir plantillas reutilizables para mostrar elementos o formularios que se repitan, facilitando en gran medida la generación dinámica de contenido que se basa en modelos de datos y mejora la eficiencia del desarrollo.
- Admite la internacionalización de las aplicaciones. El archivo se puede usar para definir y gestionar la traducción y localización de textos utilizados en la interfaz de usuario. Esto es una ventaja clave debido a que se puede adaptar la aplicación a diferentes culturas e idiomas. [5]

## ***2.8. Lenguaje de programación ABAP***

Éste viene a ser un lenguaje de programación propiedad de SAP de cuarta generación que se utiliza para programar la mayoría de sus productos. Con este lenguaje de programación, los programadores pueden crear nuevas funcionalidades, realizar modificaciones en los procesos existentes, interactuar con bases de datos y desarrollar informes y formularios personalizados. También es utilizado para integrar aplicaciones externas con los sistemas SAP.

ABAP se perfila mayormente como un lenguaje back-end, el cual posee una sintaxis similar a otros lenguajes de programación como, por ejemplo, C, lo que facilita en parte el aprendizaje y transición para los programadores familiarizados con estos lenguajes.

Por otro lado, también es interesante saber el tipo de ventajas que te proporciona dicho lenguaje de programación, por ello a continuación, se detallan algunas de ellas:

- ABAP se encuentra estrechamente integrado con los sistemas SAP, ello permite acceder y manipular fácilmente los datos y objetos del sistema. Esto permite el desarrollo de soluciones muy personalizadas y adaptadas a las necesidades de cada organización.
- ABAP ofrece un gran conjunto de bibliotecas de funciones y clases proporcionadas por SAP. Dichas bibliotecas cubren una amplia gama de funcionalidades empresariales, facilitando así el desarrollo de aplicaciones completas y sofisticadas sin tener que empezar un código desde cero.
- Se ofrece la reutilización de código reduciendo así el tiempo en el desarrollo de aplicaciones y mejorando a su vez la eficiencia.
- Existencia de herramientas de desarrollo con características avanzadas como es el seguimiento de errores y la capacidad de realizar pruebas exhaustivas, facilitando de esta manera la identificación y solución de problemas.
- ABAP se puede extender hacia otros sistemas y tecnologías, facilitando así la integración de aplicaciones externas con los sistemas SAP.
- Se puede acceder a recursos en línea, foros de discusión y documentación detallada, facilitando en gran parte la solución de los problemas. [8]

## *Capítulo 3. Desarrollo de la aplicación*

En este capítulo, se va a explicar cada uno de los pasos dados para llevar a cabo la elaboración de la aplicación, analizando en profundidad aspectos como pueden ser tanto los códigos creados en el front-end como en el back-end.

### *3.1. Descripción de la aplicación*

El objetivo de la aplicación es el conseguir que los operarios de una empresa en particular gestionen el abastecimiento de una serie de líneas de producción a través de sus dispositivos PDA en un centro de trabajo en concreto. Cabe destacar que dicho objetivo se ha establecido tras conocer los requerimientos de la aplicación que han sido proporcionados por el cliente y redactados en un documento por el consultor funcional, el cual reúne cada una de las funciones que debe tener la aplicación a desarrollar. Dicho documento funcional es transmitido al consultor técnico que se encargará del desarrollo de dicha aplicación.

Una vez se ha analizado el documento funcional, se ha conocido que dicha aplicación debe mostrar todas las órdenes de abastecimiento en función de la selección de tres filtros. Éstos vienen a ser uno que filtra por el número de orden, otro que filtra por el puesto de trabajo correspondiente a las líneas a abastecer y el número de almacén. Éste último se encontrará fijo ya que el número de almacén es el 800 (con el cual se identifica físicamente un almacén en un sitio) para todos los lotes de bobinas a reubicar si fuera necesario.

El listado de órdenes mostrará una serie de campos de información, estos son los siguientes:

- Puesto de trabajo: identifica la línea a abastecer.
- Orden: muestra el número de orden.
- Material: muestra el tipo de material de esa orden.
- Lote: identifica que lote de bobinas se quiere ubicar en la línea de abastecimiento correspondiente.
- Ubicación: muestra la ubicación actual de dicho lote de bobinas.
- Tipo de almacén: utilizados para agrupar y categorizar almacenes similares en términos de su uso, capacidad, restricciones o reglas operativas. Por ejemplo, los tipos de almacén pueden ser "almacén de productos terminados", "almacén de materias primas" o "almacén de productos peligrosos". Éste vendrá dado por un identificador.
- Operación: dispondrá de un botón que ejecute ese movimiento de traslado de lote desde una ubicación de logística (campo ubicación) a una ubicación específica creada para los puestos de trabajo (CUNAS\_D, CUNAS\_G o CUNAS\_L). En el momento de activar el botón, se activará una función que realizará la acción y mostrará un mensaje que confirme que el lote de bobinas fue ubicado exitosamente y éste desaparecerá del listado de órdenes.

El proyecto a llevar a cabo consiste en desarrollar esta aplicación multiplataforma, tanto su parte front-end como back-end siguiendo con los principios Fiori marcados por SAP.

En primer lugar, su parte front-end será desarrollada con el framework SAPUI5 diseñado para crear aplicaciones multiplataforma y responsivas del lado del cliente y que permite cumplir con los principios de experiencia de usuario SAP UX.

Y, por otro lado, la parte back-end que se encuentra en el sistema SAP de la empresa cliente se encarga de procesar y proporcionar los datos que alimentan el front-end. Se utilizarán las herramientas SAP necesarias para la creación de estructura de datos, modelo de datos y servicio oData. Cabe destacar que para las funciones back-end que hagan la lógica del movimiento de traspaso de los lotes de bobinas se utilizará el lenguaje de programación ABAP.

Por último, se deberá de establecer la comunicación entre el front-end y la fuente de datos que se encuentra en el back-end. Para ello se utilizará el protocolo oData (Open Data Protocol). Asimismo, una vez la aplicación se encuentre terminada y operativa, será desplegada en la plataforma SAP Fiori Launchpad de la empresa cliente.

### ***3.2. Cloud Connector y creación del destino***

El Cloud Connector se trata de una herramienta de conectividad proporcionada por SAP. Dicha herramienta tiene la función de actuar como un puente local entre los sistemas back-end locales y las aplicaciones en la nube.

Éste permite el establecimiento de una conexión confiable y segura entre los diferentes sistemas internos y los servicios en la nube, de esta forma se evita el exponer directamente los sistemas back-end a Internet.

Los pasos que se seguirán para la configuración del Cloud Connector son los siguientes:

- Descargar e instalar el Cloud Connector en el entorno local.
- Acceder al Cloud Connector a través de la interfaz web y crear un nuevo subdominio.
- En este caso se optará por un Sistema back-end de tipo SAP Gateway con protocolo HTTP. Donde se añadirá el servidor interno proporcionado por el cliente y una descripción del sistema que se quiere añadir.
- En cuanto a la URL se añadirá la ruta absoluta acceder al sistema back-end.

Ahora será necesario la creación de un destino cliente, ya que el Cloud Connector y la creación de un destino están directamente relacionados en el contexto de desarrollo de una aplicación SAPUI5 en el front-end. La creación de este destino define la configuración de conexión específica para acceder al sistema back-end.

Una vez creado el destino y configurado el Cloud Connector, se realiza tanto una prueba de conexión para verificar que ésta es estable entre la aplicación front-end y el Cloud Connector, así como para verificar la conexión entre el Cloud Connector y el sistema back-end al que se está accediendo.

Por último, quedaría en la creación del proyecto en el apartado de conexión, introducir el destino cliente creado.

### 3.3. Creación de la estructura y tipos de tabla del modelo de datos

Una vez determinada la función de la aplicación y en qué va a consistir, se profundizará en cada uno de los pasos a seguir, comenzando por la creación de una estructura para el modelo de datos que a posteriori se creará y se publicará para su posterior prueba en front-end.

Se accederá al sistema proporcionado por el cliente especificando el servidor de aplicación, el entorno (en este caso del de desarrollo) y el número de puerto al que se realizará la conexión. Tras tener dichos datos, mediante una VPN a la propia empresa y tras el establecimiento de la conexión en la aplicación SAP GUI, se entrará al sistema.

Una vez dentro del sistema, se accederá a la transacción “se80” para crear el paquete a utilizar, dicho paquete se utilizará para gestionar los objetos durante el desarrollo, transporte y mantenimiento en el entorno SAP.

Para el caso a tratar, se creará un paquete llamado ZWM\_ABASTECER\_PDA\_LCS, donde WM se refiere a *Warehouse Management* (gestión de almacenes).

A continuación, se creará la estructura para el almacén, línea de producción y órdenes. Dichas estructuras, deberán tener todos los campos que se quieran devolver, así como los campos de entrada pertinentes. Cabe destacar, que dichos campos se sabrán a partir de la información proporcionada por el cliente. Dichas estructuras se pueden observar en las Tablas 1, 2 y 3.

Componente	Clase tipificación	Tp.componente	Tipo de datos	Long.	Decimales	Descripción breve
WERKS	Type	WERKS_D	CHAR	4	0	Centro
LGNUM	Type	LGNUM	CHAR	3	0	Núm.almacén/Complejo alm.
NAME1	Type	NAME1	CHAR	30	0	Nombre

**Tabla 1. Creación de la estructura de almacén.**

**Fuente: Elaboración propia.**

Componente	Clase tipificación	Tp.componente	Tipo de datos	Long.	Decimales	Descripción breve
PRVBE	Type	PRVBE	CHAR	10	0	Área de suministro de producción

**Tabla 2. Creación de la estructura de línea de producción.**

**Fuente: Elaboración propia.**



Estructura	ZWM_ABAST_S_ORDEN	activo																																																																																																																																																																	
Descripción breve	Almacén para app de abastecer líneas (León)																																																																																																																																																																		
Atributos	Componentes	Ayuda p./Verif.entr. Campos de moneda/cantidad																																																																																																																																																																	
<div style="text-align: right;">Tipo instalado 22</div> <table border="1"> <thead> <tr> <th>Componente</th> <th>Clase tipificación</th> <th>Tp.componente</th> <th>Tipo de datos</th> <th>Long.</th> <th>Decimales</th> <th>Descripción breve</th> </tr> </thead> <tbody> <tr> <td>STAT</td> <td>Type</td> <td>J STATUS</td> <td>CHAR</td> <td>5</td> <td>0</td> <td>Status individual de un objeto</td> </tr> <tr> <td>TXTO4</td> <td>Type</td> <td>J TXT04</td> <td>CHAR</td> <td>4</td> <td>0</td> <td>Status individual de un objeto (breve)</td> </tr> <tr> <td>LGTYP</td> <td>Type</td> <td>LGTYP</td> <td>CHAR</td> <td>3</td> <td>0</td> <td>Tipo almacén</td> </tr> <tr> <td>SPRAS</td> <td>Type</td> <td>SPRAS</td> <td>LANG</td> <td>1</td> <td>0</td> <td>Clave de idioma</td> </tr> <tr> <td>LGNUM</td> <td>Type</td> <td>LGNUM</td> <td>CHAR</td> <td>3</td> <td>0</td> <td>Núm.almacén/Complejo alm.</td> </tr> <tr> <td>LGPLA</td> <td>Type</td> <td>LGPLA</td> <td>CHAR</td> <td>10</td> <td>0</td> <td>Ubicación</td> </tr> <tr> <td>LGORT</td> <td>Type</td> <td>LGORT D</td> <td>CHAR</td> <td>4</td> <td>0</td> <td>Almacén</td> </tr> <tr> <td>WERKS</td> <td>Type</td> <td>WERKS D</td> <td>CHAR</td> <td>4</td> <td>0</td> <td>Centro</td> </tr> <tr> <td>PRVBE</td> <td>Type</td> <td>PRVBE</td> <td>CHAR</td> <td>10</td> <td>0</td> <td>Área de suministro de producción</td> </tr> <tr> <td>MATNR</td> <td>Type</td> <td>MATNR</td> <td>CHAR</td> <td>40</td> <td>0</td> <td>Número de material</td> </tr> <tr> <td>MAKTX</td> <td>Type</td> <td>MAKTX</td> <td>CHAR</td> <td>40</td> <td>0</td> <td>Texto breve de material</td> </tr> <tr> <td>MENGE</td> <td>Type</td> <td>BSTMG</td> <td>QUAN</td> <td>13</td> <td>3</td> <td>Cantidad de pedido</td> </tr> <tr> <td>MEINS</td> <td>Type</td> <td>MEINS</td> <td>UNIT</td> <td>3</td> <td>0</td> <td>Unidad de medida base</td> </tr> <tr> <td>CHARG</td> <td>Type</td> <td>CHARG D</td> <td>CHAR</td> <td>10</td> <td>0</td> <td>Número de lote</td> </tr> <tr> <td>AUFNR</td> <td>Type</td> <td>AUFNR</td> <td>CHAR</td> <td>12</td> <td>0</td> <td>Número de orden</td> </tr> <tr> <td>CY_SEQNR</td> <td>Type</td> <td>CY_SEQNR</td> <td>NUMC</td> <td>14</td> <td>0</td> <td>Número de secuencia de orden</td> </tr> <tr> <td>TIPO_CHARACTER</td> <td>Type</td> <td>CHAR70</td> <td>CHAR</td> <td>70</td> <td>0</td> <td>Campo de texto, longitud 70</td> </tr> <tr> <td>BDMNG</td> <td>Type</td> <td>BDMNG</td> <td>QUAN</td> <td>13</td> <td>3</td> <td>Cantidad necesaria</td> </tr> <tr> <td>TIPO_ABASTECIDO</td> <td>Type</td> <td>CHAR1</td> <td>CHAR</td> <td>1</td> <td>0</td> <td>Indicador de una posición</td> </tr> <tr> <td>ABASTECIDA</td> <td>Type</td> <td>CHAR1</td> <td>CHAR</td> <td>1</td> <td>0</td> <td>Indicador de una posición</td> </tr> <tr> <td>ZPRIMAVERA</td> <td>Type</td> <td>CHAR21</td> <td>CHAR</td> <td>21</td> <td>0</td> <td>Nombre campo</td> </tr> <tr> <td>RSPOS</td> <td>Type</td> <td>RSPOS</td> <td>NUMC</td> <td>4</td> <td>0</td> <td>Número de posición de reserva/necesidades secundarias</td> </tr> </tbody> </table>			Componente	Clase tipificación	Tp.componente	Tipo de datos	Long.	Decimales	Descripción breve	STAT	Type	J STATUS	CHAR	5	0	Status individual de un objeto	TXTO4	Type	J TXT04	CHAR	4	0	Status individual de un objeto (breve)	LGTYP	Type	LGTYP	CHAR	3	0	Tipo almacén	SPRAS	Type	SPRAS	LANG	1	0	Clave de idioma	LGNUM	Type	LGNUM	CHAR	3	0	Núm.almacén/Complejo alm.	LGPLA	Type	LGPLA	CHAR	10	0	Ubicación	LGORT	Type	LGORT D	CHAR	4	0	Almacén	WERKS	Type	WERKS D	CHAR	4	0	Centro	PRVBE	Type	PRVBE	CHAR	10	0	Área de suministro de producción	MATNR	Type	MATNR	CHAR	40	0	Número de material	MAKTX	Type	MAKTX	CHAR	40	0	Texto breve de material	MENGE	Type	BSTMG	QUAN	13	3	Cantidad de pedido	MEINS	Type	MEINS	UNIT	3	0	Unidad de medida base	CHARG	Type	CHARG D	CHAR	10	0	Número de lote	AUFNR	Type	AUFNR	CHAR	12	0	Número de orden	CY_SEQNR	Type	CY_SEQNR	NUMC	14	0	Número de secuencia de orden	TIPO_CHARACTER	Type	CHAR70	CHAR	70	0	Campo de texto, longitud 70	BDMNG	Type	BDMNG	QUAN	13	3	Cantidad necesaria	TIPO_ABASTECIDO	Type	CHAR1	CHAR	1	0	Indicador de una posición	ABASTECIDA	Type	CHAR1	CHAR	1	0	Indicador de una posición	ZPRIMAVERA	Type	CHAR21	CHAR	21	0	Nombre campo	RSPOS	Type	RSPOS	NUMC	4	0	Número de posición de reserva/necesidades secundarias
Componente	Clase tipificación	Tp.componente	Tipo de datos	Long.	Decimales	Descripción breve																																																																																																																																																													
STAT	Type	J STATUS	CHAR	5	0	Status individual de un objeto																																																																																																																																																													
TXTO4	Type	J TXT04	CHAR	4	0	Status individual de un objeto (breve)																																																																																																																																																													
LGTYP	Type	LGTYP	CHAR	3	0	Tipo almacén																																																																																																																																																													
SPRAS	Type	SPRAS	LANG	1	0	Clave de idioma																																																																																																																																																													
LGNUM	Type	LGNUM	CHAR	3	0	Núm.almacén/Complejo alm.																																																																																																																																																													
LGPLA	Type	LGPLA	CHAR	10	0	Ubicación																																																																																																																																																													
LGORT	Type	LGORT D	CHAR	4	0	Almacén																																																																																																																																																													
WERKS	Type	WERKS D	CHAR	4	0	Centro																																																																																																																																																													
PRVBE	Type	PRVBE	CHAR	10	0	Área de suministro de producción																																																																																																																																																													
MATNR	Type	MATNR	CHAR	40	0	Número de material																																																																																																																																																													
MAKTX	Type	MAKTX	CHAR	40	0	Texto breve de material																																																																																																																																																													
MENGE	Type	BSTMG	QUAN	13	3	Cantidad de pedido																																																																																																																																																													
MEINS	Type	MEINS	UNIT	3	0	Unidad de medida base																																																																																																																																																													
CHARG	Type	CHARG D	CHAR	10	0	Número de lote																																																																																																																																																													
AUFNR	Type	AUFNR	CHAR	12	0	Número de orden																																																																																																																																																													
CY_SEQNR	Type	CY_SEQNR	NUMC	14	0	Número de secuencia de orden																																																																																																																																																													
TIPO_CHARACTER	Type	CHAR70	CHAR	70	0	Campo de texto, longitud 70																																																																																																																																																													
BDMNG	Type	BDMNG	QUAN	13	3	Cantidad necesaria																																																																																																																																																													
TIPO_ABASTECIDO	Type	CHAR1	CHAR	1	0	Indicador de una posición																																																																																																																																																													
ABASTECIDA	Type	CHAR1	CHAR	1	0	Indicador de una posición																																																																																																																																																													
ZPRIMAVERA	Type	CHAR21	CHAR	21	0	Nombre campo																																																																																																																																																													
RSPOS	Type	RSPOS	NUMC	4	0	Número de posición de reserva/necesidades secundarias																																																																																																																																																													

**Tabla 3. Creación de la estructura de órdenes.**

**Fuente: Elaboración propia.**

Tras crear las estructuras de datos, se crearán los tipos de tabla los cuales se encargan de proporcionar una forma organizada de almacenar grandes volúmenes de datos y representan conjuntos de registros con campos relacionados. Tras la elaboración de los pasos previos, se activan y guardan los cambios.

### 3.4. Creación de los módulos de funciones

Al tener ya la estructura del tipo de datos y los tipos de tabla para cada una de las estructuras, se crean los módulos de funciones, donde cada uno de ellos tendrá diferentes funcionalidades. A continuación, se muestra el código de cada uno de ellos y su correspondiente explicación.

La función ZWM\_MF\_GET\_ORDENES (Anexo) se trata de una función que procesa información relacionada con órdenes de producción. Hablando de forma general, este código realiza lo siguiente:

- Extrae información relevante de la base de datos sobre órdenes de producción, como el número de orden, la planta y el estado de la orden.
- Filtra las órdenes de producción obtenidas según ciertos criterios especificados en los rangos “rg\_orden”.
- Se elaboran operaciones de procesamiento de datos en las órdenes de producción restantes.

- Realiza consultas adicionales con el objetivo de obtener información detallada sobre cada orden de producción como, por ejemplo, los datos del puesto de trabajo y los números de lote asociados.
- Utiliza la información obtenida para asignar valores específicos a los campos de las órdenes de producción, como el puesto de trabajo, el número de lote, el tipo de ubicación y la cantidad requerida.
- Realiza verificaciones adicionales en la tabla “lqua” con la finalidad de obtener más detalles acerca de los números de lote y lleva a cabo operaciones de eliminación según ciertos criterios.
- Finalmente, se realiza una clasificación y se eliminan las órdenes de producción que no cumplen con ciertas condiciones especificadas.

La función ZWM\_MF\_SH\_LINEA (Anexo) proporciona una lista predefinida de líneas de producción. Estas líneas de producción se representan mediante una estructura de datos y se agregan a una tabla de salida. La lista final se utiliza para mostrar información relacionada con las líneas de producción en un contexto específico. En resumen, la función genera y devuelve una lista de líneas de producción predefinidas para su posterior visualización o procesamiento.

La función ZWM\_MF\_UBICAR (Anexo) realiza las siguientes acciones a grandes rasgos:

- Se define una serie de variables y estructuras de datos para almacenar información relevante.
- Se llama a la función “bapi\_prodord\_get\_detail” para obtener detalles de una orden de producción específica. Se proporciona el número de orden como entrada y se obtienen datos como la cabecera de la orden, los componentes y las operaciones asociadas.
- Se realizan bucles para recorrer los resultados obtenidos y extraer información específica, como el estado de la orden, el lote de componentes y el puesto de trabajo asociado.
- Se comprueba el estado de la orden y se realizan acciones en consecuencia. Si el estado cumple ciertos criterios, se ejecuta un bloque de código para realizar operaciones adicionales.
- Se realiza una consulta a la tabla “lqua” para obtener información sobre la ubicación de un lote específico.
- Se definen variables y se asignan valores en función del puesto de trabajo y otros criterios.
- Se llama a la función “l\_to\_create\_single” para crear una ubicación logística (l\_to) en base a los datos proporcionados.
- Se manejan las excepciones y se añaden mensajes de retorno en caso de error.

### 3.5. Creación del modelo de datos

Una vez se han definido los tipos de tabla a utilizar con su correspondiente estructura y los módulos de funciones, se pasará a crear el modelo de datos que se utilizará en el servicio web. Esto va a implicar la creación de entidades y atributos que representan la estructura de los datos que se expondrán a través del servicio. Por tanto, en primer lugar, se explicará el proceso de creación de los tipos de entidades importando una estructura de datos.

Se debe de crear un nuevo proyecto en la transacción “segw” y en la opción “Data Model” se deben de importar las estructuras de datos ya creadas previamente para cada una de las entidades. Las propiedades se generarán automáticamente en base a los campos de la estructura de datos importada. Un ejemplo son las propiedades generadas para la entidad de órdenes mostrado en la Tabla 4.

Properties																	
Name	Is Key	Edm Type	Prec.	Scale	Max ...	Unit Prop.	Creat...	Upda...	Sorta...	Nulla...	Filterable	Label	La...	Comp. T...	ABAP Fl...	A...	Semantics
Lgtyp	<input type="checkbox"/>	Edm.Stri...	0	0	3		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Tipo almacén	T			LGTYP	<input type="checkbox"/>
Sprasm	<input type="checkbox"/>	Edm.Stri...	0	0	2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Idioma	T			SPRAS	<input type="checkbox"/>
Lgnum	<input type="checkbox"/>	Edm.Stri...	0	0	3		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Núm.almacén	T			LGNUM	<input type="checkbox"/>
Lgpla	<input type="checkbox"/>	Edm.Stri...	0	0	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Ubicación	T			LGPLA	<input type="checkbox"/>
Lgort	<input type="checkbox"/>	Edm.Stri...	0	0	4		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Almacén	T			LGORT	<input type="checkbox"/>
Werks	<input type="checkbox"/>	Edm.Stri...	0	0	4		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Centro	T			WERKS	<input type="checkbox"/>
Prvbe	<input type="checkbox"/>	Edm.Stri...	0	0	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Línea a abastecer	T			PRVBE	<input type="checkbox"/>
Matnr	<input type="checkbox"/>	Edm.Stri...	0	0	40		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Material	T			MATNR	<input type="checkbox"/>
Maktx	<input type="checkbox"/>	Edm.Stri...	0	0	40		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Denominación	T			MAKTX	<input type="checkbox"/>
Menge	<input type="checkbox"/>	Edm.De...	13	3	0		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Cantidad pedido	T			MENGE	<input type="checkbox"/>
Meins	<input type="checkbox"/>	Edm.Stri...	0	0	3		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	UM base	T			MEINS	<input type="checkbox"/>
Charg	<input type="checkbox"/>	Edm.Stri...	0	0	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Lote	T			CHARG	<input type="checkbox"/>
Aufnr	<input checked="" type="checkbox"/>	Edm.Stri...	0	0	12		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Orden	T			AUFNR	<input type="checkbox"/>
CySeqnr	<input type="checkbox"/>	Edm.Stri...	0	0	14		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Nº de secuencia	T			CY_SEQ...	<input type="checkbox"/>
TipoCar...	<input type="checkbox"/>	Edm.Stri...	0	0	70		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	char70	T			TIPO_C...	<input type="checkbox"/>
Bdmng	<input type="checkbox"/>	Edm.De...	13	3	0		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Ctd.necesaria	T			BDMNG	<input type="checkbox"/>
TipoAba...	<input type="checkbox"/>	Edm.Stri...	0	0	1		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Carácter 1	T			TIPO_A...	<input type="checkbox"/>
Abasteci...	<input type="checkbox"/>	Edm.Stri...	0	0	1		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Carácter 1	T			ABASTE...	<input type="checkbox"/>
Zprimav...	<input type="checkbox"/>	Edm.Stri...	0	0	21		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Nombre campo	T			ZPRIMA...	<input type="checkbox"/>
Rspost	<input type="checkbox"/>	Edm.Stri...	0	0	4		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Posición	T			RSPOS	<input type="checkbox"/>

Tabla 4. Propiedades de la entidad de órdenes.

Fuente: Elaboración propia.

Donde se observa que el campo clave es el de “aufnr”, es decir, el número de pedido. Dicho campo clave es un campo que se utiliza en un modelo de datos para identificar de manera única cada registro en una tabla o entidad. Además, los campos marcados en la opción “Nullable” indican que dicho campo puede ser nulo o cero y los campos marcados por “Filterable”, indican que se puede filtrar por dichos campos.

También se crea la entidad de almacén donde las propiedades de ésta se muestran en la Tabla 5.

Properties																	
Name	Is Key	Edm Type	Prec.	Scale	Max ...	Unit Prop.	Creat...	Upda...	Sorta...	Nulla...	Filterable	Label	La...	Comp. T...	ABAP Fl...	A...	Semantics
Werks	<input checked="" type="checkbox"/>	Edm.Stri...	0	0	4		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Centro	T			WERKS	<input type="checkbox"/>
Lgnum	<input checked="" type="checkbox"/>	Edm.Stri...	0	0	3		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Núm.almacén	T			LGNUM	<input type="checkbox"/>
Name1	<input type="checkbox"/>	Edm.Stri...	0	0	30		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nombre 1	T			NAME1	<input type="checkbox"/>

Tabla 5. Propiedades de la entidad de almacén.

Fuente: Elaboración propia.

En este caso los campos claves son el centro y el número de almacén y ninguno de los campos puede ser filtrable ni nulo.

La siguiente entidad definida es la de línea de producción, donde las propiedades de ésta se muestran en la Tabla 6.

Properties																	
Name	Is Key	Edm Type	Prec.	Scale	Max ...	Unit Prop.	Creat...	Upda...	Sorta...	Nulla...	Filterable	Label	La...	Comp. T...	ABAP Fl...	A...	Semantics
Prvbe	<input checked="" type="checkbox"/>	Edm.Stri...	0	0	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Área sum.prod.	T		PRVBE		<input checked="" type="checkbox"/>

**Tabla 6. Propiedades de la entidad de línea.**

**Fuente: Elaboración propia.**

Donde el campo clave es el del puesto de trabajo, el cual no podrá ser nulo ni filtrable.

Aparte de la creación de las entidades, se debe de crear las asociaciones entre las entidades pertinentes para así poder navegar y acceder a los datos relacionados. En este caso se ha relacionado la entidad almacén con la entidad orden con una cardinalidad de 1 a N queriendo decir que un almacén puede tener varias órdenes asociadas a él, pero cada orden está vinculada a un único almacén.

Por último, se invocará una “Function Import”, la cual ejecutará la lógica necesaria para realizar la reubicación de la orden en el sistema subyacente. Dicha lógica se describe en la redefinición del método “/iwbeb/if\_mgw\_appl\_srv\_runtime~execute\_action” en el back-end de SAP. Dicho método permite implementar la lógica de ejecución de acciones personalizadas.

El método, descrito en el Anexo, toma como parámetros de entrada el nombre de la acción (iv\_action\_name) y una tabla de parámetros (it\_parameter) que contiene los valores necesarios para la acción. Dentro del método, se realiza la manipulación de estos parámetros y se ejecutan las operaciones correspondientes a la acción solicitada.

La lógica de la redefinición puede variar dependiendo de las necesidades del servicio y las acciones definidas. En el ejemplo proporcionado, se verifica si la acción es "Ubicar" y se extrae el número de orden de la tabla de parámetros. Luego, se invoca a una función ZWM\_MF\_UBICAR que realiza la reubicación de la orden. Si no hay mensajes de error, se copian los datos de la orden en una estructura de respuesta.

En caso de que se produzcan errores durante la ejecución de la acción, se generan excepciones para informar de manera adecuada y controlada los problemas encontrados.

Siguiendo con la creación del modelo de datos, se debe de elaborar el mapeo. Esto consiste en realizar la configuración que permite asociar las entidades y propiedades definidas en el modelo de datos del servicio con las estructuras de datos y tablas subyacentes en el back-end.

En primer lugar, se mapeará a su vez el “Entity Set” de órdenes, donde en este caso los inputs serán el almacén, la línea de producción y el número de orden, teniendo múltiples outputs como son los mostrados en la Tabla 10. Los inputs serán dirigidos al módulo de función ZWM\_MF\_GET\_ORDENES, creado anteriormente, el cual devolverá una tabla de salida con distintos outputs a mostrar en la aplicación (front-end).

Mapping of Operation GetEntitySet (Query) for OrdenSet			Data Source Parameter	Description
Pr...	Entity Set property	Constant Value	M... Data Source Parameter	
	Lgtyp		← ET_SALIDA\LGTYT	
	Sprax		← ET_SALIDA\SPRAS	
	Lgnum		← ET_SALIDA\LGNUM	
	Lgnum		→ RG_ALMACEN	
	Lgpla		← ET_SALIDA\LGPLA	
	Lgort		← ET_SALIDA\LGORT	
	Werks		← ET_SALIDA\WERKS	
	Prvbe		← ET_SALIDA\PRVBE	
	Prvbe		→ RG_LINEA	
	Matnr		← ET_SALIDA\MATNR	
	Maktx		← ET_SALIDA\MAKTX	
	Menge		← ET_SALIDA\MENGE	
	Meins		← ET_SALIDA\MEINS	
	Charg		← ET_SALIDA\CHARG	
	Aufnr		← ET_SALIDA\AUFNR	
	Aufnr		→ RG_ORDEN	
	CySeqnr		← ET_SALIDA\CY_SEQNR	
	TipoCaract		← ET_SALIDA\TIPO_CARACT	
	Bdmng		← ET_SALIDA\BDMNG	
	TipoAbastecido		← ET_SALIDA\TIPO_ABASTECIDO	
	Abastecida		← ET_SALIDA\ABASTECIDA	
	Zprimavera		← ET_SALIDA\ZPRIMAVERA	
	Rspos		← ET_SALIDA\RSPOS	

Data Source Parameter	Description
ZWM_MF_GET_ORDENES	MF traer listado ordenes a pantalla
RG_ALMACEN[]	
RG_LINEA[]	
RG_ORDEN[]	
ET_SALIDA[]	
STAT	Status
TXTO4	Status
LGTYT	Tipo almacén
SPRAS	Idioma
LGNUM	Núm.almacén
LGPLA	Ubicación
LGORT	Almacén
WERKS	Centro
PRVBE	Área sum.prod.
MATNR	Material
MAKTX	Denominación
MENGE	Cantidad pedido
MEINS	UM base
CHARG	Lote
AUFNR	Orden
CY_SEQNR	Nº de secuencia
TIPO_CARACT	char70
BDMNG	Ctd.necesaria
TIPO_ABASTECIDO	Carácter 1
ABASTECIDA	Carácter 1
ZPRIMAVERA	Nombre campo
RSPOS	Posición

**Tabla 7. Mapeo del “Entity Set” de órdenes.**

**Fuente: Elaboración propia.**

Por último, se mapea el “Entity Set” de la línea de producción, donde el único output e input es el puesto de trabajo como se indica en la Tabla 8. Al igual que en el anterior mapeo, los inputs serán dirigidos, en este caso, al módulo de función ZWM\_MF\_SH\_LINEA, el cual se ha creado anteriormente como ya se ha visto y que devolverá una tabla de salida con distintos outputs a mostrar en la aplicación (front-end).

Mapping of Operation GetEntitySet (Query) for SHLineaSet			Data Source Parameter	Mandatory	Description
Pr...	Entity Set property	Constant Value	M... Data Source Parameter		
	Prvbe		← ET_SALIDA\PRVBE	<input type="checkbox"/>	Search header de linea set
	Prvbe		→ RG_LINEA	<input type="checkbox"/>	Área sum.prod.

**Tabla 8. Mapeo del “Entity Set” de línea de producción.**

**Fuente: Elaboración propia.**

Con este último paso ya se ha creado el modelo de datos al completo, únicamente faltaría guardar y activar los cambios, así como generar las clases DPC (Data Provider Class) y MPC (Model Provider Class) junto con las extensiones DPC\_EXT y MPC\_EXT para agregar lógica adicional a las clases generadas sin modificarlas directamente.

A grandes rasgos, el DPC se encarga de manejar las operaciones CRUD (Create, Read, Update, Delete) y realiza las acciones asociadas con el modelo de datos. Mientras que el MPC define las entidades, asociaciones, acciones y funciones, mapeando los datos entre el modelo de datos y el back-end.

### 3.6. Publicación del modelo de datos

Una vez generado el modelo de datos, falta publicarlo para que sea posible consumirlo desde front-end por una aplicación SAPUI5. Al publicar el modelo de datos, se permite a la aplicación acceder y manipular los datos del back-end a través del servicio OData.

Para llevar a cabo la publicación del modelo de datos hay diferentes métodos de hacerlo, pero se va a comentar el más seguro. Para ello hay que dirigirse a la transacción “/n/iwfnd/maint\_services” en la cual se podrá añadir el servicio tal y como se muestra en la Figura 5.

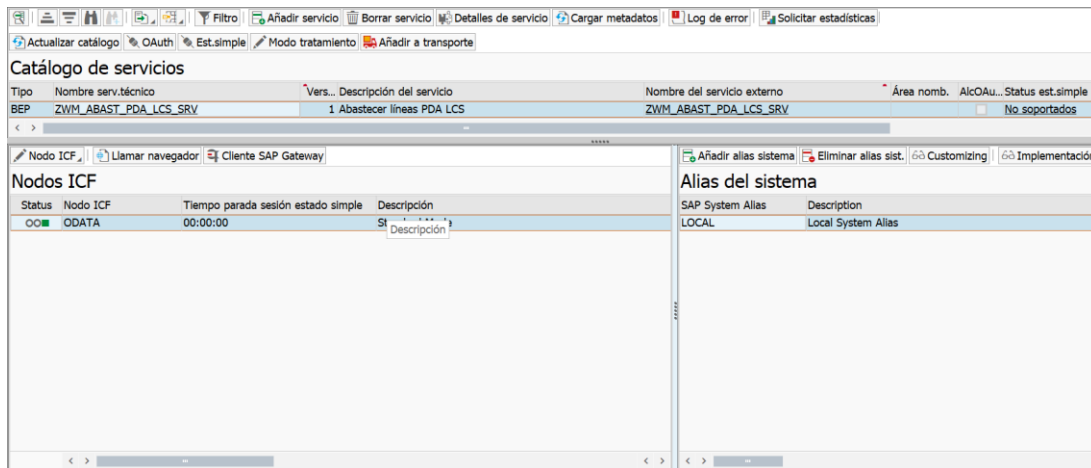
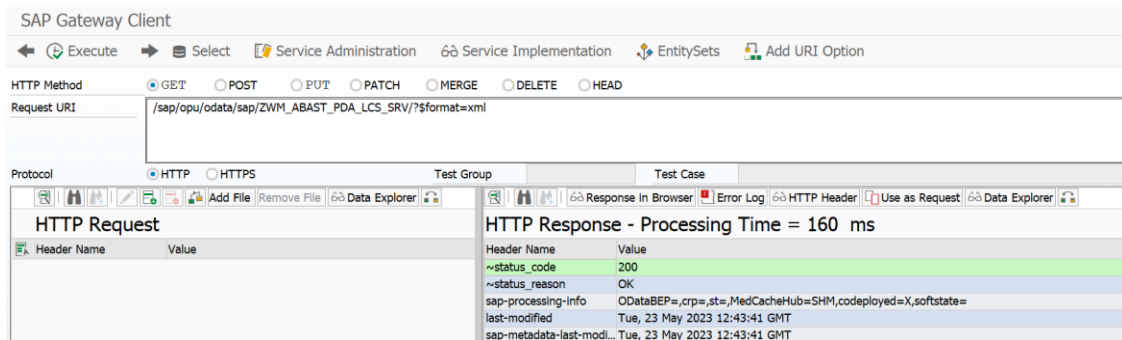


Figura 5. Publicación del modelo de datos en la transacción “/n/iwfnd/maint\_services”.

Fuente: Elaboración propia.

Como se puede observar, el servicio a añadir es el ZWM\_ABAST\_PDA\_LCS\_SRV. Una vez añadido el servicio, se accede a la pestaña Cliente SAP Gateway. Cabe destacar que mediante el SAP Gateway es posible consumir los servicios OData desde aplicaciones web y móviles, así como exponerlos desde sistemas SAP subyacentes como son SAP ERP, SAPS/4HANA o SAP CRM.

Si en la pestaña de SAP Gateway se pulsa el botón de ejecución y se recibe un valor 200 en el HTTP response, quiere decir que la solicitud de prueba de conectividad entre el cliente y el SAP Gateway se ha realizado exitosamente. Esto confirma que el cliente está correctamente configurado y puede interactuar con el back-end a través del servicio OData expuesto por el SAP Gateway. Un ejemplo de una conectividad exitosa es la mostrada en la Figura 6.



**Figura 6. Conectividad exitosa entre el cliente y el SAP Gateway.**

**Fuente: Elaboración propia.**

### 3.6.1. *Publicación del modelo de datos como un servicio OData*

Es relevante saber porque se publica el modelo de datos como un servicio OData, por ello en este apartado se darán las suficientes razones que respalden este hecho.

En aspectos generales, se debe a que de esta forma se permite el acceso y la manipulación de los datos de manera uniforme y estandarizada a través de interfaces web. Los beneficios de publicar el modelo de datos como un servicio OData son los siguientes:

- Los clientes pueden consumir el servicio OData independientemente de la plataforma o tecnología utilizada, siempre que sean compatibles con el protocolo HTTP y el formato de datos utilizado (JSON o XML).
- El servicio OData expone los datos estructurados de manera coherente y consistente. Los clientes pueden acceder a los datos utilizando operaciones CRUD estándar (crear, leer, actualizar, eliminar) y navegar por las relaciones entre las entidades. Esto simplifica el acceso y la manipulación de los datos, ya que los clientes pueden confiar en la estructura y las operaciones definidas por el servicio OData.
- El servicio OData proporciona metadatos que describen la estructura de los datos disponibles y las operaciones que se pueden realizar en ellos. Esto facilita a los clientes descubrir y comprender la estructura de los datos sin conocimiento previo. Los clientes pueden consultar los metadatos para determinar qué entidades están disponibles, qué propiedades tienen y qué operaciones se pueden realizar en ellas.
- Los servicios OData admiten capacidades de filtrado y consulta que permiten a los clientes solicitar solo los datos necesarios y realizar consultas complejas para recuperar conjuntos de datos específicos. Esto mejora la eficiencia y el rendimiento al reducir la cantidad de datos transferidos entre el servidor y el cliente.

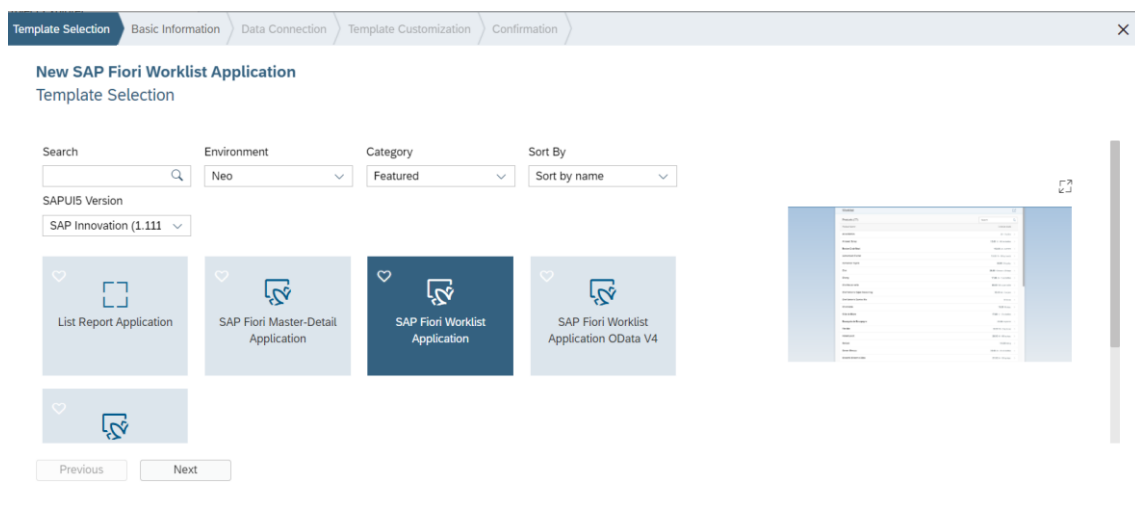
### 3.7. Front-end

Todo el desarrollo de la parte front-end se realiza usando el entorno que proporciona SAP y del cual ya se ha hablado anteriormente en el presente trabajo, se trata de SAP WebIDE. Dicho entorno es el que se utilizará para implementar la aplicación SAPUI5.

En este apartado, se detallará cada uno de los pasos seguidos para elaborar la aplicación analizando cada parte del código, el cual se adjunta en el apartado de Anexos.

#### 3.7.1. Creación del proyecto

Dentro del entorno SAP WebIDE, se crea un nuevo proyecto para la aplicación. Los pasos a seguir son los mostrados a continuación.



**Figura 7. Creación del proyecto.**

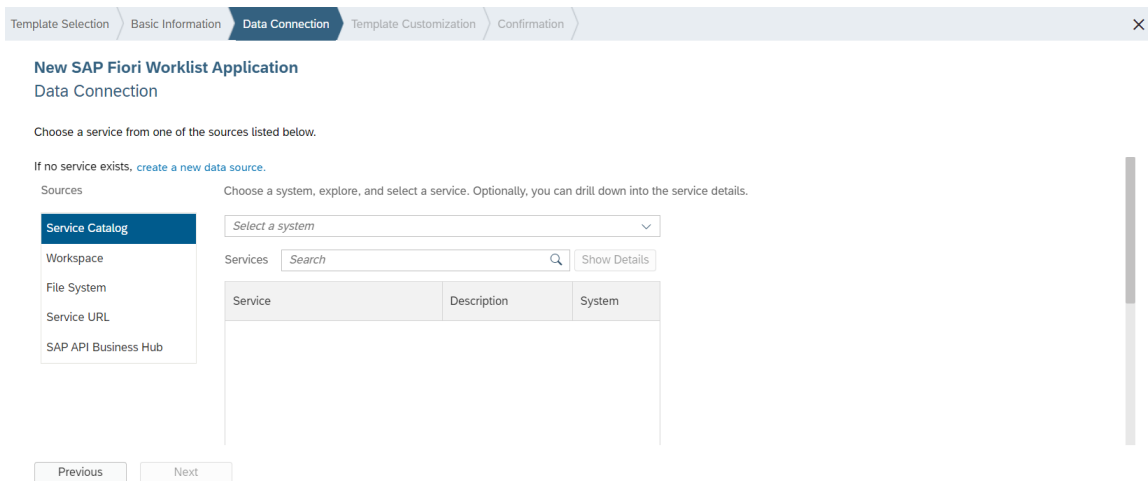
**Fuente: Elaboración propia.**

En este caso se elige una aplicación del tipo Worklist, ya que lo que se sacará por pantalla será un listado de órdenes, así como detalles de cada una de ellas. Se tiene en cuenta que la versión de SAP que se usará será la más nueva.

Tras elegir el tipo de aplicación SAP Fiori, se informa de cuál será el nombre de la aplicación, así como una descripción y la ruta donde estará alojado el proyecto.

Siguiendo con la creación del proyecto, también se debe de informar de la conexión al destino creado del cual se ha hablado anteriormente en el presente trabajo, así como al servicio que se ha publicado llamado ZWM\_ABAST\_PDA\_LCS\_SRV. Dicha acción se realiza en la Figura 8.

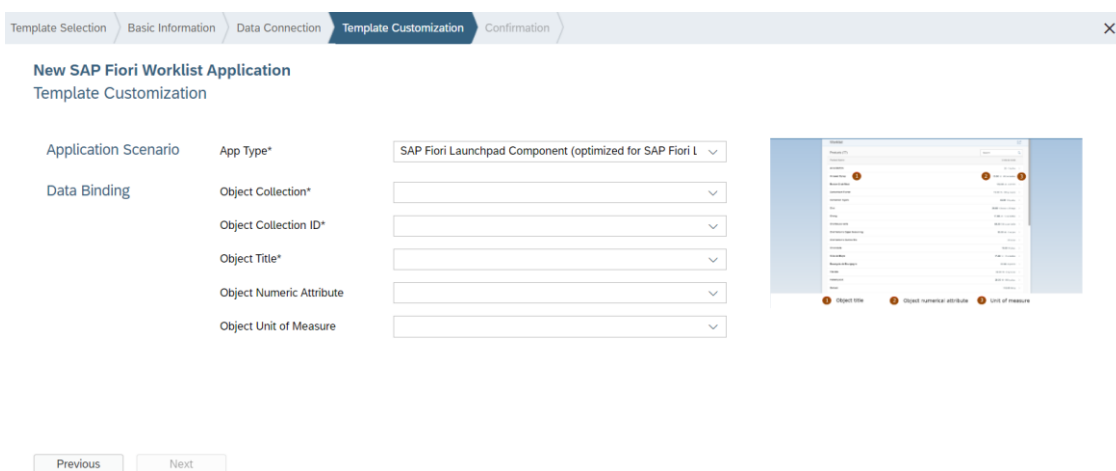




**Figura 8. Conexión al sistema y servicio publicado.**

**Fuente: Elaboración propia.**

Por último, antes de finalizar con la creación del proyecto se deberá indicar que entidades van a tomar protagonismo en el Worklist a desarrollar de la aplicación donde la colección de objeto se trata de las órdenes, así como del título y en el atributo numérico se pondrá el stock. Dichas operaciones se observan con más detalle en la Figura 9.



**Figura 9. Personalización del Worklist.**

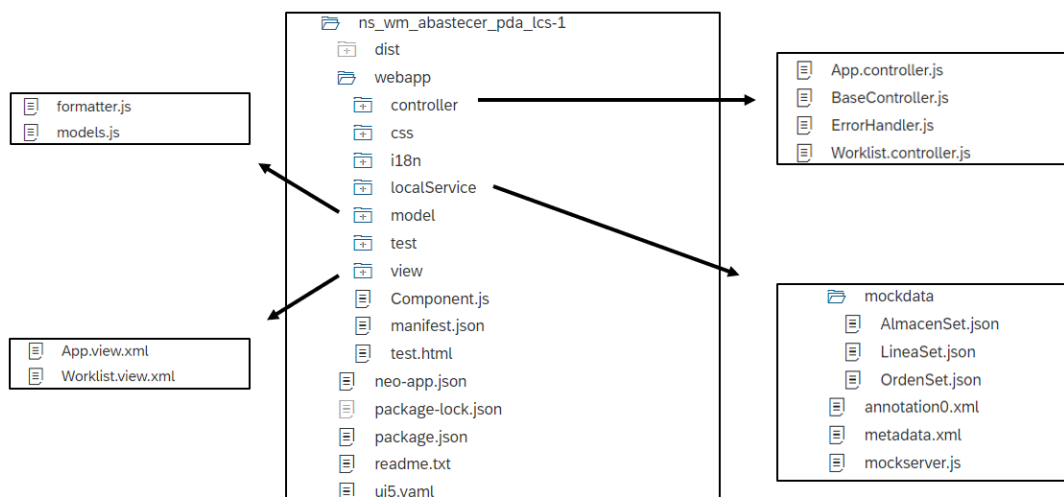
**Fuente: Elaboración propia.**

Para seguir con la explicación y el desarrollo de la aplicación, cabe explicar que es un modelo de datos, así como la funcionalidad y diseño de éste. Tras ello, se procederá a analizar el modelo de datos utilizado en el desarrollo de la aplicación (Figura 10).

Los modelos de datos en SAPUI5 se basan en el patrón de diseño Modelo-Vista-Controlador (MVC), donde los modelos de datos son los que contienen los datos, las vistas definen la presentación de éstos y los controladores son los que se encargan de manejar la lógica y la interacción. A continuación, se detalla la función de cada uno de los componentes clave que componen un proyecto creado en SAP WebIDE.

- Carpeta “view”: contiene archivos que definen la interfaz de usuario de la aplicación SAPUI5, donde cada uno de ellos tiene la extensión “.view.xml”. Por tanto, dichos archivos, se puede decir, que son los responsables de la presentación de los datos y la interacción con el usuario. Algunos de los elementos visuales que son mostrados como botones, formularios y tablas se definen en esta carpeta. Cabe destacar que también es definida la lógica de navegación entre diferentes vistas que se tengan en la aplicación. En este caso, esto último no se va a realizar, ya que solo se va a tener una vista principal donde se realizarán todas y cada una de las acciones.
- Carpeta “controller”: en esta carpeta se tiene los archivos del lenguaje de programación Javascript, los cuales actuarán como controladores de las vistas definidas en la carpeta anteriormente descrita y tienen la extensión “.controller.js”. La funcionalidad de dichos archivos es la de definir el comportamiento de la interfaz de usuario manejando eventos, realizando validaciones, así como otras tareas.
- Carpeta “model”: donde se definen los modelos de datos utilizados por la aplicación SAPUI5. Los archivos contenidos en esta carpeta, cabe destacar, que se trata de archivos Javascript, con la extensión “.js” que se encargan de conectar la aplicación con servicios de back-end o con otras fuentes de datos.

Dentro del espacio de trabajo, una vez creado el proyecto y llevada a cabo la creación de los diferentes archivos necesarios para el correcto desarrollo de la aplicación pedida, se verán los archivos mostrados en la Figura 10 dentro de éste.



**Figura 10. Archivos que conforman el proyecto.**

**Fuente: Elaboración propia.**

Donde archivos como es la vista “App” y su controlador asociado (“App.controller.js”) del mismo nombre vienen por defecto al crear el proyecto, pero se ha optado por crear otro archivo llamado “Worklist.view.xml” el cual será la vista principal, dejando a “App.view.xml” sin uso. Tampoco se hará uso del “BaseController.js”, el cual venía por defecto según la configuración dada a la hora de la creación del proyecto, debido a que “Worklist.controller.js” será el utilizado, asociándolo con la vista creada anteriormente. Asimismo, el archivo “ErrorHandler.js”, el cual viene por defecto según la configuración dada en la creación del proyecto, será el encargado de capturar y manejar errores no controlados y excepciones en la aplicación.

Por otro lado, cabe destacar que en la carpeta “localService” se creó una carpeta mockdata con un mockserver. Con ello se realizaron las pruebas pertinentes para ver el correcto funcionamiento de la aplicación cuando se desarrolló antes de probarlos con los datos reales de la empresa cliente.

Por último, hay que mencionar que archivos como son el componente (“Component.js”) y el descriptor (“manifest.json”), son estrictamente necesarios en la implementación de una aplicación SAPUI5. Dichos archivos tienen las funcionalidades descritas en el apartado “3.7.1. Implementación de una aplicación SAPUI5”.

### 3.7.2. Creación de la vista principal (Worklist.view.xml)

A continuación, se irá analizando cada parte de la vista principal. La primera parte a analizar es la mostrada en la Figura 11.

```

1 <mvc:View controllerName="net.wm.abastecer_pda_lcs.controller.Worklist" xmlns="sap.m" xmlns:mvc="sap.ui.core.mvc"
2   xmlns:core="sap.ui.core" xmlns:semantic="sap.f.semantic" xmlns:smartFilterBar="sap.ui.comp.smartfilterbar"
3   xmlns:smartTable="sap.ui.comp.smarttable"
4 <semantic:SemanticPage id="page" headerPinnable="false" toggleHeaderOnTitleClick="false"
5 <semantic:titleHeading
6   <title text="{!@!worklistTitle}" level="H2"/>
7 </semantic:titleHeading
8 <!-- Content -->
9 <semantic:content
10 <VBox id="vBox" fitContainer="true"
11 <smartFilterBar:SmartFilterBar id="smartFilterBar" initialize="onSmartFilterBarInitialise" entitySet="OrdenSet"
12   persistencyKey="SmartFilter_Orden" useToolBar="false"
13 <smartFilterBar:controlConfiguration
14 <smartFilterBar:ControlConfiguration id="sfaufnr" index="1" key="Aufnr"
15   preventInitialDataFetchInValueHelpDialog="false" visibleInAdvancedArea="true"/></smartFilterBar:ControlConfiguration>
16 <smartFilterBar:ControlConfiguration id="sfprube" index="2" key="Prube"
17   preventInitialDataFetchInValueHelpDialog="false" visibleInAdvancedArea="true"/></smartFilterBar:ControlConfiguration>
18 <smartFilterBar:ControlConfiguration id="sflgnum" index="3" key="Lgnum" preventInitialDataFetchInValueHelpDialog="false"
19   visibleInAdvancedArea="true"
20 <smartFilterBar:defaultFilterValues
21   <smartFilterBar:SelectOption low="800" high=""></smartFilterBar:SelectOption>
22 </smartFilterBar:defaultFilterValues>
23 </smartFilterBar:ControlConfiguration>
24 </smartFilterBar:controlConfiguration>
25 </smartFilterBar:SmartFilterBar>
26 <smartTable:SmartTable
27   entitySet="OrdenSet"
28   smartFilterId="smartFilterBar"
29   tableType="ResponsiveTable"
30   id="smartTable"
31   header="{worklistView/worklistTableTitle}"
32   beforeRebindTable="onBeforeRebindTable"
33   showRowCount="false"
34   persistencyKey="SmartTableAnalytical_Orden"
35   enableAutoBinding="false"
36   useExportToExcel="false"
37   ignoreFromPersonalisation="{Lgtyp,Spras,Lgnum,Lgpla,Lgort,Werks,Prube,Matnr,Mehtx,Menge,Meins,Charg,Aufnr,CySeqr,TipoCaract,Bdmg,TipoAbastecido,Abastecido,Zprimevera,Rspas"
38   requestAtLeastFields="{Lgtyp,Spras,Lgnum,Lgpla,Lgort,Werks,Prube,Matnr,Mehtx,Menge,Meins,Charg,Aufnr,CySeqr,TipoCaract,Bdmg,TipoAbastecido,Abastecido,Zprimevera,Rspas"

```

Figura 11. Definición de la vista, filtros y tabla inteligente en la vista principal.

Fuente: Elaboración propia.

Se empezarán definiendo las librerías a utilizar a lo largo del código a elaborar, así como del controlador asociado a la vista del cual se hablará más adelante.

Más adelante se define el título de la página y se establecen una serie de filtros inteligentes en los cuales se insertarán una serie de campos que quieran ser buscados. Hay tres tipos diferentes, teniendo la posibilidad de filtrar tanto por número de orden, línea de producción (solo hay 3 como se ha visto ya anteriormente) y por número de almacén, el cual será fijo (800).

Al igual que se definen los filtros inteligentes, también se debe de definir una tabla inteligente a la que ligar dichos filtros. De esta tabla inteligente se destacan los campos:

- “entitySet”: donde se pondrá la entidad a la que la tabla está vinculada. En este caso se vincula a la entidad “OrdenSet”.
- “smartFilterId”: asocia la tabla inteligente con una barra de filtros inteligente utilizando el ID de la barra de filtros. Esto permite que la tabla interactúe con los filtros definidos en la barra de filtros.
- “tableType”: define el tipo de tabla utilizado para mostrar los datos. En este caso, se utiliza una tabla sensible al tamaño de pantalla que se adapta automáticamente a diferentes dispositivos y resoluciones.
- beforeRebindTable="onBeforeRebindTable": Especifica el nombre de una función o método en el controlador asociado a la vista que se ejecutará antes de que se vuelva a vincular la tabla. Esto permite realizar acciones adicionales o ajustes en los datos o en la configuración de la tabla antes de mostrarlos.
- “showRowCount”: Determina si se muestra el recuento de filas en la tabla. En este caso, se establece en "false" para no mostrar el recuento de filas.
- “enableAutoBinding”: controla si la tabla se vincula automáticamente a los datos al cargar la página. En este caso, se desactiva la vinculación automática para permitir un control manual de cuándo y cómo se cargan los datos en la tabla.
- “ignoreFromPersonalisation”: lista de campos que los usuarios no podrán personalizar, ya sea ocultándolos o cambiando su posición en la interfaz de usuario.
- “requestAtLeastFields”: campos que se incluyen en la consulta de datos, aunque el usuario no los seleccione para su visualización en la tabla.

A continuación, se introduce la segunda parte del código de la vista principal con su correspondiente análisis.

```

40 - <Table id="table" width="auto" noDataText="{worklistView}/tableNoDataText}" busyIndicatorDelay="{worklistView}/tableBusyDelay"
41 - growing="true" growingScrollToLoad="true" updateFinished=".onUpdateFinished" sticky="ColumnHeaders,HeaderToolBar">
42 - <columns>
43 - <Column>
44 - <customData>
45 - <core:CustomData key="pi3nData" value="{columnKey: 'pue', 'columnIndex': '0'}"/>
46 - </customData>
47 - <Text text="{i18n>PuestoTrabajo}/>
48 - </Column>
49 - <Column>
50 - <customData>
51 - <core:CustomData key="pi3nData" value="{columnKey: 'ord', 'columnIndex': '1'}"/>
52 - </customData>
53 - <Text text="{i18n>Orden}/>
54 - </Column>
55 - </columns>
56 - <items>
57 - <ColumnListItem type="Inactive" press=".onPress">
58 - <cells>
59 - <Text text="{Prvbe}"/>
60 - <Text text="{Aufnr}"/>
61 - <Text text="{Matnr}"/>
62 - <Text text="{Charg}"/>
63 - <Text text="{Lgpla}"/>
64 - <Text text="{Lgtyp}"/>
65 - <ObjectNumber
66 - number="{ path:'Bdmng', formatter: '.Formatter.numberUnit'}"
67 - unit="{Meins}"/>
68 - <Button text="{i18n>Ubicar}" press="onLocatePress"></Button>
69 - </cells>
70 - </ColumnListItem>
71 - </items>
72 - </Table>
73 - </smartTable:SmartTable>
74 - </VBox>
75 - </semantic:content>
76 - <semantic:sendEmailAction>
77 - <semantic:sendEmailAction id="shareEmail" press=".onShareEmailPress"/>
78 - </semantic:sendEmailAction>
79 - </semantic:SemanticPage>
80 - </mvc:View>

```

**Figura 12. Definición de columnas y campos en la vista principal.**

**Fuente: Elaboración propia.**

Como se puede observar en la Figura 12, se define una tabla normal dentro de la tabla inteligente con columnas y elementos de la lista.

En la parte superior del código se tiene la definición de las columnas que se van a crear dentro de la tabla. En este caso solo se ha mostrado la creación de la columna que contendrá el número de órdenes y de puesto de trabajo con el objetivo de no mostrar la creación de las tres columnas restantes (material, lote, cantidad y ubicar), ya que es un proceso repetitivo.

En la parte inferior de la Figura 12, se establece el contenido de cada una de las columnas por orden. Al haber indicado en la tabla inteligente que se van a sacar los datos de la entidad Ordenset, se hará un "binding" a dicha entidad para obtener los campos buscados.

Finalmente, se introduce un botón "ubicar" en la última columna para cada una de las órdenes que al ser pulsado ejecutará la función ".onLocatePress", la cual se encuentra en el controlador y de la que se hablará más adelante.

### 3.7.3. Creación del controlador asociado a la vista principal (*Worklist.controller*)

Al crear el proyecto, hay ciertas funciones que vienen ya por defecto y que, por tanto, no hace falta crear. En este apartado se comentarán las funciones extra que se han añadido para el correcto desarrollo de la aplicación, así como simplemente dar una breve explicación de cada función que ya vienen en el código cuando se crea el proyecto. Cabe destacar que el código del controlador utilizado para el desarrollo de la aplicación se encuentra en el Anexo.

Las funciones que ya vienen, así como la explicación de cada una de ellas son las siguientes:

- “\_showObject”: Esta función se utiliza para mostrar un objeto específico en la vista detallada. Navega a la ruta "object" de la aplicación y pasa el identificador del objeto como parámetro.

```
_showObject: function (oItem) {
    this.getRouter().navTo("object", {
        objectId: oItem.getBindingContext().getProperty("Aufnr")
    });
},
```

Figura 13. Función “\_showObject”.

Fuente: Elaboración propia.

- “onUpdateFinished”: Esta función se ejecuta cuando se completa la actualización de la tabla. Actualiza el contador de objetos en la vista de lista según el número total de elementos en la tabla. Actualiza el título de la tabla en función del número de objetos.

```
onUpdateFinished: function (oEvent) {
    // update the worklist's object counter after the table update
    var sTitle,
        oTable = oEvent.getSource(),
        iTotItems = oEvent.getParameter("total");
    // only update the counter if the length is final and
    // the table is not empty
    if (iTotItems && oTable.getBinding("items").isLengthFinal()) {
        sTitle = this.getResourceBundle().getText("worklistTableTitleCount", [iTotItems]);
    } else {
        sTitle = this.getResourceBundle().getText("worklistTableTitle");
    }
    this.getModel("worklistView").setProperty("/worklistTableTitle", sTitle);
},
```

Figura 14. Función “onUpdateFinished”.

Fuente: Elaboración propia.

- “onPress: Esta función se activa cuando se presiona un elemento de la tabla. Llama a la función “\_showObject” para mostrar el objeto seleccionado en la vista detallada.

```
onPress: function (oEvent) {
    // The source is the list item that got pressed
    this._showObject(oEvent.getSource());
},
```

Figura 15. Función “oPress”.

Fuente: Elaboración propia.

- “onSearch”: Esta función se activa cuando se realiza una búsqueda en la lista. Aplica el estado de búsqueda a la lista, filtrando los elementos que coincidan con el criterio de búsqueda proporcionado.

```

onSearch: function (oEvent) {
    if (oEvent.getParameters().refreshButtonPressed) {
        // Search field's 'refresh' button has been pressed.
        // This is visible if you select any master list item.
        // In this case no new search is triggered, we only
        // refresh the list binding.
        this.onRefresh();
    } else {
        var aTableSearchState = [];
        var sQuery = oEvent.getParameter("query");

        if (sQuery && sQuery.length > 0) {
            aTableSearchState = [new Filter("Aufnr", FilterOperator.Contains, sQuery)];
        }
        this._applySearch(aTableSearchState);
    }
},

```

**Figura 16. Función “onSearch”.**

**Fuente: Elaboración propia.**

- “onRefresh”: Esta función se activa cuando se presiona el botón de actualización de la lista. Actualiza la lista volviendo a cargar los datos.

```

onRefresh: function () {
    var oTable = this.byId("table");
    oTable.getBinding("items").refresh();
},

```

**Figura 17. Función “onRefresh”.**

**Fuente: Elaboración propia.**

“\_applySearch”: Esta función es un método de ayuda interno que aplica el estado de búsqueda y filtro a la lista. Filtra los elementos de la lista según los criterios de búsqueda y actualiza el texto cuando no se encuentran resultados de filtro.

```

_applySearch: function (aTableSearchState) {
    var oTable = this.byId("table"),
        oViewModel = this.getModel("worklistView");
    oTable.getBinding("items").filter(aTableSearchState, "Application");
    // changes the noDataText of the list in case there are no filter results
    if (aTableSearchState.length !== 0) {
        oViewModel.setProperty("/tableNoDataText", this.getResourceBundle().getText("worklistNoDataWithSearchText"));
    }
}

```

**Figura 18. Función “\_applySearch”.**

**Fuente: Elaboración propia.**

Una vez analizadas cada una de las funciones que vienen por defecto una vez se crea el proyecto, se pasará a comentar las funciones añadidas con el objetivo de cumplir con los requisitos pedidos inicialmente que en este caso solo será una que se encargará de reubicar la orden elegida.

Previamente se veía como al pulsar el botón “ubicar” se ejecutaba una función del controlador. Dicha función es la mostrada en la Figura 19.

```
onLocatePress: function (oEvent) {
    //Recuperar fila en la que estamos y extraer el valor Aufnr. Recuperamos el entity type completo.
    var oSelectedItem = oEvent.getSource().getParent();
    var oContext = oSelectedItem.getBindingContext();
    var oOrden = oContext.getObject();
    var sAufnr = oOrden.Aufnr;
    var oModel = this.getView().getModel();
    // lectura del i18n model
    var oBundle = this.getView().getModel("i18n").getResourceBundle();
    var oSmartTable = this.getView().byId("smartTable");
    oModel.callFunction("/Ubicar", { // function import name
        method: "GET", // http method
        urlParameters: {
            Aufnr: sAufnr
        }, // function import parameters
        success: function (oData, response) {
            var sMsg = oBundle.getText("Ubicorrecta");
            MessageToast.show(sMsg);
            // refresco de tabla
            oSmartTable.rebindTable();
        }, // callback function for success
        error: function (oError) { // callback function for error

            sap.m.MessageBox.error("No se ha podido mover este lote, revise si la ubicación esta completa");
        }
    });
},
```

Figura 19. Función “onLocatePress”.

Fuente: Elaboración propia.

Se comienza obteniendo el elemento seleccionado que desencadenó el evento, asumiendo que éste está en la fila de una tabla y obtiene la fila al completo. Tras obtener la referencia, se tiene el contexto de enlace del elemento seleccionado, dicho contexto representa el objeto de datos asociado a la fila donde está ubicado el botón.

Con la sentencia “oContext.getObject()” se acaba obteniendo el objeto de datos del contexto. Dicho objeto contiene todos los datos de la fila seleccionada.

Por otro lado, definiendo la variable “sAufnr”, lo que se hace es extraer el valor de la propiedad “Aufnr” del objeto “oOrden”. También se obtiene una referencia al modelo de datos asociado a la vista y se realiza una llamada a la “function import” de Ubicar que se veía en apartados anteriores.

Se acaban estableciendo dos casos:

- Si la llamada a la función es exitosa se ejecuta la función y la orden se reubica, además se muestra un mensaje de éxito. Más tarde, se realiza un refresco de la tabla y, por tanto, la orden que se acaba de reubicar desaparecerá de la tabla.
- Si ocurre un error durante la llamada a la función, se ejecutaría esta función de devolución de la llamada, la cual incluye un mensaje de error.



### 3.7.4. Creación de las anotaciones locales (annotation0.xml)

Antes de nada, cabe comentar cual es la funcionalidad del archivo de anotaciones locales. Dicho archivo se utiliza para proporcionar metadatos adicionales sobre las entidades y propiedades de datos en una aplicación, por tanto, sirven de mejora de la descripción de los modelos de datos y proporcionan información adicional sobre cómo se deben presentar y manipular los datos en la interfaz usuario.

En este caso, se agrega la propiedad “Prvbe” donde el “CollectionPath” es “SHLineaSet”, el cual está mapeado con la función ZWM\_MF\_SH\_LINEA (Anexo). Por tanto, en la vista principal de la aplicación en el filtro de línea a abastecer, aparecerá una búsqueda de ayuda, donde se visualizarán 3 opciones: decapado, laminado y galvanizado.

```
*annotation0.xml x
1 <edmx:Edmx xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx" Version="4.0">
2   <edmx:Reference Uri="/sap/bc/ui5_ui5/ui2/ushell/resources/sap/ushell/components/factsheet/vocabularies/UI.xml">
3     <edmx:Include Alias="UI" Namespace="com.sap.vocabularies.UI.v1"/>
4   </edmx:Reference>
5   <edmx:Reference Uri="/sap/bc/ui5_ui5/ui2/ushell/resources/sap/ushell/components/factsheet/vocabularies/Communication.xml">
6     <edmx:Include Alias="vCard" Namespace="com.sap.vocabularies.Communication.v1"/>
7   </edmx:Reference>
8   <edmx:Reference Uri="http://docs.oasis-open.org/odata/odata/v4.0/cs01/vocabularies/Org.OData.Measures.V1.xml">
9     <edmx:Include Alias="CQP" Namespace="Org.OData.Measures.V1"/>
10  </edmx:Reference>
11  <edmx:Reference Uri="/sap/bc/ui5_ui5/ui2/ushell/resources/sap/ushell/components/factsheet/vocabularies/Common.xml">
12    <edmx:Include Alias="Common" Namespace="com.sap.vocabularies.Common.v1"/>
13  </edmx:Reference>
14  <edmx:Reference Uri="/sap/opu/odata/sap/ZWM_ABAST_PDA_LCS_SRV/$metadata">
15    <edmx:Include Alias="Metadata" Namespace="ZWM_ABAST_PDA_LCS_SRV"/>
16  </edmx:Reference>
17  <edmx:Reference Uri="https://wiki.scn.sap.com/wiki/download/attachments/448470968/UI.xml?api=v2">
18    <edmx:Include Alias="UI" Namespace="com.sap.vocabularies.UI.v1"/>
19  </edmx:Reference>
20  <edmx:DataService>
21    <Schema xmlns="http://docs.oasis-open.org/odata/ns/edm" Namespace="ns_abastecer_pda_lcs_localService.annotation0.ZWM_ABAST_PDA_LCS_SRV">
22      <Annotations xmlns="http://docs.oasis-open.org/odata/ns/edm" Target="Metadata.Orden/Prvbe">
23        <Annotation Term="Common.ValueList">
24          <Record Type="Common.ValueListType">
25            <PropertyValue Property="CollectionPath" String="SHLineaSet"/>
26            <PropertyValue Property="SearchSupported" Bool="false"/>
27            <PropertyValue Property="Parameters">
28              <Collection>
29                <Record Type="com.sap.vocabularies.Common.v1.ValueListParameterInOut">
30                  <PropertyValue Property="LocalDataProperty" PropertyPath="Prvbe"/>
31                  <PropertyValue Property="ValueListProperty" String="Prvbe"/>
32                </Record>
33              </Collection>
34            </Record>
35          </Annotation>
36        </Annotations>
37      </Schema>
38    </edmx:DataService>
39  </edmx:Edmx>
```

Figura 20. Anotaciones locales del proyecto.

Fuente: Elaboración propia.

### 3.7.5. Creación del formatter (formatter.js)

Este archivo se encarga de definir funciones reutilizables de formato y transformación de datos los cuales son utilizados principalmente en la vista para dar otro formato al tipo de datos que se muestra al usuario.

En este caso, se toma un valor numérico en forma de cadena y la función que se observa en la Figura 21 se encarga de convertirlo a un número decimal redondeado a dos decimales.

Como se ha dicho anteriormente, al formatter es común hacerle una llamada desde la vista. En este caso, la llamada se le hace desde la vista principal (Worklist.view.xml), dicha llamada ya se veía en la Figura 12, pero podía pasar de forma desapercibida, por ello se ha hecho hincapié en ello y se ha incluido la Figura 22 en la cual se puede ver de manera más clara dicha llamada al formatter.

```

*formatter.js x
1 sap.ui.define([], function () {
2     "use strict";
3
4     return {
5
6         numberUnit : function (sValue) {
7             if (!sValue) {
8                 return "";
9             }
10            return parseFloat(sValue).toFixed(2);
11        }
12    };
13
14 });

```

**Figura 21. Formatter.**

**Fuente: Elaboración propia.**

```

<ColumnListItem type="Inactive" press=".onPress">
<cells>
<Text text="{Prvbe}"/>
<Text text="{Aufnr}"/>
<Text text="{Matnr}"/>
<Text text="{Chang}"/>
<Text text="{Lgpla}"/>
<Text text="{Lgtyp}"/>
<ObjectNumber
number="{ path:'Bdmng', formatter: './formatter.numberUnit'}"
unit="{Meins}"/>

```

**Figura 22. Llamada desde la vista principal al formatter.**

**Fuente: Elaboración propia.**

### 3.8. Vista en web de la aplicación

Una vez elaborado tanto el código del controlador como de la vista con las anotaciones pertinentes para el correcto funcionamiento de la ayuda de búsqueda para el caso del filtro de la línea a abastecer, se obtiene el diseño final de la aplicación pedida por el cliente. Dicho diseño se puede observar en la Figura 23.

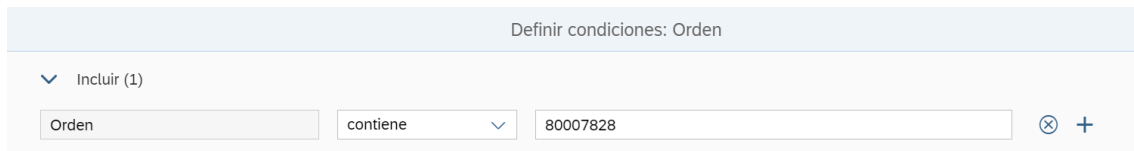
Puesto trabajo	Orden	Material	Lote	Ubicación	Tipo Almacén	Cantidad	Operación
DECAPAD1	80008092	...	0822003155	A06H1	C01	20,07	Ubicar
DECAPAD1	80008079	...	0822003241	L02H1	C01	24,62	Ubicar
DECAPAD1	80007845	...	0822003367	G19H1	C02	25,08	Ubicar
DECAPAD1	80008095	...	0822003854	A08H1	C01	23,51	Ubicar
DECAPAD1	80007889	...	0822004827	F01H1	C02	23,45	Ubicar
DECAPAD1	80007890	...	0822004830	G01H1	C02	22,06	Ubicar
DECAPAD1	80007828	...	0822005995	F08H1	C02	24,06	Ubicar
DECAPAD1	80008091	...	0822006116	AA13H2	C01	24,32	Ubicar

**Figura 23. Diseño final de la aplicación.**

**Fuente: Elaboración propia.**

Por temas de confidencialidad los materiales no serán visibles, pero para hacer las comprobaciones de que la aplicación funciona correctamente los números de orden serán visibles. A continuación, se probarán las distintas funcionalidades de la aplicación.

Se empezará hablando del funcionamiento del filtraje por número de orden. Haciendo uso de dicho filtro, se puede filtrar por cualquier orden y al aceptar, la tabla será refrescada y mostrará la línea de dicha orden. El funcionamiento de este filtro se puede observar con más detalle en las Figuras 24 y 25.



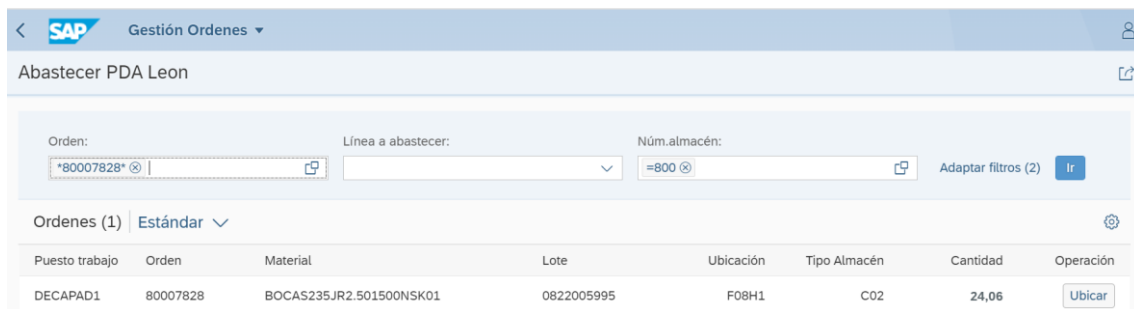
Definir condiciones: Orden

▼ Incluir (1)

Orden contiene 80007828

**Figura 24. Filtro por número de orden.**

**Fuente: Elaboración propia.**



Gestión Ordenes

Abastecer PDA Leon

Orden: \*80007828\* Línea a abastecer: Núm.almacén: =800 Adaptar filtros (2) tr

Ordenes (1) Estándar

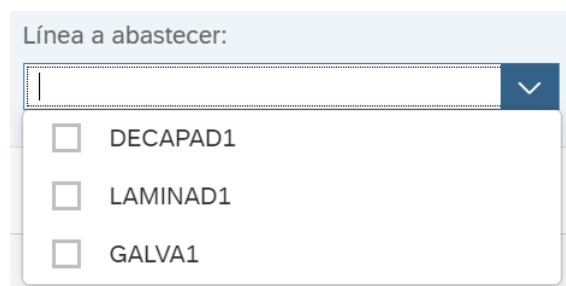
Puesto trabajo	Orden	Material	Lote	Ubicación	Tipo Almacén	Cantidad	Operación
DECAPAD1	80007828	BOCAS235JR2.501500NSK01	0822005995	F08H1	C02	24,06	Ubicar

**Figura 25. Resultado del filtraje de la orden 80007828.**

**Fuente: Elaboración propia.**

Esta funcionalidad es implementada con el objetivo de hacer más cómoda y amena la búsqueda de órdenes a la hora de querer reubicar éstas.

Por otro lado, en el filtro de línea a abastecer, aparecerán 3 opciones como ya bien se ha comentado anteriormente (laminado, decapado y galvanizado).



Línea a abastecer:

- DECAPAD1
- LAMINAD1
- GALVA1

**Figura 26. Filtraje por línea a abastecer.**

**Fuente: Elaboración propia.**

Si se selecciona cualquiera de estas tres opciones, la tabla se refrescará y aparecerán las órdenes que se tienen pendientes de ubicar en dicha línea. Cuando se pulse el botón ubicar hay dos posibilidades:

- Que la orden se ubique correctamente y, por tanto, pase de la ubicación en la que se encuentra actualmente a la nueva ubicación de la línea a abastecer: Cunas\_L, Cunas\_D o Cunas\_G. Un ejemplo de ello es, por ejemplo, querer reubicar el lote 0822003260 el cual se encuentra en la ubicación A01H1 como se puede observar en la Figura 27.

Tabla a examinar	LQUA	Cuantos	
Ctd.aciertos	1		
TmpoEjec	0	Ctd.máxima aciertos	285.279

NºA	Cuanto	Material	Centro	Lote	D	E	Nº esp.	Tp.	Ubicación	Pos	BA	BS	En	SA
800				0822003260				C01	A01H1	1				

Figura 27. Ubicación actual del lote 0822003260.

Fuente: Elaboración propia.

Dicho lote como bien se indica en la Figura 28, se requiere en la línea de producción de DECAPAD1 (línea de decapado), es decir, en la ubicación de dicha línea a abastecer, que como ya se vio anteriormente en la función ZWM\_MF\_UBICAR es la de CUNAS\_D.

Puesto trabajo	Orden	Material	Lote	Ubicación	Cantidad	Operación
DECAPAD1	80008295		0822003260	A01H1	24,52	Ubicar
DECAPAD1	80008391		0822003992	B01H3	24,56	Ubicar
DECAPAD1	80008286		0822006204	H10H1	23,82	Ubicar
DECAPAD1	80008268		0822007260	H09H1	23,63	Ubicar
DECAPAD1	80008270		0822007267	H13H1	23,87	Ubicar
DECAPAD1	80007960		0822007407	F14H1	24,62	Ubicar
			0822007407	F14H1		
				F12H1		
			0822007439	H26H1		

Figura 28. Reubicación del lote 0822003260.

Fuente: Elaboración propia.

Una vez se pulsa el botón “ubicar” se llama a la función ya analizada anteriormente “onLocatePress” y se visualiza el mensaje dando a entender que el lote ha sido reubicado correctamente.

Para verificar que ahora la ubicación actual del lote 0822003260 es la de CUNAS\_D, se accederá a la transacción “se11” o “se16n” en el back-end y se verá si en la tabla LQUA se ha actualizado la ubicación de dicho lote. Como se puede observar en la Figura 29, la ubicación ha sido actualizada correctamente a CUNAS\_D.

**LQUA: Visualización de las entradas existentes**

Tabla a examinar: LQUA      Cuantos

Ctd.aciertos: 1

TmpoEjec: 0      Ctd.máxima aciertos: 285.279

NºA	Cuanto	Material	Centro	Lote	D	E	Nº esp.	Tp.	Ubicación	Pos	BA
800				0822003260				C02	CUNAS_D	1	

**Figura 29. Nueva ubicación del lote 0822003260.**

**Fuente: Elaboración propia.**

- Que la orden no se haya podido ubicar, debido a que la ubicación de la línea a abastecer esté llena, en este caso la ubicación seguirá siendo la misma. Esto se puede ver más claramente en back-end, donde si se ejecuta la transacción “/n/iwfnfnd/error\_log” se puede observar que el SAP Gateway devuelve un error. Al producirse dicho error, en la función “onLocatePress” ya analizada anteriormente, se ejecutará directamente la sentencia de error que mostrará un mensaje diciendo que no se ha podido ubicar el lote y que se revise si la ubicación ya está completa.

El error mostrado en el SAP Gateway se puede visualizar con más detalle en la Figura 30.

SAP Gateway: Error Log

Re-Select

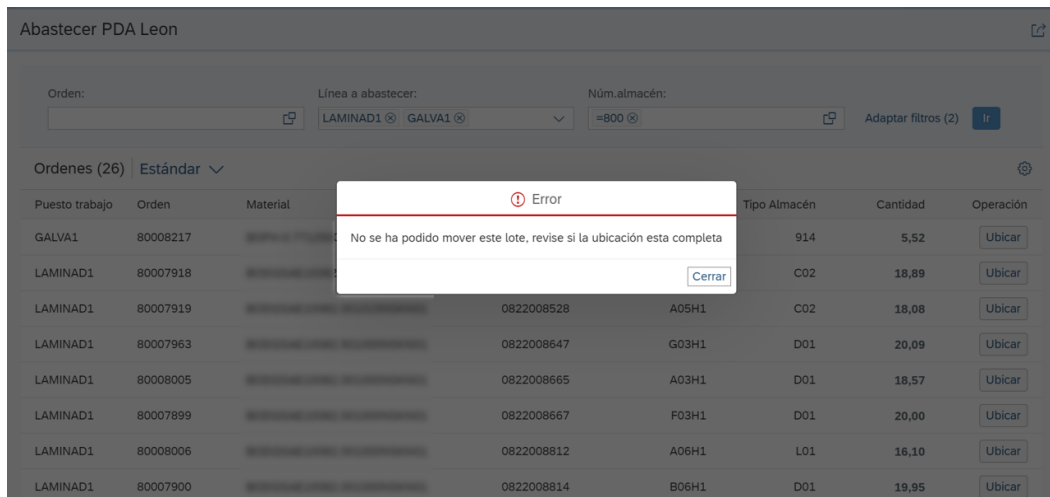
Overview

Line	Entry ...	Date	Time	User	T100 Error ID	T100 E...Error Co...	ICF No...	Request Kind	HTT...	Ba Error Text	
1		09.06.2023	08:20:07		/IWFND/CM_CONSUMER122	7	4	odata	Server OData ...	500	No se puede procesar mensaje E L3 043 en el modo plug-in HTTPS

**Figura 30. Error al reubicar el lote en el SAP Gateway.**

**Fuente: Elaboración propia.**

Por otro lado, el mensaje de error mostrado en la interfaz web, se puede ver claramente en la Figura 31.



**Figura 31. Mensaje de error al reubicar el lote en la interfaz web.**

**Fuente: Elaboración propia.**

Finalmente el filtro de número de almacén estará fijo, ya que como requisito del cliente solo se pidió que se pudiera filtrar por el número de almacén 800. [9]

## **Capítulo 4. Despliegue de la aplicación en el Launchpad del cliente**

En el presente apartado, se van a describir cada uno de los pasos dados a nivel general desde que se presenta la descripción de la aplicación hasta que ésta es desplegada en el Launchpad del cliente empresarial.

### **4.1. Pasos a seguir desde el planteamiento hasta el despliegue de la aplicación**

Los pasos que se siguen para el correcto despliegue de la aplicación en el Launchpad de la empresa cliente desde el inicio son varios y cabe resaltarlos para que se vea el proceso desde un punto de vista más general. Dichos pasos son los siguientes:

1. Desarrollo en el servidor local de SAP: El desarrollo de la aplicación SAPUI5 se lleva a cabo en el entorno local proporcionado por SAP, como WebIDE o BASS (en este caso se ha utilizado SAP WebIDE). Aquí se crea y se prueba la aplicación antes de ser desplegada.
2. Desarrollo del back-end: Junto con el desarrollo de la aplicación en el entorno local, también se desarrolla el back-end de la aplicación. Esto implica la creación y configuración de los servicios y funcionalidades necesarios para que la aplicación interactúe con los datos y procesos del sistema.

3. Generación de tile: Una vez desarrollada la aplicación, se genera un tile que servirá como acceso a la aplicación a través del Launchpad. Este tile proporciona un acceso fácil y rápido a la aplicación para los usuarios.
4. Configuración del tile en el Launchpad Designer: El siguiente paso es acceder al Launchpad Designer, que es la herramienta utilizada para personalizar y configurar el Fiori Launchpad. En el Launchpad Designer, se crea un nuevo tile y se especifica la información requerida, como el título, la descripción y la ruta de navegación de la aplicación.
5. Asociación del tile con el BSP: Una vez que se ha configurado el tile en el Launchpad Designer, se procede a asociarlo con el BSP correspondiente. Para ello, se selecciona el tile recién creado y se configura la propiedad “Navigation Targets” para especificar la ruta del BSP donde se encuentra desplegada la aplicación SAPUI5.
6. Despliegue de la aplicación en el servidor BSP: Una vez que el tile ha sido asociado correctamente con el BSP, se realiza el despliegue de la aplicación SAPUI5 en el servidor BSP. Esto implica subir los archivos de la aplicación (como los archivos HTML, JavaScript y CSS) al servidor BSP y configurar la ruta adecuada.
7. Verificación y prueba del tile: Después de haber desplegado la aplicación en el servidor BSP, se verifica que el tile se muestra correctamente en el Fiori Launchpad y que al hacer clic en él se redirige a la aplicación SAPUI5.
8. Pruebas en el entorno de calidad: Antes de pasar la aplicación al entorno de calidad, se realiza una verificación exhaustiva tanto del back-end como del front-end en el entorno de desarrollo. Se comprueba el correcto funcionamiento de la aplicación, su rendimiento y su compatibilidad con diferentes dispositivos y navegadores. Una vez que se ha verificado que la aplicación funciona correctamente, se procede a su transporte al entorno de calidad.
9. Transporte al entorno de producción: Después de que la aplicación ha sido probada y verificada en el entorno de calidad, se realiza el transporte de todos los componentes, tanto el back-end como el front-end, al entorno de producción. Aquí la aplicación estará disponible y lista para ser utilizada por los usuarios finales.

## *Capítulo 5. Conclusiones*

Una vez conseguido el objetivo principal del trabajo, se pretenden elaborar unas conclusiones, llegando a hablar de lo que ha supuesto el correcto desarrollo de la aplicación para el cliente.

### *5.1. Conclusiones*

Con el desarrollo del previo apartado en el presente trabajo, se puede dar por finalizado el proceso de desarrollo e implantación de la aplicación SAPUI5 cumpliendo con los requisitos impuestos por el cliente dándose por conseguido el objetivo principal del TFG.

El principal propósito de la aplicación fue el de mejorar la gestión de las órdenes de trabajo, permitiendo obtener un listado de órdenes las cuales eran necesarias ser reubicadas en diferentes líneas de producción. El haber conseguido tal propósito ha ayudado a la empresa cliente a obtener una mayor eficiencia en la asignación de recursos y una mejora en la planificación de las actividades logísticas.

Con el satisfactorio desarrollo de la aplicación se ha facilitado la toma de decisiones del personal encargado de la gestión del almacén. Además, se ha logrado de tal manera un seguimiento más preciso de las existencias resultando así en una reducción de errores y una optimización de la cadena de suministro.

Asimismo, también se ha logrado que la empresa cliente tenga una mayor capacidad de reorganización de los recursos y una respuesta más ágil a las necesidades de abastecimiento de las diferentes líneas de producción.

De forma resumida, se puede decir que el desarrollo de esta aplicación para el módulo SAP WM de la empresa cliente ha generado beneficios significativos para ésta al mejorar en temas de eficiencia la gestión de órdenes de trabajo contribuyendo así a obtener una mayor productividad, una mayor planificación de logística y una mayor optimización de los recursos en el almacén, fortaleciendo de tal manera el rendimiento global y la competitividad de la empresa en el mercado.



## *Bibliografía*

- [1] **Jimenez, G.** (2020, enero 10). *Integración de operaciones en la empresa. El ERP.* Portal Web Digitalicce.org. Portal de Noticias Digitales.  
<https://www.digitalicce.org/integracion-de-operaciones-en-la-empresa-el-erp-2/>
- [2] **Gonzalez, I.** (2020, junio 4). *¿Por qué centralizar los datos de mi empresa? - Corposuite. Consultoría del ERP en la nube Oracle NetSuite México.*  
<https://corposuite.com.mx/2020/06/04/por-que-centralizar-los-datos-de-mi-empresa/>
- [3] **Software de Gestion myGESTION.** *¿Por qué necesitas centralizar los datos de tu empresa?* Mygestion.com. <https://www.mygestion.com/blog/centralizar-datos>
- [4] **Romero, C.** (2022, julio 1). *Ventajas y desventajas de un ERP: todo lo que necesitas saber.* Blog del E-commerce. <https://www.tiendanube.com/blog/mx/ventajas-y-desventajas-de-erp/>
- [5] **SAP software solutions.** SAP. <http://www.sap.com/>
- [6] **Sistema SAP: características, beneficios y funciones.** Hco.com.  
<https://www.hco.com/es-us/insights/sap-beneficios-caracteristicas>
- [7] **Bengochea, D.** (2022, marzo 22). *Los 19 Mejores Módulos de SAP en 2023.* Outvio.  
<https://outvio.com/es/blog/modulos-de-sap/>
- [8] **Online, P.** (2022, febrero 7). *¿Qué es SAP - ABAP?* Profesional Online.  
<https://www.profesionalonline.com/blog/sap/que-es-sap-abap/>
- [9] **SAPUI5 SDK - Demo Kit.** <http://ui5.sap.com/>

## Anexo

### Código de las funciones en back-end.

- Función ZWM\_MF\_GET\_ORDENES:

```
FUNCTION zwm_mf_get_ordenes.
*-----
*"*Interfase local
*  IMPORTING
*    VALUE(RG_LINEA) TYPE  RSIS_T_RANGE OPTIONAL
*    VALUE(RG_ALMACEN) TYPE  RSIS_T_RANGE OPTIONAL
*    VALUE(RG_ORDEN) TYPE  RSIS_T_RANGE OPTIONAL
*  EXPORTING
*    VALUE(ET_SALIDA) TYPE  ZWM_ABAST_TT_ORDEN
*-----

DATA: lt_t320          TYPE TABLE OF t320,
      ls_return       TYPE          bapiret2,
      ls_t320         TYPE          t320,
      ls_order_objects TYPE          bapi_pp_order_objects,
      ls_operations   TYPE          bapi_order_operation1,
      lt_operations   TYPE TABLE OF bapi_order_operation1,
      lv_abastecido   TYPE char1.

SELECT caufv~aufnr, caufv~werks, jest~stat
FROM caufv AS caufv INNER JOIN jest AS jest ON caufv~objnr = jest~objnr

INNER JOIN tj02 AS tj02 ON jest~stat = tj02~istat
INNER JOIN tj02t AS tj02t ON tj02~istat = tj02t~istat
AND tj02t~spras = 'S'
WHERE werks = '8000'
AND ( jest~stat = 'I0001'
AND jest~inact <> 'X' ) OR ( jest~stat = 'I0076'
AND jest~inact <> 'X' ) AND tj02t~spras = 'S'
INTO CORRESPONDING FIELDS OF TABLE @et_salida .
DELETE et_salida WHERE aufnr NOT IN rg_orden.
LOOP AT et_salida INTO DATA(ls_stat).
  IF ls_stat-stat = 'I0076' OR ls_stat-stat <> 'I0001'.
    DELETE et_salida WHERE aufnr = ls_stat-aufnr.
  ENDIF.
ENDLOOP.

IF et_salida IS NOT INITIAL.
  SELECT afko~aufnr, afko~aufpl, afvc~arbid, crhd~arbpl FROM afko AS
afko INNER JOIN afvc AS afvc
  ON afko~aufpl = afvc~aufpl AND afvc~vornr = '0010'
  INNER JOIN crhd AS crhd ON crhd~objid = afvc~arbid
  FOR ALL ENTRIES IN @et_salida WHERE aufnr = @et_salida-
aufnr INTO TABLE @DATA(lt_puesto_trabajo).
  IF sy-subrc = 0.

    SELECT afko~aufnr, afko~rsnum, resb~rspos, resb~charg FROM afko
AS afko
    INNER JOIN resb AS resb ON afko~rsnum = resb~rsnum
```

```

FOR ALL ENTRIES IN @et_salida
WHERE afko~aufnr = @et_salida-aufnr
AND ( resb~rspos = '1' OR resb~rspos = '2' )
AND resb~charg <> ''
INTO TABLE @DATA(lt_lote).

IF sy-subrc = 0.
  LOOP AT et_salida ASSIGNING FIELD-SYMBOL(<fs_salida>).
    READ TABLE lt_puesto_trabajo INTO DATA(ls_puesto_trabajo) WI
TH KEY aufnr = <fs_salida>-aufnr.

    IF <fs_salida>-aufnr = ls_puesto_trabajo-aufnr.
      <fs_salida>-prvbe = ls_puesto_trabajo-arbpl.
    ENDIF.

    READ TABLE lt_lote INTO DATA(ls_lote) WITH KEY aufnr = <fs_s
alida>-aufnr.
    IF <fs_salida>-aufnr = ls_lote-aufnr.
      <fs_salida>-charg = ls_lote-charg.
    ENDIF.

    IF <fs_salida>-charg IS NOT INITIAL.
      SELECT SINGLE * FROM lqua INTO @DATA(ls_lqua) WHERE charg
= @<fs_salida>-charg.

      IF sy-subrc = 0.

        <fs_salida>-lgtyp = ls_lqua-lgtyp.
        <fs_salida>-lgpla = ls_lqua-lgpla .
        IF ls_lqua-lgpla = 'CUNAS_D' OR ls_lqua-
lgpla = 'CUNAS_L' OR ls_lqua-lgpla = 'CUNAS_G'.
          DELETE et_salida WHERE aufnr = <fs_salida>-aufnr.

        ELSE.
          <fs_salida>-matnr = ls_lqua-matnr.
          <fs_salida>-bdmng = ls_lqua-verme.
        ENDIF.
      ELSE.
        DELETE et_salida WHERE aufnr = <fs_salida>-aufnr.
      ENDIF.

    ENDIF.
  CLEAR ls_lqua.
  CLEAR ls_puesto_trabajo.
  CLEAR ls_lote.
ENDLOOP.
ENDIF.
ENDIF.
ENDIF.

SORT et_salida BY prvbe charg.
DELETE et_salida WHERE prvbe NOT IN rg_linea OR charg IS INITIAL.

ENDFUNCTION.

```

- Función ZWM\_MF\_SH\_LINEA:

```

FUNCTION ZWM_MF_SH_LINEA.
*"-----
--
*"**"Interfase local
*"  IMPORTING
*"    VALUE(RG_LINEA) TYPE  RSIS_T_RANGE OPTIONAL
*"  EXPORTING
*"    VALUE(ET_SALIDA) TYPE  ZWM_ABAST_TT_LINEA
*"-----
--

DATA ls_salida TYPE ZWM_ABAST_S_LINEA.
ls_salida-prvbe = 'DECAPAD1'.
APPEND ls_salida TO et_salida.
ls_salida-prvbe = 'LAMINAD1'.
APPEND ls_salida TO et_salida.
ls_salida-prvbe = 'GALVAL1'.
APPEND ls_salida TO et_salida.
CLEAR ls_salida.
APPEND ls_salida TO et_salida.

ENDFUNCTION.

```

- Función ZWM\_MF\_UBICAR:

```

FUNCTION zwm_mf_ubicar.
*"-----
--
*"**"Local Interface:
*"  IMPORTING
*"    VALUE(AUFNR) TYPE  AUFNR
*"  EXPORTING
*"    VALUE(OT_MSG) TYPE  BAPIRET2_T
*"    VALUE(OS_ORDEN) TYPE  ZWM_ABAST_S_ORDEN
*"-----
--

DATA: number TYPE bapi_order_key-order_number.
DATA: ls_order_objects TYPE bapi_pp_order_objects.
DATA: lt_header_bapi TYPE TABLE OF bapi_order_header1.
DATA: lt_component TYPE TABLE OF bapi_order_component.
DATA: return TYPE TABLE OF bapiret2.
ls_order_objects-header = 'X'.
ls_order_objects-components = 'X'.
ls_order_objects-operations = 'X'.
number = aufnr.
DATA: ls_return TYPE bapiret2.
DATA: lt_operation TYPE TABLE OF bapi_order_operation1.

CALL FUNCTION 'BAPI_PRODORD_GET_DETAIL'
  EXPORTING

```

```

        number          = number
        order_objects  = ls_order_objects
*   IMPORTING
*   return            = return
TABLES
    header           = lt_header_bapi
    component        = lt_component
    operation        = lt_operation.

LOOP AT lt_header_bapi INTO DATA(ls_header).
ENDLOOP.

LOOP AT lt_component INTO DATA(ls_component).
    EXIT.
ENDLOOP.

LOOP AT lt_operation INTO DATA(ls_operation).
    EXIT.
ENDLOOP.

DATA(lv_status) = ls_header-system_status.
DATA(lv_lote) = ls_component-batch.

DATA(lv_puesto_trabajo) = ls_operation-work_center.
DATA(lv_equivalencia) = 'GALVA1'.
DATA(lv_equivalencia_destino) = ''.

IF ( lv_status CS 'NOTI' )
OR ( lv_status CS 'CERR' )
OR ( lv_status CS 'INBO' )
OR ( lv_status CS 'PTBO' ).

    IF sy-subrc <> 0.
        APPEND VALUE bapiret2( type      = sy-msgty
                                id        = sy-msgid
                                number    = sy-msgno
                                message_v1 = sy-msgv1
                                message_v2 = sy-msgv2
                                message_v3 = sy-msgv3
                                message_v4 = sy-msgv4 ) TO ot_msg.

    ENDIF.
ELSE.

    SELECT SINGLE * FROM lqua INTO @DATA(ls_lqua) WHERE charg = @lv_lote.
    IF sy-subrc = 0.

        DATA: i_lgnum TYPE ltak-lgnum VALUE '800'.
        DATA: i_bwlvs TYPE ltak-bwlvs VALUE '999'.
        DATA: i_matnr TYPE ltap-matnr.
        i_matnr = ls_lqua-matnr.
        DATA: i_werks TYPE ltap-werks VALUE '8000'.
        DATA: i_lgort TYPE ltap-lgort VALUE '8000'.
        DATA: i_charg TYPE ltap-charg.
        i_charg = lv_lote.

```

*\*Ubicaciones origen: se saca de LQUA*

```

*I_VLTYP : TIPO almacen origen LGTYP
*I_VLPLA : tipo ubicacion origen LGPLA
  DATA: i_vltyp TYPE ltap-vltyp.
  DATA: i_vlpla TYPE ltap-vlpla.
  i_vltyp = ls_lqua-lgtyp.
  i_vlpla = ls_lqua-lgpla.

*Cunas_D para la línea Decapado
*Cunas_L para la línea Laminado
*Cunas_G para la línea Galvanizado
* Ubicaciones: Destino
  DATA: i_nltyp TYPE ltap-nltyp.
  DATA: i_nlpla TYPE ltap-nlpla.

IF lv_puesto_trabajo = 'DECAPAD1'.
  i_nlpla = 'CUNAS_D'.
  i_nltyp = 'C02'.
ELSEIF
  lv_puesto_trabajo = 'LAMINAD1'.
  i_nlpla = 'CUNAS_L'.
  i_nltyp = 'D01'.
ELSEIF
  lv_puesto_trabajo = 'GALVA1'.
  i_nlpla = 'CUNAS_G'.
  i_nltyp = 'L02'.
ENDIF.
DATA: i_altme TYPE ltap-altme.
DATA: i_anfme TYPE r103t-anfme.
i_anfme = ls_lqua-verme.

CALL FUNCTION 'L_TO_CREATE_SINGLE'
EXPORTING
  i_lgnum = i_lgnum
  i_bwlvs = i_bwlvs
  i_matnr = i_matnr
  i_werks = i_werks
  i_lgort = i_lgort
  i_charg = i_charg
  i_anfme = i_anfme
  i_altme = ' '
  i_vltyp = i_vltyp
  i_vlpla = i_vlpla
  i_nltyp = i_nltyp
  i_nlpla = i_nlpla

EXCEPTIONS
  NO_TO_CREATED           = 1
  BWLVS_WRONG            = 2
  BETYP_WRONG            = 3
  BENUM_MISSING         = 4
  BETYP_MISSING         = 5
  FOREIGN_LOCK          = 6
  VLTYP_WRONG           = 7
  VLPLA_WRONG           = 8
  VLTYP_MISSING         = 9
  NLTYP_WRONG           = 10
  NLPLA_WRONG           = 11

```

```

NLTYP_MISSING           = 12
RLTYP_WRONG             = 13
RLPLA_WRONG            = 14
RLTYP_MISSING          = 15
SQUIT_FORBIDDEN        = 16
MANUAL_TO_FORBIDDEN    = 17
LETYP_WRONG             = 18
VLPLA_MISSING          = 19
NLPLA_MISSING          = 20
SOBKZ_WRONG            = 21
SOBKZ_MISSING          = 22
SONUM_MISSING          = 23
BESTQ_WRONG            = 24
LGBER_WRONG            = 25
XFELD_WRONG            = 26
DATE_WRONG             = 27
DRUKZ_WRONG            = 28
LDEST_WRONG            = 29
UPDATE_WITHOUT_COMMIT  = 30
NO_AUTHORITY           = 31
MATERIAL_NOT_FOUND     = 32
LENUM_WRONG            = 33
OTHERS = 34

```

```

.
IF sy-subrc <> 0.
    APPEND VALUE bapiret2( type      = sy-msgty
                           id        = sy-msgid
                           number    = sy-msgno
                           message_v1 = sy-msgv1
                           message_v2 = sy-msgv2
                           message_v3 = sy-msgv3
                           message_v4 = sy-msgv4 ) TO ot_msg.
* Implement suitable error handling here
ENDIF.

ENDIF.

ENDIF.
ENDFUNCTION.

```

Método /iwbsp/if\_mgw\_appl\_srv\_runtime~execute\_action en la creación del modelo de datos:

```
METHOD /iwbsp/if_mgw_appl_srv_runtime~execute_action.
```

```
DATA:
```

```

ls_parameter TYPE /iwbsp/s_mgw_name_value_pair,
ls_msg       TYPE bapiret2,
lt_msg       TYPE bapiret2_t,
lv_error     TYPE string.

```

```
DATA:
```

```
lo_filter TYPE REF TO /iwbsp/if_mgw_req_filter,
```

```

ls_key_tab          TYPE /iwbep/s_mgw_name_value_pair,
lt_select_options  TYPE /iwbep/t_mgw_select_option.

DATA:
lv_aufnr TYPE aufnr,
ls_orden TYPE zwm_abast_s_orden.

IF iv_action_name = 'Ubicar'.      "Imprime la etiqueta seleccionada

READ TABLE it_parameter INTO ls_parameter WITH KEY name = 'Aufnr
'.

IF sy-subrc = 0.
  lv_aufnr = ls_parameter-value.
ENDIF.

CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
  EXPORTING
    input  = lv_aufnr
  IMPORTING
    output = lv_aufnr.

CALL FUNCTION 'ZWM_MF_UBICAR'
  EXPORTING
    aufnr  = lv_aufnr
  IMPORTING
    ot_msg = lt_msg
    os_orden = ls_orden.

IF lt_msg IS INITIAL.
  "Si todo ha ido bien, devolvemos la estructura de la incidencia
  copy_data_to_ref(
    EXPORTING
      is_data = ls_orden
    CHANGING
      cr_data = er_data ).
ENDIF.

ENDIF.

IF lt_msg IS NOT INITIAL.
  READ TABLE lt_msg INTO ls_msg INDEX 1.
  IF sy-subrc = 0.
    lv_error = ls_msg-message.
    RAISE EXCEPTION TYPE /iwbep/cx_mgw_busi_exception
      EXPORTING
        textid          = /iwbep/cx_mgw_busi_exception=>business
        _error_unlimited
        message_unlimited = lv_error.
  ENDIF.
ENDIF.

ENDMETHOD.

```