

# Especificación de Diseño Funcional Front End (SAP) Solicitud de Pedido

Elaboró	Revisó	Autorizó	Fecha
Ing. Bernabé Gutierrez Rodríguez	Oscar Castañeda	Arturo Eugenio Sánchez	17-DIC-18
			Revisión
			02

## Índice

### Contenido

<b>0. INTRODUCCIÓN.....</b>	<b>4</b>
<b>1. OBJETIVO GENERAL.....</b>	<b>4</b>
Objetivo específico .....	4
<b>2. ESCENARIO ACTUAL.....</b>	<b>4</b>
<b>3. DESCRIPCIÓN DE LA TECNOLOGIA APLICADA.....</b>	<b>4</b>
Front End <Capa 1> .....	5
Target devices. ....	6
Interfaz web app: Login .....	7
Interfaz web: Nueva SOLPE .....	8
RockJS <Capa 2> .....	10
SAP <Capa 3> .....	10
Historia de Usuario .....	12
<b>4. MODELO DE DISEÑO.....</b>	<b>13</b>
<b>5. REQUERIMIENTOS FUNCIONALES .....</b>	<b>14</b>
6.1 WSDL – Elemento <Definicion> .....	14
6.2 WSDL – Elemento <tipos> .....	15
6.3 WSDL – Elemento <mensaje> .....	16
6.4 WSDL – Elemento <portType> .....	16
6.5 WSDL – Elemento <binding> .....	17
6.6 WSDL – Elemento <ports> .....	18
6.7 WSDL – Elemento <service> .....	18
<b>7 Requerimiento funcional 2.....</b>	<b>19</b>
7.1 Consumo del Servicio Web SOAP .....	19
7.2 Descripción del API WSDL SAP. ....	19
7.3 Definición de la autenticación SAP .....	20
7.4 Definir WDSL .....	20

7.5	Creación del objeto SoapClient	20
7.6	Definición de la función para dar de alta una SOLPE	20
7.7	Tipos de dato del ZwsPosPedido	21
7.8	Excepciones y posibles errores	21
7.9	Datos Maestros	22
7.10	Ejemplo de consumo con PHP	23



## 0. INTRODUCCIÓN

El presente Procedimiento se alinea a la Filosofía\* Corporativa de la Organización, en donde el Personal del Área de Desarrollo de software desarrolla sus actividades guiado por la Estrategia de la Organización y toma en cuenta la Política de Calidad y el Objetivo de Calidad que en nuestro particular expresa:

### **Objetivo de Calidad de Desarrollo de software:**

*Contribuir al cumplimiento de la Política de Calidad de la Organización alcanzando la efectividad (eficacia y/o eficiencia) de sus procesos: RockJS01DS.*

En el desempeño de nuestros procesos - bajo un liderazgo comprometido - damos atención a las necesidades y expectativas de nuestros Clientes, entendiendo el entorno de la Organización y planificando la prestación del servicio y/o producto que proporcionamos, con un enfoque a procesos; verificando y validando cada etapa. Los roles, responsabilidades y autoridades de cada colaborador están definidas. Atendemos a los Indicadores de nuestros Procesos y analizamos los factores de riesgo para tomar su control y mitigar o anular sus consecuencias. Nuestro sistema de auditorías y otras formas de verificación nos permiten controlar el producto no conforme y realizar acciones correctivas que incorporan la mejora continua al SGC. Implementamos los cambios que nos lleven a superar las necesidades y expectativas de nuestros clientes internos/externos, incrementando su satisfacción.

\* Misión, Visión; y Valores

## 1. OBJETIVO GENERAL

Este documento describe en lenguaje técnico, cómo se construirá la solución para crear solicitud de pedido (**VA01**) en el sistema SAP a través de RockJS. Mediante el consumo de servicios web (REST y SOAP) consumibles desde la web.

En este proyecto se consumirán los servicios web SOAP del lado de SAP (web service) mediante los cuales se habilitará un canal de comunicación en línea para dar de alta solicitudes de pedido (SOLPE). Dicho servicio SOAP será convertido a un servicio REST mediante RockJS y consumido por interfaces web.

### **Objetivo específico**

Desarrollar una aplicación web, usando tecnologías HTML, JS y CSS3 para el consumo de Servicio Web SOAP.

## 2. ESCENARIO ACTUAL

Actualmente no hay servicios que consuman Web Services. El proyecto es nuevo.

El presente requerimiento describe técnicamente como el usuario puede dar alta ventas en dispositivos móviles o en su computadora personal.

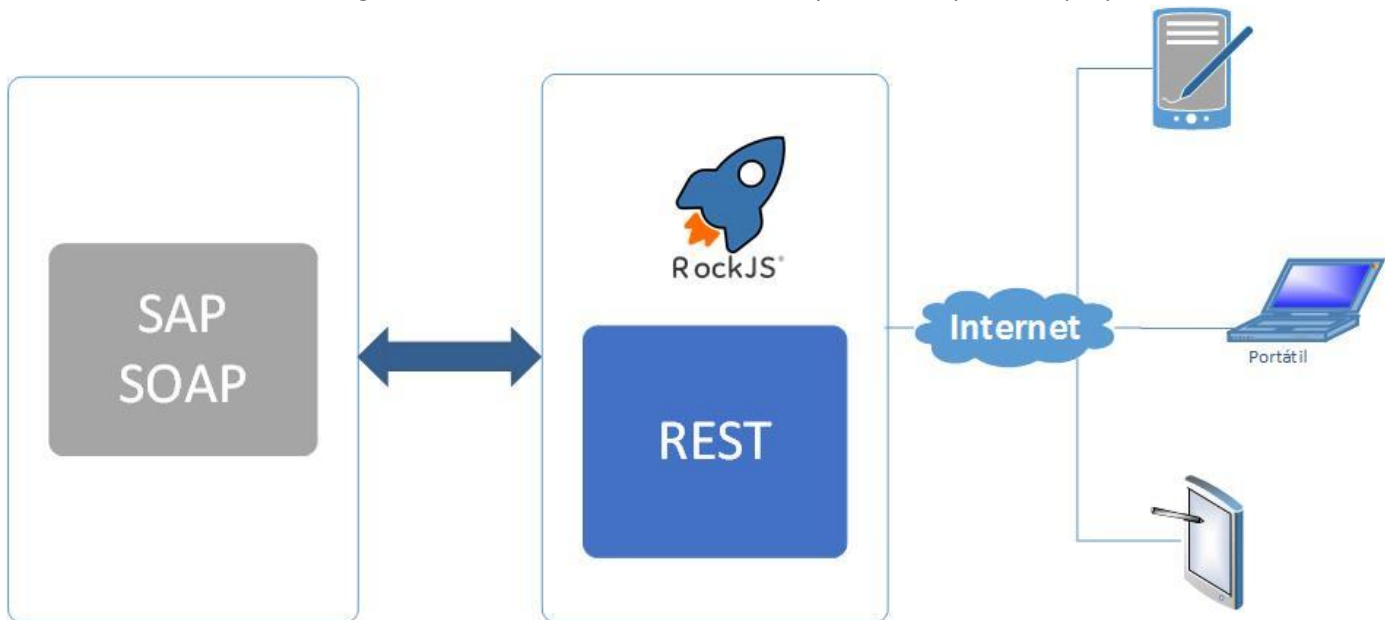
## 3. DESCRIPCIÓN DE LA TECNOLOGÍA APLICADA

Para lograr la creación de aplicaciones web/móvil que interactúan con el sistema SAP, la propuesta de solución está basada en la arquitectura de software modelo MVS (Modelo/Vista/Controlador) que se presenta a continuación.

1. **Front End <Vistas web>**
2. **Controlador <RockJS >**
3. **Back End <SAP>**



Ahora se muestra el diagrama de la estructura de la aplicación que se propone como solución.



En el diagrama se ven las fases o capas por las cuales viaja la información, cada agente en esta arquitectura trabaja de manera independiente llevando a cabo su función. El proceso integrado da como resultado el alta de una SOLPE en el sistema SAP.

#### Front End <Capa 1>

Se desarrollará una interfaz Web del modelo SAP de la SOLPE Standard para lograr transacciones en el sistema SAP expuesto mediante un servicio WEB SOAP XML, su función será:

- a. Crear Solicitudes de pedido standard (VA-01).

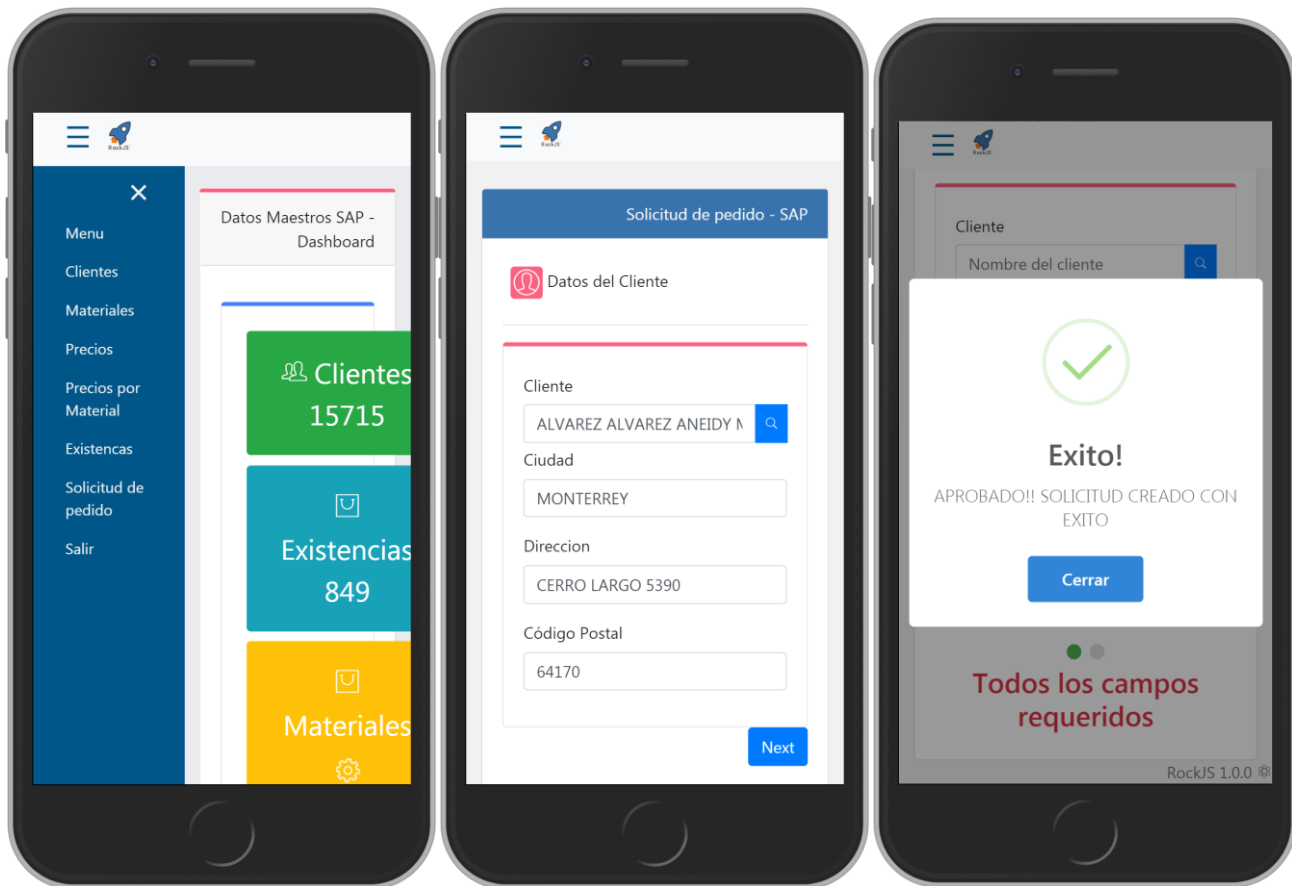
Para la creación de interfaces web se proponen los siguientes elementos:

- a. **HTML5**: es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet
- b. **CSS (Bootstrap 4)**: es un framework que simplifica el proceso de creación de diseños web combinando CSS y JavaScript.
- c. **JavaScript**: es un lenguaje orientado a objetos, que permite mejoras en la interfaz de usuario y paginas web
- d. **Ajax**: Ajax es una tecnología asíncrona, los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página
- e. **JQuery**: es una framework de interfaz gráfica de usuario basado en HTML5, diseñado para construir páginas webs y aplicaciones, que sean accesibles por teléfonos inteligentes, tabletas, y PC de escritorio.

Las Interfaces web están propuestas en base a las mejores prácticas para el desarrollo web actual. La interfaz está diseñada de manera secuencial (pasos o secciones), de modo que el usuario de la aplicación, pueda llenar los campos de manera fácil y sencilla, mostrándole durante el proceso mensajes de ayuda. Una vez que se ha llenado correctamente la forma, se envían los datos a RockJS para el procesamiento en SAP.

Las interfaces pueden ser ajustadas a los requerimientos funcionales que se requieran. De tal manera que la capa de Front End no interfiere con el Back End de SAP.

Interfaces de ejemplo:



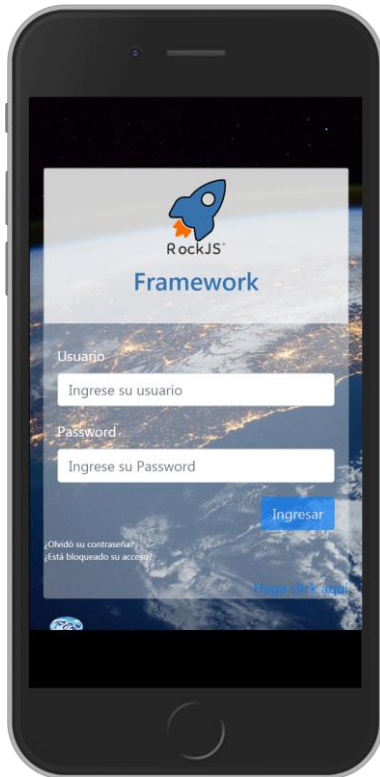
#### Target devices.

- Desktop computers
- Laptops
- Tablets (con resolución de 1024 píxeles ancho como mínimo)
- Smartphone (con resolución de 7 pulgadas)

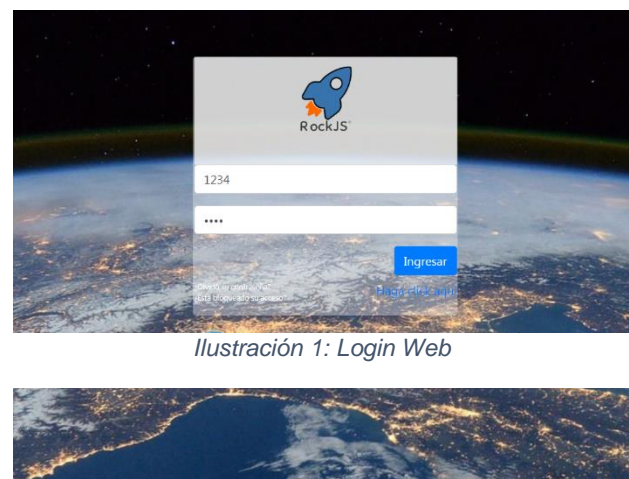
**Interfaz web app: Login**

La aplicación cuenta con una interfaz en la que el usuario podrá loguearse para entrar a los recursos. Proporcionando sus credenciales de acceso la aplicación determina si es un usuario cuenta con un registro valido para el acceso.

En esta interfaz se podrá visualizar la página que solicita los datos Usuario y Password y un botón de acceso.



*Ilustración 2: Login App*

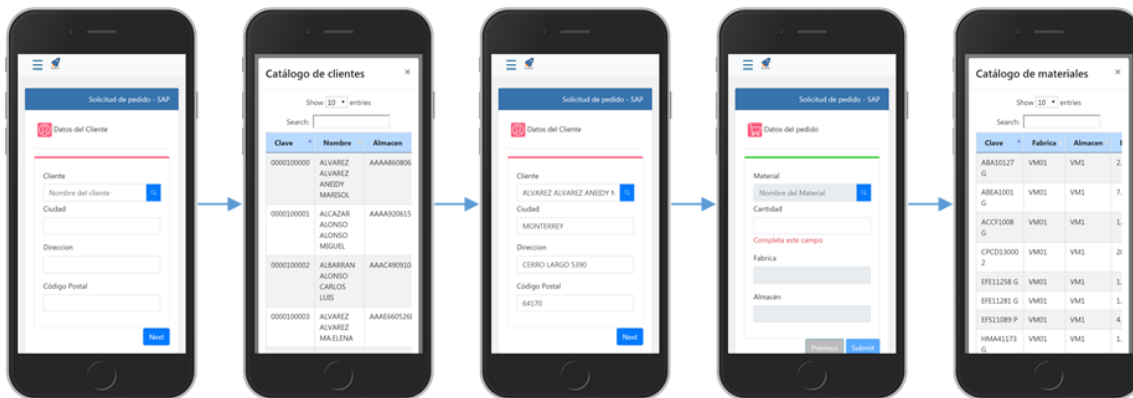


*Ilustración 1: Login Web*



### Interfaz web: Nueva SOLPE

Desde el menú superior de la aplicación, se habilitará un link que permitirá la creación de una nueva solicitud de pedido. En esta pantalla, se podrán capturar los datos necesarios para tal fin. Debido a que los campos a capturar son demasiados, la interfaz estará separada en pasos secuenciales a modo de que sea un proceso sencillo y fácil para el usuario de la aplicación.



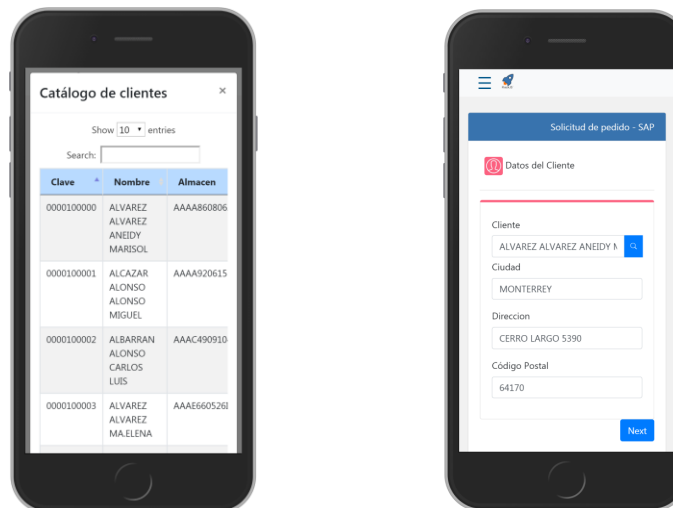
El usuario podrá consultar el catálogo de clientes pulsando el botón que se encuentra junto a la casilla de captura.

#### Cliente

Nombre del cliente



Al realizar esta acción, aparecerá una ventana emergente que contendrá una tabla con la lista de clientes, el usuario podrá filtrar en orden alfabético pulsando la columna que desea filtrar, si la columna es numérica, se filtrará de mayor a menor o viceversa. Al dar clic sobre una fila, las casillas de texto se llenarán con la información del cliente, para avanzar deberá presionar el botón **Next**







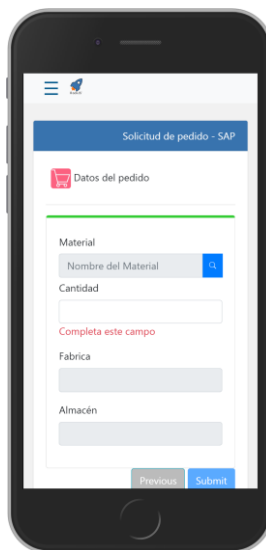
El usuario podrá consultar el catálogo de materiales pulsando el botón que se encuentra junto a la casilla de captura.

### Material

Nombre del Material

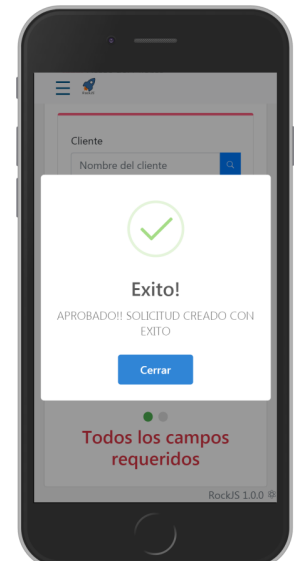
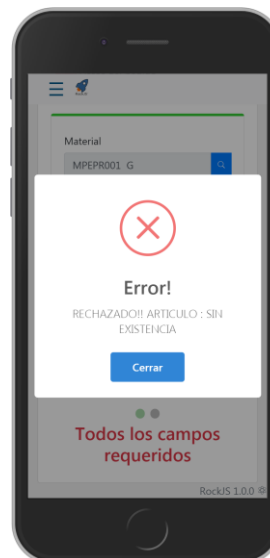


Al realizar esta acción, aparecerá una ventana emergente que contendrá una tabla con la lista de materiales, el usuario podrá filtrar en orden alfabético pulsando la columna que desea filtrar, si la columna es numérica, se filtrará de mayor a menor o viceversa. Al dar clic sobre una fila, las casillas de texto se llenarán con la información del cliente, para crear una SOLPE el usuario deberá presionar el botón **Submit**.



El usuario deberá proporcionar aquellos datos que sean obligatorios, por ejemplo la cantidad del material solicitado y en caso de no proporcionarlos, aparecerán mensajes de ayuda para el usuario.

Al pulsar el botón **Submit**, se procesará la petición y devolverá mensajes de validación al usuario en caso de éxito o error



**RockJS <Capa 2>**

Para el controlador entre el Front End y el Back End (SAP) se usara RockJS el cual está constituido por los siguientes elementos:

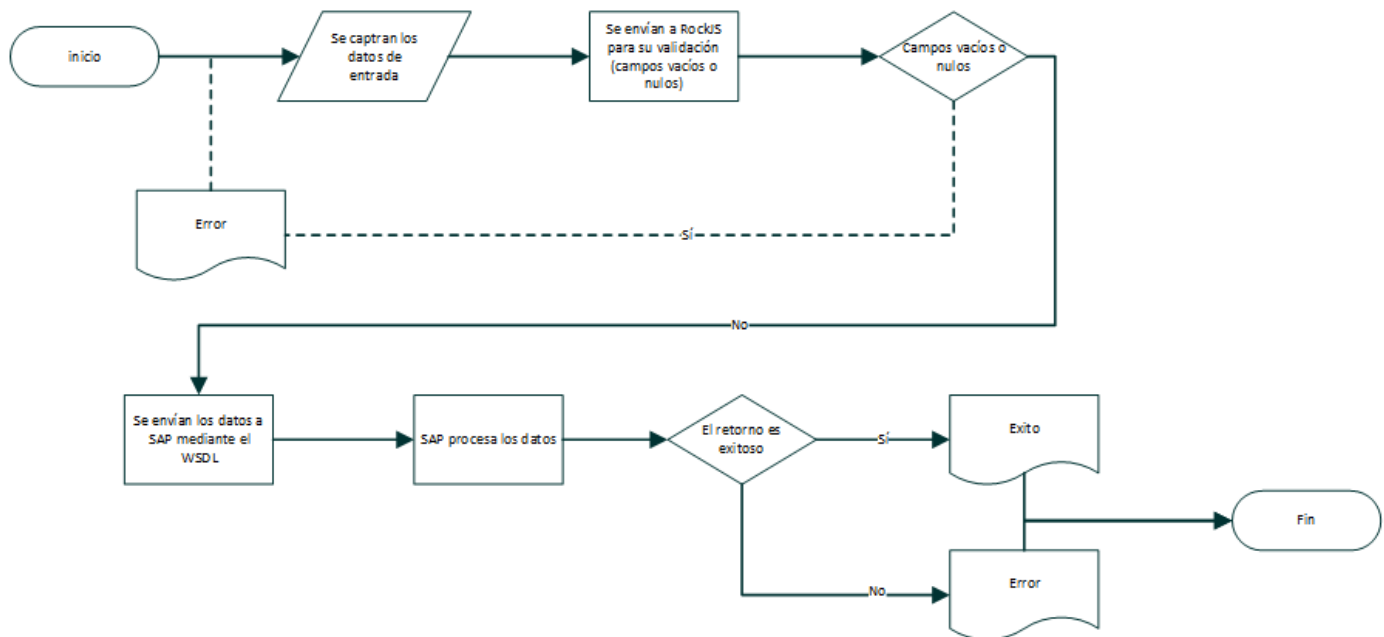
- PHP 7.1.3
- Extensión XML PHP
- Extensión JSON PHP

RockJS será quien consuma el Servicio Web SOAP, también determinara las funciones que contiene el WSDL, mismas que darán pauta a la creación de interfaces y Servicios Web propios de RockJS.

**SAP <Capa 3>**

El Back End está a cargo de SAP, el cual expone el servicio web de consumo SOAP, ofreciendo métodos y funciones que reciben parámetros (datos), para dar de alta una nueva SOLPE.

El siguiente diagrama muestra el comportamiento dinámico de la información a través del alta del servicio SOAP



*Ilustración 3: diagrama de procesos*

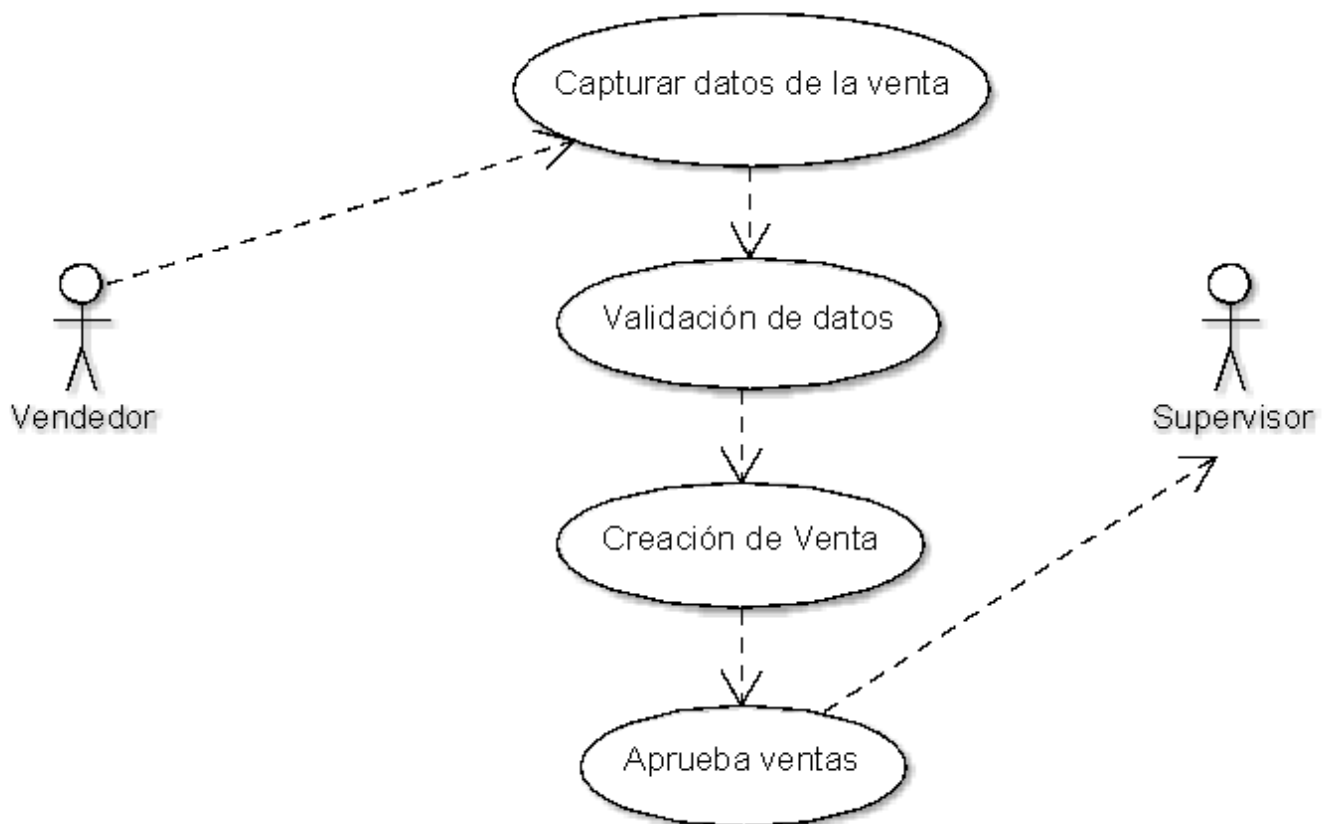


Ilustración 4: diagrama de caso de uso

El siguiente diagrama muestra la secuencia de la operación que se efectuarán con el fin de crear una venta.

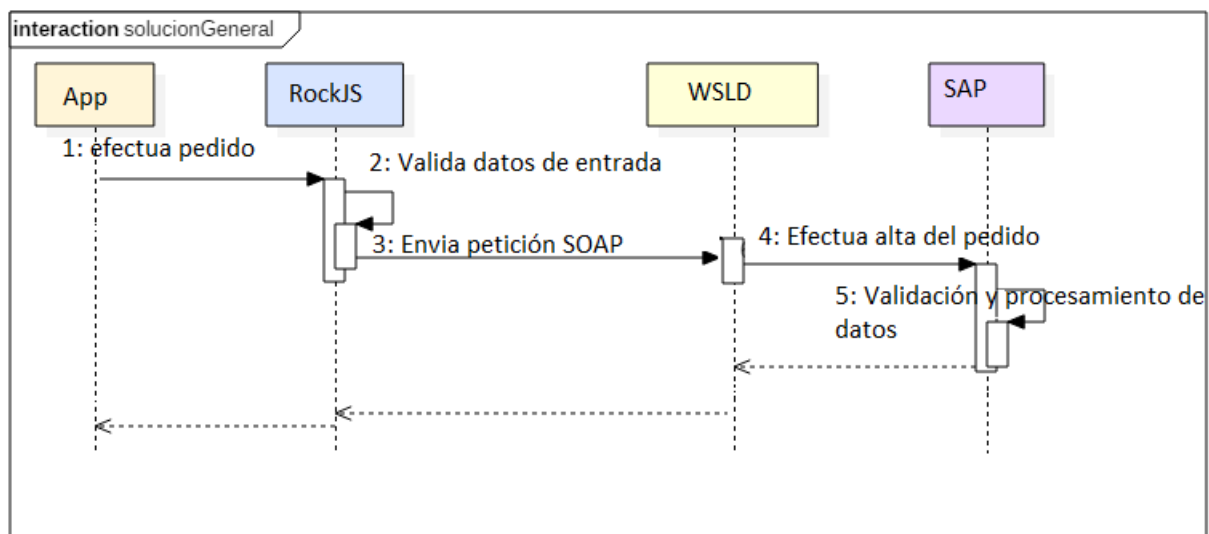


Ilustración 5: diagrama de secuencia

**Historia de Usuario**

A continuación se muestra la historia de usuario clave que servirá para mostrar la funcionalidad de la aplicación. La historia incluye la carga de trabajo necesaria que se ha estimado para implementarla y que ésta funcione correctamente.

<b>Título:</b> Crear SOLPE	
<b>Historia:</b> Como usuario quiero crear una Solicitud de pedido en línea. Podré capturar de forma fácil los datos para poder lograrlo.	
<b>Pantallas:</b> Creación de SOLPE	
Pruebas Validación: <ul style="list-style-type: none"> <li>• Validar que el usuario puede acceder a la pantalla de creación de usuario desde el menú de la aplicación.</li> <li>• Validar que el usuario capture de manera correcta los datos del formulario.</li> <li>• Validar que el usuario puede crear una nueva Solicitud de pedido.</li> <li>• Validar que los datos se almacenan correctamente.</li> <li>• Validar que la aplicación muestre al usuario el estatus del alta (Éxito o Errores).</li> </ul>	
<b>Carga de trabajo:</b> 2 Semanas	<b>Prioridad:</b> Alta



## EXPOSICION DE SERVICIOS SOAP A REST

Se consumirán una serie de servicios web SOAP expuestos por SAP, que permitirán realizar Consultas de Materiales, Precios, Existencias, Clientes y Solicitudes de Pedido. Al mismo tiempo RockJS convertirá dichos servicios de la arquitectura SOAP a REST, y de esta manera hacer la interfaz más amigable.

La exposición de los servicios SAP se presenta a continuación:

### WSDL SAP

En base al WSDL de SAP, RockJS es capaz de determinar con que funciones cuenta dicho servicio.

Método	Liga	Descripción
ZwsPosPedido	<a href="http://201.166.145.80:8000/sap/bc/srt/wsd/rlv_10002A111AD1/bndg_url/sap/bc/srt/rfc/sap/zws_crea_pedido/500/zws_crea_pedido/zws_crea_pedido?sap-client=500/&lt;Parametros&gt;">http://201.166.145.80:8000/sap/bc/srt/wsd/rlv_10002A111AD1/bndg_url/sap/bc/srt/rfc/sap/zws_crea_pedido/500/zws_crea_pedido/zws_crea_pedido?sap-client=500/&lt;Parametros&gt;</a>	Crea una SOLPE
ZwsDataPto	<a href="http://s4qa:8000/sap/bc/srt/wsd/rlv_10002A101AD1/bndg_url/sap/bc/srt/rfc/sap/zws_datos_maestros/500/zws_datos_maestros/zws_datos_maestros?sap-client=500">http://s4qa:8000/sap/bc/srt/wsd/rlv_10002A101AD1/bndg_url/sap/bc/srt/rfc/sap/zws_datos_maestros/500/zws_datos_maestros/zws_datos_maestros?sap-client=500</a>	Consulta Datos maestros SAP
ZwsPosLog	<a href="http://s4qa:8000/sap/bc/srt/wsd/rlv_10002A101AD1/bndg_url/sap/bc/srt/rfc/sap/zws_pos_log/500/zws_pos_log/zws_pos_log?sap-client=500">http://s4qa:8000/sap/bc/srt/wsd/rlv_10002A101AD1/bndg_url/sap/bc/srt/rfc/sap/zws_pos_log/500/zws_pos_log/zws_pos_log?sap-client=500</a>	Consulta estatus de Solpes

## 4. MODELO DE DISEÑO

El alcance de los requerimientos es para las siguientes funciones:

Tabla		
1	ZwsPosPedido	Crea una solicitud de pedido
2	ZwsDataPto	Consulta el maestro de datos de SAP
3	ZwsPosLog	Consulta el log de tickets generados (SOLPES)

Dentro de esta sección se detallan los requerimientos que especifican la funcionalidad del software que los desarrolladores deben construir en el producto para posibilitar a los usuarios dar de alta una SOLPE en SAP.

La pantallas indican la funcionalidad requerida, el diseño & look&feel puede variar de acuerdo a las herramientas utilizadas en la implementación.



## 5. REQUERIMIENTOS FUNCIONALES

Nombre del requerimiento	Analizar WSDL para crear una SOLPE en SAP y para definir los campos y estructura de datos
--------------------------	---

En este caso el WSDL para la creación de un SOLPE, SAP provee un solo método llamado `ZwsPosPedido` el cual recibe como parámetros un documento XML que contiene los datos para generar una SOLPE. Además de datos otros datos adicionales como los headers de autorización.

Se requiere que mediante el WSDL proporcionado por el equipo de Viko ([http://201.166.145.80:8000/sap/bc/srt/wSDL/flv\\_10002A111AD1/bndg\\_url/sap/bc/srt/rfc/sap/zws\\_crea\\_pedido/500/zws\\_crea\\_pedido/zws\\_crea\\_pedido?sap-client=500](http://201.166.145.80:8000/sap/bc/srt/wSDL/flv_10002A111AD1/bndg_url/sap/bc/srt/rfc/sap/zws_crea_pedido/500/zws_crea_pedido/zws_crea_pedido?sap-client=500)) se registre la estructura de datos, métodos y funciones.

Para obtener la información de la estructura de datos del WSDL se debe tener en cuenta

- **Definiciones**
- **Tipo:** utilizando tipos de datos integrados y se definen en XMLSchema.
- **Mensaje:** Request y Response
- **Tipo de puerto:** operación que consiste en una solicitud y un servicio de respuesta.
- **Enlace:** Dirección para utilizar el protocolo de transporte HTTP SOAP.
- **Servicio** - Servicio disponible con el URI
- **Puerto:** asocia el enlace con el URI donde se puede acceder al servicio en ejecución.

### 6.1 WSDL – Elemento <Definicion>

```
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsoap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="urn:sap-
com:document:sap:soap:functions:mc-style"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:n1="urn:sap-
com:document:sap:rfc:functions" targetNamespace="urn:sap-
com:document:sap:soap:functions:mc-style">
```

Del fragmento anterior, podemos concluir que las definiciones.

- Es un contenedor de todos los demás elementos.
- Especifica un atributo `targetNamespace`. El `targetNamespace` es una convención de esquema XML que permite el documento WSDL para referirse a sí mismo.
- Especifica un espacio de nombres predeterminado: `xmlns` = <http://schemas.xmlsoap.org/wsdl/>. Por lo tanto, se supone que todos los elementos sin un prefijo de espacio de nombres, como `message` o `portType`, forman parte del espacio de nombres WSDL predeterminado.
- Especifica numerosos espacios de nombres que se utilizan a lo largo del resto del documento.



## 6.2 WSDL – Elemento <tipos>

Un servicio web necesita definir sus entradas y salidas y cómo se asignan dentro y fuera de los servicios. El elemento <tipos> WSDL se encarga de definir los tipos de datos que utiliza el servicio web. Los tipos son documentos XML, o partes de documentos.

- El elemento **types** describe todos los tipos de datos utilizados entre el cliente y el servidor.
- WSDL no está vinculado exclusivamente a un sistema de escritura específico.
- WSDL utiliza la especificación del esquema XML del W3C como su opción predeterminada para definir los tipos de datos.
- Si el servicio utiliza solo tipos simples incorporados de esquema XML, como cadenas y enteros, no se requiere el elemento de tipos.
- WSDL permite que los tipos se definan en elementos separados para que los tipos puedan reutilizarse con múltiples servicios web.

```
<xsd:schema attributeFormDefault="qualified" targetNamespace="urn:sap-
com:document:sap:rft:functions">
  <xsd:simpleType name="char1">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="1"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="char10">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="10"/>
    </xsd:restriction>
  </xsd:simpleType>
```

...

Parámetros de entrada del WSDL, que conforma la estructura del servicio.

```
<xsd:complexType name="ZposPedido">
  <xsd:sequence>
    <xsd:element name="Mandt" type="n0:clnt3"/>
    <xsd:element name="Ticket" type="n0:char20"/>
    <xsd:element name="Factura" type="n0:char20"/>
    <xsd:element name="DocType" type="n0:char4"/>
    <xsd:element name="FechaAlta" type="tns:char10"/>
    <xsd:element name="SalesOrg" type="n0:char4"/>
    <xsd:element name="DistrChan" type="n0:char2"/>
    <xsd:element name="Division" type="n0:char2"/>
    <xsd:element name="SalesOff" type="n0:char4"/>
    <xsd:element name="PurchNoC" type="n0:char35"/>
    <xsd:element name="Pmnttrms" type="n0:char4"/>
    <xsd:element name="PartnRole" type="n0:char2"/>
    <xsd:element name="PartnNumb" type="n0:char10"/>
    <xsd:element name="Country" type="n0:char1"/>
    <xsd:element name="Langu" type="n0:lang"/>
    <xsd:element name="Name" type="n0:char35"/>
    <xsd:element name="City" type="n0:char35"/>
    <xsd:element name="Street" type="n0:char35"/>
    <xsd:element name="PostlCode" type="n0:char10"/>
```



```

<xsd:element name="Status" type="n0:char10"/>
<xsd:element name="Registros" type="xsd:unsignedByte"/>
<xsd:element name="FechaCurrent" type="tns:date10"/>
<xsd:element name="PedidoSap" type="tns:char10"/>
<xsd:element name="EntregaSap" type="tns:char10"/>
<xsd:element name="FacturaSap" type="tns:char10"/>
</xsd:sequence>
</xsd:complexType>

```

Los tipos de datos resuelven el problema de identificar los tipos de datos y los formatos que pretende utilizar con sus servicios web. La información de tipo se comparte entre el remitente y el receptor. Por lo tanto, los destinatarios de los mensajes necesitan acceso a la información que utilizó para codificar sus datos y deben entender cómo decodificar los datos.

### 6.3 WSDL – Elemento <mensaje>

El elemento <mensaje> describe los datos que se intercambian entre los proveedores de servicios web y los consumidores.

- Cada servicio web tiene dos mensajes: **entrada** y **salida**.
- La entrada describe los parámetros para el servicio web y la salida describe los datos de retorno del servicio web.
- Cada mensaje contiene cero o más parámetros <part>, uno para cada parámetro de la función de servicio web.
- Cada parámetro <part> se asocia con un tipo concreto definido en el elemento contenedor <types>.

Tomemos un fragmento de código del capítulo Ejemplo de WSDL –

```

<wsdl:message name="ZwsPosPedido">
  <wsdl:part name="parameters" element="tns:ZwsPosPedido"/>
</wsdl:message>

```

Aquí, se definen dos elementos de mensaje. El primero representa un mensaje de solicitud **ZwsPosPedido** y el segundo representa un mensaje de respuesta **ZwsPosPedidoResponse**.

Cada uno de estos mensajes contiene un elemento de una sola parte. Para la solicitud, la parte especifica los parámetros de la función; en este caso, se especifica una sola **parameters** parámetro. Para la respuesta, la parte especifica los valores de retorno de la función; En este caso, especificamos un solo valor de retorno de **ZwsPosPedidoResponse**.

### 6.4 WSDL – Elemento <portType>

El elemento <portType> combina múltiples elementos de mensaje para formar una operación completa de ida o vuelta.

Por ejemplo, un <portType> puede combinar una solicitud y un mensaje de respuesta en una sola operación de solicitud / respuesta. Esto se utiliza más comúnmente en los servicios SOAP. Un portType puede definir múltiples





operaciones.

```
<wsdl:portType name="ZWS_CREA_PEDIDO">
  <wsp:Policy>
    <wsp:PolicyReference URI="#IF__ZWS_CREA_PEDIDO"/>
  </wsp:Policy>
  <wsdl:operation name="ZwsPosPedido">
    <wsp:Policy>
      <wsp:PolicyReference URI="#OP__ZwsPosPedido"/>
    </wsp:Policy>
    <wsdl:input message="tns:ZwsPosPedido"/>
    <wsdl:output message="tns:ZwsPosPedidoResponse"/>
  </wsdl:operation>
</wsdl:portType>
```

- El elemento `portType` define una sola operación, llamada `ZWS_CREA_PEDIDO`
- La operación consiste en un solo mensaje de entrada `ZWS_CREA_PEDIDO` y un mensaje de salida `ZwsPosPedidoResponse`.

### 6.5 WSDL – Elemento `<binding>`

El elemento `< binding >` proporciona detalles específicos sobre cómo se transmitirá realmente una operación `portType` a través del cable.

Los enlaces pueden estar disponibles a través de múltiples transportes, incluidos HTTP GET, HTTP POST o SOAP. Los enlaces proporcionan información concreta sobre qué protocolo se está utilizando para transferir las operaciones `portType`.

Los enlaces proporcionan información donde se encuentra el servicio. Para el protocolo SOAP, el enlace es `<soap: binding>`, y el transporte son mensajes SOAP sobre el protocolo HTTP.

Puede especificar varios enlaces para un solo `portType`.

El elemento de enlace tiene dos atributos: `name` y `type` de atributo.

```
<wsdl:binding name="ZWS_CREA_PEDIDO_soap12" type="tns:ZWS_CREA_PEDIDO">
  <wsp:Policy>
    <wsp:PolicyReference URI="#BN__ZWS_CREA_PEDIDO_soap12"/>
  </wsp:Policy>
  <wsoap12:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document"/>
  <wsdl:operation name="ZwsPosPedido">
    <wsoap12:operation soapAction="urn:sap-com:document:sap:soap:functions:mc-
style:ZWS_CREA_PEDIDO:ZwsPosPedidoRequest" style="document"/>
    <wsdl:input>
      <wsoap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <wsoap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
```

El atributo de `name` define el nombre del enlace, y el atributo de `type` apunta al puerto para el enlace, en este caso el puerto " `tns:ZWS_CREA_PEDIDO` ".



## 6.6 WSDL – Elemento <ports>

Un elemento **<port>** define un punto final individual especificando una dirección única para un enlace.

Aquí está la gramática para especificar un puerto -

```
<xsd:import namespace="urn:sap-com:document:sap:rfc:functions"/>
```

El elemento de puerto tiene dos atributos: **name** y **enlace**.

El atributo de **name** proporciona un nombre único entre todos los puertos definidos dentro del documento WSDL adjunto.

El atributo de **enlace** se refiere al enlace mediante las reglas de enlace definidas por WSDL.

Los elementos de extensibilidad de enlace se utilizan para especificar la información de dirección para el puerto.

Un puerto NO DEBE especificar más de una dirección.

Un puerto NO DEBE especificar ninguna información de enlace que no sea información de dirección.

## 6.7 WSDL – Elemento <service>

El elemento **<servicio>** define los puertos admitidos por el servicio web. Para cada uno de los protocolos soportados, hay un elemento de puerto. El elemento de servicio es una colección de puertos.

- a. Los clientes de servicios web pueden aprender lo siguiente del elemento de servicio:
  - a. Dónde acceder al servicio,
  - b. a través de qué puerto acceder al servicio web, y
  - c. Cómo se definen los mensajes de comunicación.
- b. El elemento de servicio incluye un elemento de documentación para proporcionar documentación legible.

```
<wsdl:service name="ZWS_CREA_PEDIDO">
  <wsdl:port name="ZWS_CREA_PEDIDO" binding="tns:ZWS_CREA_PEDIDO">
    <soap:address
location="http://s4qa:8000/sap/bc/srt/rfc/sap/zws_crea_pedido/500/zws_crea_pedido/zws
crea_pedido"/>
  </wsdl:port>
  <wsdl:port name="ZWS_CREA_PEDIDO_soap12" binding="tns:ZWS_CREA_PEDIDO_soap12">
    <wssoap12:address
location="http://s4qa:8000/sap/bc/srt/rfc/sap/zws_crea_pedido/500/zws_crea_pedido/zws
crea_pedido"/>
  </wsdl:port>
</wsdl:service>
```

Los atributos de enlace del elemento de **port** asocian la dirección del servicio con un elemento de enlace definido en el servicio web. En este ejemplo, esto es **ZWS\_CREA\_PEDIDO**.



## 7 Requerimiento funcional 2

Nombre del requerimiento	Consumir el Servicio Web SOAP con RockJS Framework
--------------------------	--

De acuerdo a los tipos de datos detallados en el punto 6.2, se requiere capturar la información en un formulario, validarlo y enviarlo a SAP mediante el WSDL.

La información que viajará debe ser recibida por SAP, luego SAP enviará la respuesta de la validación y proporcionará un front al usuario con alertas para informar sobre el estatus de la transacción.

A continuación se da el proceso para la conversión del Servicio Web SOAP XML a REST JSON. Mediante el consumo del WSDL de SAP, RockJS transformará el SOAP a REST.

### 7.1 Consumo del Servicio Web SOAP

Mediante el lenguaje de programación PHP se creará una clase llamada `Wsdlsap.php`, en esta clase se implementarán las funciones nativas para la conexión con el WSDL y extraer la información necesaria del Servicio Web, mediante la función `SoapClient`.

La clase `SoapClient` proporciona un cliente para servidores » SOAP 1.1, » SOAP 1.2. Puede ser usada en modo WSDL o modo no-WSDL.

### 7.2 Descripción del API WSDL SAP.

Mediante el uso del estándar HTTP requerido para obtener datos o indicar la ejecución de operaciones sobre los datos, se crea un REST API, empleando las operaciones GET, POST, PUT y DELETE equiparables a las operaciones CRUD de base de datos (Crear, leer, actualizar y borrar) el intercambio de datos e realizar mediante JSON (JavaScript Object Notation)

Para este desarrollo, las URL que del API que pueden desarrollar son los siguientes:

- a) GET /veta → recuperará una lista de ventas.
- b) GET /veta /1 → recupera la información de una venta en específico.
- c) POST /venta → Crea una nueva venta
- d) PUT /venta /1 → Actualiza el registro con el ID 1.
- e) DELETE /venta /1 → Elimina el registro con ID 1.

El alcance de este desarrollo tan solo define la acción de crear ventas, sin embargo es posible agregar más funciones en un futuro.



### Ejemplo de estructura de controlador REST

```
public function __construct() {
    header('Content-Type: application/JSON');
    $method = $_SERVER['REQUEST_METHOD'];
    switch ($method) {
        case 'GET': //consulta
            echo 'METODO NO DISPONIBLE';
            break;
        case 'POST': //inserta
            $this->addVent();
            break;
        case 'PUT': //actualiza
            echo 'METODO NO DISPONIBLE';
            break;
        case 'DELETE': //elimina
            echo 'METODO NO DISPONIBLE';
            break;
        default: //metodo NO soportado
            echo 'METODO NO SOPORTADO';
            break;
    }
}
```

### 7.3 Definición de la autenticación SAP

Para configurar la autorización del WSDL, hay que proveer del usuario y contraseña mediante un arreglo, que al instanciar la función SoapClient, que recibirá como parámetro.

```
$SOAP_AUTH = array('login' => 'webservice',
    'password' => 'webservice',
    'encoding' => 'utf-8',
);
```

### 7.4 Definir WDSL

```
$WSDL="
http://201.166.145.80:8000/sap/bc/srt/wSDL/flv_10002A111AD1/bndg_url/sap/bc/srt/rfc/sap/zws_crea_pedido/500/zws_crea_pedido/zws_crea_pedido?sap-client=500";
```

### 7.5 Creación del objeto SoapClient

En la creación del objeto SoapClient, recibe como argumentos de la función los parámetros `$WSDL` y `$SOAP_AUTH`

```
$client = new SoapClient($WSDL, $SOAP_AUTH);
```

### 7.6 Definición de la función para dar de alta una SOLPE

`SoapClient::__getFunctions` — Devuelve una lista de todas las funciones SOAP disponibles. Mediante la siguiente línea se visualiza la función de entrada al sistema SAP.

En este proceso también se aprecia la conversión de los datos XML a JSON para una mejor lectura y procesamiento.

```
echo '<pre>' . json_encode($client->__getTypes(), JSON_PRETTY_PRINT) . '</pre>';
```



La ejecución de esta línea devolverá el nombre de la función que se debe invocar para la creación de Ventas/Pedidos (ZwsPosPedido).

### 7.7 Tipos de dato del ZwsPosPedido

`__getTypes()` Devuelve el nombre y el tipo de las variables de la función ZwsPosPedido.

```
echo '<pre>' . json_encode($client->__getTypes(), JSON_PRETTY_PRINT) . '</pre>';
```

Esta ejecución devolverá la lista de variables y su tipo de dato.

### 7.8 Excepciones y posibles errores

PHP tiene algunos problemas al cargar un archivo WSDL de un servicio web de SAP y provoca el siguiente error:

**Error grave: excepción de SoapFault no detectada: [WSDL] SOAP-ERROR: Análisis WSDL: desconocida requiere la extensión WSDL**

El mensaje "Se requiere extensión WSDL desconocida" significa que WSDL usa alguna extensión y especifica que el cliente debe entenderlo para poder usar el servicio. Aparentemente, Los lenguajes de programación, incluso PHP SoapClient no entienden la extensión, lo que provoca este error.

La solución temporal que utilizamos para evitar este problema es deshabilitar el uso de `UsingPolicy` del `sap wsd1`. Al reemplazar la palabra "`true`" por "`false`" en la url del `wsdl`, tenemos un segundo `wsdl` que no usa las políticas del Servicio Web.

```
<wsp:UsingPolicy wsdl:required="true"/>
```

Por

```
<wsp:UsingPolicy wsdl:required="False"/>
```

También es posible que deba modificar su archivo hosts el cual le permite invalidar el sistema de nombres de dominio (DNS) para un dominio en una máquina específica. Este proceso es útil cuando desea realizar las siguientes tareas:

- Probar su sitio sin el enlace de pruebas antes de publicarlo con SSL.
- Verificar que un sitio alias funcione antes de los cambios al DNS.
- Trabajar en otras tareas relacionadas con el DNS.

Modificar su archivo hosts provocará que su máquina local vea directamente hacia la dirección protocolo de internet (IP) que especifique.

Modificar el archivo hosts exige añadirle dos entradas. Cada entrada contiene la dirección IP a la que quiere que resuelva el sitio y una versión de la dirección de internet.

El archivo Host se encuentra en la dirección: C:\Windows\System32\drivers\etc y deberá agregar estas líneas y salvar los cambios.

```
201.166.145.80
```

```
s4qa
```

**7.9 Datos Maestros**

Consumo del servicio desde SAP: Datos Maestros de Existencias para La Fábrica VM01 almacén VM1.

MATNR	WERK	LGOR	CLABS	VORMG
ABA10127 G	VM01	VM1	2.000	1.000
ABEA1001 G	VM01	VM1	7.000	1.000
ACCF1008 G	VM01	VM1	1.000	1.000
CPCD13000 2	VM01	VM1	20.000	20.000
EFE11258 G	VM01	VM1	1.000	1.000
EFE11281 G	VM01	VM1	1.000	1.000
EFS11089 P	VM01	VM1	4.000	1.000
HMA41173 G	VM01	VM1	1.000	1.000
HMA43015 G	VM01	VM1	16.000	1.000
HMA43022 G	VM01	VM1	8.000	1.000
HMA45020 G	VM01	VM1	6.000	1.000
HMA45021 G	VM01	VM1	6.000	1.000
HMA45097 G	VM01	VM1	10.000	1.000
HMA45099 G	VM01	VM1	5.000	1.000
MPEPP116 G	VM01	VM1	170.000	1.000
MPEPP331 G	VM01	VM1	120.000	1.000
MPEPP426 G	VM01	VM1	100.000	1.000
MPZA0020 G	VM01	VM1	4,000.000	1,000.000
MPZA0025 G	VM01	VM1	14,000.000	1.000
MTDMC0489 G	VM01	VM1	3.000	0.520
MTDMC0541 G	VM01	VM1	2.000	0.540
MTDMC0584 G	VM01	VM1	4.000	0.560
MTDMC0729 G	VM01	VM1	10.000	0.940

**7.10 Ejemplo de consumo con PHP**

Evidencia de Salida:

URL: <http://localhost/ConsSAP/prdatosmaestros.php>

```

[item] => Array
(
    [0] => stdClass Object
        (
            [Matnr] => ABA10127  G
            [Werks] => VM01
            [Lgort] => VM1
            [Clabs] => 2.0
            [Vormg] => 1.0
        )

    [1] => stdClass Object
        (
            [Matnr] => ABEA1001  G
            [Werks] => VM01
            [Lgort] => VM1
            [Clabs] => 7.0
            [Vormg] => 1.0
        )

    [2] => stdClass Object
        (
            [Matnr] => ACCF1008  G
            [Werks] => VM01
            [Lgort] => VM1
            [Clabs] => 1.0
            [Vormg] => 1.0
        )

    [3] => stdClass Object
        (
            [Matnr] => CPCD13000 2
            [Werks] => VM01
            [Lgort] => VM1
            [Clabs] => 20.0
            [Vormg] => 20.0
        )

```



### Opciones de búsqueda del Servicio Web de datos maestros.

Opción	Clave	Descripción
<b>Materiales</b>	MA	Datos maestros de materiales
<b>Precios</b>	PS	Datos maestros de Precios
<b>Precios Material</b>	PM	Datos maestros de Precios por material
<b>Clientes</b>	BP	Datos Maestros por Cliente
<b>¿</b>	PB	
<b>Existencias</b>	EX	Datos maestros de Existencias

De manera tal que en el cuerpo del arreglo en la sección de "Opcion" debe llevar cualquiera de estas claves.  
Ejemplo

```
'Opcion' => "EX", //Existencias
'Opcion' => "MA", //Materiales
Etc.
```

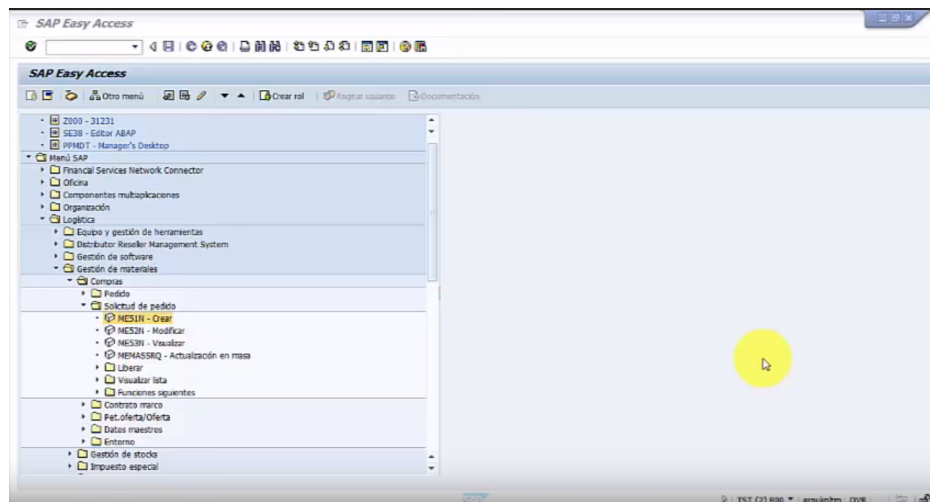
### Análisis del modelo de SOLPE

Análisis del modelo de SOLPE (Solicitud de pedidos) HANA, Distribución y diseño funcional.

La SOLPE es un paso antes del PEDIDO ya que cargaría los datos necesarios para hacer el pedido

Iniciar con la transacción ME51N

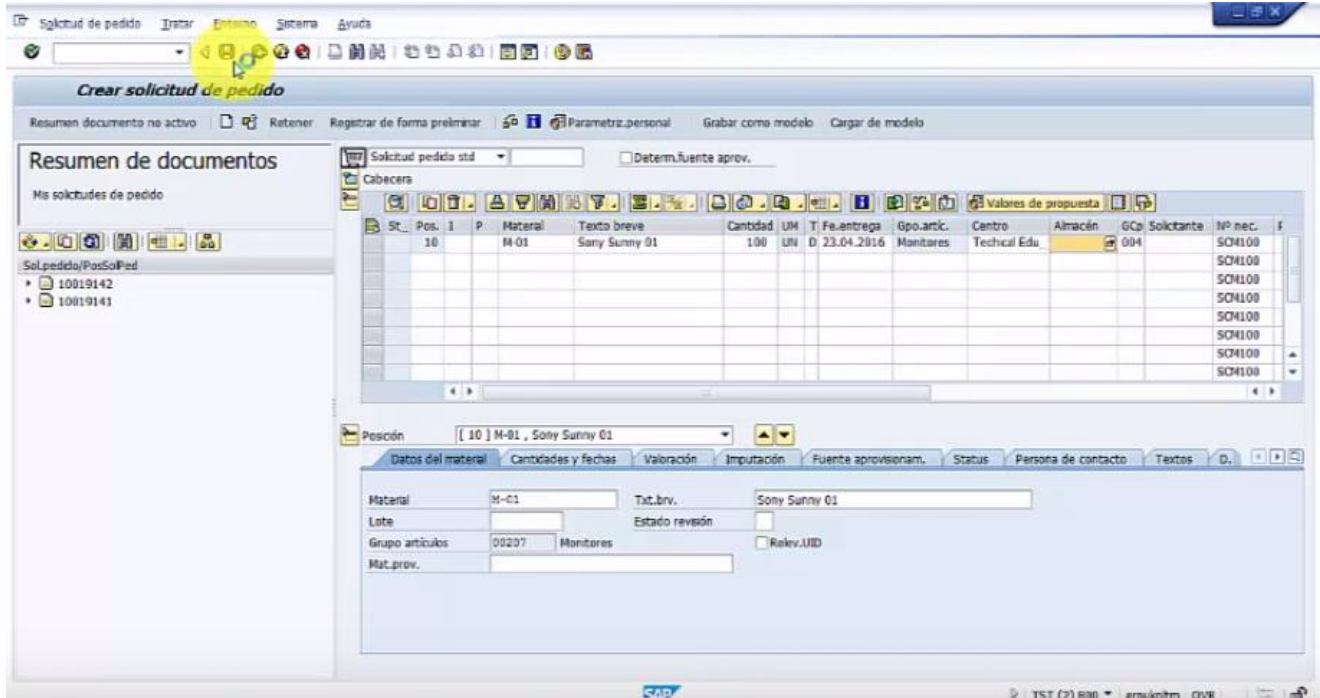
*Menú SAP > Lógica > Gestión de materiales > Compras > Solicitud de pedido > ME51N - Crear*





Se agregan las partidas linea por linea en la pantalla.

- Define la clave del material
- Cantidad
- Define el Centro

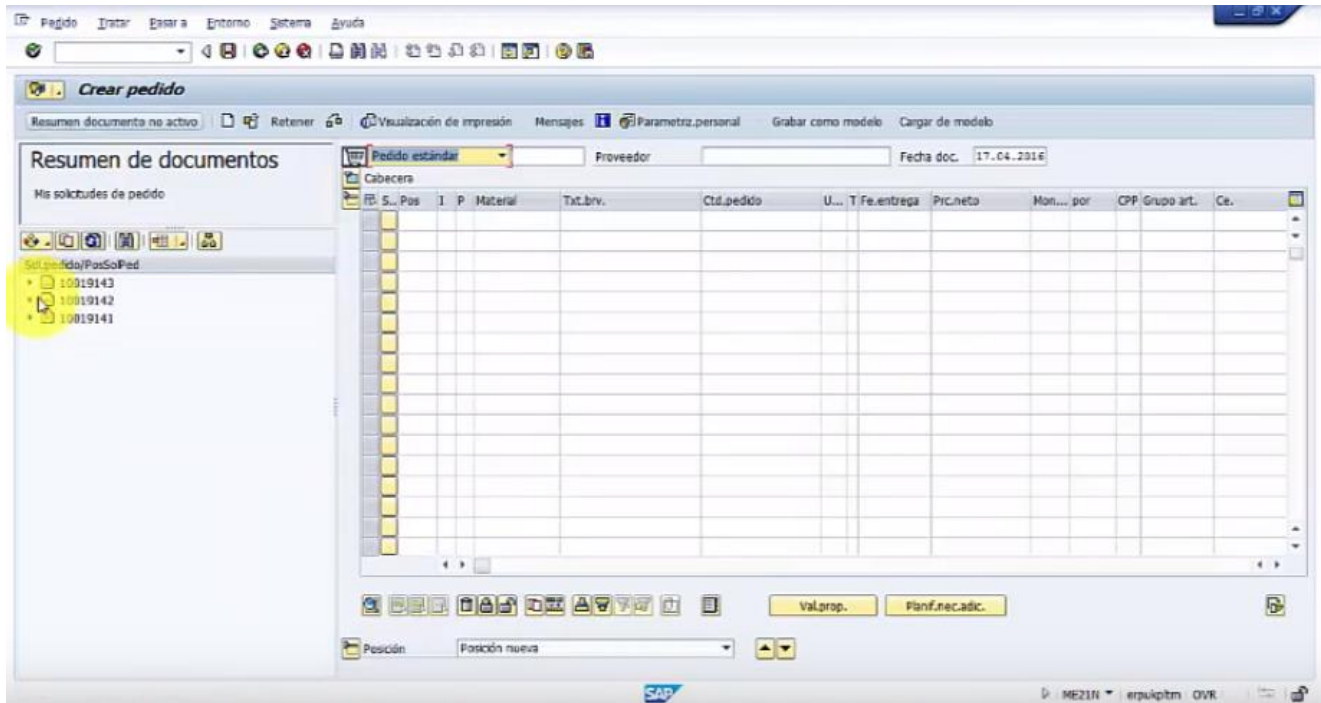


The screenshot displays the SAP 'Crear solicitud de pedido' (Create Purchase Order) screen. The interface includes a menu bar at the top with options like 'Solicitud de pedido', 'Tratar', 'Entorno', 'Sistema', and 'Ayuda'. Below the menu, there's a toolbar with various icons. The main area is divided into several sections:

- Resumen de documentos**: A sidebar on the left showing a list of documents.
- Cabecera**: A header section with fields for 'Solicitud pedida std', 'Determ.fuente aprov.', and 'Valores de propuesta'.
- Tabla de líneas**: A table with columns for 'St.', 'Pos.', 'I', 'P', 'Material', 'Texto breve', 'Cantidad', 'UM', 'T. Fa.entrega', 'Gpo.artic.', 'Centro', 'Almacén', 'GCo', 'Solicitante', 'NP nec.', and 'f'. The first line item is selected, showing details for material 'M-01', quantity '100', and center 'Tecnical Edu'.
- Datos del material**: A section at the bottom with fields for 'Material', 'Lote', 'Grupo articulos', and 'Mat. prov.'. The 'Material' field is set to 'M-01', 'Lote' to '00207', and 'Grupo articulos' to 'Monitores'.

Se guardan los cambios y luego aparece el SOLPE con la leyenda: *Creada la solicitud de pedido bajo el numero 00000000*

Ahora se navega a la opción ME21N para hacer pedidos y usar la SOLPE creada como referencia.



Seleccionar y arrastrar en número de SOLPE hasta el icono de Carrito de Compra.

- Proporcionar número de proveedor

### Ejecución de la solicitud de pedido con PHP y SAP (SOLPE)

Código fuente del programa sap2pe.php transacción exitosa

<?php

```
$SOAP_AUTH = array(
    'login' => 'webservice', //define la autenticacion
    'password' => 'webservice',
    'encoding' => 'utf-8',
    "sap-client" => "500",
);

$Cabecero = array(
    'item' => array(
        'Mandt' => '500',
        'Ticket' => 100014,
        'Factura' => 'FTT49',
        'DocType' => 'ZVCA',
        'FechaAlta' => '',
        'SalesOrg' => 1000,
        'DistrChan' => 15,
        'Division' => '0A',
        'SalesOff' => 'VM01',
        'PurchNoC' => 'GRUPO 1',
        'Pmnttrms' => 'NCON',
        'PartnRole' => '',
        'PartnNumb' => '112751',
        'Country' => '',
    )
);
```



```

        'Langu' => '',
        'Name' => 'ARANZA BARRON FRANCISCO JAVIE',
        'City' => 'MX',
        'Street' => 'AGRICOLA ORIENTAL',
        'PostlCode' => '08500',
        'Status' => '',
        'Registros' => '1',
        'FechaCurrent' => '2018-01-18',
        'PedidoSap' => '1',
        'EntregaSap' => '',
        'FacturaSap' => '',
        'Tipoerror' => ''
    ),
);

$Detalle = array(
    'item' => array(
        'Mandt' => '500',
        'Ticket' => 100014,
        'Factura' => 'FTT49',
        'DocType' => 'ZVCA',
        'ItmNumber' => '000001',
        'Material' => 'HMA45099 G',
        'Plant' => 'VM01',
        'StoreLoc' => 'VM1',
        'ReqQty' => '1',
        'SalesUnit' => 'M',
        'Kbetr' => '1',
        'Lote' => '1'
    ),
);

try {
    $wsdl_url =
'http://s4qa:8000/sap/bc/srt/wsdl/flv_10002A101AD1/bndg_url/sap/bc/srt/rfc/sap/zws_cre
a_pedido/500/zws_crea_pedido/zws_crea_pedido?sap-client=500';
    $client = new SOAPClient($wsdl_url, $SOAP_AUTH);
    $params = array(
        'EtError' => array(
            'item' => array(),
        ),
        'EtPosCabecero' => $Cabecero,
        'EtPosDetalle' => $Detalle,
        'ItPosCabecero' => '',
        'ItPosDetalle' => ''
    );
    echo "<pre>";
    print_r($params);
    //print_r($client->__getTypes());
    //echo '<pre>' . json_encode($client->__getFunctions(), JSON_PRETTY_PRINT) .
'</pre>';
    echo "</pre>";
    $return = $client->ZwsPosPedido($params);
    echo "<pre>";
    print_r($return);
    echo "</pre>";
} catch (Exception $e) {

```



```

    echo "Exception occurred: " . $e;
}

```

### Mensaje de respuesta

```

[EtError] => stdClass Object
(
    [item] => stdClass Object
    (
        [Znoerror] => 00
        [Ztipo] => A
        [Zerror] => APROBADO!! SOLICITUD CREADO CON EXITO
    )
)

```

**ZPOS\_PEDIDO: Visualización de las entradas existentes**

Tabla a examinar: ZPOS\_PEDIDO    Tabla para Pedido (Cabecera) POS

Ctd.aciertos: 1

TmptoEjec: 0    Ctd.máxima aciertos: 500

| TICKET | FACTURA | CVT  | Fecha S | OrgVt | CDIs | Se | OfVta | Ref.cte. | CPag | Func. | Cliente | Ctry.ind. | Idioma | Nombre 1                      | Población | Calle             | CP    | TRAMER No. | Registros | Fecha alta | Pedido Entrega | Pedido Tipo |
|--------|---------|------|---------|-------|------|----|-------|----------|------|-------|---------|-----------|--------|-------------------------------|-----------|-------------------|-------|------------|-----------|------------|----------------|-------------|
| 100014 | FTT49   | ZVCA |         | 1000  | 15   | 0A | VM01  | GRUPO 1  | NCON |       | 112751  |           |        | ARANZA BARRON FRANCISCO JAVIE | MX        | AGRICOLA ORIENTAL | 08500 | PENDIENTE  | 1         | 03.01.2019 | 1              |             |

*Ilustración 6: Transacción exitosa SAP*

**Descripción y definición de campos**

| Cabecera                                   |                     |           |
|--|---------------------|-----------|
| Campo                                      | Definición          | Requerido |
| 'Mandt' => '500',                          | Cliente             | Si        |
| 'Ticket' => 100014,                        | Ticket              | Si        |
| 'Factura' => 'FTT49',                      | Factura             | Si        |
| 'DocType' => 'ZVCA',                       | Tipo de documento   | Si        |
| 'FechaAlta' => '',                         | Fecha de alta       | No        |
| 'SalesOrg' => 1000,                        | Organización de Sal | Si        |
| 'DistrChan' => 15,                         | Cadena distribución | Si        |
| 'Division' => '0A',                        | División            | Si        |
| 'SalesOff' => 'VM01',                      | Ventas salientes    | Si        |
| 'PurchNoC' => 'GRUPO 1',                   | -                   | No        |
| 'Pmnttrms' => 'NCON',                      | -                   | si        |
| 'PartnRole' => '',                         | Parte de rol        | No        |
| 'PartnNumb' => '112751',                   | Numero de parte     | No        |
| 'Country' => '',                           | País                | Si        |
| 'Langu' => '',                             | Lenguaje            | No        |
| 'Name' => 'ARANZA BARRON FRANCISCO JAVIE', | Nombre              | Si        |
| 'City' => 'MX',                            | Ciudad              | Si        |
| 'Street' => 'AGRICOLA ORIENTAL',           | Calle               | No        |
| 'PostlCode' => '08500',                    | Código postal       | No        |
| 'Status' => '',                            | Estatus             | No        |
| 'Registros' => '1',                        | Registros           | No        |
| 'FechaCurrent' => '2018-01-18',            | -                   | No        |
| 'PedidoSap' => '1',                        | -                   | No        |
| 'EntregaSap' => '',                        | -                   | No        |
| 'FacturaSap' => '',                        | -                   | No        |
| 'Tipoerror' => '',                         | -                   | No        |

| Detalle                     |                   |           |
|-----------------------------|-------------------|-----------|
| Campo                       | Definición        | Requerido |
| 'Mandt' => '500',           | Cliente           | Si        |
| 'Ticket' => 100014,         | Ticket            | Si        |
| 'Factura' => 'FTT49',       | Factura           | Si        |
| 'DocType' => 'ZVCA',        | Tipo de documento | Si        |
| 'ItmNumber' => '000001',    | Numero de ítem    | No        |
| 'Material' => 'HMA45099 G', | Id de material    | Si        |
| 'Plant' => 'VM01',          | Sucursal          | Si        |
| 'StoreLoc' => 'VM1',        | Almacén           | Si        |
| 'ReqQty' => '1',            | Cantidad          | Si        |
| 'SalesUnit' => 'M',         | Unidad            | No        |
| 'Kbetr' => '1',             | -                 | si        |
| 'Lote' => '1',              | Lote              | si        |



## Procedencia y definición de campos

| Cabecera   |   |   |
|--|---|---|
| Campo  | Procedencia   | Definición  |
| 'Mandt' => '500',<br>'Ticket' => 100014,<br>'Factura' => 'FTT49',<br>'DocType' => 'ZVCA',<br>'FechaAlta' => '',<br>'SalesOrg' => 1000,<br>'DistrChan' => 15,<br>'Division' => '0A',<br>'SalesOff' => 'VM01',<br>'PurchNoC' => 'GRUPO 1',<br>'Pmnttrms' => 'NCON',<br>'PartnRole' => '',<br>'PartnNumb' => '112751',<br>'Country' => '',<br>'Langu' => '',<br>'Name' => 'ARANZA BARRON FRANCISCO JAVIE',<br>'City' => 'MX',<br>'Street' => 'AGRICOLA ORIENTAL',<br>'PostlCode' => '08500',<br>'Status' => '',<br>'Registros' => '1',<br>'FechaCurrent' => '2018-01-18',<br>'PedidoSap' => '1',<br>'EntregaSap' => '',<br>'FacturaSap' => '',<br>'Tipoerror' => '' | Valor estático<br>Consecutivo<br>Valor estático<br>Opcional<br>Valor estático<br>Valor estático<br>Valor estático<br>Valor estático<br>Valor estático<br>Valor estático<br>Valor estático<br>Valor estático<br>Opcional<br>opcional<br><br>Maestro de clientes<br>Maestro de clientes<br>Maestro de clientes<br>Maestro de clientes<br>Opcional<br>Opcional<br>Actual<br>Opcional<br>Opcional<br>Opcional | Char-3<br>Int-6<br>Char-4<br>Date dd-mm-yy<br>Int-4<br>Int-2<br>Char-2<br>Char-4<br>Char-7<br>Char-4<br>-<br>Char-6<br><br>char35<br>char35<br>char35<br>char5<br>char10<br>-<br>char10<br>char10<br>char10 |

| Detalle   |   |  |
|---|---|--|
| Campo   | Definición  | Requerido  |
| 'Mandt' => '500',<br>'Ticket' => 100014,<br>'Factura' => 'FTT49',<br>'DocType' => 'ZVCA',<br>'ItmNumber' => '000001',<br>'Material' => 'HMA45099 G',<br>'Plant' => 'VM01',<br>'StoreLoc' => 'VM1',<br>'ReqQty' => '1',<br>'SalesUnit' => 'M',<br>'Kbetr' => '1',<br>'Lote' => '1' | Valor estático<br>Consecutivo<br>Valor estático<br>Valor estático<br>Valor estático<br>Maestro de existencias (Matnr)<br>Valor estático<br>Valor estático<br>Cantidad<br>Unidad<br>Opcional<br>Opcional | clnt3<br>char20<br>char20<br>char4<br>int6<br>char40<br>Si<br>Si<br>Si<br>Si<br>No<br>si<br>si |