



**JOSÉ MANUEL RIVAS MERCURY FILHO**

**RELATÓRIO DE ESTÁGIO NA EMPRESA SOTHIS  
ENTERPRISE RESOURCE PLANNING LTDA**

**EMIÇÃO DE DOCUMENTOS ADUANEIROS UTILIZANDO A LINGUAGEM  
DE PROGRAMAÇÃO ABAP E SISTEMA ERP**

**LAVRAS – MG  
2021**

**JOSÉ MANUEL RIVAS MERCURY FILHO**

**RELATÓRIO DE ESTÁGIO NA EMPRESA SOTHIS ENTERPRISE RESOURCE  
PLANNING LTDA  
EMISSÃO DE DOCUMENTOS ADUANEIROS UTILIZANDO A LINGUAGEM DE  
PROGRAMAÇÃO ABAP E SISTEMA ERP**

Relatório de Estágio Supervisionado  
apresentado à Universidade Federal de Lavras  
como parte das exigências do curso de  
Sistemas de Informação, para obtenção do  
título de Bacharel.

Profa. Dra. Renata Teles Moreira  
Orientadora

**LAVRAS - MG  
2021**

**JOSÉ MANUEL RIVAS MERCURY FILHO**

**RELATÓRIO DE ESTÁGIO NA EMPRESA SOTHIS ENTERPRISE RESOURCE  
PLANNING LTDA  
EMISSÃO DE DOCUMENTOS ADUANEIROS UTILIZANDO A LINGUAGEM DE  
PROGRAMAÇÃO ABAP E SISTEMA ERP**

Relatório de Estágio Supervisionado  
apresentado à Universidade Federal de Lavras  
como parte das exigências do curso de  
Sistemas de Informação, para obtenção do  
título de Bacharel em Sistemas de Informação.

Aprovado em 18 de junho de 2021

Dra. Renata Teles Moreira  
Dr. Maurício Ronny de Almeida Souza  
Stella Marques

DCC/UFLA  
DCC/UFLA  
CRIA Tecnologia e Inovação

Profa. Dra. Renata Teles Moreira  
Orientadora

**LAVRAS - MG  
2021**

## **AGRADECIMENTOS**

Em memória ao meu pai José Manuel Rivas Mercury

Minha família: Priscila Marlys Sá Rivas e Raimunda Castro Sá Rivas

Aos membros da banca: Maurício Ronny de Almeida Souza,

Stella Azevedo Marques e Renata Teles Moreira

Todos do DCC, DRCA, DRI e PRG UFLA

MIC Customs Solutions / Sothis Enterprise Resource Planning

IAESTE Áustria, IAESTE Brasil e IAESTE Espanha

## LISTA DE FIGURAS

Figura 1 – Definição de contexto.	19
Figura 2 – Recuperação de Dados.	19
Figura 3 – Módulo de funções.	20
Figura 4 – Funções e Estruturas Complexas.	20
Figura 5 – Parâmetros de Entrada.	21
Figura 6 – Recuperação e Impressão de Dados - Interface.	21
Figura 7 – Parâmetros do Formulário.	22
Figura 8 – Importação de Dados de Rodapé e Cabeçalho.	23
Figura 9 – Inserção de Sub Formulários e Tabelas.	23
Figura 10 – Customização de Subformulários e Tabelas.	24
Figura 11 – Inserção do logotipo da empresa.	24
Figura 12 – Inserção do logotipo da empresa.	25
Figura 13 – Identificação do nome do logotipo.	25
Figura 14 – Seleção de Logotipo a ser inicializado no formulário.	26
Figura 15 – Transporte do formulário para o pacote de desenvolvimento do cliente.	27
Figura 16 – Transporte do formulário para o pacote de desenvolvimento do cliente.	27

## SUMÁRIO

1.	INTRODUÇÃO	8
2.	SOBRE A ORGANIZAÇÃO	10
2.1	Sothis Enterprise Resource Planning	10
2.2	Processo Organizacional	10
2.3	Produto	11
3.	TECNOLOGIAS E FERRAMENTAS	13
3.1	Framework Scrum	13
3.1.1	Eventos do Scrum	13
3.1.2	Papéis do Scrum	14
3.2	Linguagem de Programação ABAP	14
3.2.2	Herança	16
3.2.3	Interface Autônoma	16
3.2.4	Objetos Compartilhados	16
3.2.5	Variáveis de referência de objeto e polimorfismo	16
4.	ATIVIDADES DESENVOLVIDAS	17
4.1	Treinamento	17
4.2	Utilizando SAP e AdobeForms	17
4.3	Outras atividades realizadas	29
5.	CONSIDERAÇÕES FINAIS	30
	REFERÊNCIAS	31

## 1. INTRODUÇÃO

SAP é uma empresa que existe desde 1972, sendo a principal fornecedora de software de planejamento de recursos empresariais do mundo. A empresa conseguiu usar os avanços da tecnologia para melhorar seu software, por meio de inovações tecnológicas e, dessa forma, a SAP conseguiu dominar o mercado. À medida que a empresa continua a se ajustar às demandas do mercado atual, ela continua a crescer. O termo SAP vem do alemão “*Systeme, Anwendungen und Produkte in der Datenverarbeitung*” (Sistemas, Aplicações e Produtos em processamento de dados) e a multinacional possui a sua sede na cidade de *Walldorf* na Alemanha (Andera e Derringer, 1998). SAP é um Sistema de Gestão Empresarial ERP<sup>1</sup>, utilizado por gigantes de diversos segmentos de negócios como Volkswagen, Telefônica, Rhodia, Siemens, Capgemini, Avantis, entre diversas outras. No Brasil está presente em empresas como Petrobras, Vale, Arcelormittal e Eletronorte.

Até junho de 2021 a versão mais atualizada do sistema é o SAP S/4 HANA 2020. A versão utilizada para desenvolvimento neste trabalho foi o sistema SAP ECC 6.0, sobre uma nova plataforma da *Netweaver*. A empresa *Sothis* utiliza vários programas como soluções, como o SAP EWM para gestão de armazéns de forma ágil, eficiente e em tempo real. Desde o recebimento e localização da mercadoria até o despacho, passando pelos registros de suprimentos e produção ou movimentos internos, a solução SAP EWM permite executar e monitorar todas as tarefas do armazém.

Neste contexto, o discente do curso de Sistemas de Informação da Universidade Federal de Lavras realizou o estágio na empresa *Sothis Enterprise Resource Planning*. Onde aplicou diversas áreas de disciplinas cursadas como, Gerência de Projetos, Gestão de processos de software e Engenharia de requisitos de software. No estágio trabalhou com a solução SAP EWM, Gestão de transporte e programação em ambiente SAP ECC.

Dessa forma, este relatório tem como objetivo descrever as atividades desenvolvidas na empresa *Sothis Enterprise Resource Planning*, no período de outubro de 2018 a maio de 2019. Neste período, o estagiário atuou como *Desenvolvedor ABAP* no projeto denominado *SAP EWM*. O objetivo geral para a realização deste trabalho foi obter conhecimento sobre consultoria tecnológica e gestão/administração logística. Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- Realizar a revisão da literatura da empresa;
- Realizar correções de documentos funcionais;

---

<sup>1</sup> <https://www.sap.com>

- Desenvolver o trabalho em equipe.
- Desenvolver expedição de formulários;

Além deste capítulo introdutório, este relatório de estágio está organizado como segue. Na seção 2 apresenta a empresa e seu processo organizacional. Na parte 3 explica as tecnologias e ferramentas utilizadas. A parte 4 detalha as atividades desenvolvidas e treinamento dado. Por fim, o capítulo 5 apresenta as considerações finais acerca das atividades desempenhadas durante o estágio.

## 2. SOBRE A ORGANIZAÇÃO

Este Capítulo descreve sobre a empresa *Sothis Enterprise Resource Planning s.l.*, onde o autor deste documento estagiou no período de outubro de 2018 a maio de 2019. Além disso, é detalhado neste capítulo o processo organizacional da empresa.

### 2.1 Sothis Enterprise Resource Planning

*Sothis Enterprise Resource Planning*<sup>2</sup> é uma empresa de consultoria SAP, onde sua missão é fornecer soluções que melhor se adequam aos clientes, integrando as tecnologias da Informação, Gestão Industrial e Gestão. A empresa desenvolveu um modelo de gestão (baseado no Modelo de Qualidade Total) com 5 componentes: Cliente, Pessoas, Fornecedor, Sociedade e Capital.

A Sothis fornece serviços de suporte a software SAP, inovação tecnológica, gestão de inventários e gestão de materiais. A empresa se especializou na entrega de serviços de software em diversas áreas como química, relação com o cliente e logística. A *Sothis* possui diversos escritórios distribuídos pela Espanha e Portugal (Porto), em algumas cidades como: Madrid, Valência, Sevilha, Paterna, Aranda de Duero e Barcelona.

### 2.2. Processo Organizacional

A empresa tem como missão o compartilhamento de informações e ensinamento entre os membros mais experientes da empresa para com os seus novos funcionários, além de princípios de transparência, com enfoque em metas e fechamento de projetos.

O processo utilizado pela empresa para a entrega de serviços é baseado em *brainstorming*. As reuniões da equipe acontecem semanalmente ou quinzenalmente, dependendo das demandas do projeto. Durante as reuniões são definidas as funções de cada membro da equipe, mobilidade do time, planejamento de transporte, entre outros. As reuniões acontecem entre a equipe e as partes envolvidas no projeto para realizar a entrega do serviço ou software.

Para as entregas dos serviços, os pedidos são armazenados em pacotes diferenciados por módulos, como por exemplo: Planejamento de Manutenção, Gestão de Materiais, Planejamento de Produção, Vendas e Distribuição, Contabilidade Financeira, entre outros. Os programadores mais experientes verificam os módulos e seus desempenhos. Caso ocorra algum erro nos sistemas, os consultores funcionais são responsáveis pelas verificações de

---

<sup>2</sup> <https://www.sothis.tech/>

parâmetros inseridos, corrigi-los e, em casos especiais, pedir apoio aos estagiários ou desenvolvedores.

### **2.3 Produto**

O sistema SAP R/3 imprime os documentos parametrizados pelos consultores tecnológicos, com as instruções para os demais colaboradores da fábrica e todos os envolvidos em projetos envolvendo armazém e estoque de materiais com ou sem produtos. Estes documentos são distribuídos aos funcionários que devem delegar os mesmos a fiscais e regularizadores para controle e inspeção aduaneira. O Sistema R/3 é a base das operações e a partir deles se desmembraram as áreas de Finanças, Recursos Humanos e Logística.

O principal benefício do SAP R/3 é que ele oferece suporte à implementação porque é fácil começar com uma estrutura de processo com parâmetros já estabelecidos. SAP R/3 é um sistema servidor/cliente que separa as tarefas do usuário da lógica. Os aplicativos são responsáveis por gerenciar a Cadeia de Suprimentos a partir do Fornecimento de material até a entrega do produto e faturamento do cliente.

Existem também outros componentes especiais do SAP R/3 que interagem com o módulo padrão e são orientados para certos tipos de indústrias. Dentro da empresa, existiam vários projetos relacionados com diversas empresas como: JJ Forwarded e Mercadona. Os desenvolvedores com mais experiência do time se locomoviam até a região de Múrcia, para a implantação de vários módulos SAP como SD (Logística), FI (Finanças), MM (Gestão de Materiais), PM (Manutenção de Planta), PP (Gestão de Produção), HR (Recursos Humanos) e TM (Gestão de Transporte).

O desenvolvimento de formulários específicos, tanto para a gestão de produtos/materiais, expedição de pedidos e gestão de saída, são dados estratégicos para orquestrar a cadeia de suprimentos, já que devido o tipo de classe dos documentos, são executadas diversas operações. Todos estes processos são de grande importância para ambas as empresas, já que lidam diretamente com a gestão da cadeia de suprimentos.

Uma das principais atividades desenvolvidas durante o estágio foi a programação do documento de entrega de saída, onde estão disponíveis as datas estipuladas para o transporte dos materiais. A entrega é feita no posto de expedição, que pode variar, dependendo do produto escolhido, onde o posto de expedição é programado automaticamente durante a gestão do pedido e na criação do formulário.

Outra atividade desenvolvida foi o documento de expedição do pedido, que são construídos através dos documentos funcionais desenvolvidos pelos consultores, logo após a

Fentrega de saída. Ao criar uma entrega de saída são iniciadas as atividades de expedição, como *picking* ou *programação do transporte*. *Picking* em logística, consiste no processo de recolhida e combinação de cargas não unitárias para completar a preparação de pedidos.

As tarefas de transporte são uma divisão lógica de áreas geográficas para as quais você pode executar tarefas de logística. Não há limites para as áreas geográficas que você pode adicionar a uma zona de transporte. Durante o processo de venda, o sistema combina os dados da zona de transporte e os itinerários de transporte para determinar a duração do envio e os locais de origem. Esta informação é baseada na verificação de disponibilidade.

Os dados nestes documentos são de grande importância, já que utilizam o número do cliente e dos centros para identificar de forma mais simples o receptor dos materiais. Caso alguma alteração no sistema seja feita, as modificações são atualizadas em tempo real. Os estudantes do mestrado SAP (que é uma formação oferecida pela Escola de Empresários - EDEM com enfoque em venda e varejo) e consultores da empresa faziam a parametrização de documentos relevantes para a gestão logística, isto era feito através da definição de classes (de material de embalagem, de documento de vendas, de entrega, entre outros).

### 3. TECNOLOGIAS E FERRAMENTAS

Este capítulo descreve as tecnologias utilizadas e as etapas necessárias para o desenvolvimento de documentos essenciais para a gestão da cadeia de suprimentos.

#### 3.1 Framework Scrum

De acordo com Schwaber e Sutherland (2017), Scrum é um *framework* para desenvolver, entregar e manter produtos complexos, não sendo um processo, técnica ou um método definitivo. Esta ferramenta deixa de forma clara a eficácia relativa de suas práticas de gerenciamento de produto e técnicas de trabalho, de modo que você possa continuamente melhorar o produto, o time e o ambiente de trabalho. O framework consiste em times associados a papéis, eventos, artefatos e regras. Cada componente dentro da ferramenta serve a um propósito específico e é essencial para o uso e sucesso do Scrum.

Para o Scrum o conhecimento vem da experiência e da tomada de decisões baseadas no que é conhecido, emprega uma abordagem iterativa e incremental para aperfeiçoar a previsibilidade e o controle de riscos. Três pilares apoiam a implementação de controle de processo empírico: transparência, inspeção e adaptação.

- **Transparência:** Aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados. Por exemplo: Uma linguagem comum referindo-se ao processo deve ser compartilhada por todos os participantes.
- **Inspeção:** Os usuários Scrum devem, frequentemente, inspecionar os artefatos Scrum e o progresso em direção ao objetivo da Sprint para detectar variações indesejadas.
- **Adaptação:** Se um inspetor determina que um ou mais aspectos de um processo desviou para fora dos limites aceitáveis, e que o resultado do produto será inaceitável, o processo ou o material sendo produzido deve ser ajustado.<sup>2</sup>

##### 3.1.1 Eventos do Scrum

Os eventos no scrum são *time-boxed*, de tal modo que todo evento tem uma duração máxima. Uma vez que a Sprint começa, sua duração é fixada e não pode ser reduzida ou aumentada. Os eventos restantes podem terminar sempre que o propósito do evento é alcançado, garantindo que uma quantidade adequada de tempo seja gasta, não permitindo desperdícios no processo. Além da Sprint, que é um *container* para outros eventos, cada

evento no Scrum é uma oportunidade de inspecionar e adaptar. Estes eventos são projetados para permitir uma transparência e inspeção criteriosa (Schwaber e Sutherland, 2017).

O Scrum prescreve quatro eventos formais para inspeção e adaptação que são: planejamento da Sprint, reunião diária, revisão da Sprint e retrospectiva da Sprint.

- **Planejamento da Sprint:** O Planejamento da Sprint é um *time-boxed* com algumas horas de duração, cada sprint pode durar em média 1 mês; A meta da Sprint é o objetivo que será satisfeito dentro da implementação do Backlog do Produto, e que fornece a orientação para o Time de Desenvolvimento sobre o porquê dele estar construindo o incremento.
- **Reunião Diária:** É um evento *time-boxed* de 15 minutos para o time de desenvolvimento, esta reunião é realizada todos os dias. O time de desenvolvimento planeja o trabalho para as próximas 24 horas, no qual, otimiza a colaboração e desempenho do time.
- **Revisão da Sprint:** A revisão da sprint é realizada no final da Sprint para inspecionar o incremento e adaptar o Backlog do Produto. O time e as partes interessadas colaboram sobre o que foi feito e colaboram com o que pode ser feito para otimizar valor.
- **Retrospectiva da Sprint:** é uma oportunidade para o time scrum inspecionar a si próprio e criar um plano para melhorias a serem aplicadas na próxima sprint. A retrospectiva ocorre depois da Revisão da Sprint e antes do planejamento da próxima Sprint.

### 3.1.2 Papéis do Scrum

A colaboração entre profissionais qualificados é essencial para problemas complexos. As soluções criativas e pensar fora da caixa são mais facilmente descobertas quando se reúnem as perspectivas e habilidades de profissionais qualificados (Verwijs et al, 2020). O *Scrum* é utilizado para organizar e orientar a equipe de desenvolvimento em torno aos diferentes processos envolvidos em rumo ao recurso entregável.

### 3.2 Linguagem de Programação ABAP

A tradução de ABAP, em sua forma literal, tem como significado processador genérico para preparação de relatórios, sendo mais tarde o termo ABAP rebatizado em inglês para *Advanced Business Application Programming*. ABAP é uma linguagem de programação de quarta geração (4GL) criada em 1980 pela SAP, onde nos seus primórdios era apenas uma

linguagem para geração de relatórios do SAP R/2. Esta é a versão antecessora do sistema SAP R/3 e é ainda a tecnologia mais usada para construção de programas acoplados ao SAP, mesmo sendo possível o desenvolvimento de programas em Java (*Web Dynpros*) (Brener, 2016).

A arquitetura da ABAP é baseada no modelo nTier ou Three-tier, isto é, as estruturas do sistema estão divididas por funcionalidades. A definição de nTier é: Three-tier é uma arquitetura cliente-servidor ao qual a interface com o usuário, o modelo lógico funcional (regras de negócios), a base de dados e o acesso aos dados são desenvolvidos e mantidos em módulos independentes, muito frequentemente em plataformas diferentes (UNIX, Windows, etc.).

Na camada de aplicação, o SAP GUI é o sistema de apresentação, isto é, o front-end responsável pela entrada e saída de dados. Na camada de aplicação, o application server, é responsável pelo processamento, este tipo de arquitetura é chamada de nTier ou arquitetura de três camadas, se existe na empresa um ambiente SAP com distintos servidores de aplicações, haverá um ambiente chamado Service Group. Já na camada de dados, que é responsável por executar as diversas transações entre as camadas de aplicação e banco de dados, ou seja, instruções nativas da plataforma de dados - SGBD. O SAP suporta plataformas de dados como: Oracle, Microsoft SQL Server, IBM, Max DB, entre outros, sendo este último a plataforma de dados da versão SAP *Netweaver*. Além das informações de negócio, também estão armazenados diretamente no banco de dados todos os programas, modelos de dados, customizações, entre outros elementos (Brener, 2016).

As classes consistem em código-fonte contendo a definição de possíveis objetos. Todos os componentes, funcionalidades e propriedades de seus objetos são declarados em uma classe. A base para o encapsulamento em objetos ABAP é a classe, ao invés do objeto. As classes são globais para todos os programas ou locais em um único programa. Eles podem ser especializados por herança e podem incorporar interfaces autônomas e customizadas, os atributos descrevem o status e contexto de um objeto. Tecnicamente falando, atributos (atributos de instância) são as variáveis locais de um objeto, que normalmente não podem ser alterados. Uma determinada classe pública também pode conter atributos estáticos, que são usados em conjunto por todos os objetos. Os métodos permitem que os objetos executem operações. Um método-instância, sempre funciona em um objeto específico, ele lê e muda o status deste objeto e interage com outros, chamando seus métodos ou disparando eventos.

Todas as camadas do *ERP* podem estar reunidas em um único computador (servidor), o modelo de arquitetura mais comum nas empresas são os clientes (SAP GUI) instalados

localmente, o que consiste em apenas um servidor de aplicação e um servidor de banco de dados dedicado para se evitar problemas de latência de tráfego entre o processamento da aplicação e a recuperação e persistência dos dados (Brener, 2016).

### **3.2.2 Herança**

Objetos ABAP suportam herança simples, em que uma classe pode ser declarada como a subclasse direta de exatamente uma superclasse. Todas as classes de objetos ABAP são parte de uma árvore de hierarquia de herança originada em uma superclasse comum. Além de seus próprios componentes, uma subclasse também contém os componentes de sua superclasse. A implementação dos métodos da superclasse pode ser sobrescrita em subclasses (Brener, 2016).

### **3.2.3 Interface Autônoma**

A seção de visibilidade pública de uma classe é sua interface externa, objetos ABAP permitem que você crie interfaces autônomas, que pode ser usado por classes como parte de sua interface, ou mesmo como sua interface completa. Objetos pertencentes a várias classes que usam as mesmas interfaces que podem ser tratadas por usuários externos (ou em diferentes empresas, utilizando os mesmos objetos) ou no mesmo caminho. Uma interface independente também pode compreender várias outras interfaces. (Interface de Rastreamento, Monitoramento de pedidos etc. (Brener, 2016).

### **3.2.4 Objetos Compartilhados**

Objetos como instâncias de classes existem na área de memória de um programa e são excluídos o mais tardar quando o programa é encerrado. Como resultado, o acesso entre programas a objetos geralmente não é possível. No entanto, os objetos ABAP permitem o acesso entre programas com objetos, que são objetos na memória compartilhada de um aplicativo servidor, porém, acesso para diversas transações e entre aplicativos Fiori já são amplamente utilizados) (Brener, 2016).

### **3.2.5 Variáveis de referência de objeto e polimorfismo**

Os objetos em um programa só podem ser acessados por referências de objeto em variáveis de referência do objeto. O tipo das variáveis de referência do objeto determina exatamente o que um programa pode fazer com um objeto (Brener, 2016).

## 4. ATIVIDADES DESENVOLVIDAS

Neste Capítulo são apresentadas as atividades desenvolvidas durante o estágio: Leitura básica, tradução de documentos de requisitos funcionais e criação de formulários interativos.

### 4.1 Treinamento

O treinamento foi realizado em 4 meses e foi realizado em 4 etapas: *revisão da literatura, correção de documentos funcionais, programação de formulários e desenvolvimento de serviços*. Na primeira etapa foi feita a revisão da literatura e foram entregues relatórios e materiais para leitura obrigatória. Logo após, foi feita a correção de documentos funcionais. Cada documento programado possuía informações do pedido realizado ou de futuros pedidos a serem solicitados. Os documentos funcionais são de grande importância, tanto para as partes envolvidas, quanto para a equipe de gestão e desenvolvimento, e devem possuir atualização em tempo real com o estoque e banco de dados SAP.

Na terceira fase, foi realizada a programação de formulários, que possuem informações relevantes para o processo de logística, manutenção de armazém ou de customização de embalagens para diferentes empresas como Mercadona e JJ Forwarded. O treinamento foi acompanhado durante reuniões com gerentes e diversos empregados do time de logística e outros setores nas duas primeiras etapas. Na última etapa, foram desenvolvidos serviços para suporte nos módulos – Gestão de Materiais e Planejamento de Pedidos.

### 4.2 Utilizando SAP e AdobeForms

A maior parte das organizações, possuem diversos tipos de produtos que contém logísticas e formas de vendas diferentes. Para estes casos, diversos documentos são impressos para definição do transporte, embalagem e outros dados de mais-valia. A maneira de gerir a seleção de materiais dentro do armazém é realizada com ferramentas SAP e seus diferentes submódulos. O documento de entrega de saída é o principal, quando falamos sobre expedição, com base nele, são feitas diversas operações.

Na Listagem 1 definimos os dados que eventualmente serão passados para o formulário, portanto, é crucial, definir corretamente a interface no primeiro caso. A Listagem 1 também mostra os dados globais e código de inicialização. Para que possamos usar um ‘*template*’ formulário, devemos ativá-lo. Ativar um formulário implica em uma verificação de sintaxe do contexto e, é importante destacar, a geração do módulo de função. O módulo de

função gerado encapsula todas as propriedades de um formulário e é chamado quando um programa é ativado.

### Listagem 1. Exemplo de utilização de estruturas em tabelas

#### REPORT

ZSELECCION\_PERSONAS.

tables: zpersonas, zpedidos, zprodutos.

TYPES: BEGIN OF st\_persona,

numero\_persona type n,  
nombre\_persona(30) TYPE c,  
apellido(30) type c,  
direccion(40) type c, END OF  
st\_persona.

data: ti\_persona type table of st\_persona.

TYPES: BEGIN OF st\_pedido,

id\_pedido(10) TYPE n,  
num\_persona(3) type n,  
id\_producto(30) type n,  
descripcion type c, END  
OF st\_pedido.

data: ti\_pedido type table of st\_pedido.

selection-screen skip.

selection-screen begin of block param with frame title text-008.

parameters: p\_num\_p type n length 3 obligatory,  
p\_nombre type c length 10,  
p\_apelli type c length 10, p\_dir  
type c length 30, p\_idped type  
n length 10, p\_idprod type n  
length 30.

selection-screen skip.

if sy-subrc = 0.

DATA:i\_persona TYPE

ZPERSONAS, wa\_persona TYPE

ZPERSONAS.

DATA:i\_pedido TYPE

ZPEDIDOS, wa\_pedido TYPE ZPEDIDOS.

INSERT zpersonas FROM wa\_persona.

IF sy-subrc NE 0.

MOVE p\_num\_p TO

wa\_persona-numero\_persona. MOVE p\_nombre TO

wa\_persona-nombre\_persona. MOVE p\_apelli TO

wa\_persona-apellido. MOVE p\_dir TO

wa\_persona-direccion. MOVE p\_idped TO

wa\_persona-id\_pedido.

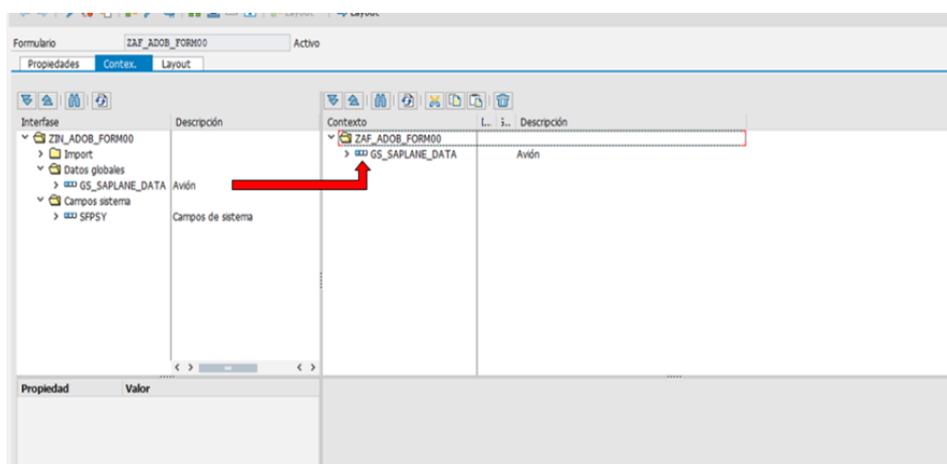
INSERT zpersonas FROM wa\_persona.

```
MOVE p_num_p TO wa_pedido-num_persona.  
MOVE p_idped TO wa_pedido-id_pedido.  
MOVE p_idprod TO wa_pedido-id_producto.  
  
INSERT zpedidos FROM wa_pedido.  
  
IF p_num_p = wa_persona-numero_persona.  
Write: / 'El numero del pedido es' ,wa_persona-id_pedido.  
ENDIF.  
  
selection-screen end of block param.  
ENDIF.  
ENDIF.
```

Em algumas partes da documentação funcional, também encontraremos um objeto de formulário em vez de um modelo de formulário. Em tempo de execução, um programa ABAP recupera os dados necessários e os passa para o módulo de função gerado. Também é possível usar a mesma interface em vários formulários, onde temos duas partes principais: um associado aos dados que se deseja visualizar no formulário e outro associado à aparência gráfica que o formulário terá.

Alternativamente, existe também a possibilidade de recuperar dados da interface. Para isso, é necessário declarar ‘Definições Globais’ e inserir o código na área ‘Inicialização’. Caso recuperemos os dados com o programa controle, não é necessário recuperar os dados na área Definições Globais. Se os dados estão na interface, declaramos esses dados nas variáveis globais na mesma interface e, em seguida, no formulário, arrastamos essa estrutura (isto é feita de forma manual - com o mouse) até a pasta onde definimos o formulário, neste caso ZAF\_ADOB\_FORM00, então os dados já são recuperados para a área do formulário que é chamada de ‘Contexto’ conforme indicado na Figura 1.

Figura 1 – Definição de contexto.



Na Figura 2, é mostrado um exemplo de como é realizada a recuperação dos dados, que pode ser de forma convencional, com chamadas às tabelas/estruturas complexas ou não, correspondentes através dos comandos SELECT e READ, que estão definidos na zona de inicialização de Código do Formulário. Normalmente a informação associada aos campos recuperados será fornecida no documento funcional entregue pelo Engenheiro de Requisitos da equipe de logística (estagiário). Este documento também é analisado antes pelos consultores funcionais. Na figura 2 podemos observar as chamadas para o módulo de função e módulo de função para abertura de trabalho de impressão 'FP\_JOB\_OPEN'.

Figura 2 – Recuperação de dados.

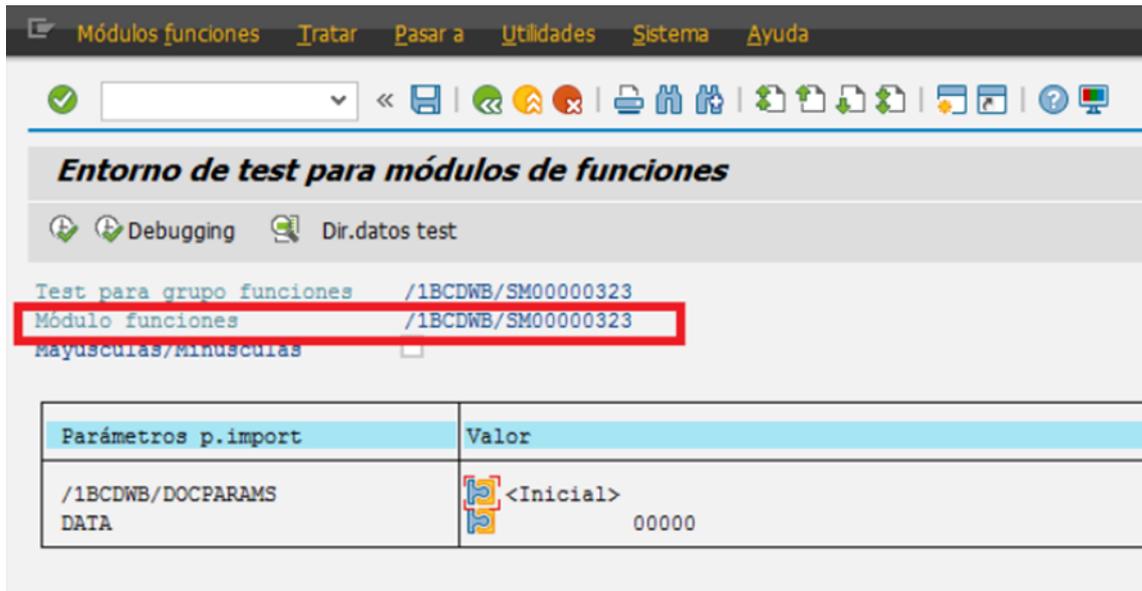
```
DATA: ls_outputparams      TYPE sfpoutputparams,
      ls_/lbcdw/foroutput TYPE fpformoutput.
DATA: lv_fm_name TYPE funcname.

CALL FUNCTION 'FP_JOB_OPEN'
  CHANGING
    ie_outputparams = ls_outputparams
  EXCEPTIONS
    cancel           = 1
    usage_error     = 2
    system_error    = 3
    internal_error  = 4
    OTHERS          = 5.
IF sy-subrc <> 0.
  lanzamos el error.
  RETURN.
ENDIF.
```

Fonte:SAP

Na Figura 3, é mostrado como consultar o nome do formulário criado automaticamente pelo sistema. Para isso, é preciso acessar a transação *osfp*, colocar o nome do formulário e pressionar a tecla F8. Feito isto, o nome do formulário associado aparecerá.

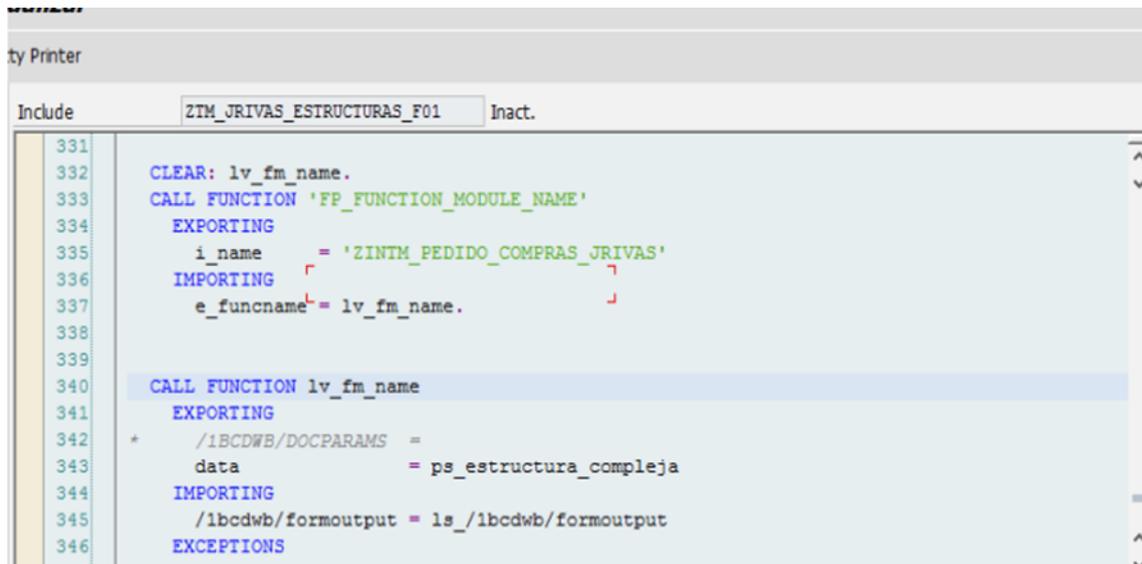
Figura 3 – Módulo de funções.



Fonte: SAP.

Na Figura 4, é mostrado um exemplo de código gerado para consultar o nome do formulário criado automaticamente pelo sistema. O Módulo de função para a obtenção do nome interno do formulário: FP\_FUNCTION\_MODULE\_NAME. O nome do módulo de funções associado aparecerá.

Figura 4 – Funções e Estruturas Complexas.



Fonte: SAP.

Na Figura 5, é mostrado como é feita a chamada para o módulo de função, que deve ser idêntico ao parâmetro definido na interface ps\_estructura\_compleja (em verde). Esses

dados que são definidos na fase 'EXPORTING' serão os parâmetros de entrada da interface, e os dados definidos na fase 'IMPORTING' serão os parâmetros de saída.

Figura 5 – Parâmetros de Entrada.

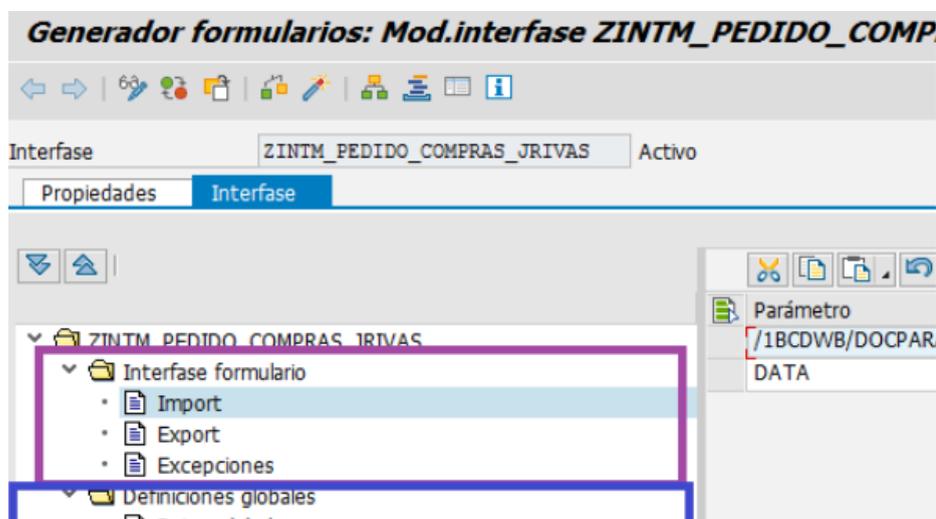
```
CALL FUNCTION lv_fm_name
EXPORTING
  /1BCDWB/DOCPARAMS =
  data                = ps estructura compleja
IMPORTING
  /1bcdwb/formoutput = ls_/1bcdwb/formoutput
EXCEPTIONS
  usage_error          = 1
  system_error         = 2
  internal_error       = 3
  OTHERS               = 4.

IF sy-subrc <> 0.
* Implement suitable error handling here
```

Fonte: SAP.

Na Figura 6, é mostrada onde serão definidos os parâmetros de importação e exportação dos dados exibidos no formulário correspondente, tanto como, suas devidas atualizações e modificações. Na cor roxa, estão selecionados os dados relacionados à recuperação usando o programa de impressão e os dados selecionados na cor azul, estão diretamente relacionados com a interface.

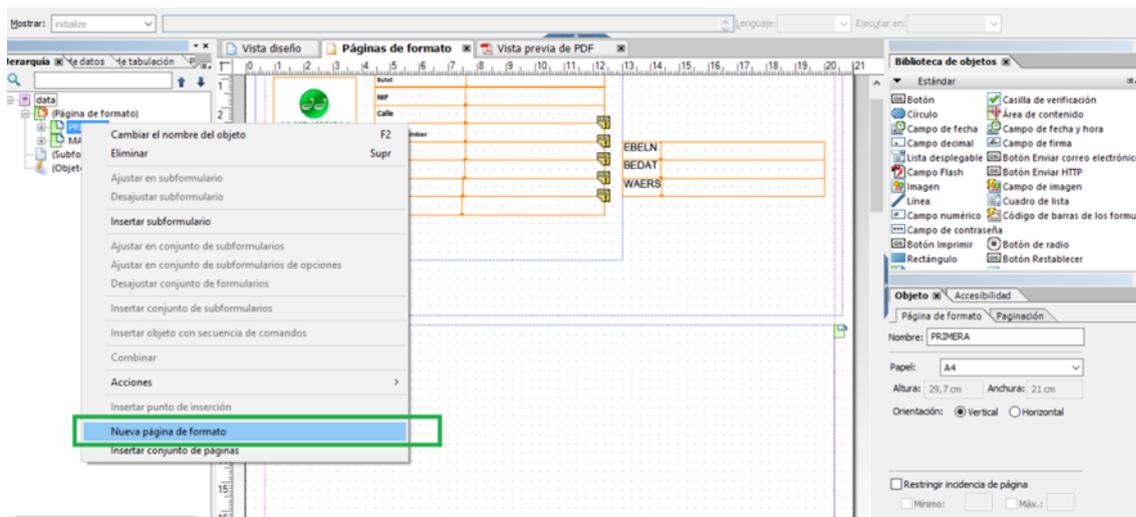
Figura 6 – Recuperação e Impressão de Dados - Interface.



Fonte: SAP

Na Figura 7, é mostrada quais atributos correspondentes com a base de dados devem estar em uma determinada input do formulário, exemplos: ebeln, que é um número identificativo do documento de compra, ebdatt que é um atributo que corresponde a data de expedição do pedido de compra e waers que é a moeda em qual o pedido é feito, todos estes atributos são da tabela EKKO da base de dados SAP.

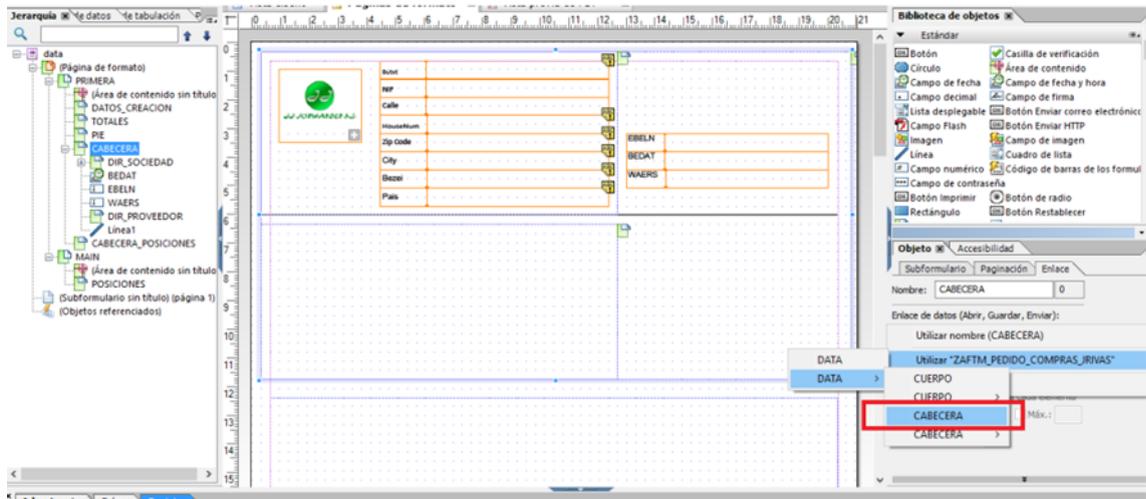
Figura 7 – Parâmetros do Formulário.



Fonte: Adobe Forms.

Na Figura 8, é mostrado como a estrutura do documento é feita. Primeiro, deve-se criar duas páginas de formato, uma com os conteúdos do rodapé e cabeçalho e outra com o conteúdo do formulário. A importação é feita através da funcionalidade de Objeto – Enlace. Anteriormente, como indicado na mesma figura, é preciso colocar um link no subformulário do cabeçalho. Sem o link do formulário, seria impossível que a informação da estrutura complexa fosse desenhada nele. Dessa forma, seleciona-se a opção de link e define-se o link do cabeçalho como indicado na Figura 8 e clica-se no subformulário > Utilizar ZAFSTM\_PEDIDOS\_COMPRAS\_JRIVAS e, depois, nas informações de cabeçalho/cabecera.

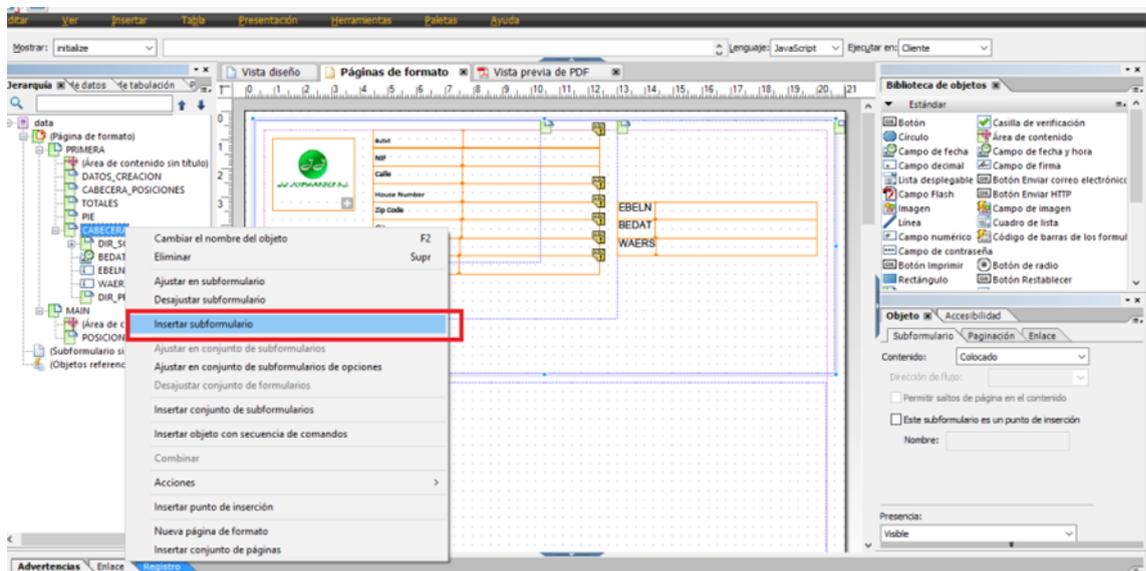
Figura 8 – Importação de Dados de Rodapé e Cabeçalho.



Fonte: Adobe Forms.

Na Figura 9, é mostrado como são criadas as diferentes tabelas com informações mais detalhadas de cada pedido de compra e opções de pagamento da empresa. Primeiramente, clica-se com o botão direito na seção 'cabeçalho' e insere-se um subformulário.

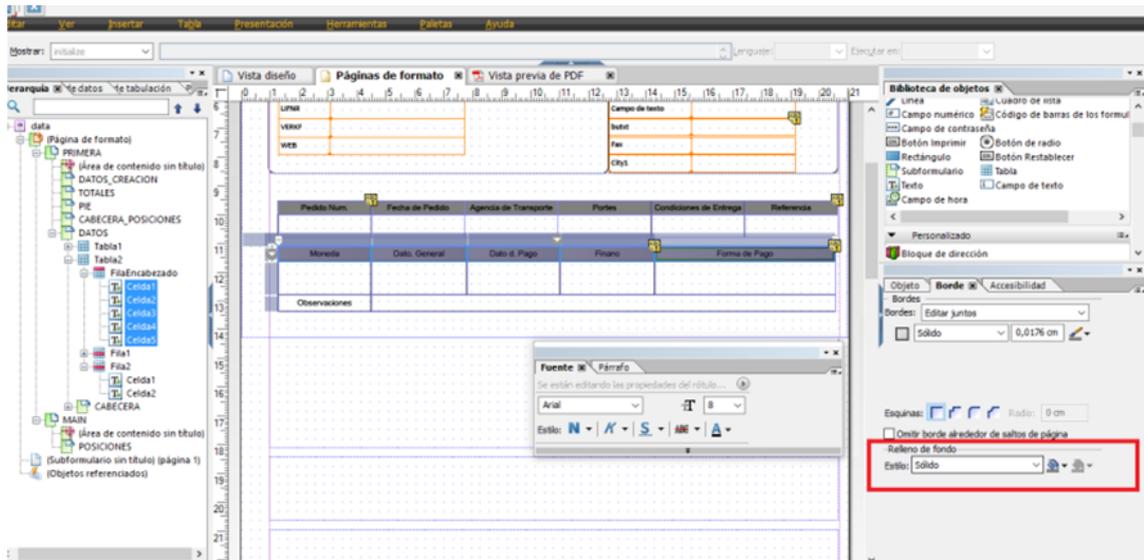
Figura 9 – Inserção de Sub Formulários e Tabelas.



Fonte: Adobe Forms.

Na Figura 10, é mostrado como customizar as tabelas criadas com os dados do pedido de compra. Como pode ser visto, do lado inferior direito, pode-se customizar o estilo da tabela e na 'tab' de objeto relaciona-se o enlace de dados com os respectivos atributos da base de dados.

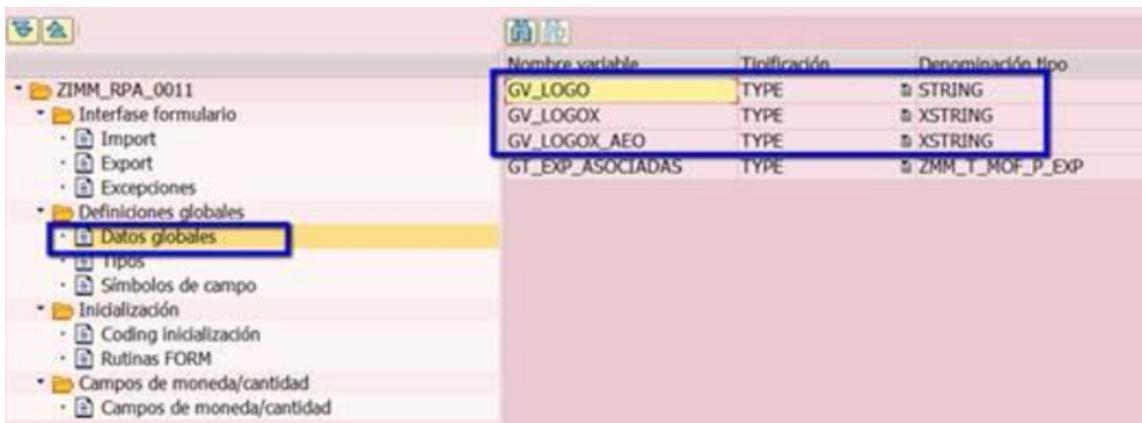
Figura 10 – Customização de Subformulários e Tabelas.



Fonte: Adobe Forms.

Na Figura 11, é mostrada como são definidas as variáveis globais como dados, fotos do logotipo da empresa e tabelas globais com informações para expedição do pedido.

Figura 11 – Inserção do logotipo da empresa.

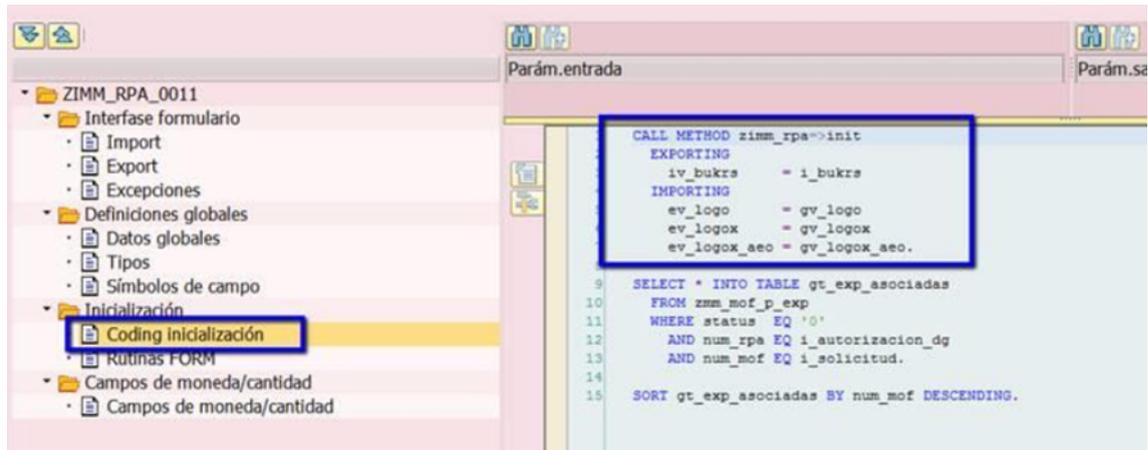


Fonte: SAP.

Na Figura 12, é mostrado como são definidas as 'triggers'. Uma trigger, neste tipo de ambiente, é definida como um objeto atribuído a uma operação no roteiro para acionar uma determinada função, ou seja, pode-se ativar determinadas funcionalidades de negócios usando *triggers* quando uma operação ou status de ordem de produção. Quando são alterados, no nosso caso, estas *triggers* deverão ser iniciadas automaticamente no formulário, com os logotipos e dados da empresa, outras informações que deverão ser carregadas – todos os

pedidos com status 'o' - abertos, com os seus respectivos números de solicitação e autorização.

Figura 12 – Inserção do Logotipo da empresa



Fonte: SAP.

Na Figura 13, é mostrado onde estão definidas o logotipo do tipo variável global dentro do contexto para exibição no formulário. Alguns dados como tipo gráfico da imagem e nome do campo aparecerão dessa forma. O status do atributo deve permanecer 'ativo' para que sejam vistas no formulário.

Figura 13 – Identificação do nome do logotipo.

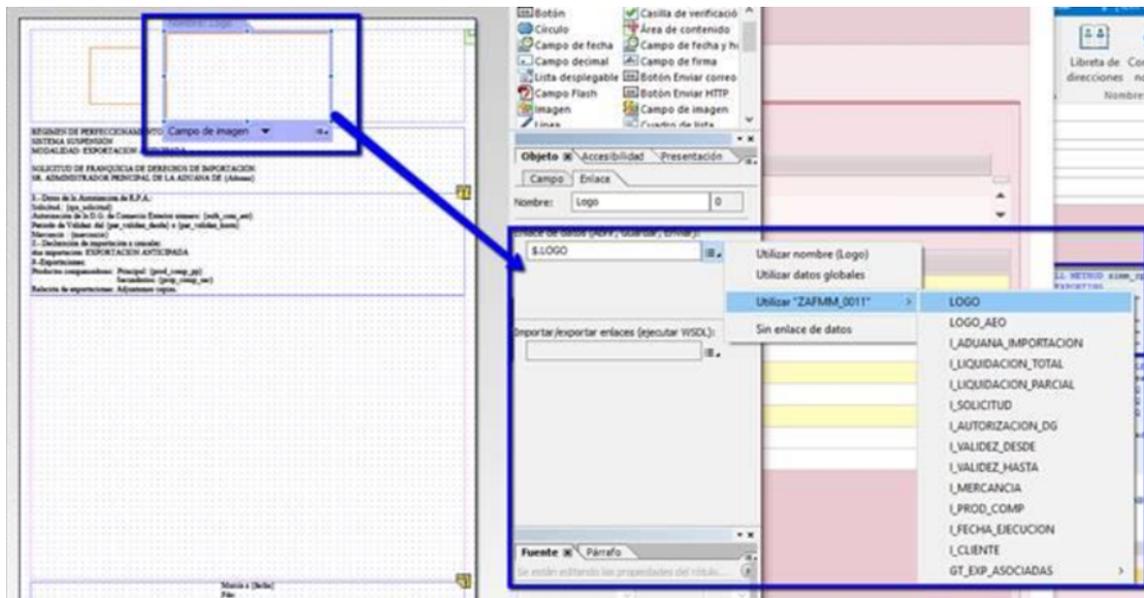


Fonte: SAP.

Na Figura 14, se ainda não se tem o campo "imagem" definido no formulário, deve-se arrastá-lo até o ponto específico onde deve-se colocar o logotipo da empresa (definido no documento funcional). Logo após, clica-se com o botão direito no objeto imagem, depois clica-se na segunda 'tab' chamada enlace onde são mostrados campos para definição do nome do logotipo. Logo abaixo, clica-se no botão à direita e vincula-se o logotipo com o nome

LOGO que já estava anexado ao formulário pela estrutura global que foi arrastado anteriormente.

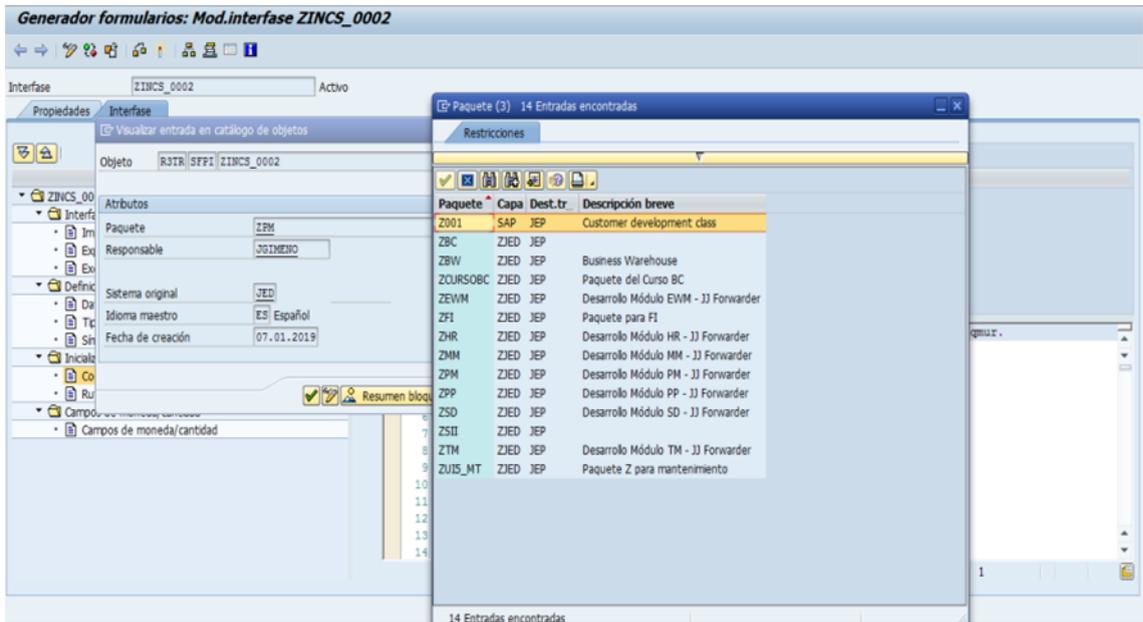
Figura 14 – Seleção de Logotipo a ser inicializado no formulário.



Fonte: Adobe Forms.

Na figura 15, é demonstrado o transporte do formulário finalizado para a máquina de produção. Primeiramente, entra-se na interface correspondente, logo após, nas opções do Menu -> Mover A -> Entrada do Catálogo de Objetos. E, logo então, seleciona-se o tipo de pacote em que o pedido será inserido, pacote Z001 (Customer Development Class).

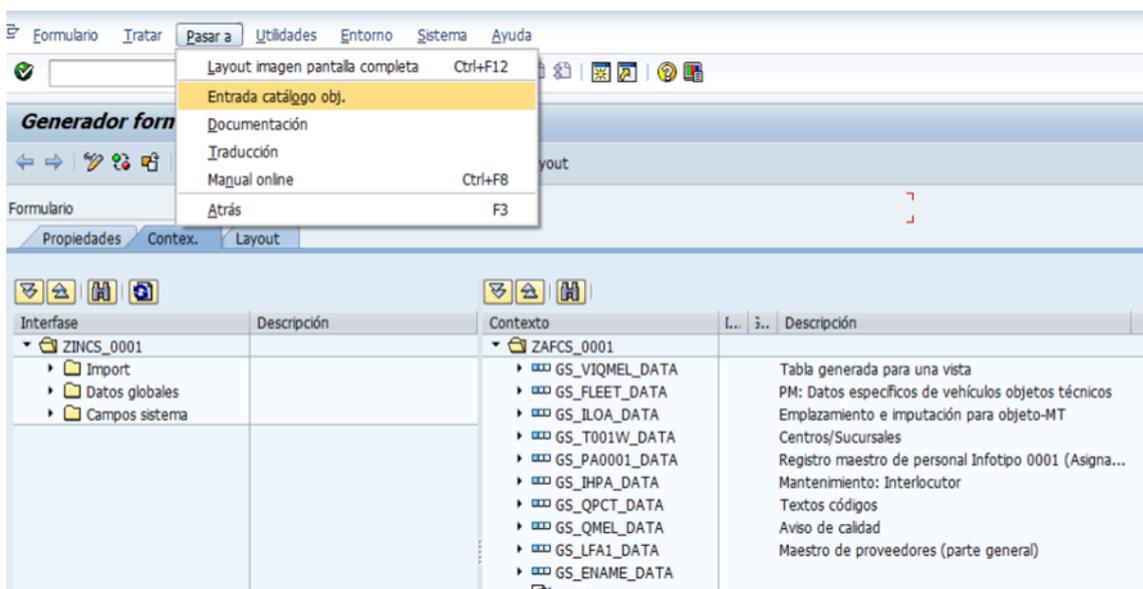
Figura 15 – Transporte do formulário para o pacote de desenvolvimento do cliente.



Fonte: SAP.

Na Figura 16, é demonstrado como passar o formulário criado para a entrada de catálogo de objetos, onde ficam armazenados outros documentos importantes para a gestão da cadeia de suprimentos. Alguns exemplos destes documentos são: *Purchase Order* (Pedido) e *Sales Document* (Documento de Vendas).

Figura 16 – Transporte do formulário para o pacote de desenvolvimento do cliente.



Fonte: SAP.

Pode-se acessar a transação *se09*, que é uma das principais transações para o sistema de transporte, para verificar informações dos pedidos de transporte e quem os criou. Para enviar os dados de forma mais organizada para o produtivo, entra-se na transação *se09*, apresenta-se o nome de usuário e depois clica-se no botão “Visualizar”, e, logo a seguir, seleciona-se o pedido criado (Por exemplo: JEDK901416) e libera-se as tarefas (F9 – Interface e Formulário) que estarão contidas no pedido diretor. Finalmente, entra-se na transação *stms*, que é uma a transação que permite realizar tanto a configuração como administração do sistema de transporte SAP, mediante este tipo de sistema, confirma-se que as modificações do ambiente de desenvolvimento sejam aplicadas ou a nível produtivo, teste, treinamento, entre outros ambientes. O AdobeForms, é um sistema de desenho de formulários que são utilizados para diversas soluções. É uma ferramenta de fácil usabilidade, que permite criar cópias exatas de documentos em papel, com código de barras e assinaturas. O desenho de formulários e seu armazenamento em arquivos magnéticos permite alcançar um maior custo-benefício em processos administrativos de aquisição e armazenamento físico de documentos. A estrutura de dados incorporada no interior do formulário, pode levar à formulação de regras de negócios. Os formulários eram criados com a transação *sfp* - smart form pdf, os smartforms são formulários criados para simular documentos como faturas, recibos, solicitações de pedidos e outros que necessitem ser impressos ou entregues de forma eletrônica. Para visualizar os formulários criados, apenas entramos na transação *sfp*, digitamos o nome do formulário em Interface e clicamos em Visualizar.

#### **4.3 Outras atividades realizadas**

Outra atividade realizada pelo estagiário foi a geração de serviços para suporte na gestão de pedidos, utilizando dados da tabela VBAK que mostra os dados do cabeçalho do documento de vendas, esta tabela também é utilizada em muitos documentos de gestão comercial, sua integração com sistemas móveis é de grande importância para os consultores, que necessitam de informações e documentos em tempo real. Esta tecnologia oferece monitoramento de atividades administrativas, suporte a investigação de incidentes e integração com os diversos documentos para controle e gestão da cadeia de suprimentos.

## 5. CONSIDERAÇÕES FINAIS

O relato descreve as atividades desenvolvidas pelo estagiário no sistema SAP ECC 6.0. Este relatório de estágio descreveu as tarefas desenvolvidas pelo estagiário na empresa *Sothis Enterprise Resource Planning*. Este trabalho colaborou para a formação técnica do estudante.

O sucesso do estágio deveu-se à qualidade no ensino das disciplinas de Gerência de Projetos de Software, Processos de Software, Engenharia de Software, Lógica de Programação, Banco de Dados, SGBD e Gestão do Conhecimento, Tecnologia e Inovação.

A oportunidade de conhecer um ambiente de inovação e gestão tecnológica é de grande importância para a formação profissional dos discentes, tendo em vista que a grade curricular do curso de Sistemas de Informação é bastante diversificada e ampliam as possibilidades de emprego no mundo da inovação. Com a criação do parque tecnológico da região do sul de Minas Gerais, o seu objetivo é coordenar o ensino da pesquisa com as necessidades da atividade econômica regional. Para isso, é necessário dotar os parques com: 1) Uma importante infraestrutura física: parques industriais, redes de comunicação (aeroportos, estradas, rodovias, telefones, fax, etc); 2) Uma infraestrutura educacional e de pesquisa, que produza pesquisadores capazes de gerar ou adotar mudanças tecnológicas, empresários capazes de inovar e trabalhadores qualificados.

Finalmente, temos o problema do financiamento dos parques tecnológicos. Está claro que as necessidades de investimento em infraestruturas e particularmente nas de comunicação da região com o resto do mundo estão fora do orçamento de muitos estados. Encontramos novamente a necessidade de cooperação entre as autoridades regionais e federais, portanto, mesmo que as políticas de parques tecnológicos seja uma política de desenvolvimento econômico regional desenhada e implementada desde o próprio estado. O seu sucesso, irá depender da cooperação do governo federal, já que o mesmo deve cooperar com o governo estadual para o financiamento da infraestrutura que o parque necessita para conseguir atingir a comunicação desta região com o resto do mundo (Gonzalez e Cabrera, 1990)

## REFERÊNCIAS

- F. Andera, D. Derringer, SAP: Implications for computer information systems. Central Michigan University. Recuperado de [Systeme, Anwendungen, Produkte in Der Datenverarbeitung](#) (“Systems, Applications, Products in Data Processing”) Sap: Implications for Computer Information Systems: Journal of Computer Information Systems: Vol 39, No 1 ([tandfonline.com](#)) (1998).
- K. Schwaber, J. Sutherland. Guia do Scrum – Um Guia definitivo para o Scrum: As regras do Jogo. Creative Commons – Estados Unidos. Recuperado de <https://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-portuguese-BR.pdf> (2017).
- M. Brener - ABAP passo a passo. Brasília (2016).
- C. Verwijs, B. Overeem and J. Schartau - Scrum: Um framework para reduzir risco e entregar valor mais cedo (2020).
- J. Gonzalez, J. Cabrera: Mercado de trabajo y progreso tecnológico en la estrategia de desarrollo económico regional. El papel de los parques tecnológicos. 2º Congreso de Economía Regional de Castilla y León (1990). Recuperado de [Acrobat Distiller, Job 10 \(jcyl.es\)](#)