



**Fundação Educacional do Município de Assis**  
**Instituto Municipal de Ensino Superior de Assis - IMESA**

**RAFAEL RODRIGUES TIZATTO**

SAP – SOLUÇÃO EMPRESARIAL

ASSIS  
2008



**Fundação Educacional do Município de Assis**  
**Instituto Municipal de Ensino Superior de Assis - IMESA**

## SAP – SOLUÇÃO EMPRESARIAL

**RAFAEL RODRIGUES TIZATTO**

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientadora: Profa. Dra. Marisa Atsuko Nitto

Analisador (1): Luiz Ricardo Begosso

Analisador (2): Alexandre Charles Cassiano

ASSIS  
2008



**Fundação Educacional do Município de Assis**  
Instituto Municipal de Ensino Superior de Assis - IMESA

**RAFAEL RODRIGUES TIZATTO**

**SAP – SOLUÇÃO EMPRESARIAL**

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientadora: Profa. Dra. Marisa Atsuko Nitto

Área de Concentração: Programação ABAP

ASSIS  
2008

*Dedico este trabalho a todos familiares e amigos que me apoiaram e ajudaram a realizar mais essa etapa da minha vida.*

## AGRADECIMENTOS

Primeiramente agradeço a Deus por ter me dado coragem e condições de fazer uma faculdade;

A Profa. Dra. Marisa Atsuko Nitto pela constante orientação neste trabalho, além da amizade, paciência e pela confiança em mim depositada;

Aos meus pais, Maria e Célio, pelo amor, motivação, apoio, compreensão nos momentos difíceis e incentivos aos estudos;

Aos meus irmãos Célio e Leandro, pela amizade, carinho e amor;

À minha namorada, Jéssica, pelo amor, compreensão, apoio e confiança nas horas mais difíceis desta jornada;

Aos meus avós maternos Oraide e João por estarem presentes em todos os momentos e aos avós paternos Ângelo e Rosa (*in memorium*);

Agradeço a meus queridos sobrinhos que foram fontes de inspiração para que nunca me desanimasse.

Aos amigos irmãos que me apoiaram e ajudaram a realizar este importante passo da minha vida. Nunca serão esquecidos.

Aos professores que tive em toda minha vida, pois sem dúvida foram os construtores da base de conhecimento existente em mim, sem esta certamente não seria possível a realização deste trabalho.

## RESUMO

Com o avanço da tecnologia e a agilidade das informações as empresas começaram a necessitar de um sistema que pudesse ajudar essas informações a ser transmitidas mais rápidas e serem utilizadas por outra área da empresa. Assim os sistemas ERP's (*Enterprise Resources Planning*) cresceram muito.

A SAP (*Systems Applications and Products in Data Processing*) é considerada a maior empresa de ERP. Essa tecnologia, apesar de ser antiga, é pouco conhecida por ser uma ferramenta muito cara e de pouco acesso.

Este trabalho tem como objetivo a demonstração de algumas ferramentas do sistema SAP e como foco a aprendizagem das mesmas e da linguagem de programação ABAP orientado a objetos.

**Palavras-chave:** Orientação a Objetos, Programação.

## ABSTRACT

With the advancement of information technology and agility of companies began to require a system that could help such information be transmitted faster and be used by another area of the company. Once the ERP`s grown a lot.

SAP is considered the largest ERP. This technology, despite being old, little is known for being a tool and very little access.

This study aims at the demonstration, some system utilities such as SAP and focus on learning the same language and the ABAP programming oriented to objects.

**Keywords:** Object-oriented, Programming

.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Empresa SAP na Alemanha.....	19
Figura 2 – Tipos de soluções.....	20
Figura 3 - Esquema adaptado do paradigma " <i>find-bind-execute</i> ".....	30
Figura 4 – Componentes e ferramentas do SAP NetWeaver.....	31
Figura 5 – Planejamento de Recursos Empresariais .....	33
Figura 6 – Hierarquia da SAP R/3. ....	37
Figura 7 – Arquitetura SAP R/3. ....	41
Figura 8 – Módulos do SAP R/3 .....	41
Figura 9 – Divisão de processo .....	42
Figura 10 – Escopo funcional do SAP R/3 .....	44
Figura 11 – Dicionário ABAP .....	45
Figura 12 – Escopo funcional do SAP R/3 .....	48
Figura 13 – Tela de definição de classes no SAP.....	59
Figura 14 – Criação de uma nova sessão interna.....	60
Figura 15 – Criação de mais do que uma sessão interna .....	61
Figura 16 – Implementação de uma classe local .....	62
Figura 17 – Classes e suas instancias. ....	64
Figura 18 – Herança visão geral.....	69
Figura 19 – Herança simples.....	70
Figura 20 – Tela inicial do SAP GUI.....	72
Figura 21 – Tela de criação de usuário do SAP GUI. ....	73
Figura 22 – Tela de finalização da criação de usuário do SAP GUI. ....	73
Figura 23 – Tela de criação de banco no Database Manager. ....	74
Figura 24 – Janela de confirmação de novo banco de dados.....	74
Figura 25 – Tela que mostra a instancia NSP.....	74
Figura 26 – Tela de criação do banco de dados. ....	75
Figura 27 – Tela de criação do banco de dados II. ....	75

Figura 28 – Tela de conexão do banco de dados. ....	76
Figura 29 – Tela de inicialização da aplicação server.....	77
Figura 30 – Tela de inicialização do SAP GUI.....	77
Figura 31 – Tela de inicialização do SAP.....	78
Figura 32 – Tela de criação de um programa em ABAP.....	79
Figura 33 – Tela de configuração de criação de programas em ABAP. ....	80
Figura 34 – Tela de escolha do pacote aonde salvar o programa. ....	80
Figura 35 – Tela do ABAP Editor.....	81
Figura 36 – Tela da execução do programa ZPCALC. ....	83
Figura 37 – Tela do resultado da aplicação ZPCALC. ....	83
Figura 38 – Tela da execução do programa ZRODIZIO. ....	85
Figura 39 – Tela do resultado da aplicação ZRODIZIO. ....	86
Figura 40 – Tela do relatório Z_RELATORIO.....	88
Figura 41 – Tela da transação SE11.....	88
Figura 42 – Tela dos campos da tabela SPFLI. ....	89
Figura 43 – Tela da escolha de campos a ser mostrado. ....	90
Figura 44 – Tela do resultado do relatório.....	90

## LISTA DE TABELAS

Tabela 1 – Tabela de transações do SAP.....	52
Tabela 2 – Tabela de comando em ABAP.....	55

## LISTA DE ABREVIATURAS E SIGLAS

ERP	Enterprise Resources Planning
SAP	Systems Applications and Products in Data Processing
IBM	International Business Machines
ABAP	Advanced Business Application Programming
GmbH	Gesellschaft mit beschränkter Haftung
NYSE	New York Stock Exchange
CeBIT	Feira Internacional de Tecnologia de Informação, Telecomunicações, Softwares e Serviços
NYSE	New York Stock Exchange
MRP	Material Requirement Planning ou Planejamento das Requisições de Materiais
MRPII	Manufacturing Resource Planning ou Planejamento dos Recursos de Manufatura
DBMS	Sistema Gerenciador de Banco de Dados
BOR	Business Object Repository
DCOM	Distributed Component Object Model
CORBA	Common Object Request Broker Architecture
CRM	Customer Relationship Management
TI	Tecnologia de Informação
TCO	Total cost of ownership
ROI	Return on Investment
SCM	Supply Chain Management
SRM	Supplier Relationship Management
RFID	Radio-Frequency Identification

SOA      Service-oriented architecture

CEO      Chief Executive Officer

## SUMÁRIO

<b>INTRODUÇÃO</b> .....	15
1.1 OBJETIVO .....	16
1.2 MOTIVAÇÃO .....	16
1.3 ESTRUTURA DO TRABALHO .....	17
<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	18
2.1 SAP (Systems Applications and Products in Data Processing) .....	18
<b>2.1.1 Experiência e tecnologia para otimizar seus negócios</b> .....	18
<b>2.1.2 Soluções SAP</b> .....	19
2.1.2.1 Soluções Corporativas.....	20
2.1.2.2 Soluções de Negócios .....	27
2.1.2.3 Soluções para Pequenas e Médias Empresas.....	27
2.1.2.4 Plataforma .....	29
2.2 ERP (Enterprise Resources Planning) .....	31
<b>2.2.1 Entendendo o ERP</b> .....	32
<b>2.2.2 Como o ERP pode melhorar a performance de uma empresa</b> .....	34
2.3 ABAP .....	35
2.4 SAP/R3 .....	36
2.4.1 Função do SAP/R3 .....	40
2.4.2 Divisão dos Processos .....	42
<b>FERRAMENTAS</b> .....	43
3.1 JAVA.....	43
3.2 ESTRUTURA ABAP .....	44
<b>3.2.1 Dicionário ABAP</b> .....	44
<b>3.2.2 Tabelas Transparentes</b> .....	46
<b>3.2.3 Estruturas</b> .....	47
3.3 ABAP ORIENTADO A OBJETO .....	55
<b>3.3.1 O ambiente de tempo de execução</b> .....	57
<b>3.3.2 A extensão da linguagem orientada a objeto</b> .....	57

<b>3.3.3 De Grupos de Funções a Objetos</b> .....	59
3.3.3.1 Declarando Métodos.....	64
3.3.3.2 Implementando Métodos .....	65
3.3.3.3 Chamando Métodos .....	66
<b>3.3.4 Herança</b> .....	67
<b>IMPLEMENTAÇÃO</b> .....	71
4.1 CONFIGURAÇÕES DO SISTEMA.....	71
4.2 INSTALAÇÃO DO MINISAP .....	72
4.3 EXEMPLOS DE APLICAÇÕES .....	79
<b>4.3.1 Aplicação 1: Programa de uma calculadora (ZPCALC)</b> .....	79
<b>4.3.2 Aplicação 2: Rodízio de placas de automóvel (ZRODIZIO)</b> .....	84
<b>4.3.3 Aplicação 3: Emissão de Relatório (Z_RELATORIO)</b> .....	86
<b>CONCLUSÃO</b> .....	92
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	93

## CAPÍTULO 1

### INTRODUÇÃO

---

A globalização e diversificação de negócios continuam a adicionar complexidade e dinamicidade nos processos empresariais. A cada dia aumentam atividades de coordenação, monitoração e simulação de novos cenários em empresas, com o objetivo de formular ações rápidas de correções de metas estabelecidas do negócio. Esse processo necessita de informação atualizada e consistente, que pode ser disponibilizada por meio de software que contenha e processe dados de diversos setores da empresa, função básica dos chamados sistemas integrados de gestão empresarial, os chamados sistemas ERP.

O software ERP baseia-se na existência de bases de dados comuns que permitam partilhar a informação com diversos usuários. Dessa forma, departamentos deixam de ser proprietários de dados ou de sistema de informação, passando a compartilhar os mesmos recursos de informação e de um conjunto de ferramentas de software de apoio à execução, controle e gestão do negócio, disponíveis no ERP, o que é um diferencial importante nestes sistemas. Exemplos desses produtos: SAP, Baan, JDEdwards, PeopleSoft, Oracle, IFS.

Mesmo as pequenas empresas produtoras de software já implementam os conceitos de integração, a diferença dessas está nas ferramentas adicionais de apoio à decisão, ainda de menor funcionalidade. Diante da crescente procura de produtos ERP no mercado, apesar de sua complexidade e custo relativamente alto, isto exige da área acadêmica um maior envolvimento na realização de estudos, oferecendo entre outros, formação de profissionais com habilidades em gestão de tecnologias de produtos ERP.

A empresa Alemã SAP, surgiu em 1972 quando cinco funcionários da IBM resolveram abandonar os seus empregos e apostaram em uma nova ideia, abrir uma empresa de software. Após 35 anos é líder mundial em sistemas empresariais.

O SAP é um sistema em ERP e é considerado o maior e mais complexo sistema de ERP no mundo. ERP é um sistema único que serve para integrar todos os departamentos e funções de uma companhia.

Nessa ferramenta é possível usar várias linguagens de programação, uma delas é a linguagem própria que foi desenvolvida por essa empresa, chamada de ABAP. Além do próprio ABAP as versões mais novas do SAP também podem ser implementadas por mais duas linguagens, JAVA e ABAP Orientado a Objetos.

O ABAP é uma linguagem que necessita de conhecimentos avançados de programação e é principalmente utilizada pelos programadores da SAP que desenvolvem o software e por consultores informáticos que adaptam o software às necessidades dos clientes.

O ABAP é a principal linguagem utilizada no software cliente-servidor SAP R/3. Em 1999, com a versão R/3 4.5, a SAP lançou uma versão de ABAP que suporta programação orientada a objetos. Objetos ABAP são conjuntos completos de comandos que foram introduzidos dentro da linguagem ABAP. Esta extensão orientada a objeto constrói sobre a linguagem existente, sendo totalmente compatível com ela. Pode se utilizar objetos ABAP em programas já existentes, e pode também usar o ABAP “convencional” em novos objetos de programas ABAP.

Os objetos ABAP suportam programação orientada a objeto. A orientação a objeto, também conhecida como paradigma orientado a objeto é um modelo de programação que unem dados e funções em objetos.

## 1.1 OBJETIVO

Este trabalho tem como objetivo pesquisar e adquirir conhecimento sobre algumas ferramentas do sistema SAP já que é considerado o maior e mais complexo sistema do mundo. Outro objetivo é aprender a linguagem de programação ABAP orientado a objetos que podem ser utilizadas nas versões mais novas do SAP.

## 1.2 MOTIVAÇÃO

A motivação para o desenvolvimento desse trabalho é a oportunidade de estar adquirindo novos conceitos e passando para a área acadêmica todo o material estudado, além da grande oferta de empregos para profissionais com conhecimento nesta área.

### 1.3 ESTRUTURA DO TRABALHO

No capítulo 1 mostramos o início do trabalho, contendo apenas a introdução.

No capítulo 2 temos a fundamentação teórica das ferramentas utilizadas, falando sobre a empresa SAP e alguns conceitos sobre ERP, SAP/R3 e ABAP

No terceiro capítulo mostraremos o funcionamento das ferramentas utilizadas, tendo como foco principal o ABAP e ABAP orientado a objeto.

No capítulo 4 será mostrado passo a passo a instalação da ferramenta MiniSAP e algumas implementações usando a linguagem ABAP.

No quinto capítulo será apresentada a conclusão, mostrando o resultado do trabalho.

## CAPÍTULO 2

### FUNDAMENTAÇÃO TEÓRICA

---

Neste capítulo será feita a fundamentação teórica das ferramentas que serão utilizadas neste trabalho. Será feita uma descrição resumida sobre SAP, ERP e SAP/R3.

#### 2.1 SAP (*Systems Applications and Products in Data Processing*)

Fundada em 1972 como *Systems Applications and Products in Data Processing* (**S**istemas, **A**plicativos e **P**rodutos para Processamento de Dados), a SAP é, reconhecidamente, líder mundial em soluções colaborativas de negócios, para todos os setores da economia e mercados verticais.

Atendendo a mais de 76.000 clientes em todo o mundo, a SAP é a maior empresa de software de gestão empresarial e a terceira maior fornecedora independente de software na classificação mundial. Ela apresenta uma notável história de inovação e crescimento que fez verdadeiros líderes de mercado. Atualmente, a SAP emprega mais de 51.800 funcionários em mais de 50 países. Os seus profissionais estão determinados a oferecer o mais alto nível de serviço e suporte aos clientes.

##### 2.1.1 Experiência e tecnologia para otimizar seus negócios

A SAP tem aproveitado a vasta experiência para oferecer um amplo leque de soluções para reforçar cada aspecto das operações corporativas. Ao usar as soluções SAP, organizações de todos os tamanhos – incluindo soluções para pequenas empresas e soluções para médias empresas – podem reduzir custos, melhorar o desempenho e ganhar agilidade para reagir às necessidades de negócios em transformação.

A SAP desenvolveu também a plataforma SAP NetWeaver, que possibilita aos clientes extrair mais benefícios de seus investimentos de TI (Tecnologia de Informação). Para assegurar a posição de liderança da SAP na área de tecnologia, a SAP Ventures investe em organizações empreendedoras emergentes que estão desenvolvendo novas e excitantes tecnologias. E, por meio da SAP Research, apresenta novas idéias para futuras soluções.

Na SAP, a atenção à qualidade e as melhores práticas são o coração de tudo que é feito. O compromisso da SAP com a qualidade se manifesta pelos prêmios de qualidade que recebe todos os anos.

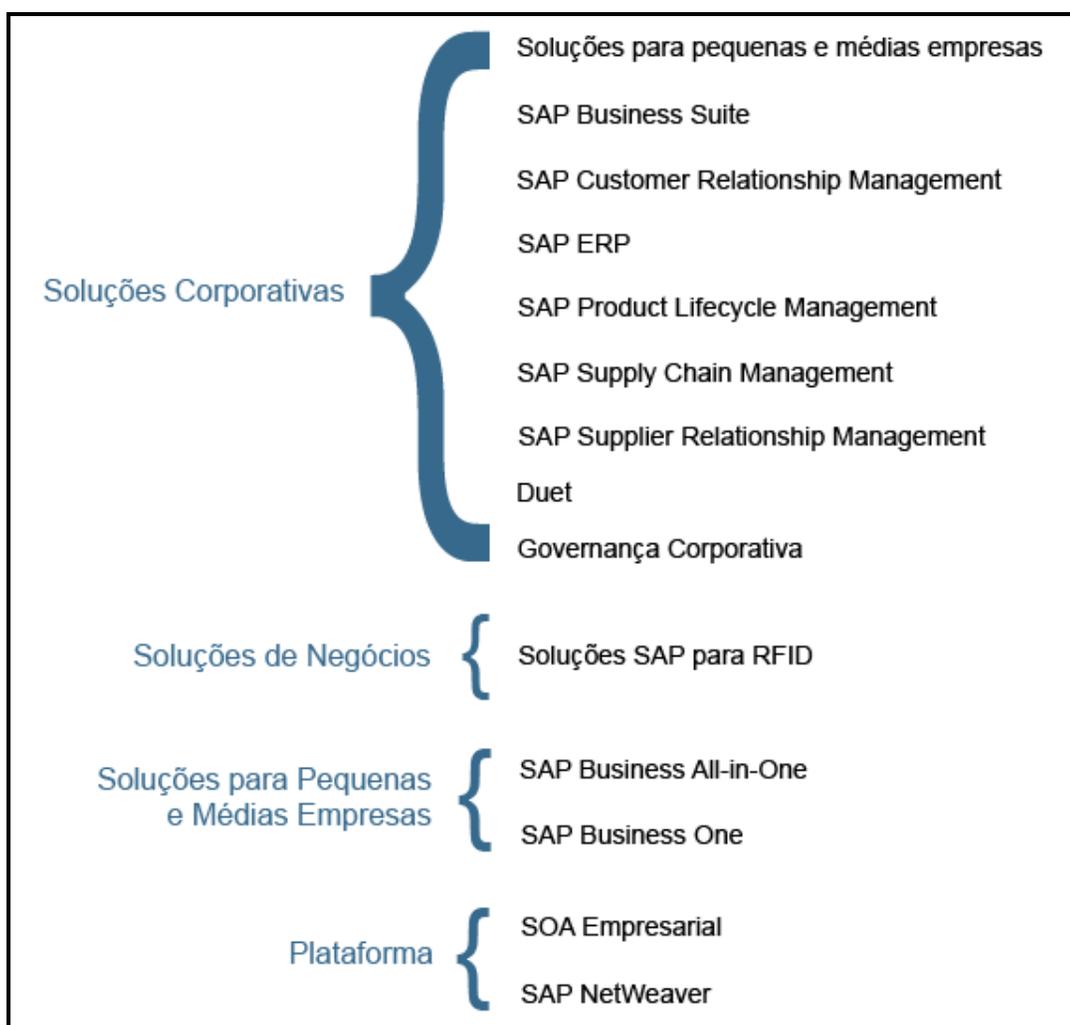
Sediada em Walldorf, Alemanha, a SAP é agora uma corporação de € 10,3 bilhões. A Figura 1 mostra a sede da SAP na Alemanha. As ações da SAP são negociadas em diversas bolsas de valores, incluindo Frankfurt e Nova York, sob o código “SAP”.



**Figura 1 – Empresa SAP na Alemanha.**

### **2.1.2 Soluções SAP**

Os softwares SAP lhe dão visibilidade em tempo real, ao longo de toda a sua empresa, para que você possa integrar sua cadeia de fornecimento, levar produtos ao mercado mais rapidamente, obter mais de suas aquisições e eliminar a duplicação de esforços. A SAP é líder mundial em soluções empresariais, oferecendo software e serviços avançados que atendem às suas necessidades específicas. A Figura 2 mostra os tipos de soluções SAP. (<http://www.sap.com/brazil/index.epx>)



**Figura 2 – Tipos de soluções.**

#### 2.1.2.1 Soluções Corporativas

- **Soluções para Pequenas e Médias Empresas**

Hoje, mais do que nunca, as empresas, qualquer que seja o seu tamanho, devem competir numa perspectiva global, dispor das tecnologias de informação mais modernas e saber aplicá-las de forma eficaz e rentável. As empresas podem ter várias dimensões, mas todas elas desenvolvem atividades similares (comprar, vender, manter relações com os seus clientes, organizar a contabilidade, gerir os recursos humanos e adaptar se às modificações das normas legais e financeiras).

Atualmente, a SAP disponibiliza as seguintes soluções:

**Médias empresas:** O SAP Business All-in-One é a solução pré-configurada para os objetivos específicos de diferentes setores de atividades e para garantir uma rápida implementação. Adicionalmente, podem ser personalizadas para se adaptarem totalmente aos requisitos individuais da sua empresa. O SAP Business All-in-One endereça os problemas reais do cliente, respondendo diretamente às suas questões, requisitos e necessidades.

**Pequenas empresas:** O SAP Business One é um software empresarial simples, mas poderoso, que vai ao encontro das necessidades únicas das pequenas e médias empresas. Com uma vasta gama de funções de negócio comuns, incluindo contabilidade, elaboração de relatórios, logística e gestão de oportunidades de vendas, entre outras. Disponível num pacote de fácil utilização e de rápida implementação, software para empresas preenche todas as suas necessidades e continuará a fazê-lo, acompanhando o crescimento do seu negócio.

- **SAP Business Suite**

O SAP Business Suite é a família mais abrangente e flexível de softwares de negócios do mercado, com as melhores funcionalidades desenvolvidas para assegurar a integração, os recursos específicos por setor vertical, a escalabilidade ilimitada e a colaboração muito simples pela Internet. Os softwares empresariais SAP Business Suite permitem o gerenciamento dos processos de negócios mais críticos. Juntos, eles formam um conjunto fortemente integrado, que agrega valor a

cada aspecto de sua organização, tanto internamente como junto à cadeia externa de valor.

Os aplicativos da família SAP Business Suite se baseiam na plataforma SAP NetWeaver, a mais completa plataforma de integração e de aplicações. Isto reduz o custo total de propriedade (TCO) de todo o ambiente de TI e suporta a evolução do SAP Business Suite para uma arquitetura baseada em serviços.

- **SAP Customer Relationship Management**

O SAP Customer Relationship Management (SAP CRM) é a única solução capaz de viabilizar processos de negócios totalmente confiáveis, abrangentes e centrados no cliente, proporcionando um benefício real de cada cliente. O software SAP CRM se aplica a processos dirigidos aos mercados verticais para suportar departamentos que interagem diretamente com os clientes, especificamente nas áreas de marketing, vendas e serviços. A adoção do SAP CRM assegura uma visibilidade de 360 graus de todos os pontos de contato do cliente e dos canais de interação, incluindo o campo, a Internet, centros de interação e parceiros do canal. A solução de Customer Relationship Management oferece ainda poderosos recursos analíticos. Só o SAP CRM é capaz de:

**Suportar processos relacionados com clientes, de ponta a ponta:** O SAP CRM assegura a orquestração de todas as tarefas relacionadas aos clientes, de um departamento para outro, incorporando, de forma transparente, atividades tais como o *fulfillment*, a distribuição, o faturamento e contas a receber.

**Suprir toda a organização com informações de clientes:** A solução SAP CRM reúne todas as fontes relevantes de informações dos clientes, distribuídas por toda a empresa, contribuindo para um melhor processo de tomada de decisões.

**Oferecer benefícios imediatos:** Só o SAP CRM permite que as empresas resolvam, em primeiro lugar, as prioridades estratégicas e cumpram com os objetivos mais rapidamente. A solução pode ser expandida de forma gradual

e cada etapa trará, de uma maneira tangível, o correspondente retorno sobre os investimentos.

- **SAP ERP**

O aplicativo SAP ERP é um software integrado de planejamento de recursos corporativos, de qualidade mundialmente reconhecida, destinado a atender aos principais requisitos de software das mais exigentes empresas de médio e grande porte, de todos os setores e mercados verticais, em qualquer país do mundo. O software SAP ERP é constituído de quatro soluções individuais que sustentam as principais áreas funcionais das organizações.

- ERP Financials
- SAP ERP Human Capital Management
- SAP ERP Operations
- SAP ERP Corporate Services

O software SAP ERP atende às principais necessidades de empresas de todo tipo, como resultado de três décadas de experiência acumulada da SAP.

Desenvolvido sobre a plataforma SAP NetWeaver – Reduz a complexidade de TI, suportando, ao mesmo tempo, a escalabilidade e o crescimento, por meio de uma abrangente plataforma de integração e aplicativos.

Software fortemente integrado, para otimizar a interação dos processos de forma multifuncional – Possibilita uma estreita colaboração dentro e fora dos limites das organizações.

Software otimizado por recursos específicos e melhores práticas de cada setor – Permite que as empresas reduzam seu custo total de propriedade (TCO) e tenham um retorno mais rápido sobre seus investimentos (ROI), beneficiando-se de uma infra-estrutura de TI mais flexível, própria para a inovação.

Concebida para suportar operações internacionais – Propiciando grande eficiência e resultados bem sucedidos nas operações e concorrências globais.

- **SAP Product Lifecycle Management**

O SAP Product Lifecycle Management (SAP PLM) oferece uma base sólida para o desenvolvimento e introdução de novos produtos (new product development and introduction, NPDI) de sucesso, permitindo que você administre pessoas e informações em um único e integrado processo. Com o SAP PLM, você pode conectar e envolver todos os departamentos, inclusive marketing e vendas, planejamento e produção, aquisições e manutenção. Além disso, o SAP PLM permite uma fácil colaboração entre parceiros, fornecedores, produtores e provedores de serviços – e até mesmo consumidores.

O SAP PLM fornece a grupos de colaboração uma visão holística de processos empresariais relacionados a produtos e informações em todo o ciclo de vida de produtos e bens – desde conceitualização, design e engenharia até o agregamento do produto e gerenciamento de mudanças, serviço e manutenção.

Como a solução mais abrangente para gerenciamento de ciclo de vida de produtos no mercado hoje, o SAP PLM pode ajudar a sua organização das seguintes formas:

- Permitindo engenharia colaborativa, desenvolvimento de produtos e gerenciamento de projetos, bens e qualidade.
- Permitindo a administração de requisitos de ambiente, saúde e segurança junto a diversos parceiros
- Trabalhando em harmonia com SAP CRM, SAP SCM, SAP SRM, e SAP ERP — melhorando assim a comunicação entre os departamentos de marketing, vendas, engenharia, produção e serviços
- Trazendo uma solução completa para a administração de todo o ciclo de vida do produto, reduzindo custos de integração e pondo o serviço à disposição de consumidores mais rapidamente

- **SAP Supply Chain Management**

O software de gestão da cadeia de fornecedores SAP Supply Chain Management (SAP SCM) pode ajudar a sua empresa a transformar cadeias lineares de fornecimento em uma rede de cadeias de suprimentos adaptáveis, permitindo que

você acesse os conhecimentos e recursos de seus colegas, ajuste-se de forma inteligente a mudanças nas condições de mercado e mantenha-se focado no cliente. Tudo isso traz uma vantagem competitiva à sua empresa. O software SAP SCM possibilita redes de fornecimento adaptáveis ao oferecer à sua empresa capacidades de planejamento e execução para gerenciar operações empresariais, assim como tecnologias de coordenação e colaboração para essender estas operações além das fronteiras corporativas. Como resultado, a sua empresa pode atingir melhorias mensuráveis e sustentáveis com a redução de custos, aumentos nos níveis de serviços e ganhos de produtividade – levando a maiores margens de lucros.

- **SAP Supplier Relationship Management**

O SAP Supplier Relationship Management (SAP SRM) automatiza processos entre a procura por fornecedores e aquisições, dentro da empresa e ao longo de toda a base de oferta. Aumentando a visibilidade da cadeia de produção e lhe dando uma visão completa de gastos globais.

Com o software SAP SRM, você pode controlar todo o ciclo de produção – desde a estratégia até a execução – para lhe ajudar a otimizar a seleção de fornecedores, diminuir o tempo dos ciclos e então construir relacionamentos funcionais e sustentáveis com fornecedores. EE, com as vantagens exclusivas de conteúdo consolidado e dados mestres, o SAP SRM lhe ajuda a tomar e executar decisões que se alinham com a estratégia corporativa. Para ver exatamente como o SAP SRM agrega valor à sua empresa, veja os SAP Business Maps.

- **Duet**

O software Duet proporciona um acesso totalmente transparente aos dados e processos de negócios SAP por intermédio do Microsoft Office. Esta iniciativa foi desenvolvida para revolucionar a maneira como os profissionais da informação interagem com os aplicativos corporativos. O Duet é o resultado de uma colaboração pioneira entre a SAP e a Microsoft, e é o primeiro produto criado e suportado em conjunto por esses dois líderes do setor de tecnologia.

Pelo fato de utilizar a familiar interface do Microsoft Office para conectar os profissionais da tecnologia da informação aos aplicativos SAP, o software Duet reduz o tempo de aprendizado e proporciona um maior acesso às informações e diretrizes da empresa, resultando em um elevado índice de adoção. Com o Duet, as organizações podem cumprir com mais rigor suas normas e regulações, aprimorar seus processos de tomada de decisões e se beneficiar da economia de tempo e recursos viabilizada por esta interação.

- **Governança Corporativa**

Novas regulamentações governamentais, pressão crescente de mercados financeiros e exigências cada vez maiores por parte de acionistas são aspectos que exigem que as empresas encarem a governança corporativa, o cumprimento de normas e os requisitos regulatórios sob novo prisma. As organizações mais avançadas estão assumindo uma posição mais pró-ativa e estratégica em relação a requisitos de governança, reconhecendo que as exigências regulatórias representam uma oportunidade para que elas se destaquem no mercado.

É por isso que as organizações mais eficientes estão recorrendo à SAP. Líder mundial em soluções corporativas de negócios, a SAP permite que as empresas desenvolvam uma estratégia eficaz e sustentável de conformidade e governança. A combinação imbatível de conhecimentos sobre processos de negócios, tecnologias integradas e presença global transforma a SAP no parceiro mais confiável para aquelas organizações determinadas a transformar requisitos regulatórios em vantagem competitiva.

Mas a parceria com a SAP vai muito além de assessoria à sua empresa no cumprimento das normas. A SAP trabalha lado a lado com sua empresa na criação de uma estratégia integrada e flexível para aproveitar todos seus investimentos de TI na administração pró-ativa de seus processos: isso permite reduzir discrepâncias entre os diferentes processos, aumentar sua eficiência operacional e, em última instância, gerar maior valor da organização perante os acionistas.

### 2.1.2.2 Soluções de Negócios

- **Soluções SAP para RFID**

As soluções SAP para RFID são soluções de nível corporativo que dão suporte a RFID (identificação por radiofrequência), códigos de barra e outras tecnologias de Auto-ID, bem como à serialização. As soluções permitem monitoramento de uso real por sensores e respondem à presença e ao movimento de objetos em tempo real.

As soluções SAP para RFID se baseiam em dois componentes principais:

- SAP Auto-ID Infrastructure – Facilita a captura de dados serializados de aparelhos em pontos locais e fornece o contexto de negócios para transformar os dados em eventos de negócio significativos.
- SAP *object event repository* – Dá suporte a aplicativos que precisam de visibilidade entre pontos na empresa ou entre a empresa e outros parceiros comerciais, sustentando uma ampla variedade de processos de negócio.

### 2.1.2.3 Soluções para Pequenas e Médias Empresas

- **SAP Business All-in-One**

As soluções do software SAP Business All-in-One são pré-configuradas para responder aos objetivos específicos de diferentes setores de atividades e para garantir uma rápida implementação. Adicionalmente, essas soluções pré-configuradas são complementadas pelos parceiros SAP para se adaptarem totalmente às necessidades individuais da sua empresa. Focadas nos principais processos de negócio do setor, elas visam cobrir praticamente todas as necessidades.

As empresas procuram essencialmente experiência e baixo risco. Experiência no sentido de que exigem do seu parceiro de negócio a compreensão das suas necessidades e requisitos, as necessidades típicas do seu setor - os seus problemas. risco porque procuram soluções pré-configuradas de baixo custo, que

possam ser implementadas rapidamente e que lhes permitam atender às suas necessidades relativas a processos de negócio, tendo um parceiro em quem confiar.

Todas as soluções pré-configuradas do SAP Business All-in-One são desenvolvidas para responder a estes requisitos, porque cada solução contém:

- documentação de marketing especialmente concebida para cada setor de atividades;
- equipe de vendas que conhece as características específicas do seu setor e fala a sua linguagem e terminologia;
- um modelo de implementação que permite repetições;
- uma seleção de clientes referenciáveis em seu próprio setor;
- uma equipe de suporte que compreende as necessidades de uma solução para cada setor.

Uma empresa média necessita do mesmo conjunto de funcionalidades que uma grande empresa. O que elas não necessitam é da mesma infra-estrutura e complexidade de implementação. O software SAP Business All-in-One é um conjunto de soluções pré-configuradas qualificado pela SAP, mas desenvolvido e entregue ao mercado pelos parceiros, sendo estes igualmente responsáveis pela sua venda, implementação, manutenção e suporte, o que permite reduzir os custos e o tempo de implementação.

- **SAP Business One**

O SAP Business One é um aplicativo de software de gestão empresarial de custo acessível e muito fácil de usar. Trata-se de um sistema único, que cobre praticamente todas as operações vitais das empresas, como vendas, finanças e contabilidade, incluindo até CRM pela Internet.

O SAP Business One permite que sua empresa gerencie melhor seus processos, adquira maior visibilidade sobre todas suas operações e receba as informações necessárias para liberar seus profissionais para iniciativas de planejamento estratégico, expansão dos negócios e desenvolvimento de clientes.

O SAP Business One confere uma visibilidade clara sobre todas as operações de sua empresa, o que facilita a tomada de decisões fundamentadas, com grande rapidez. Os processos de negócios, antes demorados e complexos, foram totalmente automatizados. Isso libera os profissionais para dedicarem mais tempo às principais competências da empresa e ao seu crescimento. Os principais desafios são:

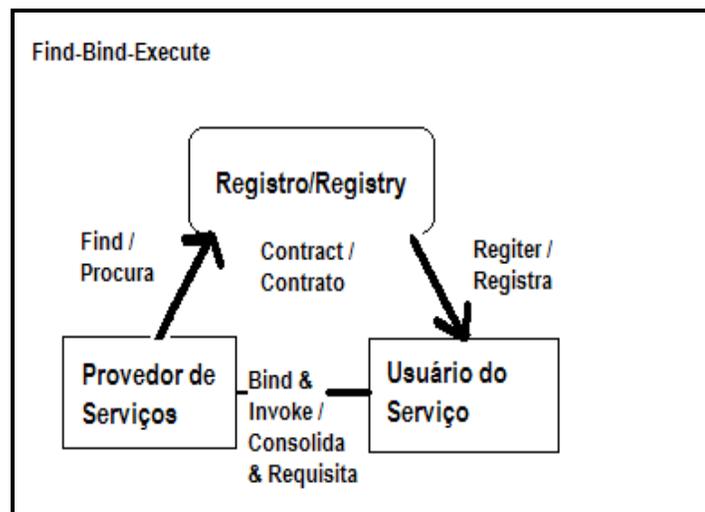
- **Integração com fornecedores** – Fortalecimento de todos os elos de sua cadeia de suprimentos;
- **Visibilidade das transações** – Monitoramento de dados referentes ao transporte e à distribuição, ao recebimento de mercadorias, embalagem e atendimento de pedidos, tudo automaticamente, com sensível redução de erros e de custos;
- **Controle de custos** – Automação de todos os processos do atacado para redução geral de gastos;
- **Serviços de valor agregado** – Estoque administrado pelo fornecedor, e faturamento detalhado.

#### 2.1.2.4 Plataforma

- **SOA Empresarial**

Service-oriented architecture (SOA), pode ser traduzido como arquitetura orientada a serviços, e é um estilo de arquitetura de software cujo princípio fundamental preconiza que as funcionalidades implementadas pelas aplicações devem ser disponibilizadas na forma de serviços. Frequentemente estes serviços são organizados através de um "barramento de serviços" (enterprise service bus, em inglês) que disponibiliza interfaces, ou contratos, acessíveis através de web services ou outra forma de comunicação entre aplicações. A arquitetura SOA é baseada nos princípios da computação distribuída e utiliza o paradigma request/reply para estabelecer a comunicação entre os sistemas clientes e os sistemas que implementam os serviços (Krafzig et alli, 2004 e Keen et alli, 2007).

A Arquitetura Orientada a Serviços pode ser bem representada a partir do seguinte processo, chamado de "find-bind-execute paradigm" o que significa aproximadamente paradigma de "procura-consolida-executa". Tal conceito é análogo a "Ciclo de Deming" aplicado aos serviços, que define o ciclo que envolve o planejamento, a execução, o monitoramento e a tomada de ação pró ativa para a melhoria da qualidade.



**Figura 3 - Esquema adaptado do paradigma "find-bind-execute". (Krafzig et alli, 2004)**

Tecnicamente falando, o processo preconiza que os provedores de serviços registrem informações em um registro central, com suas características, indicadores, e aspectos relevantes às tomadas de decisões. O registro é utilizado pelo cliente para determinar as características dos serviços necessários, e se o mesmo estiver disponível no registro central, como por exemplo por um catálogo de serviços, o cliente poderá utilizá-lo, sendo este oficializado através de um contrato que enderece este serviço.

- **SAP NetWeaver**

Uma plataforma abrangente de aplicativos e integração, o SAP NetWeaver trabalha com a sua infra-estrutura de TI já existente para permitir e administrar mudanças.

Com o SAP NetWeaver, você pode projetar, construir, implementar e executar novas estratégias empresariais de forma rápida e flexível, assim como executar novos processos e estratégias empresariais. Você também pode conduzir a inovação ao longo de toda a sua empresa ao combinar sistemas existentes, enquanto mantém uma estrutura de custos sustentável.

O SAP NetWeaver usa padrões da Internet, como HTTP, XML e serviços Web, permitindo abertura e interoperabilidade com Microsoft .NET e ambientes Java 2 Platform Enterprise Edition (J2EE), como a IBM WebSphere.

O SAP NetWeaver é a base técnica das soluções SAP Business Suite, soluções parceiras e aplicativos sob medida para consumidores. Ele também torna possível a arquitetura empresarial orientada a serviços, o modelo SAP para soluções empresariais orientadas a serviços.

O SAP NetWeaver inclui os seguintes componentes e ferramentas, como mostra a Figura 4.

SAP NetWeaver	
Componentes	Ferramentas
<ul style="list-style-type: none"> <li>• SAP NetWeaver Application Server</li> <li>• SAP NetWeaver Business Intelligence</li> <li>• SAP NetWeaver Portal</li> <li>• SAP NetWeaver Exchange Infrastructure</li> <li>• SAP NetWeaver Master Data Management</li> <li>• SAP NetWeaver Mobile</li> <li>• SAP Auto-ID Infrastructure</li> <li>• SAP Web Application Server</li> <li>• SAP NetWeaver Identity Management</li> </ul>	<ul style="list-style-type: none"> <li>• Adaptive Computing Controller</li> <li>• SAP NetWeaver Composition Environment</li> <li>• SAP NetWeaver Developer Studio</li> <li>• SAP NetWeaver Visual Composer</li> <li>• SAP Solution Manager</li> </ul>

**Figura 4 – Componentes e ferramentas do SAP NetWeaver.**

## **2.2 ERP (*Enterprise Resources Planning*)**

No final da década de 50, a tecnologia vigente era baseada nos gigantescos mainframes, que rodavam os primeiros sistemas de controle de estoque. Na época essa tecnologia que era bastante cara e lenta e poucas empresas aderiram, mesmo

assim já se observava uma maior agilidade nos processos automatizados do que nos processos manuais.

No início da década de 70, a expansão da economia e a maior divulgação computacional geraram um conceito denominado de MRP. Este conceito consistia em elaborar os sistemas em forma de módulos ou pacotes de software interligados, que possibilitavam automatizar processos relacionados ao uso de materiais pertinentes às mais diversas etapas de produção.

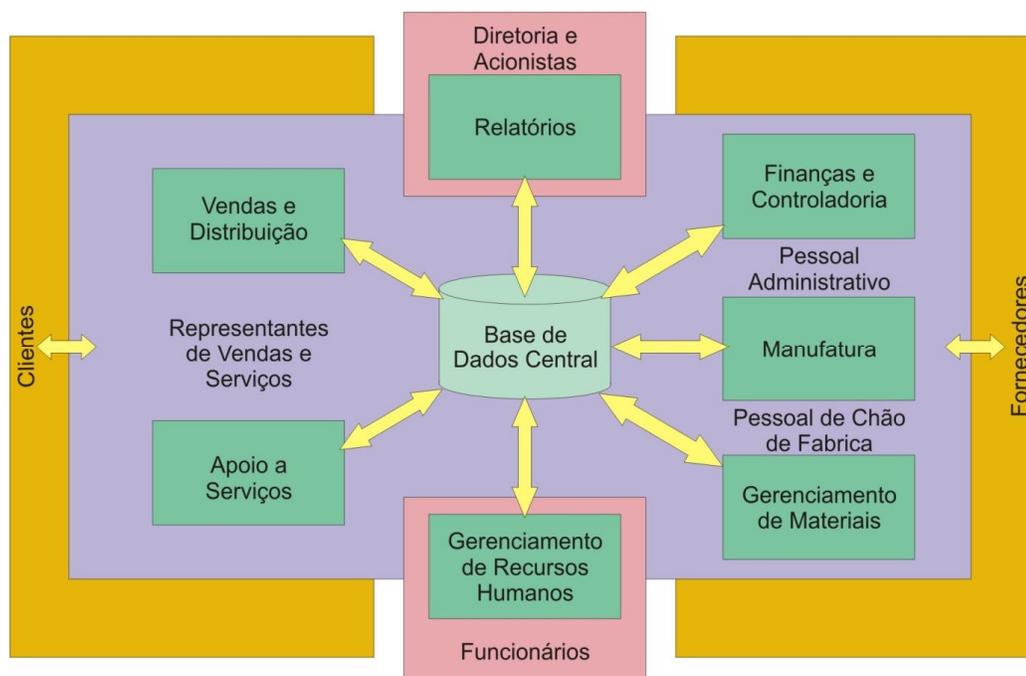
Uma década depois começaram a surgir as redes de computadores, que permitiram maior usabilidade e redução de custos na arquitetura computacional dentro das organizações. Paralelamente a esta revolução tecnológica, os processos de gerenciamento de produção e logística foram agregados aos sistemas de gestão. Dessa forma, o então denominado MRP, assumi à denominação de MRPII, controlando outros processos como a mão de obra e o chão de fábrica.

De forma geral; o MRPII já podia ser chamado de ERP, pela sua abrangência de controles e gerenciamento, mas somente em 1975 ele recebeu oficialmente esta denominação. Ao mesmo tempo, surgiu à empresa alemã SAP, que lançou o software R/2, entrando assim para a história, no que se diz respeito à tecnologia de gestão corporativa, isto é, os ERP's. Entretanto, esse conceito tornou mais difundido a partir dos anos 90, junto com a evolução das redes de computadores e com o aparecimento da arquitetura cliente/servidor. Outro motivo que levou o mercado a investir nessa tecnologia, foi o custo com equipamentos (estações de trabalho, servidores, aplicações, etc.), razão essa, que chamou a atenção dos empresários para as vantagens de se utilizar sistemas integrados.

Surgiram a partir de então, novos fornecedores e a diversidade dos ERP's aumentaram significativamente, já que diversas empresas adotaram o uso de sistemas integrados, influenciado também pelo do bug do milênio, em que os sistemas poderiam apresentar falhas por utilizarem a data do sistema com apenas dois dígitos.

### **2.2.1 Entendendo o ERP**

Para começarmos a entender o ERP, é importante sabermos que ele não possui nenhuma ligação direta com a sua sigla. Esqueça a palavra planejamento, ele não faz isso, e esqueça a palavra recurso, um termo descartável. Mas lembre-se da parte empresarial. Ele serve para integrar todos os departamentos e funções de uma companhia em um simples sistema de computador que pode servir a todas as necessidades particulares de cada uma das diferentes seções. A Figura 4 mostra o planejamento de recursos empresariais.



**Figura 5 – Planejamento de Recursos Empresariais – Fonte [www.ogerente.com.br](http://www.ogerente.com.br).**

É um grande desafio construir um único programa de software que supra as necessidades do departamento financeiro, assim como dos trabalhadores de recursos humanos e também do depósito e é isso que o ERP faz. Cada um desses departamentos, tipicamente, possui seu próprio sistema de computador, cada um aperfeiçoado para cada necessidade, para a forma de trabalho de cada departamento. O ERP combina todos eles juntos em um só programa de software integrado que trabalha com um banco de dados comum. Dessa forma, os vários departamentos podem mais facilmente dividir informações e se comunicar entre si.

Essa abordagem integradora pode dar um grande retorno financeiro se as companhias instalarem o software adequadamente. Pegue o pedido de um cliente como exemplo: tipicamente, quando um cliente faz um pedido, aquele pedido

começa uma jornada em papel, de um lugar para outro na empresa, sendo digitado e redigitado em vários computadores ao longo do caminho. Toda essa jornada causa atrasos e perdas de pedidos, e cada digitação, em um diferente sistema, é convidativa a erros. Ao mesmo tempo, nenhuma companhia sabe realmente em que estágio um pedido se encontra em um determinado momento porque não há como o departamento financeiro, por exemplo, entrar no computador do depósito para ver se o item foi embarcado. "Você terá que ligar para o depósito", é a resposta familiar dada aos frustrados consumidores.

### **2.2.2 Como o ERP pode melhorar a performance de uma empresa**

O ERP automatiza as tarefas envolvendo a performance de um processo, tal qual a finalização de um pedido, o qual envolve pegar o pedido de um cliente, enviá-lo e cobrá-lo. Com o ERP, quando um representante recebe o pedido de um cliente, ele ou ela, tem todas as informações necessárias para completá-lo. Todas as pessoas na empresa vêem o mesmo visor e têm acesso a um único banco de dados que guarda o novo pedido do cliente. Quando um departamento termina a sua parte em um pedido, este é enviado automaticamente para o próximo departamento via ERP. Para saber em que ponto está um pedido, em um determinado momento, é só checar no ERP. Com sorte, o processo se move como um raio dentro da organização, e os clientes recebem seus pedidos mais rapidamente que antes. O ERP consegue aplicar essa mesma magia à maioria dos processos empresariais, tal qual manter os funcionários informados sobre seus benefícios ou sobre decisões financeiras em geral.

Há três razões principais pelas quais firmas adotam o ERP:

**Para integrar dados financeiros:** Como o CEO tenta entender a performance geral da companhia, ele ou ela podem encontrar diferentes versões da verdade. O financeiro tem os seus números, vendas tem outra versão, e as diferentes unidades podem, cada uma, ter a sua própria versão do quanto eles podem contribuir para a receita. O ERP cria uma única versão da verdade que não pode ser questionada porque todos estão usando o mesmo sistema.

**Para uniformizar o processo de manufatura:** Empresas de manufatura, especialmente aquelas com um grande apetite por fusões e aquisições, geralmente descobrem que diferentes unidades da empresa usam diferentes métodos e sistemas de computador. Uniformizar esses processos, usando um único e integrado sistema de computador, pode economizar tempo, aumentar a produtividade e reduzir gastos.

**Para uniformizar as informações de RH:** Principalmente em firmas com múltiplas unidades de negócio, o departamento de Recursos Humanos talvez não tenha um único e simples método para acompanhar o tempo dos empregados e comunicá-los sobre seus benefícios e serviços. O ERP pode fazer isso.

O ERP é um sistema único que serve para integrar todos os departamentos e funções de uma companhia. Nesse sistema é possível programar em três linguagens: ABAP, ABAP orientado a objetos e JAVA.

O sistema ERP da SAP constitui o sistema de gestão empresarial mais difundido em nível mundial. O sistema força uma estrutura de gestão de negócios orientada a processos, onde a empresa é compelida a definir suas operações de forma muito precisa, enfatizando como as tarefas são realizadas, o que precisa ser feito, quem é responsável por elas e quais as informações necessárias para a sua realização (Gupta e Kohli, 2006). O sistema exige a estrita aderência a padrões ao longo de toda a empresa o que, associado à sua dimensão e complexidade, interpõe dificuldades de gestão muito significativas para a área de tecnologia de informação. Essas características, associadas ao fato de o sistema constituir um objeto relativamente uniforme ao longo das empresas, faz dele um interessante contexto para a pesquisa dos aspectos relacionados com as capacidades e atores do modelo de pesquisa.

## 2.3 ABAP

O ABAP é uma linguagem de programação dentro da categoria das linguagens de quarta geração desenvolvidas nos anos 80. Foi originalmente concebida como uma

linguagem para construção de relatórios para o SAP R/2. O objetivo original da linguagem era ser uma ferramenta de programação suficientemente simples para ser usada pelos utilizadores finais. No entanto, o ABAP é uma linguagem que necessita de conhecimentos avançados de programação e é principalmente utilizada pelos programadores da SAP que desenvolvem o software e por consultores informáticos que adaptam o software às necessidades dos clientes.

O ABAP é a principal linguagem utilizada no software cliente-servidor SAP R/3. Em 1999, com a versão R/3 4.5, a SAP lançou uma versão de ABAP que suporta programação orientada a objetos.

A versão mais recente da plataforma de desenvolvimento da SAP, o NetWeaver, suporta programação em ABAP e em Java.

Todos os programas de ABAP residem dentro da base de dados do SAP. Não são armazenados em arquivos separados como programas de Java ou de C++. Na base de dados todo o código de ABAP existe em dois formulários: código fonte, que pode ser visto e editado com as ferramentas da ABAP(*workbench*), e o código gerado, uma representação binária comparável com um *bytecode* do Java. Os programas de ABAP são executados sob um sistema de *runtime*, que é parte do SAP kernel. O sistema de *runtime* é responsável para processar indicações de ABAP, controla a lógica do fluxo das telas e de responder pelos eventos. Um componente chave do sistema de *runtime* em ABAP é a relação com base de dados, que converte indicações da base de dados independentes de ABAP nas indicações compreendidas pelo DBMS subjacente. A interface com a base de dados contém funcionalidades extra tais como a proteção de dados freqüentemente alcançados na memória local do servidor de aplicações. SAP tem três camadas diferentes: a camada de apresentação (GUI), a camada de aplicação (onde são executados os programas) e a camada onde todos os dados são armazenados e recuperados das condições conduzidas aos usuários finais e programadores através dos níveis de programação.

## 2.4 SAP/R3

O sistema SAP R/3 é composto por um conjunto de módulos de software integrados iterativamente. É um sistema abrangente e complexo. Pode tratar atividades desde a cadeia produtiva até relacionamentos com clientes da organização. As aplicações partilham dados de bases comuns aos módulos. As alterações feitas em bases de dados por determinado programa aplicativo não compromete a funcionalidade de outros módulos do sistema. A implementação do SAP R/3 em qualquer empresa requer a análise e a modificação de formas de realização de processos do negócio.

A arquitetura do sistema SAP R/3 é dividido em três camadas:

- **Camada de dados:** fornece dados ao sistema, incluindo tabelas, meta-dados e dados sobre as transações da empresa incluídas no sistema.
- **Camada de aplicação:** processa e faz intercâmbio de dados entre servidor de aplicação e servidor de base de dados para execução de ferramentas do software.
- **Camada de apresentação:** atua como interfaces de terminais e recursos de usuários, executando tarefas de entrada e saída de dados/informação.

A Figura 5 mostra a hierarquia da arquitetura do sistema SAP/R3.

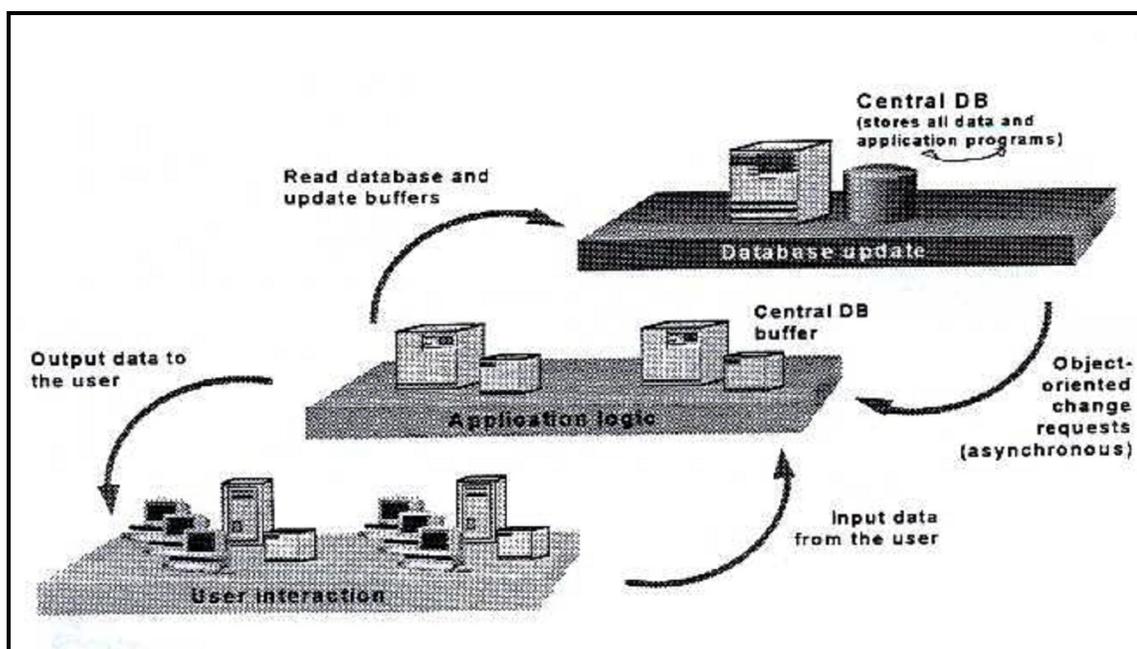


Figura 6 – Hierarquia da SAP R/3 – Fonte: Apostila de treinamento PAUT.

O Sistema R/3 é distribuído de acordo com os princípios de tecnologia de client/Server, onde seus componentes são divididos em uma hierarquia de computador de três-níveis.

Um computador central funciona como um servidor de banco de dados, isto é contém processos que constituem o serviço de banco de dados. Este computador também aloja o serviço de atualização do R/3 o qual é responsável por executar atualizações de banco de dados. O servidor de banco de dados pode ser unido a vários servidores de aplicação. Nos servidores de aplicação, é processada a lógica de aplicação atual.

Cada servidor de aplicação pode ser conectado a vários *front-ends* (por exemplo *workstations*, PCs) nos quais os usuários estão trabalhando. Todas as tarefas de apresentação são processadas nestas máquinas individuais.

Essa abordagem separa funcionalidades de programas dos servidores, colocando bases de dados em camada à parte. A divisão em camadas é feita com o objetivo de não sobrecarregar aplicações individuais de usuários. Assim, se for necessário pode-se acrescentar novos recursos tecnológicos (Ex.: servidores de arquivos, servidores de aplicações, terminais e outros recursos), sem causar maiores impactos no ambiente do sistema. Portanto, pode-se planejar a incorporação de recursos tecnológicos para cada uma das camadas sem levar em conta os efeitos de alterações sobre camada adjacente.

O sistema SAP R/3 é aberto e os programas das aplicações são concebidos em linguagem de programação ABAP/4. É uma plataforma comum que possui um conjunto integrado de ferramentas de desenvolvimento de quarta geração, que permite programar e adaptar aplicações necessárias à organização.

O sistema SAP R/3 apresenta pontos positivos e pontos negativos sobre sua utilização. Dentre os pontos positivos destacam se:

- **Nível de integração:** dados produzidos em qualquer ponto da cadeia são centralizados, o que elimina redundâncias e favorece a integridade da informação.
- **Modularidade e flexibilidade:** possui uma divisão em módulos aplicativos que permitem a implementação evolutiva de componentes.

- **Sistema aberto:** adota diferentes plataformas de hardware, software, tipos de banco de dados, sistemas operacionais. Pode ser flexibilizado por meio de parametrização e customização (modificação e desenvolvimento de programas aplicativos).
- **Apoio à gestão/decisão:** pode combinar informação interna e/ou externa e assim produzir resultados para a gestão do negócio, por meio de módulos de análise e apoio à gestão da empresa.

O sistema SAP R/3 também apresenta desvantagens, dificuldades e problemas. Por exemplo, sabe-se que um software proprietário desenvolvido para empresa específica é mais flexível que pacote *standard* comercializado por fornecedores, como o SAP R/3 e outros produtos similares. Adicionalmente, um software vertical implementado com base em atividades específicas de uma empresa pode atender mais facilmente aos requisitos particulares. Por exemplo, um software desenvolvido para atividades de uma indústria de calçados, usina de açúcar, criação de camarões ou laminação de aço. No entanto, componentes de pacote *standard* como o SAP R/3 ou outro ERP qualquer, muitas vezes não atendem as necessidades reais de uma organização, pois representam generalidades de um setor e não especificidades de uma empresa. Por isso, a implementação de pacote ERP exige adaptações por meio de parametrizações e customizações que, geralmente, envolve muito trabalho e custo excessivo para a instituição.

Além disso, o sistema SAP R/3 foi criado para atender a grandes corporações. Requer definições precisas de processos e percepções corretas de funções do negócio. É um software complexo, caro, de implementação de longo curso, em média de seis a 18 meses. Assim, tem sido pouco adaptável a empresas menores, principalmente, devido aos custos e adaptações de processos da empresa que precisam ser feitas na entrada do software (Santos 1999).

A implementação é função da complexidade e do número de módulos adquiridos. Exige trabalho em equipe e envolvimento de pessoas com diversas valências profissionais. Para ajudar na implementação de produtos ERP, firmas de consultoria especializadas são contratadas.

Os módulos do sistema SAP R/3 não são estanques ou de funcionamento individualizados, embora possam ser utilizados separadamente. Representam áreas

e tentam materializar a integração de unidades do negócio. Dependendo do interesse, necessidades e recursos, novos módulos podem ser implementados para expandir funções do software e assim atender a requisitos adicionais.

### **2.4.1 Função do SAP/R3**

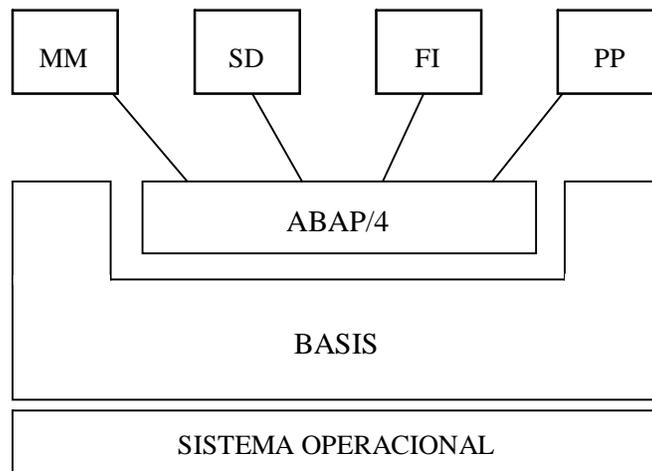
A função do R/3 é oferecer um conjunto integrado compacto, de aplicativos empresariais de grande escala. A Figura 6 mostra a arquitetura do SAP/R3.

O conjunto padrão de aplicativos integrados com cada sistema R/3 é o seguinte:

- PP (Production Planning - planejamento de produção)
- MM (Materials Management - gerenciamento de materiais)
- SD (Sales and Distribution - vendas e distribuição)
- FI (Financial Accounting - contabilidade financeira)
- CO (Controlling - controladoria)
- AM (Fixed Assets Management - gerenciamento de ativos fixos)
- PS (Project System - sistema de projeto)
- WF (Workflow - fluxo de trabalho)
- IS (Industry Solutions - soluções de indústria)
- HR (Human Resources - recursos humanos)
- PM (Plant Maintenance - manutenção das instalações)
- QM (Quality Management - gerenciamento da qualidade)

Estes aplicativos são chamados áreas funcionais, áreas de aplicativo ou, às vezes, módulos funcionais do R/3. Todos estes termos são sinônimos.

Tradicionalmente, as estruturas empresariais desenvolvem um conjunto de aplicativos de processamento de dados avaliando produtos individuais e comprando esses produtos separadamente de diversos fabricantes de software. Interfaces entre eles são então necessárias.



**Figura 7 – Arquitetura SAP R/3.**

O R/3 vem predefinindo com os aplicativos empresariais básicos necessários à maioria das grandes corporações. Esses aplicativos coexistem em um ambiente homogêneo. Eles são projetados para funcionar utilizando desde um único banco de dados e um conjunto (muito grande) de tabelas. Os tamanhos de banco de dados de produção atuais variam de 12 gigabytes a aproximadamente 3 terabytes. Cerca de 8.000 tabelas de banco de dados são distribuídas com o produto R/3 padrão.



**Figura 8 – Módulos do SAP R/3 – Fonte: Apostila de treinamento PAUT.**

## 2.4.2 Divisão dos Processos

Cada processo de trabalho é responsável por somente um tipo de *request*. Tipos de *request* incluem *dialog*, *background*, *update*, *enqueue* e *spool*.

O *dispatcher* distribui as tarefas adequadamente para os processos definidos. Porém, esta distinção só se refere aos papéis diferentes ou serviços especiais associados aos processos de trabalho através da estratégia de *dispatch*.

Como os processos de trabalho contêm aplicações escritas em ABAP/4, um grupo de processos que consiste em um *dispatcher* e um número definido de processos de trabalho, também é conhecido como um servidor de aplicações.

O servidor de mensagem permite os servidores de aplicações para trocar mensagens internas pequenas.

O servidor de acesso facilita comunicação entre R/3, R/2 e outros sistemas.

Em programas de aplicações, os processos de aplicações podem ser divididos em processos de atualização (V1, V2) e processos de background (B).

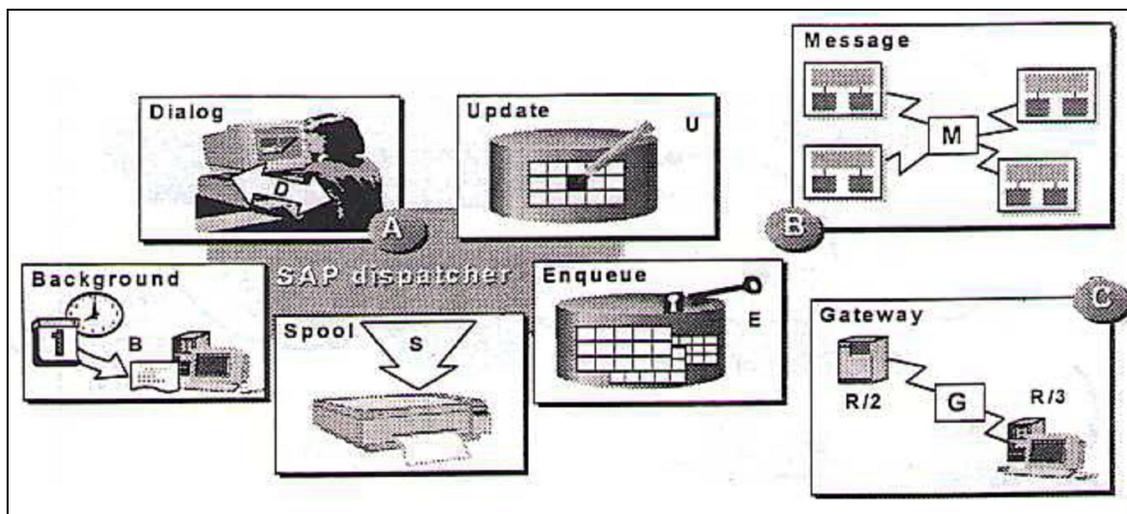


Figura 9 – Divisão de processo – Fonte: Apostila de treinamento PAUT.

## CAPÍTULO 3

### FERRAMENTAS

---

Neste capítulo serão apresentadas as ferramentas que foram utilizadas neste trabalho. Será feita uma descrição mais detalhada sobre JAVA, ABAP e ABAP Orientado a Objetos.

#### 3.1 JAVA

Java é uma linguagem de programação orientada a objeto desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Diferentemente das linguagens convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um "*bytecode*" que é executado por uma máquina virtual. A linguagem de programação Java é a linguagem convencional da Plataforma Java, mas não sua única linguagem.

Em 1991, na Sun Microsystems, foi iniciado o Green Project, o berço do Java. Os mentores do projeto eram Patrick Naughton, Mike Sheridan, e James Gosling. O objetivo do projeto não era a criação de uma nova linguagem de programação, mas antecipar e planejar a "próxima onda" do mundo digital. Acreditava que, em algum tempo, haveria uma convergência dos computadores com os equipamentos e eletrodomésticos usados pelas pessoas no seu dia-a-dia.

A velocidade dos acontecimentos seguintes foi assustadora, o número de usuários cresceu rapidamente, grandes fornecedores de tecnologia, como a IBM anunciaram suporte para a tecnologia Java.

Desde seu lançamento, em maio de 1995, a plataforma Java foi adotada mais rapidamente do que qualquer outra linguagem de programação na história da computação. Em 2003 Java atingiu a marca de 4 milhões de desenvolvedores em todo mundo. Java continuou crescendo e hoje é uma referência no mercado de desenvolvimento de software. Java tornou-se popular pelo seu uso na Internet e hoje possui seu ambiente de execução presente em web browsers, mainframes, SO's, celulares, palmtops e cartões inteligentes, entre outros.

### 3.2 ESTRUTURA ABAP

Davenport, Thomas (1998) divide os ERP em quatro blocos: financeiro, recursos humanos, operações e logística e vendas e marketing. Os módulos do bloco financeiro seriam divididos em contabilidade, contas a pagar, contas a receber e fluxo de caixa. Os módulos do bloco de recursos humanos seriam a folha de pagamento, gerenciamento de recursos humanos e controle de despesas de viagem. Os módulos de operações e logística seriam o gerenciamento de estoques, o MRP e o faturamento. Os módulos de vendas e marketing seriam processamento de pedidos e gerenciamento e planejamento de vendas.



**Figura 10 – Escopo funcional do SAP R/3 – Fonte: Mião 2007.**

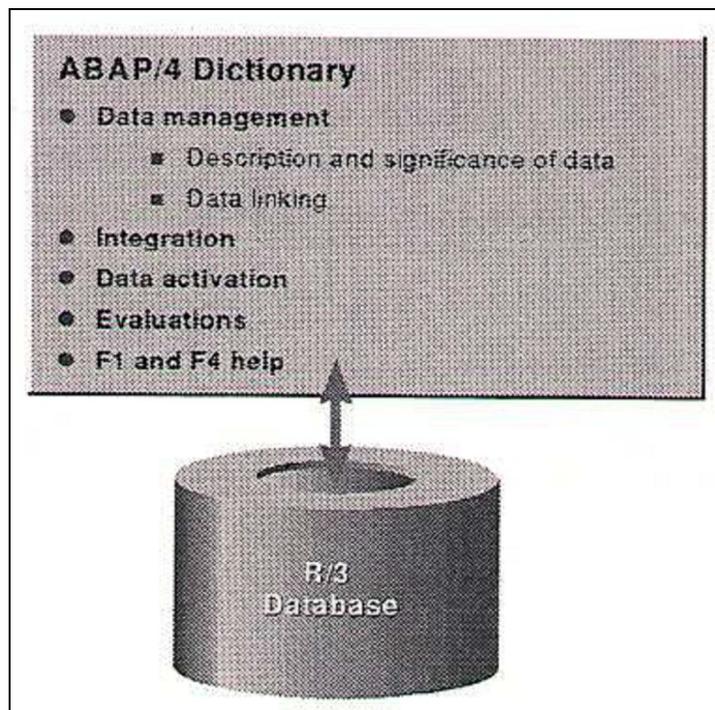
#### 3.2.1 Dicionário ABAP

O Dicionário ABAP é usado para criar e administrar definições de dados. Habilita todos os dados usados no sistema a serem descritos centralmente, sem redundância.

Todos os dados que entram no Dicionário ABAP estão disponíveis ao longo do sistema SAP.

Alterações entram em vigor em todos os componentes pertinentes ao sistema pelo mecanismo de ativação.

Pelo Sistema de Informação Repositório, o Dicionário pode ser analisado de ângulos diferentes. A figura 11 mostra a estrutura do Dicionário ABAP.



**Figura 11 – Dicionário ABAP – Fonte: Apostila de treinamento PAUT.**

O Dicionário ABAP é ativamente integrado no ABAP Development Workbench. Cada mudança é efetuada imediatamente nos programas e telas de ABAP.

Linha após linha, este é o modo interpretativo de operação do Sistema R/3. O intérprete não trabalha diretamente com o código original de um programa ou tela de ABAP. Ao invés, interpreta uma representação interna gerada.

Quando são feitas mudanças no Dicionário ABAP, uma geração automática subsequente destas representações garantem que o intérprete trabalhe com os dados em dia.

Toda a informação que é importante para o ambiente de operação, como tamanho de campo e tipo de campo, é tirada do Dicionário ABAP. Os objetos básicos para definir dados no Dicionário ABAP são: tabelas, domínios e elementos de dados.

Domínios são usados para a definição técnica (por exemplo: tipo de campo, tamanho de campo) de um campo de tabela, e elementos de dados para a definição semântica (por exemplo: descrição pequena).

Campos não são objetos independentes. Eles são dependentes em tabelas e podem ser mantidos então só dentro de uma tabela; eles se referem a elementos de dados. Cada elemento de dados é dependente na existência de um domínio. Domínios podem ser definidos sem referências adicionais para outros objetos do Dicionário.

O Dicionário ABAP emprega dois níveis de conceito de domínio para definição de dados, com domínios técnicos (identificados como domínios no Sistema SAP) e domínios semânticos (identificados como elementos de dados no Sistema SAP).

O domínio é o objeto central para descrever as características técnicas de um atributo de um objeto empresarial. Descreve o alcance de valor de um campo. Isto é, determinando e especificando características formais como formato externo, duração e assim por diante. Além disso, o valor de alcance pode ser restringido especificando valores fixos ou um valor da tabela.

O elemento de dados é o domínio semântico. Descreve um papel de um domínio exatamente em um contexto empresarial particular para seus campos dependentes.

### **3.2.2 Tabelas Transparentes**

Uma tabela transparente é criada automaticamente no banco de dados quando é ativado no Dicionário ABAP. A descrição do banco de dados independente da tabela no Dicionário ABAP é traduzida ao idioma do sistema de banco de dados usado.

A tabela de banco de dados tem o mesmo nome como a tabela no Dicionário ABAP. Os campos também têm os mesmos nomes no banco de dados e o Dicionário ABAP. Os dados digitados no Dicionário ABAP é transferido aos tipos de dados correspondentes do sistema de banco de dados.

A sucessão de campo no Dicionário ABAP pode diferir da sucessão de campo no banco de dados. Isto torna possível inserir campos novos sem converter a tabela. Quando um campo novo é somado, o ajuste é feito mudando o catálogo de banco de dados (ALTERE TABELA). O campo novo é juntado à tabela de banco de dados, independente da posição do campo novo no Dicionário ABAP.

Quando uma tabela transparente é ativada no Dicionário ABAP, a tabela é criada automaticamente no banco de dados pelo utilitário de DB. Só complexas operações

de banco de dados, como conversões de tabelas, precisam acontecer diretamente pelo utilitário de DB. Operações simples, como criar objetos, acontecem automaticamente pela ativação dos objetos no Dicionário ABAP.

### **3.2.3 Estruturas**

No Dicionário ABAP a estrutura produz dados quando cálculos são levados a cabo dentro de programas ou quando dados são transferidos entre programas podendo também ser definidos globalmente. Tal definição é conhecida como uma estrutura.

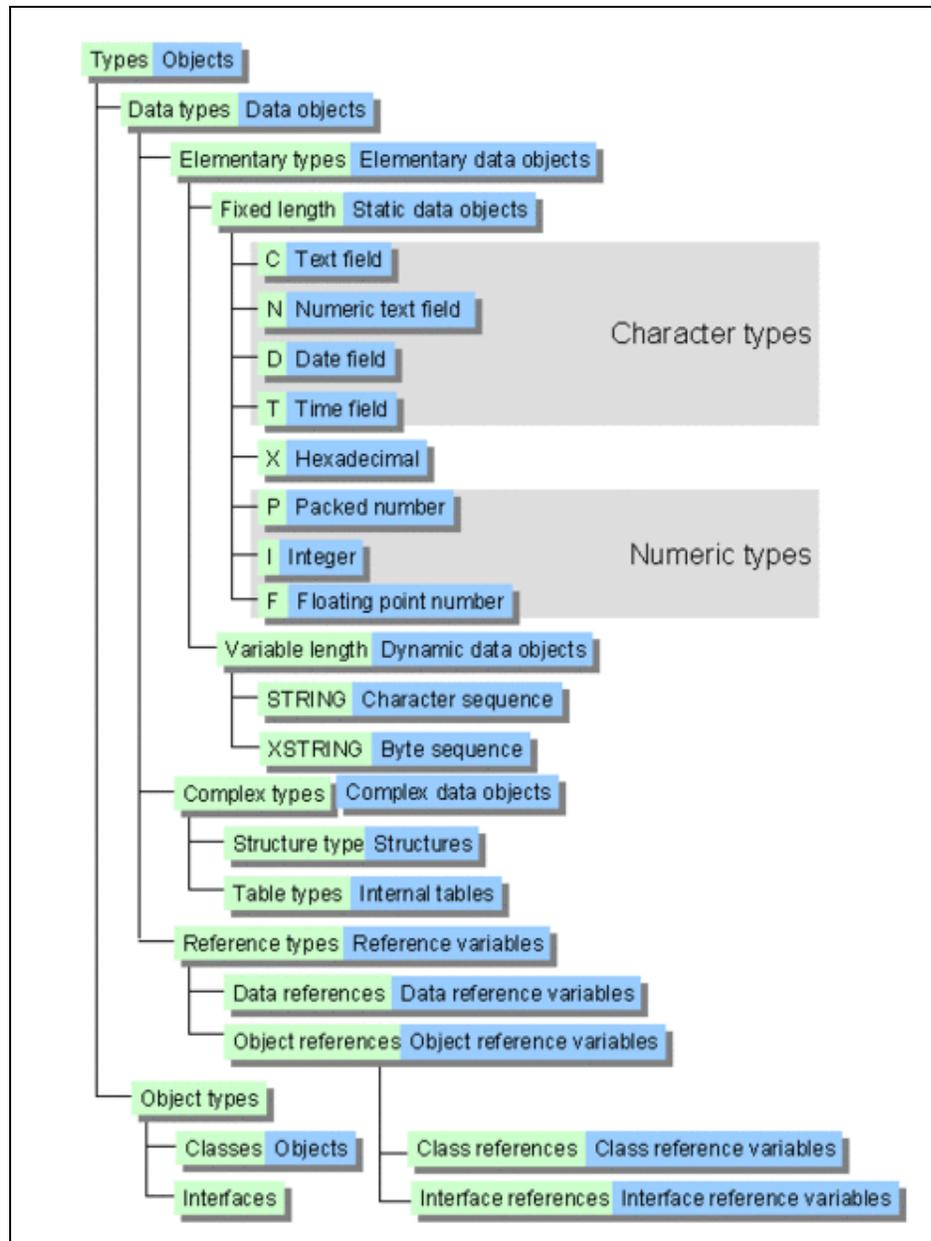
Estruturas são definidas da mesma maneira que tabelas. Porém, uma estrutura não corresponde a um objeto no banco de dados. Uma estrutura pode ser usada como uma referência em programas de ABAP.

Estruturas são definidas e são ativadas no Dicionário ABAP. Eles podem responder a programas ABAP como tabelas nas instruções de TABELAS.

Em tabelas transparentes, dados podem ser salvos no banco de dados, mas estruturas só contêm dados temporariamente durante a execução de um programa.

Estruturas são definidas desde que sejam usadas mais de uma vez centralmente, elas também podem ser mudadas centralmente. Estas mudanças são efetuadas então a todos pontos pertinentes pelo Dicionário ABAP ativo.

A figura 12 mostra o escopo de variáveis do sistema SAP/R3.



**Figura 12 – Escopo funcional do SAP R/3 – Fonte: Mião 2007.**

A Tabela 1 mostra algumas transações do ABAP.

SE09/SE10	Transport Organizer - Utilizada para verificarmos as ordens.
SE11	Abap Dictionary - Utilizada para exibir/modificar/criar tabelas, visões, elementos de dados, domínio, ajudas de pesquisa e objetos de bloqueio.
SE12	Abap Dictionary - Utilizada para exibir tabelas, visões, elementos de dados, domínio,

	ajudas de pesquisa e objetos de bloqueio.
SE13	Abap Dictionary - Utilizada para exibir/modificar opções técnicas.
SE14	Abap/4 Dictionary: utilitário DB - Utilizada para eliminar objetos, bem como para ativar/ajustar as tabelas, visões, etc, conservando ou eliminado seus dados.
SE15	Sistema Info Repository.
SE16	Data Browser - Utilizada para checar o conteúdo de tabelas.
SE17	Exibição geral de tabela.
SE18	Add ins: Definições.
SE19	Add ins: Implementações – BADI.
SE24	Class Builder: 1ª tela.
SE29	Exibir Pacote de Aplicações.
SE30	Análise do tempo de execução: 1ª tela - Utilizada para analisar o tempo de execução de programas, transações e módulos de funções.
SE32	Atualização de elementos de texto (de programas e classes).
SE33	Contexto: 1ª tela.
SE35	Módulos de diálogo ABAP/4.
SE36	Bancos Lógicos de Dados.
SE37	Function Builder: 1ª tela - Utilizada para exibir/modificar/criar módulos de funções.
SE38	Edito ABAP: 1ª tela - Utilizada para exibir/modificar/criar programas (texto fonte, variantes, características, documentação, elementos de texto).
SE39	Abap editor de tela dividida: Utilizada para comparar programas.
SE40	Menu Painter atualização de regras: 1ª tela.
SE41	Menu Painter: 1ª tela.
SE43	Atualização de Menu de Área - Utilizada para inclusão/alteração de menus (alteração do menu Itaú, por exemplo).

SE48	Análise de Programas: hierarquia de chamadas: 1ª tela.
SE49	Análise de Programas: manipulação tabelas.
SE51	Screen Painter: 1ª tela - Utilizada para exibir, modifica e criar lógica de processamento, lista de elementos, características, layout de tela.
SE52	Chamada parametrização de screenpainter.
SE54	Geração visão de tabelas: 1ª tela tabela/visão - Utilizada para criar/eliminar diálogos de atualização para tabelas.
SE55	Chamada interna atualizar visão tabelas.
SE56	Chamada interna exibir visão tabelas.
SE57	Chamada interna eliminar visão tabelas.
SE61	Atualização de documentos: 1ª tela.
SE62	Utilitários setor industrial.
SE63	Tradução: 1ª tela.
SE71	Form Painter (Formulário SapScript): 1ª tela - Utilizada para criar/modificar/exibir formulários e seus sub-objetos (cabeçalho, páginas, janela, etc. ).
SE72	Estilo(Estilos SAPScript): requisito.
SE73	Atualizar família de caracteres SAPscript: 1ª tela.
SE74	Conversão de formato SAPscript.
SE75	Opções SAPscript.
SE76	SAPscript: tradução de formulários.
SE77	SAPscript: tradução de estilos.
SE78	SAPscript: administração map bit (Inclui figuras no SAP).
SE80	Object Navigator - Utilizado para navegar dentre os objetos de um programa tipo MODULE POOL.
SE81	Hierarquia de aplicações: Exibir.

SE83	Exibir Reuse Library.
SE84	Sistema Info Repository.
SE85	ABAP/4 dictionary sistema info.
SE86	ABAP/4 sistema info.
SE87	Sistema info Data Modeler.
SE88	Coordenação desenv. Sistema info.
SE89	Atualização de árvore sistema info.
SE8I	Pool listas sistema info Repository.
SE90	Sistema info modelo de processo.
SE91	Atualização de mensagem: 1ª tela - Utilizada para exibir/modificar/criar mensagens.
SE92	Sistema info modelo de processo.
SE93	Atualização de transação - Utilizada para exibir/modificar/criar códigos de transação.
BIBS	Exemplos de layout de interfaces.
LIBS	Layout de listas: síntese - Utilizada para verificar exemplos de relatórios.
SHDB	Registrador de gravação: síntese de gravação - Utilizada para gravar/processar mapeamentos.
SM30	Atualização de visão de tabelas: 1ª tela - Utilizada para exibir/modificar/criar conteúdo de tabelas.
SM35	Batch input: síntese de pastas - Utilizada para analisar/processar pastas.
SM37	Síntese de jobs em background.
SQ00	SAP Query: executar queries.
SQ01	SAP Query: atualizar queries.
SQ02	SAP Query: atualizar InfoSet.
SQ03	SAP Query: atualizar grupos usuários.

SQ07	ABAP Query: ajuste idioma.
SQ09	SAP Query: atualizar arquivo ampliado.
SQ10	SAP Query: administração de funções.
SQCIT	Atualização templates CI.
SQ_LEGEND	Plano de seqüências: seleção elementos gráficos.
SQLR	Interpretador de trace SQL.
SQVI	QuickViewer
ZFTP	Abap transferência de arquivos Servidor X PC - Utilizada para executar transferência de arquivos (upload/download).
SA38	ABAP/4: execução de programas - Utilizada para executar programas tipo 1.
PE03	Características: 1ª tela - Utilizada para exibir/modificar Features.
PO10	Atualizar unidade organizacional.
PP02	Atualizar dados de plano: modo especialista.
PP03	Atualizar dados de plano: executar medidas.
PM01	INFOTIPO PA/PB.
PPCI	INFOTIPO PD.
OOVK	Atualiza ligações.
OOIT	Atualiza customização para INFOTIPOS.

**Tabela 1 – Tabela de transações do SAP.**

A Tabela 2 mostra alguns comandos em ABAP:

Write	Para escrever.
Uline	Aciona uma linha horizontal.
Skip	Saltar linha.

Under	em baixo "a baixo".
occurs	Controle de registro na tabela. Delimita as linhas da tabela, para melhorar a performance do programa.
E000	Esta expressão é usada para mensagem de erro.
S000	Esta expressão é usada para mensagem de rodapé.
I000	Esta expressão é usada para mensagem de informação.
W000	Esta expressão é usada para mensagem de aviso, congela a imagem, aguarda retorno do usuário.
Vline	Vertical " insere uma linha vertical".
"   "	Igual ao V_line.
Clear V_Var1	Limpa" zera uma variável.
[ ]	Compara o conteúdo total da tabela.
Select	Selecionar. "Varre o banco de dados".
New-Page	Abrir um nova pagina.
Standard Page Heading	Desabilita cabeçalho padrão.
Top - Of - Page	Para desenvolver um novo cabeçalho.
End-of-page	Encerra a página "Rodapé".
Format Color 1	Insere uma nova cor que é selecionada pelo numero.
Format Intensified off	Desabilita o formato padrão da letra.
Format Intensified on	Ativa o formato padrão.
Data	Dados.
Like	Como.
Move	Mover.
Move Correspond	Move apenas os dados que corresponde a seleção.

Is Initial	Comparação do vazio. Testa se a tabela interna esta vazia no IF
Refresh	Limpa todo o conteúdo da tabela interna.
Clear	Apaga o cabeçalho (limpa) de uma tabela interna.
Parameter	"são os texts boxes" --> entrada de dados.
Default	Padrão.
At First	Primeiro registro da tabela.
At New	Primeiro registro da quebra.
At Last	Ultimo registro da quebra.
At And Of	Ultimo registro da quebra.
Order	Ordenar.
Append	Gravar um registro após a execução.
ON CHANGE	Gravar o cabeçalho de uma tabela interna
Describe	Mostra o total do registro na tabela.
Condense	Comando que serve para juntar as palavras de forma ordenada.
Using	Usar.
OkCode	Como se fosse Enter - "Isto um codigo interno do abap".
Ws - Download	Envia "Arquivo".
Upload	Importa "Arquivo".
Loop	Comando execução.
Form	Sub-rotinas.
At line selection	Comando utilizado no relatório, quando uma linha é selecionada.
At user command	Controla os botões que forem adicionados ao código.
Call Transaction	Possibilita o processamento de uma tabela BDC.

Bach Input	Método de entrada de dados.
------------	-----------------------------

**Tabela 2 – Tabela de comando em ABAP.**

### 3.3 ABAP ORIENTADO A OBJETO

Orientação a objeto, ou para ser mais preciso, programação orientada a objeto, é um método de resolução de problemas no qual a solução do software reflete objetos do mundo real. Uma introdução abrangente à orientação a objeto como um todo iria muito além dos limites da introdução aos objetos ABAP. Esta documentação introduz uma seleção de termos que são usados universalmente na orientação a objeto e também ocorre em objetos ABAP. Em sessões subsequentes, continua-se a discutir em mais detalhes como esses termos são usados em objetos ABAP. O fim desta seção contém uma lista de leitura posterior, com uma seleção de títulos sobre orientação a objeto.

- **Objetos:** Um objeto é seção de código que contém dados e fornece serviços. Os dados formam os atributos do objeto. Os serviços são conhecidos como métodos. Tipicamente, métodos operam em dados privados, que é apenas visível para os métodos do objeto. Logo os atributos de um objeto não podem ser modificados diretamente pelo usuário, mas apenas pelos métodos do objeto. Isso garante a consistência interna do objeto.
- **Classes:** Descrevem objetos. De um ponto de vista técnico, objetos são instâncias em tempo de execução de uma classe. Em teoria, pode-se criar qualquer número de objetos baseados em uma única classe. Cada instância de uma classe tem uma identidade única e seu próprio conjunto de valores para seus atributos.
- **Referências a Objetos:** Em um programa, identifica-se e endereça objetos usando referências únicas a objetos. Referências a objetos permitem que acesse os atributos e métodos de um objeto. Em programação orientada a objeto, objetos geralmente têm as seguintes propriedades:

- ✓ **Encapsulamento:** Objetos restringem a visibilidade de seus recursos (atributos e métodos) aos outros usuários. Todo objeto tem uma interface, que determina como os outros objetos podem interagir com ele. A implementação do objeto é encapsulada, isso é, invisível for a do próprio objeto.
- ✓ **Polimorfismo:** Métodos idênticos (mesmo nome) se comportam diferentemente em diferentes classes. Orientação orientada a objeto contém construções chamadas interfaces. Elas permitem que enderece métodos com mesmo nome em diferentes objetos. Apesar de a forma de endereçamento é sempre a mesma, a implementação do método é específica a uma particular classe.
- ✓ **Herança:** Pode-se usar uma classe existente para derivar uma classe nova. Classes derivadas herdam os dados e métodos da superclasse. No entanto, eles podem substituir métodos existentes, e também adicionar novos.

Abaixo estão algumas vantagens da programação orientada a objeto:

- Sistemas de software complexos se tornam mais fáceis de serem compreendidos, já que a estrutura orientada a objeto fornece uma representação muito mais próxima da realidade do que as outras técnicas de programação;
- Em um sistema orientado a objeto bem desenvolvido, é possível implementar mudanças em nível de classe, sem ter que realizar alterações em outros pontos do sistema. Isto reduz a quantidade total de manutenção requerida;
- Através do polimorfismo e herança, a programação orientada a objeto permite a reutilização de componentes individuais;
- Em um sistema orientado a objeto, a quantidade de trabalho de manutenção e revisão envolvido é reduzida, já que muitos problemas podem ser detectados e corrigidos em fase de projeto;

Para atingir estes objetivos requer:

-Linguagens de programação orientada a objetos. Técnicas de programação orientadas a objeto não necessariamente dependem em linguagens de programação orientada a objeto. No entanto, a eficiência da programação orientada a objeto

depende diretamente de como as técnicas de programação orientada a objetos são implementadas no sistema kernel;

- Ferramentas de orientação a objeto permitem que se crie programas orientados a objetos em linguagem orientada a objetos. Eles permitem que se modele e guarde objetos e relações entre eles;

- Modelagem orientada a objeto de um sistema de software é o mais importante, mais demorado, e o requerimento mais difícil para alcançar acima dos objetivos. Design orientado a objeto envolve mais do que apenas programação orientada a objeto, e fornece vantagens lógicas que são independentes da verdadeira implementação.

Objetos ABAP são um novo conceito do R/3 Release 4.0. O termo tem dois conceitos. No primeiro, os objetos compõem todo o ambiente de tempo de execução ABAP. No segundo, ele representa a extensão orientada a objeto da linguagem ABAP.

### **3.3.1 O ambiente de tempo de execução**

O novo nome ABAP Objects para todo o ambiente de tempo de execução é uma indicação do modo que o SAP R3 tem, por algum tempo, movendo em direção a orientação a objeto, e de seu compromisso em perseguir esta linha. O ABAP Workbench permite que você crie objetos R/3 Repository, como programas, objetos de autorização, objetos de travamento, objetos de customizing, e muitos outros. Usando módulos de funções, você pode encapsular funções em programas separados com uma interface definida. O Business Object Repository (BOR) permite que você crie SAP Business Objects para uso interno e externo (DCOM/CORBA). Até agora, as técnicas de orientação a objeto têm sido usadas exclusivamente em projeto de sistemas, e ainda não foram suportadas pela linguagem ABAP.

### **3.3.2 A extensão da linguagem orientada a objeto**



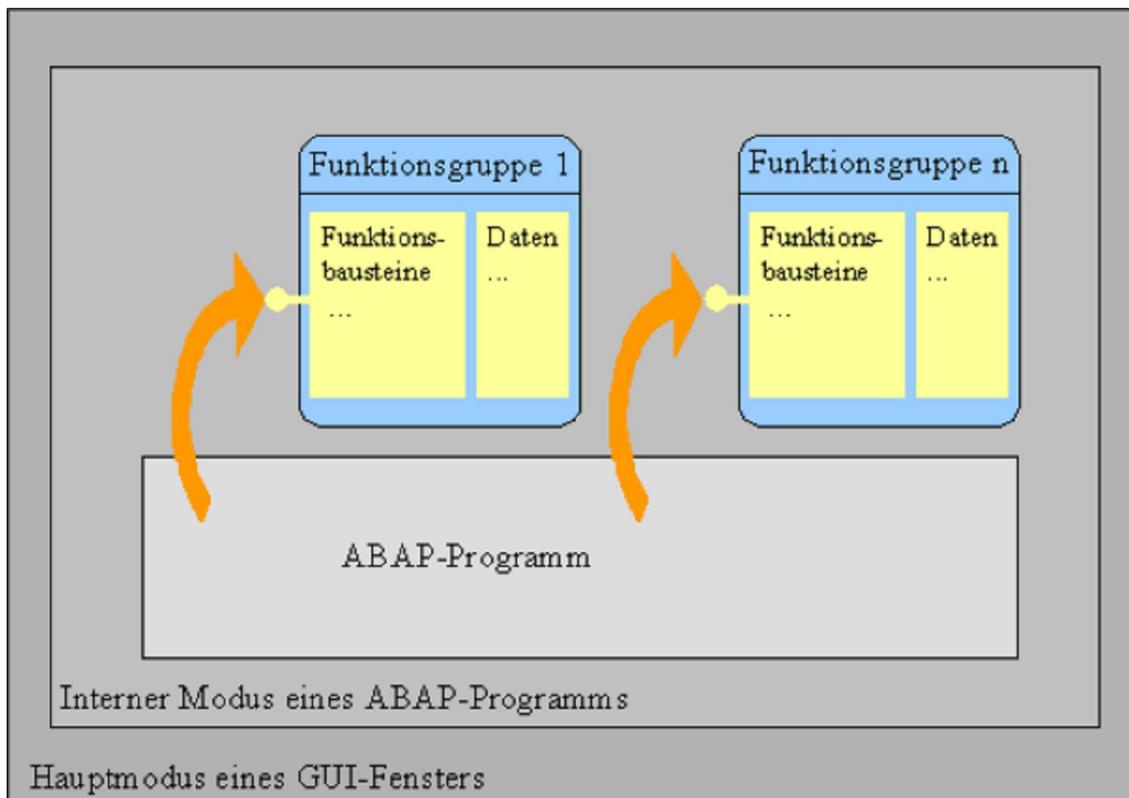
### Figura 13 – Tela de definição de classes no SAP.

#### 3.3.3 De Grupos de Funções a Objetos

No centro de qualquer modelo orientado a objeto estão os objetos, que contém atributos (dados) e métodos (funções). Objetos deveriam permitir programadores a mapear um problema real e soluções propostas do software em uma base um-por-um. Objetos típicos no ambiente de negócios são, por exemplo, 'consumidor', 'ordem', ou 'pedido'. Do release 3.1 em diante, o Business Object Repository (BOR) contém exemplos de tais objetos. O objeto modelo dos objetos ABAP, a extensão orientada a objeto do ABAP, são compatíveis com o objeto modelo do BOR.

Antes do R/3 Release 4.0, o que existia mais próximo de objetos no ABAP eram módulos de funções e grupos de funções. Suponha que tenhamos um grupo de funções para processar ordens. Os atributos de uma ordem correspondem aos dados globais do grupo de função, enquanto os módulos de funções individuais representam ações que manipulam aqueles dados (métodos). Isto significa que a verdadeira ordem dos dados é encapsulada no grupo de funções. Deste modo, os módulos de funções podem garantir que os dados são consistentes.

Quando executa um programa ABAP, o sistema inicia uma nova sessão interna. A sessão interna tem uma área de memória que contém o programa ABAP e os dados associados. Quando você chama um módulo de função, uma instancia de um grupo de funções mais seus dados, é carregado na memória da sessão interna. Um programa ABAP pode carregar várias instancias através da chamada de módulos de funções de diferentes grupos de funções. A figura 14 mostra a criação de uma nova instancia.



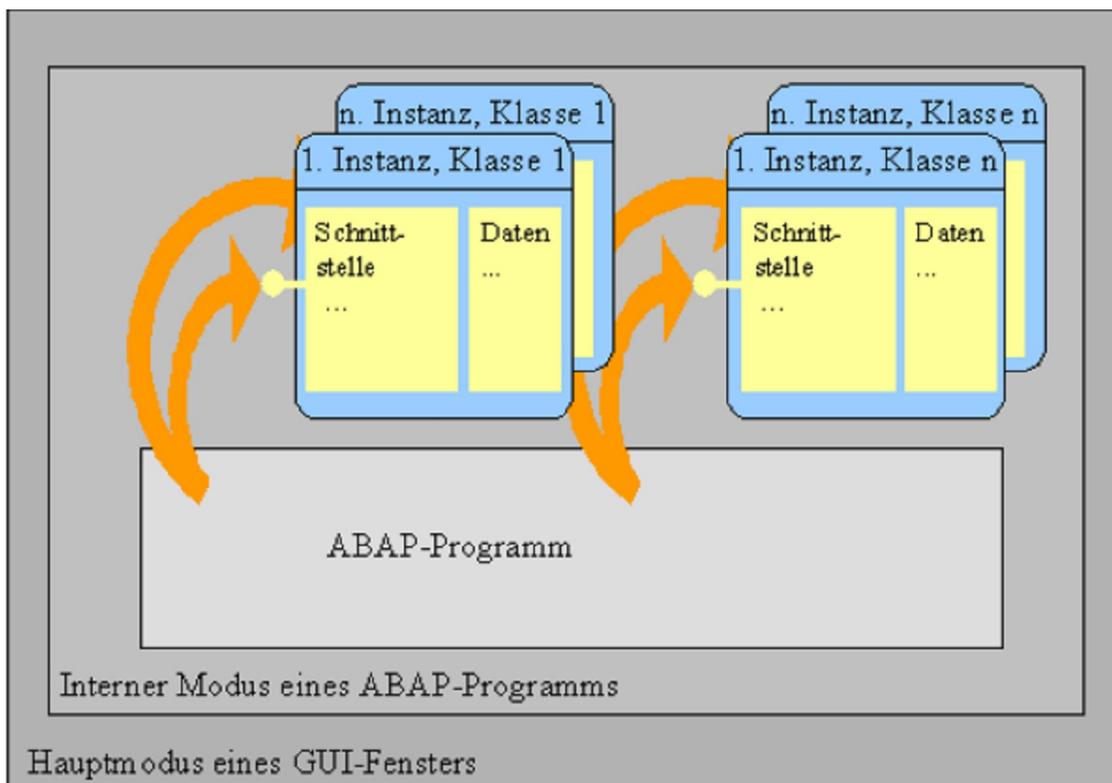
**Figura 14 – Criação de uma nova sessão interna – Fonte: Aspen Procwork.**

A instância de um grupo de funções na área de memória de uma sessão interna quase representa um objeto no senso de orientação a objeto. Quando chama-se um modulo de função, o programa chamador usa a instancia de um grupo de funções, baseada na descrição usada no Function Builder. O programa não pode acessar os dados no grupo de funções diretamente, mas apenas através do modulo de função. Os módulos de funções e seus parâmetros são a interface entre o grupo de funções e o usuário.

A diferença principal entre a real orientação a objeto e grupo de funções é que apesar de um programa poder trabalhar com instâncias de diversos grupos de funções simultaneamente, ele não pode trabalhar com diversas instancias do mesmo grupo de funções. Analisando que um programa necessite vários contadores independentes, ou processar várias ordens ao mesmo tempo. Neste caso, teria que adaptar o grupo de funções a incluir administração de instancia, por exemplo, usando números para diferenciá-las.

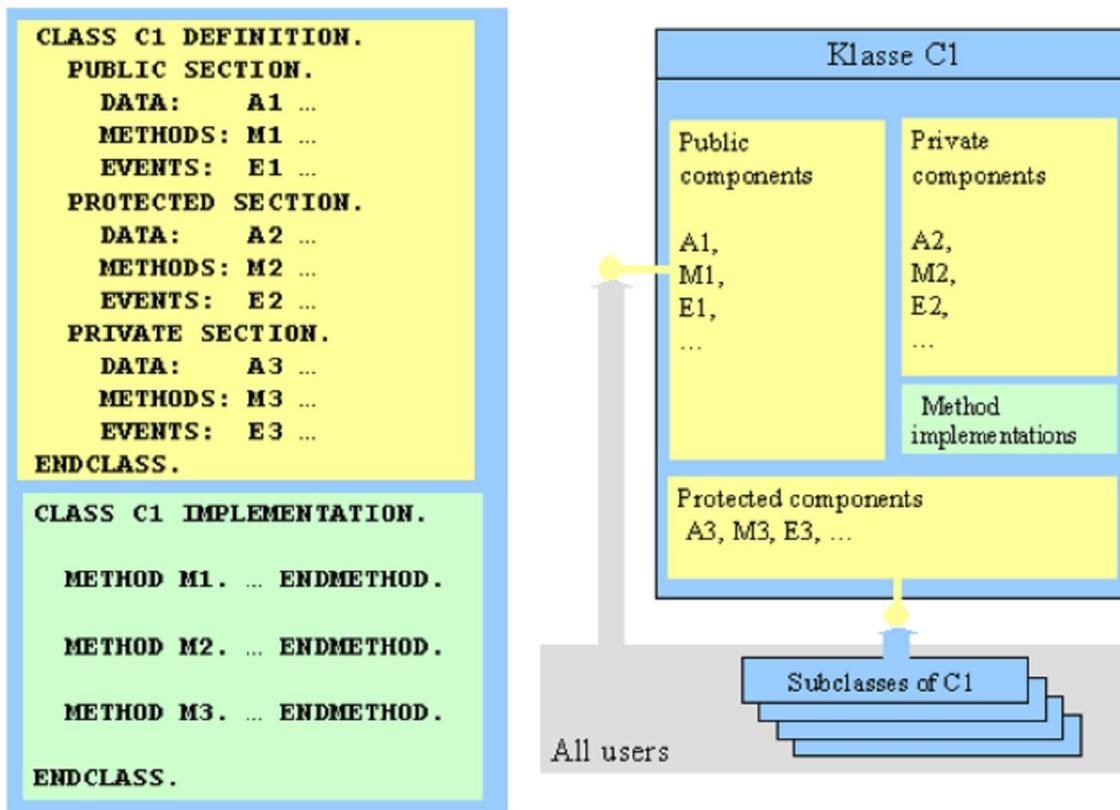
Na prática, isto é muito estranho. Conseqüentemente, os dados são usualmente guardados no programa chamador, e os módulos de funções são chamados para trabalhar com ele (programação estruturada). Um problema é, por exemplo, que todos os usuários do módulo de função devem utilizar os mesmos dados assim como o grupo de funções. Modificando a estrutura interna dos dados de um grupo de funções afeta vários usuários, e é freqüentemente difícil de prever implicações. O único modo de evitar é depender muito de interfaces e de uma técnica que garante que as estruturas internas das instâncias vão permanecer ocultas, permitindo que as modifique mais tarde sem causar problemas.

Este requisito é atingido por orientação a objeto. Objetos ABAP permitem que defina dados e funções em classes ao invés de grupo de funções. Usando classes, um programa ABAP pode trabalhar com qualquer número de instancias (objetos) baseados na mesma classe.



**Figura 15 – Criação de mais do que uma sessão interna – Fonte: Aspen Procwork.**

Ao invés de carregar uma única instância de um grupo de funções dentro de uma memória implícita quando um módulo é chamado, o programa ABAP pode agora gerar as instancias de classes explicitamente usando o novo comando ABAP CREATE OBJECT. As instancias individuais representam objetos únicos. Você endereça estes usando referências a objetos. As referências aos objetos permitem que o programa ABAP acesse as interfaces das instancias.



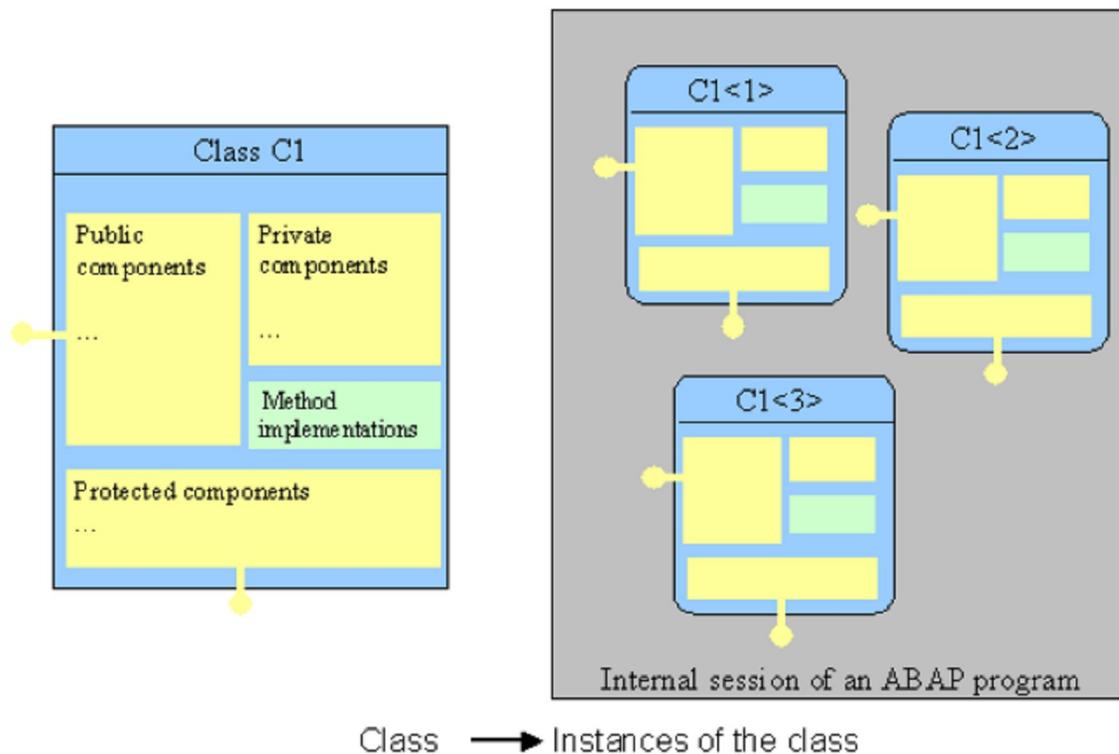
**Figura 16 – Implementação de uma classe local – Fonte: Aspen Procwork.**

A figura a esquerda demonstra as partes de declaração e implementação de uma classe local C1. A figura a direita ilustra a estrutura da classe com os componentes em suas respectivas áreas de visibilidade, e a implementação de métodos.

Os componentes públicos da classe formam a interface entre a classe e seus usuários. Os componentes protegidos são uma interface para que as subclasses de C1 acessem atributos da superclasse. Os componentes privados não estão visíveis externamente, e estão completamente encapsulados dentro da classe. Os métodos, na parte de implementação, tem acesso irrestrito a todos os componentes da classe.

```
CLASS C_COUNTER DEFINITION.  
  
PUBLIC SECTION.  
METHODS: SET_COUNTER IMPORTING VALUE (SET_VALUE) TYPE I,  
         INCREMENT_COUNTER,  
         GET_COUNTER EXPORTING VALUE (GET_VALUE) TYPE I.  
  
PRIVATE SECTION.  
DATA COUNT TYPE I.  
  
ENDCLASS.  
  
CLASS C_COUNTER IMPLEMENTATION.  
  
METHOD SET_COUNTER.  
COUNT = SET_VALUE.  
ENDMETHOD.  
  
METHOD INCREMENT_COUNTER.  
ADD 1 TO COUNT.  
ENDMETHOD.  
  
METHOD GET_COUNTER.  
GET_VALUE = COUNT.  
ENDMETHOD.  
  
ENDCLASS.
```

A classe C\_COUNTER contém três métodos públicos - SET\_COUNTER, INCREMENT\_COUNTER, e GET\_COUNTER. Cada um destes trabalham com o campo inteiro privado COUNT. Dois dos métodos tem parâmetros de entrada e saída. Estes formam a interface de dados da classe. O campo COUNT não é visível externamente.



**Figura 17 – Classes e suas instancias – Fonte: Aspen Procwork.**

A ilustração acima mostra uma classe C1 a esquerda, com suas instancias representadas na sessão interna de um programa ABAP a direita. Para distinguí-los de classes, instancias são desenhadas com cantos redondos. Os nomes das instancias acima usam a mesma notação como é usada para variáveis de referência no Debugger.

### 3.3.3.1 Declarando Métodos

Você pode declarar métodos na parte de declaração de uma classe ou em uma interface. Para declarar métodos de instancia (pertencentes ao objeto), use o seguinte comando:

```
METHODS <meth> IMPORTING.. [VALUE(<i>)] TYPE type [OPTIONAL]..
EXPORTING.. [VALUE(<e>)] TYPE type [OPTIONAL]..
CHANGING.. [VALUE(<c>)] TYPE type [OPTIONAL]..
RETURNING VALUE(<r>)
EXCEPTIONS.. <e>..
```

Para declarar métodos estáticos(pertencem à classe), use o seguinte comando:

```
CLASS-METHODS <meth>...
```

Ambos têm a mesma sintaxe.

Quando você declara um método, você também define sua interface de parâmetros usando as adições IMPORTING, EXPORTING, CHANGING, e RETURNING. As adições definem parâmetros de entrada/saída, e o código de retorno. Elas também definem os atributos dos parâmetros da interface, principalmente quando um parâmetro é passado por referência ou valor (VALUE), seu tipo(TYPE), e se é opcional (OPTIONAL, DEFAULT). Ao contrário de módulos de funções, o modo default de passar um parâmetro em um método é por referência. Para passar um parâmetro por valor, você deve explicitamente especificar a adição VALUE. O valor de retorno (RETURNING) deve sempre ser passado explicitamente como um valor. Isto é ótimo para métodos que retornam um único valor de saída. Se você usá-lo, você não pode usar EXPORTING ou CHANGING.

Assim como em módulos de funções, você pode usar parâmetros de exceções (EXCEPTIONS) para permitir que o usuário reaja a situações de erro quando o método é executado.

### 3.3.3.2 Implementando Métodos

Deve-se implementar todos os métodos dentro de uma classe na parte de implementação.

```
METHOD <meth>.  
...  
ENDMETHOD.
```

Quando implementa o método, não precisa especificar quaisquer parâmetros de interface, já que são definidos dentro da declaração do método. Os parâmetros de interface de um método comportam-se como variáveis locais dentro do método de implementação. Pode-se definir variáveis locais adicionais dentro de um método usando o comando DATA.

Assim como módulos de funções, você pode usar o comando RAISE <exception> e MESSAGE RAISING para lidar com situações de erro. Quando você implementa um método estático, lembre-se que isto pode apenas trabalhar com atributos estáticos da sua classe. Métodos de instancias podem trabalhar com ambos atributos estáticos ou de instancia.

### 3.3.3.3 Chamando Métodos

Para chamar um método, use o seguinte comando:

```
CALL METHOD <meth> EXPORTING... <i> =.<f i>...
      IMPORTING... <e> =.<g i>...
      CHANGING ... <c> =.<f i>...
      RECEIVING      r = h
      EXCEPTIONS... <e> = rc i...
```

O modo como se acessa o método <meth> depende de onde está chamando. Dentro da parte de implementação da classe, pode-se chamar os métodos da mesma classe diretamente usando o nome <meth> deles.

```
CALL METHOD <meth>...
```

Fora da classe, a visibilidade do método depende da visibilidade atribuída a ele(public). Métodos de instancia visíveis podem ser chamados de fora da classe usando:

```
CALL METHOD <ref>-><meth>...
```

Onde <ref> é uma variável de referência cujo valor indica uma instância da classe. Métodos de classe visíveis podem ser chamados de fora da classe usando:

```
CALL METHOD <class>=><meth>...
```

onde <class> é o nome da classe relevante.

Quando chama um método, deve-se passar todos os parâmetros de entrada obrigatórios usando EXPORTING ou CHANGING no comando CALL METHOD. Pode-se importar os parâmetros de saídas para dentro de seu programa usando IMPORTING ou RECEIVING. Igualmente, pode-se manipular quaisquer exceções disparadas pelos métodos usando a adição EXCEPTIONS. No entanto, isto é

altamente recomendado. Passa-se e recebe valores em métodos da mesma maneira dos módulos de funções, incluindo a sintaxe após a adição correspondente:

```
... <Parâmetro Formal > = <Parâmetro Atual>
```

Os parâmetros de interface (Parâmetros Formais) estão sempre do lado esquerdo do sinal de igualdade. O novo parâmetro está sempre a direita. O sinal de igual não é um operador de atribuição neste contexto, ele meramente serve para copiar variáveis do programa para os parâmetros de interface de método.

Se a interface de um método consiste apenas de um simples parâmetro IMPORTING, você pode usar a seguinte forma resumida de chamada de método:

```
CALL METHOD <method>( f).
```

O parâmetro <f> é passado para os parâmetros de entrada do método. Se a interface de um método consiste apenas de parâmetros IMPORTING, você pode usar a seguinte forma resumida:

```
CALL METHOD <method>(....<ii> =.<f i>...).
```

Cada actual parameter <f i > é passado correspondenteis ao formal parameter <i i >.

### 3.3.4 Herança

A herança permite que você derive uma nova classe de uma já existente. Você faz isso usando a adição INHERITING FROM do modo a seguir:

```
CLASS <subclasse> DEFINITION INHERITING FROM <superclasse>.
```

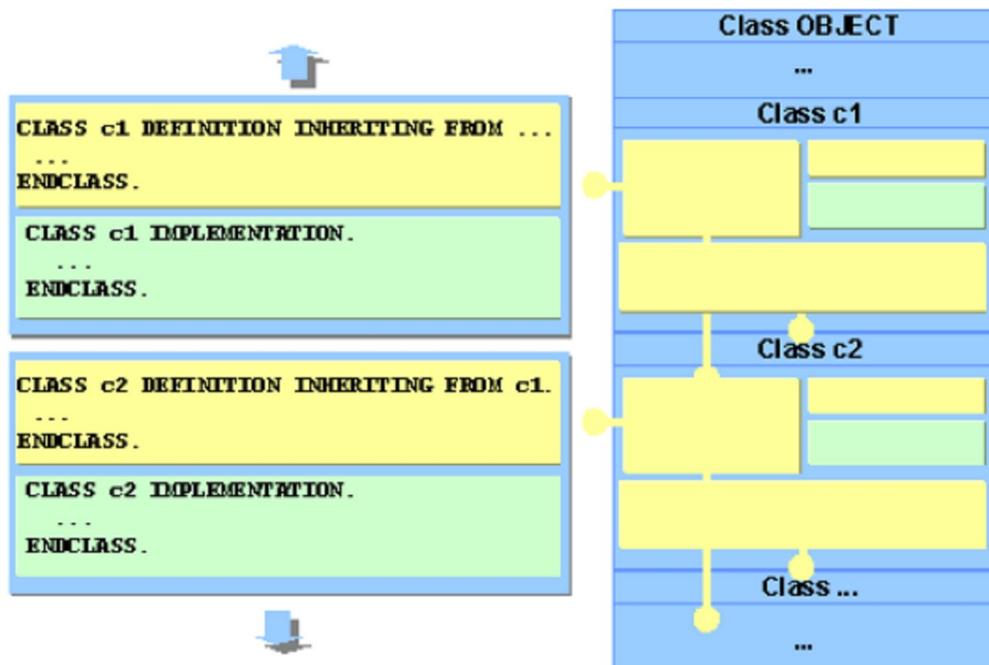
A nova classe <subclasse> herda todos os componentes da classe existente <superclasse>. A nova classe é chamada derivada da qual ela herdou. E a original é chamada superclasse da nova classe.

Se você adicionar novas declarações à subclasse ela conterà os mesmos componentes da superclasse, mais as adições. No entanto, apenas os componentes públicos e protegidos da superclasse estão visíveis na subclasse. Apesar dos componentes privados da superclasse existirem na subclasse, eles não estão visíveis. Você pode declarar componentes privados em uma subclasse que tem os

mesmo nomes dos componentes privados da superclasse. Cada classe trabalha com seus próprios componentes privados.

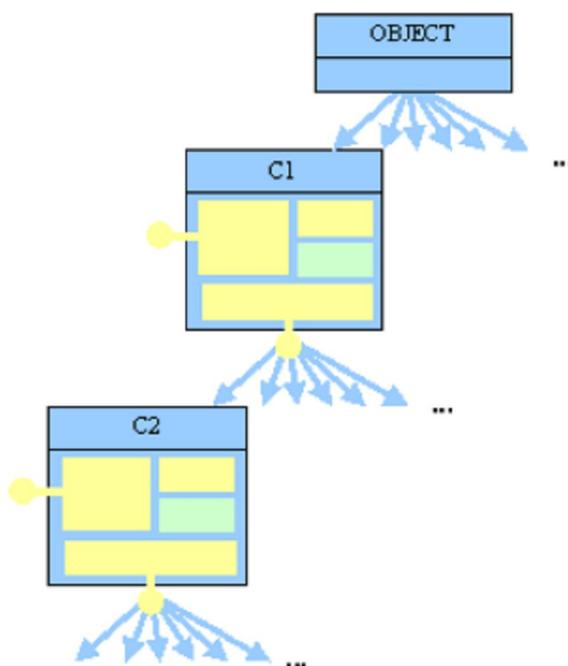
Se a superclasse não tem uma seção de visibilidade privada, a subclasse é uma réplica exata da superclasse. No entanto, você pode adicionar novos componentes à subclasse. Isto lhe permite que torne uma subclasse em uma versão especializada da superclasse. Se uma subclasse é também uma superclasse, você introduz um novo nível de especialização.

Uma classe pode ter mais do que uma subclasse direta, mas apenas uma superclasse direta. Isto é chamada herança simples. Quando subclasses herdam de superclasses e a superclasse é também uma subclasse de outra classe, todas as classes envolvidas formam uma árvore de herança, cujos degraus de especialização aumentam com cada novo nível hierárquico que você adicionar. Ao contrário, as classes se tornam mais generalizadas até se chegar a raiz da árvore de herança. A raiz de todas as árvores de herança em Objetos ABAP é a predefinida classe vazia OBJECT. Esta é a classe mais generalizada possível, já que não contém métodos nem atributos. Quando você define uma classe, você não precisa especificar explicitamente a superclasse - a relação é sempre definida implicitamente. Dentro de uma árvore de herança, dois nós adjacentes são a superclasse direta ou subclasse direta uma da outra. Outros nós relacionados são referidos como superclasse ou subclasse. A declaração de componentes de uma subclasse são distribuídas através todos os níveis da árvore de herança.



**Figura 18 – Herança visão geral – Fonte: Aspen Procwork.**

A parte esquerda do gráfico demonstra como você pode derivar uma subclasse `c2` de uma superclasse `c1` usando a adição `INHERITING FROM` no comando `CLASS`. A parte direita do gráfico demonstra como é a distribuição da subclasse dentro da árvore de herança, que alcança a classe vazia default `OBJECT`. Uma subclasse contém todos os componentes declarados acima dela na árvore de herança, e pode acessar aqueles que são declarados públicos ou protegidos.



**Figura 19 – Herança simples – Fonte: Aspen Procwork.**

Este gráfico ilustra herança simples. Uma classe pode apenas ter uma superclasse direta, mas pode ter mais do que uma subclasse direta. A classe vazia OBJECT é o nó-raiz de toda árvore de herança em objetos ABAP.

## CAPÍTULO 4

### IMPLEMENTAÇÃO

---

Neste capítulo serão apresentadas as implementações realizadas utilizando a linguagem de programação ABAP.

#### 4.1 CONFIGURAÇÕES DO SISTEMA

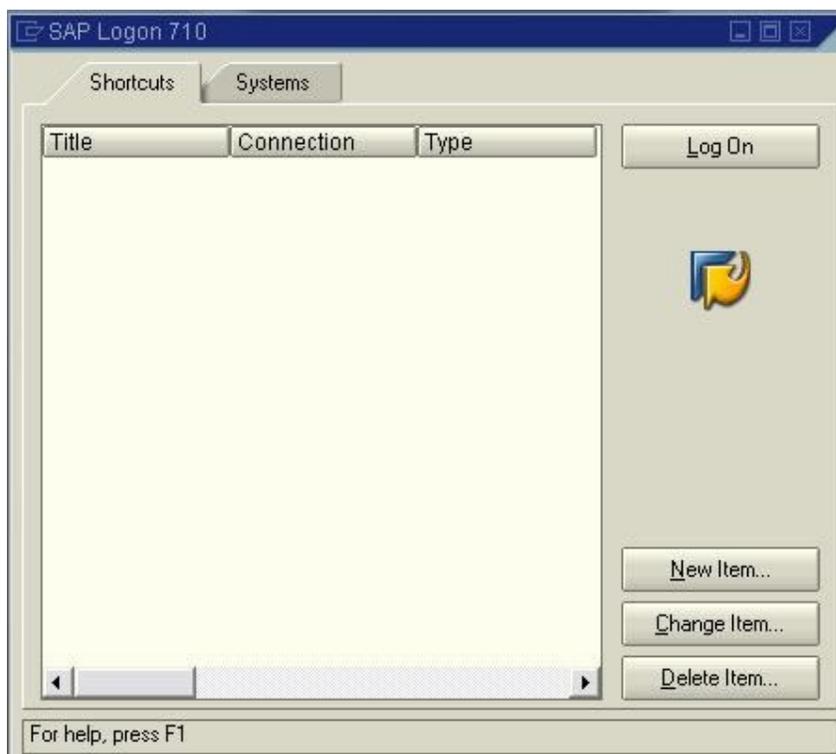
Neste trabalho foi utilizado o software MINISAP, que é uma versão mais enxuta do SAP. O MINISAP é um ERP da SAP que imita o sistema original, com algumas diferenças, sendo que a versão *full* chega a possuir até *terabyte* de tabelas e a versão Mini possui somente 22 Gigabytes de tabelas no banco.

As configurações mínimas necessária para o computador executar o MINISAP são:

- Sistema operacional: Windows XP Professional (Service Pack 2) ou Windows Server 2003;
- Hostname não deve exceder 13 caracteres;
- Partição tipo NTFS do sistema operacional;
- Internet Explorer 5.5 ou superior ou Firefox 1.0 ou superior;
- Pelo menos 1 Gb de memória RAM;
- Intel Pentium III/1.1 GHz ou superior (ou compatível);
- Disco rígido 22 GB temporário durante a instalação - 10 Gigabyte permanente;
- Alta resolução do monitor (1024x768 ou superior, 256 cores);
- Garantir que nenhum outro sistema SAP está instalado no seu computador;
- Se o servidor DHCP não está disponível na sua rede ou o computador não está ligada à rede, precisa-se instalar o adaptador à interface virtual MS Loopback conversor.

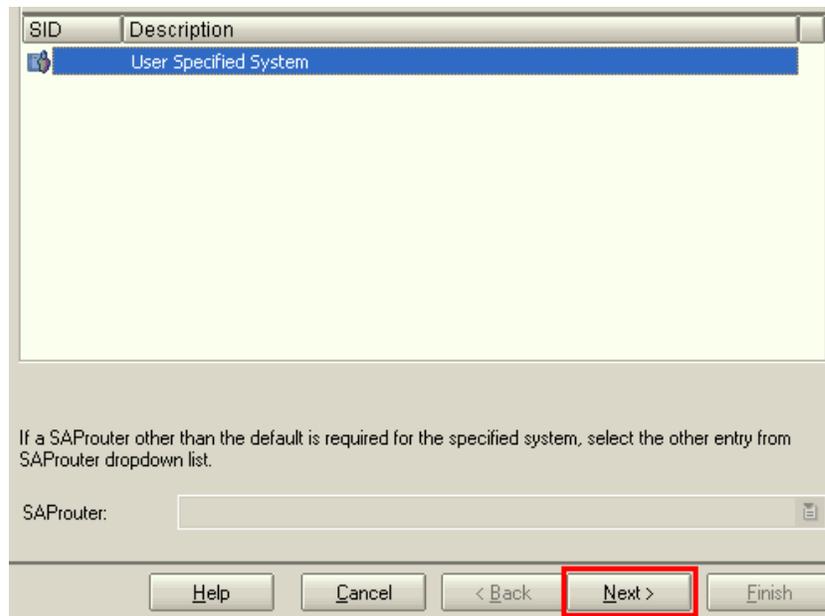
## 4.2 INSTALAÇÃO DO MINISAP

Após a instalação do MINISAP é necessário instalar o SAP GUI, o SAP GUI serve para autenticar os usuários que conectam na sessão. Após a instalação deve se criar um usuário, clicando em *New Item*, conforme Figura 20.



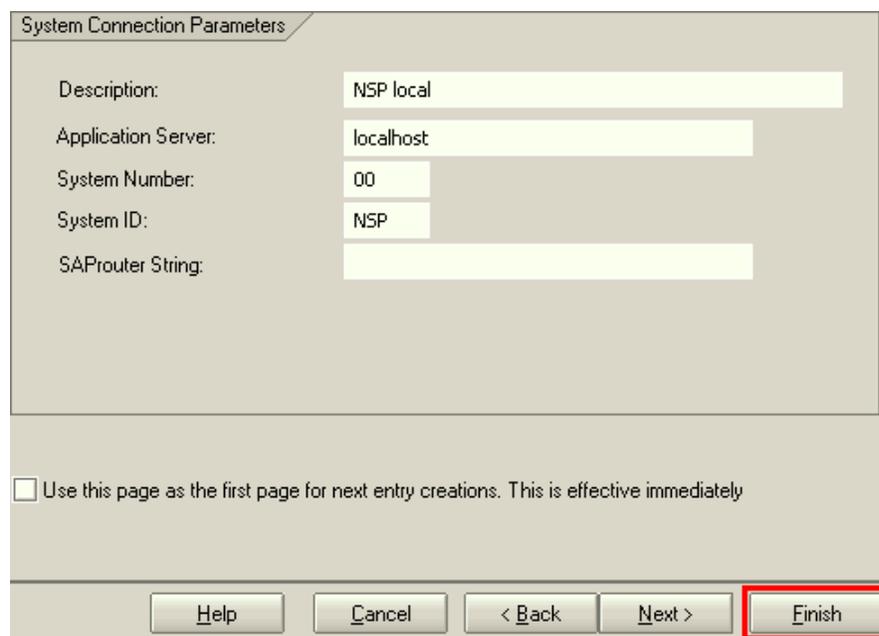
**Figura 20 – Tela inicial do SAP GUI.**

Para selecionar um usuário específico do sistema é preciso escolher a opção *next*, como mostra a Figura 21.



**Figura 21 – Tela de criação de usuário do SAP GUI.**

Para finalizar a criação de um usuário no SAP GUI é necessário preencher os campos com os valores indicados, como mostra a Figura 22 Após o preenchimento correto escolher *finish*.



**Figura 22 – Tela de finalização da criação de usuário do SAP GUI.**

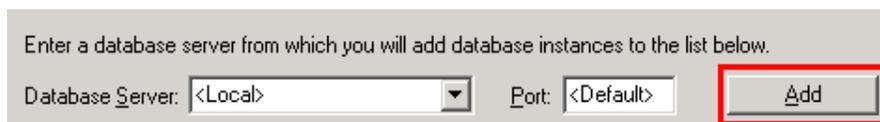
Após a instalação do SAP GUI, é preciso instalar o banco de dados SQL Studio. A instalação de bases de dados MaxDB Manager e MaxDB SQL Studio pode ser encontrado na pasta MaxDBTools. A instalação do MaxDB Manager é feita executando o arquivo DBM76.exe e em seguida é feita a instalação do arquivo SQLSTD76.exe que é o SQL Studio, sendo que ambos devem ser instalado nessa ordem.

Para executar o Database Manager será selecionado o menu Iniciar → Programas → MaxDB → Database Manager. Após a criação do banco no Database Manager é necessário clicar em ADD , como mostra a Figura 23.



**Figura 23 – Tela de criação de banco no Database Manager.**

Ao finalizar a execução aparece uma janela pop-up e é preciso clicar novamente em ADD, conforme Figura 24.



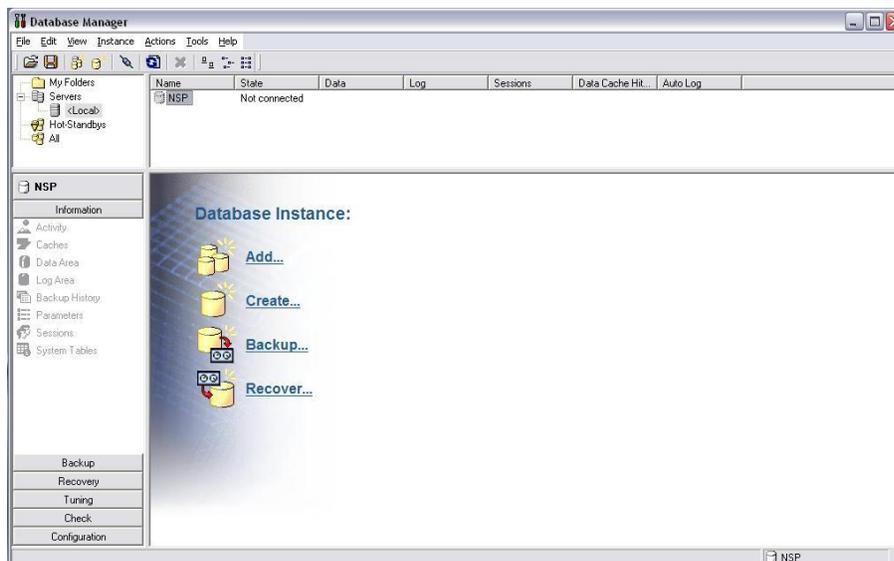
**Figura 24 – Janela de confirmação de novo banco de dados.**

A Figura 25 mostra que a base de dados instância NSP. Clique na OK para confirmar o presente diálogo.



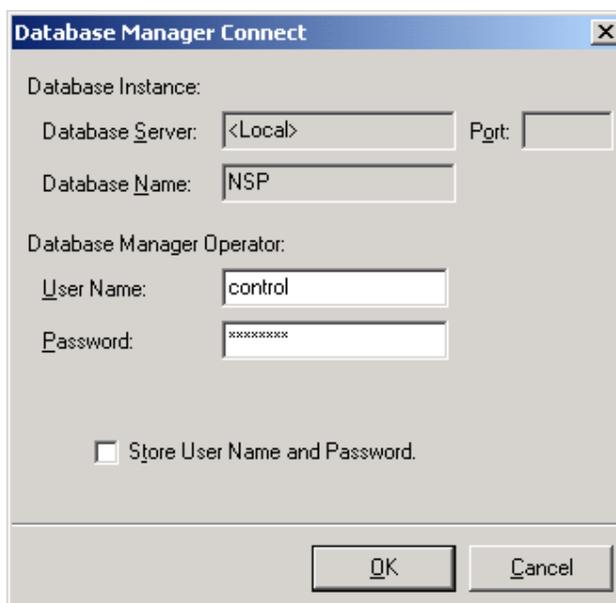
**Figura 25 – Tela que mostra a instancia NSP.**

Na Figura 26 será mostrado uma entrada para a NSP database é adicionado à janela principal. A solicitação de um usuário e senha é obtida com um duplo clique.



**Figura 26 – Tela de criação do banco de dados.**

A Figura 27 apresenta a tela de criação do banco de dados. O usuário é default do sistema e a senha utilizada é aquela que foi colocada durante a instalação.



**Figura 27 – Tela de criação do banco de dados II.**

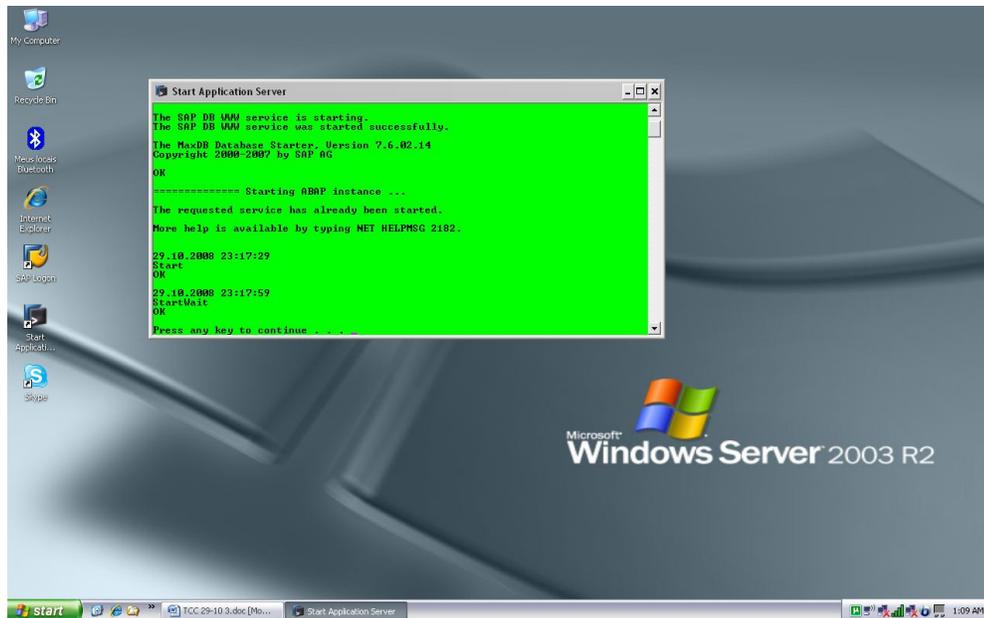
Para configurar o SQL Studio é preciso primeiramente inicializar o SQL Studio no menu Iniciar → Programas → MaxDB → SQL Studio.

Clicar em  para conectar o SQL Studio e ele vai pedir uma senha que é a mesma da instalação, como na Figura 28.



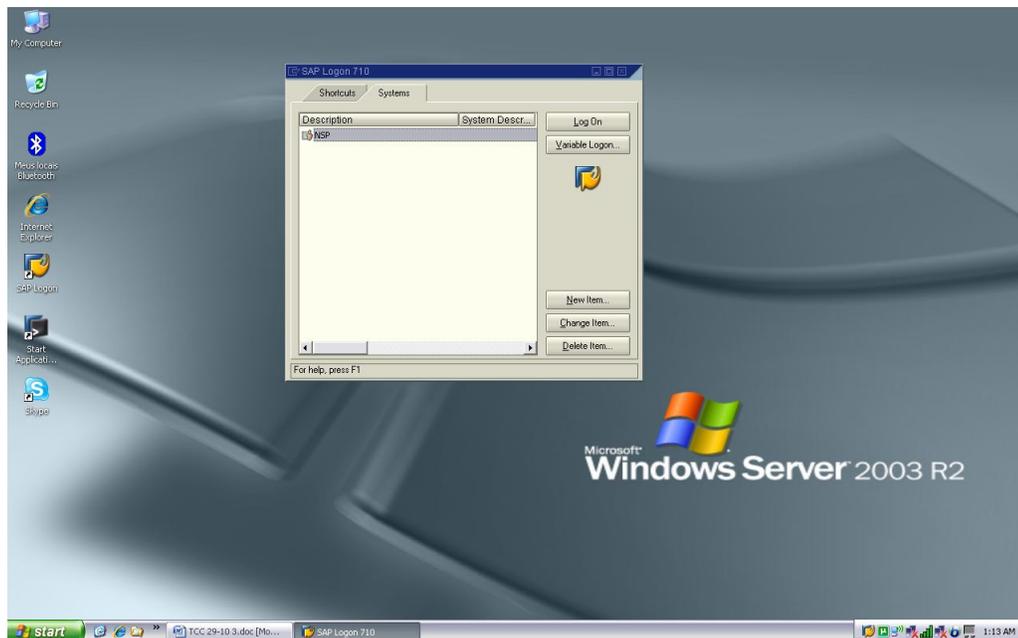
**Figura 28 – Tela de conexão do banco de dados.**

Após a instalação dos aplicativos citados anteriormente é só começar a utilizar o MiniSAP. Para inicializar o MiniSAP é necessário executar o aplicativo *Start Application Server*. Para executar: menu Iniciar → Programas → SAP Netweaver 7.0 ABAP Trial Version → NSP → Start Application Server, conforme a Figura 29.



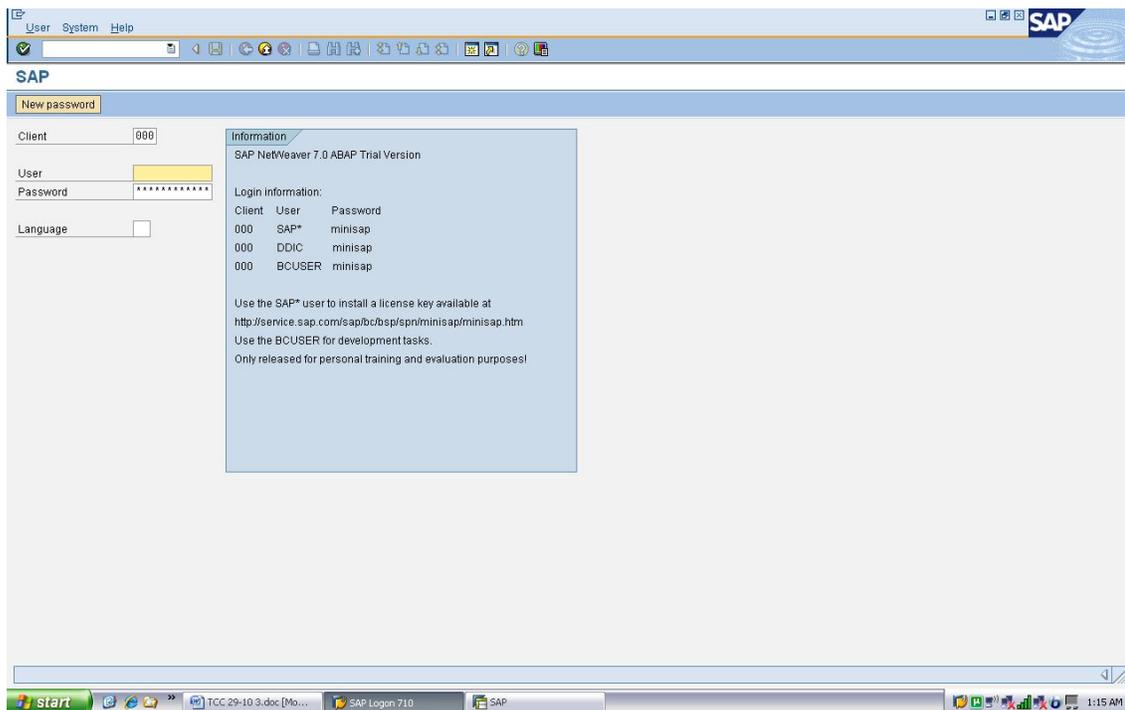
**Figura 29 – Tela de inicialização da aplicação server.**

Em seguida menu Iniciar → Programas → SAP → SAP Front End → SAP Logon.



**Figura 30 – Tela de inicialização do SAP GUI.**

A Figura 30 mostra que após a execução vai aparecer o usuário que foi criado durante a instalação. Após abrir o SAP Logon basta clicar em Log On. Em seguida, vai aparecer a tela inicial do MiniSAP com os campos de usuário e senha.



**Figura 31 – Tela de inicialização do SAP.**

O MINISAP possui por default 3 usuários, sendo eles:

- SAP\* – É o administrador do MiniSAP, onde tem permissão para instalar novas licenças que são renovadas a cada noventa dias;
- DDIC – É o usuário de manutenção do sistema;
- BCUSER – É o usuário desenvolvedor, quem cria os programas em ABAP.

Uma observação importante é que todos os programas em ABAP devem começar com a Z ou Y.

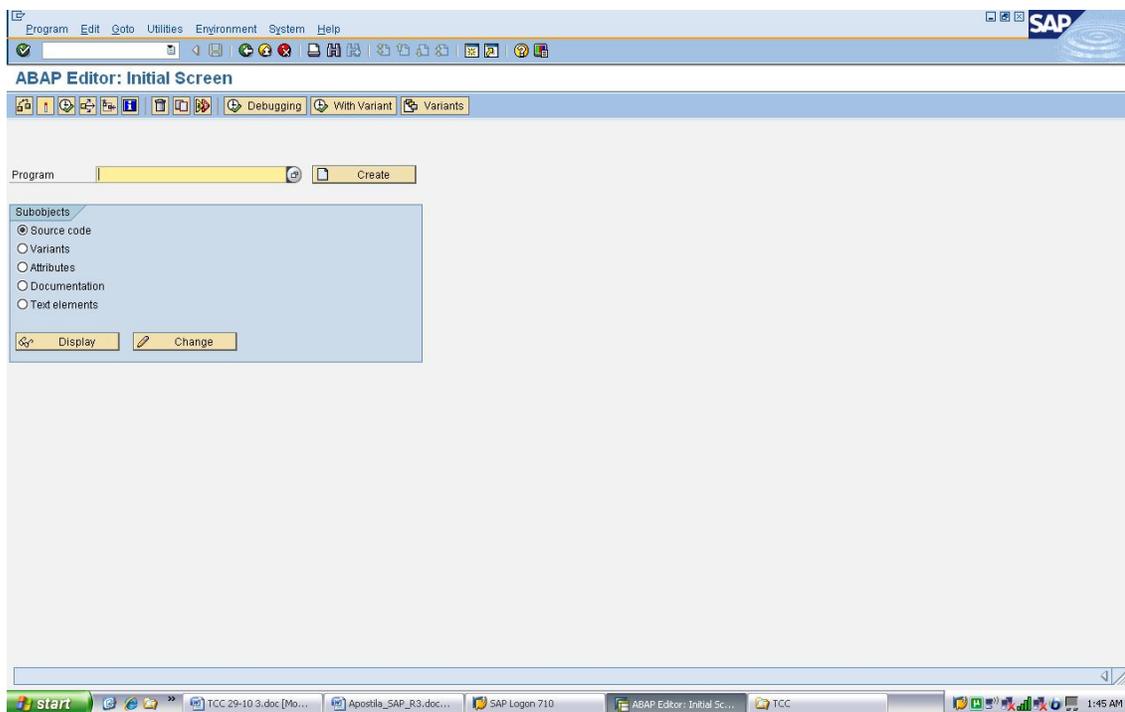
## 4.3 EXEMPLOS DE APLICAÇÕES

Nesta seção serão apresentadas as implementações desenvolvidas em linguagem ABAP/4. As aplicações desenvolvidas neste trabalho foram para mostrar como são feitas as criações de um programa ABAP.

### 4.3.1 Aplicação 1: Programa de uma calculadora (ZPCALC)

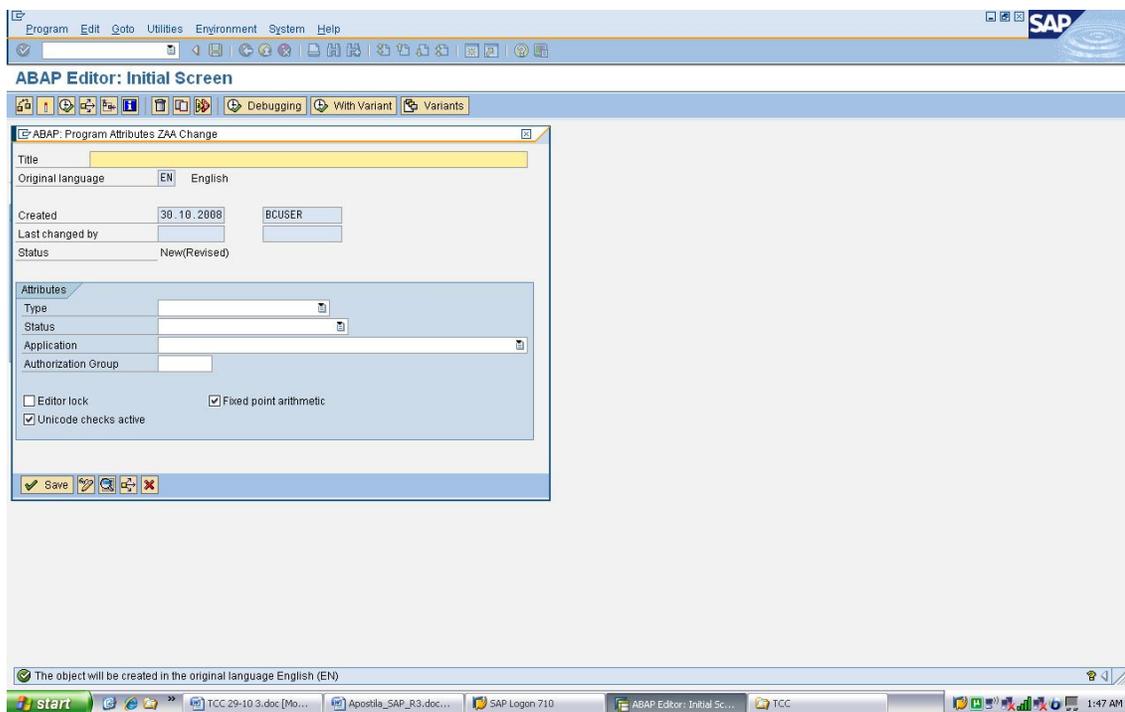
Esta aplicação tem a função de executar cálculos matemáticos básicos.

A Figura 32 mostra o campo de comando execute da transação SE38 e pressione a tecla *Enter*.



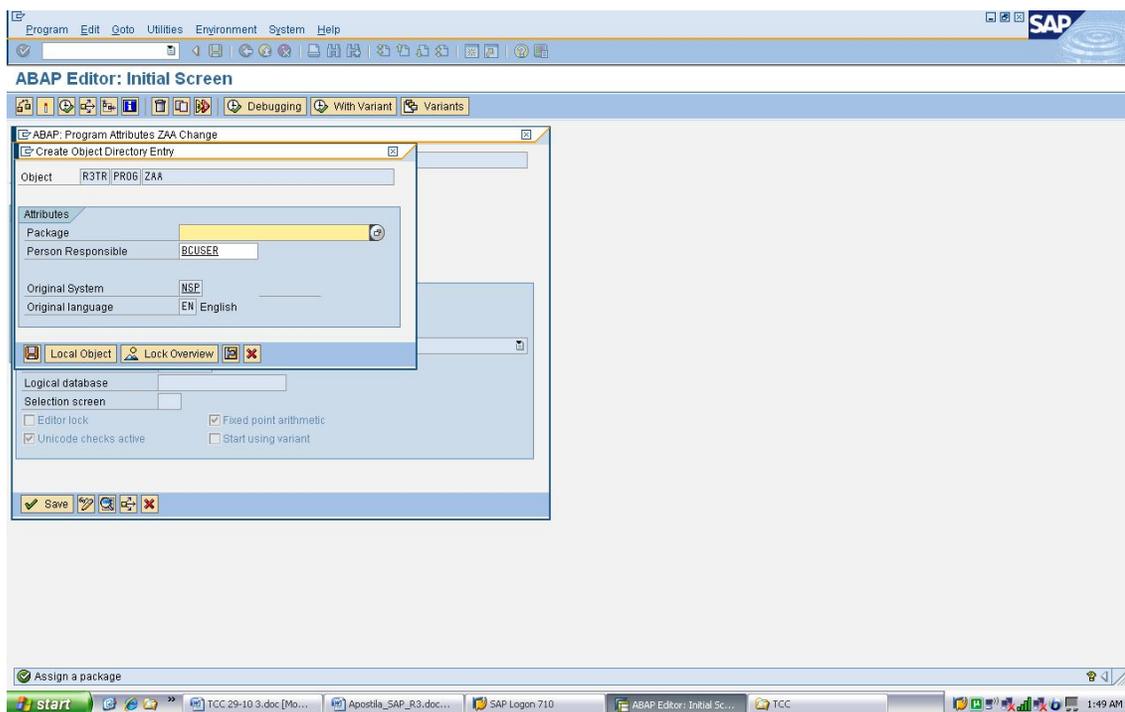
**Figura 32 – Tela de criação de um programa em ABAP.**

A Figura 32 mostra como colocar o nome que deseja criar e clicar em *Create*.



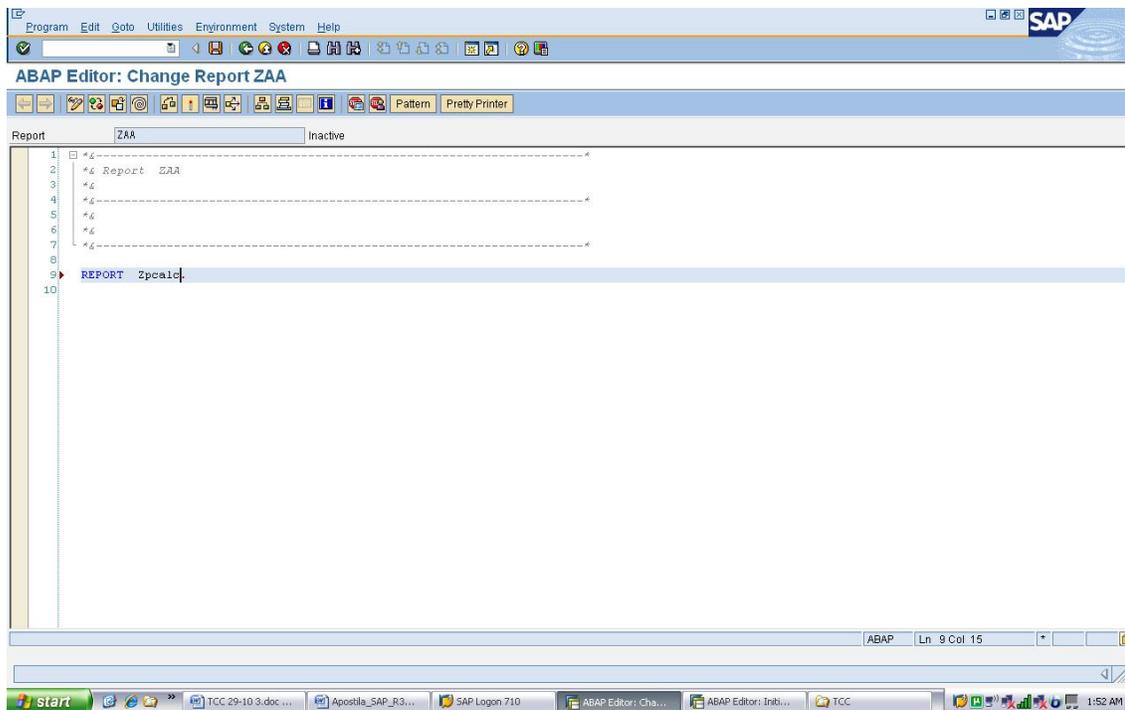
**Figura 33 – Tela de configuração de criação de programas em ABAP.**

O próximo passo é colocar o título do programa, tipo, status e aplicação. Após o preenchimento dos campos clique em Save como mostra a Figura 33.



**Figura 34 – Tela de escolha do pacote aonde salvar o programa.**

A Figura 34 mostra a escolha do pacote aonde deseja salvar, neste caso será salvo no pacote default do sistema: \$TMP, após a escolha clique em salvar (disquete). Logo em seguida aparecerá a tela de programação em ABAP.



**Figura 35 – Tela do ABAP Editor.**

Para obter o resultado com sucesso é necessário inserir o código fonte do programa.

```
*&-----*
*& Report ZPCALC
*&-----*
*&
*&-----*
```

```
REPORT ZPCALC.
```

```
* Parametros.
```

```
PARAMETERS:
```

```
  p_val1 TYPE i,
```

```
  p_val2 TYPE i,
```

```
  p_op   TYPE c.
```

```

* Definicao de variaveis locais.
DATA: vl_result TYPE p DECIMALS 2.

* Verifica se o usuario escolheu opcao valida.
IF NOT ( p_op = '+' OR
        p_op = '-' OR
        p_op = '/' OR
        p_op = '*' ).

    WRITE: 'No valid operation'(iop).

* Vefirica divisao por zero.
ELSEIF p_op = '/' AND p_val2 = 0.

    WRITE: 'Error division by zero'(zer).

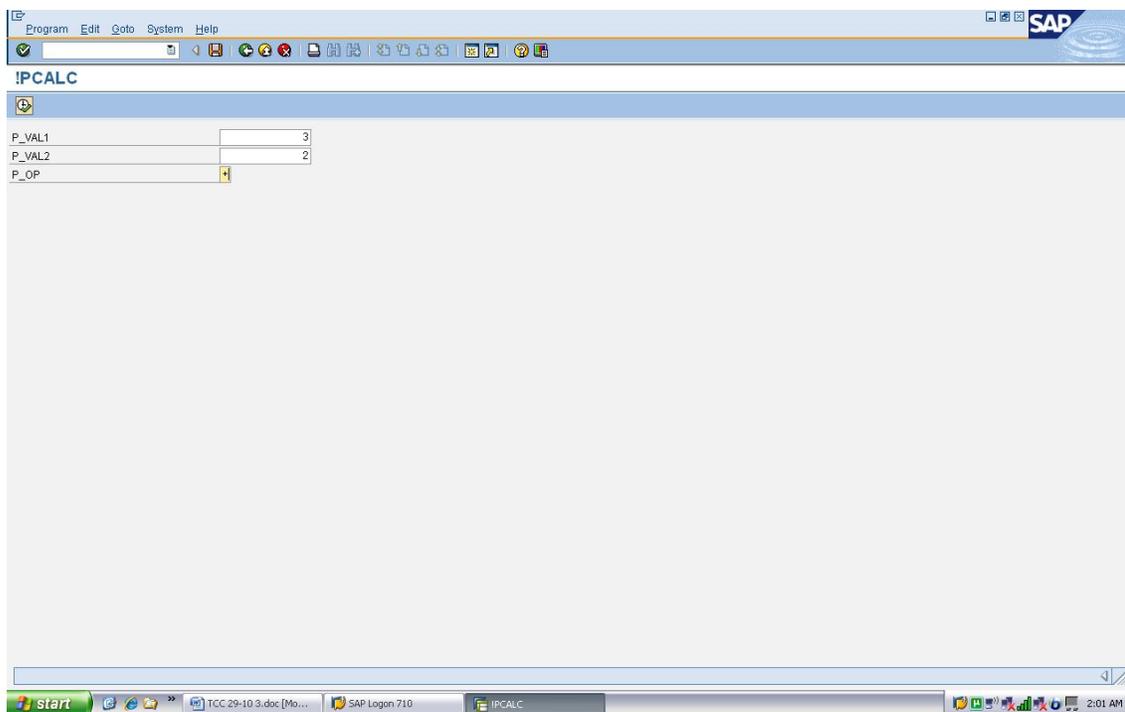
ELSE.

* Verifica a opcao correta escolhida pelo usuario.
CASE p_op.
    WHEN '+'.
        vl_result = p_vall + p_val2.
    WHEN '-'.
        vl_result = p_vall - p_val2.
    WHEN '/'.
        vl_result = p_vall / p_val2.
    WHEN '*'.
        vl_result = p_vall * p_val2.
ENDCASE.

* Imprime o resultado.
WRITE: 'Result'(res), vl_result.
ENDIF.

```

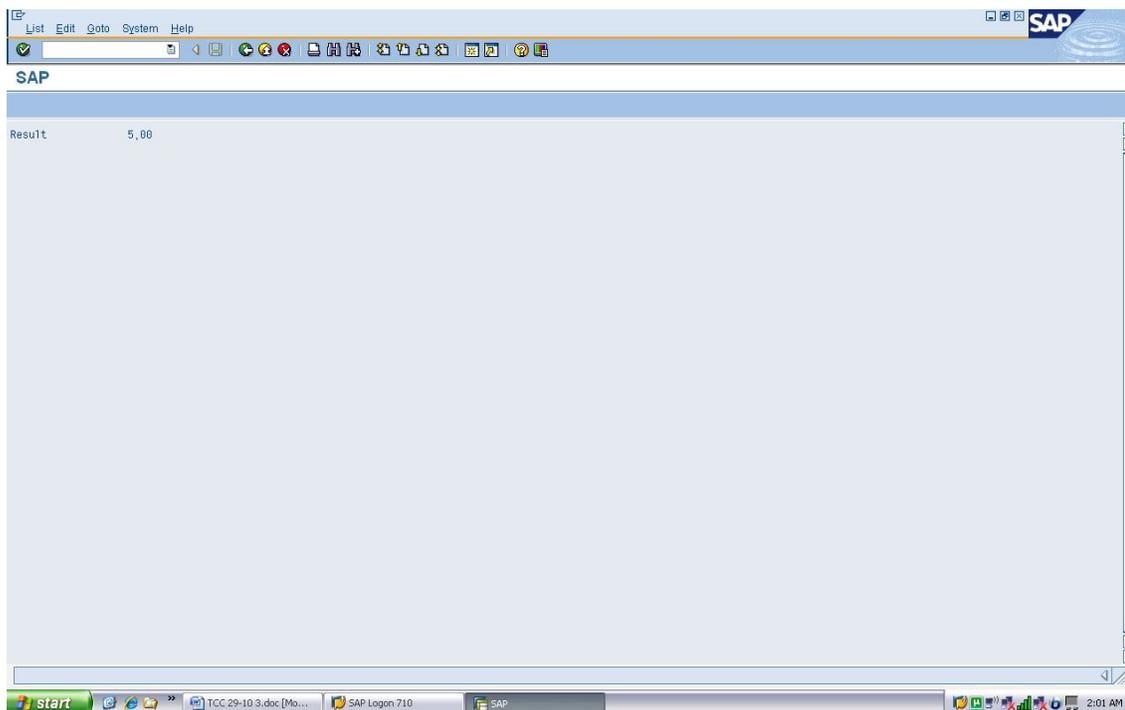
Após criar o programa, salvar ou utilizar o atalho Ctrl+S, em seguida executar o programa pressionando a tecla F8. A Figura 36 apresenta a tela de execução do programa da calculadora chamada ZPCALC.



**Figura 36 – Tela da execução do programa ZPCALC.**

Agora basta colocar os valores dos campos e pressionar a tecla *Enter*. Aparecerá o resultado da operação matemática que o usuário escolheu.

Na Figura 37 mostra a tela de resultado da aplicação ZPCALC.



**Figura 37 – Tela do resultado da aplicação ZPCALC.**

### 4.3.2 Aplicação 2: Rodízio de placas de automóvel (ZRODIZIO)

Esta aplicação tem a função de validar e mostrar quando uma determinada placa de um veículo não pode percorrer as ruas em dias da semana que possui um rodízio.

Os procedimentos para a execução da aplicação são os mesmos apresentados na aplicação 1. Após seguir todos os passos é só inserir programa código fonte.

```

*&-----*
*& Report  ZRODIZIO
*&-----*
*&
*&
*&-----*

REPORT  ZRODIZIO.

DATA:
  v_letras(3)  TYPE c,
  v_numeros(4) TYPE c,
  v_result(1)  TYPE c.

PARAMETER
* Digite a placa referente de seu veículo.
  p_placa(7) TYPE c OBLIGATORY.

START-OF-SELECTION.
  v_letras  = p_placa+0(3).
  v_numeros = p_placa+3(4).
  v_result  = v_numeros+3(1).

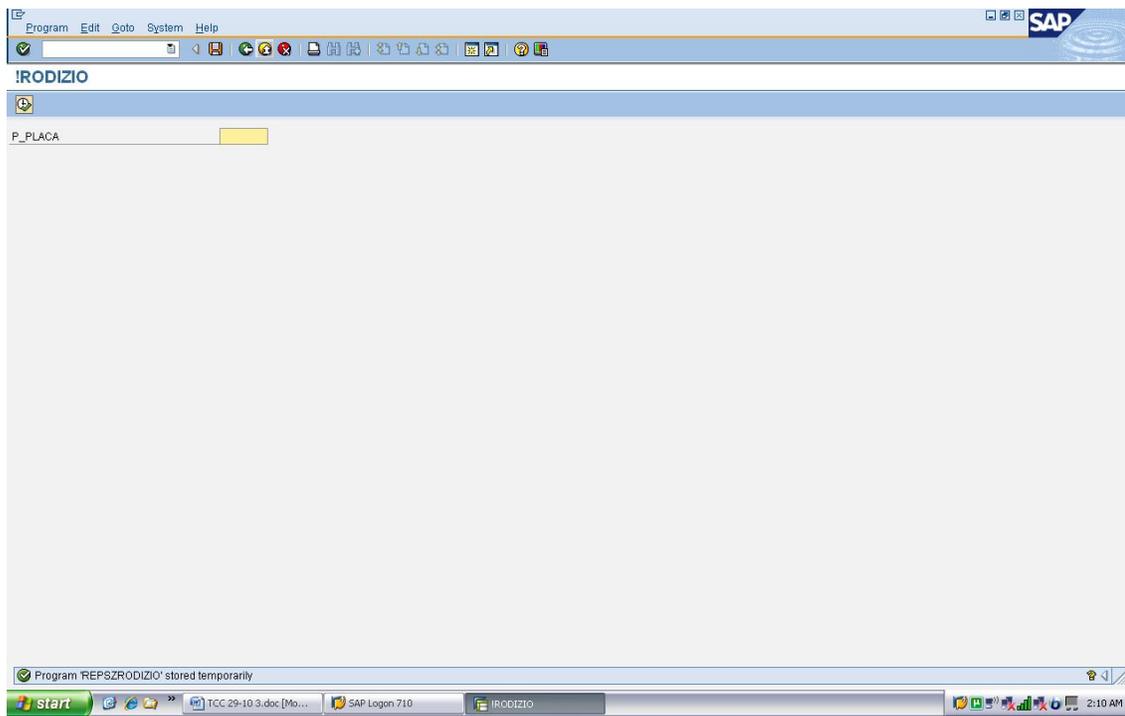
  IF NOT ( v_letras CA sy-abcde AND v_numeros CA '0123456789' ).
    WRITE: / 'Formato da placa está incorreto.'.
  ELSE.

    WRITE:
      'Placa : ', p_placa, /,
      'Letras : ', v_letras, /,

```

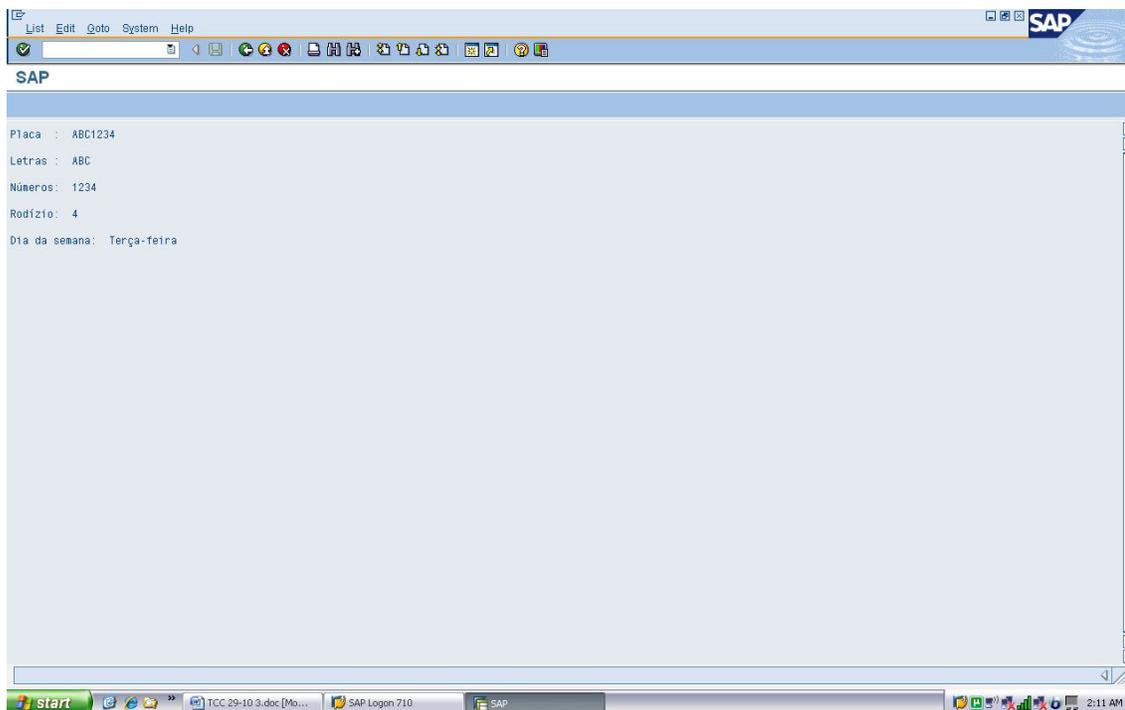
```
'Números: ', v_numeros, /,  
'Rodízio: ', v_result, /.  
  
WRITE 'Dia da semana: '.  
CASE v_result.  
  WHEN 1 OR 2.  
    WRITE 'Segunda-feira'.  
  WHEN 3 OR 4.  
    WRITE 'Terça-feira'.  
  WHEN 5 OR 6.  
    WRITE 'Quarta-feira'.  
  WHEN 7 OR 8.  
    WRITE 'Quinta-feira'.  
  WHEN OTHERS.  
    WRITE 'Sexta-feira'.  
ENDCASE.  
ENDIF.
```

Em seguida salvar e executar o programa. A Figura 38 mostra a tela de execução do programa ZRODIZIO.



**Figura 38 – Tela da execução do programa ZRODIZIO.**

Para obter o resultado da aplicação ZRODIZIO, Figura 39, preencher o campo com P\_Placa e pressionar a tecla *Enter*,



**Figura 39 – Tela do resultado da aplicação ZRODIZIO.**

O programa mostrará os seguintes dados:

- Placa – é a placa que o usuário digitou;
- Letras – são as letras da placa;
- Números – são os números da placa;
- Rodízio – mostra o final da placa;
- Dia da Semana – o dia que o veículo não pode circular.

#### **4.3.3 Aplicação 3: Emissão de Relatório (Z\_RELATORIO)**

Essa aplicação tem como objetivo executar um comando SQL e fazer relatório de uma tabela (spfli) do próprio sistema MiniSAP. Podendo escolher somente alguns campos da tabela a ser mostrado. Para emissão de relatórios existem duas

maneiras de extrair o relatório: uma é por meio de implementação de código fonte e outra através da transação SE11.

Primeiramente deve ser criado o programa como nome Z\_RELATORIO e em seguida inserir o código fonte.

```

*&-----*
*& Report  Z_TESTE
*&-----*
*&
*&
*&-----*

REPORT  Z_RELATORIO.

DATA: it_spfli TYPE TABLE OF spfli WITH HEADER LINE.

SELECT * FROM spfli INTO TABLE it_spfli.

LOOP AT it_spfli.

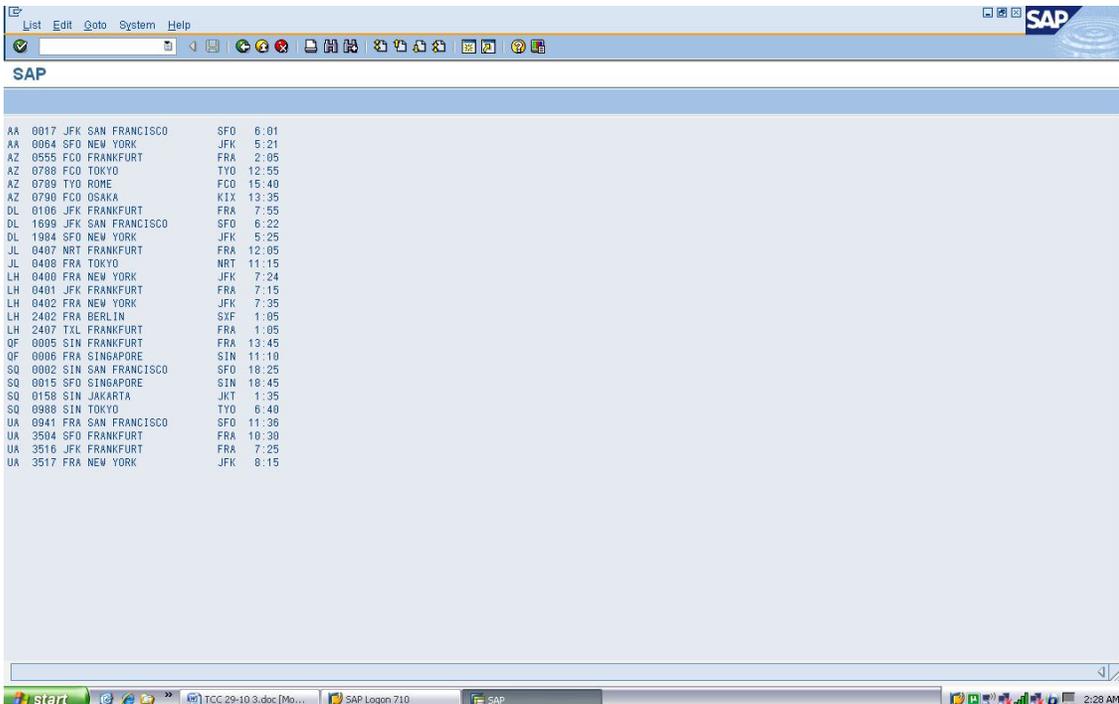
    WRITE: / it_spfli-carrid,
            it_spfli-connid,
            it_spfli-airpfrom,
            it_spfli-cityto,
            it_spfli-airpto,
            it_spfli-fltime.

ENDLOOP.

```

Em seguida é necessário salvar e executar o programa, e após a execução aparece a tela mostrada na Figura 40.

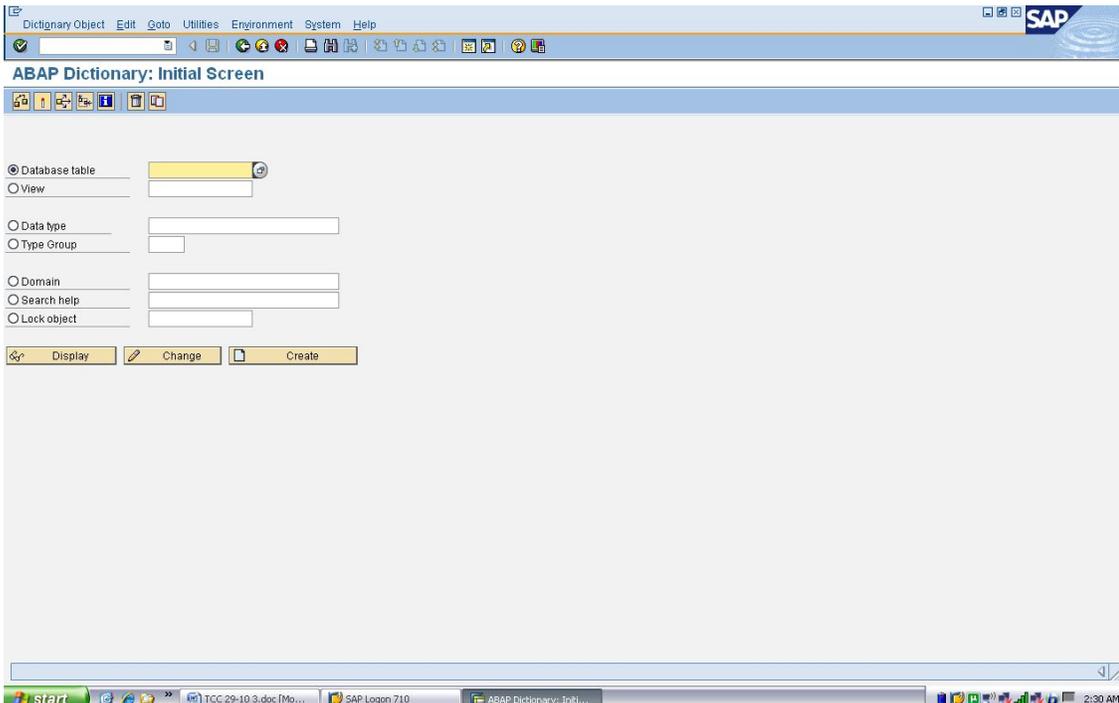
Essa é a maneira de mostrar o relatório usando um comando SQL.



Flight Number	Origin	Destination	Time
AA 0017	JFK	SAN FRANCISCO	SFO 6:01
AA 0064	SFO	NEW YORK	JFK 5:21
AZ 0555	FCO	FRANKFURT	FRA 2:05
AZ 0788	FCO	TOKYO	TYO 12:55
AZ 0789	TYO	ROME	FCO 15:40
AZ 0790	FCO	OSAKA	KIX 13:35
DL 0106	JFK	FRANKFURT	FRA 7:55
DL 1599	JFK	SAN FRANCISCO	SFO 6:22
DL 1984	SFO	NEW YORK	JFK 5:25
JL 0407	NRT	FRANKFURT	FRA 12:05
JL 0408	FRA	TOKYO	NRT 11:15
LH 0480	FRA	NEW YORK	JFK 7:24
LH 0401	JFK	FRANKFURT	FRA 7:15
LH 0402	FRA	NEW YORK	JFK 7:35
LH 2402	FRA	BERLIN	SXF 1:05
LH 2407	TXL	FRANKFURT	FRA 1:05
QF 0005	SIN	FRANKFURT	FRA 13:45
QF 0006	FRA	SINGAPORE	SIN 11:10
SQ 0002	SIN	SAN FRANCISCO	SFO 18:25
SQ 0015	SFO	SINGAPORE	SIN 18:45
SQ 0158	SIN	JAKARTA	JKT 1:35
SQ 0988	SIN	TOKYO	TYO 6:40
UA 0941	FRA	SAN FRANCISCO	SFO 11:36
UA 3504	SFO	FRANKFURT	FRA 10:30
UA 3516	JFK	FRANKFURT	FRA 7:25
UA 3517	FRA	NEW YORK	JFK 8:15

**Figura 40 – Tela do relatório Z\_RELATORIO.**

Outra maneira de realizar emissão de relatório é utilizando uma transação diferente da SE38. E no campo de comando entrar com a função SE11, como mostra a Figura 41.

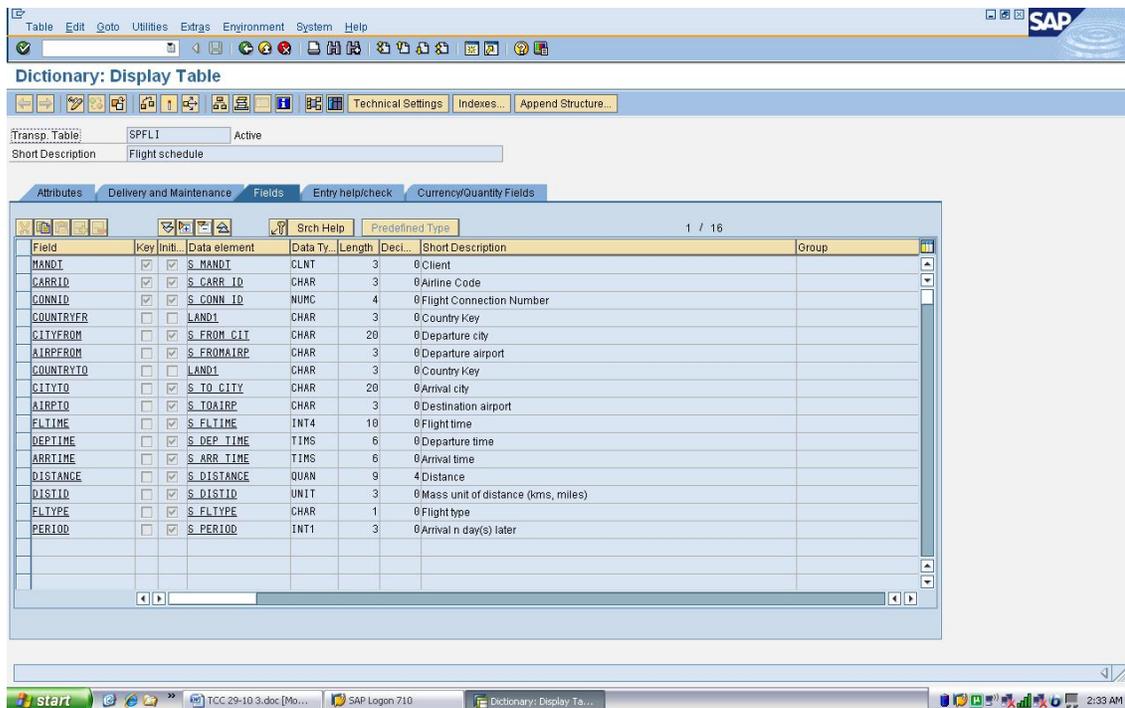


**ABAP Dictionary: Initial Screen**

Database table   
 View   
 Data type   
 Type Group   
 Domain   
 Search help   
 Lock object

**Figura 41 – Tela da transação SE11.**

Agora é só escolher a tabela que deseja emitir o relatório e as colunas aparecem listadas como na Figura 42.



**Figura 42 – Tela dos campos da tabela SPFLI.**

Em seguida deve escolher qual campo que deseja ver os dados no ícone  ou use a tecla de atalho Ctrl+Shift+F10.

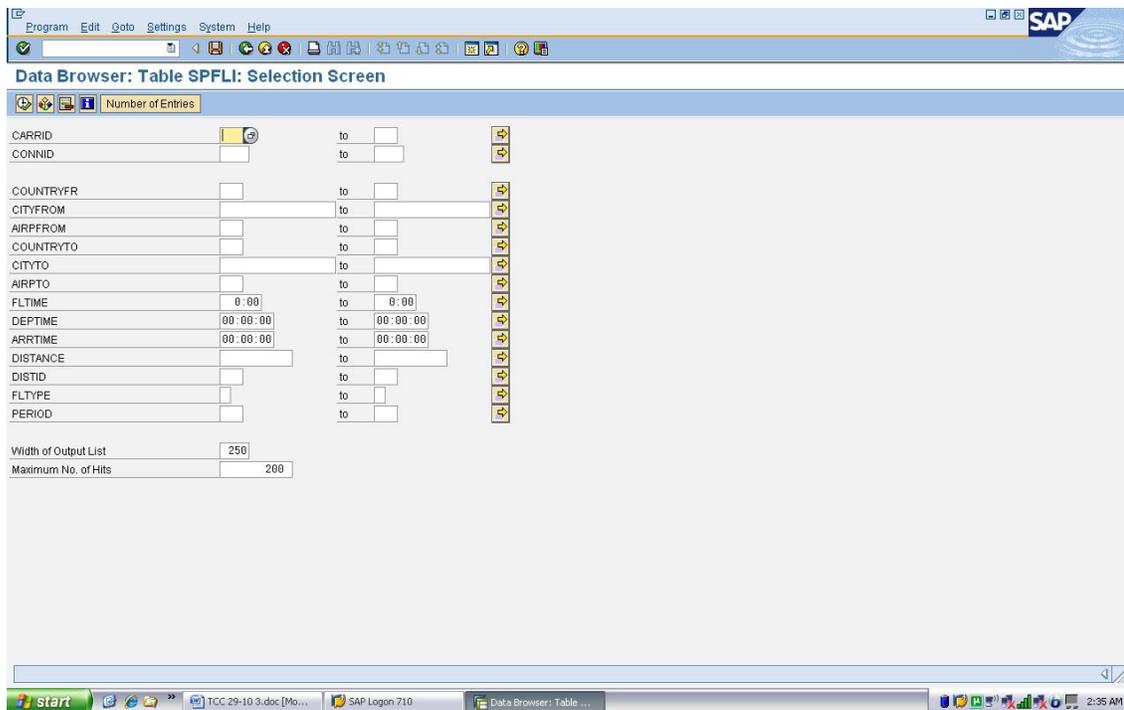


Figura 43 – Tela da escolha de campos a ser mostrado.

Na Figura 43 são mostrados todos os campos da tabela, podendo assim ser escolhido qual deles é necessário para emissão do relatório.

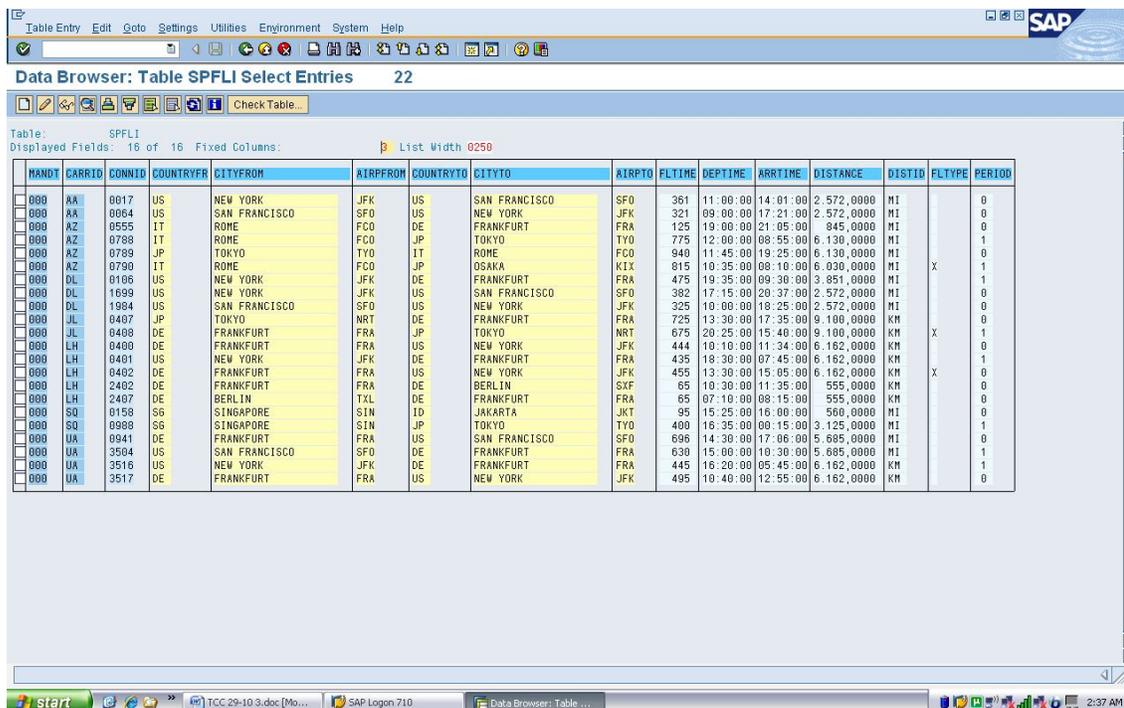


Figura 44 – Tela do resultado do relatório.

Está pronto mais um relatório emitido de outra maneira que não seja através de código fonte.

## CAPÍTULO 5

### CONCLUSÃO

---

Neste trabalho foram apresentados vários conceitos sobre o ERP chamado SAP, onde foram mostradas algumas vantagens sobre a implantação de um sistema.

Foram mostradas também algumas aplicações e suas facilidades no sistema SAP com a linguagem de programação ABAP. O funcionamento da arquitetura SAP/R3 foi explicado com alguns detalhes importantes.

As aplicações apresentadas neste trabalho foram de suma importância para o aprendizado. Sendo que são aplicações importantes para o dia-a-dia, na aplicação ZPCALC foi mostrado uma calculadora de operações matemáticas fundamentais. Na aplicação ZRODIZIO foram mostradas as importâncias da aplicação nos municípios que a utilizam para organizar o trânsito.

Os relatórios das tabelas presentes no sistema MiniSAP são simples de elaborar, onde foi apresentado duas formas de emitir relatórios, sendo uma através de programação ABAP (SQL) e a outra através de uma transação que escolhe os parâmetros necessários para emissão do relatório.

O desenvolvimento deste trabalho foi de primordial importância para o crescimento pessoal e profissional, contribuindo para que as pessoas interessadas no assunto possam dar continuidade na pesquisa.

## REFERÊNCIAS BIBLIOGRÁFICAS

- SANTOS, A. A. (1999) - *Sistemas integrados de gestão empresarial: Uma visão do uso de produtos ERP no Brasil*. Anais do VI Congresso Internacional de Custos, Braga, Setembro, 95-105.
- GUPTA, M. e KOHLI, A. (2006) Enterprise resource planning systems and its implications for operations function. *Technovation*, 26, 687–696.
- SOUZA, C. A. (2007) Capacidades e atores na gestão de sistemas ERP: Um estudo exploratório entre usuários corporativos do ERP da SAP.
- KRAFZIG, D. e BANKE, K. e SLAMA, D. *Enterprise SOA: Service-Oriented Architecture Best Practices*. 1.ed. Estados Unidos da América: Prentice Hall, 2004. ISBN 0131465759.
- Martin Keen, Susan Bishop, Alan Hopkins, Sven Milinski, Chris Nott, Rick Robinson, Jonathan Adams, Paul Verschueren, Amit Acharya. *Patterns: Implementing an SOA using an Enterprise Service Bus* (em inglês). Página visitada em 4 de junho de 2007.
- BELLA, V. Análise do ciclo de vida e fases de implementação de sistemas ERP.*
- MIÃO, R. Implementação de sistemas ERP SAP/R3 e suas tecnologias Middleware.*
- SANTOS, A. A. Um enfoque sobre a utilização do SAP R/3 em contabilidade e custos.*
- Blog de ABAP, Dicas Sobre Programação em ABAP. Disponível em <http://www.infocms.com.br/abap/cgi/sumario.php>. Acesso em 25 de Março de 2008.
- CBSCONSULTING, Consultoria em ERP's. Disponível em <http://www.cbsconsulting.com.br/erp.htm>. Acesso em 03 de setembro de 2008.
- [SOUZA&ZWICKER99] SOUZA, César Alexandre de. ZWICKER, Ronaldo. Um Modelo de Ciclo de Vida em Sistemas ERP: Aspectos Relacionados à sua Seleção, Implementação e Utilização. IV SEMEAD. 1999

SAP Brasil, Software de gestão empresarial, soluções, aplicativos e serviços para grandes, médias e pequenas empresas. Disponível em <http://www.sap.com/brazil/index.epx>. Acesso em 21 de junho de 2008.

SAP R/3 Módulo ABAP Introdução – 21 de Maio de 2007. Plaut Consultoria.

OLIVEIRA, C E C - Programação orientada a objeto em ABAP/4 – Aspen Procwork.