

MARCELO HENRIQUE CELERI

ADEMPIERE: UM ESTUDO DA ORGANIZAÇÃO DO AMBIENTE

Assis
2009

ADEMPIERE: UM ESTUDO DA ORGANIZAÇÃO DO AMBIENTE

MARCELO HENRIQUE CELERI

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:

Prof. Ms. Osmar Aparecido Machado
Imesa-FEMA

Nome: (Analisador1)
Imesa-FEMA

Nome: (Analisador2)
Imesa-FEMA

Assis
2009

MARCELO HENRIQUE CELERI

ADEMPIERE: UM ESTUDO DA ORGANIZAÇÃO DO AMBIENTE

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:

Prof. Ms. Osmar Aparecido Machado
Imesa-FEMA

Assis
2009

Dedico este trabalho a Deus, presença constante, sem Ele, nada seria possível.

A toda minha família, pelo apoio e incentivo incondicional em todas as horas.

Muito obrigado.

AGRADECIMENTOS

A paciência, sabedoria e a concentração, que puderam tornar este projeto uma realidade, pois encontrei o apoio dos amigos e incentivo de várias pessoas. Reconheço o meu agradecimento:

Ao Prof. Ms. Osmar Aparecido Machado, pelo apoio como orientador, a minha gratidão pelo apoio, incentivo, sugestões e pelas sábias orientações em todo o projeto que permitiram avançar em meu saber e ultrapassar obstáculos;

A todos os professores que contribuíram decisivamente para a minha, formação acadêmica, profissional e pessoal.

Aos meus familiares, que em todos os momentos de fraqueza me deram forças para seguir, encorajando-me, respeitando-me nas minhas ausências, os meus silêncios, preservando o meu tempo e espaço, a vocês todo o meu carinho e dedicação, em especial a minha tia Maristela, minha avó Marina, meu irmão Marcio e minha namorada Miriana, obrigado pelo estímulo como ser humano e pela confiança, estando sempre ao meu lado me ajudando a enfrentar os obstáculos do dia-a-dia, vivenciando cada desafio e comemorando cada vitória.

RESUMO

O objetivo do presente trabalho é um estudo e análise da organização do Adempiere, um sistema ERP, para que o mesmo possa ser utilizado por empresas de pequeno porte e também, ser utilizado em meio acadêmico. Os ERPs proprietários, comercializados no mercado, possuem altos custos de aquisição e implementação, assim, espera-se que este trabalho possa contribuir para promover o uso de sistemas ERPs, disponibilizados em código aberto. O trabalho foi desenvolvido por meio de testes de uso, análise do ambiente e testes de implantação. Os resultados apontaram que o sistema Adempiere possui características que o tornam apto, em termos de organização e qualidade de padrões, a ser utilizado por quaisquer tipos de organização. Além disso, pode ser também utilizado para o ensino acadêmico, como uma ferramenta pronta que pode ser desmontada e estudada. Espera-se também, que este trabalho possa cumprir com outros objetivos, não elencados no trabalho, como o de ser um instrumento que contribua para a disseminação do software livre.

Palavras-chave: ERP, Adempiere, Software Livre.

ABSTRACT

The aim of this paper is a study and analysis of the Adempiere organization, an ERP system so that it can be used by small companies and also be used in university. The ERPs softwares, traded on the market, have high acquisition costs and implementation as well, it is expected that this work will help to promote the use of ERP systems, available in open source. The work was developed through usability testing, environmental analysis and testing to deployment. The results showed that the Adempiere system has features that make it suitable in terms of organization and quality standards to be used for any type of organization. Moreover, it can also be used for academic teaching, as a ready tool that can be disassembled and studied. It is also hoped that this work can meet other goals, not listed in the work, as being an instrument that contributes to the spread of free software.

Key-Words: ERP, Adempiere, Free Software.

LISTA DE ILUSTRAÇÕES

Figura 1 – Integralização de atividades pelo ERP	14
Figura 2 – Evolução do Sistema ERP	17
Figura 3 – Estrutura Funcional do Adempiere	23
Figura 4 – Exemplo de Indicador do ERP Adempiere.....	24
Figura 5 – Fluxo do processo de entrada.....	24
Figura 6 – Fluxo do processo de saída	25
Figura 7 – Funcionamento da JVM	26
Figura 8 – Passos da Especificação MDA	32
Figura 9 – Modelo Tela – Cadastro de Produtos.....	34
Figura 10 – Modelo Form – Fatura Mensal de Produtos.....	35
Figura 11 – Tela Principal – Apresentação Web.....	36
Figura 12 – Colunas padrões do banco de dados – tabela ad_menu.....	38
Figura 13 – Funcionalidades do Adempiere.....	39
Figura 14 – Arvore da Organização do Menu Principal.....	40
Figura 15 – Tela responsável pela configuração de menus.....	40
Figura 16 – Organização do Código Fonte	43
Figura 17 – Estrutura do Código Fonte	43

LISTA DE GRÁFICOS

Gráfico 1 – Comparativo no Mercado Nacional.....	19
Gráfico 2 – Uso de Banco de Dados pelos ERP	20

LISTA DE QUADROS

Quadro 1 – Prefixos da Tabela de Banco de Dados Adempiere	37
Quadro 2 – Colunas Padrões das Tabelas	37

LISTA DE ABREVIATURAS E SIGLAS

SGBD	Sistema Gerenciador de Banco de Dados
SBD	Sistema de Banco de Dados
TPD	Tecnologia em Processamento de Dados
BCC	Bacharelado em Ciência da Computação
SQL	Structure Query Language

SUMÁRIO

1. INTRODUÇÃO	13
2. ERP – ENTERPRISE RESOURCE PLANNING.....	16
2.1. Histórico da evolução dos ERPs	16
2.2. E.R.P. no Brasil	18
2.3. Principais Produtos / Fornecedores	19
3. ADEMPIERE	22
3.1. Evolução do Adempiere.....	22
3.2. Ambiente de Desenvolvimento	25
3.2.1. Linguagem.....	25
3.2.2. Banco de dados	26
3.2.3. Software Livre	27
4. DESCRIÇÃO DO AMBIENTE ADEMPIERE	29
4.1. Padrões de Projeto	30
4.2. Padrão MDA	31
4.3. Organização do Código do Adempiere	33
4.4. Tipos de Dados.....	36
4.5. Organização	38
4.6. Estrutura do Código.....	42
5. CONCLUSÃO	44
6. REFERÊNCIAS BIBLIOGRAFICAS	46

1. Introdução

Toda informação obtida por uma empresa é imprescindível para seu sucesso. É por meio delas que as empresas tomam suas decisões, a falta de informação correta e precisa pode, portanto, acarretar em prejuízos, queda de crescimento e até mesmo o fim de uma empresa [3].

Por isso os sistemas computacionais voltados à gestão das organizações tem sido importantes no sentido de prover informações de qualidade tanto para os processos operacionais quanto os de tomada de decisões nas organizações. Investimentos em sistemas, sejam eles grandes ou pequenos, necessitam de um planejamento detalhado dos diferentes ambientes de trabalho onde será implantado. Os ERPs são sistemas que, cada vez mais oferecem soluções tecnológicas e de gestão alinhados para atender às necessidades organizacionais.

Estes sistemas surgiram de uma necessidade de gerenciar informações de maneira clara, completa e eficiente das informações que trafegam dentro de uma empresa. É uma ferramenta que pode ser utilizada tanto para os processos de tomada de decisão quanto para os processos transacionais, ou seja, das operações rotineiras das organizações. Esta característica é facilitada em função de o software utilizar um banco de dados único e da tecnologia cliente/servidor. Neste ambiente o usuário acessa uma rotina que faz a interação entre o banco e os diversos módulos do sistema, disponibilizando informações precisas, com integridade e sem redundâncias. Desta forma, é possível o acesso das mesmas informações por todos os módulos da empresa. [1]

As informações corretas e disponibilizadas no tempo adequado trazem muitas vantagens para as organizações, como a flexibilidade, economia de custos, relatórios consistentes que geram respostas de acordo com as necessidades da empresa.

Estes sistemas já vêm sendo utilizados por grandes organizações desde os anos 80. Por conta dos altos custos dos ERPs, as micros e pequenas empresas tem dificuldades em adotá-los, mas com o desenvolvimento de TI (Tecnologia da Informação) já é possível encontrá-los em versões de software livre, com custos reduzidos quando comparados aos aplicativos proprietários.

ERP (*Enterprise Resource Planning*) é um termo genérico usado para representar o conjunto de atividades executadas por um software de vários módulos, integrando os dados e processos de vários departamentos ou todos de uma organização em um único

sistema [3]. No Brasil é conhecido também como Sistema Integrado de Gestão, cuja principal característica, em termos organizacionais é a capacidade de integrar num mesmo ambiente diversas funções de uma organização, como RH, Finanças, Produção, Marketing, etc.

Este tipo de sistema tem por objetivo auxiliar o gestor de uma empresa nas importantes fases de seu negócio, incluindo desenvolvimento de produto, compra de itens, manutenção de inventários, interação com fornecedores, serviços a clientes e acompanhamento de ordens de produção. A figura 1 representa a idéia de um E.R.P.

Tipicamente um sistema ERP usa e está integrado a uma única base de dados. O banco de dados interage com todos os aplicativos do sistema. Desta forma, se reduz o número de redundâncias de informações e retrabalhos de digitação dos dados, o que assegura maior qualidade das informações obtidas. A implantação de um sistema ERP envolve uma extensa análise dos processos da empresa, são necessários treinamentos de operadores e principalmente do pessoal administrativo, que são os principais responsáveis pelo sucesso da implantação. Quando estes funcionários não se envolvem no processo de implantação, são grandes as chances de boicote ao sistema por conta das mudanças que ocorrem tanto na gestão como nos processos de negócios.

Além disso, são necessário investimentos em hardware (equipamentos), investimentos em equipes de TI e reestruturação nos métodos de trabalho. [12]



Figura 1: Integralização de atividades pelo ERP
 Fonte: http://www.fagus-ti.com.br/images/erp_modules.gif

Dentre alguns dos principais motivos que levam uma empresa a usar um ERP, podem ser citados a busca de competitividade, melhora na produtividade e qualidade dos serviços prestados aos clientes, na redução de custos, melhor gerenciamento de estoques e no planejamento e alocação de recursos.

Dentre as opções disponibilizadas em software livre, destaca-se o Adempiere, um ERP de código aberto, que surgiu em 2006, sendo um fork, uma versão do projeto Compiere. O grande motivador para a criação do Adempiere foi o fato do código do Compiere ser muito centralizado e as sugestões de melhoria apresentadas pela comunidade de software livre não estavam sendo implementadas no software, pois o dono do projeto não as aceitava, alegando que as contribuições só estavam sendo abertas a parceiros autorizados para efetuar as melhorias sugeridas pela comunidade. Assim o Adempiere, com maior participação da comunidade livre, vem se desenvolvendo mais rapidamente, com várias modificações e novas implementações, melhorando sua estabilidade e funcionalidade. [2]

Este trabalho trata-se de um estudo sobre ERP (*Enterprise Resource Planning*), contudo, serão abordados diversos assuntos no decorrer do trabalho, como software livre, banco de dados, Java, ERP, etc. Desta forma, nas próximas seções será realizada uma revisão dos conceitos sobre o assunto, a fim de situar os leitores sobre o contexto no qual o assunto principal do trabalho se desenvolve. Assim, esta seção, o Capítulo 1, a introdução faz uma rápida contextualização do trabalho. O Capítulo 2 descreve os conceitos de ERP, faz um histórico do software no Brasil. O capítulo 3 trata da apresentação do Adempiere, que é o produto a ser estudado neste trabalho, e o capítulo 4 detalha as principais características do ambiente Adempiere. Para finalizar, capítulo 5 traz as conclusões obtidas ao longo deste trabalho.

2. ERP – *Enterprise Resource Planning*

2.1 Histórico da evolução dos ERPs.

No início da década de 60, quando os conceitos modernos de controle tecnológico e da gestão corporativa começaram a surgir, a tecnologia vigente na época era a dos gigantescos mainframes, onde foram executados os primeiros sistemas de controle de estoques, foi a primeira tecnologia que uniu a atividade de gestão e tecnologia da época. Esta tecnologia exigia altos valores financeiros, altos investimentos em infra-estrutura capacitação de pessoal, era muito lenta, ainda assim os altos investimentos justificavam-se, pois demandava de menos processos manuais [7]. Contudo uma das principais vantagens desta tecnologia era a agilidade que dava aos processos de negócios.

Entre a década de 60 e até o final da década de 70 com a expansão econômica e uma maior participação computacional nas empresas surgiram primeiras tecnologias que antecederam os ERP's, demonstrados na Figura 2. A primeira delas foi o MRP (Material Requirement Planning ou planejamento das requisições de materiais), cujas funcionalidades consistiam em processos lógicos de cálculo, a fim de converter as previsões de demanda em programação de necessidade. Além disso tinha outras funções como calcular a quantidade de componentes necessários a produção de um determinado produto sem que houvesse falta de componentes no processo ou final da produção [3]. Como se observa, este conceito foi muito utilizado pelas grandes organizações da época, muito utilizado especialmente para as operações de chão de fábrica.

A década de 70 foi marcada pelo início da entrada no mercado de computadores bem mais baratos e fáceis de usar, comparados aos mainframes, o que possibilitou uma revolução nas atividades de produção e logística. Surgiu nesta época o MRP II (Manufacturing Resource Planning ou planejamento de recursos de manufatura), considerado uma evolução do MRP I. Por sua vez, o MRP II também incluía mais atividades diferentes de controle, como a gestão de custos de mão-de-obra e maquinários utilizados nos processos de produção [13].

Um fator fundamental para o surgimento do ERP foi a criação, em 1975, da empresa alemã SAP (Systemanalyse and Programmentwicklung ou no português, Análise de Sistemas e Desenvolvimento de Programas), que criou um software chamado R/2 e entrou para a história da área de ERPs sendo ainda hoje, uma das mais importantes

fornecedoras de ERP do mercado mundial [13].

Como se observa na figura 2 e no histórico relatado, o ERP é na realidade uma evolução dos MRP e MRP II.

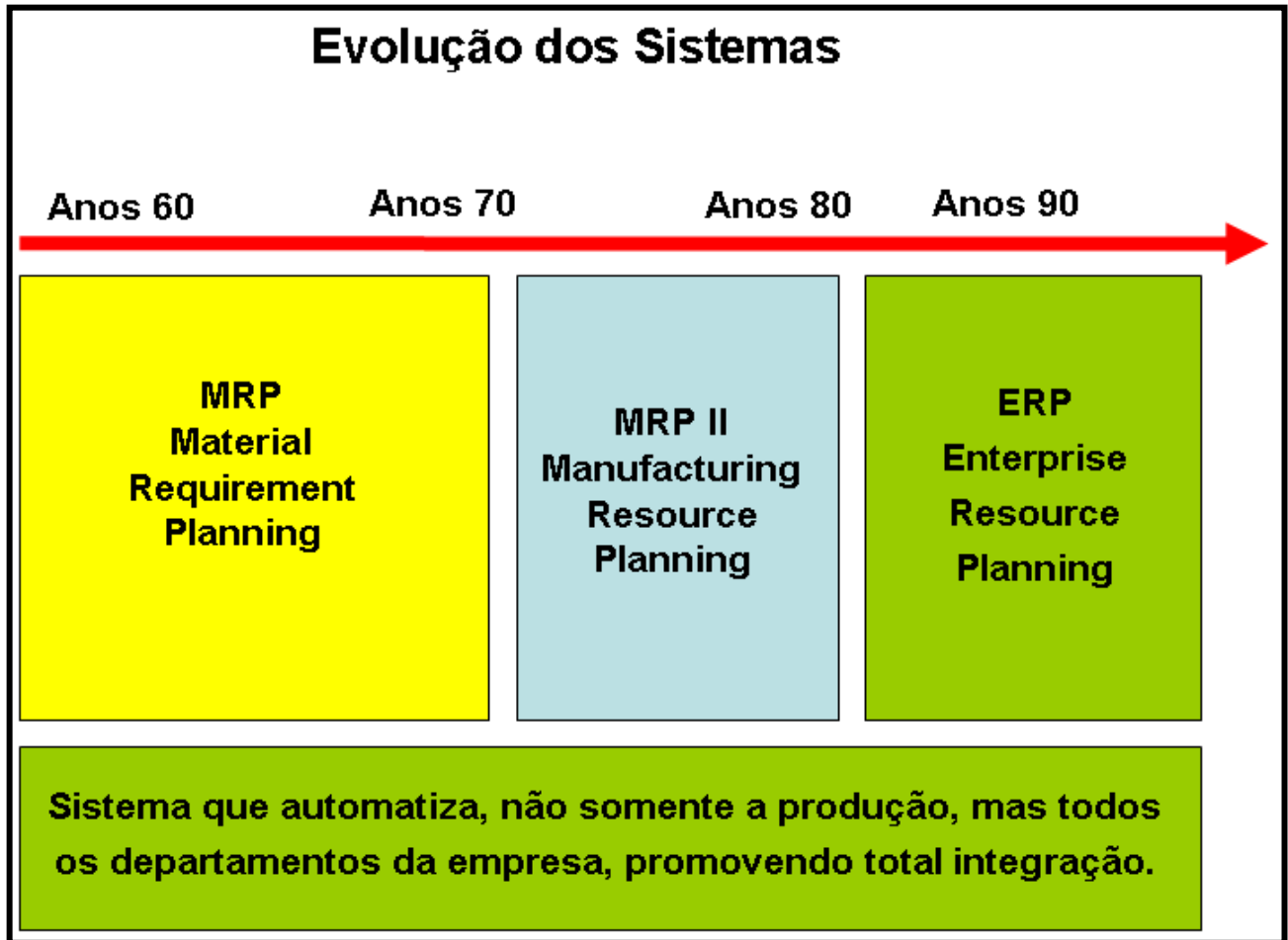


Figura 2: Evolução do Sistema ERP

Já a década de 80 serviu para consolidar o conceito de ERP como um sistema capaz de integrar e agilizar a comunicação entre os departamentos de uma organização, já que foram agregados novos sistemas como a área de finanças, compras, vendas, recursos humanos entre outras, ou seja, setores das áreas administrativas e de apoio a produção foram integrados a automação.

Somente nos anos 90 é que o ERP ganharia mais força. Consolidado como uma ferramenta importante de controle e gestão dos setores corporativos [3], foi incorporado à evolução constante das redes de comunicação entre computadores e a disseminação da arquitetura cliente/servidor, junto com o surgimento dos microcomputadores com valores mais atrativos e conectados a servidores comparados aos mainframes.

Na segunda metade da mesma década houve um grande estouro de vendas dos pacotes de gestão dos ERPs, surgindo diversos fabricantes internacionais, junto vieram fornecedores brasileiros que lucraram com suas vendas como um substituto dos sistemas que poderiam falhar devido ao bug do ano 2000.

Estas empresas cresceram influenciadas pelo bug do milênio. Este bug, problema relacionado às datas dos sistemas antes do ano 2000, que utilizavam apenas dois (2) dígitos para registrar o ano, ao invés de quatro (4). Como os sistemas existentes daquela época não eram adaptados para tal problema, isto ocasionou uma disputa de mercado e estes sistemas tornaram se vantajosos.

2.2. E.R.P. no Brasil

Devido à maioria dos sistemas de ERP serem importados, com um alto custo de importação e implantação e a dificuldade de adaptação com um idioma estrangeiro, foram surgindo no mercado nacional no início dos anos 90 empresas como a Microsig, Datasul [12].

Assim o número de fornecedores aumentou e devido à grande concorrência no setor surgiu, para os fornecedores um novo nicho de mercado, as empresas de médio e até pequeno porte. Atualmente diversos fornecedores disponibilizam aplicativos ERP para empresas focadas neste nicho de mercado [11].

Originado primeiramente à atender necessidades industriais, hoje os ERPs no Brasil, também conhecidos como SIGE - Sistemas Integrados de Gestão Empresarial, se expandem atingindo outras áreas de negócio, como comércio, finanças, serviços e outras, possibilitando uma ampliação na área de atuação ocasionando uma maior aceitação por parte das empresas.

Também é necessário esclarecer que os ERPs não são desenvolvidos para clientes específicos cabendo esta tarefa a equipe de consultoria desenvolver soluções de acordo com a necessidade de cada cliente. [12]

2.3 Principais Produtos / Fornecedores

Uma pesquisa realizada no ano de 2008 pela Fundação Getúlio Vargas (SP), sobre Administração de Recursos de Informática, apontou o cenário dos fornecedores de ERP no Brasil, com a seguinte participação:

- O grupo Totvs lidera o mercado nacional com 24%;
- Em segundo lugar está a SAP, com 23%;
- A Oracle com 17% do mercado ocupa a terceira colocação;
- A Datasul representa 16% do mercado;

O restante está dividido entre outras empresas, como se observa no gráfico 1.

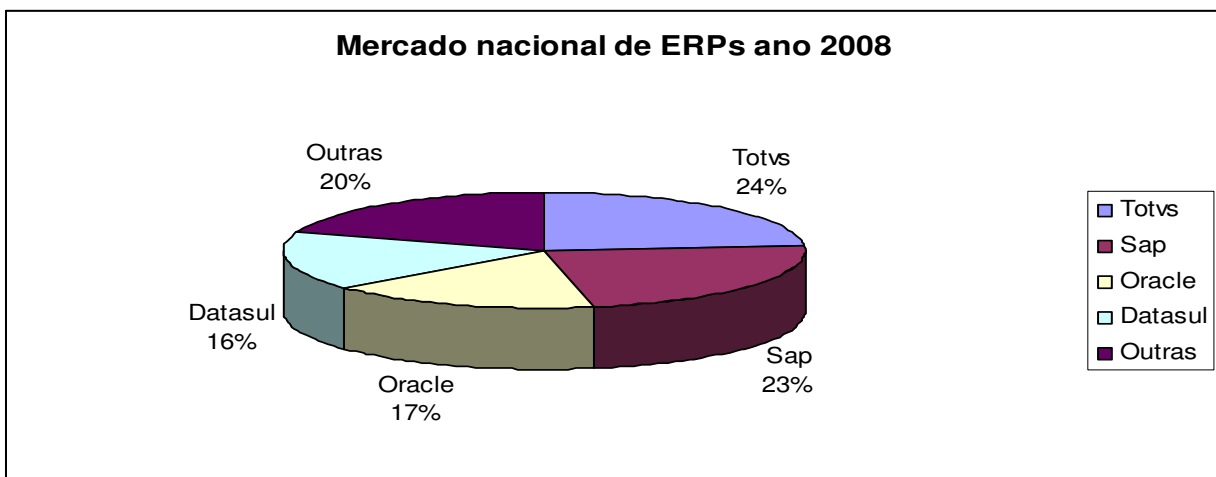


Gráfico 1: Comparativo no Mercado Nacional

Cada ERP tem suas características particulares em relação à adoção das ferramentas de depósito de dados, ou seja, das suas bases de dados. A Oracle destaca-se no cenário por fornecer além do sistema de banco de dados o sistema ERP.

Com relação ao mercado de banco de dados, a Oracle lidera o mercado nacional com 35%, registrando um aumento de 2 pontos percentuais em relação à pesquisa realizada no ano de 2007. O banco de dados da Microsoft SQL Server, fica em segundo lugar com 27% [4]. Os outros 38% estão divididos entre outros fornecedores empresas, como demonstra o gráfico 2.

Em relação aos sistemas operacionais, a pesquisa confirma a tendência de que a Novell está desaparecendo e dando lugar para um leve crescimento em Windows, Linux e Unix, proporcionalmente.

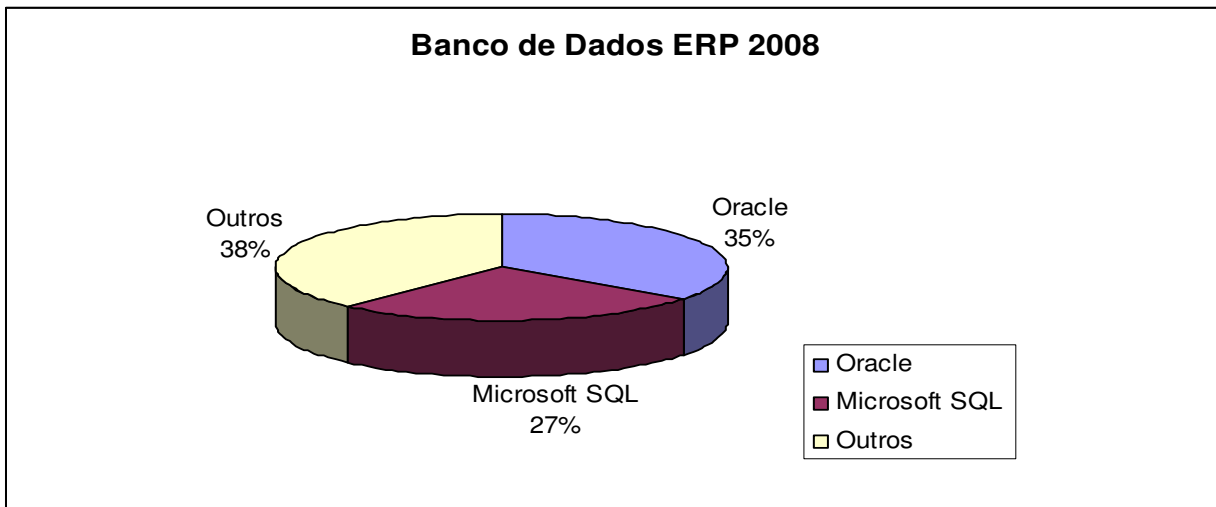


Gráfico 2: Uso de Banco de Dados pelos ERP

Quanto ao mercado nacional de ERP, Samuel Carvalho, analista da IDC, classifica como "bem provável" que o mercado nacional de ERP tenha ultrapassado a marca de US\$ 500 milhões, em 2008. Cinco fornecedores detêm 90% das receitas do setor, que deve registrar um crescimento anual de 12%, até 2012.

As grandes corporações, quase que em sua totalidade, já consolidaram seus ERPs. Mas ainda há um grande mercado a ser explorado entre as pequenas e médias. Pelas estimativas da IDC, o nicho composto por companhias que possuem entre 100 e 999 funcionários representa 50% dos gastos em tecnologia. [12]

Cada vez mais o mercado nacional de software de gestão caminha rumo à consolidação, nas mesmas proporções que os ERPs estão mais enraizados nas companhias. Alguns fornecedores vêm se consolidando no mercado com soluções para grandes e pequenas organizações. Um exemplo é a Auland, com sede em Curitiba – PR e com 50 funcionários, que está no mercado de informática a mais de 20 anos, possui cerca de 80 clientes, distribuídos entre empresas de médio e pequeno porte.

ABC71 originária da cidade de São Paulo - SP trabalha a 38 anos no mercado, com 80 funcionários e uma cartela de mais de 630 clientes ativos, tendo como público-alvo empresas do ramo de metalurgia, plásticos, químico, farmacêutico, cosméticos, autopeças, eletroeletrônicos, seu faturamento em 2008 foi de R\$ 7,5 milhões. [12]

A Oracle desenvolvedora de vários segmentos, já há 21 anos no Brasil possui mais de 320 mil clientes espalhados por todo o mundo, e tem como público atender as áreas de manufatura, varejo, construção civil, finanças, telecom, agronegócio, petroquímico.

O grupo Linx de origem brasileira há 24 anos no mercado nacional conta com 400 funcionários para atender mais de 3000 clientes, tem como foco a área de vestuário, calçados e acessórios, seu faturamento em 2008 foi de R\$ 66 milhões.

A SAP, vinda da Alemanha, é considerada a pioneira no ramo de ERP, está no Brasil há 14 anos e conta com mais de 800 funcionários no país para atender 1,9 mil clientes, atua na área de saúde, governo, óleo & gás e varejo. Em 2008 fechou o exercício com um faturamento de 11,5 bilhões de euros em números globais.

A Totvs, de origem brasileira, já atua no mercado há 26 anos conta com mais de 9 mil colaboradores e uma quantidade de clientes perto de 23 mil, entre o público alvo estão a agroindústria, saúde, finanças, jurídico, construção civil, educação e logística. Seu faturamento no ano de 2008 foi na casa de R\$ 844,854 milhões.

A Senior Sistemas, também de origem nacional, situada em Santa Catarina, atua no mercado há 12 anos, conta com 500 funcionários e mais de 1000 clientes, atua nas áreas de logística, calçadista, varejo, têxtil, saúde, agronegócios e educação. Obteve um faturamento de R\$ 267 milhões em 2008.

Dentre estas a Lawson é a mais nova no país com pelo menos seis meses no mercado nacional conta com 20 clientes locais tendo como foco empresas do ramo da moda, obteve em 2008 um faturamento de US\$ 852 milhões nível global. [12]

Existem outras fornecedoras de ERPs no mercado atualmente, todas estas citadas tem em comum ainda um grande valor de custo, tanto para implantação quanto para treinamento dos usuários. Tendo em vista estes problemas, este trabalho visa observar o funcionamento de outro tipo de ERP, robusto, proveniente de software livre, o Adempiere. Este sistema pode ser uma opção interessante para empresas de pequeno porte e ainda, a custos bastante reduzidos se comparados às opções proprietárias.

3. Adempiere

3.1 Evolução do Adempiere

O projeto do Compiere foi idealizado e desenvolvido por Jorg Janke, com 20 anos de experiência com ERPs, certificado em ferramentas Oracle e Java. Criou uma versão beta do sistema, e em 1999, o Compiere foi trabalhado e desenvolvido para atender a necessidade específica da empresa Goodyear Alemanha, de sistematizar a gestão de seus distribuidores de produtos. Sua implementação ocorreu em 2000 e foi a primeira versão do Compiere.

Após o sucesso da primeira versão do projeto Compiere, Janke abriu o código fonte e fundou o Compiere Inc., sendo o primeiro ERP open-source, líder mundial na categoria. Nos dias atuais o projeto conta com 75 desenvolvedores ativos, cadastrados no site oficial do projeto. [11]

Mesmo o projeto sendo de código aberto suas alterações são muito demoradas, pois são exaustivamente testadas, a comunidade não ativa do projeto também tem dificuldades em enviar suas idéias, sendo as decisões de atualização centralizadas para o dono do projeto, que por sua vez ele alega que as contribuições de melhorias estão disponíveis apenas aos parceiros.

Devido a estas dificuldades a comunidade de software desenvolveu um novo projeto baseado de uma versão estável do projeto Compiere, esta por sua vez batizada de Adempiere. [2]

O Adempiere surgiu em setembro de 2006 contando com um amplo conjunto de recursos, dirigidos aos processos de gestão de empresas dos mais diversos segmentos, como mostrado na figura 3.

Uma das diferenças apresentadas pelo sistema é a forma de sua organização através de processos, e não por de módulos, como é o que acontece com a maioria das soluções de ERP existentes no mercado. O sistema é uma ferramenta única, totalmente integrada, a qual permite ao usuário uma visão única e centralizada de seus negócios, possibilitando a visualização de indicadores como demonstra a figura4.

Essa integração se estende desde os recursos padrões comumente disponíveis em sistemas dessa natureza, como controle de pedidos, contas a pagar e a receber, controle de estoque etc. até os recursos de gestão de relacionamento com clientes (CRM) e Loja Web (Web Store). [11]

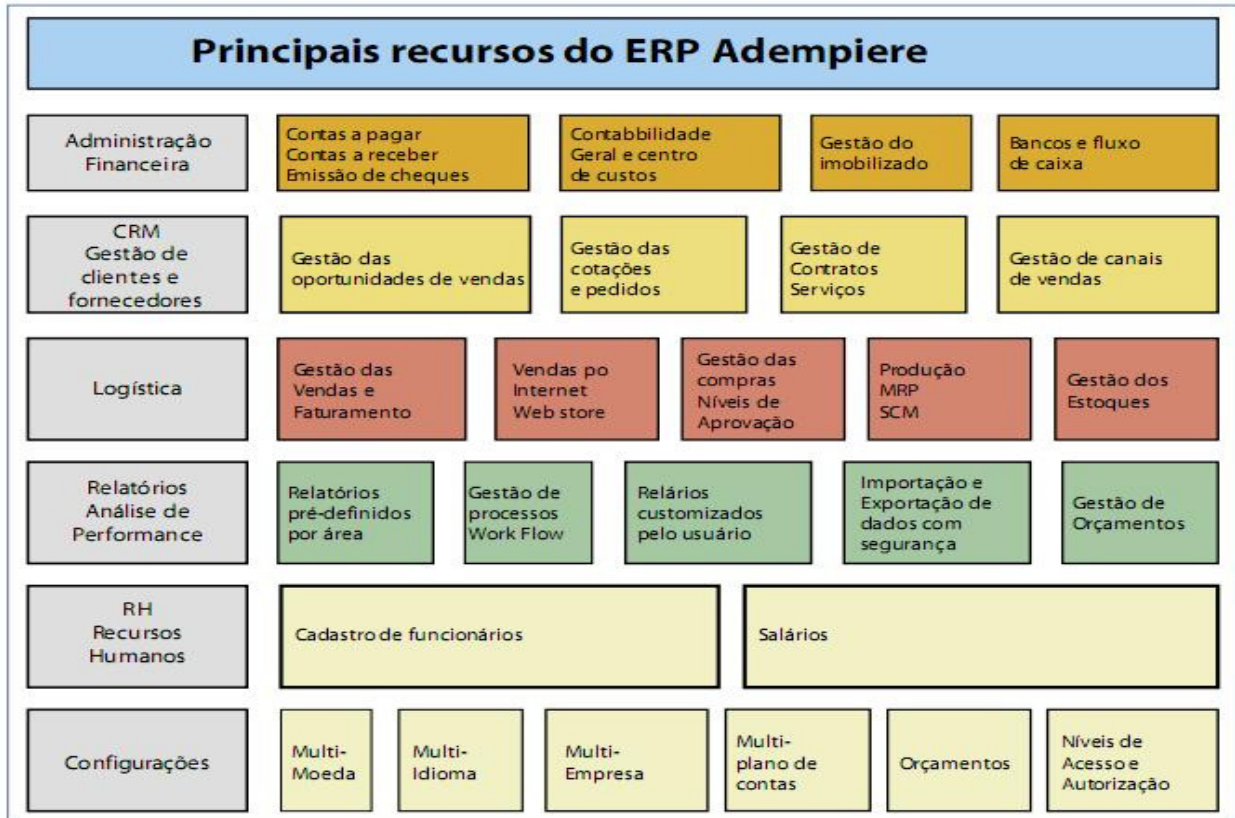


Figura 3: Estrutura Funcional do Adempiere

Desenvolvido com arquitetura baseada em modelos (MDA, Model Driven Architecture) e um dicionário de dados ativo da aplicação, fornece benefícios de utilização ao manter a lógica e a estrutura das regras de negócio em alto nível, resultando em um único conjunto de informações e código, resultando, um modelo que é automaticamente documentado. A aplicação pode utilizar as informações do dicionário para navegação dinâmica nos dados, resultando em uma extração de relatórios extremamente simplificada. Por exemplo, a identificação de quais transações (pedidos, faturas, etc.) utilizam uma determinada condição de pagamento pode ser efetuada por uma simples consulta, sem a necessidade de uma codificação lógica pré-definida.



Figura 4: Exemplo de Indicador do ERP Adempiere.

A figura 5 apresenta parte do fluxo de dados de entrada no sistema e a figura 6 demonstra o fluxo de saída.

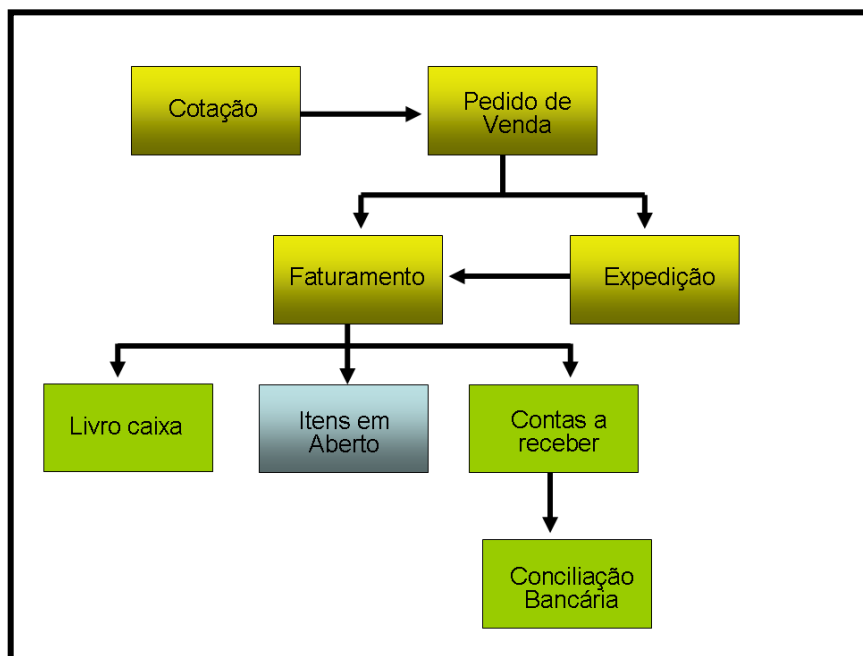


Figura 5: Fluxo do processo de entrada

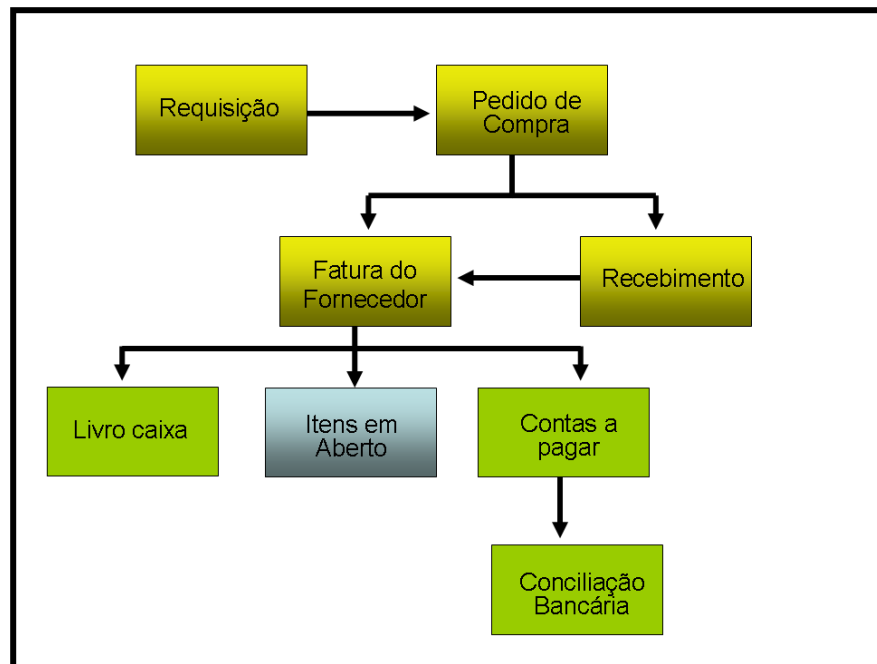


Figura 6: Fluxo do processo de saída

No Adempiere é possível manter e garantir regras de segurança de forma consistente e ainda proporcionar uma produtividade do desenvolvedor muito alta. O Dicionário ativo desenvolvido com a metodologia MDA representa a plataforma da aplicação. Assim, o próprio Adempiere é um conjunto de ferramentas para desenvolvimento de alta produtividade. [11]

3.2 Ambiente de Desenvolvimento

3.2.1 Linguagem

O Adempiere assim como o compiere é desenvolvido em Java, uma linguagem criada pela Sun Microsystems nos anos 90. Originalmente foi desenvolvida para programar eletrodomésticos como vídeos-cassete, torradeiras, etc, e por isso deveria ser, simples, confiável, compacta e independente de plataforma.

A linguagem começou a ser escrita no ano de 1991, seu primeiro nome foi Oak e foi projetada por James Gosling. Devido a explosão das redes de computadores em 1993 o projeto foi direcionado à aplicações de internet, assim assumindo também o nome de Java em 1994.

Possui muitas características que faz dela nos dias atuais ser uma das linguagens mais difundidas do mercado, como a orientação a objetos, não utilizar ponteiros, ter uma

boa performance, tratar exceções. É ótima para programar dispositivos móveis, ideal para integral com HTML e outras tecnologias para utilização na internet e principalmente, por ser livre de plataforma de execução e gratuito. [10]

Um programa escrito em Java é compilado e traduzido para uma linguagem de máquina especial, denominada bytecode. Este por sua vez não é interpretado diretamente por nenhum computador, mas por uma Máquina Virtual específica, chamada de JVM (*Java Virtual Machine* ou Máquina Virtual Java), é um programa que traduz dinamicamente o bytecode para a linguagem de máquina correspondente à plataforma em que ele está sendo executado. Assim, cada tipo de computador possui sua própria Java Virtual Machine, sendo a JVM a grande responsável pelo sucesso da linguagem. A figura 8 demonstra como é o funcionamento da JVM. [10]



Figura 7: Funcionamento da JVM.

3.2.2 Banco de Dados

O Adempiere pode ser usado com duas opções de Banco de Dados (SGBD), o Oracle e o PostgreSQL.

A Oracle é uma empresa de quase trinta anos de mercado, seu criador Larry Ellison observou uma oportunidade que outras empresas ainda não haviam percebido,

quando analisou uma descrição de um protótipo funcional de um banco de dados relacional, e que nenhuma empresa tinha tido interesse em comercializar essa tecnologia. Ellison e seus co-fundadores Bob Miner e Ed Oates, perceberam o potencial de negócios no modelo de banco de dados relacional, mas sem dar conta de que mudariam a face da computação empresarial para sempre.

Hoje a tecnologia da Oracle pode ser encontrada em quase todos os setores de atividades comerciais e de serviços. Sendo a principal fornecedora de software para gerenciamento de informações e a segunda maior empresa de software independente do mundo. Entre seus produtos a Oracle disponibiliza uma versão XE (*Express Edition*) que é limitada em questão de alguns recursos, como o limite de 4GB de espaço para armazenar informações, que pode ser utilizada para a instalação da base de dados do Adempiere. Por essa limitação a instalação desta versão de banco de dados só é aconselhada para fins de acadêmicos ou testes, já que uma versão completa desta ferramenta tem um valor um pouco elevado. [9]

A outra solução disponível em nível de SGBD para o Adempiere é o PostgreSQL, uma ferramenta gratuita e sem limitações de uso, sendo um poderoso sistema gerenciador de banco de dados objeto-relacional com mais de 15 anos de desenvolvimento ativo e uma arquitetura que comprovadamente ganhou forte reputação de confiabilidade, integridade de dados e conformidade a padrões [8].

Para este estudo a ferramenta de banco de dados a ser utilizada será o PostgreSQL. Com a execução deste trabalho será possível também fazer uma análise deste banco de dados, entender sua estrutura, facilidade de uso e configuração, que será explanada nas próximas seções.

3.2.3 Software Livre

A principal característica do Adempiere é que foi desenvolvido sob o conceito de software livre. Esta é uma vantagem significativa, para tanto é importante se entender o contexto do Software Livre. O movimento organizado deste ambiente teve início em 1983, quando Richard Stallman deu início ao Projeto GNU e, posteriormente, à Free Software Foundation. Conforme a definição de software livre criada pela Free Software Foundation diz que o software livre é o software que pode ser usado, copiado, estudado, modificado e redistribuído sem restrição. A forma usual de um software ser distribuído livremente é

sendo acompanhado por uma licença de software livre (como a GPL), e com a disponibilização do seu código-fonte. [5]

Existem quatro liberdades predominantes associadas ao software livre, que se identificam como: a liberdade de executar o programa, para qualquer propósito, a liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades disponibilizando o acesso ao código-fonte como um pré-requisito, a liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo, e a liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie. [5]

Um programa só é considerado mesmo um software livre se o usuário tem todas essas liberdades. Portanto, deve ser livre para redistribuir cópias, seja com ou sem modificações, seja de graça ou cobrando uma taxa, para qualquer um em qualquer lugar que seja, entre outras palavras, ser livre para fazer o que quiser sem ter que pedir permissão ou pagar por isso, uma vez que tenha posse do programa. Para que essas liberdades sejam reais, elas têm que ser irrevogáveis. [6]

4. Descrição do Ambiente Adempiere

Este trabalho tem como objetivo o estudo e análise da organização do ambiente da solução Adempiere, bem como da organização do seu código fonte e de sua estrutura, além de uma incursão pelo banco de dados. Como se trata de um software já em uso por muitas organizações, muitos estudos e pesquisas vem sendo realizadas sobre este software.

Por meio de pesquisas realizadas em artigos e matérias disponibilizadas principalmente na internet, é possível observar que o software pode ser utilizado também no ambiente acadêmico para o ensino e uso de um aplicativo comercial. Como já relatado anteriormente, trata-se de um software livre, os códigos fontes tanto do sistema como do banco de dados estão disponibilizados na rede e podem ser utilizados para o estudo de ambos.

Desta forma, este trabalho objetiva realizar um estudo da estrutura do Adempiere e mapeá-lo de forma que possa ser utilizado nos cursos de computação. Fazendo uma analogia com a medicina, este trabalho pretende disponibilizar o Adempiere como um corpo a ser estudado, onde os alunos dissecarão toda a sua estrutura e funcionalidade. Por tratar-se de um software consolidado no mercado, com inúmeras organizações utilizando-o, é um ambiente adequado para os alunos possam, ainda dentro da graduação, experimentar e testar sua estrutura.

Estudar um ambiente em funcionamento pode trazer grandes benefícios para os alunos, pois pegar um projeto já iniciado e identificar sua estrutura não é tão fácil como pode parecer. Além disso, será estudado como é caracterizado o padrão do código, sendo este um grande desafio por se tratar de algo existente e de pouca literatura na língua nacional.

Para este projeto será utilizada a última versão estável, baixada diretamente do site do projeto para execução. Será utilizada também uma IDE freeware chamada Eclipse, para a análise da estrutura do código. O banco de dados utilizado será o PostgreSQL e o sistema operacional Microsoft Windows XP.

4.1 Padrões de Projeto

O Adempiere foi desenvolvido seguindo o padrão MDA (Model Driven Architecture), que será descrito na próxima seção. Nas palavras de *Metsker* “Um padrão é uma maneira de fazer algo, ou de buscar um objetivo. Tal idéia se aplica a cozinhar, fazer fogos de artifício, desenvolver software e qualquer outro ofício.” [14(METSKER, 2004, p.17)].

Devido à preocupação de se produzir softwares com os maiores índices de qualidade, há vários anos a engenharia de software vem construindo técnicas de desenvolvimento, sempre com a objetividade de se ter um software livre de erros, de uma manutenção fácil e bem organizada, enfim um software que atenda todas as especificações para que o foi projetado, obviamente sendo excluída a possibilidade de má utilização do software pelo usuário.

Mesmo com a ajuda da engenharia de software é muito difícil de conseguir um software totalmente livre de defeitos, mas as técnicas são fundamentais na organização da construção das aplicações, é altamente recomendada a grandes projetos pois podem auxiliar em um possível reuso de código fonte, expansão do software, fácil integração de novos programadores no projeto.

Padrão não necessariamente deve ser aplicado à área de informática, mas sim em problemas variados de nosso cotidiano diário. O simples fato de executarmos uma tarefa sempre cumprindo uma seqüência de ações nos leva a criar um padrão que pode ser entendido e utilizado por outras pessoas que conheçam aquela rotina.

Christopher Alexander, citado por (METSKER, 2004, p.17) foi um dos primeiros escritores a encapsular as melhores práticas de um ofício por meio da documentação de seus padrões. Seus trabalhos estão relacionados à arquitetura de edifícios, não de software.

“[...] Alexander deixa claro que seus padrões o ajudam a servir e inspirar as pessoas que irão ocupar as devidas construções.” [14(METSKER, 2004, p.18)].

Segundo esta afirmação, é possível comparar e relacioná-la com o projeto de um software, em que nele se procura atingir um objetivo: que o software seja de suma importância para nossos clientes, cumprindo com os prazos e objetivos para conseguir uma aprovação em um nível mais satisfatório possível.

A comunidade de software fez ressoar a abordagem de Alexander e criou muitos livros que documentam padrões de desenvolvimento de software. Estes livros registram as melhores praticas para processo de software, análise de software e projetos de alto nível e no nível de classe. Um padrão de projeto é um padrão uma maneira de alcançar um objetivo o que utiliza classes e seus métodos em uma linguagem orientada a objeto. [14(METSKER, 2004, p. 18)].

Na maioria das vezes os desenvolvedores preferem soluções que aparentemente são mais fáceis, como o acesso direto ao código fonte para solucionar seus problemas, devido a terem algum tipo de experiência com algum tipo de linguagem ou por já terem alguma experiência de programação esquecendo que adotando um padrão de projeto se torna mais fácil a compreensão como um todo e facilitando assim a solução de problemas com o projeto.

Devido ao aumento e complexidade das aplicações desenvolvidas nos dias de hoje, adotar um padrão de projeto torna-se inevitável, pois dependem de características como herança e polimorfismo, e que tudo seja abordado igualmente em todo o projeto de software, de maneira a contribuir com o andamento e cumprimento dos objetivos.

4.2 Padrão MDA

A arquitetura *Model Driven Architecture* foi proposta pelo *Object Management Group* (OMG), e vem a reconhecer a importância dos modelos em um processo, sendo ponto chave no desenvolvimento de um projeto, nele é definido que seu desenvolvimento deve-se ter um direcionamento na modelagem de nível conceitual, não dependendo de nenhum tipo de plataforma ou implementação, assim realizando transformações no modelo conceitual, sejam produzidos novos modelos com níveis de abstração mais específicos e ligados a implementação possam ser gerados, de forma que ao final o sistema possa ser gerado automaticamente, partindo do modelo conceitual. Os modelos construídos são mais formais, o que evita ambigüidade, MDA vem a ser uma especificação a apoiar o desenvolvimento dirigidos por modelos *Model-Driven Development* (MDD), não sendo uma técnica recente, mas que ganhou destaque após o OMG lançar a especificação da MDA no ano de 2001. A arquitetura do MDA é baseada nas seguintes etapas [20].

O primeiro conceito para se usar a arquitetura MDA é a geração do modelo *Computation Independent Model* (CIM), construindo uma visão do sistema com um ponto-de-vista independente de computação. O CIM apresenta somente os requisitos do sistema sem mostrar os detalhes da sua estrutura. Apesar de constar na especificação

MDA, o CIM ainda é um conceito pouco usado sendo omitido em grande parte pela maioria das ferramentas e publicações sobre MDA.[21]

O segundo conceito é o da geração de um novo modelo a partir do CIM, o modelo agora é chamado de *Platform Independent Model* (PIM), sendo definido em um alto grau de abstração, independente de qualquer tipo de plataforma ou tecnologia, responsável por definir a melhor maneira de se modelar o negócio.[21]

Após ser gerado o PIM, o terceiro conceito é o da geração de um ou mais *Platform Specific Models* (PSMs). Os PSM são gerados especificamente levando em conta tecnologias específicas a serem utilizadas para a implementação, podendo ser gerado para cada PIM existente vários PSMs.

Concluindo, o último passo tem como objetivo a geração de código, a partir de cada PSM. Esta geração de código não apenas constrói estruturas básicas, o código gerado no MDA deverá ser o mais próximo possível da solução definitiva de implementação do software, incluindo suas regras de negócio. A Figura 8 demonstra as etapas da arquitetura MDA.

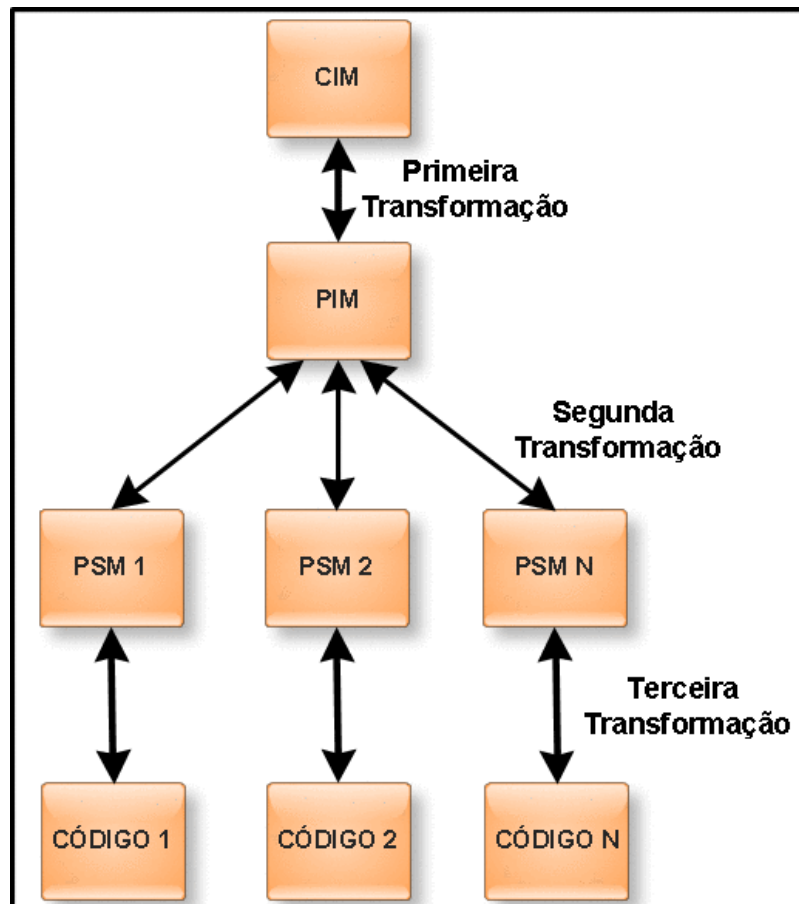


Figura 8: Passos da Especificação MDA.

Entre as principais vantagens do uso do MDA pode-se citar a produtividade, pois mesmo que na geração e transformação entre os modelos PIM, PSM e do código, é realizada por apenas uma vez para cada plataforma utilizada, uma vez bem definidas as regras de negócio, as transformações se tornam automáticas notando-se que apenas o modelo de conceito PIM deve ser alterado enquanto os outros modelos são atualizados automaticamente reduzindo o tempo no desenvolvimento do projeto.

Outra grande vantagem é a portabilidade, já que o PIM é por definição independente de tecnologia ou plataforma, gerando um único PIM para diversos PSMs. A interoperabilidade também é possível através do uso de bridges que utilizam códigos de diferentes plataformas entre os PSMs, mas para que isso ocorra, além de gerar os PSMs as ferramentas de transformação devem gerar bridges entre os PSMs.

Sua manutenção e documentação também não é feita no código do projeto e sim no nível mais alto do modelo (PIM), desta maneira a manutenção da documentação é constantemente atualizada, caso ocorra de uma manutenção ser realizada diretamente em um PSM, é possível que tenha atualização automática do PIM deixando os modelos consistentes.

4.3 Organização do Código do Adempiere

O Adempiere possui um diretório de aplicativos denominado AD (*Application Directory*). Este diretório é usado extensivamente permitindo criar novas entradas, telas do sistema e modificar as já existentes. Neste contexto, uma tela é uma janela criada usando o diretório de aplicativos, isto significa que sua estrutura é armazenada no diretório de aplicativos. As telas são criadas utilizando a classe *org.compiere.apps.AWindow*.

A tela é um quadro que tem um menu padrão. As abas da janela são definidas no AD. Os separadores são definidos pelo *org.compiere.grid.GridController* que por sua vez cria campos usando a classe *org.compiere.grid.ed.VEditorFactory*.

Os retornos obtidos pela classe *VEditorFactory.class* são de diferentes tipos, dependendo do que o campo editado representa, ou seja, se é definido um campo do tipo data, um retorno do tipo *org.compiere.grid.ed.VDate* é devolvido. Se o campo for numérico, um retorno do tipo *org.compiere.grid.ed.VNumber* é devolvido assim sucessivamente. A figura 9 mostra um exemplo de tela de cadastro de produtos.

Figura 9: Modelo Tela – Cadastro de Produtos

Na figura 9 os itens destacados em vermelho permite analisar os atalhos padrões e a barra de menus criados junto com a tela, que facilita o acesso aos recursos do Adempiere, pois são encontrados em todas as telas do sistema.

Em contraste com as telas os formulários (forms) são gerados de maneiras diferentes em comparação com as telas, uma vez que não são definidos no diretório de aplicativos. Estruturalmente, os forms são encontrados no pacote do código fonte *org.compiere.apps.form*. pois são janelas de propósitos específicos para gerar relatórios, não sendo práticos para serem criados como se fossem uma tela comum. A figura 10 demonstra um modelo de formulário.

Arquivo Visualizar Ir Ferramentas Janela Ajuda

Fatura Mensal de Produtos

ADempiere Fatura Mensal de Produtos Página 1(1,1) de 2(1,2)

Data Faturada	Margem	Margem %	Val. Linha	Desconto	Desconto %	Valor Limite	Valor da Lista	Produto
01/01/2003	30,00	5,26	570,00	30,00	5,0	540,00	600,00	Rose Bush
01/02/2002	-250,00	-125,0	200,00	300,00	60,0	450,00	500,00	Plum Tree
01/02/2002	7,50	5,26	142,50	7,50	5,0	135,00	150,00	Plum Tree
01/08/2002	3,25	5,26	61,75	3,25	5,0	58,50	65,00	Oak Tree
01/01/2003	42,00	11,76	357,00	63,00	15,0	315,00	420,00	Lawn Tiller
01/08/2002	2,25	5,26	42,75	2,25	5,0	40,50	45,00	Planting Service
01/01/2003	30,00	11,76	255,00	45,00	15,0	225,00	300,00	Hoe 4 ft
01/01/2003	24,00	11,76	204,00	36,00	15,0	180,00	240,00	Grass Seeder
01/01/2003	48,75	5,26	926,25	48,75	5,0	877,50	975,00	Oak Tree
01/01/2003	13,50	11,76	114,75	20,25	15,0	101,25	135,00	Rake Bamboo
01/01/2003	7,50	5,56	135,00	15,00	10,0	127,50	150,00	Mulch 10#
01/01/2003	18,75	5,26	356,25	18,75	5,0	337,50	375,00	Azalea Bush
01/01/2003	20,00	5,26	360,00	20,00	5,0	360,00	400,00	Holly Bush
01/01/2003	30,00	5,26	570,00	30,00	5,0	540,00	600,00	Plum Tree
01/01/2003	40,00	5,56	720,00	80,00	10,0	680,00	800,00	Fertilizer #50
01/08/2002	2,50	5,26	47,50	2,50	5,0	45,00	50,00	Plum Tree
01/01/2003	9,00	11,76	76,50	13,50	15,0	67,50	90,00	Weeder
01/09/2002	120,00	33,33	360,00	240,00	40,0	240,00	600,00	Oak Tree
01/11/2003	11,30	5,23	215,90	11,40	5,02	204,60	227,30	Azalea Bush
01/01/2003	45,00	5,26	855,00	45,00	5,0	810,00	900,00	Elm Tree
01/01/2003	96,00	16,67	576,00	384,00	40,0	480,00	960,00	Grass Seed Container
01/01/2003	24,00	11,76	204,00	36,00	15,0	180,00	240,00	Rake Metal
01/01/2003	10,50	11,76	89,25	15,75	15,0	78,75	105,00	Transplanter

na-letter - 8.5x11.0 " ((0.5,0.5)->(7.5,10.0))" - Paisagem - Colunas de Dados=-1, Linhas de Dados=23

Página 1(1,1) de 2(1,2)

Figura 10: Modelo Form – Fatura Mensal de Produtos

Como se observa o uso de um AD como referência a produção de telas e formulários, demonstra que a arquitetura MDA é fundamental no desenvolvimento do Adempiere, pois cada tela ou formulário usa da mesma classe para ser obtida, é possível que se construa usando esta arquitetura telas com as mesmas funcionalidades com tecnologias diferentes. A figura 11 mostra a tela principal do sistema em formato web, podendo o sistema ser configurado também para funcionar via internet, tornando o sistema muito mais compatível com outras tecnologias.

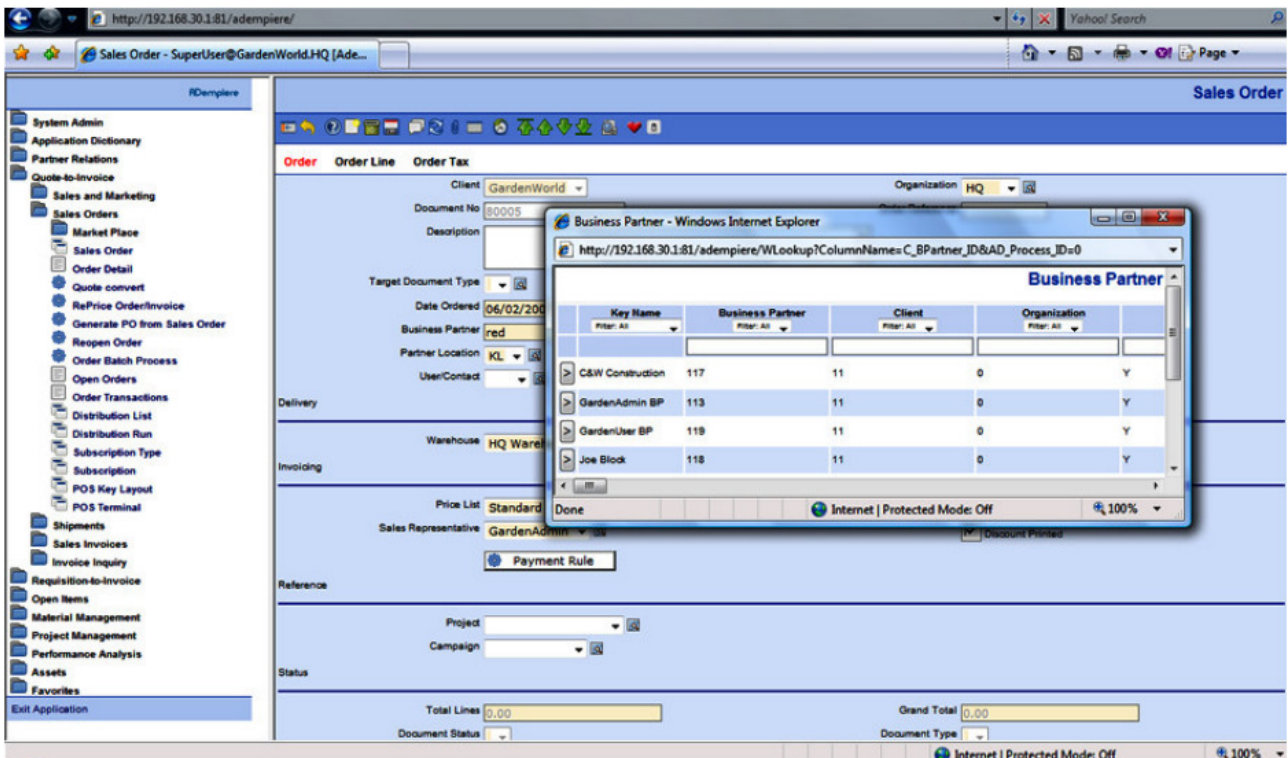


Figura 11: Tela Principal – Apresentação Web

4.4 Tipos de Dados

Cada campo em uma aba de uma tela tem um tipo de dado associado. O tipo de dado determina como o campo será representado na tela. Como mencionado, o editor apropriado para cada campo é selecionado pelo *org.compiere.grid.ed.VEditorFactory*, que por sua vez aponta para o tipo de dado. Os tipos de dados são definidos na tabela *AD_Reference* do banco de dados. Cada tipo de dados tem um tipo de dados correspondente definido no *org.compiere.util.DisplayType*.

Para utilização do banco de dados são estabelecidos prefixos (AD_, ASP_ CM_) que são usados no conceito organizacional, facilitando a manutenção e o entendimento da estrutura do banco de dados, conforme pode-se observar no quadro 1.

Prefixos da tabela de banco de dados	
Sigla	Descrição
AD	Application Dictionary (Dicionário da Aplicação)
A	Assets Management (Gerenciamento de Ativos)
ASP	Application Service Provider (Provedor de Serviço de Aplicação)
B	Marketplace ()
C	Common or Core Functionality (Comum ou Funcionalidade)
CM	Collaboration Management (Colaborador de Gestão)
FACT	Multi-Dimensional Cube()
GL	General Ledger (Registro Geral)
HR	Human Resource (Recurso Humano)
I	Import (Importação)
K	Knowledge Management (Gestão do Conhecimento)
M	Material Management (Gestão de Material)
PA	Performance Analysis (Análise de Desempenho)
PP	Production Planning (Planejamento da Produção)
R	Requests (Pedidos)
RV	Report View (Relatório)
S	Service (Serviço)
T	Temporary Tables (Tabelas temporárias)
W	Web
WS	Web Service

Quadro 1: Prefixos da Tabela de Banco de Dados Adempiere

Alguns campos são padrões nas tabelas do banco de dados da aplicação, o que dá muita segurança em utilizar o sistema, pois estes campos fornecem informações importantes, pois garantem a unicidade de cada elemento de dado e são destacados como mostra o quadro 2 e a figura 12.

Colunas Padrões das Tabelas	
Campo	Tipo de Dado
AD_Client_ID	Number(10) NOT NULL
AD_Org_ID	Number(10) NOT NULL
IsActive	Char(1) - Valor: Y/N
Created	Date NOT NULL
CreatedBy	Number(10) NOT NULL
Updated	Date NOT NULL
UpdatedBy	Number(10) NOT NULL

Quadro 2: Colunas Padrões das Tabelas

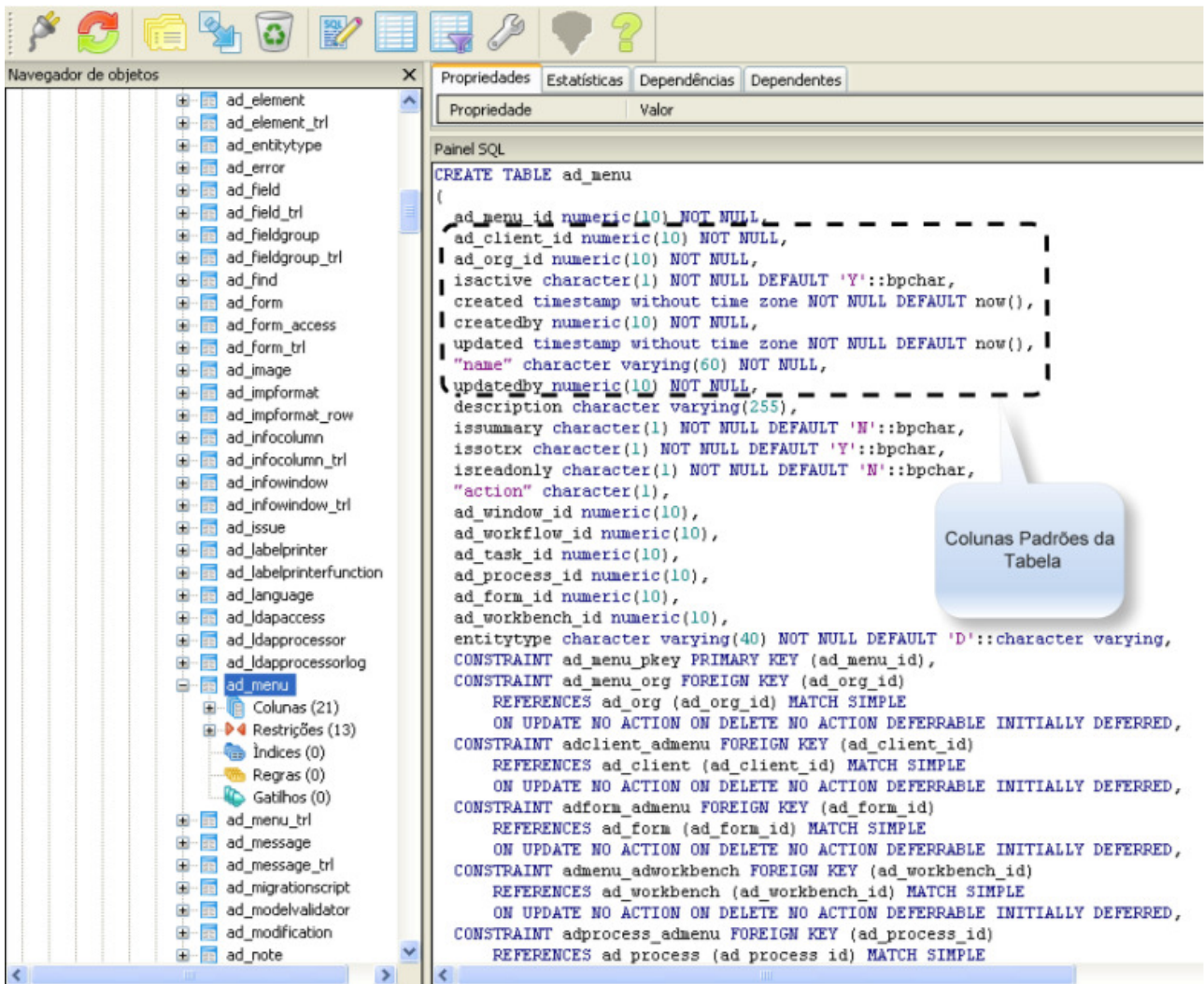


Figura 12: Colunas padrões do banco de dados – tabela ad_menu

A utilização de campos padrões nas tabelas do banco nos transmite maior segurança em relação ao funcionamento do sistema, pois estes campos são responsáveis pela autenticação das informações do cliente e nela também são gerados os arquivos de log (alterações) executadas em uma determinada tela.

4.5 Organização

Os tópicos anteriores demonstram parte da organização do Adempiere e como se observa o sistema é organizado de forma clara e concisa, o que ajuda e muito no processo de aprendizagem, pois é possível encontrar um padrão e até certo ponto uma rotina de trabalho.

Na tela de menu principal estão localizados a maior parte das funcionalidades do sistema, apresentados nas Figuras 13 e 14, de onde é possível acessar a maioria das configurações do sistema.

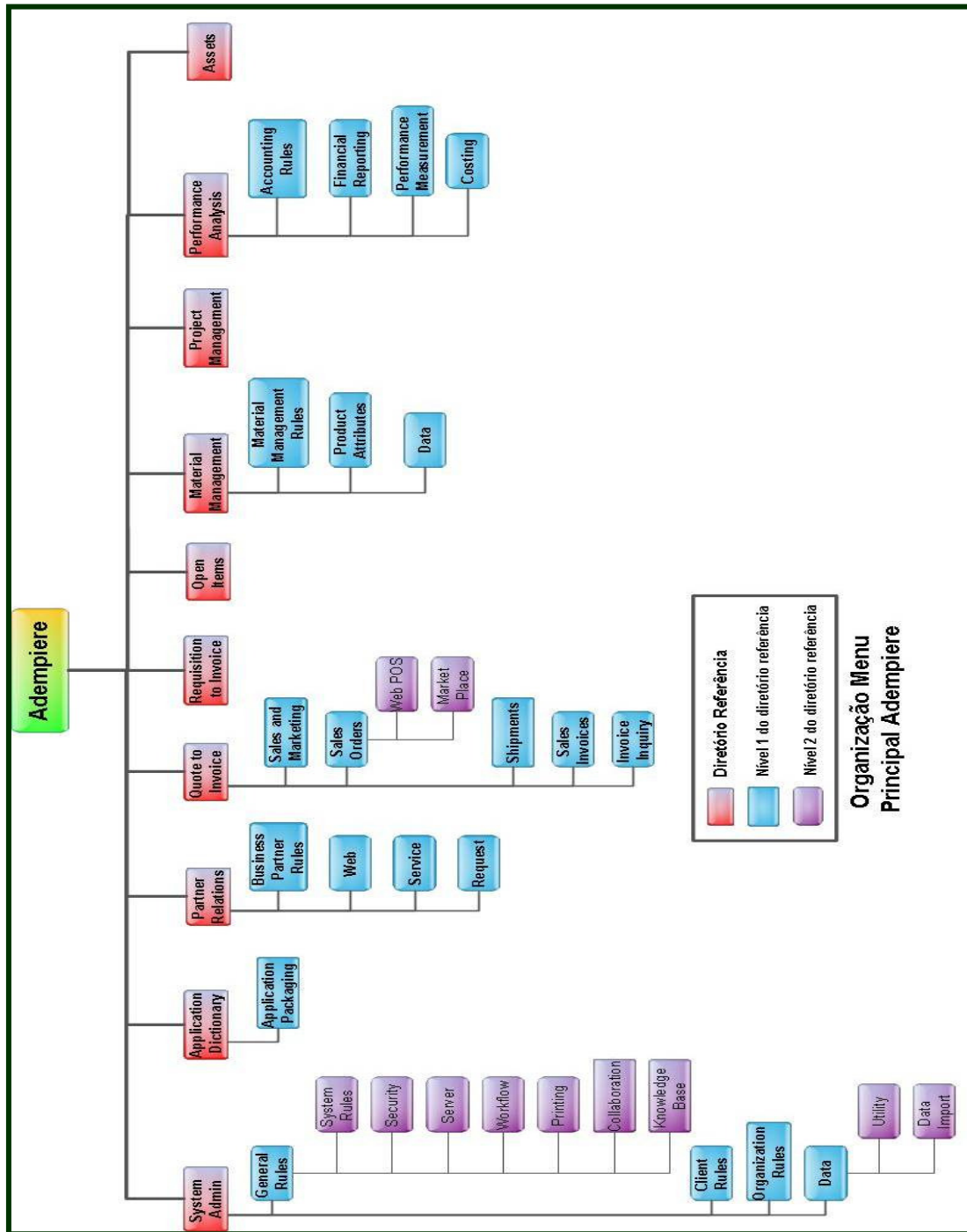


Figura 13 - Funcionalidades do Adempiere

A mesma estrutura pode ser vista em forma de árvore, também no menu principal do sistema (figura 14). Como exemplo, a tela apresentada na figura 15 é acessada da seguinte forma: Menu → Administrador do Sistema → Regras Gerais → Regras do Sistema, MENU.

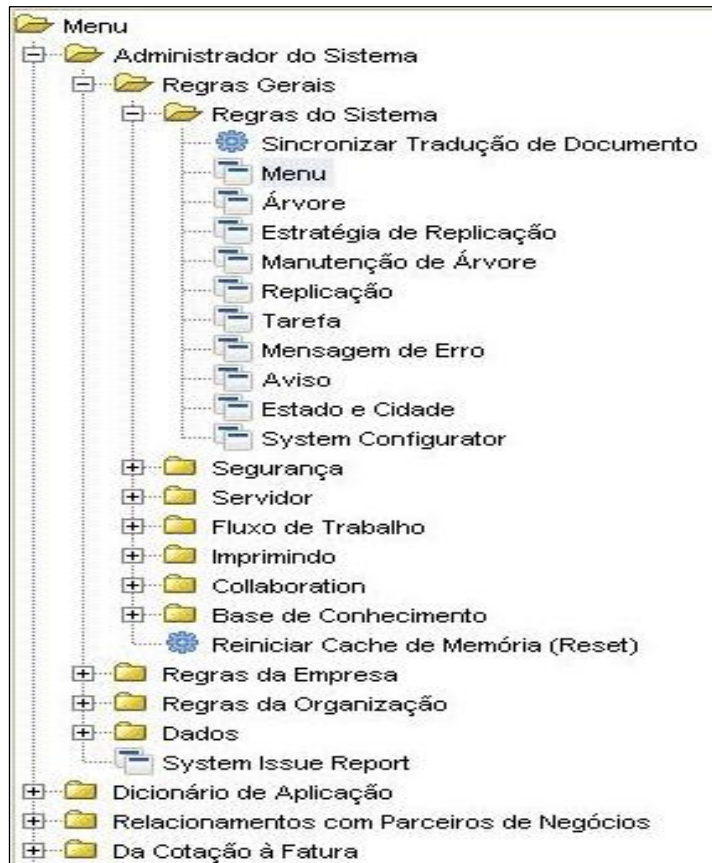


Figura 14: Arvore da Organização do Menu Principal

Esta opção permite que se crie e configure outras telas do sistema.

Figura 15 – Tela responsável pela configuração de menus

A tabela de referencia a esta tela é a AD_Menu, ela é responsável pela estrutura da organização e da arvore estrutural que será utilizada para a Organização selecionada. Não serão apresentados, neste trabalho, detalhes dos campos e ou estrutura das telas, mas para facilitar o entendimento, excepcionalmente serão detalhados os campos desta tela:

- Empresa (*Client*) Um cliente é uma empresa ou uma entidade jurídica, não podendo entre varias Empresas compartilharem dados entre si.
- Organização (*Organization*) são entidades que pertencem ao cliente como lojas, departamentos. Nome (*Name*) utilizado para pesquisa padrão de Menus salvos.
- Descrição (*Description*) é um campo opcional para descrição das configurações.
- Ativo (*Active*) existem duas modalidades de registro disponíveis no sistema, a exclusão e a desativação do registro, quando o registro é desativado, ele se torna indisponível para uma seleção em uma pesquisa, podendo ser acessado apenas por meio de relatórios. Há duas razões para a desativação e não exclusão de registros. Primeiro que o sistema exige o registro para fins de auditoria, e segundo se o registro for referenciado por outros registros, por exemplo, não se pode excluir um cliente, se há faturas para este cliente existentes no sistema, desta forma é possível desativar o cliente e impedir que este registro seja utilizado para vendas futuras.
- Tipo de Entidade (*Entity Type*) determina qual a propriedade e sincronização pertence (aplicativos, dicionários, extensões).
- Somente Leitura (*Read Only*) significa que o registro está habilitado apenas para consulta e não poderá ser atualizado ou alterado enquanto estiver marcado.
- Ação (*Action*) este campo habilita uma opção a ser executada pelo item (formulário, processo, relatório) que deverá ser escolhida conforme a necessidade.
- Nível de Resumo (*Resume Level*) representa a síntese de uma filial em uma arvore ao invés de um fim de um nó.

4.6 Estrutura do Código

Todo software de nível comercial deve ter uma arquitetura que suporte as ferramentas com as quais irá se relacionar, como linguagens, bancos de dados, ambiente corporativo, manutenções, etc. Além disso, deve obedecer as diretrizes que regem as linguagens de programação, desta forma o Adempiere é um caso de total padronização, pois seu código fonte obedece aos requisitos que a linguagem Java determina, como organização via pacotes, código comentado para geração de documentação, etc.

Sua estrutura de código é organizada por pastas e dentro destas pastas encontram-se os pacotes (*packages*) de onde estão localizados os códigos do sistema. Alguns destes pacotes são o *base/src*, *jbossfact/src*, *client/src*, apresentados na figura 16. Como já citado anteriormente, não serão detalhadas as características do Adempiere, por duas razões: não fazem parte dos objetivos do trabalho e principalmente, pela dimensão do produto. Seria necessário maior período de tempo e de estudo para documentar tais características de forma detalhada, por isso limitaremos a ilustrar pequenos exemplos do sistema, como os relatados a seguir:

- O pacote *base/src* é basicamente responsável por guardar os beans do projeto, que são as classes que possuem os *gets* e *sets* do Adempiere.
- Já o pacote *jbossfact/src* é responsável por ter em seu conteúdo as classes que fazem a comunicação do servidor web com o aplicativo.
- O *client/src* é responsável pelo interfaceamento do usuário com o sistema, pois é onde está localizado os pacotes de criação de telas e forms, é nele também que se encontram as classes responsáveis pelas regras de negócio do Adempiere.

Pode-se notar uma curiosidade em sua organização, sendo um projeto derivado do Compiere, o Adempiere na sua estrutura demonstra isto de forma clara, pois os pacotes que estavam estáveis do projeto compiere não foram alterados e é possível encontrar vários pacotes com a descrição compiere em seu nome.

O trecho de código da figura 17 demonstra a padronização do código quanto as características do dicionários de aplicação o AD, pode-se verificar os prefixos que antecedem a declaração das variáveis como o *m_checks*, *m_batch*, *m_C_BankAccount*, que fazem referencia a tabela ao AD *C_ → Common* e *m_ → Material Management*.

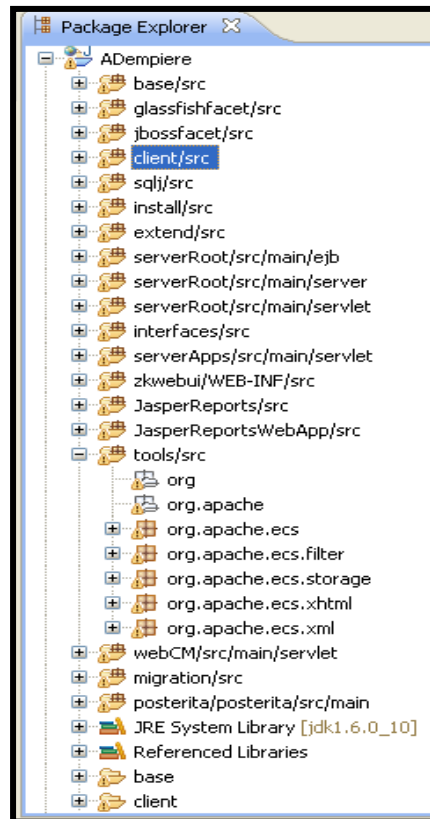


Figura 16: Organização do Código Fonte

```

import java.math.BigDecimal;

public class PayPrint {

    /** Window No */
    public int m_WindowNo = 0;
    /** Used Bank Account */
    public int m_C_BankAccount_ID = -1;

    /** Payment Information */
    public MPaySelectionCheck[] m_checks = null;
    /** Payment Batch */
    public MPaymentBatch m_batch = null;
    /** Logger */
    public static CLogger log = CLogger.getLogger(PayPrint.class);

    public ArrayList<KeyNamePair> getPaySelectionData()
    {
        ArrayList<KeyNamePair> data = new ArrayList<KeyNamePair>();

        log.config("");
        int AD_Client_ID = Env.getAD_Client_ID(Env.getCtx());

        // Load PaySelect
        String sql = "SELECT C_PaySelection_ID, Name || ' - ' || TotalAmt FROM C_PaySelection "
            + "WHERE AD_Client_ID=? AND Processed='Y' AND IsActive='Y' "
            + "ORDER BY PayDate DESC";

        try
        {
            PreparedStatement pstmt = DB.prepareStatement(sql, null);

```

Figura 17: Estrutura do Código Fonte

5. Conclusão

No transcorrer do desenvolvimento deste trabalho foi possível concluir que o mercado tende a evoluir no ramo dos ERPs, não mais no sentido de oferecer soluções para grandes empresas, mas sim buscando se expandir no nicho das médias e pequenas empresas.

Por muitas vantagens já citadas de que os softwares livres nos proporcionam, as empresas que optarem por adquirir soluções como estas obterão menores custos de implantação e poderão investir em outras áreas, sem depender de fornecedores e de altas taxas de manutenção. É possível dizer ainda que uma empresa que tenha uma boa equipe de TI possa por si mesma adaptar a solução de acordo com suas necessidades.

Uma vantagem interessante desta solução é a possibilidade de troca de prestadora de serviço, por se tratar de um software livre, o que não é possível acontecer com fornecedoras de código fechado.

Conceitos adicionais importantes foram estudados para o desenvolvimento do projeto, como o da arquitetura MDA, provou-se que para se ter sucesso em uma atividade é necessário ter um padrão a ser seguido, não por questões de linguagem de programação, mas sim em um conceito organizacional de produção, pois torna mais fácil a inclusão de novos colaboradores em projetos já existentes.

Apesar de o código ser muito extenso, e inicialmente ter encontrado dificuldades para entender a sua organização, após o estudo dos conceitos relativos a organização foi possível notar com mais clareza que não é muito difícil notar como as coisas funcionam dentro dele, e também como um bom conhecimento de linguagem Java é possível começar a implementar e a alterar de acordo com as próprias necessidades.

O projeto foi todo desenvolvido utilizando a tecnologia de máquinas virtuais, para tanto foi utilizado o *virtual box* da *Sun Microsystems*, a idéia de se desenvolver em meio a uma máquina virtual é a de disponibilizar não só um trabalho manual escrito, mas sim algo pronto de fácil acesso para que se possa pular processos relativos a configuração e instalação que demandam muito tempo.

Este processo de virtualização permitiu gerar uma mídia, anexo a este trabalho, que será disponibilizado com a instalação do projeto Adempiere, a IDE Eclipse com o código fonte importado do projeto na internet, instalação do banco de dados PostgreSQL,

e outros aplicativos que ajudam a configuração do mesmo. A princípio a idéia da mídia é de usá-la como uma matriz de segurança, possibilitando a quem a utilize testar e analisar sem medo de danificar o código fonte ou a versão instalada do Adempiere, pois caso ocorra algum erro a reconfiguração da máquina virtual é bastante simples.

Em termos gerais, espera-se que este trabalho possa ser útil para os alunos dos cursos de computação e outros cursos da instituição ou mesmo fora dela. As disciplinas que tratam de projetos de software, que utilizam conceitos relacionados a gestão, projetos e ferramentas de desenvolvimento de sistemas de informação em suas ementas (Java, qualidade de software) poderão utilizá-lo para demonstrar os conceitos em um produto acabado. Isto com certeza trará benefícios para os alunos no processo de aprendizagem.

Este trabalho não esgota o assunto, na realidade possui muitas limitações. Por exemplo, faltam ainda elementos sobre o funcionamento do sistema em ambiente Linux, pois todos os testes e implementações aqui realizados foram feitos no Windows. Falta também investigar sua relação com o banco Oracle; realizar testes de implantação e uso em ambiente real. Enfim, espera-se que este trabalho possa fomentar o interesse da comunidade acadêmica dos cursos de informática a realizarem outros estudos que possam se aprofundar ainda mais nos conceitos deste produto, que com certeza gerará muitos frutos.

6. Referências Bibliográficas

- [1] ALBERTÃO, S. E. ERP Sistemas de gestão empresarial. IN: **Metodologia para avaliação, seleção e implantação: Para pequenas e médias empresas**. São Paulo: Iglu, 2001.
- [2] XAVIER, F. Projeto adempiere < http://www.adempiere.com.br/artigos/projeto_adempiere.php >, acesso em abril de 2009.
- [3] MENDES, J. V. and ESCRIVAO FILHO, E. Sistemas integrados de gestão ERP em pequenas empresas: um confronto entre o referencial teórico e a prática empresarial. In: **Gestão e Produção**. São Paulo, 2002.
- [4] 19ª pesquisa: Administração em Recursos tecnológicos. FGV, 2008 – disponibilizada em < http://www.fgv.br/fgvportal/principal/idx_materia.asp?str_chave=11269&sessao=2 >, acesso junho de 2009.
- [5] MIRANDA, I. T. P. *Et all*. Softwares livres: vantagens. IN: **Maringá Management. Revista de Ciências Empresariais**. v.2, n.1, Maringá 2005.
- [6] O que é software livre. Disponível em:< <http://br-linux.org/faq-softwarelivre> >. Acesso maio. 2009.
- [7] HABERKORN, E. **Teoria do ERP – Enterprise Resource Planning**. São Paulo : MAKRON Books, 1999.
- [8] Sobre o Postgresql. Disponível em < <http://www.postgresql.org/sobre> >. Acesso em maio 2009.
- [9] A história do Oracle: Inovação, liderança e resultados. Disponível em < <http://www.oracle.com/global/br/corporate/story.html> >. Acesso em maio 2009.
- [10] 1995 Começa a grande revolução da tecnologia Java. Disponível em < <http://br.sun.com/aboutsun/historia.jsp> >. Acesso em junho 2009.
- [11] Montenegro, E. **Uma visão geral do ERP e CRM Adempiere**: Linux Magazine, n39 Fevereiro de 2008
- [12] Raio X: confira o perfil de fornecedores de ERP no Brasil. Disponível em < <http://www.itweb.com.br/noticias/index.asp?cod=56127> >. Acesso em maio 2009.

- [13] PESSÔA, M. S. P. **Uma visão geral dos sistemas ERP Pessoa**, Depto Eng. de Produção/Escola Politécnica-USP. São Paulo, 2000.
- [14] METSKER, Steven John. **Padrões de projeto em Java**. Porto Alegre: Bookman, 2004.
- [15] GAMMA, Erich et al. **Padrões de Projeto: Soluções reutilizáveis de software Orientado a Objetos**. Porto Alegre: Bookman, 2000.
- [16] SOMMERVILLE, Ian. **Engenharia de Software**. São Paulo: Addison Wesley, 2003.
- [17] FREEMAN & FREEMAN, Eric & Elizabeth. **Padrões de Projetos: Seu cérebro em padrões de projetos**. Rio de Janeiro: ALTABOOKS, 2007.
- [18] PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Pearson Education do Brasil, 1995.
- [19] KLEPPE, A.; WARMER, J.; BAST, W. **MDA explained: the model driven architecture: practice and promise**. Boston: Pearson Education, 2003.
- [20] MELLOR, S. J. et al. **MDA distilled: principles of Model-Driven Architecture**. Boston: Addison-Wesley, 2004.
- [21] OBJECT MANAGEMENT GROUP. **MDA guide version 1.0.1**. OMG: 2003. Disponível em <http://www.omg.org/docs/omg/03-06-01.pdf>
- [22] **Developer Corner**. Disponível em <http://www.adempiere.com/index.php/Adempiere_Developer_Corner#Model> Acesso em setembro 2009.
- [23] **Table Prefix**. disponível em < http://www.adempiere.com/index.php/Table_Prefix >. Acesso em setembro 2009.