

CASE "Computer-Aided Software Engineering" ,a base para a reengenharia de negócios.

Autor: Belmiro do Nascimento João

Resumo:

Produtividade e Qualidade são questões atuais. A reengenharia exige mudanças radicais na organização, no entanto o desenvolvimento tradicional de sistemas é lento e rígido. As soluções existem. CASE vem de encontro a essas necessidades, sendo a automação da automação. São analisados os aspectos de sua categorização, seleção e implementação bem como aspectos da engenharia reversa, reusabilidade, repositórios e CASE corporativo bem como seus impactos.

Palavra-chave:

CASE, reengenharia, produtividade, desenvolvimento de sistemas, metodologias, seleção, implementação, Engenharia Reversa, Repositórios

1. INTRODUÇÃO

Este artigo é baseado na principal ferramenta para as modernas técnicas de reengenharia de negócios, onde a nível corporativo, podemos definir os modelos atuais da empresa e o reutilizá-lo em todos os novos sistemas ou processos da organização, com ganhos significativos de produtividade e abandono do antigo e podendo repensá-la à partir de um novo ponto de vista. Se você acredita que os complexos modelos organizacionais podem ser descritos com papel e lápis este artigo não o auxiliará em nada.

A época favorece as discussões a respeito das mudanças qualitativas em todas as áreas do conhecimento humano. Palavras de ordem são, reengenharia, globalização, competitividade, qualidade total, produtividade. As empresas tornam-se flexíveis e ágeis, integrando a produção na estratégia dos negócios e permitindo mudanças rápidas no produto, atendendo às rápidas demandas do mercado. No entanto não é assim com o desenvolvimento de sistemas e temos que encontrar caminhos alternativos para essa crise.

1.1. Solucionando as crises do software.

Um sentimento reinante nas empresas, é a sensação de que muito está sendo gasto em computação, e ainda assim, o resultado da área de processamento nunca é o mais adequado. Dentre os principais problemas, vistos pela administração e pelos usuários, temos: Os usuários não conseguem obter aplicações quando precisam. difícil, ou impossível, obter modificações que a administração precisa em tempo

razoável. Os sistemas entregues dificilmente atendem às reais necessidades do negócio. Os sistemas custam muito mais para desenvolver e manter do que o previsto. O problema principal, está nos métodos de trabalho. O desenvolvimento de sistemas tradicional é lento e rígido. Seu uso existe nas empresas através de padrões e procedimentos, tais como abertura de uma ordem de serviço, estudos de tempos e custos de desenvolvimento, análise, programação, implementação e testes. De um modo geral, esses procedimentos não funcionam de modo adequado. As Soluções existem mas requerem novos softwares, novos procedimentos e novas estruturas gerenciais, muitas vezes entrando em choque com a própria Cultura da organização. A hora é de um rompimento com os antigos métodos, ainda mais que a revolução gerada pelos microcomputadores permite a automação das antigas metodologias e incentiva a criação de novas técnicas para a automação do desenvolvimento de software. No entanto, a maioria das empresas e dos profissionais de processamento de dados não acompanha a velocidade da proliferação dessas novas tecnologias. As ferramentas CASE, por serem um conjunto integrado de ferramentas que podem atuar em todas as fases de desenvolvimento de software, tem um impacto profundo, exigindo novas metodologias. Straub e Wetherbe (6), através dos resultados de um painel de especialistas, nos mostram as principais tecnologias que influenciarão as empresas nos anos 90, sendo CASE considerada como uma tecnologia de impacto indireto, do ponto de vista organizacional, mas afirmando-se que é importante e estará em crescimento acelerado nos anos 90. Afirmam também que o valor de CASE consiste em "automatizar o projeto humano e codificá-lo", o que significa uma mudança fundamental no modo dos sistemas serem construídos. De outro modo os produtos CASE e as capacidades de Engenharia Reversa, terão um forte impacto nas profissões ligadas ao desenvolvimento de sistemas até meados da década de 90, sendo então, seus efeitos sentidos pela comunidade de usuários finais.

1.2. A motivação para produtividade e qualidade.

Dentre as grandes questões em desenvolvimento de software hoje, destacam-se: melhor qualidade, maior produtividade, melhor gerenciamento e custos menores e a manutenibilidade.

Simultaneamente a uma diminuição do custo dos equipamentos de processamento de dados e da sua disseminação nas áreas usuárias, tem crescido o interesse e o conhecimento por parte dos usuários da tecnologia e isso os torna mais exigentes. Requerem melhores informações gerenciais, automatizam processos, aumentando a produtividade.

Tudo isso nos leva novamente à necessidade de novos métodos de trabalho, com novas tecnologias e softwares ou ferramentas, deixando os métodos artesanais de desenvolvimento de lado; A chave da produtividade é a automação, e isto nos leva a desenvolver softwares, como qualquer outro tipo de produto manufaturado.

A manutenção tende a crescer à medida que cresce o número de programas e, imaginando-se um mercado de livre concorrência, onde os competidores travam uma luta sem fim, a tendência é de crescimento dos programas e, conseqüentemente, dos problemas de manutenção e novos "back-logs". É comum encontrarmos empresas onde o tempo total gasto com manutenção corresponde a 80% de todo o tempo gasto pelo pessoal de sistemas. No entanto, existem já

ferramentas CASE e técnicas na qual muito do custo de manutenção pode ser evitado. difícil imaginar o que seria a área de sistemas, dentro de uma década, sem a utilização de novas técnicas; sómente com as metodologias convencionais, certamente seria o caos.

A primitiva organização de Sistemas de Informação Gerencial não pode apropriar-se das vantagens de CASE e novas tecnologias de engenharia de software.

CASE automatiza partes de uma ou mais metodologias de desenvolvimento de software. A metodologia é a base, a fundação, sendo CASE a automação da metodologia.

2. METODOLOGIA E CASE.

A implementação de ferramentas CASE sem uma forte e fundamental metodologia , torna vulnerável os seu sucesso sendo improvável obter os benefícios desejados.

Uma escolha por metodologias pode revelar-se inapropriada para automação com ferramentas CASE, caso se escolha uma metodologia manual, o tédio envolvido pode desviar-se para a metodologia e conseqentemente para CASE.

Por causa do refletido processo de automação inevitavelmente mudanças no processo, tecnologia CASE e metodologia são melhor implementados juntos.

2.1. Automação das metodologias

As Metodologias, de um modo geral, estão ultrapassadas, por perpetuarem técnicas manuais do passado. Por outro lado a grande transformação pela qual passa a área de processamento de dados, nos faz pensar na automação das metodologias.

Segundo MARTIN, em Manifesto, "deve-se procurar acima de tudo, numa metodologia, a automação", precisando automatizar o trabalho do analista e do programador o mais possível, sendo que esta automação precisa ser do conhecimento dos usuários finais, e ser adaptável ao mercado de maneira flexível.

2.2. As metodologias mudam por causa de CASE

Quando as pessoas usam o termo metodologia, eles podem estar se referindo às metodologias de análise e projeto ou às metodologias que suportem todo o desenvolvimento.

As ferramentas de desenvolvimento de sistemas suportadas por computador poderão ser um fator de disciplina do trabalho, porque obrigam a seguir um roteiro pré-estabelecido, e com praticamente toda a documentação técnica gerada pela ferramenta. A linha mestra da análise orientada a processos e os métodos de projeto não foram alterados significativamente em dez anos. O diagrama de fluxo de dados, segundo Yourdon e as suas extensões para processos real-time, e os diagramas estruturais de PAGE-JONES são predominantes em análise estruturada e projeto estruturado.

Por outro lado, os métodos orientados a dados estão mudando, com variações dos diagramas de entidades relacionadas de CHEN, que dominam o mundo de projetos de dados hoje.

Os esforços para aumentar o rigor e a riqueza dessas técnicas de modelagem de dados e para se conseguir diagramas computáveis são base para algumas inovações.

Uma coisa certa: Como a tecnologia CASE incrementa em poder e capacidade, metodologias complementares continuarão a se desenvolver rapidamente.

Entretanto, programadores e analistas tradicionalmente treinados, provavelmente relutarão em absorver os novos métodos. Arquiteturas cliente/servidor e ambientes gráficos (GUI) tumultuam definitivamente esse cenário. O paradigma muda das metodologias estruturadas para as metodologias orientadas a objeto.

3. INTRODUÇÃO A CASE

CASE é automação de software. A tecnologia CASE é talvez a mais profunda transformação ocorrida na comunidade de software. CASE é a automação da automação. Ele oferece uma resposta prática aos problemas de produtividade de software dos últimos vinte e cinco anos.

A tecnologia CASE é uma combinação de ferramentas de software com metodologia. Deste modo, CASE é diferente das primeiras tecnologias de software, porque enfoca o problema da produtividade de software e não só nos problemas da implementação.

O termo foi criado no começo dos anos oitenta, quando a idéia de que ferramentas gráficas, como a dos diagramas de fluxo de dados, os diagramas de entidades e relacionamentos e gráficos estruturais, poderiam ser úteis na análise e projeto de sistemas.

Uma definição talvez mais simples e completa é a que nos é dada por McClure (5): "CASE é a automação do desenvolvimento de software". Nesta concepção CASE é uma nova abordagem para o ciclo de vida de software, que é baseada na automação. A idéia básica é que CASE proveria um conjunto integrado de Ferramentas de economia de trabalho, ligando e automatizando todas as fases do ciclo de vida de software.

Isso evidencia que a tecnologia de ferramentas automáticas continuará avançando nos próximos 10 anos. Os ferramentais do analista em meados da década de 1990, certamente terão características sofisticadas de verificação de erro, tão bem como habilidade em gerar código e mesmo sugerem (usando técnicas de inteligência artificial) possibilidades de código reutilizável a partir de Bibliotecas de Software. Muitas ferramentas CASE incluem poderosos editores gráficos que implementam uma interface com o usuário transformando o desenvolvimento de software num processo visual.

3.1. CASE e o Ciclo de Vida

CASE altera o ciclo de vida do sistema. Com CASE, o projetista do sistema pode focalizar mais o trabalho da análise e projeto e menos a codificação e teste.

Programas codificados manualmente e alterados por geradores de código que automaticamente produzem código a partir das especificações do projeto.

Especificações do projeto tomam a forma de diagramas estruturados, como DFD e DER, e modelos de protótipos. Como as especificações são armazenadas no repositório CASE, elas são automaticamente verificadas para a correção e referência cruzada com outro sistema de informação. Toda a informação do sistema é armazenada em repositório CASE onde ele pode ser automaticamente gerenciado e distribuído.

Finalmente, CASE reduz o sério, e custoso, problema de manutenção de sistemas,

pois as ferramentas CASE podem documentar, analisar e reestruturar os programas existentes agilizando o tempo de entrega para o mercado.

3.2. Categorização

Há poucos anos atrás, dificilmente algum analista de sistemas teria ouvido o termo CASE. O mercado de ferramentas CASE tem um dos mais altos graus de crescimento das empresas que atuam no setor de desenvolvimento de softwares. Como o mercado nasce, a definição de CASE continua a mudar e não há um padrão definido para a sua categorização, no entanto os termos abaixo são os que melhor o identificam.

- a) Front End ou Upper CASE; são aquelas que apóiam as etapas iniciais de criação dos sistemas, são elas as fases de planejamento, análise e projeto do programa ou aplicação.
 - b) Back End ou Lower CASE; são aquelas que dão apoio a parte física, isto é, a codificação testes e manutenção da aplicação, normalmente encontram se produtos de ambos os lados, mas o difícil é se obter uma compatibilidade entre ambas, isto é, conseguir com o auxilio de uma única ferramenta.
 - c) I-CASE ou Integrated CASE; classifica os produtos que cobrem todo o ciclo de vida do software, desde os requisitos do sistema até o controle final da qualidade. Outra classificação nos é dada segundo a fase do ciclo de vida que apoia: Planning Workstation, Analysis Workstation, Design Workstation, Ferramentas de programação.
-

3.3. Critério de seleção.

Dentre os critérios para a seleção de ferramentas CASE, destacamos, segundo Gibson (1):

- a) a ferramenta é um DBMS ou um sistema de dicionário de software.
- b) Quais as direções para o futuro e funcionalidade da ferramenta.
- c) O fabricante da ferramenta tem uma filosofia de arquitetura aberta.
- d) A ferramenta produz software utilitário e procedimentos de leitura, bibliotecas de fontes e cria especificações de componentes CASE para sistemas existentes.
- e) A ferramenta tem uma interface efetiva com outras ferramentas.
- f) A ferramenta dispõe de metodologias gráficas, capazes de explodir os diagramas do projeto e especificações do dicionário, para um sistema de dimensões moderadas.
- g) O modelo de planejamento no componente de planejamento do CASE é compreensivamente abrangente ao nível de corporação e unidades funcionais, plano estratégico e de sistema.
- h) É provido de capacidade de Prototipação.
- i) É capaz de gerar automaticamente um escopo das especificações do projeto físico a partir das especificações do projeto.
- j) A ferramenta de projeto é provida de suporte a análise e documentação do projeto.
- k) A ferramenta tem capacidade de gerar automaticamente relatórios das especificações do projeto.
- l) O componente de desenvolvimento de baixo-nível provê métodos para conveniencia e compreensiva customização do sistema gerado.

- m) A ferramenta permite a distribuição de responsabilidades pelo projeto e desenvolvimento.
 - n) As ferramentas de projeto e desenvolvimento tem a capacidade de enviar porções do projeto e especificações do dicionário de desenvolvimento.
 - o) Pode a ferramenta de interface do projeto e especificações de desenvolvimento para um DBMS funcional, ser usada para manter dados de uma companhia.
 - p) Tem capacidade de processamento de texto.
 - q) A ferramenta destaca o gerenciamento do projeto.
 - r) É possível modificar o projeto CASE e ferramentas de desenvolvimento relativas ao ambiente interno da empresa ou metodologia existente.
 - s) Pode a ferramenta gerar automaticamente projeto, operação e documentação para o usuário final.
 - t) A ferramenta tem facilidades para manter o projeto tão bem quanto sistemas.
 - u) Pode a ferramenta gerar programas para uma ampla faixa de sistemas.
-

3.4. Critérios para implementação

- Dentre os critérios para a implementação de ferramentas CASE , segundo Gibson (2):
- a) Estabelecimento e aplicação de critérios para avaliação das Ferramentas. b) Inclusão de um Treinamento extensivo.
 - c) Seleção de um grupo experiente para Treinamento e acompanhamento durante a implementação.
 - d) Formulação de um conjunto de Padrões e procedimentos.
 - e) Grupo de Consultores customizando o Treinamento e Padrões de documentação e procedimentos.
 - f) Seleção de um grupo experiente (Líderes de Projeto), como primeiro grupo a receber treinamento.
 - g) Casa haja treinamento interno deve haver a inclusão de Trainees.
 - h) Antes do início de um projeto, ter um grupo treinado.
 - i) O projeto piloto deve ser relativamente simples, de 3 a 6 meses, para não haver desgaste.
 - j) O grupo diretamente ligado ao piloto deve ser treinado por consultores.
 - k) Após o término do primeiro projeto, dever ser reavaliados e alterados os padrões e procedimentos da empresa de acordo com a experiência alcançada.
 - l) Deverá ser feita uma reavaliação do treinamento após o primeiro projeto.
-

3.5. O papel da Gerência

Quando se fala em implementação de CASE, o apoio da alta gerência é fundamental e, sem esse apoio, pode haver uma grande probabilidade de fracasso a curto prazo. O convencimento parte do princípio de que desenvolver sistemas de forma tradicional não é mais viável.

Vários fatores devem ser considerados:

- a) Um extenso "back-log", isto é , aplicações pendentes de desenvolvimento.
- b) Uma comunidade de usuários mais exigente e que passa a questionar os custos de computação, os usuários se tornaram extremamente críticos principalmente com o tempo de desenvolvimento. Aqui uma questão é fundamental: o desenvolvimento de aplicativos não acompanhou o desenvolvimento de Hardware, e o CASE seria uma resposta nessa direção.

c) O mercado é cada vez mais competitivo, o qual gera expectativas de resposta cada vez menor e desenvolvimentos simultaneos em novas tecnologias para se ganhar vantagem competitiva (Time-to-Market).

A utilização de recursos financeiros representa um dos aspectos mais significativos na utilização de ferramentas CASE, sabendo-se de antemão que será necessário gastar recursos em metodologias, o que é apenas um compromisso. Treinamento adequado ao pessoal que desenvolve aplicações, mudanças organizacionais, que vem conjuntamente com as metodologias, dado que esta requer um rigor e disciplina para que os sistemas de informação trabalhem de forma conjunta. CASE realmente necessita ser adotado em um nível estratégico porque altera o ciclo de vida de desenvolvimento de software e muitas das práticas fundamentais de desenvolvimento de software.

Watts Humphrey, em seu recente livro "Managing the Software Process", adverte: "Insanidade é o processo de repetir a mesma coisa, sempre e sempre, e aguardar resultados diferentes". Esta é uma referência a que as práticas de desenvolvimento não mudaram realmente nos últimos 20 anos.

O pessoal de processamento de dados e os gerentes necessitam aceitar suas responsabilidades para o processo de desenvolvimento de software, seja que o processo pode e deve ser melhorado através de uma abordagem tradicional, sem o uso de ferramentas CASE ou outra tecnologia.

4. AVANÇOS FUTUROS EM CASE

Passaremos a descrever algumas das principais características das ferramentas CASE, que estão em evolução neste momento, como os repositórios e o conceito de reusabilidade de software.

4.1. CASE Corporativo.

Ainda sobre o estudo de STRAUB e WETHERBE (6), o grande impacto organizacional de CASE será sentido quando este tiver o status de CASE Corporativo, fazendo com que os altos investimentos feitos tenham retorno, dentre as direções temos: visão estratégica da empresa, fixação de uma metodologia, utilização de métricas (p.ex.: APF), utilização do repositório corporativo, reengenharia e engenharia de reutilização.

4.2. Engenharia reversa

Engenharia reversa de programas existentes, tem um amplo espaço futuro para desenvolvimento. A inabilidade para automatizar a manutenção dos milhões de linhas de código existentes é uma séria falha em tecnologia na maioria das empresas. Estimam que são gastos pelo menos 60 % das suas atividades de software na manutenção ou modificação dos sistemas existentes. Ferramentas de engenharia reversa de código procedural poder vir bem mais tarde. No inicio dos anos 90, ferramentas para auxiliar na compreensão do código existente através da conversão para a forma gráfica e editores inteligentes começam a aparecer, algumas ferramentas CASE poderão assistir o programador na edição de novo código a sistemas existentes.

4.3. Software Reusável

Uma questão que domina o desenvolvimento de software na atualidade é o seu alto custo em relação ao do Hardware. Entre os fatores que contribuem para uma dramática redução no custo de software está a reusabilidade dos programas. A reusabilidade é um atributo ou qualidade de programas que podem ser usados em múltiplas aplicações. A reusabilidade pode ser obtida tanto com a geração automática de programas quanto manualmente, através de programação parametrizada. A reusabilidade é uma resposta importante aos problemas de produtividade de software. Possibilidades para reusabilidade de software incluem: código reusável, projetos, especificações formais, pacotes de software, protótipos e geradores de aplicação e sistemas de transformação. Ferramentas CASE jogam com uma importante regra, fazendo da reusabilidade uma estratégia prática de construção de sistemas. O repositório CASE forma um coração das necessidades das ferramentas CASE para reusabilidade.

4.4. Repositórios

O repositório CASE é o mecanismo para armazenamento e organização de todas as informações concernentes ao sistema de software, incluindo planejamento, análise, projeto e construção. Alternativamente o repositório CASE é referenciado como um dicionário de dados, um banco de dados, um banco de dados orientado a objeto, uma base de conhecimento ou enciclopédia, dependendo de como o fornecedor de software o chame. É importante notar que o conceito de repositório transcende o conceito de um simples dicionário de dados, desde que o repositório armazena mais tipos de informações do sistema, os relacionamentos entre os vários componentes de informação e as regras para uso dos componentes ou seu processamento.

5. CONCLUSÃO

As conclusões a seguir nos mostram um quadro referente à introdução de CASE (3):

- a) Falhas na formação que tem a maioria dos profissionais de desenvolvimento de sistemas, principalmente no que se refere a metodologias estruturadas.
- b) Deve ser incentivada a criação de pequenos planos pilotos para implantação de CASE, fortemente apoiados em metodologias. E aqui surgem os primeiros problemas referentes a falta de vivência dos profissionais com os "novos métodos", onde necessitam de treinamento sistemático para a sua assimilação e posterior utilização.
- c) O apoio da alta gerência passa a ter um papel fundamental na implementação de CASE, e neste caso, incluindo a "venda" da idéia para a alta direção da empresa, usuários e toda a gerência responsável por sistemas dentro da organização.
- d) As ferramentas CASE, estão apenas começando a se sedimentar no mercado, e normalmente a primeira ferramenta CASE implementada por uma organização de modo algum é a escolha definitiva e sim resultado de um plano de implementação conjunta de uma cultura voltada para metodologias estruturadas e da utilização de ferramentas.
- e) Empresas que não tem tradição em metodologias ou não tenham uma metodologia própria não se aventuram em metodologias novas ou pouco difundidas no mercado, deixando para aquelas que tem experiência a adoção de metodologias novas.

- f) Os aspectos de prototipação são encarados como importantes, assim com a adoção de bancos de dados relacionais em todos os novos projetos. h) Não há controles efetivos de custos de desenvolvimento de sistemas, e a sua utilização, quando existe, se resume a gerenciadores de projeto, não há a utilização de métricas para controle efetivo de projeto, custos, prazos, alocações de recursos, etc.
- i) Existe pouco esforço em termos de planejamento para a implementação de metodologias estruturadas e tampouco de ferramentas CASE na maioria das empresas, resultando as implementações existentes, próprias de poucas organizações que as estão desenvolvendo devido a necessidade crescente de redução de custos, ganhos de produtividade e credibilidade da área.
- j) A grande velocidade com que os recursos e técnicas computacionais evoluem. A falta de uma cultura de metodologias estruturadas, em todos os níveis. A falta de instrumentos adequados para o planejamento de informática.

Entre os benefícios podemos citar (4):

- a) Tornar a prática de técnicas estruturadas, existentes desde a década de 60 e 70 e simultaneamente estimular novas metodologias.
- b) Aplicar a Engenharia de Informações e a de software.
- c) Melhorar a qualidade de software através da verificação automatizada, eliminando a possibilidade de erros.
- d) Tornar prática a prototipação, acelerando o desenvolvimento de sistemas junto ao usuário.
- e) Simplificar a manutenção de programas
- f) Agilizar o processo de desenvolvimento.
- g) Liberar o analista de desenvolvimento para a parte criativa do desenvolvimento de software.
- h) Encorajar a evolução e desenvolvimento incremental.
- i) Deixar disponível o reuso de componentes de sistemas.
- j) Importância do aspecto da documentação de sistemas e diminuição de "Back-Log".
- k) Ainda não há confiança plena na tecnologia CASE atual e o caminho para a sua implementação dever ser acompanhado passo a passo juntamente com as metodologias, mas acredita-se nas suas possibilidades, principalmente após a segunda metade desta década.
- l) necessária a criação de uma base educacional para metodologias estruturadas incluindo a mudança de atitude do profissional de desenvolvimento de sistemas.
- m) Há um forte impacto na tecnologia com a implantação de CASE.
- n) Há resistência a mudanças na implantação de CASE: de um lado o pessoal técnico, que não está acostumado com especificações de sistemas, e da gerência, que não está acostumada a controlar projetos e utilizar metodologias estruturadas com conseqüente perda de prestígio e poder dentro da organização.
- o) Constitui a base da moderna reengenharia de processos, desde que trabalhado a nível corporativo.
- p) Permitir a construção de sistemas realmente inovadores atendendo aos processos de negócio básicos da organização e com máxima reação para o mercado.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- (1) **GIBSON**, Michael L., A Guide to Selecting CASE Tools, in Datamation, Julho 1988, p. 65 a 66.
 - (2) **GIBSON**, Michael L., Implementing The Promise, in Datamation, Fevereiro 1989, p. 65 a 67
 - (3) **JOÃO**, Belmiro N., A Introdução da Cultura CASE "Computer- Aided Software Engineering" no Brasil, dissertação de Mestrado, São Paulo, PUC/SP, 1992, 335p.
 - (4) **JOÃO**, Belmiro N., Metodologias de Desenvolvimento de Sistemas, São Paulo, Ed. Érica, 1993, 155p.
 - (5) **McCLURE**, Carma, CASE is Software Automation, New Jersey, Prentice-Hall, 1989, 290p.
 - (6) **STRAUB**, Detmar W. & **WETHERBE**, James, Information Technologies for the 1990s: An Organizational Impact Perspective, Communications of the ACM, Novembro 1989, p.1328 a 1339.
-

Publicado originalmente em: REVISTA TIBIRIÇÁ N. 25 Agosto/94, págs. 85 a 96, podendo ser reimpresso, observado os créditos e desde que citada a fonte