

# Engenharia de Software

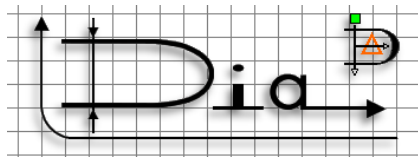
**Professor:**

Zady Castaneda Salazar



# Aula 5

## Ferramentas da Engenharia de Software



◆ **MÉTODOS**: prescrevem os detalhes sobre como fazer para construir o software.



◆ **PROCEDIMENTOS**: constituem o elo de ligação entre os métodos e ferramentas.



◆ **FERRAMENTAS**: dão suporte automatizado aos métodos.

**MÉTODOS:** prescrevem os detalhes sobre como fazer para construir o software.

- Engenharia de requisitos
- Análise de requisitos
- Projeto de software
- Codificação
- Teste
- Manutenção

...

**PROCEDIMENTOS:** constituem o elo de ligação entre os métodos e ferramentas.

- Sequência em que os métodos são aplicados.
- Produtos que se exige que sejam entregues.
- Controles que ajudam assegurar a qualidade e coordenar as alterações.

**FERRAMENTAS:** dão suporte automatizado aos métodos.

- Existem atualmente ferramentas para apoiar cada um dos métodos.
- CASE (Computer Aided Software Engineering) é uma classificação que abrange **todas ferramentas baseadas em computadores que auxiliam atividades de engenharia de software**, desde análise de requisitos e modelagem até programação e testes, tais como: desenho de diagramas, checagem da consistência dos diagramas e manutenção do programa.

## ◆ Ferramentas CASE

- São ferramentas que dão apoio à engenharia de software
- Têm o objetivo fornecer um suporte automatizado para as atividades de processo de software.
- Automatizam e padronizam as modelagens desenvolvidas
- Verificam consistência, integridade e metodologia de diagramas e dados

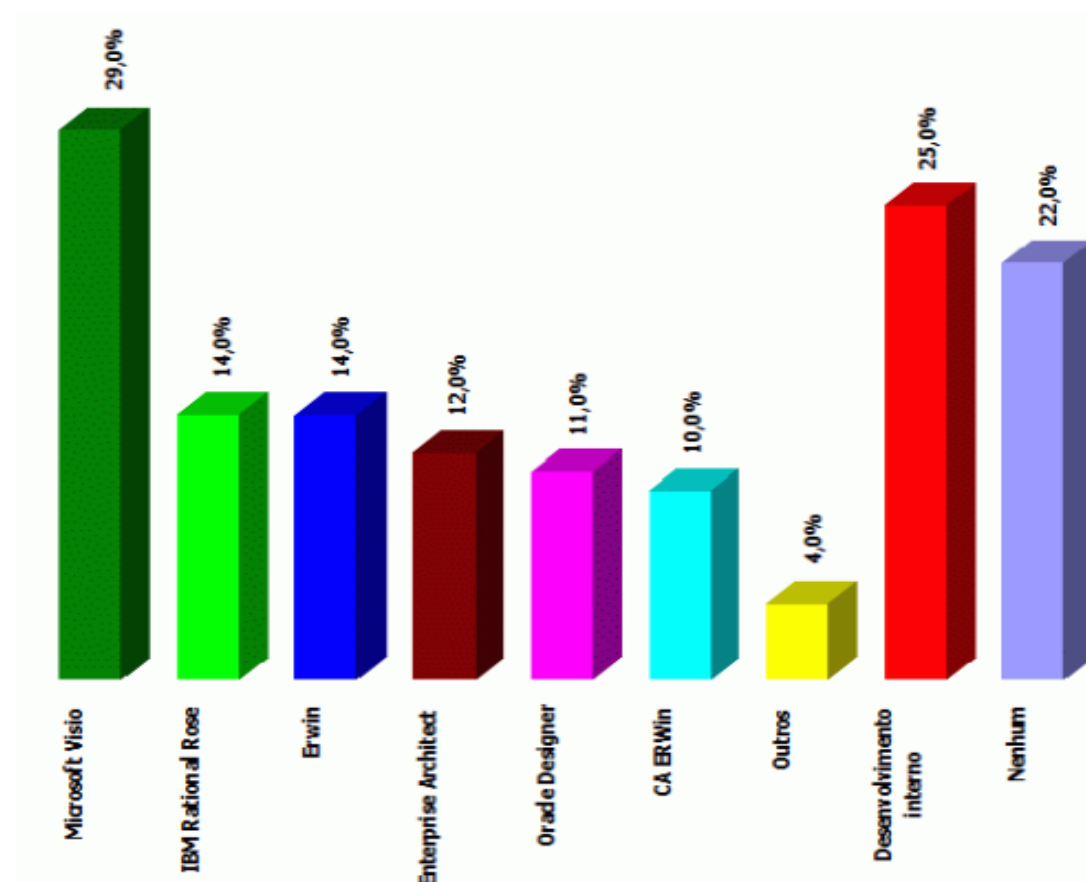
## ◆ Ferramentas CASE

- Operam em dois níveis:
- **Alto nível:** ferramentas que suportam as atividades iniciais de requisitos e projetos. Ex:
  - Na fase de Engenharia de Requisitos temos a ferramenta **RE-Tools**. Ela constitui-se de um kit de ferramentas, *open source*, para modelagem de Requisitos em múltiplas notações.
- **Baixo nível:** ferramentas que suportam as atividades de programação, depuração e testes.



## ◆ Ferramentas CASE

- A presença de ferramentas CASE é vital para o bom funcionamento de uma empresa desenvolvedora de software.



- ◆ **Exemplo de atividades que podem ser automatizadas com o uso de CASE.**
- O desenvolvimento dos modelos gráficos de sistemas como parte da especificação de requisitos ou do projeto de software.
- A **compreensão de um projeto** com informações sobre as entidades e as relações em um projeto.
- A **geração de interfaces** com o usuário com base em uma descrição de interface gráfica criada interativamente pelo usuário.

- ◆ **Exemplo de atividades que podem ser automatizadas com o uso de CASE.**
- O *debugging* do programa por meio do fornecimento de informações sobre um programa em execução.
- A **tradução automática de programas** a partir de uma versão antiga de uma linguagem de programação.

- A tecnologia de Ferramentas Case trouxe melhorias significativas para o processo de Software (qualidade e produtividade) como aprimoramentos de até 40%;
- Entretanto os aprimoramentos são limitados por dois fatores:
  - A engenharia de software requer pensamento criativo - **isso não é facilmente automatizado**;
  - Engenharia de software é uma atividade de equipe e, para grandes projetos, - **muito tempo é gasto em interações de equipe.**

# Vantagens

---

- **Maior qualidade dos produtos finais:** diminuem a probabilidade de erros, uma vez que podem ajudar no controle de consistência dos dados;
- **Produtividade:** ao ajudar e até mesmo ao realizar algumas tarefas automaticamente;
- **Eliminação de trabalho monótono:** podem realizar algumas tarefas cansativas para os desenvolvedores, tais como procurar informações e desenhar símbolos de um diagrama;
- **Mais tempo para a tomada de decisão:** – em consequência de as ferramentas realizarem certas atividades pelas pessoas, estas ficam liberadas para outras tarefas, que exigem tomadas de decisão e criatividade, ao invés de tarefas repetitivas;

# Vantagens

---

- **Flexibilidade para mudanças:** permitem que sejam mudados dados e diagramas de maneira mais rápida e fácil;
- **Menos programação:** eliminam muito do trabalho de programação, deixando mais tempo para que a equipe técnica se preocupe com a Análise do Sistema, que é onde se define como solucionar o problema do usuário;
- **Melhor documentação:** por armazenarem dados e diagramas, as ferramentas também contribuem para uma melhor documentação do sistema, agilizando relatórios, busca de informações e alterações;
- **Uniformização:** do processo de desenvolvimento, das atividades realizadas, e dos artefatos produzidos.

- Incompatibilidade de ferramentas;
- Elevado custo da ferramenta e do treino para a sua utilização;
- Limitações na flexibilidade da documentação;
- Impossibilidade de utilizar vários métodos na mesma ferramenta;
- Modificação cultural;
- Ferramentas de gerência não integradas às atividades;
- Armazenamento de dados inadequados.

- ❑ **Lower-CASE** – ferramentas de ambientes mais simples, provêm suporte à codificação, teste, depuração e manutenção do código do software;
- ❑ **Upper-CASE** – ferramentas de ambientes mais complexos, automatizam diversas tarefas de análise e de projetos de sistemas e são capazes de gerar código automaticamente, a partir das especificações dadas;
- ❑ **I-CASE** (Integrated-CASE) – ambiente caracterizado por um grupo de Ferramentas CASE integradas, isto é, que se relacionam entre si (entradas e saídas) e que permitem controlar a consistência dos dados quando uma metodologia é seguida. A ideia de metodologia embutida no ambiente garante uma maior integração entre as ferramentas, permitindo que as próprias ferramentas façam a verificação de consistência dos dados gerados por elas.



- As ferramentas CASE podem ser:

## Horizontais:

- oferecem serviços utilizados durante todo o processo de software;
- Ex: suporte à documentação; gerenciamento de versões; configurações;

## Verticais:

- utilizadas em fases específicas do processo de software;
- Ex: análise de requisitos e teste de software;

# Classificação de Ferramentas CASE

A classificação nos ajuda a compreender os diferentes tipos de ferramentas CASE e o seu **apoio às atividades do processo;**

## Perspectiva funcional:

- As ferramentas são classificadas de acordo com sua função específica.

## Perspectiva do processo

- As ferramentas são classificadas de acordo com as atividades de processo que são suportadas.

## Perspectiva de Integração:

- As ferramentas são classificadas de acordo com sua organização em unidades integradas.

# Classificação Funcional

Ferramentas de	Exemplos
Planejamento	Ferramentas PERT, ferramentas de estimativa, planilha de cálculo
Edição	Editores de textos, editores de programas, processadores de texto
Gerenciamento de mudança	Ferramentas de controle de requisitos, sistemas de controle de mudanças
Gerenciamento de configuração	Sistemas de gerenciamento de versão, ferramentas de construção de sistemas
Prototipação	Linguagens de nível muito alto, geradores de interfaces com o usuário
Apoio a métodos	Editores de projeto, dicionário de dados, geradores de códigos
Processamento de linguagem	Compiladores, interpretadores
Análise de programa	Geradores de referência cruzada, analisadores estáticos, analisadores dinâmicos
Testes	Geradores de dados de testes, comparadores de arquivos
Depuração	Sistemas interativos de depuração
Documentação	Programas de layout de página, editores de imagem
Reengenharia	Sistemas de referência cruzada, sistemas de reestruturação de programas

# Classificação baseada em Atividades

Ferramentas de reengenharia			•	
Ferramentas de testes			•	•
Ferramentas de depuração			•	•
Ferramentas de análise de programa			•	•
Ferramentas de processamento de linguagem		•	•	
Ferramentas de apoio a métodos	•	•		
Ferramentas de prototipação	•			•
Ferramentas de gerenciamento de configuração		•	•	
Ferramentas de gerenciamento de mudança	•	•	•	•
Ferramentas de documentação	•	•	•	•
Ferramentas de edição	•	•	•	•
Ferramentas de planejamento	•	•	•	•
	Especificação	Projeto	Implementação	Verificação e Validação

- **Ferramentas**

- Suporte a tarefas de processo individuais, como verificação de consistência de projeto, edição de texto, etc.

- **Workbenches**

- Suporte a uma fase de processo, como especificação ou design, normalmente incluem um número de ferramentas integradas.

- **Ambientes**

- Apoiar todos ou uma parte substancial de um processo de software inteiro. Normalmente incluem várias workbenches integrados.

# Ferramentas, Workbenches, Ambientes



# Ferramentas de Engenharia de Processo de Negócio

- **Processo de Negócio:** atividades previamente estabelecidas cujo objetivo é determinar como o trabalho será realizado em uma organização;
- Objetivo principal é **representar objetos de dados de negócio**, seus **relacionamentos** e como esses objetos de dados **fluem** entre as diferentes áreas de negócio, dentro de uma companhia;
- **Organização:** **entidades** (fornecedores, clientes, funcionários, produtos/serviços) e **funções básicas** (produção, marketing e vendas, contabilidade e finanças, recursos humanos)
- Modelam os requisitos de informação estratégica de uma organização;
- Foco na informação do negócio e não nos requisitos de uma aplicação específica;

# Ferramentas de Gerenciamento de Projetos

- Gerência de Projetos de Desenvolvimento de Software;
- Armazenam/Gerenciam informações sobre **orçamento** (custos e investimentos), **cronogramas**, **atividades**, **recursos**, documentos e decisões;
- Deve oferecer recursos para que sejam feitos gráficos (ou gerar automaticamente) tais como:
  - ✓ atividades X pessoal;
  - ✓ atividades X recursos;
  - ✓ pessoal X recursos ou deve ela;
- Quando houver algum problema (por exemplo, atraso no cronograma ou gastos excedentes), a ferramenta deve avisar o Gerente de Projeto e recalcular as consequências (o novo prazo ou o aumento nos custos).
- Armazenamento dos documentos do Projeto (relatórios de andamento/atividades, planos, atas de reuniões, ...);
- **Controle de Versões:** informações sobre quais módulos foram utilizados, onde eles estão guardados, quem os fez e como eles foram integrados;



- **Ferramentas de produção de documentos** e de editoração eletrônica (usadas em praticamente todas etapas);
- Em média 20% ou 30% de todo o esforço de desenvolvimento de software é gasto em documentação;
- Ferramentas de documentação podem oferecer uma importante oportunidade para melhorar a produtividade;

- **Ferramentas de métricas** que fazem a auditoria do código-fonte para assegurar o atendimento das normas de linguagem;
- Extraem métricas técnicas num esforço de projetar a qualidade do software que está sendo construído;
- Encontrar erros no software;

# Ferramentas de Gestão de Configuração de Software

---

- A **gestão de configuração** de software fica no miolo de qualquer ambiente CASE.
- As ferramentas podem assistir as cinco principais tarefas de SCM (Software Configuration Management) (identificação, controle de versão, controle de modificação, auditoria e listagem de categorias);

- Definir o layout de tela rapidamente para aplicações interativas;
- As **ferramentas CASE de prototipação** mais sofisticadas permitem a criação de projeto de dados acoplado com layout, tanto de tela quanto de relatório.
- Algumas ferramentas podem gerar estruturas de código-fonte construir aplicações (de tempo real).

- Manutenção de Sistemas;
- Entender os “segredos” do sistema através da análise do programa, buscando criar uma representação do mesmo em um nível de abstração mais elevado (ex: criar diagramas UML a partir de um código fonte);
- Categorizadas como ferramentas de análise estática ou de análise dinâmica;
- Análise Estática: extrai informações do código fonte;
- Análise Dinâmica: extrai informações de um programa em execução;
- Essas ferramentas são limitadas a linguagens de programação específicas;

- Criar modelos do sistema a ser construído;
- Representação dos dados, funções e comportamento do sistema;
- Verificações de consistência dos modelos;
- Eliminar erros, antes que se propaguem para o projeto, ou para a implementação propriamente dita;

- Edição de código fonte;
- Compiladores;
- Debug;
- Editores de interface gráfica integrado;

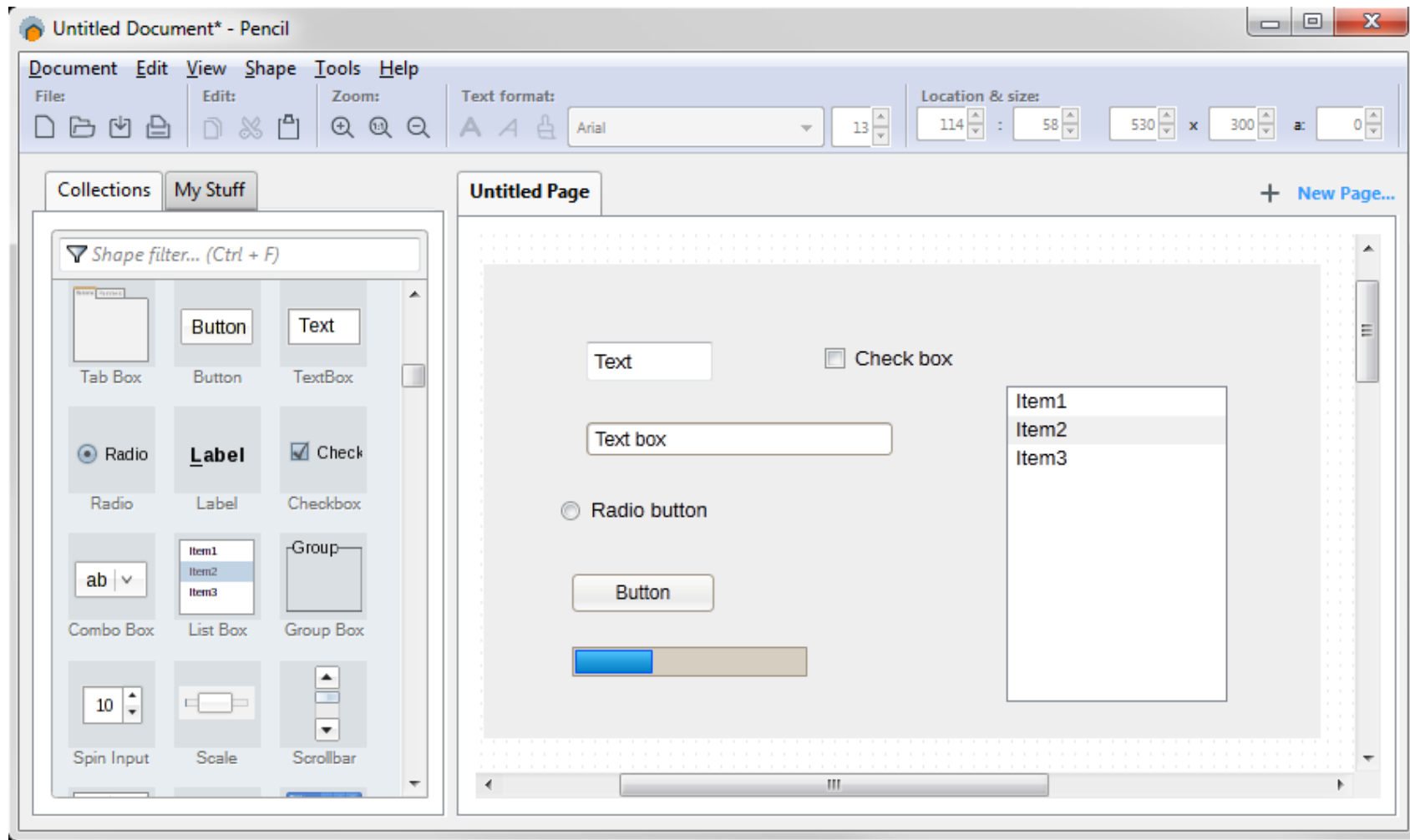
- Auxiliam na execução de testes de software;
- Podem possuir funcionalidades como:
- Gerar dados a serem usados durante o teste;
- Análise do código-fonte sem executar casos de teste;
- Análise do código-fonte durante a execução;
- Simulação de funções de hardware e outros dispositivos externos;



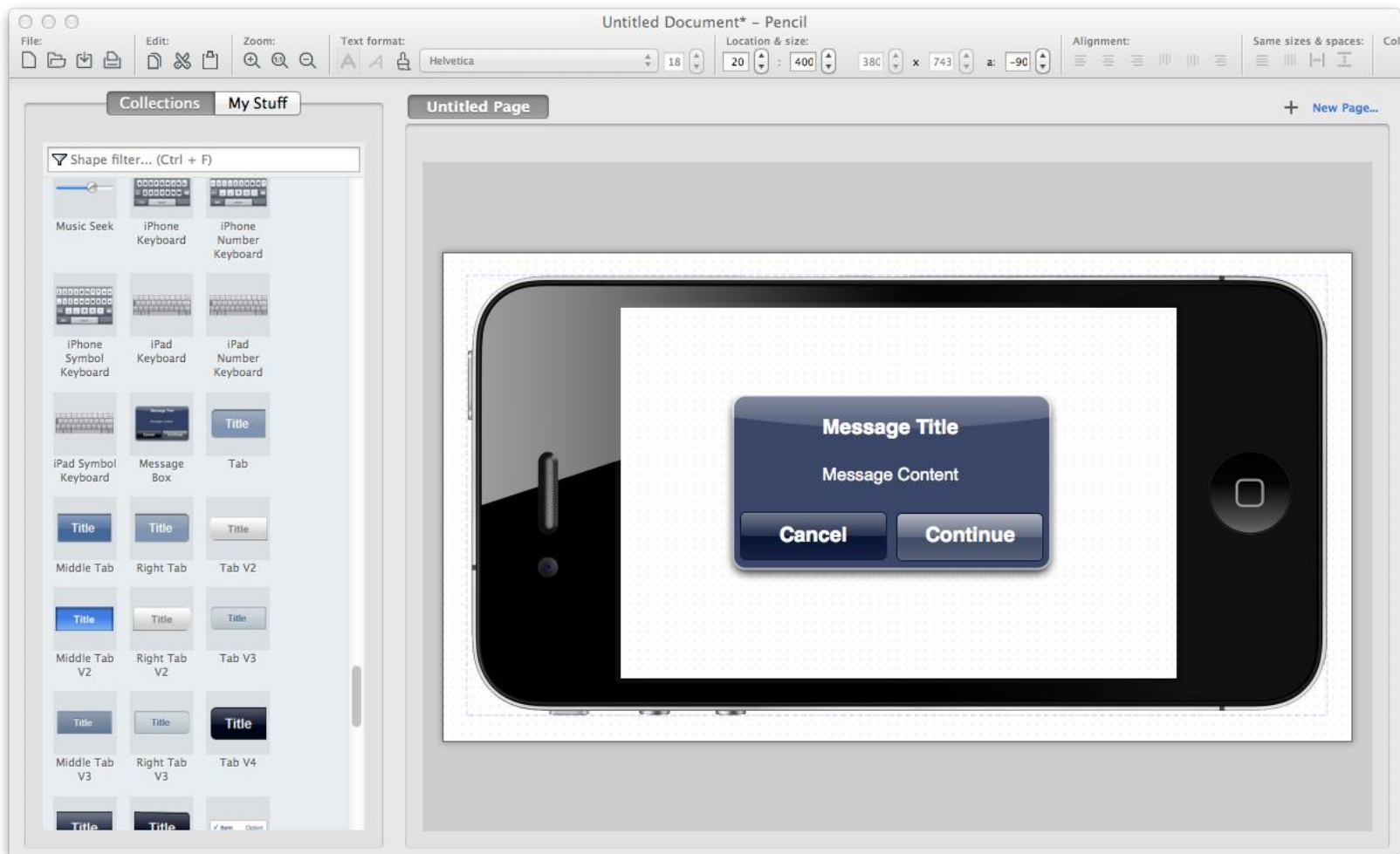
- **Exemplos de ferramentas CASE :**
  - ◆ **Pencil Project**
  - ◆ **BPWin**
  - ◆ **Rational Rose**
  - ◆ **ERWin**
  - ◆ **ArgoUML**
  - ◆ **Creately**
  - ◆ **Astah**

- É uma ferramenta gratuita, é extremamente útil e muito fácil de usar.
- Pencil Project é uma ferramenta utilizada no desenvolvimento de protótipo de interfaces gráfica;
- 
- Ele oferece uma “coleção de moldes; permitindo a rápida produção do rascunho de um desenho, diagrama ou da estrutura uma página Web;
- Possui uma interface bastante amigável e intuitiva, permitindo fácil acesso às ferramentas como: caixas de diálogo, abas, caixas de seleção, painéis e etc.

## ◆ Pencil Project



## ◆ Pencil Project



- Exemplos :

- ◆ **BPWin**

- ◆ BPwin é uma poderosa ferramenta de modelagem utilizada para analisar, documentar e melhorar os processos de negócios complexos.
- ◆ Um modelo BPwin permite documentar claramente fatores importantes, tais como quais atividades são necessárias, como elas são executadas e quais os recursos necessários.

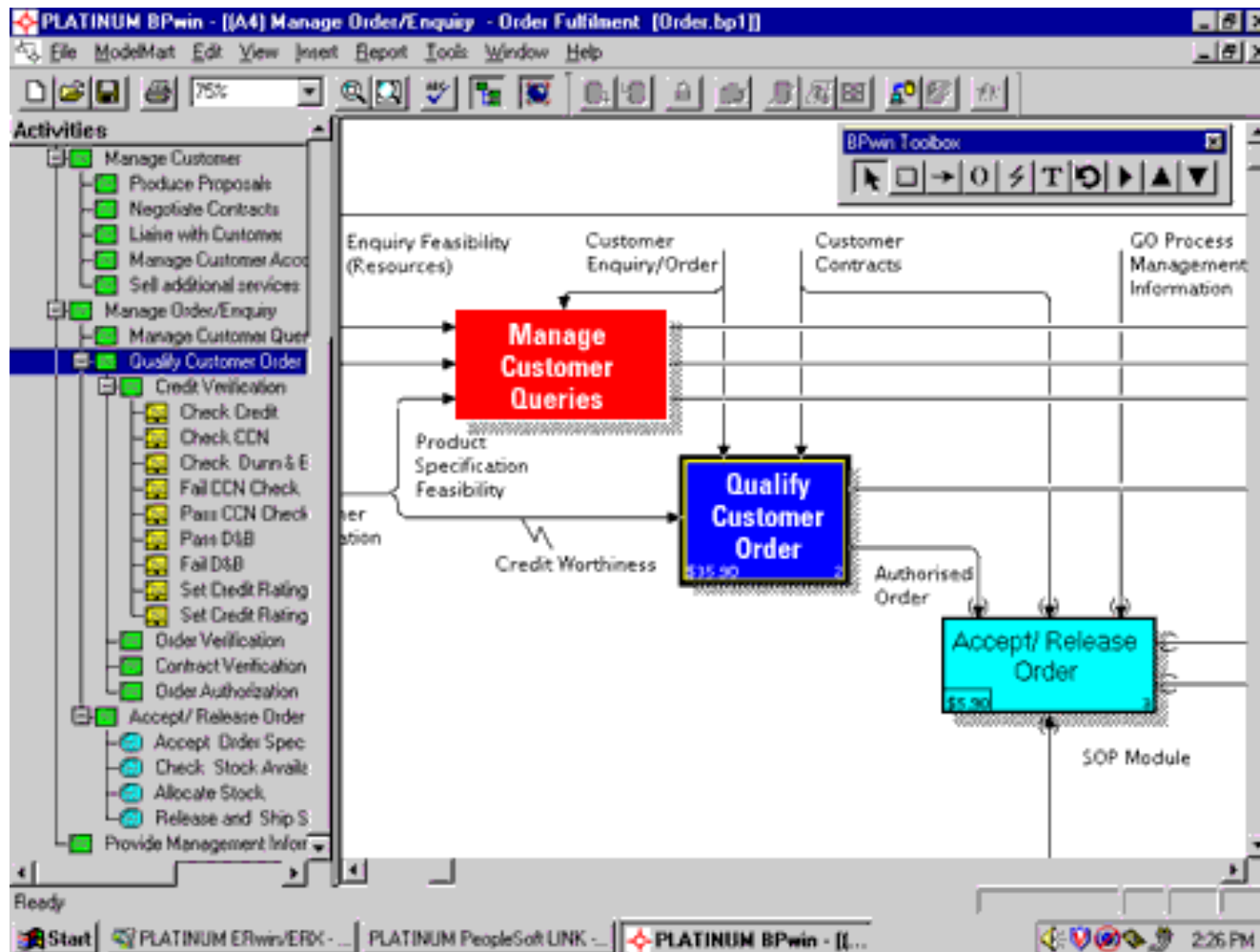
- Exemplos :

## ◆ BPWin

- Modelagem de processos que permite aliar estratégia e tecnologia da informação
- Atividades e objetos comuns podem ser criados com um simples clicar do mouse
- Flexibilidade de apresentação e documentação com variedades de fontes, cores e formas

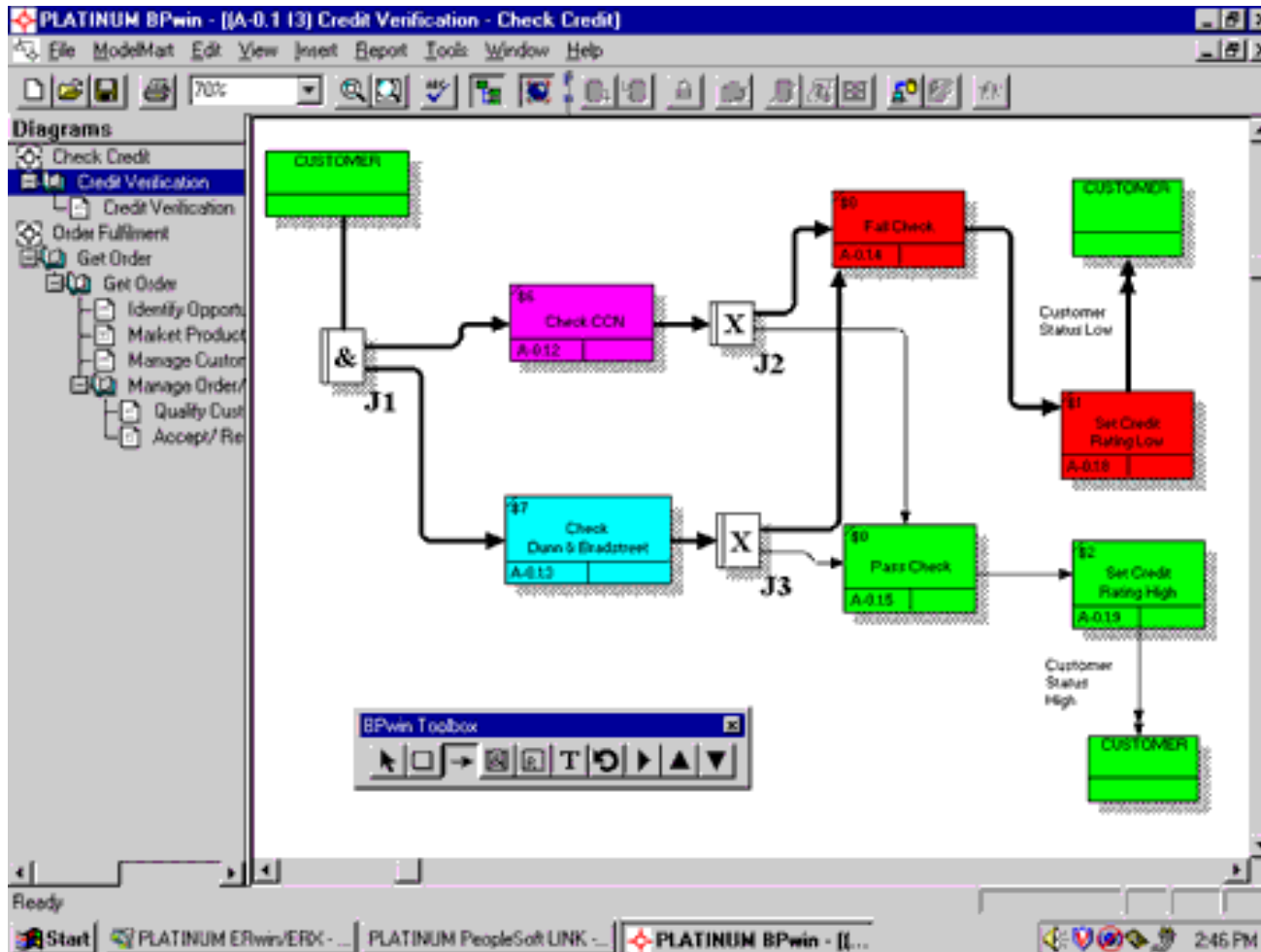
# Engenharia de software

## ◆ BPWin



Permite fazer um diagrama que mostra claramente actividades empresariais, os seus resultados e os recursos necessários para apoiá-los

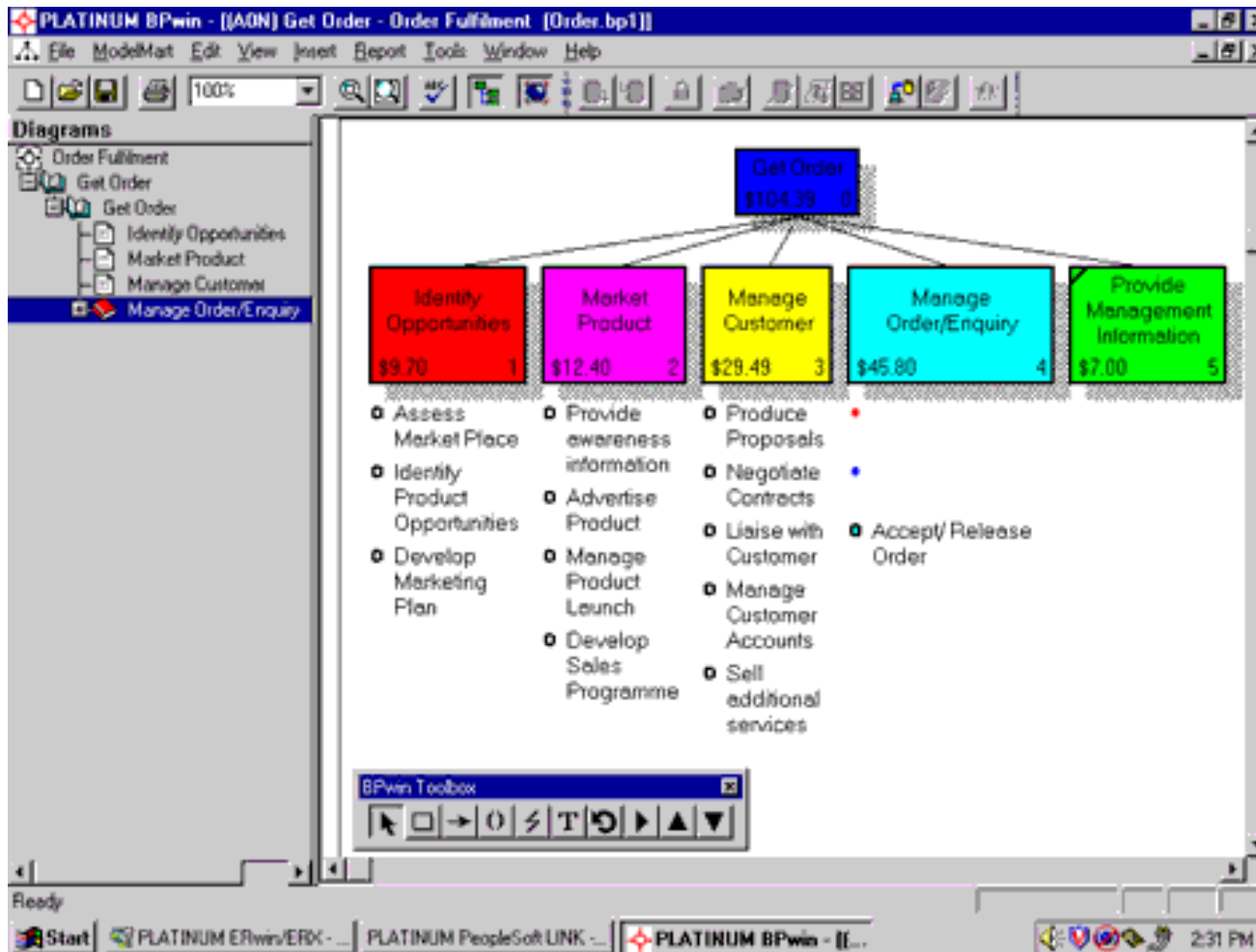
## ◆ BPWin



Suporta a função ,  
fluxo de dados e fluxo  
de trabalho de  
modelagem em uma  
única ferramenta ,  
integrando três  
perspectivas.



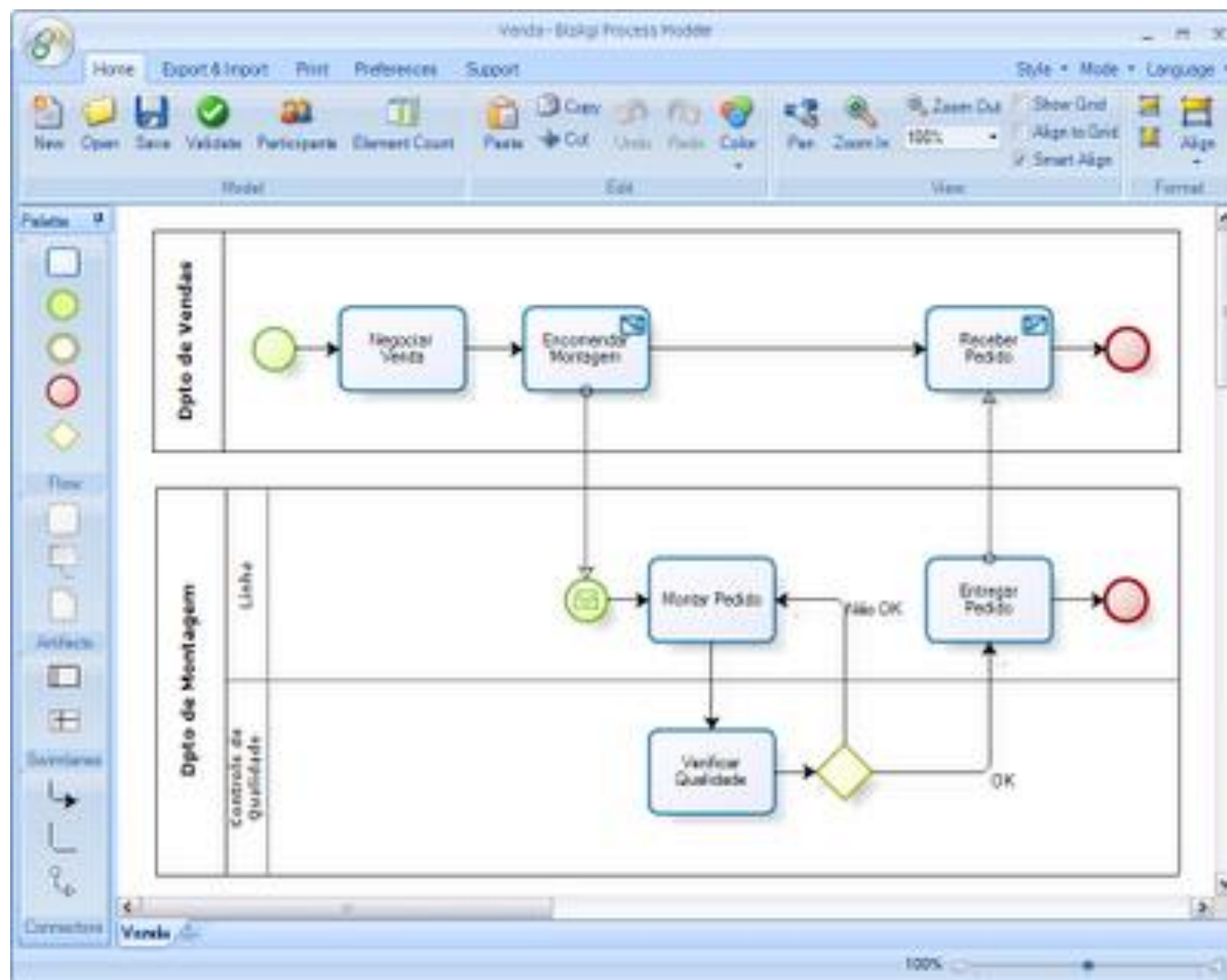
## ◆ BPWin



Oferece inúmeras maneiras de representar processos de negócios, de modo que todos os envolvidos com o processo de modelagem podem fornecer sua perspectiva única.

- “BizAgi é um software BPM (Business Process Management), que permite automatizar os processos de negócio de forma ágil e simples em um ambiente gráfico intuitivo.”
- BizAgi® foi idealizado para modelar processos em BPMN, definir regras de negócio, definir interface do usuário, otimização e balanceamento de carga de trabalho, indicadores de desempenho de processos, monitor de atividades e muito mais.





- Exemplos :

## ◆ Rational Rose

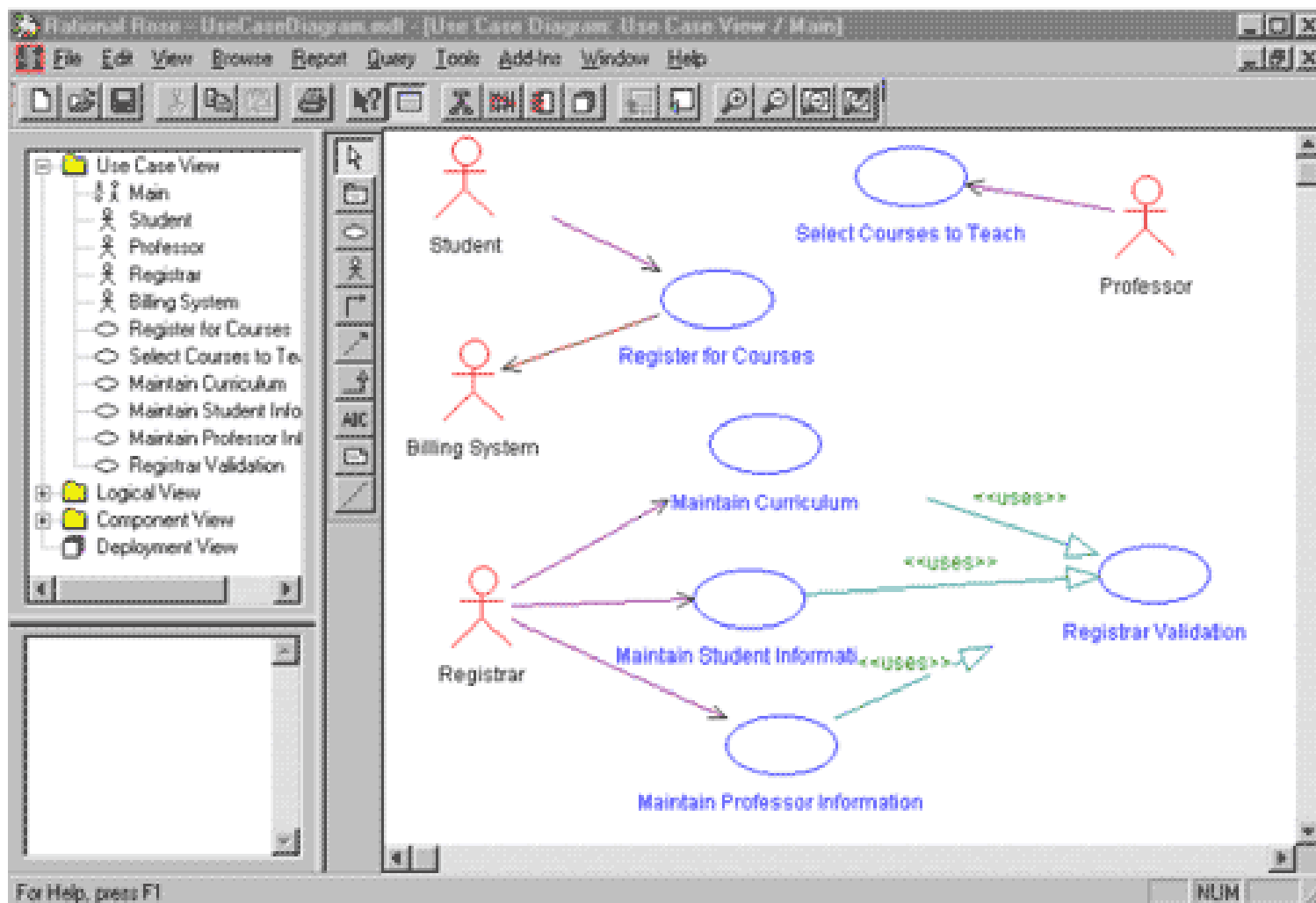
- Ferramenta que unifica toda a equipe de desenvolvimento a partir da modelagem
- UML (Unified Modeling Language)
- Gerencia todas as etapas da engenharia de software
- O Rational Rose Modeler oferece uma ferramenta e linguagem de modelagem comum para simplificar o ambiente de trabalho e permitir a criação mais rápida de software de qualidade.

- Exemplos :

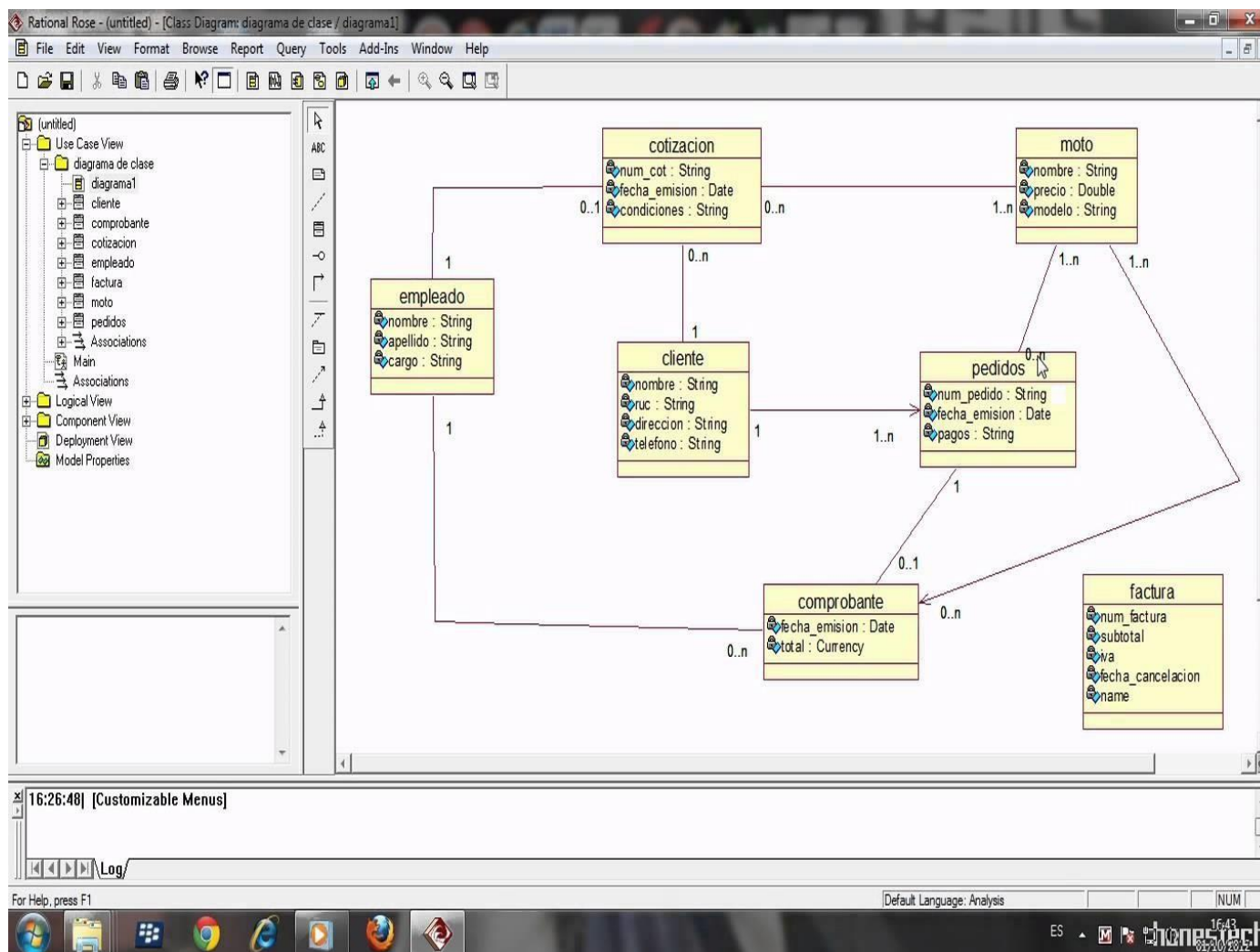
- ◆ **Rational Rose**

- Permite a modelagem com os nove diagramas da UML (Diagramas de Casos de Uso, Diagramas de Classe, Diagrama de Componentes, Diagrama de Desenvolvimento, Diagrama de Objetos, Diagrama de Sequência, Diagrama de Colaboração, Diagrama de Estados e Diagrama de atividades).

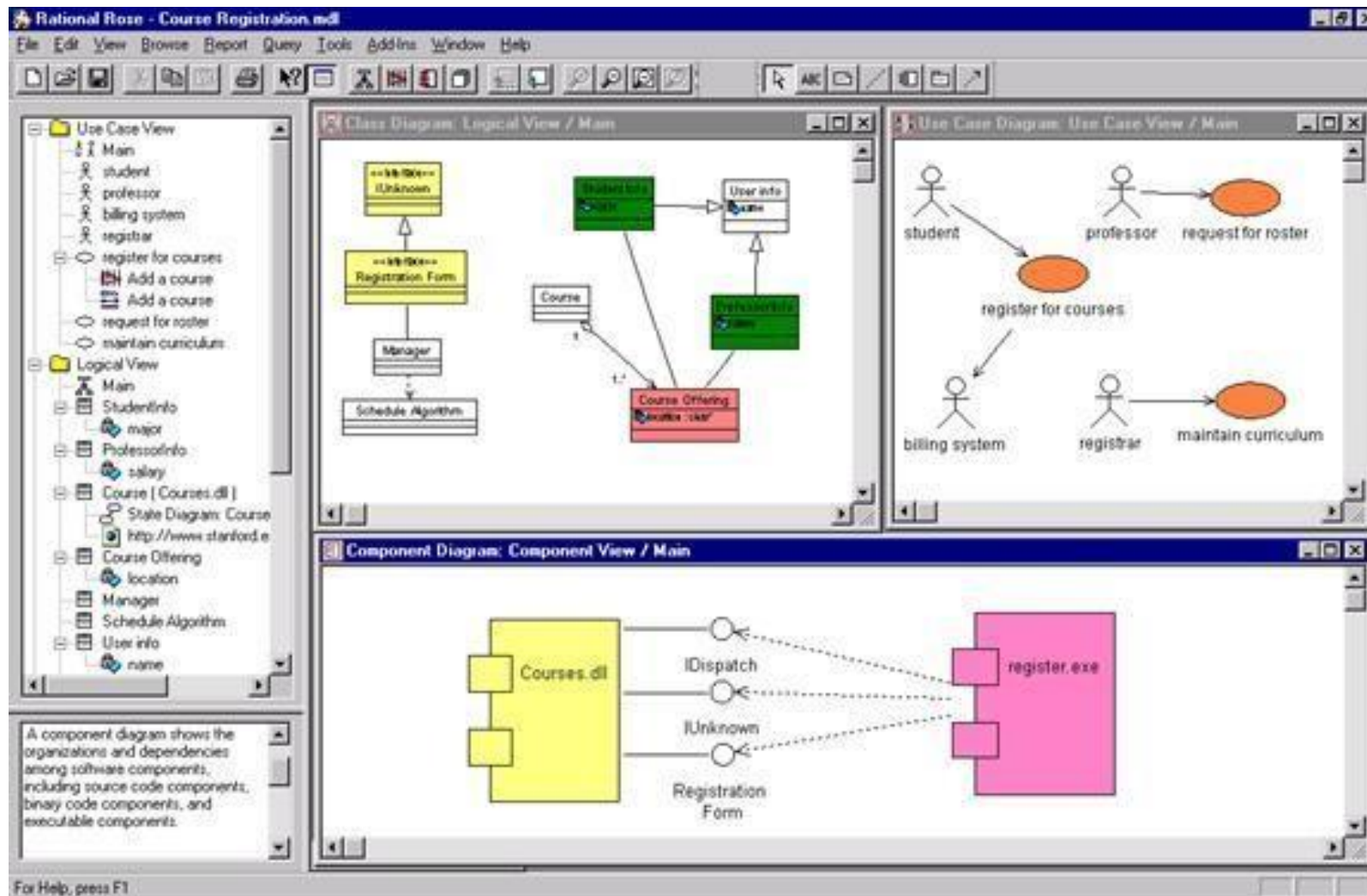
## ◆ Rational Rose



## ◆ Rational Rose



## ◆ Rational Rose



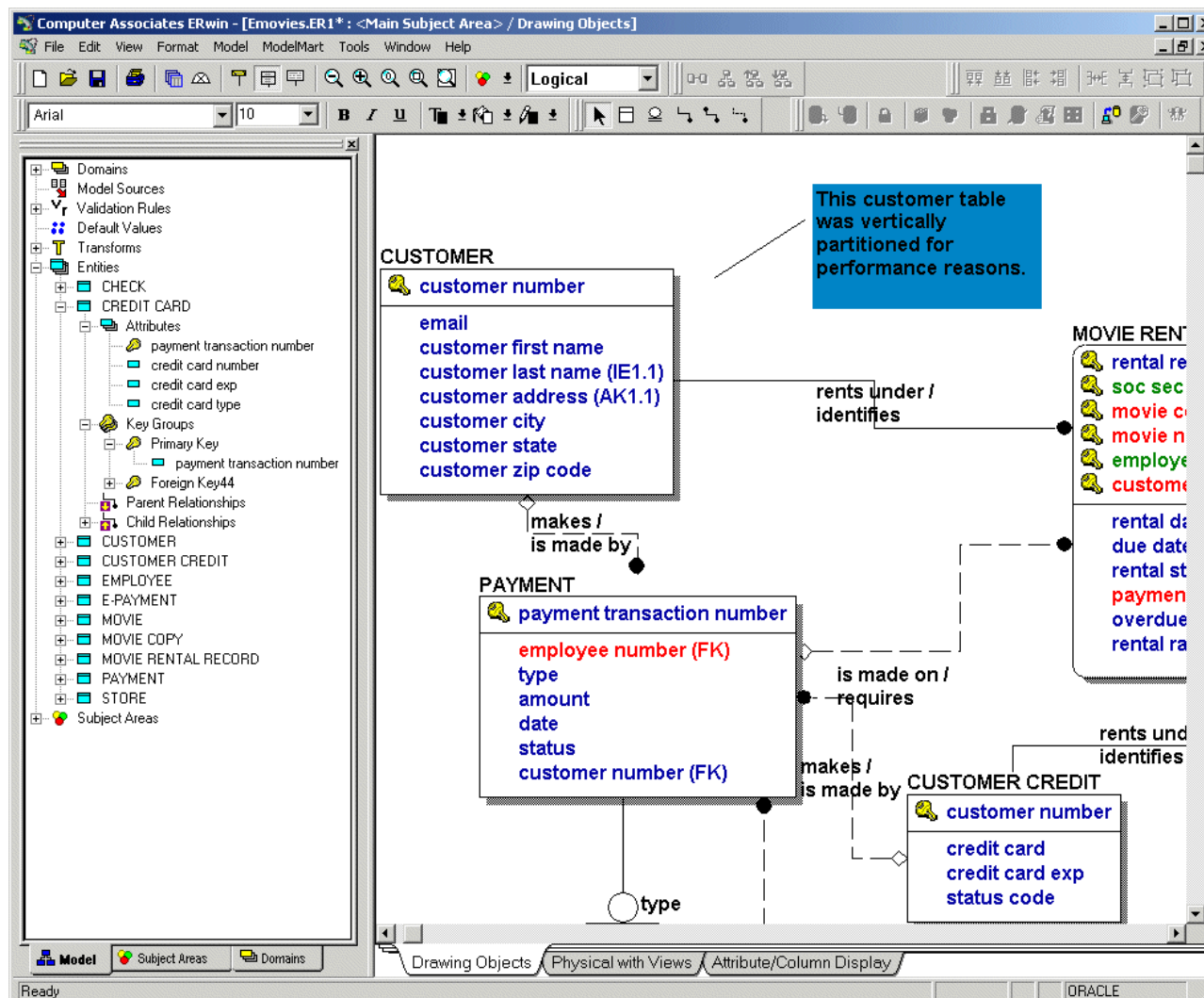


- Exemplos :

## ◆ ERWin

- Poderosa ferramenta de modelagem de dados para ambientes cliente/servidor
- Facilidade de uso com interface intuitiva
- Poderosa ferramenta para documentar sistemas já implantados (sistemas legados)
- Integração com ferramentas front-ends(Delphi, Visual Basic)
- Total independência do banco de dados

## ◆ ERWin

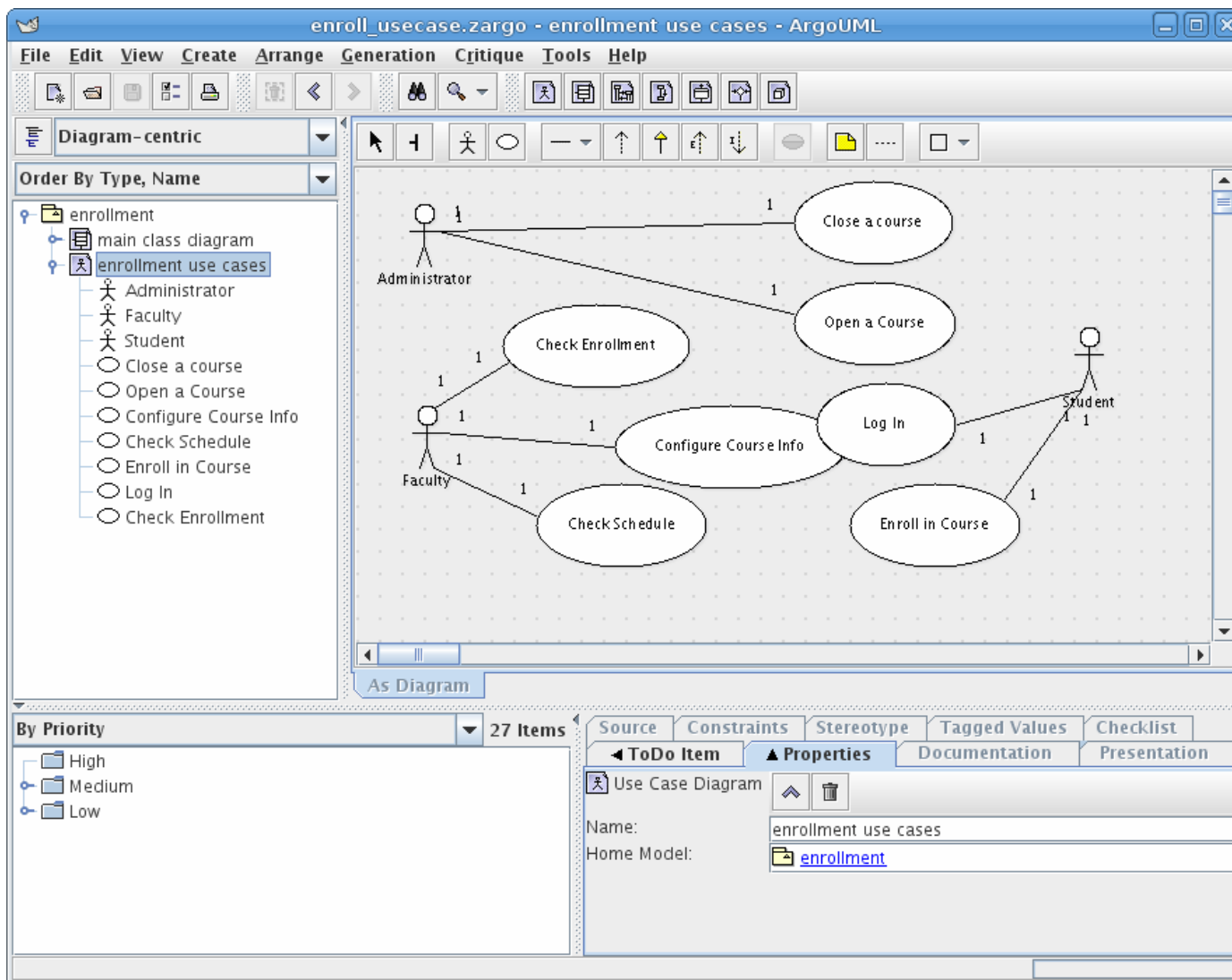


- Exemplos :

## ◆ ArgoUML

- Editor UML open source
- Multiplataforma: **JAVA**
- Visões múltiplas e sobrepostas
- Geração de código em 5 linguagens (Java, C++, C#, PHP4 e PHP5)
- Engenharia reversa para JAVA

## ◆ ArgoUML



- Exemplos :

## ◆ Creately


- Ferramenta de diagramação de propósito geral
- Aplicação nas nuvens (cloud computing)
- Multiplataforma
- Foco em equipes virtuais
- Diagramas em UML, ER, DFD, fluxogramas. Mind maps
- Suporte a trabalho em equipes, controle de versões

# Engenharia de software

## ◆ Creately



Community Browser

creately  Diagram Community

Search types... **Select or Search Diagram Types**

- All
- Flowcharts
- UI Mockups
- Mind Maps
- Site Maps
- Database Diagrams
- Organizational Charts
- Entity Relationship Diagrams
- SWOT Diagrams
- Venn Diagrams
- Concept Diagrams
- Class Diagrams (UML)
- Work Breakdown Structure
- Gantt Charts
- Use Case Diagrams (UML)
- Data Flow Diagrams
- Block Diagrams
- Network Diagrams
- Business Process Management
- Logic Gates
- Object Diagrams (UML)
- PERT Chart

Search diagrams... **Search for Diagrams**

Q Template Popular All My

**Filter by types, including your own**

**Select Diagram to View it in a Separate Tab or Use It**

Web App Support Process ★★★★★

Flow chart Template ★★★★★

Understanding Flow Charts ★★★★★

Support Process - Swimlane ★★★★★

Repair Service - Swim Lane ★★★★★

Banking Teller ★★★★★

ITIL version comparison ★★★★★

Flowchart - basic ★★★★★

Harry Potter and the Deathly Hallows ★★★★★

Video Upload Process ★★★★★

Social Network Flowchart ★★★★★

Shipment process ★★★★★

What is Creately ★★★★★

Recruitment Process ★★★★★

Goal Funnel Process ★★★★★

Credit Card Transaction Process  
★★★★★  
A swimlane flowchart on the Credit Card Transaction process

View Use

- Exemplos :

## ◆ Astah

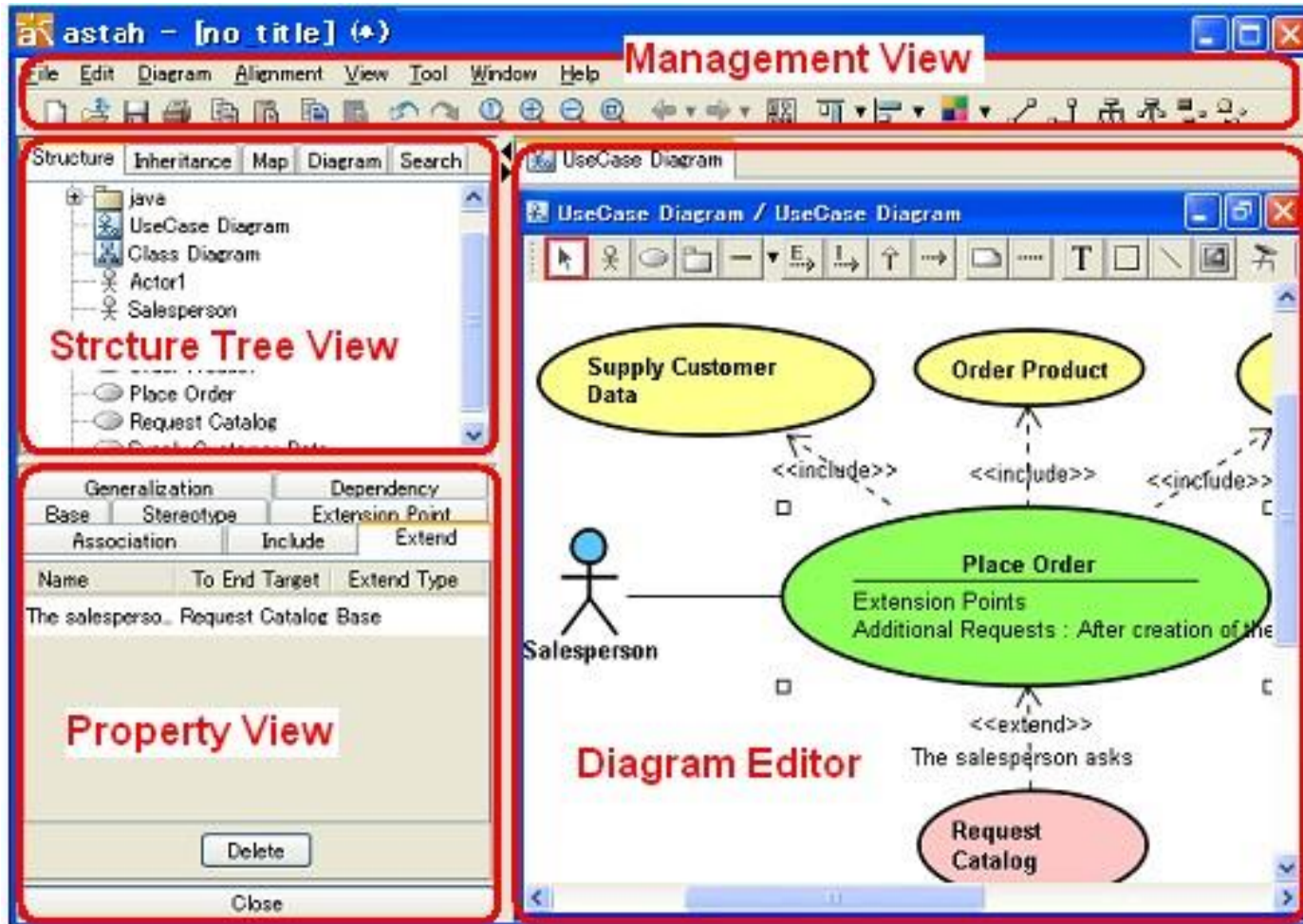
- Astah é uma ferramenta CASE de modelagem UML.
- Astah Community é a versão gratuita para estudantes e uso pessoal.
- **Recomendado para uso educacional e pequenos projetos**
- Interface separada por visões: gerenciamento, projeto e propriedades e editor.
- Diagramas de classe, caso de uso, de estados, de atividades, de sequencia, de componentes, de objetos...



# Engenharia de software

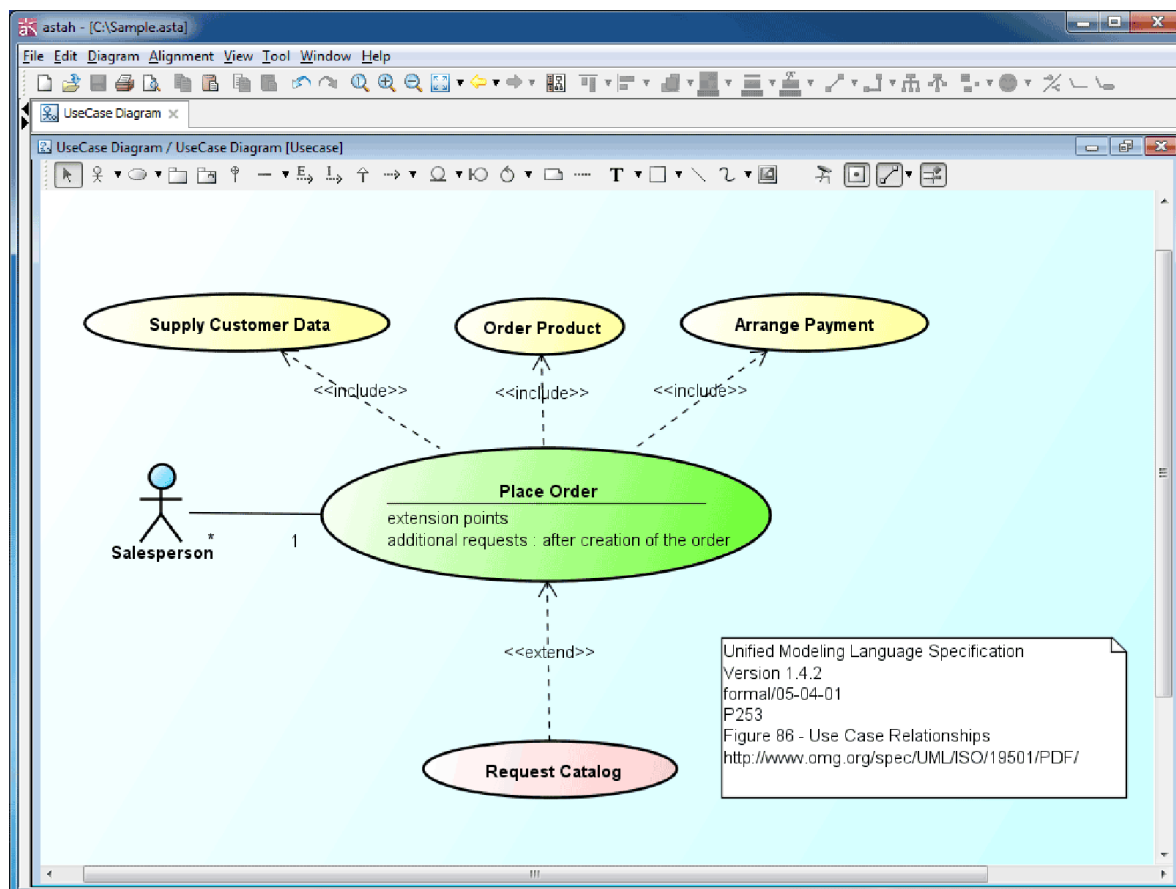
## ◆ Astah

astah  
community





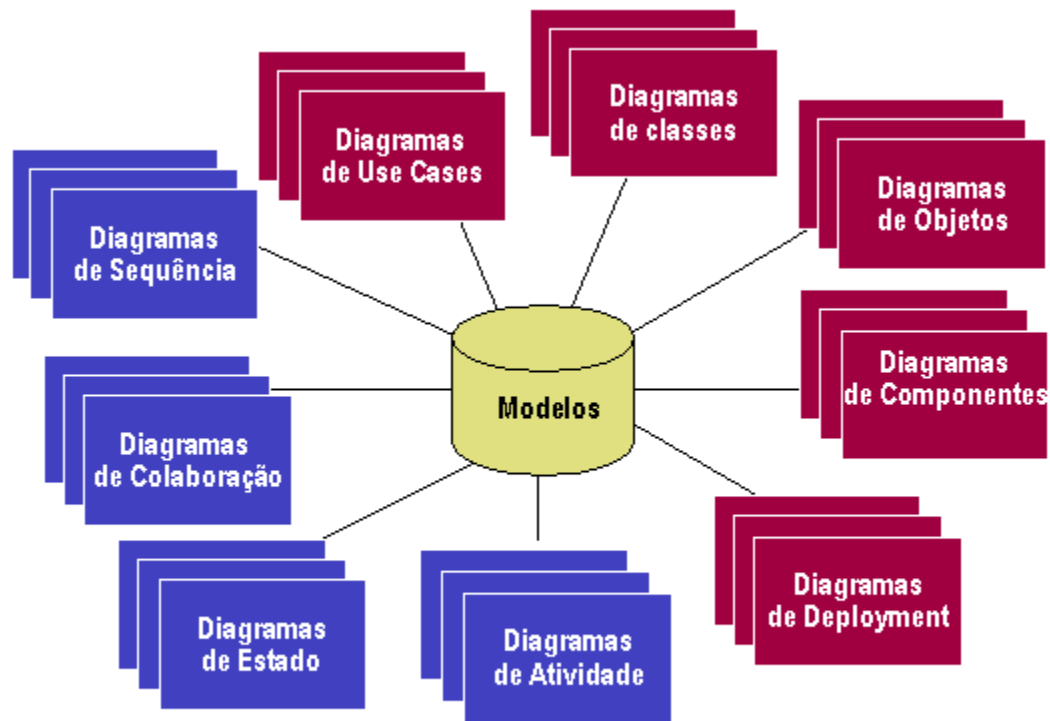
## ◆ Astah



## ■ Unified Modeling Language

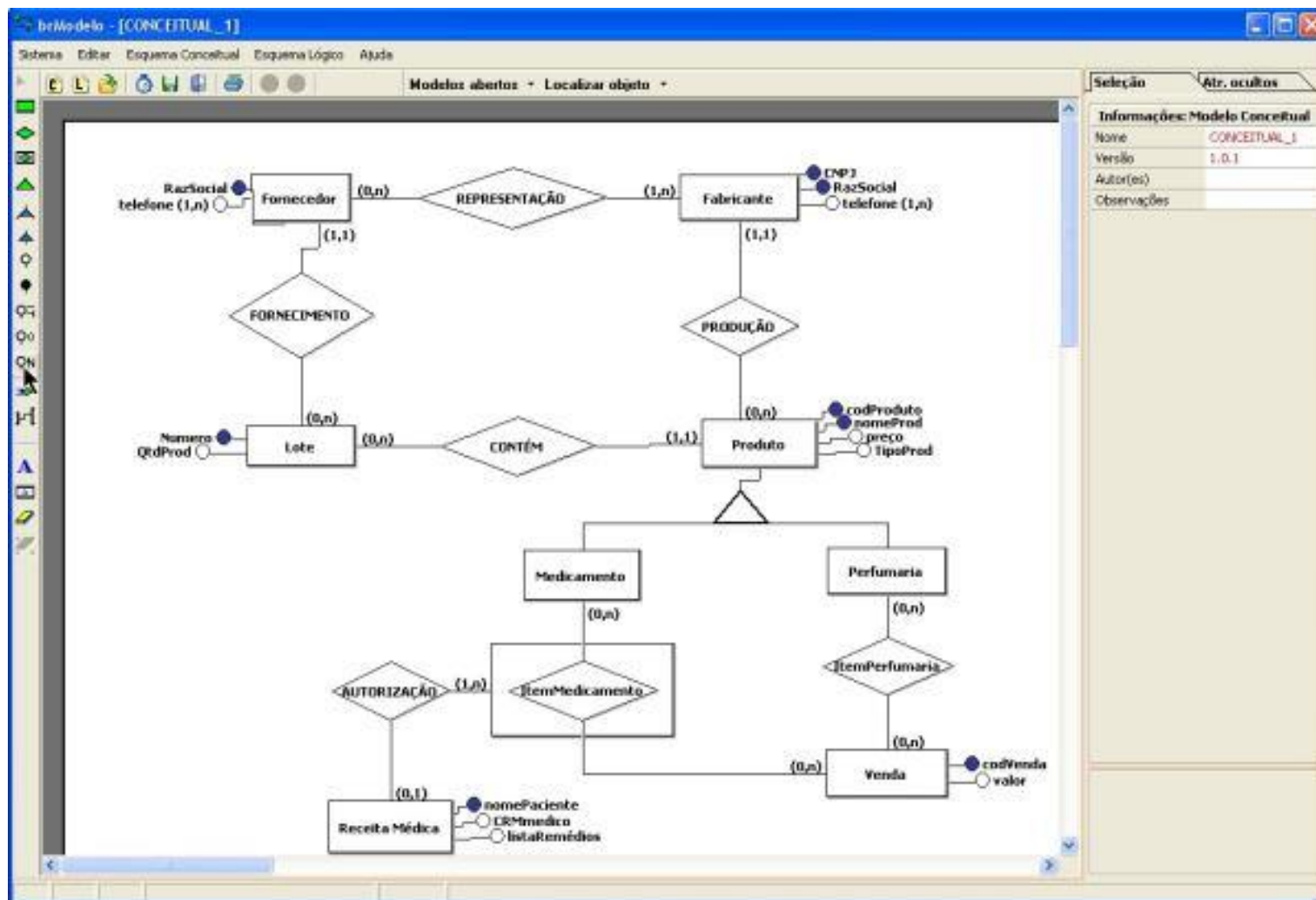
- UML é uma linguagem para visualização, especificação, construção e documentação de artefatos de um software orientado a objeto.
- Sua grande vantagem é que ela é única para as quatro atividades: análise, “design”, implementação e teste.
- O modelo da linguagem é composto de:
  - *itens;*
  - *relacionamentos;*
  - *diagramas.*

- O modelo do sistema é representado pelos dois conjuntos de diagrama, estático e dinâmico (ver figura abaixo).



*Fig1: Vermelho: estático (ou estrutural); Azul: dinâmico (ou comportamental).*

## ◆ brMODELO



## **NORMA ISO/IEC 14102 – AVALIAÇÃO E SELEÇÃO DE FERRAMENTAS CASE**

A norma ISO/IEC 14102 – Avaliação e Seleção de Ferramentas CASE encontra-se em estudo na ABNT (Associação Brasileira de Normas Técnicas). Com o propósito de aplicar esta norma no Brasil a fim de servir como referencial para aqueles que desejam avaliar e selecionar ferramentas CASE.

A norma ISO/IEC 14102 define uma sequência de processos e um conjunto estruturado de características de ferramentas CASE, para uso na avaliação técnica e seleção definitiva de uma ferramenta.

As informações apresentadas na norma devem levar a seleções mais eficientes em termos de custos e a uma maior uniformidade na descrição de funções e características de ferramentas CASE.

- **Bibliografia**

- Paulo Filho, Wilson de Pádua. *Engenharia de Software*. LTC, 2003
- Pressman, Roger S. *Engenharia de Software*. 6ª edição. McGraw-Hill, 2006.
- Sommerville, Ian. *Engenharia de Software*. 8ª edição. Pearson Education, 2007.
- Carvalho, Ariadne M. B. Rizzoni & Chiossi, Thelma C. dos Santos. *Introdução à Engenharia de Software*. Unicamp, 2001.