

# Sugestões de projetos

Aqui, faço algumas sugestões para trabalhos práticos que podem complementar o ensino em um curso de engenharia de software. Estas sugestões vão desde projetos substanciais, que seriam normalmente realizados por um grupo de estudantes, até projetos mais simples, individuais e curtos.

A dificuldade enfrentada na hora de estabelecer qualquer trabalho prático é que os cursos têm períodos de tempo relativamente curtos (de 10 a 12 semanas), e é simplesmente impossível completar industrialmente projetos realísticos no tempo disponível. Se você ministra um curso que se divide em dois semestres, isso simplifica o problema do estabelecimento de projetos mais substanciais. Contudo, se está ministrando um curso de um semestre, então tem de lidar com esse tipo problema. Nas minhas aulas, tenho levado duas abordagens para o problema:

1. Projetos em grupo são conjuntos que envolvem apenas um único aspecto (sem programação) da construção de um sistema. Pode ser a elaboração de uma especificação de requisitos, a construção de uma especificação formal não trivial, a elaboração de um manual de usuário etc. Nesses documentos, pede-se também que os estudantes planejem e elaborem o custo de seu trabalho, ilustrando, portanto, alguns dos problemas de gerenciamento discutidos no livro.
2. Dos estudantes envolvidos em outros cursos, onde o trabalho prático é direcionado, espera-se que, ao fazer este trabalho, empreguem as técnicas discutidas no curso de engenharia de software. Portanto, um curso de técnicas de compilação, por exemplo, pode sugerir que uma abordagem orientada a objetos seja levada para projeto de compiladores; um curso de construção de sistemas embutidos pode envolver a produção de uma especificação de teste abrangente para o sistema que está sendo construído.

## Projetos em grupo

Penso que é importante dar aos estudantes alguma experiência de trabalho em grupo, bem como projetos individuais. Eu prefiro definir os grupos de estudantes ao invés de deixá-los se agruparem. Essa definição pode, naturalmente, ser muito difícil, já que, idealmente, os grupos devem ter uma combinação de personalidades, de capacidades e de gêneros. As informações devem ser extraídas de outras matérias estudadas, de tal modo que exista tempo disponível para reuniões de grupo e trabalho colaborativo. Não existem soluções fáceis para esse problema – você tem de usar seu próprio conhecimento sobre os estudantes para trabalhar com eles, a fim de formar grupos úteis.

Um dos objetivos do projeto é ilustrar os problemas do trabalho em grupo e, em algumas ocasiões, confrontos de personalidades podem surgir. Dependendo das pessoas

envolvidas, os instrutores podem se envolver para resolver questões pendentes ou, simplesmente deixar que os grupos resolvam.

Um aspecto da engenharia de software no qual os estudantes se acham desconfortáveis, é o fato de os usuários de sistema e os compradores terem, geralmente, um conjunto de requisitos vagos e potencialmente contraditórios. A abordagem adotada pelo autor é agir como um usuário e ser deliberadamente vago, contraditório, e apresentar requisitos impossíveis. Esta abordagem conduz às discussões em classe!

O argumento contra projetos em grupo é, naturalmente, que eles tornam a avaliação individual mais difícil. Isso é correto. Contudo, a opinião deste autor é que a educação é mais importante que a avaliação, e que a perda de precisão na avaliação é compensada pela educação melhorada, fornecida pelos trabalhos em grupo. Em geral, os estudantes que trabalham em grupo acham a experiência muito instrutiva e agradável, especialmente quando estão trabalhando com os aspectos do assunto que não envolvem programação.

As pessoas dos grupos podem ser avaliadas por meio de entrevistas individuais, logo após a nota ser dada ao grupo como um todo. Durante essa entrevista, a contribuição individual dos membros do grupo pode ser avaliada e a nota, reavaliada. Com a minha experiência, percebo que os estudantes estão satisfeitos com esse sistema (especialmente depois que a imprecisão inerente de qualquer sistema de atribuição de notas é explicada a eles) e poucas reclamações surgem a respeito disto.

## **Produtos de projeto**

A dificuldade enfrentada na hora de propor projetos é que a maioria dos exemplos reais requer algum conhecimento de domínio de aplicação e existem poucas aplicações gerais compreensíveis. Por isso, sistemas de monitoração de pacientes e sistemas de e-mail são frequentemente usados como fonte de exemplos em livros-texto, simplesmente porque os conceitos são facilmente compreendidos pelos estudantes.

Os projetos sugeridos aqui se classificam em duas categorias. Não são muito gerais e, também, não são implementáveis na maioria dos laboratórios das universidades e faculdades. Eles podem ser especificações grandes, podem requerer acesso a hardware especial ou conhecimento de domínio detalhado para realizar uma implementação. Espera-se que os estudantes elaborem especificações e projetos de alto nível para tais sistemas. O objetivo do trabalho é, ao invés de projetar um sistema válido, ilustrar os problemas de elaboração de especificações e projetos. Em alguns casos, quando linguagens e equipamentos estão disponíveis, é possível fazer o protótipo de partes do sistema.

Os documentos que poderiam ser produzidos são:

- Definição e especificação (parcial) de requisitos.
- Projeto de arquitetura preliminar.
- Plano de projeto e cronograma.
- Protótipo de parte da interface do usuário do sistema.

Em todos os casos, é recomendado que uma definição de requisitos seja produzida para expandir as propostas de projeto sugeridas. Naturalmente, isso pode permitir uma pequena semelhança com os requisitos verdadeiros do sistema, porém, como os requisitos são razoáveis, isso não é, de fato, importante.

Uma segunda classe de projeto, são os projetos em escala menor no qual um estudante ou um grupo poderia explicar desde a especificação inicial, até a implementação. Dependendo do tempo disponível, esses projetos podem ser considerados simplesmente como projetos de especificação. Em tais projetos, os estudantes poderiam produzir alguns dos seguintes documentos:

- Uma especificação de requisitos que expanda (abaixo) os documentos preliminares em mais detalhes.
- Uma especificação formal para parte do projeto.
- Um projeto de arquitetura preliminar.
- Uma especificação de projeto detalhada.
- Uma especificação de testes.
- Um manual de usuário e quadros de auxílio associados.
- Um plano de projeto e um cronograma que definam os marcos, o uso de recursos e os custos estimados.
- Um plano de qualidade que defina o procedimento de garantia de qualidade.

Eu, normalmente, não espero que os grupos de estudantes produzam todos esses documentos. Você tem de, dependendo da orientação do seu curso, escolher os mais apropriados.

Embora, pessoalmente, eu não tenha usado esta abordagem, alguns dos meus colegas pediram aos grupos que produzissem uma apresentação em pôster descrevendo seu trabalho. Enquanto o pôster de um dos grupos era apresentado, os membros dos outros grupos poderiam levantar questionamentos sobre aquele trabalho. Ambos responderam positivamente, e os melhores pôsteres eram, subsequente, apresentados nos laboratórios de computação. Isso significava que esses estudantes, anos mais tarde, poderiam facilmente compreender o padrão de trabalho que foi esperado deles.

## **Projetos**

### **Um sistema de comando e controle de veículo policial**

É requisito de um serviço policial que ele responda o mais rápido possível aos incidentes reportados, e que o objeto de um sistema de comando e controle assegure que os incidentes serão registrados e direcionados ao veículo policial mais apropriado. Os fatores que devem ser levados em conta na hora de decidir para qual veículo enviar cada incidente são:

- **O tipo de incidente.** Alguns incidentes são mais sérios que outros, e necessitam de uma resposta mais urgente. É recomendado que classes de respostas sejam identificadas e que os incidentes sejam alocados à essas classes.
- **A posição de veículos disponíveis.** Em geral, a melhor estratégia é enviar a mensagem ao veículo mais próximo para este possa responder ao incidente. Leve em conta que a posição dos veículos pode não ser conhecido exatamente, e que pode ser necessário enviar uma mensagem aos veículos para determinar sua posição atual.
- **Os tipos de veículos disponíveis.** Alguns incidentes requerem uma série de veículos, outros, tais como acidentes de trânsito, podem requerer veículos especializados, e assim por diante.
- **A localização do incidente.** Em algumas áreas, pode ser imprudente responder a um incidente enviando um único veículo. Em outras áreas, um único veículo ou policial pode ser suficiente para responder ao mesmo tipo de incidente.
- **A necessidade de alertar outros serviços de emergência, tais como serviços de bombeiro e ambulância.** O sistema deve, se necessário, alertar automaticamente esses serviços.
- O sistema deve permitir que o relator registre os detalhes do incidente.

Um sistema tal como esse é aberto para expansão quase indefinida. Por exemplo, os veículos policiais incluem terminais de fax, de modo que informações escritas podem ser enviadas via fax quando um veículo for encaminhado a um incidente. Uma interface de usuário para uma sala de controle de sistema poderia ser prototipada, e assim por diante.

## Um sistema de monitoração de alarme de fogo e proteção

Um grande edifício pode requerer um sistema de alarme automatizado, que monitora e controla todos os alarmes de fogo e proteção do edifício. Normalmente, o edifício é dividido em zonas, cada uma com uma série de alarmes associados à ela. Os alarmes alertam uma área central controlada por pessoas que podem transmiti-los para os serviços de emergência, ou respondê-los pessoalmente.

Os fatores que devem ser levados em consideração em um sistema de edifício são:

- Se a área de controle não for controlada por pessoas e, caso um alarme seja ativado, esse alarme não deve ser ignorado, já que é potencialmente sério. Os serviços de emergência devem ser chamados imediatamente.
- Algumas, mas nem todas as partes do edifício, podem ser equipadas com sistemas de *sprinkler* ou com sistemas para a desativação de equipamentos elétricos. Esses sistemas devem ser ativados se um alarme de fogo for confirmado. Porém, não devem ser ativados caso haja pessoas na mesma sala.
- O edifício pode ser equipado com indicadores de direção que iluminam o caminho para a saída mais próxima. Eles devem ser ativados quando um alarme de fogo é confirmado. Ao mesmo tempo, um sinal audível deve ser emitido alertando as pessoas a deixarem o edifício.

- Um alarme de proteção pode causar o fechamento automático de algumas portas internas. Deve ser possível isolar zonas completas pelo fechamento automático de portas.
- Alarmes falsos são comuns e deveria ser uma prática normal ter a confirmação do alarme antes de alertar os serviços de emergência. Existem caminhos diferentes de confirmação de alarme. No caso de um alarme de fogo, pode ser confirmado por múltiplos sensores que detectam um problema.

## **Um sistema integrado de informações do departamento da universidade**

Os departamentos de ciência e de engenharia da computação das universidades têm, freqüentemente, orçamentos que são comparáveis a pequenas empresas, e um sistema integrado para apoiar todos os aspectos da administração do departamento é necessário. Ele deve apoiar os registros dos estudantes, a administração do laboratório, os pedidos de produtos e serviços, os pagamentos de entrada e saída, os pagamentos feitos para os professores, os relatórios de pesquisa, etc. Deve estar ligado a um sistema mais amplo da universidade, responsável pelos salários do pessoal, etc.

Os fatores que devem ser levados em consideração são:

- Os chefes de departamento que usam tais sistemas são, em geral, muito ocupados. Mesmo que eles estejam familiarizados com a informática, necessitam de um sistema com interface de usuário direta.
- Os usuários do sistema variam, desde secretárias, até técnicos para ensino e pessoal administrativo.
- Em alguns países, a tendência é as universidades passarem a responsabilidade da administração da organização central para os departamentos individuais. O sistema deve ser capaz de ser expandido para tratar tarefas futuras e imprevistas.
- Os subsistemas devem ser automaticamente ligados de tal modo que, por exemplo, os custos de uma classe ou laboratório particular sejam calculados, considerando os pagamentos feitos e recebidos.

## **Um sistema de gerenciamento de conferências**

Atualmente, sistemas simples de conferência ou quadro de boletins são amplamente usados, mas, em geral, têm interface baseada em caracteres (e, por esse fato, não podem ser acessados por meio de linhas discadas), uma linguagem de comando criptografada e apresentada na forma somente texto. Este projeto envolve a construção de um sistema de gerenciamento de conferências para ser usado por gerentes via suas estações de trabalho. Deve possuir suporte para conferências múltiplas.

Esse sistema é diferente dos sistemas de conferências gerais em vários aspectos:

- Todas as estações de trabalho têm telas gráficas de bitmap e são conectadas por uma rede banda larga local. O apoio de linhas discadas não é necessário.

- Os gerentes que usam o sistema necessitam de uma interface de usuário de fácil uso. Uma interface de linguagem de comando baseada em texto não é aceitável.
- A mistura de texto e gráficos deve ser suportada.
- Deve incorporar um recurso de e-mail.
- A comunidade de usuários é fechada. É possível incorporar conhecimento de perfis particulares e individuais no sistema. Isso pode levar a algumas pessoas relacionadas somente com conferências particulares.

## **Um sistema de automação de biblioteca**

Uma biblioteca requer um sistema integrado, completamente automatizado e que controle pedidos, catalogações, publicações e devoluções de livros. A intenção é que o catálogo seja acessível aos usuários, e que eles sejam capazes de acessar esse sistema remotamente, por meio de linhas discadas. Os usuários devem ser capazes de solicitar livros que estão atualmente emprestados, fazendo a reserva do livro no catálogo da biblioteca.

Os fatores que devem ser levados em consideração são:

- Os usuários da biblioteca podem não ter experiência específica em informática. Contudo, a necessidade de um de browser de catálogo remoto, significa que uma interface baseada em texto é necessária.
- O pedido de livro depende do orçamento de compra. Se não existir orçamento, nenhum livro pode ser pedido. Pode haver múltiplos orçamentos de compra de livros para serem administrados.
- O sistema deve ser capaz de gerar, para os funcionários da biblioteca, relatórios sobre pedidos de livros pendentes, sobre a frequência de empréstimos, etc.
- Os livros trazem um código de barras quando comprados E esse código é utilizado para registrar o empréstimo do livro no sistema. Os usuários da biblioteca também possuem cartões de identificação que incorporam um código de barras.

## **Um sistema de informações de museu**

Um museu requer um sistema de informações automatizado para uso dos visitantes e que ajude a localizar itens e descobrir mais sobre os itens que estão no display. O sistema é destinado ao uso em um museu de computadores que contém muitos computadores antigos, máquinas modernas que executam simulações, livros e registros de áudio dos pioneiros no desenvolvimento de computadores. Fatores que devem ser levados em consideração no projeto são:

- O sistema deve ter um sistema de ‘ande e use’ para os visitantes que não têm treinamento, independente do sistema de computador em uso.
- Você precisará gerenciar os andares do museu e a localização de itens nesses andares. As informações sobre itens específicos devem estar acessíveis por meio da indicação de sua localização.

- O sistema terá de gerenciar informações multimídia – sons, imagens e vídeo.
- Deve ser possível para os funcionários do museu, mudar e adicionar informações sobre novas exposições.
- Quando você fornece às pessoas informações sobre onde encontrar um item, elas respondem melhor à instruções simples, como a referência a edifícios antigos famosos, que a mapas, que freqüentemente elas encontram dificuldade para ler.

## **Um sistema de entrega de jornais**

Este sistema se destina ao gerenciamento da entrega de jornais e revistas em uma cidade pequena ou em alguma área de uma grande cidade. É voltado para uso de distribuidores que são usuários casuais de sistemas de computador, e que devem operar um ou PC ou um hardware similar. Fatores que devem ser levados em consideração na especificação e projeto deste sistema são:

- O sistema deve imprimir, a cada dia, as publicações a serem entregues em cada endereço.
- O sistema também deve imprimir, para o distribuidor, um resumo de quem recebeu quais publicações em cada dia.
- Uma vez por mês, as contas são entregues aos clientes juntamente com seus jornais. Essas notas devem ser calculadas automaticamente pelo sistema.
- Os clientes viajam, e podem estar fora temporariamente, de férias ou em viagens de negócio.
- Nem todos os clientes têm entregas diárias.
- O sistema deve ser capaz de gerenciar algumas informações geográficas simples, de tal modo que imprima informações sobre as pessoas, para que as publicações sejam entregues a elas.

É um sistema relativamente pequeno em comparação com alguns outros, portanto, seria possível completar sua implementação em um projeto de dois semestres.

## **Um sistema automatizado de pedágio.**

Em muitos países, tais como EUA e França, é comum termos de pagar pedágios nas rodovias interestaduais ou nas rodovias expressas. O pedágio aplicado é referente à distância viajada. Contudo, as estações de pedágio precisam que os veículos parem, o que, quando o tráfego está intenso, pode causar um congestionamento significativo. O objetivo desse projeto é automatizar a coleta de pedágios dos automóveis usando a comunicação da rodovia e os computadores de bordo, de tal modo que não haja necessidade de os automóveis pararem quando estiverem passando por uma estação de pedágio. Os fatores que devem ser levados em consideração ao projetar o sistema são:

- Os automóveis podem estar equipados com um sistema de bordo que estabeleça uma comunicação entre o identificador do automóvel e o equipamento de pedágio ao lado da rodovia.
- A tecnologia *smart-card* disponível atualmente, permite aos motoristas comprar um cartão para ‘carregá-lo’ em qualquer posto bancário.
- Você precisa ter um mecanismo para cobrar os veículos que não estão equipados com um sistema de bordo.
- As estações de pedágio devem estar situadas nas intersecções e, normalmente, não devem ser operadas por pessoas. Os sistemas devem realizar uma auto-monitoração e relatar os defeitos a um centro de controle.
- Câmeras podem fotografar os números da placa e reconhecer automaticamente a licença dos automóveis.
- O sistema pode ter de se integrar a outros sistemas computacionais, particularmente, aqueles que mantêm informações sobre licenciamento de automóveis.

Existe, nesse projeto, um escopo para vários exercícios de simulação de apoio ao sistema. Por exemplo, é possível solicitar que os estudantes escrevam um simulador que simule o comportamento dos grupos de veículos, emitindo sinais de seus equipamentos de bordo.

Para maiores detalhes sobre esse sistema, você pode baixar meus slides sobre engenharia de sistemas baseada em computadores.

## Projetos de conclusão

Agrupei os projetos aqui sob uma série de categorias e, obviamente, existe um escopo para a integração de projetos dentro de uma mesma categoria e modificação com idéias para outros projetos. O problema enfrentado naturalmente, é que os projetos muito específicos de domínio não podem ser definidos para um grupo de estudantes. Portanto, escolhi categorias onde os instrutores podem fornecer recomendação de domínio ou onde os próprios estudantes são capazes de usar seu conhecimento genérico para desenvolver uma especificação de sistema mais detalhada. As categorias são:

1. *Componentes de software.* São os componentes usados diretamente como parte de outros sistemas, ou os que podem gerar código de programa ou dados.
2. *Ferramentas de engenharia de software.* São ferramentas que apóiam atividades de engenharia de software específicas, tais como requisitos, projeto, etc. Você pode, naturalmente, sugerir também que as capacidades oferecidas pelas ferramentas CASE comerciais sejam implementadas.
3. *Sistemas de apoio à universidade.* São sistemas de software que apóiam algumas funções de um departamento da universidade. Aqui, o instrutor é o especialista de domínio e deve estar preparado para oferecer recomendações aos estudantes sobre o que é requisitado.
4. *Sistemas de gerenciamento de informações.* São sistemas que gerenciam alguns tipos de informações, como por exemplo, fotografias. Podem ser úteis se você



pretende desenvolver projetos relativamente simples que gerenciem, por exemplo, informações de texto, no qual você deseja usar os conceitos de interface de usuário ou um banco de dados e 4GLs. Você pode também estendê-los às informações multimídia, o que aumenta o nível de complexidade para os estudantes. São todos adequados para implementações baseadas em Internet.

5. *Sistemas de simulação.* São sistemas que simulam alguma outra máquina. Eles são particularmente úteis para proporcionar experiência em desenvolvimento de sistemas de tempo real.
6. *Sistemas Web.* São sistemas baseados em Web sites ou em HTML. São particularmente úteis para dar aos estudantes a experiência de descobrir componentes existentes que são disponibilizados gratuitamente.

Uma maneira possível de lidar com grupos maiores é um grupo implementar componentes de framework e outro construir um sistema usando esses componentes. Isso é muito eficiente na hora de ilustrar para os estudantes a importância de interfaces bem definidas.

Assumi que os projetos serão desenvolvidos em computadores pessoais (PC ou Macintosh) ou em estações de trabalho Unix ou Linux. Sistemas Unix e Linux são mais fáceis quando os estudantes não têm experiência em programação de interfaces gráficas, e também quando você não deseja desenvolver essa habilidade particular no projeto. Qualquer linguagem de programação apropriada que você apóie pode ser usada.

Obviamente, muitos desses projetos são adequados para implementações baseadas em Web usando CGI, Javascript e Java applets. Em alguns casos, outras linguagens de *scripting* tal como Tcl/Tk também podem ser usadas.

## **Componentes de software**

### **Uma árvore abstrata de sintaxe para algumas linguagens de programação**

O objetivo deste projeto é construir uma representação abstrata da sintaxe de uma linguagem de programação. Recomendo C ou Pascal para os projetos de estudantes, já que são linguagens relativamente simples e é possível completar o projeto em um período ou semestre. Também é possível usar linguagens mais complexas, tais como Ada ou C++, mas levam mais tempo. Os estudantes devem ter algum conhecimento sobre construção de compiladores e, recomendo o uso de ferramentas, tais como Lex e Yacc para construir um analisador de linguagem. Uma interface gráfica simples deve ser desenvolvida para apresentar a árvore abstrata de sintaxe. Uma vez construída, essa árvore pode ser a base para uma série de outros projetos de construção de ferramentas orientadas a linguagens.

## **Um conjunto de componentes para apoiar a edição de gráficos**

Sistemas de edição gráfica envolvem a manipulação de objetos gráficos simples, tais como círculos, retângulos, etc. Várias ferramentas e um conjunto de componentes básicos podem ser usados para simplificar a construção desses editores. O objetivo deste projeto é desenvolver um conjunto de objetos ou tipos de dados abstratos que possam ser usados como base para a construção de editores gráficos. Você deve incluir componentes para representar retângulos, elipses, polígonos, etc., com a capacidade de conectar esses componentes usando linhas de espessuras diferentes. Deve ser possível colorir entidades, movê-las, redimensioná-las, agrupá-las e desagrupá-las, etc.

## **Um sistema de auxílio e de mensagem integrados**

Esse sistema deve ser projetado como um conjunto de tipos abstratos de dados, de tal modo que possa ser incorporado como um componente em outros sistemas. Seu projeto deve levar em consideração que o sistema pode ter de apresentar auxílio e mensagens em níveis de detalhes e em linguagens diferentes.

## **Um gerador de tela e de formulários**

O objetivo desse projeto é construir um sistema que permita ao usuário criar formulários e projetos de tela de maneira interativa, por meio da especificação de campos no formulário, bem como seus conteúdos e suas restrições. O sistema pode interagir com algum sistema de gerenciamento de banco de dados. Preste atenção ao fato de que o texto apresentado nos campos pode exceder o tamanho do campo e, assim, o *scrolling* de cada campo individual deve receber suporte

## **Ferramentas de software**

### **Ferramentas de análise de programa**

O objetivo deste projeto é construir ferramentas de análise estática e/ou dinâmica de programa. Essas ferramentas podem prover recursos de análise estática, conforme discutidos no capítulo 24, ou análise dinâmica, mostrando com que frequência cada declaração foi executada no programa. Você pode adicionar alguns recursos de análise, tais como um recurso que mostre onde os nomes definidos no programa são usados, ou um recurso que analise a alocação dinâmica da memória no programa (todo armazenamento que é alocado deve ser desalocado). Uma árvore abstrata de sintaxe é necessária para esse projeto.

## Ferramentas de visualização de programa

O objetivo deste sistema é gerar representações visuais de um programa em uma linguagem de programação. Várias representações poderiam ser desenvolvidas, tais como uma representação gráfica da estrutura de chamada do programa, uma representação da estrutura de composição, uma representação da hierarquia de escopo, etc. Para linguagens orientadas a objetos, você poderia gerar diagramas de uso de serviços ou diagramas de agregação. Esse tipo de sistema tem escopo amplo, que vai desde sistemas simples, com uma única janela, até sistemas complexos com múltiplas janelas e uma grande variedade de opções de *display*. Uma árvore abstrata de sintaxe é necessária para este projeto.

## Um sistema de teste de compilador

O objetivo deste sistema é construir um gerador de entrada de compilador que pode ser usado para testar sistemas de compilação. Você usa a gramática da linguagem definida para gerar *strings* que representam os programas corretos e incorretos, e os integrar a um sistema que submeta as entradas e processe as saídas do compilador para verificar quais *strings* sintaticamente corretos foram classificados sem mensagens de erro, e quais mensagens de erro foram geradas para programas incorretos.

## Workbenches de projeto

Existe uma variedade de possíveis projetos onde *workbenches* de vários tipos são construídos. Podem incluir os componentes discutidos no capítulo 26, tais como editores de projeto para representações diferentes, geradores de relatórios, geradores de código, etc. Tal projeto pode envolver grupos substanciais (até 10 ou 12 estudantes), onde os subgrupos são responsáveis pelo desenvolvimento de componentes de infra-estrutura e outras ferramentas de *workbench*.

## Sistemas de apoio à CBSE

O objetivo deste projeto é construir um catálogo de componentes de software no qual os usuários fazem pesquisas para descobrir quais são os componentes de software reusáveis. Pode ser acessado usando teclados associados a cada componente, e deve conter a opção de entrar e classificar os componentes. A interface para o sistema pode ser, ou uma interface de linguagem de consulta, ou uma interface baseada em formulários. Uma extensão do projeto é implementar a capacidade de pesquisa gráfica baseada em Web para catálogo de componentes, e permitir que os componentes sejam baixados de lugares remotos.

## Sistema de controle de mudanças e de geração de relatórios

Esse sistema dedica-se a apoiar um sistema de controle de mudanças e geração de relatórios, conforme discutido no capítulo 32. Ele deve permitir que as mudanças sejam

registradas, submetidas a aprovação e, subseqüentemente, acompanhadas. Ele poderia ser integrado a um sistema de e-mail, de modo que os relatórios de mudanças possam ser passados automaticamente ao CCB.

### **Um sistema de rastreabilidade de requisitos**

Esse sistema deve ser projetado para analisar requisitos escritos em linguagem natural e para descobrir os relacionamentos entre eles. A idéia básica é que os requisitos relacionados devem se referir aos mesmos sistema, entidades de domínio e relacionamentos e, desse modo, o sistema varre os requisitos em linguagem natural a procura de requisitos que usam os mesmos termos técnicos. Você precisa ser capaz de treinar o sistema para operar em um domínio particular e definir exclusões – palavras que são tão comuns que devem ser sempre ignorados.

### **Um sistema de base de projeto**

Uma base de projeto está relacionada a fornecimento das razões das tomadas de decisão particulares de um projeto. Um sistema de base de projeto permite que as propostas sejam feitas, e que os argumentos prós e contras dessas propostas sejam mantidos. Você poderia, também, incluir a noção de riscos em tal sistema onde, além de simplesmente colocar uma proposta de projeto, o proponente deve também ser capaz de descrever os riscos e alguns valores numéricos associados a eles. O sistema de base deve ser capaz de ajudar na tomada de decisões baseadas nesses fatores de risco, bem como em argumentos técnicos.

## **Sistemas de apoio à universidade**

### **Um sistema de registros de estudantes**

O objetivo deste projeto é manter um sistema de registros de estudantes, de maneira que esses registros fiquem dentro de um único departamento da universidade ou faculdade. O sistema deve permitir o registro de detalhes pessoais, bem como as classes, as notas, etc. Ele deve fornecer, ainda, recursos de sumarização que permitam a recuperação de informações sobre um grupo de estudantes. Admita que o sistema é operado pelo pessoal administrativo do departamento, que não possui experiência em informática. Assim, esse projeto poderia ser implementado em uma linguagem de programação de banco de dados, ou na linguagem de VisualBasic.

## **Um sistema de gerenciamento de laboratório**

Esse sistema é dedicado a apoiar a administração de um laboratório de graduação, onde equipamentos diferentes estão disponíveis aos estudantes para a realização de experimentos diferentes em tempos diferentes. O sistema deve ajudar com a programação dos experimentos e manter os registros das compras de equipamento, da confiabilidade e da manutenção. É basicamente um projeto de banco de dados e, assim, poderia ser implementado usando uma linguagem de programação de banco de dados.

## **Um sistema de submissão de trabalho de curso**

Esse sistema deve gerenciar a entrega eletrônica de trabalhos dos estudantes para os instrutores. Deve manter o acompanhamento de quais estudantes entregaram os trabalhos, e deve permitir aos instrutores gerenciar as notas desses trabalhos. Deve, ainda, fornecer resumos das notas, além de recursos para retornar eletronicamente informações sobre os estudantes. Você poderia considerar algum tipo de sistema de anotações onde os instrutores poderiam comentar sobre os trabalhos em versões eletrônicas. Se você aborda proteção em seus cursos, você poderia incluir recursos de criptografia.

## **Um sistema de admissão**

Esse sistema deve auxiliar o pessoal administrativo do departamento a gerenciar o processo de admissão de novos estudantes de graduação e graduados. Ele deve manter um acompanhamento de formulários recebidos, bem como o progresso e as informações que foram enviadas em cada formulário. Deve incluir avisos automáticos ao pessoal quando ações particulares devem ser tomadas e quando prazos são atingidos. O sistema deve gerar automaticamente as cartas padrões para os formulários.

# **Sistemas de gerenciamento de informações**

## **Sistema de agenda e compromissos de grupo**

Esse projeto está relacionado ao fornecimento de uma agenda compartilhada, onde os compromissos de um grupo são registrados. Quando uma reunião que envolve mais de um membro de grupo é necessária, suas agendas são consultadas, e uma data adequada é mutuamente descoberta. Os compromissos podem ser movidos para constituir uma data adequada. É preciso atentar para o fato de que alguns compromissos não podem ser movidos.

## **Processador de idéias**

Este projeto está relacionado ao fornecimento de um meio de organização de idéias. Ele deve apoiar a entrada e a movimentação de títulos, bem como a associação de texto com títulos, etc. Não creio que uma estrutura simples hierárquica seja apropriada. Deve ser possível formar uma rede de idéias. Por isso, faça uso de capacidades gráficas, se estas estiverem disponíveis.

## **Sistema de gerenciamento de dados pessoais**

O objetivo deste sistema é construir um pequeno sistema de gerenciamento de banco de dados usado para registrar informações pessoais, tais como detalhes de registros e livros, informações sobre receitas, e assim por diante. Eles podem ter diversas informações registradas e o sistema deve ser usado por aqueles que não possuem experiência em informática. Uma abordagem possível seria construir um sistema de HyperCard parecido com Macintosh.

## **Sistema de biblioteca de fotografias**

Este sistema deve registrar e recuperar informações sobre fotografias em uma biblioteca. Deve apoiar classificação de palavras-chave e recuperar solicitações de tipos diferentes de fotografia. Novamente, o sistema deve ser projetado para o uso de pessoas que não são familiarizados com conceitos de informática. Ele deve ser desenvolvido em um sistema de computador pessoal, e você poderia usar alguns sistemas de Photo-CD de domínio público para testar o sistema. Dependendo da plataforma de implementação, você pode desejar incluir vários recursos de manipulação de imagens no sistema.

## **Sistemas de simulação**

### **Simulação de gerenciamento de memória**

Um sistema de memória virtual pode ser baseado em páginas de tamanho fixo que são movidos entre a RAM e o disco. O objetivo desse sistema é construir um simulador de gerenciamento de memória para testar os efeitos das estratégias da substituição de páginas diferentes, dos tamanhos de páginas, etc. O sistema deve possuir recursos para apresentar a memória não usada e ilustrar o tráfego de páginas para, e do disco. Ele deve ser dirigido por comandos simples que solicitam e liberam páginas em um sistema particular. O sistema pode ter associada a ele, uma interface simples, textual ou gráfica.

### **Sistema de entrega de remédios**

Esse projeto simples simula um sistema de entrega de remédios conforme discutido no capítulo 21. As partes diferentes do sistema devem ser simuladas por

processos separados, e deve ser possível carregar um software de controle diferente em cada componente do sistema para testar sua operação.

## **Componentes de alarme de sistema**

Esse sistema deve prover simulações de uma série de tipos diferentes de sensores e atuadores, tais como detectores de fumaça, sensores de movimento, sirenes, *sprinklers*, etc. Você pode obter informações sobre o desempenho desses dispositivos a partir dos manuais do fabricante, ou simplesmente constituir cenários confiáveis para uso na simulação.

## **Sistemas *WEB***

### **Verificação de *Web site* e reparo**

Muitos Web sites interromperam links, e os arquivos de armazenamento associados contêm arquivos não usados e não desejados. O objetivo desse projeto é desenvolver um sistema que verifique um Web site em busca de erros desse tipo, e que forneça uma interface de controle simples para que os usuários reparem esses erros. O sistema deve ‘aprender’ quando o usuário repara um erro, de tal modo que, se, em qualquer ocasião, um link for substituído por outro, o sistema proponha automaticamente essa mudança caso a ocorrência de erros posteriores envolvendo os mesmos arquivos seja detectada.

### **Referência cruzada de Web site**

O objetivo deste projeto é desenvolver um sistema de referência cruzada que possa atravessar um Web site e produzir uma tabela mostrando os links que são usados em cada página, e mostrar cada página Web de um site que faz referência a outra em particular. A referência cruzada externa limitada também é possível quando você segue um link de ida e, se esse link possui um link de volta, pode-se então fazer a referência cruzada desse site.

### ***Layout* de página Web de programas em Java**

Neste projeto, seu objetivo é levar um programa escrito em Java como um texto e convertê-lo em um conjunto de páginas Web onde o Java é apresentado como HTML. Você precisa ser capaz de descobrir todos os objetos referenciados no programa (normalmente em arquivos separados) e apresentar esses objetos em páginas separadas. Você pode criar diversos índices diferentes dentro das páginas do programa em Java e seu sistema deve lidar com o *layout* dos programas em Java de até 10.000 linhas de código.

Para fazer esse projeto, você pode ter de classificar o código Java para descobrir os objetos usados. Existe uma série de componentes em Java para classificação que estão disponíveis para download gratuito.