

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/28229130>

O ensino da disciplina linguagem de programação em escolas técnicas

Article · January 2008

Source: OAI

CITATIONS

8

READS

548

2 authors, including:



Fabio Wellington Orlando Da Silva

Centro Federal de Educação Tecnológica de Minas Gerais

39 PUBLICATIONS 171 CITATIONS

SEE PROFILE

Artigo Científico

O ensino da disciplina linguagem de programação em escolas técnicas

Teaching aspects of computer programming languages in vocational schools

Cléber R. Fontes e Fabio W.O. da Silva

Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG), Belo Horizonte,
Minas Gerais, Brasil

Resumo

Este trabalho apresenta os resultados de um estudo de campo sobre textos usados no ensino da disciplina linguagens de programação, em algumas escolas técnicas. Foi investigado o tipo de material didático adotado nessa disciplina, os critérios usados na escolha, as principais características do material, e a viabilidade de uma proposta alternativa para utilização desse material, inspirada nos estudos de David Ausubel sobre psicologia educacional e aprendizagem significativa. A pesquisa de campo foi desenvolvida em uma escola técnica de ensino médio, ao longo de dois meses, obtendo-se um desempenho superior dos alunos submetidos à estratégia alternativa. Espera-se que os resultados contribuam para o planejamento de aulas de linguagens de programação, bem como para o planejamento curricular de escolas de ensino médio voltadas à formação de profissionais de computação. © Cien. Cogn. 2008; Vol. 13 (2): 84-98.

Palavras-Chave: aprendizagem significativa; linguagem de programação; metodologia de ensino; Ausubel; livros didáticos.

Abstract

This paper presents the results of a field study on textbooks adopted in the discipline Computer Programming Languages in some technological vocational schools. This investigation focused on the type of didactic material adopted, the criteria underlying the choice, the main characteristics of the material, and the viability of an alternative proposal for usage of this material inspired in the studies of David Ausubel on Educational Psychology and Meaningful Learning. The empirical research was developed in a Technological Vocational School, during two months, and the students submitted to the alternative strategy have achieved better performance as compared to students in conventional learning contexts. It is expected that these results may contribute for computer programming languages lesson planning as well as for the curriculum planning in Vocational Schools aiming at the education of professionals in the computing technology field. © Cien. Cogn. 2008; Vol. 13 (2): 84-98.

Keywords: meaningful learning; programming language; teaching methodology; Ausubel; textbooks.

1. Introdução

As linguagens de programação de computadores evoluíram muito nos últimos cinco anos. Atualmente elas oferecem uma vasta gama de recursos, que permitem desde a elaboração de simples programas para cálculos matemáticos até a construção de sistemas de computação com engenharia complexa, a integração de dados alfanuméricos, geográficos, fotográficos ou sonoros, que podem ser lidos e armazenados em bases de dados localizadas em qualquer parte do planeta.

Os textos sobre informática, por sua vez, em geral não acompanharam essa evolução, e o ensino da estrutura e dos comandos das diversas linguagens de programação é ministrado de forma descontextualizada. Esses recursos são expostos aos estudantes sem aplicação prática, sem um objetivo específico ou um desafio ao qual devam responder. Aprende-se a linguagem por ela mesma, e, em diversos aspectos, o texto proposto para uso didático se confunde com o manual do fabricante, ou seja, uma lista dos comandos disponíveis.

De acordo com nossa experiência educacional, essa metodologia tem-se revelado monótona para os estudantes e ineficaz para o aprendizado. É possível que essa característica seja inerente ao ensino de linguagens de programação, ligada ao seu conteúdo árido, determinado por uma lógica inexorável e artificial. Entretanto, pode ser que os motivos estejam ligados a uma metodologia de ensino inadequada.

Diante deste problema inicial, nossos questionamentos são: Qual é o tipo de material didático utilizado em disciplinas de linguagens de programação em escolas técnicas de informática? Qual é o motivo dessa preferência? Quais são as principais características desse material? Esse material é adequado? Se não for adequado, o que pode ser feito para melhorá-lo?

Pretendemos ainda contribuir para a elucidação dessas questões, comparando o método de ensino usualmente adotado com uma estratégia alternativa inspirada nos estudos de David Ausubel (1980) sobre psicologia educacional e ensino significativo. Nossa hipótese de trabalho é que, ao tornar o ensino mais significativo para os estudantes, haja uma conseqüente elevação no interesse e no aproveitamento da disciplina.

A pesquisa sobre o material didático foi realizada em diversas escolas técnicas de Belo Horizonte, e a pesquisa empírica foi desenvolvida durante dois meses, no segundo semestre letivo de 2007, em uma escola técnica. Os resultados parecem corroborar nossa hipótese, esperando-se que contribuam para o planejamento de aulas de linguagens de programação, para o planejamento curricular de escolas de ensino médio para formação de profissionais de informática, bem como para o aprimoramento de livros didáticos nessa área.

2. O ensino de linguagens de programação

O ensino de linguagens de programação de computadores, nas escolas e faculdades voltadas para a formação de profissionais da área de informática, vem obedecendo à mesma linha metodológica do início da década de 1990. Em geral, os livros técnicos e apostilas contêm modelos prontos de programas muito elementares, aos quais se seguem algumas séries de exercícios, que não passam de variações dos modelos.

Um exemplo muito comum é o “*Hello World*”, que habitualmente aparece nos livros como um dos primeiros programas a ser desenvolvidos. Sua única função é escrever a mensagem “*Hello World*” na tela do computador. Os alunos são exaustivamente orientados a elaborar esse programa em diversas linguagens, mas isso não significa que eles estejam entendendo a função do programa, que é escrever qualquer mensagem ou frase na tela. Por apresentar palavras em idioma estrangeiro – muitos autores nem se preocupam em traduzir as mensagens – alguns alunos se confundem na identificação da mensagem e na identificação

dos códigos de programação, já que o contexto é o de um ambiente onde se inicia a aprendizagem de uma linguagem de programação.

Essa metodologia de ensino talvez fosse satisfatória quando os programas de computadores apresentavam uma *interface* muito simples com o usuário, por meio de telas com informações apenas sob a forma de textos, e os objetivos em termos de programação eram muito limitados. Isso, contudo, mudou.

O aluno que estiver estudando sob essa metodologia não terá oportunidades de aprendizagem para desenvolver sua capacidade de compreensão e abstração do pensamento lógico-computacional, pois a literatura técnica disponível emprega exemplos prontos, distantes de sua realidade. Ao ingressar no mercado de trabalho, em geral como estagiário de programação, esse aluno terá dificuldades em relacionar o conhecimento recebido com as necessidades reais da empresa como, por exemplo, criar uma mensagem de aviso de que um sistema está em manutenção e não pode ser utilizado no momento.

Utilizando um mecanismo de buscas da Internet com os termos “Hello World”, rapidamente encontramos muitos resultados e comentários em fóruns e discussões de usuários, comprovando a confusão que tal programa causa. Alguns autores já não trazem mais exemplos com “*Hello, World*”, mudando o formato das mensagens ao leitor. Em livros traduzidos, porém, essas mensagens habitualmente permanecem na língua original, preservando o problema.

Outro exemplo muito freqüente são os programas que resolvem problemas matemáticos a partir das informações digitadas pelo usuário, como o cálculo de somas e de médias aritméticas, cálculos de fatoriais e da sequência de Fibonacci. Esse tipo se mostra ineficaz ao aluno por não conseguir atrair sua atenção para um universo compreensível. Funciona como um livro de receitas, em que os passos são seguidos em uma ordem pré-determinada, com os ingredientes e medidas já informados, e o resultado é um produto que, muitas vezes, o aluno não sabe se está correto ou não.

Diante de uma situação problema, o aluno necessita utilizar toda sua estrutura cognitiva para descrever para o computador os passos para a resolução do problema, utilizando uma linguagem de programação. O computador executará a descrição da resolução do problema e fornecerá uma “resposta” somente daquilo que lhe foi solicitado. O aluno deverá, então, refletir sobre o que foi produzido pelo computador. Caso os resultados não correspondam ao desejado, o aluno terá que buscar novas informações para incorporá-las ao programa e repetir a operação (Valente, 1998).

Apesar do avanço da informática e do uso dos recursos computacionais cada vez mais difundido no cotidiano de todos, com o uso de computadores pessoais, acesso à Internet, troca de mensagens “instantâneas” ou por e-mail, o conhecimento técnico da população, de um modo geral, ainda é muito pequeno e limitado a poucas aplicações, como editores de texto, planilhas, navegadores de páginas de Internet e gerenciadores de música. A experiência de utilização de programas de computadores voltados ao gerenciamento de bancos de dados e extração de informações não é percebida ou assimilada diretamente pelo público. Como, então, ensinar a um aluno sobre o desenvolvimento de um sistema de computação, implementado por uma linguagem de programação, se ele nunca utilizou um sistema desse tipo e não tem formado o seu conceito sobre o assunto?

Assim, ao ingressar no curso, o aluno tem pouco conhecimento prévio sobre o tema, seus conceitos estão desconectados e, infelizmente, os materiais didáticos adotados fazem uma exposição sequencial dos conteúdos, não realizam a interligação entre os conceitos, nem se preocupam em informar ao estudante para que serve o que ele está aprendendo.

Segundo Braathen (2003:37) as pessoas constroem e reconstróem o seu conhecimento ao longo da vida, e “os estudantes que não aprendem satisfatoriamente o que tentamos ensinar, muito provavelmente não o fazem por não terem o conhecimento prévio necessário para uma aprendizagem significativa”. Essa é uma referência à teoria elaborada por David

Ausubel, centrada no contexto educativo, e que se ocupa especificamente dos processos de ensino/aprendizagem dos conceitos científicos pelo aluno a partir dos conceitos previamente formados em sua vida cotidiana. Ele abre sua Psicologia Educacional (Ausubel, 1980: viii) afirmando a importância dos conhecimentos prévios: “Se eu tivesse que reduzir toda a psicologia educacional a um único princípio, diria isto: o fator isolado mais importante que influencia a aprendizagem é aquilo que o aprendiz já conhece. Descubra o que ele sabe e baseie nisso os seus ensinamentos”.

De acordo com essa teoria, toda aprendizagem em sala de aula poderia ser analisada em um gráfico contínuo de duas dimensões ortogonais: o eixo horizontal, aprendizagem por recepção-descoberta; e o eixo vertical, aprendizagem por repetição-significativa. Na aprendizagem por recepção, o conteúdo é apresentado ao aluno em sua forma final, e espera-se que o aluno relacione esse conteúdo aos aspectos relevantes de sua estrutura cognitiva. Na aprendizagem por descoberta, o aluno deve descobrir independentemente o conteúdo a ser aprendido. Na aprendizagem por repetição (ou automática), a tarefa consiste em realizar associações puramente arbitrárias, ou falta ao aluno o conhecimento prévio necessário para torná-la significativa. No extremo oposto, estaria a aprendizagem significativa, em que uma informação nova pode se relacionar de forma não arbitrária e substancial (não literal) ao que o aluno já sabe, e quando o aluno adota um estratégia compatível com esse objetivo. Isto significa que as novas idéias expressas simbolicamente são relacionadas a algum aspecto relevante de sua estrutura cognitiva, seja uma imagem, um conceito etc.

A aprendizagem significativa requer simultaneamente que:

- 1) o material a ser aprendido seja significativo, ou seja, incorporável à estrutura cognitiva do aluno por meio de uma relação não arbitrária e não literal;
- 2) o aluno manifeste uma disposição para esse tipo de aprendizagem.

Acreditamos que, no ensino de informática, essas condições estejam mais distantes de serem preenchidas do que em outras áreas. Um aluno de engenharia, quando se depara com o primeiro projeto a construir, tem que formar seus conceitos e significados sobre técnicas de planejamento, cálculos, desenhos e construção, mas já passou pela experiência de ter entrado, morado e usufruído de uma residência e possui os principais conceitos sobre o objeto real: sabe o que são portas, janelas, pias, lâmpadas e paredes. Um aluno de medicina, ao entrar para a faculdade, muito provavelmente, já teve a experiência de ter entrado em consultórios médicos para se consultar e sabe que é preciso haver uma consulta, dessa consulta sai um diagnóstico, que pode requisitar um ou mais exames, que podem resultar no uso de medicamentos e em consultas de retorno. Um aluno de informática, ao entrar para o curso técnico ou superior, contudo, não tem noção do que seja um sistema, uma linguagem de programação, ou um banco de dados, a não ser que já trabalhe na área de informática e tenha optado pelo curso para a obtenção de um certificado.

3. Metodologia

A pesquisa de campo foi realizada em escolas de ensino técnico de informática de Belo Horizonte e compreendeu as seguintes etapas:

- a) levantamento de dados em escolas técnicas de Belo Horizonte, com ensino técnico de informática, sobre os livros didáticos adotados em disciplinas de linguagem de programação;
- b) entrevistas com um coordenador de ensino e com um professor de disciplina de linguagem de programação das escolas do item (a);
- c) Experimento e testes de conhecimento com alunos de duas turmas de linguagem de programação.

3.1. Levantamento de dados nas escolas com ensino técnico de informática

O levantamento sobre os textos didáticos adotados em disciplinas de linguagem de programação foi realizado em seis escolas de nível médio e pós-médio, com cursos técnicos de informática, de Belo Horizonte. Essa etapa consistiu de uma consulta enviada à coordenação de ensino das escolas, pelos e-mails institucionais disponibilizados em seus *sites*, solicitando que indicassem a bibliografia adotada nas disciplinas de Linguagem de Programação do curso e, se fossem livros, indicar o título, o autor e a editora. Posteriormente, foi realizado um contato telefônico com a secretaria das escolas para complementar as informações e agendar uma entrevista.

3.2. Entrevistas

As entrevistas tiveram o objetivo de refinar o conhecimento sobre a concepção que os coordenadores e docentes das instituições têm a respeito da bibliografia adotada nas disciplinas de linguagens de programação, e os critérios que norteiam essa escolha. As entrevistas foram realizadas de forma estruturada, tendo como interlocutores um coordenador de ensino e um professor com experiência no ensino de linguagens de programação, contemplando os seguintes temas:

1. Qual é a bibliografia adotada nas disciplinas de Linguagem de Programação do Curso (indicar o nome do curso)? (Se forem livros, favor indicar nome, autor e editora.)
2. Quais são os motivos da adoção desse tipo de material (próprio ou livros didáticos)?
3. Quais são as vantagens e desvantagens de utilização desse material (próprio ou livros didáticos)?
4. A escola utiliza o mesmo tipo de material (próprio ou livro didático) para as disciplinas da Base Nacional Comum?
5. Quais são as vantagens e desvantagens de utilizar esse tipo de material (próprio ou livro didático) nas disciplinas da Base Nacional Comum?
6. Qual(is) disciplina(s) de Linguagem de Programação você leciona ou já lecionou?
7. A escolha do material didático é uma definição da instituição ou do professor?

3.3. Experimento e teste de conhecimento com os alunos

A terceira etapa da pesquisa foi realizada em duas turmas de ensino técnico de informática de um colégio de Belo Horizonte, que nesse artigo chamaremos de Turma 1 e Turma 2, durante as aulas da disciplina de Linguagem de Programação (Delphi). Os alunos estavam tecnicamente aptos a desenvolver sistemas, mas nunca tinham tido a experiência de utilizar um sistema como um usuário final. Essa é uma situação comum entre alunos do ensino técnico de informática.

O objetivo da etapa é comparar o desempenho da Turma 1, usando uma estratégia alternativa, denominado Sistema Modelo, com o da Turma 2, com aula convencional, empregando somente o material adotado pela escola. As duas turmas apresentam características muito semelhantes, seguem o mesmo calendário escolar e possuem número aproximado de alunos, sendo a primeira com 25 alunos e a segunda com 23.

O Sistema Modelo (Apêndice B) é o protótipo de um sistema de cadastramento de clientes, um exemplo muito utilizado em disciplinas de linguagens de programação, contendo uma base de dados com várias tabelas, e disponível via Internet no endereço <http://www.cleberfontes.com/projetos/prj001>, com controle de acesso de usuários por meio de senhas. O sistema é montado em uma linguagem voltada para a Web (PHP), e busca expor aos alunos os mesmos conceitos desenvolvidos na disciplina (porém sem a preocupação de estrutura e sintaxe de programação): formulários de entrada de dados, bancos de dados, registros, tabelas, utilização de tabelas para campos de escolha (combobox). Aqui, o importante é levar o aluno a compreender os conceitos e a utilização dos sistemas antes de construí-los, independentemente da ferramenta utilizada.

As duas turmas foram avaliadas, por meio de testes, antes (Apêndice A) e após (Apêndice C) freqüentar as aulas de linguagem de programação. Por serem turmas muito semelhantes, a divisão das tarefas entre elas foi escolhida por sorteio, de acordo com a descrição a seguir.

Tarefas da Turma 1:

- Responder ao questionário de conhecimento prévio.
- Executar as tarefas contidas em uma folha recebida, que consistiam na alimentação e operação do Sistema Modelo. Isso inverte o papel do aluno e o faz desempenhar a função de usuário. O Sistema Modelo fica disponível na Internet e pode ser acessado a qualquer hora pelos alunos, mesmo após o dia marcado para a execução da tarefa.
- Assistir às aulas previstas para o conteúdo de programação de banco de dados uma semana após a experiência.
- Responder ao questionário sobre o aprendizado.

Tarefas da Turma 2:

- Responder ao questionário de conhecimento prévio.
- Assistir às aulas previstas inicialmente sem alteração.
- Responder ao questionário sobre o aprendizado.

Como visto acima, ambas as turmas responderam aos mesmos questionários, prévio e final, permitindo a comparação entre o nível de acerto das respostas entre a turma que acessou e a turma que não acessou o sistema.

Para que os alunos possam se habituar com um ambiente mais próximo de um sistema corporativo, foram criados dois usuários para acesso ao sistema, com nível de privilégios diferentes, através da implementação de uma segurança no sistema que não permite que os dados digitados por um usuário sejam modificados pelo outro.

Além disso, na folha de tarefas distribuída aos alunos, foi dada a opção de criar um usuário e senha individuais para cada aluno que assim o desejasse.

4. Resultados e discussão

4.1. Levantamento nas escolas

O levantamento de dados nas escolas sobre os livros didáticos utilizados em disciplinas de linguagem de programação foi a etapa mais difícil de ser realizada. O questionário foi enviado por correio eletrônico a seis escolas, mas somente duas responderam. Diante desse baixo retorno, foi realizada uma nova tentativa via contato telefônico e mais três escolas enviaram resposta.

Assim, o resultado do levantamento foi: 5 escolas responderam; 1 escola não respondeu; entre as cinco escolas que responderam, todas adotam apostilas elaboradas pelos professores e não utilizam livros didáticos disponíveis no mercado.

As três escolas que responderam via contato telefônico informaram que utilizam texto próprio e “os professores disponibilizam o material para os alunos, conforme sua programação anual”, uma vez que não há uma apostila padronizada pela escola. Em uma das escolas, cada professor pode elaborar sua apostila, o que faz com que turmas diferentes, com professores diferentes, tenham apostilas e conteúdos lecionados diferentes.

As duas escolas que responderam por correio eletrônico possuem apostila elaborada e padronizada pela escola, sendo a mesma apostila para todas as turmas, independentemente do professor que leciona a disciplina e da unidade em que a turma estuda (ambas têm mais de uma unidade do colégio, em bairros diferentes da cidade).

Percebeu-se também uma certa dificuldade para as escolas admitirem que não adotam livros nas disciplinas técnicas.

4.2. Entrevistas

Os entrevistados revelaram que suas escolas utilizam apostilas próprias, elaboradas pelo corpo docente das disciplinas. Os textos das apostilas são discutidos e aprovados pela coordenação de ensino e coordenação técnica de informática, e utilizadas de forma padronizada em todas as turmas da escola.

O motivo mais importante para adoção de apostilas, e não de livros, é o alto custo de aquisição dos livros, o que poderia tornar o curso oneroso para o aluno. É possível elaborar e entregar ao aluno apostilas que chegam a custar 10% (dez por cento) do preço de um livro equivalente. Como exemplo foi citado o alto preço dos livros “Dominando o Delphi 7 – A Bíblia” (Cantu, 2003) e “Java, como programar” (Deitel e Deitel, 2005), que são referências atualmente no ensino de linguagens de programação.

Além disso, os livros de linguagem de programação atualmente disponíveis no mercado possuem um conteúdo muito extenso, que não coincide com o programa da escola, pois os livros têm um conteúdo muito maior que o conteúdo programático previsto no calendário escolar.

O uso de apostila faz com que o material didático possa ser direcionado ao que está sendo visto pelo aluno, garantindo um material didático adequado ao curso. Na verdade, os livros disponíveis no mercado não são livros didáticos de linguagem de programação, mas guias completos sobre softwares específicos da linguagem de programação. Não é como no caso da Matemática, por exemplo, que tem livros voltados para o conteúdo lecionado em cada série do ensino fundamental e médio.

As escolas prevêm ainda que as apostilas sejam atualizadas anualmente pelos professores e permitem que elas sejam elaboradas em formato de guias de aulas.

Para as disciplinas da Base Nacional Comum (BNC), as escolas utilizavam o material didático disponível no mercado.

A utilização de apostilas, em vez de livros editados, também apresenta algumas desvantagens.

O primeiro problema é que a atualização anual nem sempre ocorre como a escola gostaria. Por mais aberta que seja a participação de todos os professores na tarefa de atualização das apostilas, na prática ela fica limitada a um grupo restrito de participantes, o que torna o campo de visão limitado ao desse grupo. Isso acarreta um segundo problema, pois, apesar de ser um dever do professor procurar extrapolar o material da apostila, como ele recebe um texto que contém uma visão muito restrita à do grupo que o elaborou, essa função fica prejudicada.

Sobre os livros de informática disponíveis no mercado, os entrevistados os consideram de qualidade muito alta, com uma abordagem profunda dos conteúdos, o que os torna guias de referência aos profissionais de informática. São livros que se propõem a solucionar problemas ou dúvidas através de uma leitura que não precisa ser seqüencial: pode-se buscar o assunto desejado e conhecer seu funcionamento. São livros muito direcionados a quem já conhece uma linguagem de programação e precisa conhecer os comandos e estruturas de outra linguagem de programação. O mercado de trabalho exige livros dessa natureza, mas escola precisa de material mais didático, mais voltado ao processo de ensino-aprendizagem do conceito de programação. Por esse motivo, as escolas optam por utilizar apostilas adequadas a esse objetivo, e indicam aos alunos que usem os livros para referência externa.

As apostilas elaboradas pelas escolas seriam baseadas nas referências bibliográficas dos livros disponíveis, sendo que cada apostila busca o conteúdo mais próximo à proposta apresentada pelo colégio.

Não foi possível fazer uma avaliação explícita sobre a contextualização das apostilas, pois não estão disponíveis ao público externo, sendo considerado um material de propriedade intelectual das escolas, nem é possível avaliar os livros nas quais elas buscam suas referências, pois esses livros não foram citados nas consultas feitas às escolas.

De toda forma, avaliamos que não exista, por estes motivos, uma padronização na formação de técnicos de informática quanto ao conteúdo mínimo que um técnico deva conhecer para ingressar no mercado de trabalho. A disciplina de Linguagem de Programação não é tratada como uma disciplina como Português ou Matemática, em que o conteúdo a ser visto por um aluno do ensino médio em um colégio é muito próximo ao dos demais alunos da mesma série de outros colégios.

4.3. Experimento e teste de conhecimento com os alunos

4.3.1. Questionário de conhecimento prévio

As percepções dos próprios alunos sobre seu conhecimento de conceitos de informática, para as turmas 1 e 2, obtidas por meio do questionário sobre conhecimento prévio, estão sintetizadas no Quadro 1.

4.3.2. Questionário sobre o aprendizado

A avaliação final compreendeu questões de percepção dos próprios alunos (parte 1) e questões de conhecimento (parte 2). Alguns desses conceitos apareciam no questionário de conhecimento prévio, e o objetivo foi observar como ocorreu, na percepção dos alunos, a evolução do seu próprio conhecimento. Os resultados da turmas 1 estão sintetizados na figura 2 e os resultados da turma 2 estão sintetizados na figura 3.

A segunda parte do questionário foi elaborada para servir como um teste de conhecimentos. O percentual de acertos das duas turmas às 7 questões propostas está representado na figura 4. Observamos um aproveitamento superior da Turma 1, que utilizou o Sistema Modelo como forma de experimentação, em todos os itens do questionário. Ressaltamos particularmente a diferença na questão 7, que trata do tema central apresentado pelo Sistema Modelo. A média da turma 1 foi superior a 90%, enquanto a da turma 2 ficou em 71%.

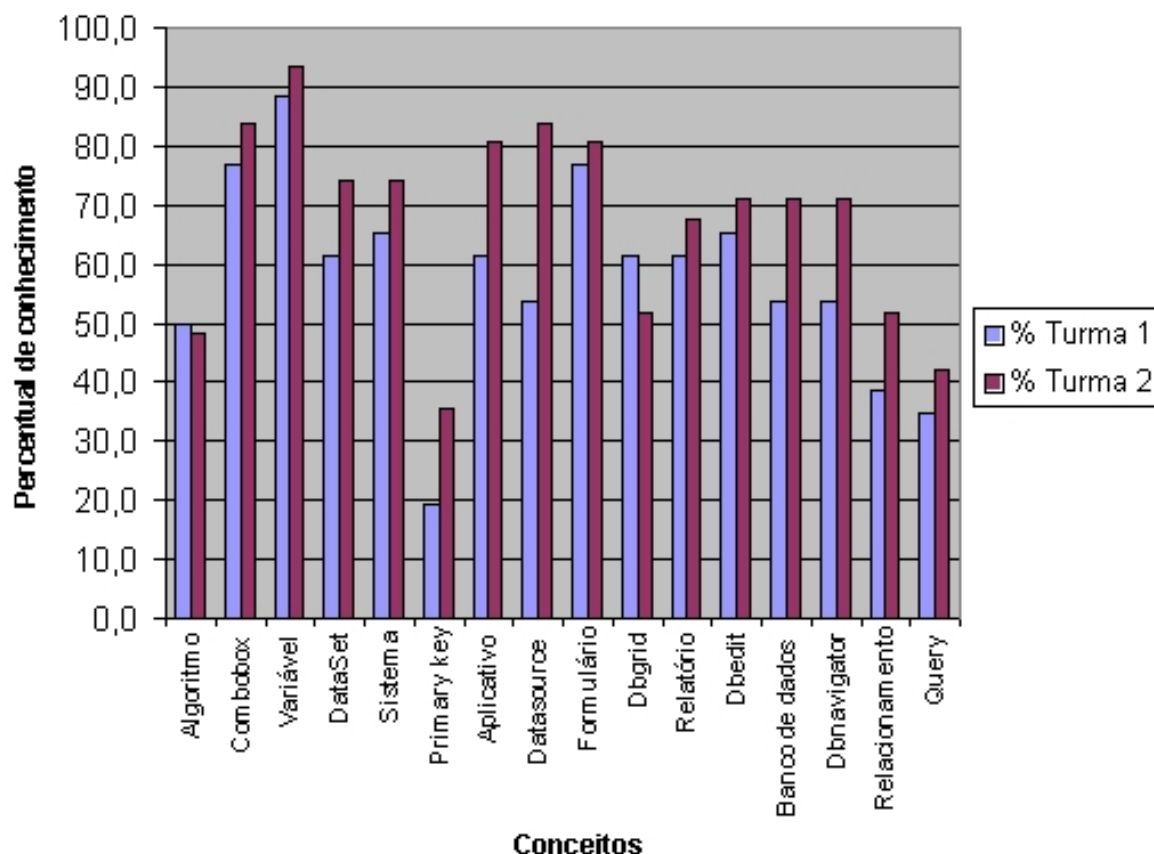


Figura 1 – Respostas ao questionário prévio das turmas 1 e 2. Comparativo de percepção de conceitos dos alunos.

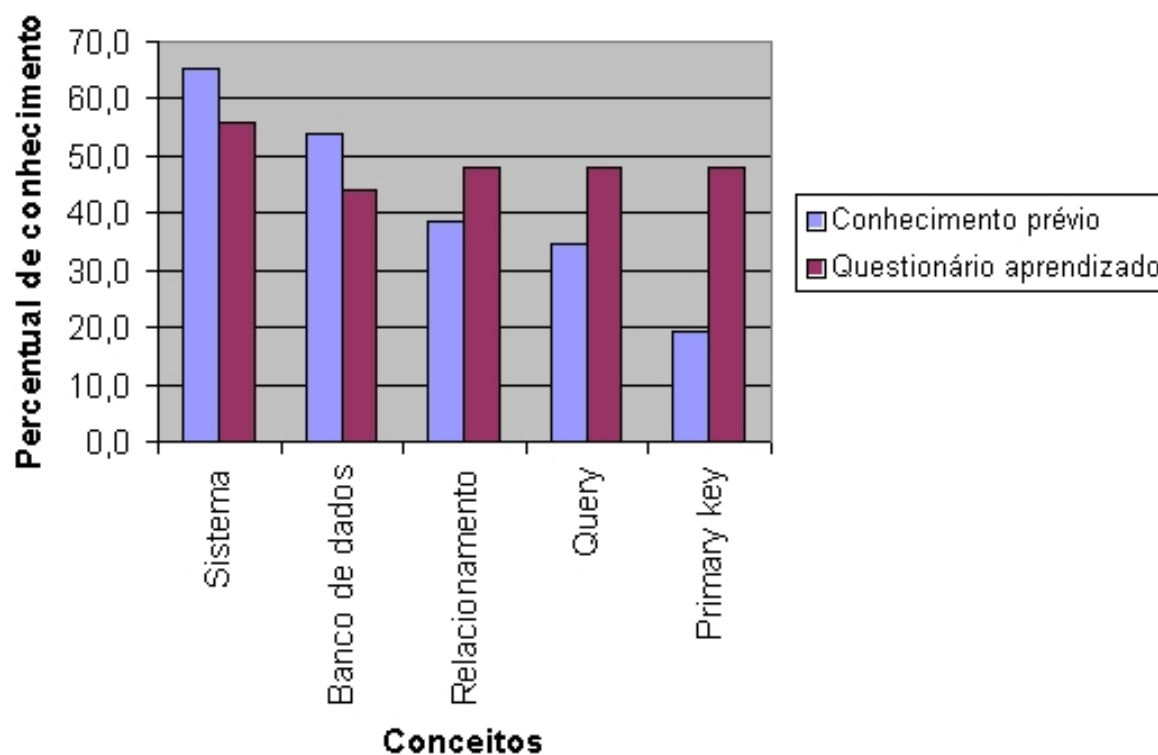


Figura 2 – Comparativo de percepção de conhecimento e questionário de aprendizado da Turma 1.

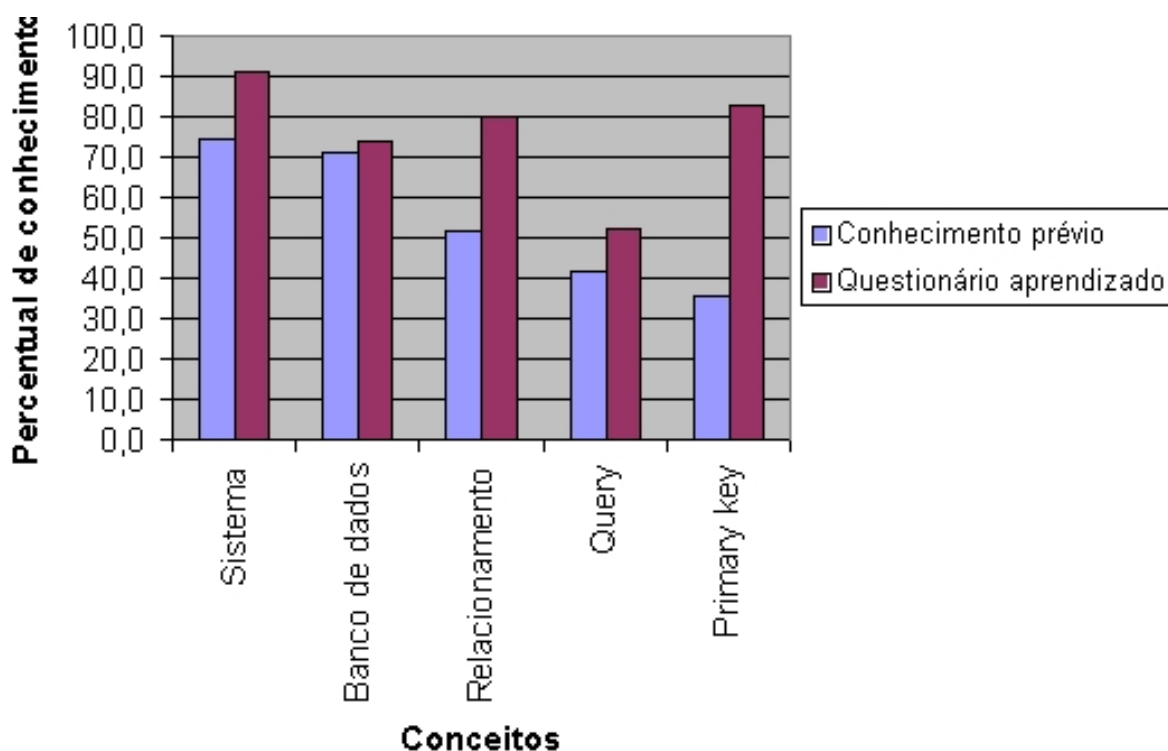


Figura 3 – Comparativo de percepção de conhecimento e questionário de aprendizado da Turma 2.

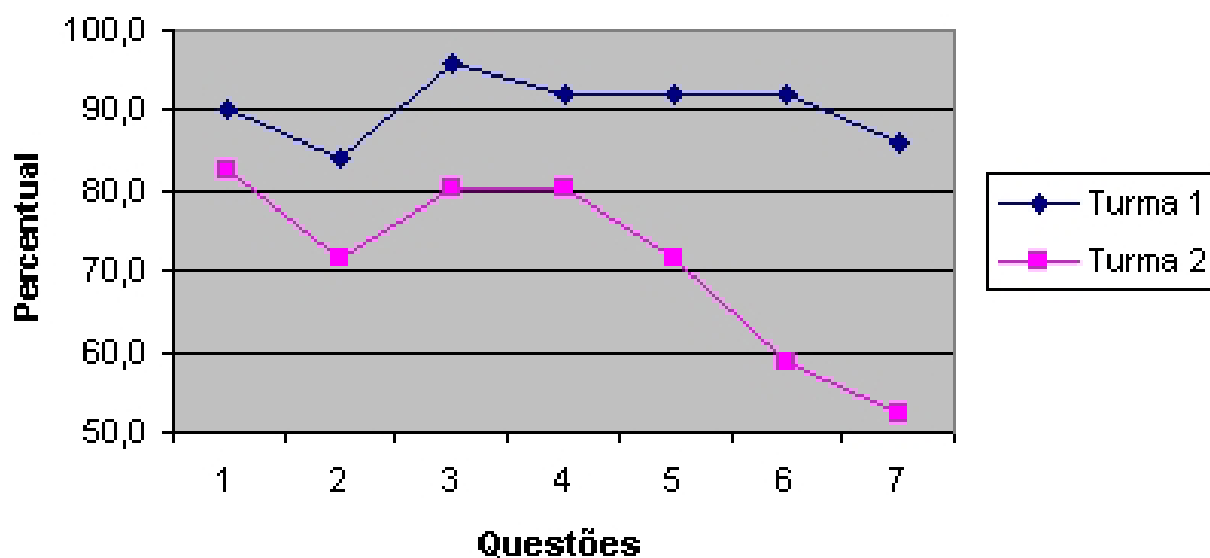


Figura 4 – Desempenho das turmas – Questão 2, Questionário 2.

Analisando os gráficos dos questionários de conhecimento prévio, percebe-se que a Turma 2 se auto-avalia com notas um pouco mais altas do que as da Turma 1. Na percepção final, após freqüentar a disciplina, a auto-avaliação da turma 2, porém, é muito superior à da turma 1. No entanto, o questionário sobre o aprendizado mostrou um rendimento bem menor da Turma 2. Talvez isso se deva a um maior nível de exigência da Turma 1, por conhecer melhor as dificuldades da área.

Com o objetivo de descartar uma possível coincidência de resultados, o aproveitamento dos alunos nas disciplinas de linguagem de programação de 2007 foi comparado ao de anos anteriores, desde 2002, em que as aulas foram ministradas no estilo convencional da escola. Mais uma vez, verifica-se um aumento significativo na média de aproveitamento da Turma 1, em relação à média de aproveitamento de turmas de anos anteriores.

Turmas	Aproveitamento (%)
Turmas 2002	77,80
Turmas 2003	72,77
Turmas 2004	60,90
Turmas 2005	72,70
Turmas 2006	74,40
Turma 2007 (sem experimento)	71,10
Turma 2007 (com experimento)	90,30

Tabela 1 – Série histórica de aproveitamento entre 2002 e 2007.

5. Conclusão

Concluimos que as escolas técnicas pesquisadas utilizam, como material nas disciplinas de linguagens de programação, manuais e apostilas de elaboração própria, buscando uma adequação desse material ao seu planejamento curricular. O principal motivo alegado para essa preferência é a indisponibilidade no mercado de livros voltados à educação técnica. A opção pelas apostilas tem como consequência imediata a falta de parametrização para a formação técnica de informática de alunos de instituições diferentes.

Uma parte significativa do material não é constituída por uma produção inédita das instituições, mas de exemplos pesquisados em fontes diversas, sob contextos diferentes e de autores variados. Possivelmente, em virtude desse fato, o aluno pode vir a ter uma experiência descontextualizada, gerando um aprendizado monótono e de mais difícil aproveitamento.

Acreditamos que, se houvesse livros disponíveis com objetivos didáticos, voltados à experimentação real por parte dos alunos, as escolas os adotariam, pois, nas escolas pesquisadas, as disciplinas onde há esse material, como Português, Matemática e Física, os livros são adotados e não se criam apostilas.

Concluimos também que o conhecimento prévio real dos alunos admitidos no ensino técnico é muito menor que o conhecimento que eles consideram que possuem. É como se o aluno considerasse que já tem bastante conhecimento sobre informática e procurasse o curso técnico apenas para aprimoramento e durante o curso descobrisse que o “universo” da informática é diferente de sua concepção inicial.

Esta pesquisa procurou analisar o aproveitamento dos alunos através da inserção de significado e contextualização do ensino de informática, comparando o desempenho com outra turma que não teve a mesma inserção e teve seu conteúdo aplicado conforme métodos já utilizados pela instituição. Observou-se um desempenho superior na turma que passou pela experiência da utilização do Sistema em teste.


Dessa forma, acreditamos que seja possível melhorar o aproveitamento dos alunos por meio da construção de exercícios de laboratórios de informática que primeiro contextualizem uma situação do uso da informática – visto que esse conhecimento não é prévio – para depois

iniciar as situações de construção de aplicativos, pois os laboratórios como meios de ensino implicam mais do que o contato direto e a observação, envolvendo também a experiência da descoberta e a relação com aspectos como formação e testes de hipóteses, controle e manipulação de variáveis e possibilidades de fazer inferências a partir de dados. Não tendo o professor um material adequado para o desempenho de suas atividades, uma alternativa para melhor aproveitar o material é a construção desses laboratórios de observação e experimentação, pois é muito mais difícil ensinar a um aluno a construção de um objeto que ele nunca tenha visto, nunca tenha usado e cujo primeiro contato seja com o objeto por ele construído.

Esta pesquisa indica que novas investigações devam ser feitas na área, para que novas formas de construção de conhecimentos, voltadas para disciplinas de linguagens de programação, possam ser descobertas, acompanhando o ritmo de evolução da indústria de informática.

6. Referências bibliográficas

- Ausubel, D.P. (1980). *Psicologia Educacional*. 2ª Ed. Rio de Janeiro: Interamericana, 625 pp.
- Braathén, C.O. (2003). Processo ensino aprendizagem em disciplinas básicas do terceiro grau. *Ed. Tecnol.*, 8 (1), 34-41. Retirado em 10/08/2007, de *World Wide Web*: <http://www2.cefetmg.br/dppg/revista/index.html>.
- Cantu, M. (2003). *Dominando o Delphi 7: a Bíblia*. São Paulo: Makron Books, 801 pp.
- Deitel, H.M. e Deitel, P.J. (2005). *Java: como programar*. Furmankiewicz, E. (Trad.). 6ª Ed. São Paulo: Pearson Prentice-Hall, 1152 pp.
- Valente, J.A. (1998). Análise dos diferentes tipos de softwares usados na educação. Em: *III Encontro Nacional do PROINFO – MEC*, Pirenópolis (GO).

 - **F.W.O. da Silva** é Graduado e Mestre em Física (Universidade Federal de Minas Gerais, UFMG), Doutor em Física (Universidade de Montpellier, França). Atua como Professor dos Cursos de Graduação, do Mestrado e da Especialização em Educação Tecnológica (CEFET-MG). Endereço para correspondência: Centro Federal de Educação Tecnológica de Minas Gerais, Av. Amazonas, 7675, Bairro Nova Gameleira, Belo Horizonte, MG 30.510-000. Telefone para contato: 55-31-3319-6796. *E-mail* para correspondência: fabow@des.cefetmg.br. **C.R. Fontes** é Graduado em processamento de dados (Curso Superior de Tecnologia em Processamento de Dados, Fundação Educacional Dom André Arcoverde), com Pós-graduação em Engenharia de Sistemas de Computação (Universidade Estácio de Sá, UNESA) e em Educação Tecnológica (CEFET-MG). Atua como Professor (Colégio Cotemig) e Supervisor de Informações Técnicas (URBEL). *E-mail* para correspondência: cleberfontes@gmail.com.

Apêndice A - Questionário 1: Conhecimento Prévio

Número do questionário:

O objetivo deste teste é verificar o nível de informação e conhecimento prévio sobre os tópicos relacionados com as disciplinas de **Linguagem de Programação**. Servirá como referência para os professores quanto à forma de abordar os conteúdos previstos. Não se preocupe se você não tiver conhecimento sobre qualquer item da lista. Não se trata de uma avaliação: o mais importante é expressar exatamente o seu nível de conhecimento e informação sobre os tópicos. **Não é necessário identificar-se.**

1. A lista abaixo mostra uma relação de conceitos e termos relacionados com o desenvolvimento de sistemas. Avalie seu grau de conhecimento / experiência em relação a cada um deles, conforme as seguintes opções:

- (1) – Desconheço; não sei do que se trata.
- (2) – Conheço superficialmente: já estudei ou li a respeito, mas não domino o assunto.
- (3) – Tenho algum conhecimento, mas pouca experiência.
- (4) – Tenho conhecimento e experiência a respeito.

Item	Minha avaliação (De 1 a 4)	Item	Minha avaliação (De 1 a 4)
▪ Algoritmo		▪ ComboBox	
▪ Variável		▪ Dataset	
▪ Sistema		▪ Primary key	
▪ Aplicativo		▪ Datasource	
▪ Formulário		▪ Dbgrid	
▪ Relatório		▪ Dbedit	
▪ Banco de dados		▪ Dbnavigator	
▪ Relacionamento		▪ Query	

2. A lista abaixo mostra tarefas / atividades relacionadas com o conhecimento e experiência em informática de modo geral. Informe seu nível de conhecimento, experiência ou desenvoltura para realiza-las, conforme o critério:

- (1) – Não sei como realizar (ou nunca realizei).
- (2) – Poderei realizar se for orientado (a).
- (3) – Realizo algumas vezes, porém com dificuldades ou limitações.
- (4) – Realizo sem dificuldades.

Tarefa / Atividade	Avaliação (1 a 4)
▪ Formatar textos com diferentes estruturas, incluindo gráficos, figuras, tabelas	
▪ Criar e formatar planilhas e gráficos no tipo Microsoft Excel	
▪ Criar e preparar apresentações no Microsoft PowerPoint	
▪ Criar e gerenciar pastas no disco do computador	
▪ Realizar buscas avançadas na Internet, a partir de palavras-chave	
▪ Receber/enviar mensagens e arquivos anexados através de e-mail	
▪ Estudar e pesquisar um assunto novo por conta própria	
▪ Analisar resultados de pesquisas, com dados e informações, e elaborar conclusões	
▪ Apresentar oralmente os resultados de um estudo, trabalho ou pesquisa	
▪ Fazer downloads de arquivos de documentos, vídeos ou músicas	
▪ Instalar e configurar aplicativos no sistema operacional	
▪ Construir aplicativos de cálculos	

3. Marque uma ou mais alternativas, se necessário, nas perguntas abaixo:

- 3.1. Onde você têm acesso ao computador?

Em casa	Em casa de amigos / Lan houses	No trabalho
---------	--------------------------------	-------------
- 3.2. Onde você têm acesso à Internet?

Em casa	Em casa de amigos / Lanhouses	No trabalho	Na escola
---------	-------------------------------	-------------	-----------
- 3.3. Para se comunicar via Internet você possui:

E-mail	Messenger (identificar)	Comunidades virtuais (Orkut, MySpace)	Nenhum
--------	--------------------------	---------------------------------------	--------

Apêndice B - Folha de Tarefas com o Sistema Modelo

Código do questionário: A001

O objetivo desta tarefa é utilizar e alimentar um sistema simples de Cadastro de Clientes, sem a preocupação de saber como foi desenvolvido.

Você será o usuário deste sistema e poderá cadastrar e alimentar o sistema normalmente, fazer alterações e propor sugestões sobre o seu funcionamento ou melhorias. Não se trata de uma avaliação.

1. Acesse o endereço <http://www.cleberfontes.com/projetos/prj001> (ou entre no endereço <http://www.Cleberfontes.com> e acesse o link [Projetos / 001-Bibliotecas]).
2. Entre no *site* utilizando o login “**admin**”, conforme as indicações da tela inicial.
 - 2.1. Cadastre um cliente, com seu nome e seus dados e salve as informações digitadas.
 - 2.2. Acrescente mais de um tipo de telefone para este cliente.
 - 2.3. Entre no cadastro de bibliotecas e acrescente um item a uma das bibliotecas (à sua escolha):
 - tipo de logradouro;
 - tipo de telefone;
 - estado civil;
 - unidade da federação,
 - área de interesse;
 - assuntos a informar.
 - 2.4. Volte ao cadastro de clientes e altere o cliente que você tinha cadastrado e observe os atributos que você alterou na biblioteca.
3. Saia do sistema e entre novamente como o usuário “**comum**”.
 - 3.1. Tente alterar o cliente que você cadastrou, anteriormente quando estava conectado como “**admin**”.
 - 3.2. Acesse o cadastro de biblioteca e tente fazer alguma alteração.
4. Conecte-se novamente como “**admin**” e tente fazer alterações nos clientes e biblioteca que você cadastrou com seus dados.
5. Caso tenha interesse em ter um login próprio no sistema, envie um e-mail para cleber@cleberfontes.com solicitando a criação de um login para o sistema, indicando o nome que deseja como *login* e a *senha* a ser utilizada neste projeto.

Apêndice C – Questionário 2: Aprendizado

Número do questionário:

O objetivo deste teste questionário é medir o conhecimento da turma em relação ao desenvolvimento de sistemas, com implementação de relacionamento de tabelas em bancos de dados relacionais, de acordo com os tópicos relacionados à disciplina de **Linguagem de Programação**. Não se trata de uma avaliação: o mais importante é expressar exatamente o seu nível de conhecimento e informação sobre os tópicos. **Não é necessário identificar-se.**

1. A lista abaixo mostra uma relação de conceitos e termos relacionados com o desenvolvimento de sistemas. Avalie seu grau de conhecimento / experiência em relação a cada um deles, conforme as seguintes opções:

- (1) – Desconheço; não sei do que se trata.
- (2) – Conheço superficialmente: já estudei ou li a respeito, mas não domino o assunto.
- (3) – Tenho algum conhecimento, mas pouca experiência.
- (4) – Tenho conhecimento e experiência a respeito.

Item	Minha avaliação (De 1 a 4)	Item	Minha avaliação (De 1 a 4)
▪ Sistema		▪ Table.MasterSource	
▪ Banco de dados		▪ Table.MasterFields	
▪ Relacionamento		▪ Table.IndexFieldsName	
▪ DataModule		▪ Query	
▪ Primary key		▪ Foreign key	

2. Uma Clínica médica gostaria de informatizar o controle das consultas e atendimentos aos seus pacientes. Hoje, cada médico controla seus pacientes através de fichas de papel que ficam em seu consultório. Quando um paciente é atendido por um outro médico, é criada uma nova ficha para este paciente. Nesta ficha, constam os seguintes dados:

1. Nome do paciente
2. Endereço completo (logradouro, numero, complemento, bairro, cidade, UF, CEP, etc)
3. Consultas do paciente

Para cada consulta são guardadas as seguintes informações:

- Data e hora da consulta
- Tipo da consulta (primeira consulta, retorno, acompanhamento, etc)
- Anotações do médico (campo livre)
- Exames solicitados
- Medicamentos receitados
- Previsão para a próxima consulta

Considerando que você desenvolveria este sistema para o médico, responda às seguintes perguntas:

- 2.1. De acordo com sua experiência no aprendizado de construção de sistemas, qual a vantagem de informatizar este sistema?
- 2.2. Seria possível a implementação de controle de acesso ao Sistema?
 - a) Sim
 - b) Não
 - c) Não é possível responder apenas com as definições acima
 Justifique:
- 2.3. Se um paciente já cadastrado marcasse uma consulta com outro médico, seria necessário que o paciente fosse cadastrado novamente?
 - a) Sim
 - b) Não
 - c) Não é possível responder apenas com as definições acima
 Justifique:
- 2.4. É possível configurar o sistema para que cada médico só possa alterar os dados da sua consulta?
 - a) Sim
 - b) Não
 - c) Não é possível responder apenas com as definições acima
 Justifique:
- 2.5. Como seria possível a um médico verificar se um paciente já é cadastrado ou não no Sistema?
- 2.6. Em relação ao banco de dados, como ficariam os dados de pacientes e consultas?
 - a) numa mesma tabela
 - b) em tabelas separadas
 Justifique:
- 2.7. Existe algum atributo que possa ser implementado através de uma biblioteca?
 - a) Sim
 - b) Não
 Justifique: