

METODOLOGIA DE PROJETO APLICADA AO ENSINO DE LINGUAGEM DE PROGRAMAÇÃO VISUAL: UM ESTUDO DE CASO

Tâmara Frazão Pinto

Departamento Acadêmico de Informática – CEFET-MA
Av. Getulio Vargas, 04 Monte Castelo São Luis – MA
E-mail: tamaralina@gmail.com

Omar Andres Carmona Cortes

Departamento Acadêmico de Informática – CEFET-MA
Av. Getulio Vargas, 04 Monte Castelo São Luis – MA
E-mail: omar@cefet-ma.br

RESUMO

O presente artigo sugere uma proposta metodológica para auxiliar o processo de ensino-aprendizagem da Informática em cursos profissionalizantes de nível técnico, objetivando beneficiar principalmente o aluno. Esta proposta metodológica fundamenta-se na Pedagogia de Projetos e considera como aspecto importante o desenvolvimento das habilidades e competências dos alunos, além de possuir a teoria aliada à prática e a aprendizagem dos conteúdos curriculares como colaboradores na melhoria do cenário educacional. A proposta metodológica é aplicada na disciplina Linguagens de Programação Visual dirigida para o curso técnico de Programação de Computadores do CEFET-MA com o objetivo de ser experimentada. Dois projetos são considerados no desenvolvimento da disciplina: um editor de textos e um sistema de banco de dados. Uma avaliação do curso, efetuada pela turma, também é apresentada.

PALAVRAS-CHAVE: Pedagogia de Projetos; Educação Profissional; Linguagens de Programação; Delphi

1. INTRODUÇÃO

Educar nos dias de hoje é educar na sociedade da informação. Fato este que tem levado muitos educadores a pensar, analisar e realmente agir de acordo com um sistema educativo baseado em novas concepções pedagógicas, epistemológicas, novas interações sociais e até mesmo novas situações afetivas e emocionais. Isto porque se sabe que atualmente no cenário educacional, não se encaixa mais velhos pensamentos, práticas conservadoras em sala de aula, métodos tradicionais e antigos hábitos didático-pedagógicos. Desse modo, o trabalho de Barbosa (2005) nos garante que um novo espaço epistemológico está em fase de gestão, com as seguintes características: construção, capacitação, aprendizagem, desenvolvimento de competências e habilidades, respeito ao ritmo individual, formação de comunidades de aprendizagem e redes de convivência, educação aberta e à distância e gestão do conhecimento.

Nesse contexto, percebe-se que desde os primórdios tempos de Platão nossa herança cultural é norteada pela íntima relação existente entre o pensar e o agir, entre a ciência e a concretização da mesma através da tecnologia, entre a observação e a aplicação. Seguindo estas idéias têm-se a afirmação de Sancho (2001) mostrando que esta fusão indissolúvel (e aparentemente indispensável) entre ciência e técnica abre um novo espaço de conhecimento, o da tecnologia como uma técnica que emprega conhecimentos científicos e que, por sua vez, fundamenta a ciência quando lhe dá uma aplicação prática. Assim, idéias que combinem teoria e prática fazem-se presentes ao longo da nossa trajetória cultural, principalmente levando-se em consideração as atuais reformas educacionais, pois hoje é muito difícil separarmos uma da outra como se fazia na pedagogia tradicional. É a prática que vem revelar se a elaboração de um pensamento é verdadeira ou falsa, pois o conhecimento só é aceito como tal se a teoria e a concretização desta estiverem em sintonia, ou seja, se o conhecimento for visto como uma verdade que foi pensada e corroborada pela sua aplicação.

A tarefa de ensinar linguagens de programação (LP) exige uma grande carga cognitiva (B. du Boulay, Sothcott, 1988), já que não se resume apenas a seguir os formalismos sintáticos da linguagem, mas também em aprender a decompor problemas e depois compô-los na linguagem considerada. Em outras palavras, deve-se necessariamente juntar os aspectos sintáticos e semânticos de um determinado problema.

Normalmente uma disciplina de LP é desenvolvida em cinco passos: (1) Aulas expositivas; (2) Aulas práticas de programação em laboratório; (3) Estudo em um livro texto e práticas de programação; (4) Desenvolver trabalhos em casa; (5) Métodos de avaliação e com isso observa-se que as dificuldades encontradas no ensino de uma LP, seja ela visual ou não, abrangem três aspectos: aulas formais, atividades em laboratório e práticas de programação. Aliada a estes aspectos está a literatura para o ensino de LP Visual, onde normalmente centenas de componentes estão agrupados sem uma contextualização.

Para muitos professores ensinar centenas de regras de sintaxe e de semântica parece ser uma tarefa tediosa. Sob o ponto de vista do aluno, aprender essas centenas de regras de sintaxe e de semântica em um curto espaço de tempo não é uma tarefa trivial. Em LP Visual a dificuldade também está atrelada à quantidade de componentes disponíveis para a criação de aplicações. Além disso, a capacidade de transmissão de conhecimento em aulas expositivas está intimamente ligada à aptidão do professor. No entanto, uma discussão sobre a aptidão do professor não faz parte deste trabalho.

As atividades em laboratório normalmente são realizadas objetivando dois aspectos: pôr em prática o que foi passado nas aulas expositivas e fornecer dicas de programação. O grande problema, nesse caso, é como atrair o aluno para as aulas de laboratório se na maioria das vezes o aluno pode fazer as atividades em sua própria residência. Em outras palavras, o desafio é tornar as aulas de laboratório mais importantes e definitivamente mais interessantes para o aluno.

Reservar um tempo de estudo para prática de programação é essencial, pois programação só se aprende programando. Entretanto, o tempo exigido nessa aprendizagem normalmente é maior do que o tempo gasto em sala de aula, pois o aluno sem prática comete muitos erros e não encontra ajuda imediata nas dificuldades encontradas durante o seu estudo. Nesse contexto, o aluno possui alguns recursos para tentar sanar suas dúvidas: mandar um e-mail para o professor/monitor, consultar um livro ou pesquisar na internet (Web ou fóruns). No primeiro caso, dependendo da dúvida, alguns professores podem considerá-la elementar demais e adiar sua resposta para o dia de sua aula. Consultar um livro depende se o aluno o adquiriu, o emprestou da biblioteca ou se o aluno encontra-se na instituição de ensino e a biblioteca encontra-se em funcionamento. E, a pesquisa na Internet depende do acesso a um provedor. Em qualquer que seja o caso, o aluno pode não conseguir tirar sua dúvida em um curto período de tempo. Esses são os mesmos problemas encontrados quando os alunos têm que realizar as tarefas propostas pelo professor.

Na literatura, centenas de componentes podem estar agrupados sem uma contextualização, ou seja, para cada componente ou grupo de componentes existe um modelo diferente. Por exemplo, no livro Spanghero (2003) para cada componente mostra-se um programa diferente. Para um aluno sem experiência pode ser difícil “juntar as peças” de modo a construir o conhecimento necessário para o desenvolvimento de uma aplicação. Na melhor das hipóteses, o livro pode ser de ajuda para aquele aluno que já possui um conhecimento prévio, esta desenvolvendo uma aplicação e que deseja apenas entender o funcionamento de um determinado componente. Em casos extremos, todos os

componentes são apresentados na primeira parte do livro e na segunda parte desenvolve-se uma aplicação com apenas uma parte dos componentes.

Nesse âmbito, este artigo propõe uma proposta metodológica para o ensino e aprendizagem de LP Visual na educação profissional de alunos de nível médio. Dessa forma, pretende-se contribuir com melhorias no processo de ensino-aprendizagem da Informática ministrada nos cursos profissionalizantes, com a finalidade de aperfeiçoar o aprendizado dos alunos através de uma prática educativa orientada por técnicas e meios adequados, enriquecendo o processo de construção do conhecimento. Em outras palavras, pretende-se tornar a aprendizagem de uma linguagem de programação mais atraente ao aluno.

Para cumprir seu objetivo este artigo está dividido da seguinte maneira: a Seção 2 aborda os aspectos da aprendizagem de conteúdos curriculares; na Seção 3 discute-se a respeito da Pedagogia de Projetos; na Seção 4 apresenta-se a metodologia proposta; na Seção 5 ilustra-se a aplicação desta metodologia na disciplina de Linguagem de Programação Visual; na seção 6 expõem-se os resultados do projeto; finalmente, na Seção 7 têm-se as conclusões deste trabalho.

2. APRENDIZAGEM DE CONTEÚDOS CURRICULARES

Sabe-se que muitos professores colocam em segundo plano os conteúdos curriculares. Outras vezes esses conteúdos são ministrados de maneira aleatória. O que ocorre é que esses educadores acabam fazendo uma interpretação errônea da visão construtivista, onde o educando deve ter autonomia para buscar as informações e o professor é um facilitador no processo educacional. Assim, esses professores acabam por gerar um esvaziamento dos conteúdos porque não combinam o saber com a prática educativa: ou são conteudistas ou entretêm o aluno de tal forma que não se passa conteúdo algum, quando não acontece do próprio aluno ministrar aulas em decorrência da ausência de intervenção do educador durante as apresentações de trabalhos.

Os conteúdos não são criados ao acaso, estes são produzidos de maneira sistemática. Independentes da área a que se destinam, eles interpretam e explicam fatos e fenômenos que regem um objeto ao mundo que o rodeia. Desta maneira, os conteúdos curriculares devem fazer parte da aprendizagem escolar sendo ministrados de forma responsável e significativa para os alunos à medida que valoriza e possibilita uma contextualização. Além disso, os currículos devem incentivar os alunos a cultivar uma vontade incessante de aprender (Somerville, 2005).

Atualmente, novas idéias de educação começaram a surgir e conceitos como os de habilidades e competências (capacidade do indivíduo de prover os recursos necessários) agora também estão incluídos nesse processo educacional (Campos, 2004).

Nesse contexto, a educação deve proporcionar uma formação mental-cognitiva, social, e de capacidade de realização (Campos, 2004). Isto se torna possível mediante o despertar e trabalhar das habilidades e competências a serem desenvolvidas. Sendo assim, ao desenvolver projetos, os estudantes com a ajuda do professor, podem mobilizar os conhecimentos adquiridos e colocá-los em prática para enfrentar tais situações e a conseqüente mobilização de suas estruturas conceituais (Campos, 2004).

3. A PEDAGOGIA DE PROJETOS

A Pedagogia de Projetos surgiu da necessidade de desenvolver uma proposta metodológica que pudesse reconhecer que tanto a participação do educador quanto a do aluno são essenciais no processo ensino-aprendizagem. Na verdade, por trás dos verdadeiros trabalhos com projetos encontra-se uma concepção pedagógica que analisa o conhecimento à medida que este é repassado ao aluno, com o intuito de transformar e reconstruir este conhecimento para que o mesmo possa ser enriquecido e expandido com a intervenção do educando.

Por serem trabalhados de maneira contextualizada, na Pedagogia de Projetos os conteúdos curriculares tornam-se realmente significativos para os alunos, despertando-lhes suas múltiplas inteligências através da orientação do professor, o qual ajuda os mesmos a estabelecerem relações entre as situações do dia-a-dia deles com situações que são observadas e construídas em sala de aula. Isto ocorre porque o educador, sob a ótica construtivista, vai resgatando os conhecimentos prévios dos estudantes e confrontando-os com as novas informações que estão sendo ministradas e assim possibilita uma real aprendizagem na qual os alunos adquirem condições de resolverem de maneira crítica e consciente situações cotidianas, habilidade esta muito importante e valorizada pela sociedade vigente.

4. PROPOSTA METODOLÓGICA

A proposta metodológica envolve cinco passos básicos conforme mostra a Figura 1. Os passos podem ser aplicados a qualquer disciplina que envolva a informática, sendo que cada um dos passos será detalhado a seguir.

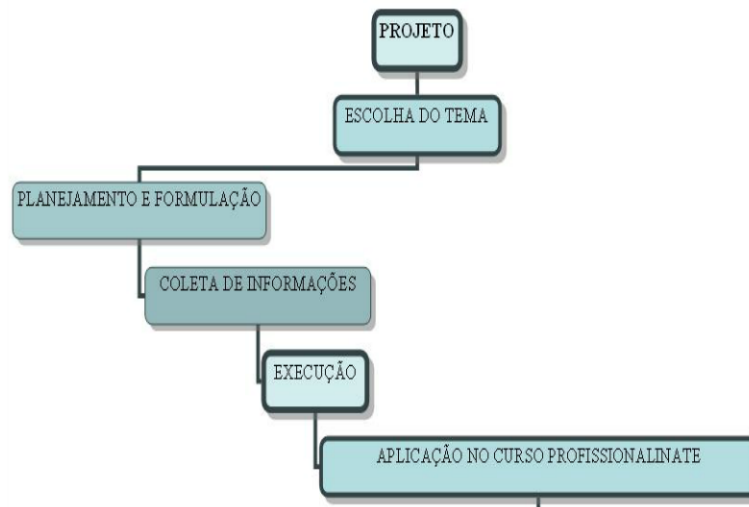


Figura 1. Proposta metodológica.

4.1 A Escolha do Tema

A escolha do tema gerador do projeto pode partir tanto do professor quanto dos alunos, ou dependendo da situação, de ambos. Na escolha deve-se levar em conta a organização curricular do curso e disciplina que está sendo tratada. O tema pode envolver uma única disciplina do curso profissionalizante em questão ou abranger várias disciplinas deste mesmo curso. Por exemplo, na disciplina de Modelagem de Sistemas (Módulo III) (DAI, 2006) pode-se solicitar o projeto de um sistema de acordo com a realidade que cada aluno vive como, por exemplo: um sistema de controle de vendas de uma frutaria, de uma farmácia, de uma pequena venda, etc.

4.2 Planejamento e Formulação do Projeto

Deve-se planejar e definir a atividade que os alunos irão desenvolver a partir do tema que foi gerado. Nesta etapa o professor deve fazer um levantamento (sondagem) do que os alunos já sabem (pré-requisito), o que ainda não sabem sobre o tema e quais são suas hipóteses e referências de aprendizagem. O professor precisa ter clareza de sua intencionalidade, ou seja, saber o que espera que os alunos aprendam com o projeto que será realizado e ter certeza do que o aluno está se propondo a desenvolver.

4.3 Coleta de Informações e Materiais Necessários

Com o auxílio do professor, os alunos irão trabalhar questões de pesquisa e investigações, pois vão ter que buscar, relacionar e articular informações com conhecimentos que já possuem para compreender melhor essas questões e chegar a construir novos conhecimentos. Neste passo é fundamental que o professor não se ausente e deixe a aquisição de conhecimentos apenas com os alunos, pois os alunos não possuem uma experiência suficiente para buscar sozinhos os conhecimentos necessários.

4.4 Execução do Projeto

Deve-se seguir um cronograma onde os alunos possam desenvolver e executar a atividade que foi pedida. É o momento que permite ao aluno representar o seu conhecimento, de modo que ele possa identificar o que sabe e o que precisa buscar para aprofundar esse conhecimento. Nesta etapa o professor observa o que os alunos estão aprendendo, como estão acompanhando o sentido do projeto e busca o desenvolvimento de competências nos alunos. Neste passo também

é possível identificar alguns aspectos importantes na formação por competência, tais como: a capacidade de trabalho em grupo, pontualidade e responsabilidade, aspectos esses essenciais para um bom posicionamento no mercado de trabalho.

4.5 Aplicação do Projeto no Curso Profissionalizante

É importante o professor destacar o sentido funcional do projeto para que os alunos possam perceber um significado no conteúdo que estão estudando e no projeto que estão desenvolvendo para poderem relacioná-los à vida profissional e cotidiana deles. A melhor maneira de se fazer isso é utilizando o projeto desenvolvido dentro do próprio curso profissionalizante relacionando o tema com a área do curso em questão.

5. APLICAÇÃO DA PROPOSTA METODOLÓGICA

Entendendo-se que a corroboração e aperfeiçoamento desta proposta metodológica só seriam possíveis após sua aplicação, esta foi feita no ensino da disciplina Linguagens de Programação Visual dirigida para o curso técnico de Programação de Computadores do CEFET-MA, especificamente para trabalhar com Delphi 7. Entretanto, nada impede que a proposta metodológica seja usada para outra disciplina de informática, assim como para outra linguagem de programação visual, sendo necessária apenas a adaptação dos componentes ensinados. O importante na escolha da linguagem é que ela permita que os alunos possam implementar uma grande variedade de algoritmos e que possam utilizar diferentes técnicas em seus programas (Minor e Gewali, 2004).

No processo de contextualização são utilizados dois projetos: um editor de texto e um sistema com acesso a banco de dados. Os componentes vão sendo mostrados na medida em que as aplicações vão sendo desenvolvidas e incrementadas. Em outras palavras, as aplicações começam simples e vão se tornando mais complexas à medida que novos componentes vão sendo apresentados, sendo que a prática de melhorias deve ser feita pelo aluno ou grupo de alunos logo após a explanação sobre o funcionamento do componente. Considera-se também que o aluno já possui uma desenvoltura na utilização da linguagem Pascal.

5.1 O Editor de Texto

O primeiro passo no ensino de uma LP Visual é mostrar o ambiente de desenvolvimento. Nessa etapa, mostra-se onde e como os componentes são posicionados (noção de formulários) além de como alterar as propriedades e eventos associados aos componentes considerados.

A Tabela I mostra a seqüência de componentes a serem apresentados. A coluna Item ressalta a ordem, sendo que nesta seção todos os Itens (componentes) apresentados referem-se à Tabela 1. Nesse contexto, a aula deve ser dividida em duas partes. A primeira onde o professor apresenta a utilização dos componentes. E, a segunda onde os alunos devem implementar o que acabou de ser apresentado. Em cada aula o professor deve apresentar no máximo dois itens, considerando que cada aula possui duas horas/aula. Na parte expositiva o professor deve mostrar tanto os componentes quanto os eventos associados. Os códigos que devem ser utilizados devem também ser explicados cuidadosamente. Sendo que esse procedimento torna o aprendizado mais agradável e interessante já que o aluno consegue visualizar rapidamente o resultado do conhecimento que está adquirindo.

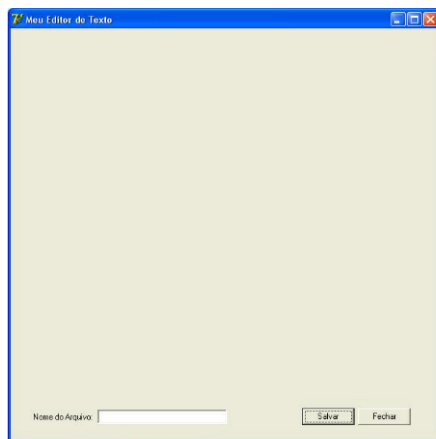
Os componentes do item 1 são os primeiros a serem apresentados. O aluno começa a se familiarizar com o ambiente e começa a entender a programação orientada a eventos. Os primeiros passos na programação visual são dados mostrando mensagens ao se clicar no botão Salvar. Este passa a ser o primeiro programa do aluno, o famoso “*Hello Word!*”. A Figura 2a mostra como fica o esqueleto do editor de texto.

Em seguida apresenta-se o componente do item 2. Neste ponto, adiciona-se um botão Abrir e a função do botão Salvar é incrementada, ou seja, esses botões vão efetivamente abrir e salvar um texto, respectivamente, através dos métodos *SaveToFile* e *LoadFromFile*. Nesse contexto, a aparência do editor passa a ser o mostrado na Figura 2b.

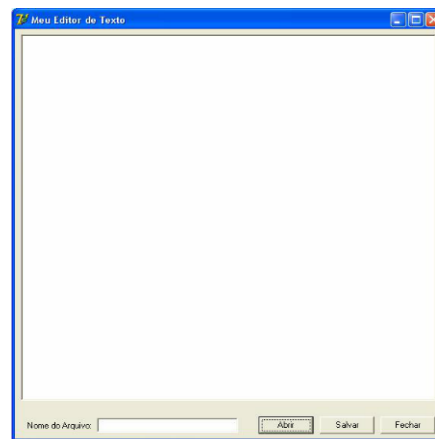
Como observado, os itens vão sendo apresentados gradativamente. Nesse contexto, o editor de texto passa pelo formato apresentado na Figura 3a e termina com um layout semelhante ao apresentado na Figura 3b. Durante esse processo, o professor deve solicitar exercícios onde os componentes apresentados em itens anteriores devem ser substituídos por novos itens, como por exemplo, pedir a substituição do *ListBox* (item 6) pelo *ComboBox* (item7), e assim sucessivamente.

Tabela I: Componentes básicos.

Item	Componente	Propriedades	Comandos/Métodos	Eventos
1	Botões, Label e Edit	Caption, Name, Text, Font	ShowMessage, Close	OnClick, OnKeyPress e OnCahnge (opcional)
2	Memo	WantReturns, WantTabs, WordWrap e Lines	SaveToFile e LoadFromFile	
3	Criando formulários	Name, Caption	Show, ShowModal, Use Unit	
4	GroupBox e RadioButtons	Name, Caption, Checked		
5	RadioGroup	Items, ItemIndex		
6	ListBox	Items, ItemIndex, Item.String, DropDownCount, Sorted	Add	
7	ComboBox	Items, ItemIndex, Text, Maxlength	Add	
8	SaveDialog e OpenFileDialog	Filter, FilterIndex Save – Option.ofFileMustExist Open – Optinos.ofOverWritePromt	Execute	
9	FindDialog	FindText, Position, Top, Left, frHideMatchCase, frHideUpDown,, frHideWholeWord Novas propriedades do Memo: SelStart, SelLength e SetFocus	Execute	OnFind
10	FontDialog	Device, Font, Options.fdLimitSize, MaxFontSize, MinFontSize. Método usado no Memo : Font.	Assign	
11			MessageDlg e MessaeBox	
12	Menus	Itens		
13	RichEdit	HideScrollBar, Paragraph, SelAttributes, ScrollBars, HideScrollBars		
14	SpeedButton	Glyph, Layout, Margin, GroupIndex, Down e AllowAllUp		

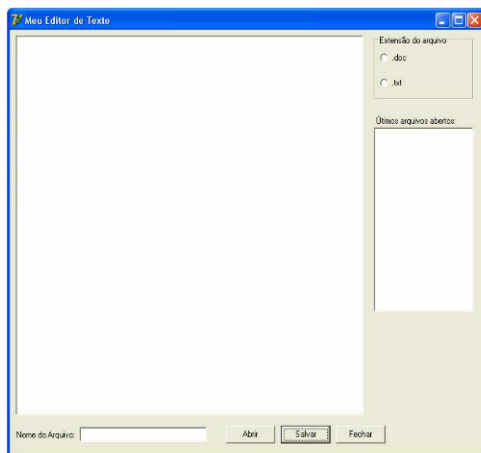


(a) Primeira versão



(b) Segunda versão

Figura 2 – Aparência inicial do editor de texto



(a) Versão intermediária.



(d) Versão Final

Figura 3 – Aparência final do editor de texto

5.2 O sistema de banco de dados

A construção do sistema de banco de dados é composta de duas partes. A primeira parte trabalha com os componentes BDE e forma a base para utilização dos componentes de acesso a banco de dados. A segunda parte refere-se ao acesso a banco de dados utilizando o Firebird (versão livre do Interbase) como sistema gerenciador. Não obstante, a metodologia pode ser adaptada para outros bancos de dados de acordo com as necessidades do curso. Por exemplo, a primeira parte pode ser formada pelos componentes ADO e acesso ao Microsoft Access e segunda parte pode ser dirigida para acesso a outros sistemas gerenciadores de banco de dados, tais como: MySQL, SQL Server, Oracle, etc.

O conhecimento sobre modelagem e SQL advém da disciplina de Banco de Dados que é ministrada no período anterior ao de Linguagem de Programação Visual. Antes de se iniciar o desenvolvimento do sistema, pode se ministrar uma aula de revisão de SQL, especialmente no que se refere a junções e *sub-queries*.

O modelo do sistema a ser desenvolvido é apresentado na Figura 4. A partir dele os componentes vão sendo apresentados e as tarefas vão sendo designadas, sendo que diferentemente do editor de textos, o *Layout* da aplicação é deixado por conta do aluno.

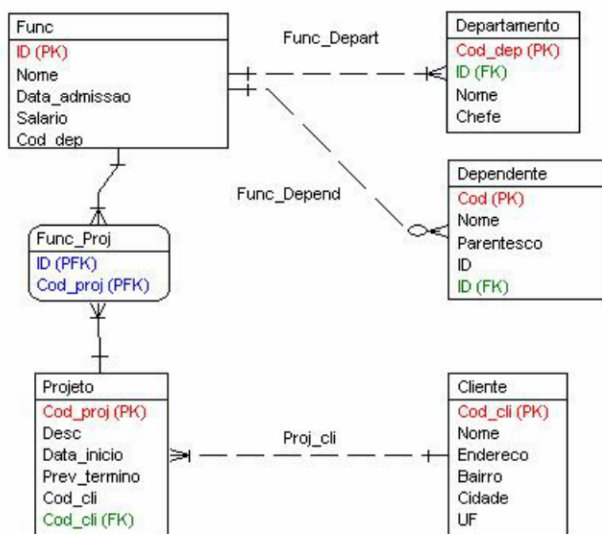


Figura 4 – Modelo relacional do BD a ser implementado.

A Tabela II apresenta os componentes relacionados à utilização do BDE e a Tabela 3 mostra os componentes relacionados à utilização do FireBird, respectivamente.

Tabela II: Componentes BDE.

Item	Componente	Propriedades	Comandos/Métodos	Eventos
1			Database Desktop - Ambiente para desenvolvimento de BD nativos.	
2	Table, Datasource, DBGrid e DBNavigator	DatabaseName, Dataset, TableName, ReadOnly, Active DBGrid – Align, DataSource e dgTitle DBNavigator – DataSet e ConfirmDelete		
3	DBEdit e DBText	Name, Datasource e DataField		
4	Query	DataBaseName, SQL e Active	SQL.Clear, SQL.Add, SQL.Append, Open, SQLExecute e Close	
5	DBLookupComboBox	DataField, ListSource, ListField e KeyField		OnChange ou OnClick
6	DataModule	Name		OnCreate
7	Page Control			

O primeiro passo para a construção do sistema, de acordo com a Tabela II, é aprender a utilizar o *Database Desktop* para a criação de tabelas e relacionamentos. O professor deve mostrar o processo para duas tabelas, por exemplo, *Func* e *Departamento* ou *Func* e *Dependente*. Com essas duas tabelas é possível mostrar a criação de relacionamentos e as restrições de integridade. Como tarefa após a exposição os alunos devem criar as demais tabelas, seus respectivos relacionamento e inserir dados para realizar as primeiras consultas utilizando a LP Visual, o que será feito em aulas posteriores. Esse mesmo esquema deve ser seguido para os demais componentes, ou seja, mostrar a utilização do componente em uma tabela e o restante deve ser feito pelos alunos.

No item 2 são explanadas as regras de acesso a BD, ou seja, como os dados são ligados a LP Visual através de Datasets e como o acesso deve ser feito através de um *Datasource*. Aqui se apresentam os dois primeiros componentes de acesso a dados *Data Controls* (*DBGrid* e *DBNavigator*) que inicialmente não exigem nenhuma linha de código. No item 3 os dados começam a ser manipulados isoladamente, isto é, por campos. Nesta etapa os alunos devem construir os formulários para inserir dados. Esse processo é facilitado pelo *DBNavigator* que também não exige codificação.

Os primeiros passos de codificação começam a ser apresentados no Item 4, onde são efetuadas consultas simples e mostradas as primeiras junções. A implementação da tabela *Func_Proj* deve ser feita pelo professor, onde serão apresentados os componentes do item 4, sendo que a inserção de dados passa a ser não mais pelo *DBNavigator* mas por um comando SQL (*Insert*) associado a um botão.

Em seguida trabalha-se com o componente *DBLookupComboBox*, sendo utilizado para fazer a busca nos dados que já existem facilitando a inserção dos dados relacionados. Os conceitos envolvidos na utilização de *DataModules*, onde o sistema irá concentrar todos os componentes de acesso aos dados, é apresentado em seguida.

Pra finalizar, mostra-se o componente *PageControl*, o qual permite mostrar em um único formulário diversos tipos de relações, por exemplo, no formulário de *Funcionario* são apresentados todos os dependentes em uma aba e todos os projetos associados ao funcionário, consolidando os conhecimentos da LP Visual com os conhecimentos adquiridos na disciplina de banco de dados.

A segunda etapa no desenvolvimento de um sistema de banco de dados é o desenvolvimento da aplicação utilizando o Firebird. O primeiro passo desta etapa é mostrar como utilizar o sistema gerenciador, o qual pode ser acessado por uma interface gráfica como, por exemplo, o IBOConsole (Firebase, 2006).

Diferente do *Database Desktop*, no IBOConsole as tabelas e seus relacionamento devem ser criadas utilizando-se SQL. Nesse contexto, os alunos já devem ter certo domínio em SQL. Por trabalhar apenas com SQL a quantidade de métodos para os componentes de controle dados é menor, sendo necessário, basicamente, dois componentes o *SQLConnection* (Item 2) e o *SimpleDataSet* (Item 3). O *SQLConnection* controla a conexão com o banco de dados. Já, o *SimpleDataSet*

pode ser utilizado tanto para acessar tabelas quanto para realizar consultas (*queries*). A interface desses componentes com os componentes de controle de dados também é feito através de um *Datasource*.

Tabela III: Componentes FireBird.

Item	Componente	Propriedades	Comandos/Métodos	Eventos
1			Utilização do IBOConsole e Firebird.	
2	SQLConnection	ConnectionString, DriverName, Database, UserName e password		
3	SimpleDataSet	Connection, DataSet.CommandType, SQLQuery e SQLTable.		
4			Introdução à Store Procedures no Firebird	
5	SQLStoredProc	DatabaseName, SQLConnection e StoredProcName		

Os demais componentes não são fundamentais, pois são unidirecionais. Em outras palavras, só aceitam a movimentação do cursor para frente. Isso é um problema quando se deseja utilizar componentes bidirecionais como por exemplo o *DBGrid*.

No último passo apresenta-se a utilização de *Store Procedures*, cuja função é centralizar as regras de negócio no banco de dados ao invés de mantê-las na aplicação, em especial quando se trata de aplicações *Desktop*. No entanto, a utilização das mesmas e dos componentes apresentados nos itens 4 e 5 são opcionais e podem ser apresentados de acordo com a evolução da turma. Sendo assim é facultado ao professor solicitar como exercício que as regras de negócio sejam colocadas em *Store Procedures*.

5.3 A avaliação

O processo de avaliação continuada deve se dar constantemente durante uma disciplina, ou seja, não mais pontualmente apenas com provas em determinadas datas. Não obstante, as provas podem ser utilizadas como um instrumento auxiliar para identificar pontos vulneráveis na evolução da turma. Nesse contexto, o processo de avaliação se dá em diversos aspectos, tais como: capacidade de trabalho em grupo, responsabilidade, pontualidade e participação. Como a aula é dividida em teoria e prática, a avaliação é praticamente diária sendo feita principalmente durante a parte prática. Em outras palavras, deve se observar se o aluno está efetivamente desenvolvendo o programa de acordo com o especificado pelo professor.

A capacidade de trabalho em grupo é avaliada de acordo com o comportamento durante a implementação. Nesse caso, observa-se que existe sempre um líder e deve-se ter cuidado para que o líder não monopolize o computador, o qual deve ser utilizado por todos os membros do grupo. Nesse contexto, o professor pode exigir um rodízio dos alunos que controlam o computador. Neste momento é possível também verificar a participação efetiva do aluno dentro da equipe.

A responsabilidade é avaliada de acordo com as atividades que foram solicitadas. Caso atividades de outras disciplinas estejam sendo feitas durante a parte prática deve haver uma intervenção do professor. Já, a pontualidade é avaliada de acordo com as datas estipuladas para a entrega dos incrementos nos sistemas. A cada certa quantidade de dias atrasados novas funções (não apresentadas pelo professor) devem ser solicitadas.

6. RESULTADOS E DISCUSSÃO

A metodologia foi aplicada à disciplina de Linguagem de Programação Visual do Centro Federal de Educação Tecnológica do Maranhão. Nesse âmbito, dadas as dificuldades no ensino de LP, alguns resultados foram verificados.

- A exposição de regras (sintáticas e semânticas) e de componentes é menor, concentrando o esforço do professor na qualidade dos incrementos da aplicação considerada.
- As atividades em laboratório deixam de ser convencionais, pois os alunos passam a depender delas para poder realizar tanto os incrementos solicitados em sala quanto os incrementos solicitados como exercícios em casa. Sendo assim, as aulas tornam-se não somente importantes, mas essenciais para o processo de aprendizagem. Outro ponto a considerar é que o aluno passa a ver seu programa funcionando desde o primeiro dia, o que torna a aula definitivamente mais interessante.

- A prática de programação é muito mais agradável tendo em vista que o professor está presente para tirar as dúvidas do aluno. É óbvio que apenas aulas de laboratório não são suficientes para fornecer a prática necessária para a formação de um desenvolvedor de sistemas. Não obstante, os alunos cometem menos erros quando estão praticando no laboratório em horários livres ou em casa em suas horas vagas. Esse fato se estende também aos incrementos que o professor solicita.
- As aulas práticas no laboratório fornecem um *background* para que o aluno possa buscar conhecimento em livros de forma mais rápida, pois a construção do seu conhecimento já está contextualizada.

A aplicação da metodologia também apresentou dificuldades. Especialmente as inerentes à aplicação de uma nova metodologia de ensino, já que a grande maioria dos professores atuais pode ter estudado via métodos tradicionais de ensino, onde o professor é o detentor de um conhecimento inquestionável. Outras dificuldades encontradas são: adequar-se a uma nova forma de avaliação, a falta de recursos fornecidos pela instituição para ministrar a disciplina e a resistência de alguns alunos à nova metodologia.

A dificuldade de se adequar à nova forma de avaliação vem do fato da mesma ser constante. Esse tipo de avaliação demanda bastante trabalho em sala de aula. Por outro lado, ganha-se tempo, pois não se mostrou necessária a utilização de provas convencionais, as quais exigem tempo extra (fora da sala de aula) para sua correção.

A falta de recursos para o ensino favorece o aparecimento de dúvidas, pois é extremamente complicado mostrar o funcionamento de componentes visuais apenas com o quadro branco/negro. A dificuldade de mostrar o funcionamento de componentes para uma turma de mais de 20 alunos em um monitor de 17 polegadas também não é menos complexo. Nesse contexto, a distribuição de transparências com os passos necessários para a utilização desses componentes (propriedades e métodos mais importantes) é essencial para facilitar o processo de ensino-aprendizagem.

A resistência de alguns alunos à nova metodologia pode ser vista nos gráficos apresentados na Figura 5. Vê-se na Figura 5(a) que a metodologia foi considerada boa de maneira geral. Nenhum aluno a considerou a metodologia excelente, talvez por ser a primeira experiência de ambos (professor e aluno) nesse tipo de metodologia. Apesar disso, menos da metade sugerem mudanças como pode ser visto na Figura 5(b).

Outro ponto que pode levar à resistência de alguns alunos é a falta de costume de pesquisar conhecimento, pois os mesmos estão acostumados a aulas expositivas. Um resultado interessante é que dos vinte e dois alunos que iniciaram o curso, vinte e um foram aprovados. Isso representa um percentual de 95,4% de aprovação. Em contrapartida, a turma que foi submetida à forma tradicional de avaliação teve um percentual de apenas 50% de aprovação.

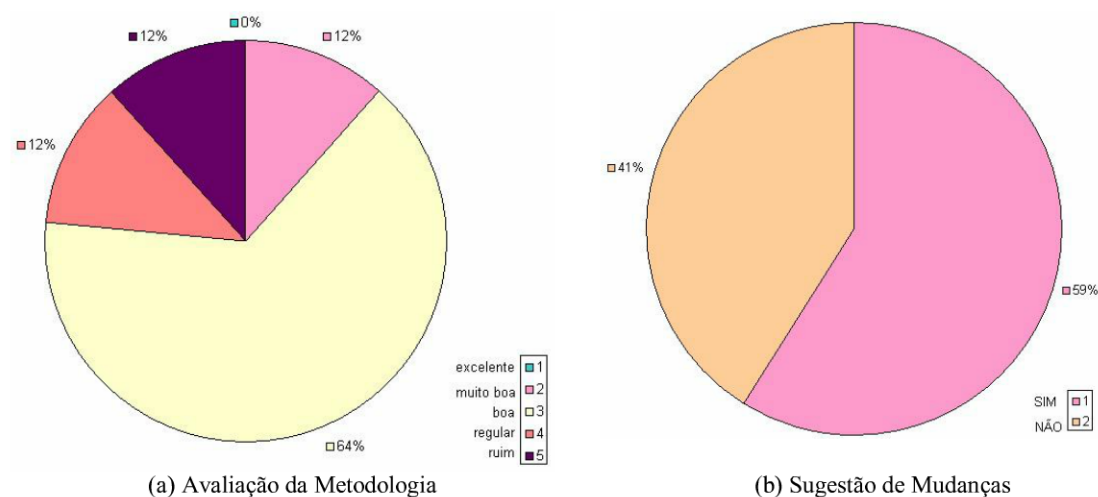


Figura 5 – Resultados da avaliação.

7. CONCLUSÕES

Neste trabalho desenvolveu-se um recurso metodológico voltado para o ensino de Informática nos cursos profissionalizantes de nível médio, escolhendo-se aplicar no ensino da Linguagem de Programação Visual, trabalhando o ambiente Delphi, mas nada impede que a proposta metodológica seja utilizada em outra disciplina de informática.

Espera-se que as contribuições deste projeto possam se fazer presentes à medida que ele oportunize maior integração professor-aluno e proporcione melhores condições de ensino e aprendizagem, além de facilitar novas construções de conhecimento pelos alunos, possibilitando-lhes uma qualificação para o mundo do trabalho atuando de maneira consciente na sociedade, com incentivo e orientação dos educadores e da instituição chamada escola.

Agradecimentos

Os autores agradecem o apoio dado pelo CNPq que tornou possível a execução deste trabalho.

8. REFERÊNCIAS

Almeida, F. J. de. , Junior, F. F. **Aprendendo com projetos**. Brasília, PROINFO/MEC, 2000.

Antunes, C. **A Avaliação da Aprendizagem Escolar**. 3ª ed. Petrópolis: Vozes, 2003.

B. du Boulay, C., Sothcott, C. **Intelligent systems for teaching programming**. In Ercoli, P. and Lewis, R., editors, *Artificial Intelligence Tools in Education*. Elsevier Science Publishers, 1988.

Barbosa, Rommel Melgaço. **Ambientes Virtuais de aprendizagem**. Porto Alegre: Artmed, 2005.

Campos, G.H.B. de. **A EAD e o modelo de competências**. Artigo. COPPE/Universidade Federal do Rio de Janeiro (UFRJ), 2004.

DAI. [Online]. Disponível em: <http://www.dai.cefet-ma.br/site/paginas/index.jsp>. Visitado em 28 de outubro de 2006.

Duarte, A.L.C. **Curso de Didática: Do desgaste conceitual na escola à revitalização da Prática Pedagógica**. In Reunião Regional da SBPC - MA (UEMA, março, 2004).

Firestore, Firestore's Home Page, <http://www.firestore.com.br/fb/>, Visitado em 26/10/2006.

J. T. Minor, L. P. Gewali, **Proceedings, Pedagogical Issues in Programming Languages**, In Proceedings of the IEEE International Conference on Information Technology: Coding and Computing, v. 1, p. 562 – 565, 2004.

Sancho, J.M. **Para uma Tecnologia Educacional**. Porto Alegre: ArtMed, 2001.

Spanghero, A. **Aprendendo Delphi 7: Guia Prático**, Futura: São Paulo, 2003.

Somerville, M., Anderson, D., Berbeco, H., Bourne, J. R. , Crisman, J. **The Online Curriculum**, *IEEE Transaction on Education*, v. 48, n. 1, p. 198-205. 2005.