

SISTEMA PARA AUTOMAÇÃO DE TRANSPORTE ESCOLAR: T-ESCOLAR

Michel Rosin Martins, Solange Maria Cottica, Fábio Alexandre Spanhol,
Juliano Rodrigo Lamb, Rui Rossi dos Santos da Silva, e-mail:
michelrm@gmail.com

Faculdade Sul Brasil – Fasul/Sistemas de Informação – Toledo – PR.
Universidade Tecnológica Federal do Paraná – Toledo - PR.

Palavras-chave: Linguagem Java, *MySQL*, Desenvolvimento de *software*.

Resumo

O presente trabalho desenvolveu um software para a automação do transporte escolar, denominado T-Escolar, que pode ser utilizado por prefeituras ou empresas que necessitem de um controle eficaz de trajetórias, usuários e outros itens relacionados ao transporte de passageiros. Após feito o levantamento de requisitos e elaborado o projeto de banco de dados, fez-se a implementação do sistema, através da linguagem de programação Java. Entre as funcionalidades do Sistema para a Automação do Transporte Escolar, o T-Escolar, enquadra-se a qualidade da interação homem-máquina apresentada e, tomando-se por base a plataforma de desenvolvimento *web*, tem-se um sistema consistente e de fácil operação por qualquer usuário com o mínimo de treinamento para fazer as operações que o sistema exige do usuário que irá operá-lo.

Introdução

Nos dias de hoje, é de fundamental importância uma empresa se informatizar para obter resultados eficazes e confiáveis em todo o processo de sua administração, seja ela contábil, estoque, contas a receber e a pagar, entre outros serviços.

Com o grande crescimento da indústria de *software*, fica fácil encontrar um sistema que possa suprir as necessidades de determinada empresa em seu processo administrativo. Porém, isso pode gerar alguns transtornos, pois os sistemas que estão disponíveis no mercado, muitas vezes não atendem por completo as necessidades específicas da empresa, haja vista que o sistema não foi projetado para tal. Desta forma, a falta de um sistema adequado faz com que não haja um bom controle, por exemplo, no serviço de transporte escolar de determinada empresa, fazendo com que informações sejam perdidas e mal gerenciadas.

A proposta deste projeto foi desenvolver um sistema *web*, utilizando a linguagem Java, juntamente com o gerenciador de banco de dados *MYSQL* e um sistema gerenciador de transporte escolar, com a finalidade de gerenciar serviços de transporte escolar e gerenciar trajetos, o custo do

mesmo e alunos que dependem desse serviço. Para tanto, fez-se inicialmente um levantamento dos requisitos, elaborou-se o projeto de banco de dados e por fim implementou-se o sistema, ou seja, construiu-se o sistema proposto.

Materiais e Métodos

Análise de Requisitos

Inicialmente fez-se um levantamento dos requisitos necessários para a elaboração do sistema. Esta etapa buscou junto ao usuário, seus sistemas e documentos, todas as informações possíveis sobre as funções que o sistema deve executar e as restrições sob as quais o sistema deve operar (WASLAWICK, 2004). Desta forma, definiram-se como requisitos para este trabalho cadastrar alunos, usuários, transportadoras, escolas, trajetos, custo por trajeto, calcular custo por trajeto, emitir relatórios de custo por trajeto, trajeto por aluno, emitir relatórios de alunos, de escolas, de transportadoras e abrir movimentação do dia.

Uma vez que os requisitos foram levantados, organizou-os em grupos correlacionados, de forma a abordá-los nos ciclos iterativos, sendo portanto, agrupados em casos de uso, conceitos e consultas.

Linguagem Utilizada

A tecnologia de desenvolvimento de *softwares* utilizada foi a linguagem Java. Esta linguagem é empregada em aplicações *desktop*, *móvel* e *web*, sendo desenvolvida pela Sun Microsystems em 1991. A linguagem Java ganhou força em 1995, com a expansão da Internet e com o suporte que os principais programas utilizados para navegar pela *web* passaram a dar à execução de pequenos aplicativos escritos em Java, chamada de *applets* (SANTOS, 2007).

A linguagem Java compõe-se de três partes distintas que são elas: um ambiente de desenvolvimento, uma linguagem de programação e uma interface de programas aplicativos *API (Applications Programming Interface)*. O ambiente de desenvolvimento é o conjunto de ferramentas utilizadas para a construção de aplicativos, chamado de *kit* de desenvolvimento padrão Java que são eles: um compilador (*javac*), um interpretador de aplicativos (Java), um visualizador de *applets (appletviewer)* e por fim um gerador de documentação (*javadoc*). A linguagem Java, por sua vez, é composta por um conjunto de palavras e símbolos utilizados pelos programadores para escrever os programas. A interface da linguagem tem um sentido particular em Java e não deve ser confundida com a interface gráfica de usuário. Do mesmo modo que uma classe, uma interface pode abrigar atributos e métodos (SANTOS, 2004).

A edição Java utilizada neste trabalho foi a *JSE (Java Platform, Standard Edition)*. Esta é uma edição padrão da plataforma Java e provê ferramentas e *APIs* para o desenvolvimento de aplicações *desktop* e alguns

tipos de aplicações servidoras. Ela representa a base de sustentação da plataforma *JEE (Java Platform, Enterprise Edition)*, uma vez que esta depende e faz uso do núcleo central de *API*, da plataforma *JSE* (SANTOS, 2007).

Sistema de Gerenciamento de Banco de Dados

O sistema de gerenciamento de banco de dados utilizado foi o *MySQL*, que é um sistema relacional, multiencadeado, de código-fonte aberto e nível corporativo. Ele possui uma agilidade muito grande, é flexível e confiável em se tratando de desenvolvimento *web* (GONÇALVES, 2006) e foi utilizado nesse projeto para o armazenamento dos registros do serviço de transporte escolar.

Desenvolvimento de Diagramas

Para o desenvolvimento de diagramas como casos de uso, modelo conceitual, diagrama de classes e seqüência foi utilizada a ferramenta *CASE (Computer Aided Systems Engineering)*, desenvolvida pela Sybase Inc, denominada Sybase Power Designer 12.1. Esta é uma ferramenta que serve para modelagem padrão da *UML (Unified Modeling Language)* e modelagem de dados nos mais diversos níveis de sistemas. Esse *CASE* também foi utilizado como a modelagem física do banco de dados.

Para construir vários tipos diferentes de diagramas, tais como entidade relacionamento, diagramas de *UML*, fluxogramas, diagramas de rede, entre outros, será utilizado o programa Dia. Este programa é baseado em *GTK (Gimp Toolkit)*, para a criação do diagrama, liberado sob a licença do *GPL (General Public License)*, inspirado no programa MS Visio. O Dia também será utilizado para desenvolvimento do modelo conceitual de banco de dados.

Interface de Desenvolvimento do Ambiente

Como Interface de Desenvolvimento do Ambiente será utilizado o *Netbeans*, desenvolvido pela Sun Micro System, *IDE (Interface Development Environment)*, *open-source*, escrito totalmente em Java e utilizado para construir aplicações *desktop*, *web* e portáteis. Esta é uma ferramenta que permite aos desenvolvedores escrever, compilar e debugar suas aplicações (GONÇALVES, 2006), sendo adotada para o desenvolvimento de todo o *software* proposto nesse contexto, por ser uma *IDE* completa, robusta e eficaz na produção de aplicativos.

Ferramenta para o Desenvolvimento de Aplicações para a Web

A ferramenta para o desenvolvimento de aplicações para a *web* empregada foi a *Adobe Dreamweaver CS3*, que pode ser utilizada em modo visual, modo de código, ou ambos na mesma janela. Esta pode possuir

várias funções pré-definidas, assim como os componentes padrões de *HTML (Hyper Text Markup Language)*, *JSP (Java Server Pages)* e também *beans* e acessos a banco de dados diretamente. Este *software* servirá para a modelagem de *layout* do sistema e de todo o gerenciamento de tabelas e dados dentro da página *web*.

Paradigma de Linguagem

No presente trabalho será adotada a *POO* Linguagem de Programação Orientada a Objetos Java. O paradigma de linguagem *OO* (Orientação a Objetos), surgiu por volta de 1967, com a criação da Linguagem de Programação Simula-67 (*Simulation Language*). No início da década de 80 esse paradigma tomou força com a introdução da Linguagem Smalltalk. Esse paradigma provém da necessidade de solucionar problemas existentes no desenvolvimento de softwares complexos e confiáveis, com o melhor controle nos custos de desenvolvimento e de manutenção.

O projeto *OO* oferece uma maneira mais natural e intuitiva para visualizar o processo de projeto, ou seja, modelando objetos do mundo real, seus atributos e seu comportamento. A *OO* também modela a comunicação entre objetos, da mesma maneira que as pessoas enviam mensagens umas às outras, como por exemplo, o sargento que ordena ao soldado para se manter em posição de sentido, os objetos também se comunicam através de mensagens (DEITEL, 2003).

Modelagem de Sistemas Orientados a Objetos

A modelagem de sistemas Orientados a Objetos utilizada foi a *UML*, que é uma linguagem desenvolvida em meados da década de 90, sob a direção inicial de um trio de metodologistas de *software*, Grady Booch, James Rumbaugh e Ivar Jacobson (DEITEL, 2003).

A *UML* tornou-se o esquema de representação gráfica mais amplamente utilizado para a modelagem de sistemas orientados a objetos. Ela unificou os diversos esquemas de notação mais populares, aqueles que projetam e usam a linguagem sob a forma de diagramas gráficos para modelar seus sistemas (DEITEL, 2003).

Por essa linguagem possuir essas características de orientação a objetos e por possuir uma flexibilidade para tal, foi escolhida para a modelagem e estrutura de documentação deste trabalho.

Modelo Relacional de Dados

O modelo relacional de dados foi utilizado para a modelagem do projeto de banco de dados neste trabalho por apresentar simplicidade e robustez. O mesmo foi introduzido por Codd em 1970, e tornou-se um padrão de fato para o desenvolvimento de aplicações comerciais, devido a sua simplicidade e performance. Este modelo é baseado na teoria matemática das relações, considerando que um banco de dados relacional

consiste em uma coleção de tabelas, que podem ser relacionadas através de seus atributos, ou seja, uma linha de uma tabela pode ser relacionada com uma outra linha em uma outra tabela.

Resultados e Discussão

Em um sistema de informações, estão contidas todas as informações necessárias ao objetivo do sistema, sendo que os dados originados dessas informações são processados pelo sistema criado.

A primeira etapa do projeto foi o modelo conceitual (Figura 1) e este descreve a realidade do ambiente do problema, constituindo-se em uma visão global dos principais dados e relacionamentos, independente das restrições de implementação.

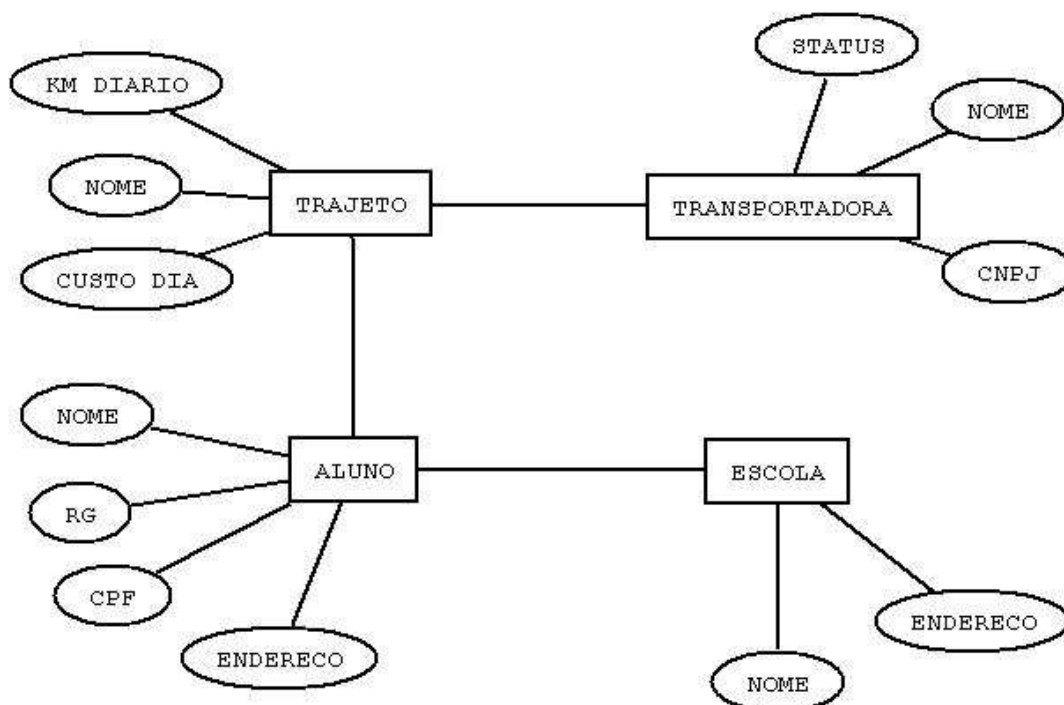


Figura 1 – Modelo Conceitual de Dados

O modelo conceitual de dados descreve as informações contidas em uma realidade, as quais estão armazenadas em um banco de dados. É também uma descrição de alto nível, mas que tem a preocupação de captar e retratar toda a realidade de uma organização, setor, repartição, departamento, entre outros (MACHADO & ABREU, 1996).

Determinado o modelo conceitual, iniciou-se o modelo lógico, que levou em consideração uma das abordagens atualmente possíveis, como Relacional, Hierárquica e Rede, dentre outros. Coletaram-se informações das características de cada entidade, ou seja, de cada tabela do banco de dados. Partiu-se do modelo lógico para implantar o modelo físico, que descreve as estruturas físicas de armazenamento de dados, tais como:

tamanho de campos, índices, tipo de preenchimento destes campos, nomenclaturas, entre outros. Além de ser projetadas de acordo com os requisitos de processamento e uso mais econômico dos recursos computacionais, este modelo detalhou o estudo dos métodos de acesso do *SGDB*, para elaboração dos índices de cada informação colocada nos Modelos Conceitual e Lógico. Esta foi a etapa final do projeto de banco de dados, na qual foi utilizada a linguagem de definição de dados do *SGDB*, *DDL (Data Defenition Language)*, para a realização da sua montagem no dicionário de dados (MACHADO & ABREU, 1996).

Em ambiente de banco de dados relacional denomina-se de *script* de criação de banco de dados, o conjunto de comandos em *SQL DDL*, que é executado no Sistema Gerenciador de Banco de Dados para a criação de banco de dados correspondente ao Modelo Físico (MACHADO, 2004). Na Figura 2, apresenta-se como foi estruturado o modelo físico para a realização deste trabalho.

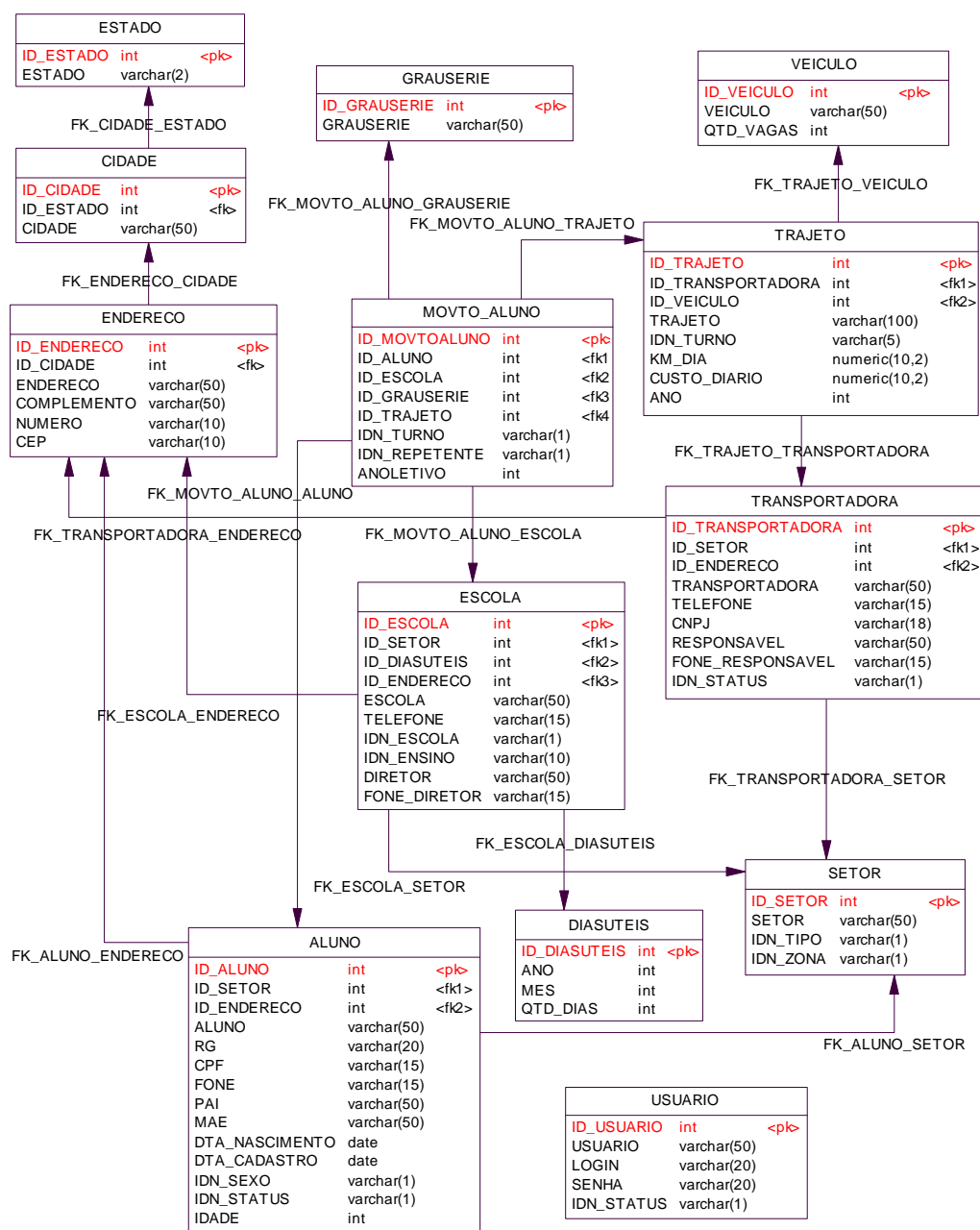


Figura 2 – Modelo Físico de Dados

Após estruturado o modelo físico de dados, implementou-se o sistema de transporte escolar. No menu cadastro aluno (Figura 3), o usuário pode cadastrar um novo aluno e após ele pode salvar esse novo registro. No botão buscar o usuário pode buscar todos os dados cadastrados e no botão voltar pode retornar à página principal do sistema.

Cadastro de Aluno

Nome Aluno:	<input style="width: 90%;" type="text"/>		
Endereco:	<input style="width: 90%;" type="text" value="RUA MEN DE SA"/>		
Setor:	<input style="width: 80%;" type="text" value="JARDIM PRIMAVERA"/>	Ex:	<input style="width: 20px;" type="text"/>
Data Nascimento:	<input style="width: 80%;" type="text"/>	Ex:	<input style="width: 20px;" type="text"/>
Data Cadastro:	<input style="width: 80%;" type="text"/>	Ex:	<input style="width: 20px;" type="text"/>
Idade:	<input style="width: 90%;" type="text"/>		
Sexo:	<input style="width: 80%;" type="text"/>	Ex:	<input style="width: 20px;" type="text"/>
Status:	<input style="width: 80%;" type="text"/>	Ex:	<input style="width: 20px;" type="text"/>
RG:	<input style="width: 80%;" type="text"/>	Ex:	<input style="width: 20px;" type="text"/>
CPF:	<input style="width: 80%;" type="text"/>	Ex:	<input style="width: 20px;" type="text"/>
Nome Pai:	<input style="width: 90%;" type="text"/>		
Nome Mae:	<input style="width: 90%;" type="text"/>		
Tel. Contato:	<input style="width: 80%;" type="text"/>	Ex:	<input style="width: 20px;" type="text"/>
<input type="button" value="Gravar"/> <input type="button" value="Voltar"/>			

Figura 3 – Janela de Cadastro de Aluno

Entre as funcionalidades do Sistema para Automação de Transporte Escolar, o T-Escolar, enquadra-se a qualidade da interação homem-máquina apresentada. Tomando-se por base a plataforma de desenvolvimento *web*, tem-se um sistema consistente e de fácil operação por qualquer usuário com o mínimo de treinamento para fazer as operações que o sistema exige do usuário que irá operá-lo. No cadastro, o usuário pode encontrar situações que quando ocorrer o não preenchimento dos campos o sistema informa o não preenchimento e não permite prosseguir no processo do cadastramento.

Além disso, a utilização de cores padronizadas para todas as telas na apresentação dos dados torna a interpretação da informação clara e de fácil entendimento pelo usuário que está operando o sistema.

Conclusões

O sistema desenvolvido para gerenciar o transporte escolar, que utilizou a linguagem Java juntamente com o gerenciador de banco de dados *MySQL*, o T-Escolar, apresentou-se de fácil utilização e praticidade,

podendo ser utilizado por prefeituras ou empresas relacionadas ao ramo de transporte de pessoas.

Agradecimentos

Agradecemos à Prefeitura de Marechal Cândido Rondon pelo auxílio prestados na coleta de informações e dados utilizados no desenvolvimento deste software.

Referências

- Deitel, H. M. *Java Como Programar*. 4. Ed. Porto Alegre: Bookman, 2003.
- Gonçalves, Edson. *Dominando Netbeans*. Rio de Janeiro: Ciência Moderna, 2006.
- Machado, Felipe Nery Rodrigues. *Banco de Dados: Projeto e Implementação*. São Paulo: Érica, 2004.
- Machado, Felipe Nery Rodrigues; Abreu, Mauricio Pereira de. *Projeto de Banco de Dados: Uma Visão Prática*. São Paulo: Érica, 1996.
- Santos, Rui Rossi Dos. *Programando em Java 2: Teoria e Aplicações*. Rio de Janeiro: Axcel Books, 2004.
- Santos, Rui Rossi Dos. *Java na Web: Programando Sites Dinâmicos*. Rio de Janeiro: Axcel Books, 2007.
- Wazlawick, Raul Sidnei. *Análise e Projetos de Sistemas Orientados a Objetos*. Rio de Janeiro: Elsevier, 2004.