

ETEC “FRANCISCO GARCIA”

CURSO TÉCNICO EM INFORMÁTICA

JAQUELINE ZANIN

JOILSON ANTONIO JULIÃO

JONATAS PRADES FERREIRA

NATANAEL MAGALHÃES CARDOSO

**TRABALHO DE CONCLUSÃO DO CURSO DE
INFORMÁTICA**

Software para Automação de Biblioteca da Escola

Alice Autran Dourado

MOCOCA – SP
2012
ETEC “FRANCISCO GARCIA”

CURSO TÉCNICO EM INFORMÁTICA

PROJETO DE INFORMÁTICA 2012
Software para Automação de Biblioteca da Escola
Alice Autran Dourado

Projeto apresentado aos alunos, professores, coordenadores e diretor da ETEC “**Francisco Garcia**”, como requisito final para a Aprovação no Componente Curricular de Trabalho de Conclusão de Curso do Curso Técnico em Informática.

Orientadora do projeto:
Prof. Gustavo Mas

TERMO DE APROVAÇÃO

Projeto do Curso de (Informática) Intitulado **Software para Automação de Biblioteca da Escola Alice Autran Dourado** elaborado pelos **alunos indicados na folha de rosto** e apresentado publicamente perante a Banca Avaliadora, como parte dos requisitos para conclusão do Curso de (Informática) da ETEC “Francisco Garcia”.

Aprovado em ____ de _____ de 2012

Banca Avaliadora:

.....
.....
Prof - Orientador

Gustavo Masquietto

Prof. - Examinador

.....
.....
Prof. - Examinador

.....
.....
Prof. – Examinador

DEDICATÓRIA

Dedicamos este trabalho a nossos pais, que nos apoiaram ao longo do projeto, a nossos professores que estiveram presentes a todo o momento, aos nossos colegas que nos apoio para chegar aonde chegamos e principalmente a Deus que iluminou nosso caminho durante essa jornada.

AGRADECIMENTOS

Agradecemos:

Em primeiro lugar, a Deus, por nos ter guiado e iluminado em cada decisão a ser tomada; - ao nosso orientador Prof.. Gustavo Maschietto, que, com toda paciência e dedicação, nos acompanhou nessa caminhada; - e também a todos os demais professores, pela ajuda e compreensão e - às nossos familiares, que foram as bases de toda nossa formação.

FRASE CIENTÍFICA

Cada sonho que você deixa para trás, é um pedaço do seu futuro que deixa de existir. (Steve Jobs)

RESUMO

Atualmente, tecnologias se tornaram ferramentas necessárias para boa parte da população. O uso da tecnologia para praticamente tudo o que fazemos, envolve uma preocupação, em inseri-las na Educação, no nosso caso em particular bibliotecas. Esse trabalho tem como objetivo, apresentar um considerável melhoramento em relação ao gerenciamento de bibliotecas. E com a finalização do sistema, ficou evidente sua necessidade, principalmente quando ela não possuía nenhum software. É claramente perceptível a agilidade que o sistema traz na automação de uma biblioteca, já que todos os cálculos são feitos automaticamente. Além disso, com a tecnologia utilizada no desenvolvimento do sistema, pode se considerar aspectos de segurança e integridade dos dados, diferente da antiga forma de armazenamento de informações ali contidas com cadastros de novos alunos e alugueis de livros e material de pesquisa.

Palavras-chave: Organização, agilidade.

Sumário

1. Ferramentas.....	15
1.1. MySQL.....	15
1.1.1. Breve História do MySQL.....	15
1.1.2. O Banco de Dados MySQL.....	16
1.1.3. Características do MySQL.....	17
1.1.4. O que o MySQL faz de melhor.....	18
1.1.5. Segurança no MySQL.....	19
1.1.6. O MySQL é gratuito ?.....	19
1.1.7. O MySQL é um banco de dados departamental ?.....	19
1.1.8. Empresas Que o Utilizam:.....	19
1.1.9. Porque a NASA Utiliza o MySQL.....	20
1.1.10. Conclusão MySQL.....	20
1.2. JAVA.....	21
1.2.1. Introdução ao Java.....	21
1.2.2. Histórico da Linguagem.....	21
1.2.3. Características da Linguagem.....	22
1.2.4. Garbage Collection.....	23
1.2.5. Palavras-Chaves.....	23
1.2.6. Por que JAVA?.....	24
1.3. Netbeans.....	25
1.3.1. Introdução ao Netbeans.....	25
1.3.2. História.....	25
1.3.3. A Plataforma NetBeans.....	26
1.4. JDBC.....	26
1.4.1. Introdução JDBC.....	26
1.4.2. JDBC = Java Data-Base Connectivity.....	26
1.4.3. Tipos de Drivers JDBC:.....	27
2. Desenvolvimento.....	28
2.1. Fluxograma.....	28
2.2. Modelo DER.....	29
2.3. Tela de Login.....	30
2.4. Tela principal.....	31
2.5. Acervo.....	32
2.5.1. Gerenciamento do acervo.....	32
2.5.2. Gerenciamento de autores.....	33
2.5.3. Gerenciamento de editoras.....	33
2.5.4. Gerenciamento de classificações literarias.....	33
2.5.5. Gerenciamento de tipos de itens.....	33
2.5.6. Cadastro de itens no acervo.....	34
2.6. Leitores.....	35
2.6.1. Gerenciamento de leitores.....	35

2.6.4. Cadastro de leitores.....	36
2.7. Movimentações.....	37
2.7.1. Cadastro de empréstimo.....	37
2.7.2. Cadastro de devolução.....	38
2.7.3. Cadastro de reservas.....	38
2.7.4. Gerenciamento de empréstimo.....	38
2.7.5. Gerenciamento de reservas.....	39
2.8. Personalização.....	39
2.8.1. Otimização de preenchimento de formulário.....	39
2.8.2. Configuração da interface gráfica.....	40
2.9. Configurações do sistema.....	40
2.9.1. Configuração de acesso ao banco de dados.....	40
2.9.2. Gerenciamento de usuários.....	41
3. Conclusão.....	41
4. BIBLIOGRAFIA:.....	43
5. Cronograma.....	44

LISTA DE FIGURAS

Figura 1 - Logo MySQL.....	15
Figura 2 - Logo Java.....	21
Figura 3 - Logo NetBeans.....	25
Figura 4 - Fluxograma.....	28
Figura 5 - Modelo DER.....	29
Figura 6 - Tela de Login.....	30
Figura 7 - Tela Principal.....	31
Figura 8 - Acervo.....	32
Figura 9 - Cadastro de itens no acervo.....	34
Figura 10 - Gerenciamento de leitores.....	35
Figura 11 - Cadastro de leitores.....	36
Figura 12 - Cadastro de empréstimo.....	37
Figura 13 - Cadastro de devolução.....	38
Figura 14 - Otimização de preenchimento de formulário.....	39
Figura 15 - Configuração de acesso ao banco de dados.....	40

INTRODUÇÃO

Problemática

O gerenciamento da biblioteca era feito de forma manual. Com isso, a biblioteca perdia cerca de três mil livros por ano. Uma quantidade muito elevada e que gerava enormes prejuízos. Segundo a bibliotecária, o principal motivo desta perda é a forma de gerenciamento da biblioteca, e ainda que, por não possuir uma forma concreta de comprovar a dívida, o leitor alegava que já havia devolvido o livro, desta forma, a biblioteca não tinha como recuperar seu acervo.

Hipótese

Diante do exposto, surgiu a necessidade de fazer o controle da biblioteca através de um software. Que, por sua vez, é a maneira mais segura de gerenciar a biblioteca.

Objetivos

O objetivo que a equipe almeja é a criação de um sistema que atenda às necessidades cotidianas de uma biblioteca, auxiliando, assim, os funcionários de uma biblioteca para que a tarefa possa ser concluída da forma mais rápida e eficiente possível.

Justificativa

Em plena era digital, a biblioteca não possuía um software para o seu gerenciamento. Desta forma, ficou por conta da equipe projetar e programar um sistema para informatizar a biblioteca, onde todos os registros eram feitos manualmente, na qual há maior dificuldade e riscos no gerenciamento manual. A ocorrência de erros e perda de informações são alguns exemplos. Por conta disso, a prática do gerenciamento manual está sendo extinta, pois o computador se encarrega de guardar todas as informações de uma maneira segura.

Artigo I. Ferramentas

Seção I.1 MySQL



Figura 1 - Logo MySQL

(a) Breve História do MySQL

O MySQL surgiu a partir da necessidade da equipe que criou o SGBD, de utilizar algum mecanismo que permitisse a conexão de tabelas criadas na linguagem SQL para um determinado fim. A princípio, o grupo iria utilizar o mSQL, mas logo perceberam que esta ferramenta não era rápida o suficiente para atender às necessidades do projeto. O jeito foi criar uma solução própria. Nascia o MySQL.

O MySQL foi criado por Michael Widenius na companhia suíça TcX. Por volta de 1979 Michael desenvolveu um banco de dados chamado UNIREG, sendo rescritos em várias linguagens desde então" [YAR 99]. Em 1994, a empresa TcX começou o desenvolvimento de aplicações baseadas na Web, tendo como base o banco UNIREG, porém esse banco possuía muito "overhead" para obter sucesso em uma aplicação para geração de páginas dinâmicas na Web. Então a empresa TcX começou a procurar por outro banco o mSQL, uma ferramenta baseada em SQL mas com características pobres não possuindo por exemplo suporte a índices, e com desempenho inferior ao UNIREG.

Foi então que o desenvolvedor do banco UNIREG contatou o David Hughes criador do mSQL, para saber do interesse dele em unir os dois bancos. Sendo positivo o interesse de David, a empresa TcX resolveu desenvolver um novo banco, mas

mantendo ao máximo a compatibilidade com mSQL. TcX foi esperta o suficiente para não reinventar o que já estava bem feito, ela construiu seu servidor baseado na estrutura que já estava montada do UNIREG e utilizou grande número de utilitários escritas para mSQL e fez API's para o novo servidor praticamente iguais ao mSQL. Como resultado usuários do mSQL que decidissem mudar para o novo servidor da TcX, teriam apenas que fazer pequenas e simples mudanças nos códigos existentes.

Em maio de 1995 que foi lançada a primeira versão do MySQL

Então foi me maio de 1995 que, definitivamente, a primeira versão do MySQL foi lançada. Um dos parceiros da TcX sugeriu a distribuição do servidor na Internet, o objetivo disso era a utilização de um modelo pioneiro desenvolvido por Aladdin Peter Deutsch. O resultado foi um maior flexibilidade em sem "copyright", que fez do MySQL mais difundido gratuitamente do que mSQL.

(b) O Banco de Dados MySQL

O MySQL foi desenvolvido pela TCX em 1996. Atualmente a MySQL AB desenvolve o programa. MySQL AB é a companhia dos fundadores e principais desenvolvedores do MySQL. Eles criaram-no porque precisavam de um banco de dados relacional que pudesse tratar grandes quantidades de dados em máquinas de custo relativamente barato. O MYSQL é um dos bancos de dados relacionais mais rápidos do mercado, apresenta quase todas as funcionalidades dos grandes bancos de dados . MySQL é uma linguagem simples, em que você facilmente pode gravar, alterar e recuperar informações num web site com segurança e rapidez O MYSQL é executado, principalmente, em sistemas que participam da filosofia UNIX, embora outros sistemas S.O também fornecem suporte, como Windows, por exemplo.

O MYSQL é um sistema de gerenciamento de banco de dados relacional multiencadeado, de código fonte aberto e nível corporativo. O MySQL não é apenas um banco de dados, mas sim um gerenciador de banco de dados. Com este SGBD (Sistema Gerenciador de Banco de Dados), também pode ser utilizado para aplicações corporativas, o qual, necessitam de varias conexões simultâneas, que possibilita 101

conexões simultâneas. Uma conexão é o tempo que leva para o usuário receber o dado solicitado.

MySQL é a solução robusta para quase todo tipo de aplicação, combine a estabilidade do MySQL com seu baixo custo de propriedade e rapidamente você irá considerá-lo indispensável. O MySQL oferece o melhor cenário de todos SGBD, executa em muitas plataformas, oferece um baixo TCO (custo total de propriedade) e é muito estável .

O MySQL é um sistema de gerenciamento de bancos de dados relacional.

Um banco de dados relacional armazena dados em tabelas separadas em vez de colocar todos os dados um só local. Isso proporciona velocidade e flexibilidade.

O Servidor MySQL foi desenvolvido originalmente para lidar com bancos de dados muito grandes de maneira muito mais rápida que as soluções existentes e tem sido usado em ambientes de produção de alta demanda por diversos anos de maneira bem sucedida. Apesar de estar em constante desenvolvimento, o Servidor MySQL oferece hoje um rico e proveitoso conjunto de funções. A conectividade, velocidade, e segurança fazem com que o MySQL seja altamente adaptável para acessar bancos de dados na Internet.

O Programa de Banco de Dados MySQL é um sistema cliente/servidor que consiste de um servidor SQL multitarefa que suporta acessos diferentes, diversos programas clientes e bibliotecas, ferramentas administrativas e diversas interfaces de programação (API's). Também concedemos o Servidor MySQL como uma biblioteca multitarefa que você pode ligar à sua aplicação para chegar a um produto mais rápido, menor e mais facilmente gerenciável.

(c) Características do MySQL

É um software multiplataforma, tem suporte a múltiplos processadores, possui um sofisticado sistema de senhas criptografadas, as tabelas criadas podem possuir até 4GB de tamanho, possibilita conexões através do protocolo TCP/IP, suporta até 16 índices por tabela, por ser um SGBD robusto, o MySQL possibilita a manipulação de banco de dados com até 50 milhões de registros, no entanto, sua interface é de fácil manipulação e reduz gastos com técnicos. Seu código foi escrito em C/C++ e testada com vários compiladores. Além disso, o MySQL tem suporte à diversas API, o que possibilita a conexão com o Java, por exemplo.

(d) O que o MySQL faz de melhor

O MySQL se adapta à qualquer tipo de projeto, seja grandes ou pequenos. Este SGBD é muito reconhecido em aplicações WEB e de nível corporativo. Por possuir uma grande estabilidade e gerenciamento de grande fluxo de informações. Além disso, o MySQL é um software de código aberto, e isso reduz custos.

(e) Segurança no MySQL

O MySQL possui componentes de segurança contra ameaças externas como crackers e outros, e também proteger os dados dos próprios usuários. O mysql apresenta vários níveis de segurança em relação ao acesso. Todas as informações de segurança estão armazenadas no banco MySQL.

A filosofia de segurança em banco de dados refere-se a fornecer ao usuário apenas o que é essencial para o seu trabalho.

(f) O MySQL é gratuito ?

Pessoas confundem "free" com "grátis" o que é comum aqui no Brasil. Mas em se tratando de software este "free" é de open source e não gratuito. Para poder utilizar o MySQL sob a licença GPL e não precisar pagar, o produto desenvolvido precisa ser GPL também, senão, orientamos a compra da licença comercial, com baixo custo, sendo comercializada por servidor, sem limites de usuários e processadores e ainda com garantia perpétua de atualização de versão para o resto da vida.

(g) O MySQL é um banco de dados departamental ?

O MySQL é regido pela licença GPL, portanto é open source. Muitas e analisarmos o investimento necessário para aquisição do mesmo, podemos dizer que cabe como uma solução departamental. Entretanto o que muitos assustam é que empresas como Nasa, Cisco, Telefônica da Suécia, Yahoo Finance, Nokia e outras mais tem utilizado o MySQL em aplicações de missão crítica.

(h) Porque a NASA Utiliza o MySQL?

Custo: Porque o custo total de MySQL é limitado ao custo da sustentação técnica, dado que MySQL está disponível para o download e o uso livres em a maioria de casos.

Sustentação: devido à comunidade grande do colaborador que pode ser encontrada no Internet. Embora a NASA usasse a sustentação direta dos colaboradores de MySQL infreqüentemente, indicou que "eram muito úteis e responsivos quando needed". Além disso, colaborador John Sudderth de NAIS indicado em um artigo que discute o interruptor que o custo para a sustentação oficial era aproximadamente 1 por cento da despesa técnica da sustentação para o oracle .

Compatibilidade: MySQL pode facilmente conectar com a maioria de aplicações SQL-sQL-compliant com ODBC.

Talvez uma quarta vantagem a fazer o interruptor a MySQL podia ser atribuída ao desempenho. "nós observamos um aumento [velocidade de] no desempenho desde que a mudança e não experimentamos nenhuns problemas com o produto", não dizemos os sistemas computadorizados analista de NAIS e o líder de projeto Dwight Clark indicado em um artigo para a semana federal do computador

(i) Conclusão MySQL

É incrível como um produto como o SGBD MySQL exerce tão bem sua função e ainda é gratuito, superando assim a concorrência, talvez por esse mesmo motivo de ele ser open source, e que todo desenvolvedor possa dar seu toque ao sistema é o que o torna tão eficiente. Este documento resume um pouco a vida desse notório produto que só tende a crescer mais e mais, e que nós estaremos acompanhado e ajudando em sua transição para a excelência.

Seção I.2 JAVA

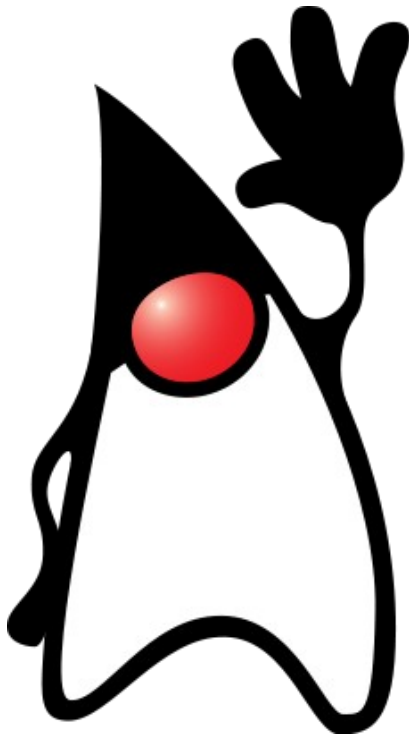


Figura 2 - Logo Java

(a) Introdução ao Java

A linguagem Java foi desenvolvida pela Sun Microsystems em 1995. Apesar de relativamente nova, a linguagem obteve uma espetacular aceitação por programadores do mundo inteiro, tendo se difundido como nunca antes ocorreu com uma linguagem de programação. Um fator que colaborou com isso, é o fato da linguagem possuir vantagens agregadas tais como: orientação a objetos, independência de plataforma, multitarefa, robusta, segura e distribuída. Com o advento da Internet, que tornou ainda mais necessário a comunicação entre plataformas heterogêneas, estes fatores fizeram com que o Java fosse a tecnologia perfeita para este novo cenário.

(b) Histórico da Linguagem

No início da década de 90, um pequeno grupo de projeto da Sun pretendia criar uma nova geração de computadores portáteis inteligentes, que pudessem se comunicar entre si de diversas formas. Para isso, decidiu-se criar uma plataforma de desenvolvimento onde o software pudesse ser executado em diversos tipos de

equipamentos. Para o desenvolvimento desta plataforma foi escolhida a linguagem de programação C++.

A linguagem C++ não permitia realizar com facilidade tudo o que o grupo pretendia. Neste ponto, James Gosling, coordenador do projeto, decidiu criar uma nova linguagem de programação que pudesse atendê-los em suas necessidades. A criação dos chips inteligentes foi abandonada devido ao alto custo de produção. Posteriormente, a explosão da Internet no mundo todo fez surgir a necessidade de uma tecnologia onde computadores de diferentes plataformas pudessem conversar. Surge daí, baseada na linguagem criada para o projeto de Gosling, a linguagem Java.

(c) Características da Linguagem

- Orientada a Objetos : Paradigma atual mais utilizado na construção de softwares. Dentre suas vantagens, podemos citar reaproveitamento de código e aumento da manutenibilidade dos sistemas assim desenvolvidos.
- Simples e Robusta : Java representa em muitos aspectos um aperfeiçoamento da linguagem C++. Ela possui certas características que permitem a criação de programas de forma mais rápida, pois tiram do programador a possibilidade de cometer erros que são comuns de ocorrer em C++. Algumas dessas características são o tratamento obrigatório de exceções e o gerenciamento automático de memória.
- Gerenciamento Automático de Memória : Em Java não existe ponteiros, isto é, não é permitido ao programador acessar explicitamente uma posição de memória. Java automaticamente gerencia o processo de alocação e liberação de memória, ficando o programador livre desta atividade. O mecanismo responsável pela liberação de memória que não está mais sendo utilizada é conhecido como Garbage Collector.
- Independência de Plataforma : Um dos elementos chave da linguagem Java é a independência de plataforma. Um programa Java escrito em uma plataforma pode ser utilizado em uma outra distinta da original. Este aspecto da linguagem é geralmente referenciado como “write once, run anywhere”. Isto é conseguido através da utilização da Java Virtual Machine (JVM) a qual roda numa plataforma específica e interpreta um programa Java

para código de máquina específico da plataforma em questão. Como os programas em Java executam sob o controle da JVM, eles podem rodar em qualquer plataforma que possua uma disponível.

(d) Garbage Collection

No C e C++ (e em outras linguagens) o programa desenvolvido é responsável pela alocação e desalocação da memória, o que é um dos principais pontos causadores de erros nestes programas. Em Java, quando um objeto é criado (e quase tudo em Java é um objeto), a memória necessária é alocada automaticamente para ele, pois não há forma de se manipular a memória diretamente. Podem existir inúmeras variáveis no programa referenciando um mesmo objeto e, durante o ciclo de execução do programa, o Java verifica se um objeto ainda está sendo referenciado por alguma variável e, caso não esteja, o Java libera automaticamente esta área que não está sendo utilizada. Ou seja, quando não resta mais nenhuma referência para um objeto, ele é marcado para ser coletado pelo garbage collector ou “coletor de lixo”, que libera a memória ocupada por ele.

O coletor de lixo é executado de tempos em tempos num processo de baixa prioridade. Quando a JVM não está realizando nenhum processamento, ela executa o coletor de lixo que vasculha a memória em busca de algum objeto criado e não referenciado.

Quando uma grande quantidade de objetos ou objetos muito grandes não são mais necessários e não se quer esperar até que o coletor de lixo seja executado para liberar essa memória, pode-se chama-lo explicitamente no programa da seguinte forma:

```
System.gc();
```

(e) Palavras-Chaves

Toda linguagem tem um grupo de palavras que o compilador reserva para seu próprio uso. Essas palavras-chaves não podem ser usadas como identificadores em seus programas. Em Java, as palavras reservadas são:

Abstract	do	implements	package	throw
boolean	double	import	private	throws

break	else	*inner	protected	transient
byte	extends	instanceof	public	try
case	final	int	*rest	*var
*cast	finally	interface	return	void
*catch	float	long	short	volatile
Char	for	native	static	while
Class	*future	new	super	
*const	generic	null	switch	
Continue	*goto	operator	synchronized	
Default	if	*outer	this	

(f) Por que escolhemos o JAVA?

Acreditamos que as principais vantagens em java é a portabilidade e um maior controle sobre o ambiente de desenvolvimento. Por ser uma linguagem de maquina virtual, é bem fácil levar o programa de um sistema operacional para outro. Além disso, Java oferece um ambiente de desenvolvimento no qual a maquina virtual se encarregue de diversas atividades e evita erros que o programador pode vir a cometer. E também por ser uma ferramenta gratuita.

Seção I.3 Netbeans



Figura 3 - Logo NetBeans

(a) Introdução ao Netbeans

O **NetBeans** IDE é um ambiente de desenvolvimento integrado (IDE) gratuito e decódigo aberto para desenvolvedores de software nas linguagens Java, C, C++, PHP, Groovy, Ruby, entre outras. O IDE é executado em muitas plataformas, como Windows, Linux, Solaris e MacOS. O NetBeans IDE oferece aos desenvolvedores ferramentas necessárias para criar aplicativos profissionais de desktop, empresariais, Web e móveis multiplataformas.

(b) História

O NetBeans foi iniciado em 1996 por dois estudantes tchecos na Universidade de Charles, em Praga, quando a linguagem de programação Java ainda não era tão popular como atualmente. Primeiramente o nome do projeto era Xelfi, em alusão ao Delphi, pois a pretensão deste projeto era ter funcionalidades semelhantes aos IDEs então populares do Delphi que eram mais atrativas por serem ferramentas visuais e mais faceis de usar, porém com o intuito de ser totalmente desenvolvido em Java.

Em 1999 o projeto já havia evoluído para uma IDE proprietário, com o nome de NetBeans DeveloperX2, nome que veio da idéia de reutilização de componentes que era a base do Java. Nessa época a empresa Sun Microsystems havia desistido de sua IDE Java Workshop e, procurando por novas iniciativas, adquiriu o projeto NetBeans DeveloperX2 incorporando-o a sua linha de softwares.

Por alguns meses a Sun mudou o nome do projeto para Forte for Java e o manteve por um bom tempo como software proprietário, porém, em junho de 2000 a

Sun disponibilizou o código fonte do IDE NetBeans tornando-o uma plataforma OpenSource. Desde então a comunidade de desenvolvedores que utilizam e contribuem com o projeto não parou de crescer, tornando-se uma das IDEs mais populares atualmente.

(c) A Plataforma NetBeans

A construção de uma aplicação desktop em cima da plataforma NetBeans pode economizar anos de tempo de desenvolvimento.

A Plataforma NetBeans é uma estrutura genérica para aplicações Swing. Ele fornece o "encanamento" que, antes, cada desenvolvedor tinha que escrever-se economia de estado, conectando ações para itens de menu, barra de ferramentas e itens atalhos de teclado; gerenciamento de janelas, e assim por diante.

A Plataforma NetBeans fornece todos estes para fora da caixa. Você não precisa codificar manualmente estas ou outras características básicas, você mesmo, nunca mais. Veja o que alguns aplicativos baseados NetBeans parecer . A plataforma não acrescenta muita sobrecarga para a sua aplicação - mas ele pode salvar uma enorme quantidade de tempo e trabalho.

A Plataforma NetBeans fornece uma arquitetura de aplicativo confiável e flexível. O aplicativo não tem que olhar qualquer coisa como um IDE . Ele pode te salvar anos de tempo de desenvolvimento. A Plataforma NetBeans dá-lhe uma arquitetura testada pelo tempo de graça. Uma arquitetura que incentiva práticas de desenvolvimento sustentável. Como a arquitetura da plataforma NetBeans é modular, é fácil criar aplicações que são robustos e extensível.

Seção I.4 JDBC

(a) Introdução JDBC

Diferentes bancos de dados relacionais possuem diferentes formas de se comunicar com uma aplicação que necessite acessar os seus dados.

Isto causa um grande problema de codificação e manutenção nas aplicações que precisam trabalhar com diversos bancos de dados e também requer o aprendizado de uma nova API para cada BD diferente.

Isso não torna a aplicação flexível.

A Sun desenvolveu a API JDBC, com a intenção de uniformizar os acessos aos Diferentes bancos de dados relacionais, dando maior flexibilidade aos sistemas.

(b) JDBC (Java Data-Base Connectivity).

Com JDBC as chamadas ao BD são padronizadas, apesar de que os comandos SQL podem variar de banco para banco, se não for usado o SQL padrão. A biblioteca da JDBC provê um conjunto de interfaces de acesso ao BD. Uma implementação em particular dessas interfaces é chamada de driver. Os próprios fabricantes dos bancos de dados (ou terceiros) são quem implementam os drivers JDBC para cada BD, pois são eles que conhecem detalhes dos BDs. Cada BD possui um Driver JDBC específico (que é usado de forma padrão - JDBC). A API padrão do Java já vem com o driver JDBC-ODBC, que é uma ponte entre a aplicação Java e o banco através da configuração de um recurso ODBC na máquina. Os drivers de outros fornecedores devem ser adicionados ao CLASSPATH da aplicação para poderem ser usados. Desta maneira, pode-se mudar o driver e a aplicação não muda.

(c) Tipos de Drivers JDBC

Tipo 1 - Driver Ponte JDBC-ODBC

É uma implementação nativa que conecta uma aplicação Java a um banco de dados através de ODBC configurado na máquina.

Tipo 2 - Driver API-Nativa Parcialmente Java

É uma “casca” sobre uma implementação nativa de um driver de acesso ao Banco (ex: este driver utiliza o OCI para Oracle). Um erro neste driver nativo pode derrubar a JVM.

Tipo 3 - Driver Java c/ Net-Protocol

Utiliza um middleware para a conexão com o banco de dados.

Tipo 4 - Driver Java Puro

Driver totalmente implementado em Java. Conhece todo o protocolo de Comunicação com o BD e pode acessar o BD sem software extra. É o tipo de driver mais indicado.

Desenvolvimento

Introdução

Buscando o planejamento de um sistema o mais completo possível, houve a necessidade de criar um sistema que tenha funções avançadas e que, também, possuisse uma interface gráfica intuitiva.

A distribuição das tarefas foi essencial para que o projeto pudesse ser concluído da maneira mais eficiente possível. Abaixo está exposto um fluxograma desenvolvido pela equipe, que ilustra os passos feitos pela equipe para a execução do projeto.

O design do banco de dados foi projetado para atender a demanda da biblioteca da melhor maneira possível, sem deixar que fatores essenciais fossem esquecidos. As principais tabelas são `Item` e `Leitor`, e a relação entre elas formam as tabelas `Empréstimo` e `Reserva`. Além disso, existem outras tabelas, que dão funcionalidades mais específicas ao sistema, como, por exemplo, a tabela `Cep` que se encarrega de preencher automaticamente o estado e a cidade do leitor após a inserção do cep.

A parte de programação abrange desde funções essenciais, como o cadastro de leitores e de obras, o gerenciamento dos mesmos, o controle das movimentações da biblioteca e a geração de relatórios. Além dessas funcionalidades, o sistema possui um otimizador de preenchimento de formulários, que agiliza o processo de cadastro, o usuário pode escolher os efeitos visuais em que mais agrada, pode escolher uma foto de perfil. A conexão com o banco é configurada pelo próprio sistema através de uma interface visual. Isso facilita o processo de migração do banco.

O sistema ainda possui um algoritmo que é carregado na inicialização do sistema e que se encarrega de verificar se todas as configurações necessárias para o funcionamento do sistema estão disponíveis. Caso não esteja, o software se encarrega de realizar a reparação de erros.

Por exemplo, se uma máquina possui o JRE – máquina virtual do Java – e o MySQL instalado, a única coisa que precisa ser feita é dar um clique duplo com no arquivo executável do Java (.JAR), que o sistema se encarrega de fazer o resto.

Seção I.5 Fluxograma

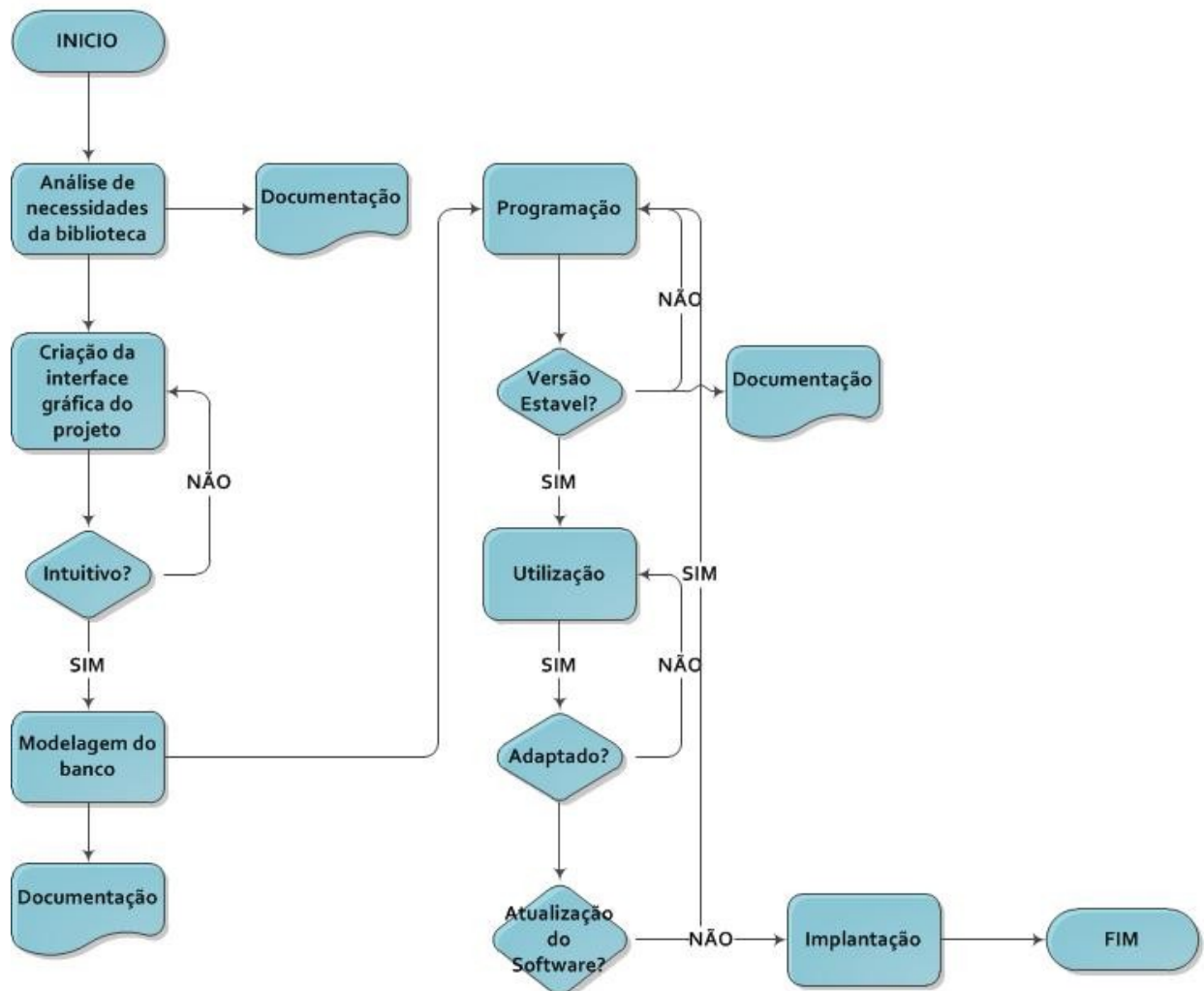


Figura 4 - Fluxograma

Seção I.6 Modelo DER

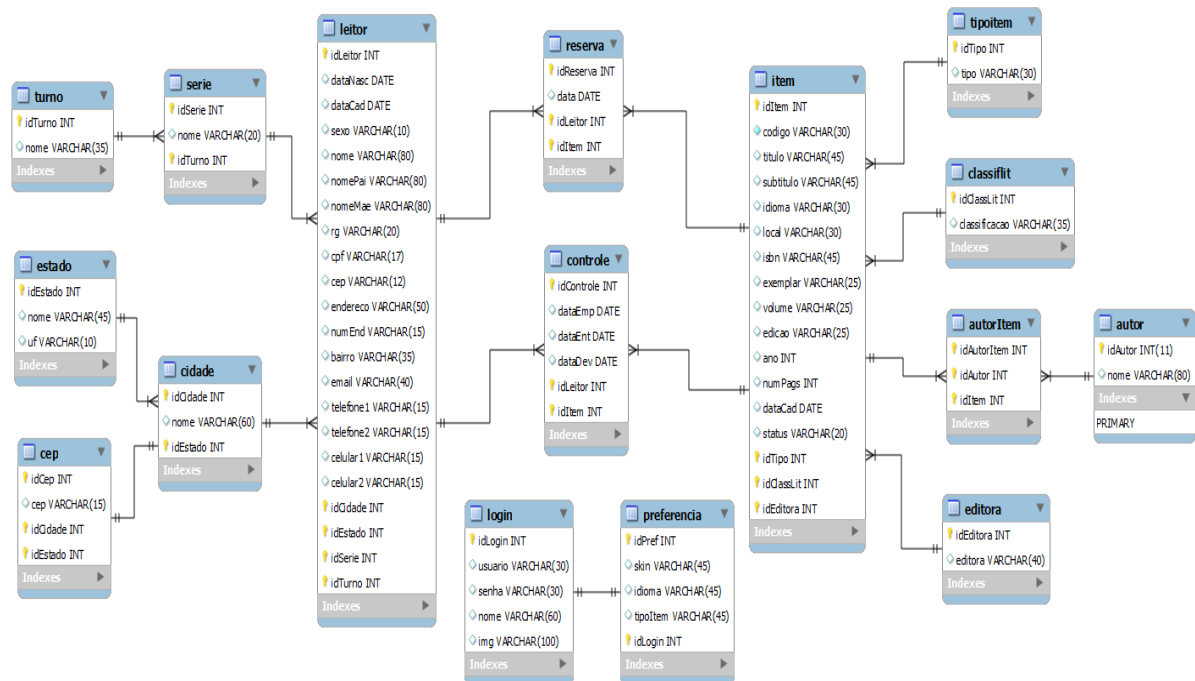
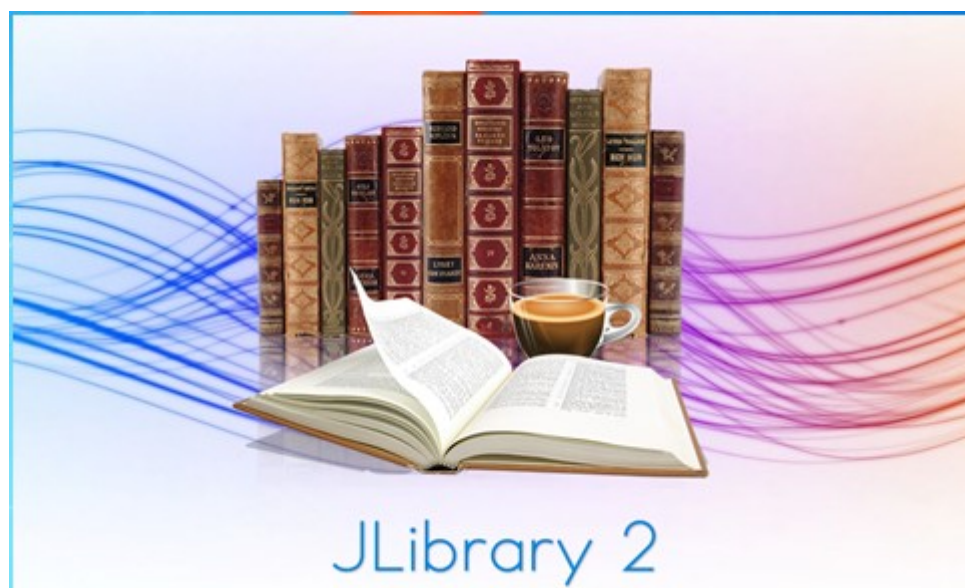


Figura 5 - Modelo DER

Tela de Splash



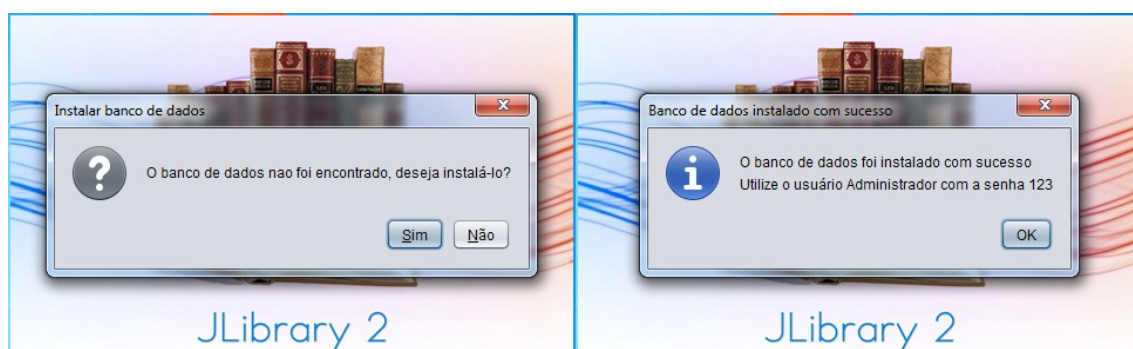
A tela de Splash, além de apresentar o sistema, tem como objetivo a busca por erros de conexão com banco de dados e o carregamento prévio da interface gráfica do formulário principal. Se não for encontrado nenhum erro, seu tempo de execução pode variar entre um e dois segundos. Alguns dos erros verificados pelo algoritmo são:

Dados incorretos de conexão com o banco



Ao tentar fazer a conexão com o MySQL e não conseguir, o algoritmo exibe uma mensagem de erro perguntando se o usuário deseja configurar o acesso ao banco de dados (quadro da esquerda). Se a resposta for positiva, abrirá uma janela para que a configuração possa ser feita (quadro da direita).

Banco de dados inexistente



Caso o banco de dados não seja detectado, o sistema exibe uma mensagem informando este erro e pergunta se deseja instalá-lo (quadro da esquerda). Se a resposta for positiva, o software se encarrega de executar o algoritmo que instala o banco de dados. Após a instalação, é informado ao usuário o login e a senha do administrador (quadro da direita).

Tela de Login

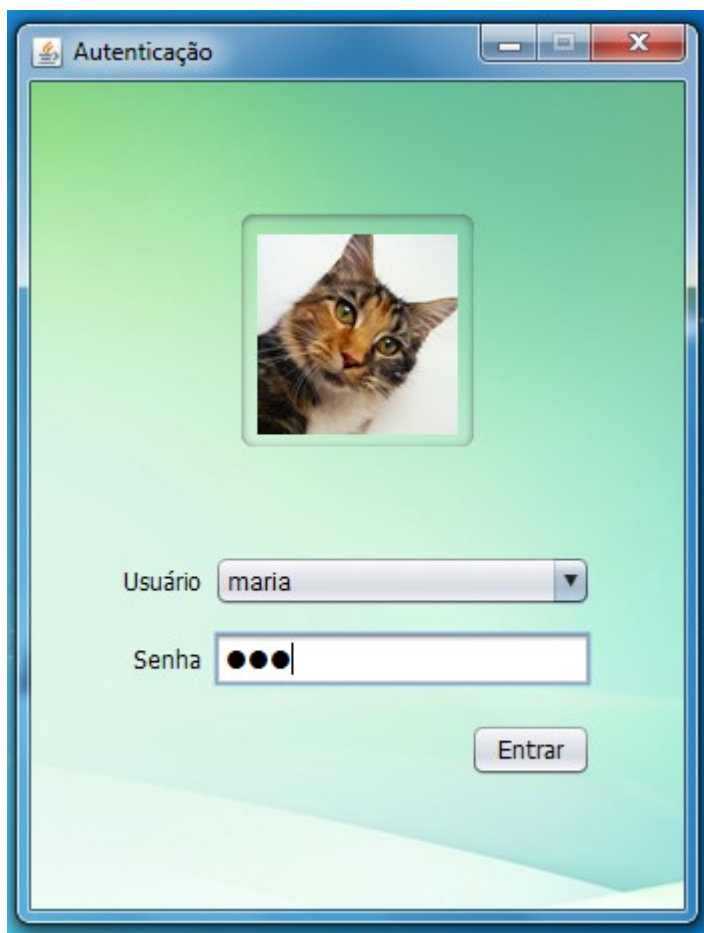


Figura 6 - Tela de Login

Cada funcionário da biblioteca tem seu nome de usuário e senha. As preferencias de cada usuário são salvas e carregadas após a inicialização do sistema

Seção I.7 Tela principal

O sistema possui a interface intuitiva, na qual as janelas mais importantes estão localizadas na janela principal.

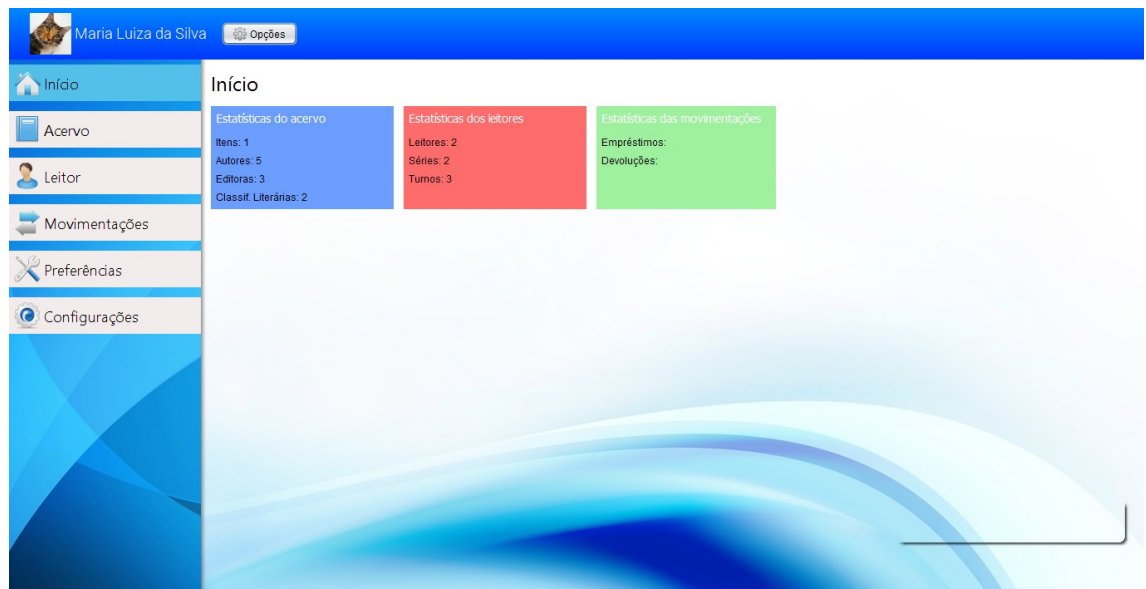
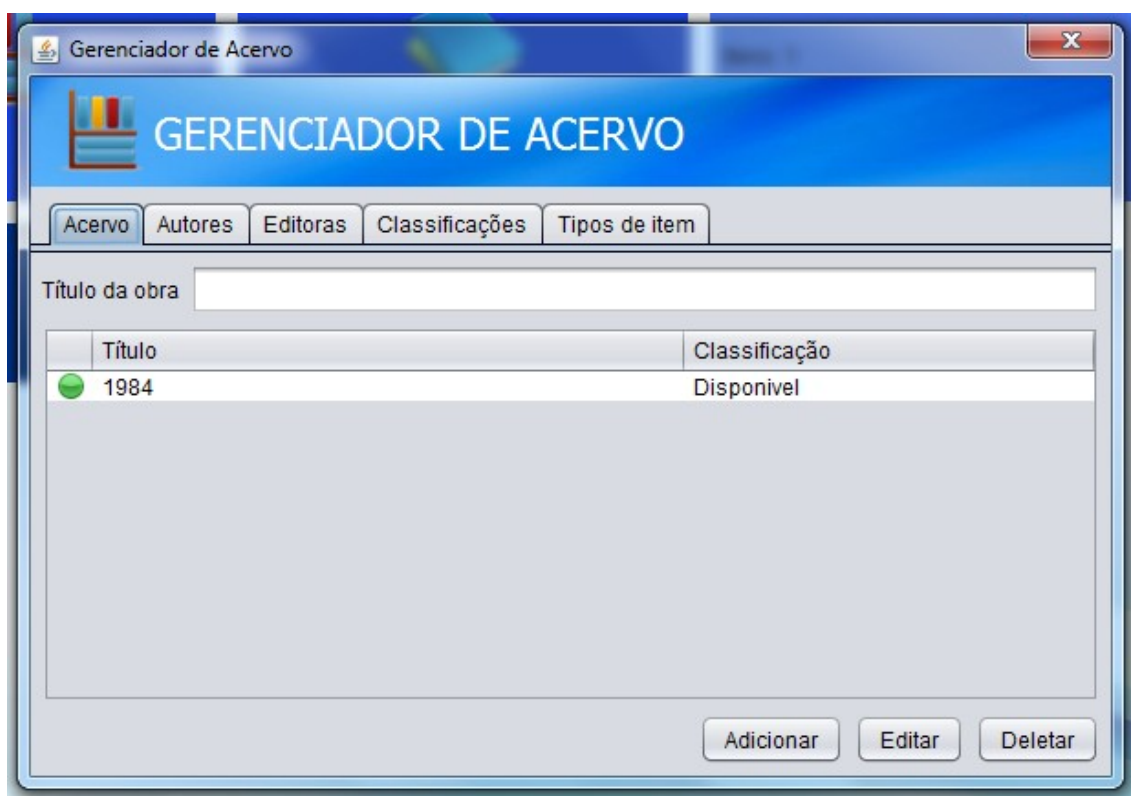


Figura 7 - Tela Principal

A tela principal é o resumo das demais funcionalidades do sistema.

A interface foi criada para ser o mais o possível. A disposição do menu fica à esquerda, o nome e a imagem de perfil do usuário do sistema se localiza ao topo, juntamente com as opções, tais como: trocar de usuário, configurar a conta, ou sair do sistema. O menu foi dividido em seis categorias, sendo que cada menu abre, na própria tela principal, todos os subitens de cada categoria.

Seção I.8 **Acervo****Figura 8 – Acervo****(a) Gerenciamento do acervo**

Gerenciar o acervo evita qualquer forma de ambiguidade entre termos, seja por erro de digitação ou, até mesmo, de acentuação. Além disso, esta ferramenta possibilita verificar a quantidade de algo que está contido no acervo.

O gerenciamento do acervo auxilia no momento do cadastro de itens, pois, ao invés do funcionário da biblioteca digitar o nome do autor – ou da editora, da classificação literária, entre outros – ele irá apenas selecionar um termo cadastrado no banco de dados do sistema. Caso não tenha nenhum registro cadastrado, o usuário poderá cadastrar rapidamente um registro no momento da inserção de um item no acervo. Desta forma, uma vez que um termo é cadastrado, ele fica disponível para todos os seguintes cadastros de itens no acervo. Todos os registros cadastrados podem ser editados e deletados.

No acervo existem alguns subitens que podem ser gerenciados, tais como:

(b) Gerenciamento de autores

São armazenados todos os autores que publicaram os itens da biblioteca

(c) Gerenciamento de editoras

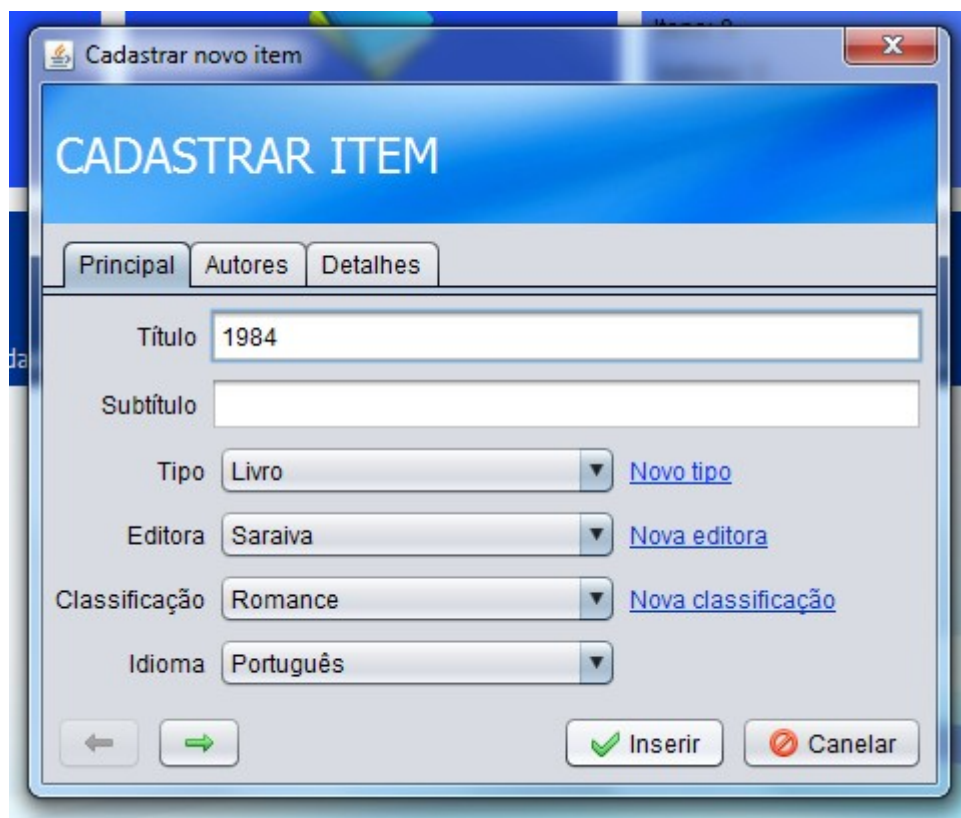
São armazenadas todas as editoras que vende os itens presentes no acervo

(d) Gerenciamento de classificações literárias

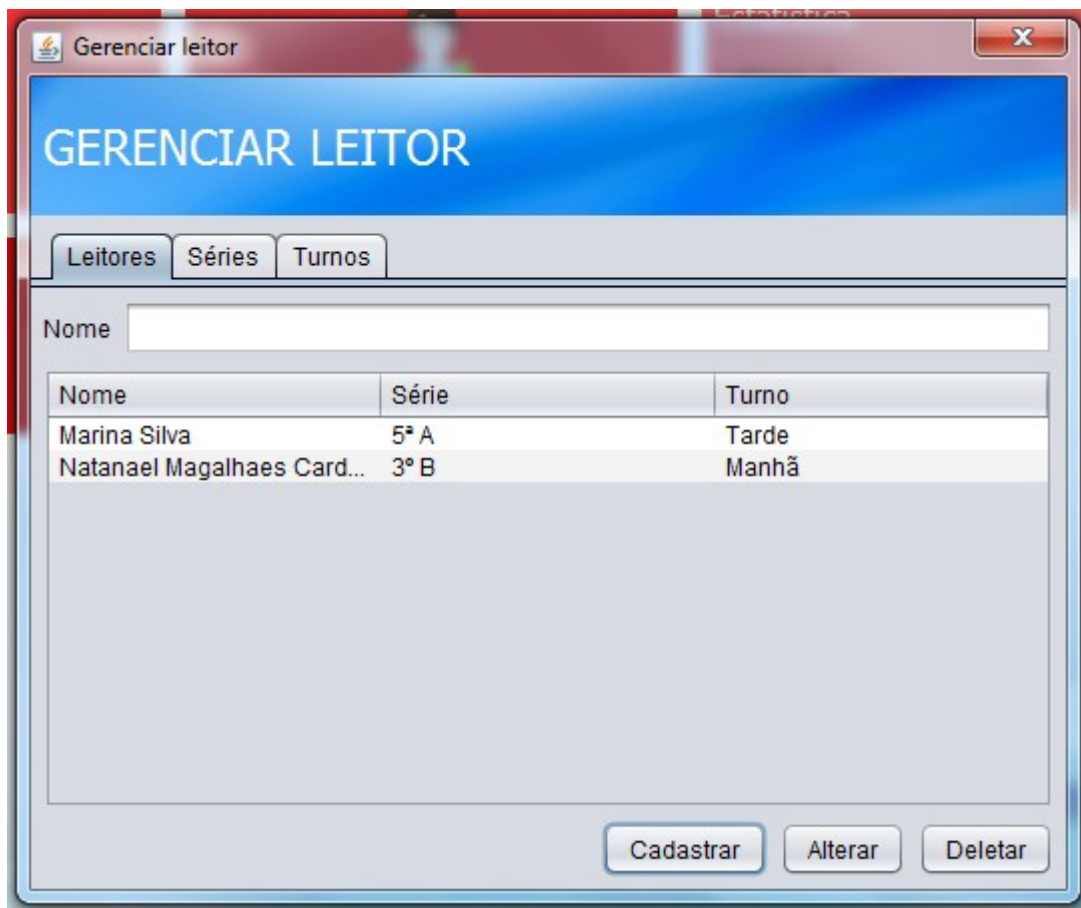
São armazenadas todas as classificações literárias pertencentes aos itens presentes no acervo, tais como romance, realismo, entre outros.

(e) Gerenciamento de tipos de itens

São armazenados todos os tipos de itens do acervo, tais como livro, revista, dvd, entre outros.

(f) Cadastro de itens no acervo**Figura 9 - Cadastro de itens no acervo**

O cadastro de itens no acervo é uma das principais funcionalidades do sistema. É aqui que será cadastrado todos os itens pertencentes à biblioteca.

Seção I.9 Leitores**(a) Gerenciamento de leitores**

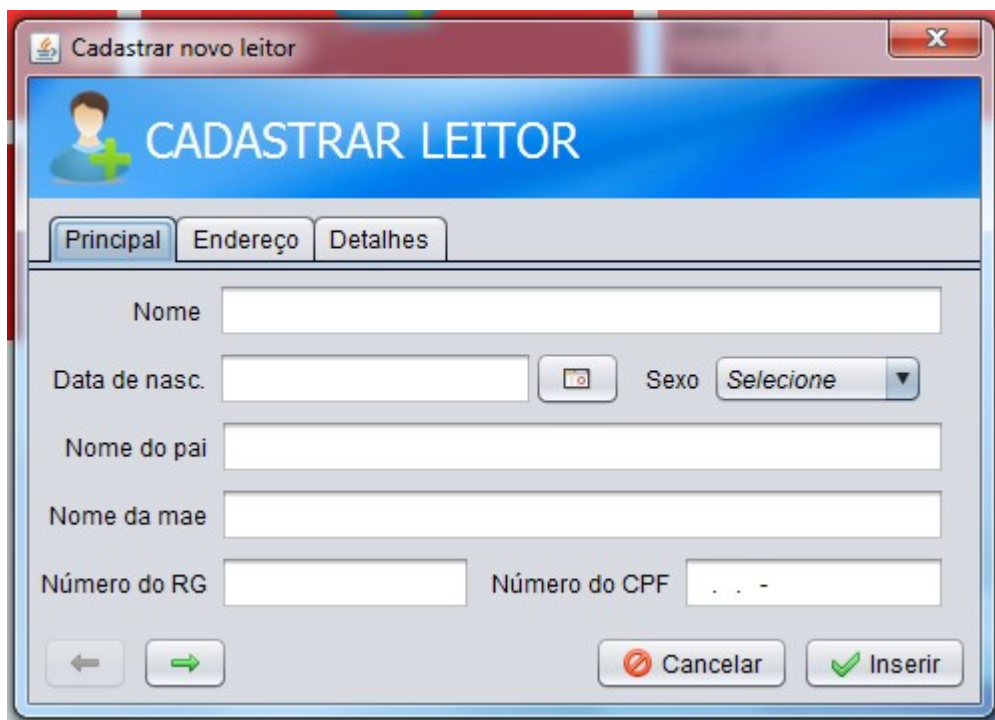
Nome	Série	Turno
Marina Silva	5ª A	Tarde
Natanael Magalhaes Card...	3º B	Manhã

Figura 10 - Gerenciamento de leitores**2.5.1. Gerenciamento de séries**

São armazenadas todas as séries da escola. Além disso, são calculados quantos alunos cadastrados no sistema possui em uma determinada série.

2.5.2. Gerenciamento de turnos

São armazenados todos os turnos de funcionamento da escola. Além disso, o gerenciador calcula quantas séries tem em cada turno e, ainda, a quantidade de alunos cadastrados no sistema que estudam em um determinado turno.


(b) Cadastro de leitores

Cadastrar novo leitor

CADASTRAR LEITOR

Principal Endereço Detalhes

Nome

Data de nasc.  Sexo Selecione ▼

Nome do pai

Nome da mae

Número do RG Número do CPF





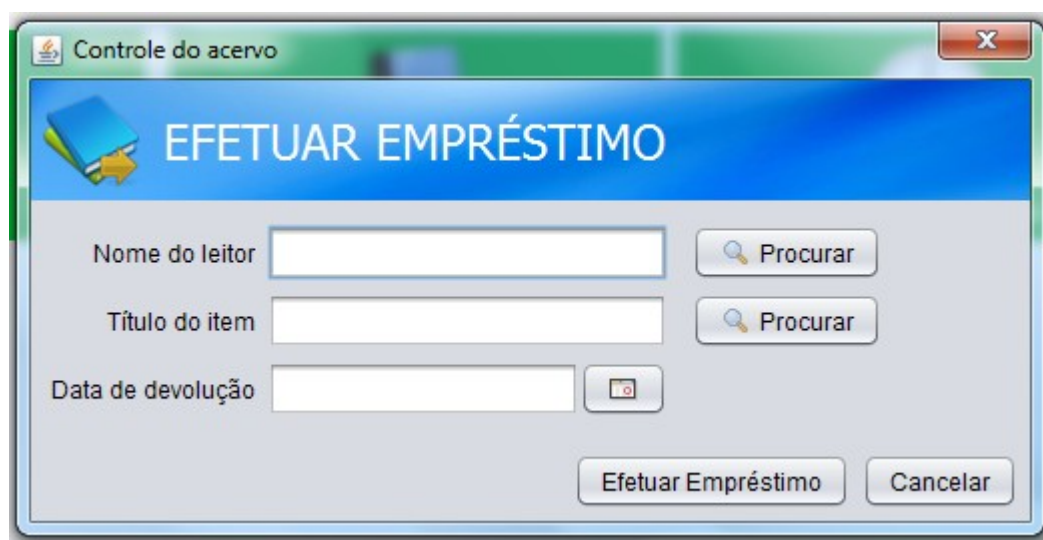
   Cancelar  Inserir

Figura 11 - Cadastro de leitores

Esta parte do sistema é responsável por armazenar as informações de todos os leitores cadastrados na biblioteca.

Seção I.10 Movimentações

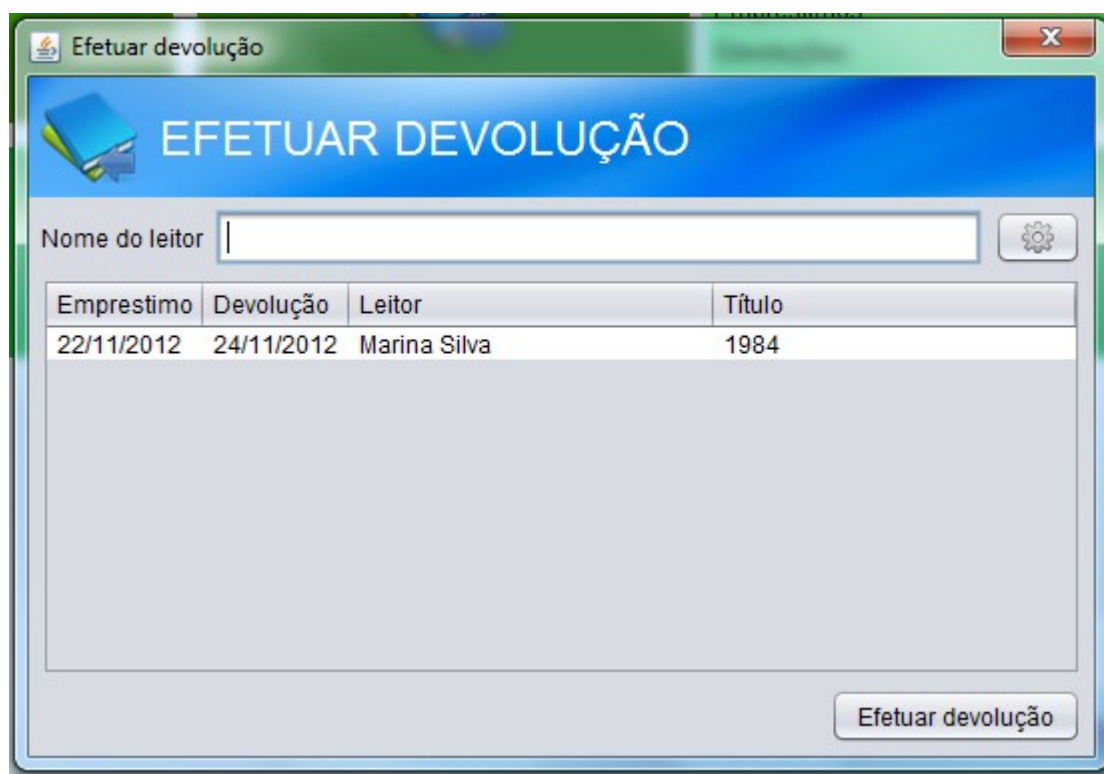
(a) Cadastro de empréstimo



A imagem mostra uma janela de software intitulada "Controle do acervo" com o botão de fechar "X" no canto superior direito. O título da janela é "EFETUAR EMPRÉSTIMO", acompanhado de um ícone de livros. O formulário contém três campos de entrada: "Nome do leitor", "Título do item" e "Data de devolução". Cada um dos primeiros dois campos possui um botão "Procurar" com um ícone de lupa. O campo "Data de devolução" possui um ícone de calendário. Na base da janela, há dois botões: "Efetuar Empréstimo" e "Cancelar".

Figura 12 - Cadastro de empréstimo

Esta janela tem função de capturar o nome do leitor, o título do item e a data de devolução. Depois disso, ele armazena um novo movimento no acervo da biblioteca. Além disso, se algum item estiver sendo emprestado e outra pessoa já tiver reservado ele, o sistema avisa ao usuário que o item já está reservado.

(b) Cadastro de devolução

Emprestimo	Devolução	Leitor	Título
22/11/2012	24/11/2012	Marina Silva	1984

Figura 13 - Cadastro de devolução

Após devolver um item, ele, automaticamente, fica disponível para o empréstimo novamente.

(c) Cadastro de reservas

Para adicionar uma nova reserva na lista de espera, basta o usuário selecionar o leitor e a obra desejada, que o sistema já organiza a lista de espera automaticamente, levando em consideração a data que a reserva foi efetuada.

(d) Gerenciamento de empréstimo

Aqui é possível administrar todos os livros acessados, o sistema mostra a data de empréstimo e a data da devolução, além disso, o usuário pode obter informações detalhadas tanto do leitor quanto da obra emprestada.

(e) Gerenciamento de reservas

Para gerenciar uma fila de espera por um determinado item no acervo, o sistema possui a ferramenta de gerenciamento de reservas, para que o cadastro de uma nova reserva, ou visualização do andamento de seu andamento possa ser organizado de uma melhor forma.

Seção I.11 Personalização

(a) Otimização de preenchimento de formulário

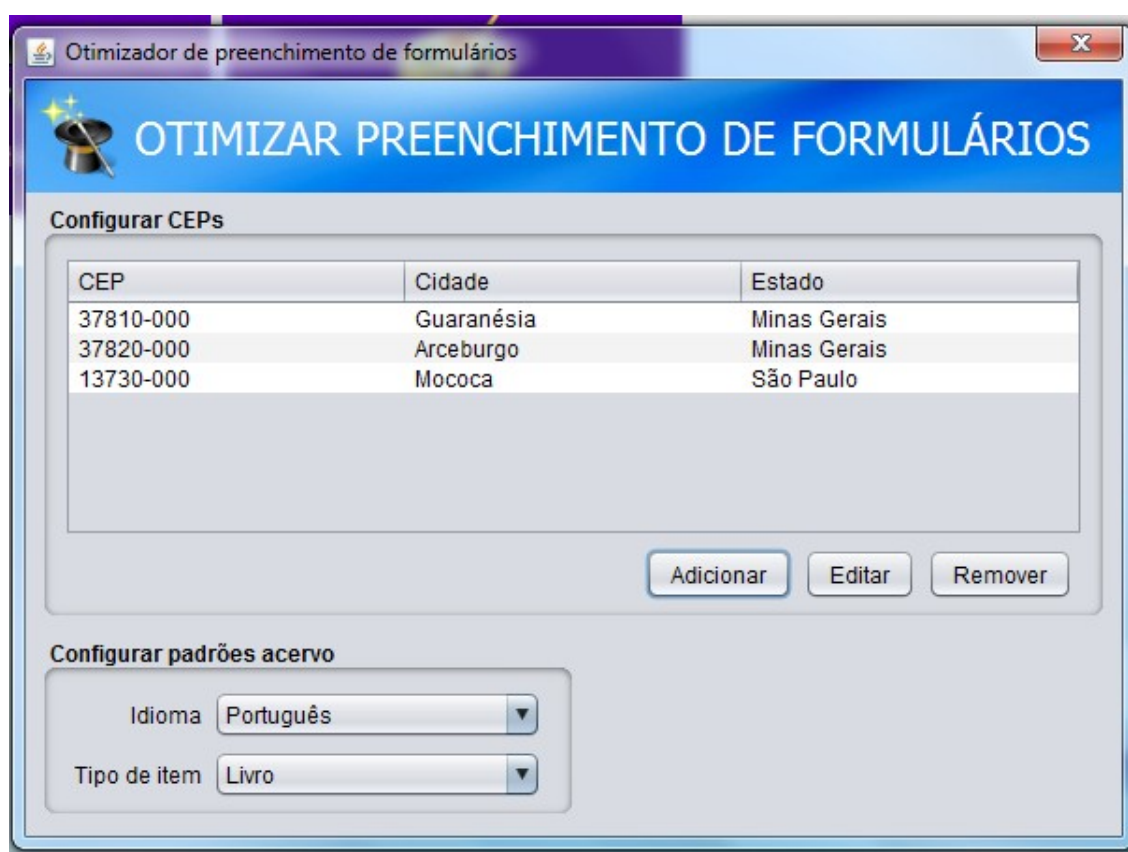


Figura 14 - Otimização de preenchimento de formulário

Para agilizar o processo de preenchimento de formulário, o sistema possui a ferramenta de otimização, que alimenta o banco de dados com informações que são carregadas no momento que o formulário é aberto. Esta ferramenta guarda os CEP's mais utilizados pelo usuário e, no momento do preenchimento do formulário de cadastro de leitores, quando a lacuna do CEP é preenchida, o sistema preenche

automaticamente o estado e a cidade, além de posicionar o cursor para a digitação do endereço. Além disso, há também outras padronizações que podem ser realizadas pelo usuário.

(b) Configuração da interface gráfica

O usuário do sistema pode configurar uma interface do seu gosto gráfica para que o sistema fique o mais intuitivo possível. Esta ferramenta consta com 8 temas diferentes que podem ser modificados pelo usuário, para que a interface do sistema não se torne algo desgastante ao usuário. As alterações são ocorridas assim que o usuário confirma a alteração do tema, não sendo necessário, assim, reinicializar o programa.

Seção I.12 Configurações do sistema

(a) Configuração de acesso ao banco de dados

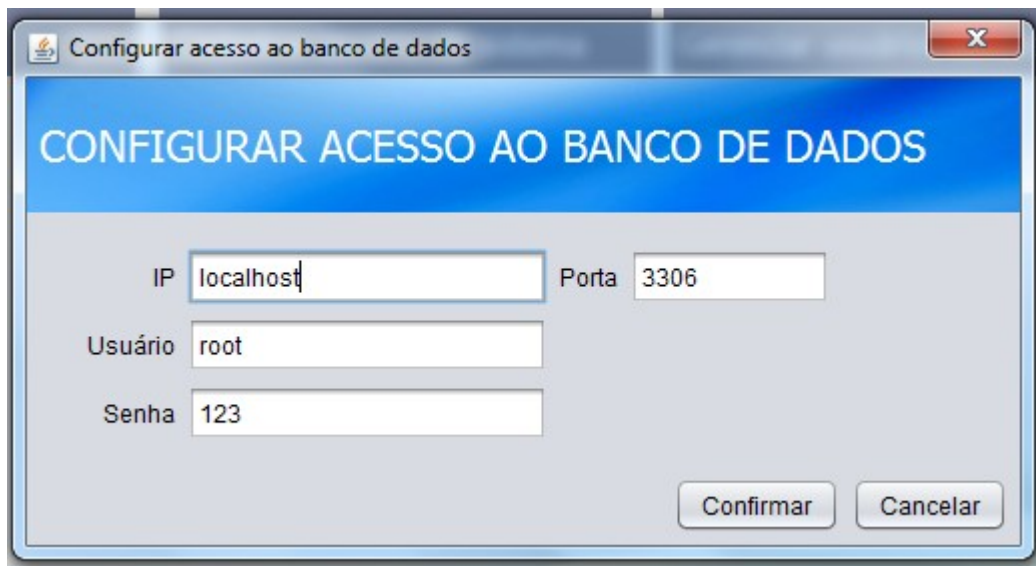
A imagem mostra uma janela de software com o título "Configurar acesso ao banco de dados". O cabeçalho interno da janela é azul e contém o texto "CONFIGURAR ACESSO AO BANCO DE DADOS" em letras brancas. Abaixo do cabeçalho, há quatro campos de entrada: "IP" com o valor "localhost", "Porta" com o valor "3306", "Usuário" com o valor "root" e "Senha" com o valor "123". No canto inferior direito da janela, há dois botões: "Confirmar" e "Cancelar".

Figura 15 - Configuração de acesso ao banco de dados

Esta ferramenta facilita a configuração de acesso ao banco de dados, os dados coletados é o IP e a porta da máquina onde se localiza o banco, o usuário e senha do MySQL. Desta forma, não é necessário fazer nenhuma alteração no código com

relação a isso após a implantação do sistema. Além de facilitar a migração do banco de dados para outro local.

(b) Gerenciamento de usuários

Esta parte se encarrega de cadastrar, editar e remover usuários do sistema. O cadastro de usuário que possibilita uma pessoa acessar o software. Nesta etapa, o usuário coloca seu nome, usuário, senha e imagem de perfil. Para deletar e remover um usuário, é preciso possuir a senha do usuário que será deletado ou removido, por forma de segurança.

Relatórios

JLibrary

Relatório detalhado do acervo

1984	
Código	A1234
Autor	George Orwell
Classificação	Romance
Editura	Moderna
Idioma	Português
ISBN	0451524934
Exemplar	3
Volume	1
Edição	1
Ano	2012
Núm. Págs.	100
Data de Cad.	04/12/12 00:00
Tipo de obra	Livro
Localização	V1H3

Força Brava 44 Dezembro 2012

Página 1 de 1

JLibrary

Relatório do acervo

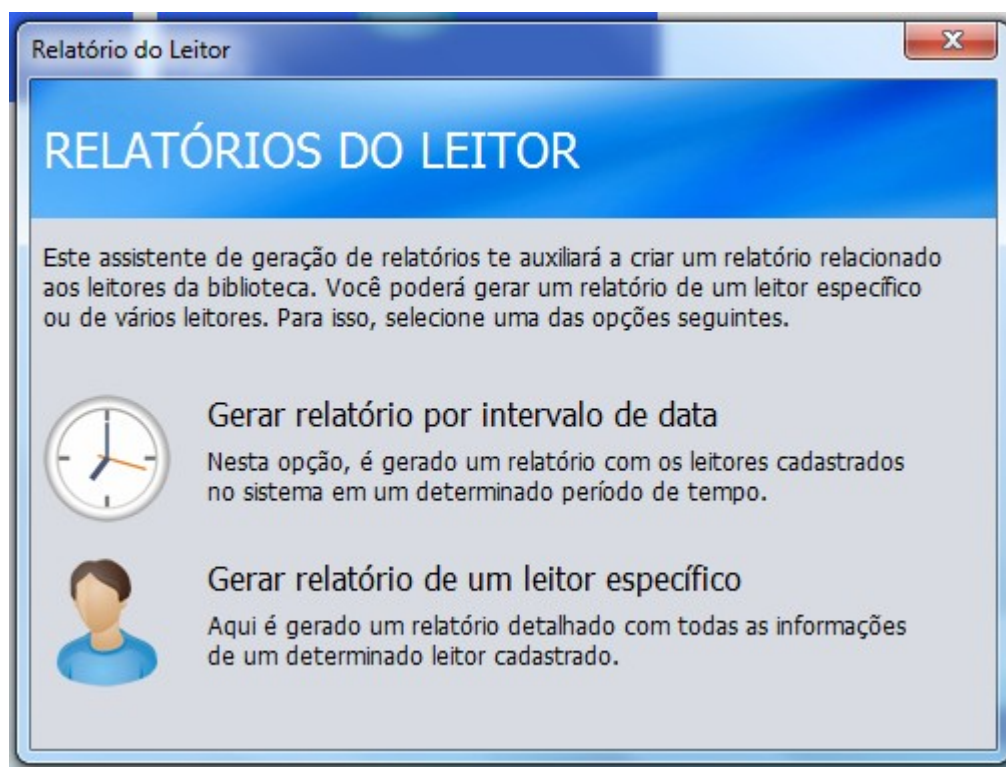
Data de cadastro	Título	Código
04/12/12	1984	A123
04/12/12	Viagens Na Minha Terra	B321
04/12/12	Til	C523
04/12/12	Memórias de Um Sargento de Milícias	G324
04/12/12	Memórias Póstumas de Bras Cubas	F437

Força Brava 04

Página 1 de 1

Os relatórios estão separados em três categorias: Acervo, Leitores e Movimentações. Sendo que cada categoria pode gerar diversos tipos de relatórios. Para isso, o usuário conta com um assistente que auxilia o processo da geração do relatório. Existem relatórios detalhados (esquerda) que diz respeito à uma determinada. Há outros que são mais abrangentes (direita) na qual são colocados os termos mais importantes de vários registros.

Assistente de criação de relatórios



O assistente de geração de relatórios tem a função de auxiliar o usuário criar um relatório, passando por etapas até que se chegue ao resultado final. Desta forma, a criação de um relatório se torna uma prática fácil, rápida e descomplicada.

Artigo II. Conclusão

O sistema de automação da biblioteca desenvolvido pela equipe foi nomeado de JLibrary. Sendo que a primeira letra (J) se refere ao Java – linguagem de programação adotada para o desenvolvimento do software – e Library – biblioteca, em inglês – por ser um sistema desenvolvido para bibliotecas.

Segundo pesquisas realizadas, o Java é a linguagem de programação mais utilizada no mundo. Isso se deve a diversos fatores positivos, tais como estabilidade, orientação ao objeto e uma grande comunidade mundial, além de ser interpretado por uma máquina virtual. Por conta disso, o Java foi escolhido para ser a linguagem de programação deste sistema.

Da mesma empresa desenvolvedora do Java, o MySql foi o Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido para o desenvolvimento do projeto. O MySql é muito popular e conta com mais de 10 milhões de usuários. Entre eles estão as instituições bancárias e governamentais, e grandes empresas como Google, Cisco, HP, Nokia, Sony, entre outras.

No decorrer da programação do sistema gerenciador da biblioteca, foi observado o quanto é importante o diálogo entre o cliente e o analista de sistemas. Pois esta é a única forma possível de programar um sistema que atenda as necessidades do usuário. Além disso, a análise de um projeto é essencial para eliminar as discrepâncias e redundâncias e implantar funcionalidades que sejam úteis ao cliente. Deste modo, a Engenharia de Software foi fundamental para o progresso do projeto.

Assim que era lançada uma nova versão, o software em funcionamento na biblioteca da escola passava por uma atualização. Com todos os paradigmas da Engenharia de Software, como, por exemplo, análise de requisitos, design do software, codificação, testes, treinamento, implantação e manutenção.

A análise de requisitos foi obtida através de reuniões feitas entre os desenvolvedores do sistema e a bibliotecária. O design do software foi baseado em recursos visuais disponíveis em outros softwares utilizados no cotidiano, bem próximo da nova interface de usuário da Microsoft, para que a adaptação ocorresse da forma mais rápida possível. A codificação consistiu na criação programação do sistema e criação de algoritmos que deixassem o sistema apto a desenvolver algumas atividades

automaticamente. A fase de testes foi realizada pela própria equipe desenvolvedora do sistema, que teve o objetivo de descobrir erros e aprimorar os algoritmos do software. O treinamento teve a finalidade de adaptar o usuário à nova interface e funcionalidades do sistema. A implantação consistiu na substituição do software que estava em funcionamento na biblioteca por uma nova versão. A manutenção foi feita com o propósito de sempre aprimorar o sistema, para que se torne uma ferramenta mais útil no cotidiano da biblioteca, auxiliando, assim, o trabalho dos profissionais.

No início do desenvolvimento, o sistema contava com poucas funcionalidades, apenas cadastro e gerenciamento de obras e leitores. Atualmente, o sistema conta com recursos muito avançados. Deste uma verificação de conexão com o banco de dados na tela de splash até a geração de gráficos e relatórios avançados de vários setores da biblioteca.

Após a implantação do sistema, foi notado que as atividades decorrentes da biblioteca foram realizadas com mais eficiência. Sendo que, anteriormente não era utilizado nenhum software para o gerenciamento da biblioteca, assim, todos os registros eram feitos manualmente. Diante disso, o software foi muito útil para a biblioteca, pois possibilita gerenciar todos os setores da biblioteca rapidamente, além de ser uma maneira de armazenamento de informações muito mais segura e organizada que a utilizada anteriormente.

Artigo III. Bibliografia

Pamela, Priscila, Charles, Jorge (2009) Trabalho de B.D.
<http://www.apostilaz.com.br/programacao/mysql-para-iniciantes.html>. Acesso em 15 de agosto de 2012.

Christian Cleber Masdeval Braz, (2010) Introdução à Linguagem Java.
<http://www.ebah.com.br/content/ABAAAAtO4AF/introducao-a-linguagem-java#>. Acesso em 20 de Agosto de 2012.

Comunidade NetBeans, (2008) Todas as Funcionalidades e Tecnologias Suportadas.
http://netbeans.org/features/all_pt_BR.html. em 1 de Setembro de 2012.

Daniel Destro do Carmo (2008) Acessando dados em Java com JDBC.
www.softechnetwork.com.br/java/CursoJDBC.pdf. Acesso em 10 de Setembro de 2012

Artigo IV. Cronograma

ATIVIDADE	PRAZO	PRECEDÊNCIA	RESPONSÁVEL
1	7 dias	Escolha do tema	Ronaldo
2	5 dias	Escolha da Linguagem	Ronaldo
3	1 dia	Distribuir as tarefas entre o grupo	Ronaldo
4	1 mês	Análise de Requisitos	Ronaldo
5	7 dias	Criação da interface gráfica	Ronaldo
6	6 dias	Modelagem do Banco de Dados	Ronaldo
7	3 meses	Programação	Ronaldo
8	15 dias	Documentação do Projeto	Ronaldo