Universidade Federal do Rio Grande do Sul — Instituto de Informática
INF01120 – Técnicas de Construção de Programas — 2018/2
Diego Dimer Rodrigues — 00287690
Augusto Peiter — 00287685
Eduardo Paim — 00277322

Trabalho Final - Sistema de Biblioteca

O que é

Escolhemos criar um sistema que poderia ser usado em bibliotecas para controle de alugueis de livros. A motivação do assunto veio de percebermos que muitas bibliotecas ainda tem o controle dos alugueis dos seus livros por fichas de papel guardadas junto aos livros. Baseamos parte do sistema também no SABi[1], da UFRGS, onde podemos consultar livros e nossos aluguéis.

Diagrama de Classes

Para modelar o sistema, usamos dois diagramas de classes: um com as classes referentes a interface gráfica e persistência de dados e outro com tudo relativo ao uso do sistema. Para representar o livro, criamos uma classe chamada Titulo, dessa classe, estendemos três classes novas: ExemplarFisico, ExemplarOnline e ExemplarAlugado. Escolhemos estender nas três classes distintas pelas diferenças entre elas; exemplares físicos participam de alugueis e tem uma variável inteira com o número de livros disponíveis; Exemplares online tem link para acessar e não participam de alugueis (o aluguel dele é simplesmente a impressão do link na tela), e exemplar alugado é um título (exemplar físico) que realizou um aluguel, tem datas de empréstimo e devolução. Um aluguel é representado pelo exemplar alugado, e guardado numa tabela específica na base de dados.
O uso de três classes estendidas de título semelhantes fica mais visível durante o uso do sistema: quando quisermos listar os alugueis ativos de um usuário, usamos o exemplar alugado. Quando buscamos um livro, estamos buscando um livro físico ou online. O livro online não entra para alugueis ativos, o físico sim.
Na parte gráfica, representamos interfaces para base de dados, funcionários e usuários. O objetivo de ter interface e implementação separadas é dar transparência aos métodos públicos que aquelas classes implementam, facilitando também mudanças no software. Foram criadas classes LoginUI e RegistroUI para registrar e logar usuários dentro do sistema, são as interfaces gráficas que usam funções da base de dados. A class Database se tornou extensa, porém, acreditamos que tudo nela ainda faz sentido para a aplicação: são métodos de consultas à base de dados.
Em relação a etapa 1, nossos diagramas foram atualizados, tiramos algumas classes que durante a

[1] https://sabi.ufrgs.br

implementação não fizeram sentido (exemplarAlugavel e Aluguel) e alguns métodos foram modificados (tiramos o login da classe Usuario, o login agora é uma função da interface que chama uma função de consulta sql à tabela usuarios, por exemplo). A essência do projeto continuou igual, apenas remodelamos aquilo que se fez necessário para dar sentido a implementação.

O que usamos

Persistência de dados

Para guardar os registros, usamos um banco de dados gratuito do site Heroku[2]. Se o sistema fosse ser *realmente* implementado e vendido, procuraríamos uma base de dados paga para termos mais segurança e suporte. Usamos também a linguagem SQL para acessar e manipular dados do banco de dados. A conexão com a base de dados foi pelo JDBC[3] usando as credenciais do heroku. Usamos uma classe Database que implementa uma interface DatabaseInterface, a implementação facilita podermos mudar, se um dia quisessemos usar outro método, poderíamos apenas reimplementar a interface DatabaseInterface e pouco código precisaria ser modificado.

Documentação

Para a documentação, usamos Javadoc[4], que extrai direto do código todos os métodos e seus comentários. Na documentação, encontram-se todas as descrições dos métodos e árvores de herança das classes.

Interfaces Gráficas

Para interface gráfica, usamos a extensão do eclipse windowbuilder[5] com o Swing[6], onde todas interfaces são JFrames.

Versionamento

Usamos o GitHub para versionamento do nosso software. Criamos um repositório em que todos do grupo foram adicionados como contribuidores. O repositório pode ser acessado pelo link: https://github.com/diegodimer/Trabalho-TCP.

Singleton

Usamos na classe de database singleton, para tornar única a conexão, representada pela variavel *conec* do tipo *Connection*. Para isso, a variável começa com null, e quando o construtor da classe é

[2]"https://dashboard.heroku.com/"
[3]https://www.oracle.com/technetwork/java/javase/jdbc/index.html
[4]https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html
[5]https://www.eclipse.org/windowbuilder/
[6]http://download.eclipse.org/windowbuilder/lastgoodbuild/

chamado pela primeira vez, estabelece-se a conexão. Depois disso, nada é feito quando chama-se o construtor novamente.

Interfaces

Para possíveis alterações de implementação, foram usadas interfaces para Usuário, Funcionário e Database. Qualquer alteração na tecnologia usada nessas classes, seriam apenas implementações dessas interfaces, garantindo o funcionamento do software.

Testes

Usamos o JUnit 4 para testes. O problema de testes quando envolve database é que todo teste pode ser executado somente uma vez (a função de adicionar algo numa tabela, se aquela linha é única, só pode ser adicionada uma vez). Então, a estratégia adotada foi comentar o teste após ele passar e ser garantido que está funcionando. Usamos setup antes de todos os métodos para conectar a base de dados, e teardown para fechar a conexão após os testes. A implementação dos métodos foi baseada em TDD, escrevíamos o teste e depois implementávamos o método na classe Database.

Conclusão

Esse trabalho foi a primeira experiência com Java para todos do grupo. Não conhecíamos também nada sobre banco de dados até o início do semestre e nada sobre conexão entre banco de dados e java até o início do trabalho. Acreditamos que ainda há muito o que melhorar no projeto (tornar ele mais seguro à SQLinjection, adicionar sistema de multas automatizado, criptografia na senha dos usuários e etc). O resultado foi satisfatório para o escopo da disciplina, acreditamos que conseguimos por em prática coisas aprendidas sobre orientação a objetos e técnicas aprendidas em aula.