

Pedro Felipe Froes Silva

Desenvolvimento de Aplicações em Java e Utilizando Conceitos de UI e UX

Belo Horizonte

2017

Pedro Felipe Froes Silva

Desenvolvimento de Aplicações em Java e Utilizando Conceitos de UI e UX

Relatório apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a aprovação na disciplina Estágio Supervisionado.

Centro Federal de Educação Tecnológica de Minas Gerais

Departamento de Computação

Curso de Engenharia de Computação

Orientador: Kécia Aline Marques Ferreira

Empresa: Avenue Code Desenvolvimento e Comércio de Softwares Ltda

Belo Horizonte

2017

Resumo

Resumo a ser escrito. **Palavras-chave:** Estágio supervisionado. Java. *User interface*. *User experience*.

Lista de abreviaturas e siglas

API	Interface de programação de aplicações (<i>application programming interface</i>)
DAO	Objeto de acesso a dados (<i>data access object</i>)
JDBC	Java Database Connectivity
UI	Interface de usuário (<i>user interface</i>)
UX	Experiência de usuário (<i>user experience</i>)
WORA	<i>Write once, run anywhere</i>

Sumário

1	INTRODUÇÃO	7
2	ESTÁGIO SUPERVISIONADO	9
2.1	Sobre a empresa: Avenue Code	9
2.2	Sobre o estágio: Programa de Estágio Jedi Internship	10
3	FUNDAMENTAÇÃO TEÓRICA	11
3.1	Java	11
3.2	<i>User interface e user experience</i>	13
4	ATIVIDADES DESENVOLVIDAS	15
4.1	Migração de bancos de dados em Java	15
4.2	Desenvolvimento de interfaces e experiências de usuário	17
5	CONCLUSÃO	19
	REFERÊNCIAS	21

1 Introdução

O presente trabalho é um relatório da disciplina de Estágio Supervisionado pertencente ao curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG), realizado pelo discente Pedro Felipe Froes Silva na empresa Avenue Code Desenvolvimento e Comércio de Software Ltda. Este relatório reflete o período de Março a Maio de 2017 durante o estágio do aluno na empresa, contemplando parte do Programa de Estágio Jedi Internship.

O Programa de Estágio Jedi Internship da Avenue Code corresponde a uma rotação dos participantes por diferentes tecnologias presentes na área de Computação. Durante o período do programa, o estagiário passa por cinco áreas distintas, obtendo um aprendizado de linguagens de programação como (i) Java e (ii) Ruby, (iii) do *framework* .NET, (iv) de conceitos de garantia de qualidade de software, e (v) de conceitos e *frameworks* para o desenvolvimento de interfaces de usuário (*user interface*, UI) e experiência de usuário (*user experience*, UX). O estudante exercita o aprendizado de cada área por meio de projetos internos da empresa, e apresenta um *workshop* com o conteúdo aprendido ao final de cada etapa.

O objetivo desse trabalho é relatar as experiências do discente nas áreas em que o mesmo participou durante o período do Estágio Supervisionado, que correspondem ao aprendizado de Java e de conceitos de UI e UX. O aprendizado em cada uma das áreas é apresentado por meio do processo de desenvolvimento de um migrador de banco de dados em Java e da construção de interfaces de usuário por meio de *frameworks* de UI e ferramentas de UX.

O restante desse trabalho está organizado de forma que o Capítulo 2 apresenta a empresa e o seu Programa de Estágio, enquanto o Capítulo 3 aponta conceitos básicos de Java e de UI e UX. O Capítulo 4 detalha as atividades em cada uma das áreas, enquanto o Capítulo 5 conclui o trabalho.

2 Estágio supervisionado

Neste capítulo, a empresa concedente Avenue Code e o Programa de Estágio Jedi Internship são apresentados. Tópicos envolvendo a história, especialidade e iniciativas da empresa, bem como a estrutura do programa de estágio são ilustrado nas seções a seguir.

2.1 Sobre a empresa: Avenue Code

A Avenue Code Desenvolvimento e Comércio de Softwares Ltd é uma empresa consultora de softwares especializada no ramo de *e-commerce* da indústria varejista. Fundada em 2008 pelo CEO Zeo Solomon na cidade de San Francisco (Califórnia, EUA), a Avenue Code atendia clientes americanos, abrindo seu primeiro escritório no Brasil somente um ano depois, na cidade de Belo Horizonte. Nos anos seguintes, a empresa expande e inaugura um escritório na cidade de São Paulo, além de adicionar Amir Razmara e Chase Hill ao time de CEOs. Em 2017, a Avenue Code é formada por mais de 230 consultores em sua equipe, além de inaugurar um quarto escritório, agora na cidade de Nova York (AVENUE CODE, 2017).

A Avenue Code é especialista no desenvolvimento e utilização de diversos tipos de tecnologias da área de Computação, como aplicações Web e móveis, automação de infraestruturas, sistemas de *backend*, implementações de plataformas, *coaching* Agile e DevOps e integrações corporativas. A Metodologia Ágil, oriunda do Manifesto Ágil para o Desenvolvimento de Software (FOWLER; HIGHSMITH, 2001), é altamente aplicada na empresa, que busca maximizar sua eficiência através da utilização de princípios Agile no desenvolvimento de projetos (AVENUE CODE, 2017).

Prezando tanto pela qualidade da tecnologia utilizada quanto pelo ambiente de trabalho dos consultores, a Avenue Code possui uma gama de empresas parceiras e de prêmios obtidos em sua história. Dentre as empresas de tecnologia parceiras da Avenue Code, figuram a Mulesoft ¹, SAP Hybris ², CHEF ³, Oracle ⁴, Amazon Web Services ⁵ e Adobe ⁶. Já entre os prêmios conquistados pela empresa, estão o reconhecimento pelo

¹ MuleSoft: Integration platform for connecting SaaS and enterprise applications. Disponível em: <<https://www.mulesoft.com/>>

² SAP Hybris: E-commerce solutions. Disponível em: <<https://www.hybris.com/en/>>

³ Chef: automate IT infrastructure. Disponível em: <<https://www.chef.io/chef/>>

⁴ Oracle: Integrated cloud applications and platform services. Disponível em: <<https://www.oracle.com/>>

⁵ Amazon Web Services: Cloud computing services. Disponível em: <<https://aws.amazon.com/>>

⁶ Adobe: Creative, marketing and document management solutions. Disponível em <www.adobe.com/>

LoveMondays⁷, InfoMoney⁸ e San Francisco Business Times's Fast 100⁹ em 2016, além de ser agraciada como uma das Melhores Empresas para Trabalhar¹⁰ em 2016 (*Great Place to Work*) (AVENUE CODE, 2017).

Por fim, a Avenue Code participa de iniciativas de inclusão digital por meio do programa AC Social, que oferta aulas de introdução à tecnologias e computação em escolas carentes. A empresa também oferece cursos ministrados pelos próprios consultores por meio do AC Community, além de ofertar dois programas de estágios distintos, o AC Wonder Women e o Jedi Internship, que será apresentado na seção seguinte (AVENUE CODE, 2017).

2.2 Sobre o estágio: Programa de Estágio Jedi Internship

O Programa de Estágio Jedi Internship consiste de uma rotação (*job rotation*) por cinco áreas de diferentes tecnologias presentes na área de Computação. São elas:

- Conceitos e *frameworks* de *user interface* (UI) e *user experience* (UX)
- Conceitos e *frameworks* de garantia de qualidade de software
- Linguagem de programação Java
- Linguagem de programação Ruby
- *Framework* .NET

Cada área dura cerca de três meses, e em cada uma delas o estagiário aprende conceitos introdutórios e avançados da tecnologia, aplicando-os em projetos internos da empresa. Cada área possui consultores que atuam como mentores para os participantes e, ao final da área, o estagiário elabora um *workshop* para ser apresentado tanto para seus mentores quanto para os outros participantes do Programa, exibindo conceitos, aplicações desenvolvidas e desafios encontrados ao longo dos três meses.

Esse trabalho apresentará os conceitos aprendidos e aplicações desenvolvidas pelo autor ao longo das áreas de Java e UI/UX, sendo que uma fundamentação teórica para ambas as áreas é exibida no capítulo seguinte.

⁷ Love Mondays: A empresa ideal, avaliada por profissionais como você. Disponível em: <<https://www.lovemondays.com.br/>>

⁸ InfoMoney: Notícias, ações e muito mais sobre investimentos. Disponível em: <www.infomoney.com.br/>

⁹ San Francisco Business Time Fast 100. Disponível em: <<http://www.bizjournals.com/sanfrancisco/blog/2016/10/bay-area-fast-growing-private-companies-fast-100.html>>

¹⁰ Great Place to Work. Disponível em: <www.greatplacetowork.com.br/>

3 Fundamentação teórica

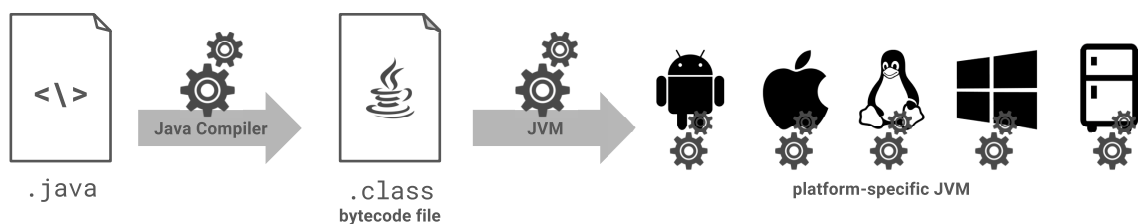
Neste capítulo, a fundamentação teórica das tecnologias abordadas neste trabalho é realizada. Conceitos básicos sobre a linguagem de programação Java são mostrados, abordando tanto os pilares da orientação a objetos presente na linguagem quanto bibliotecas relevantes para as aplicações desenvolvidas na área. Posteriormente, conceitos relevantes para a o desenvolvimento de interfaces de usuário (*user interface*, UI) e experiência de usuário (*user experience*, UX) são apresentados, apontando características e ferramentas utilizada no processo de construção de interfaces.

3.1 Java

Uma das tecnologias presentes no Programa de Estágio Jedi Internship e abordadas nesse trabalho corresponde ao aprendizado e aplicação de conceitos da linguagem de programação Java. Originada inicialmente em 1991 com o codinome de Oak, e nomeada de Java somente em 1995, atualmente a linguagem é uma das mais utilizadas na área de Computação, sendo administrada pela empresa Oracle e estando em sua oitava versão (BOYARSKY; SELIKOFF, 2015).

Em suas primeiras versões, o propósito inicial do Java era conectar diferentes tipos de micro-sistemas da empresa Sun, sendo uma linguagem comum entre eles. A habilidade de escrever um código que pode funcionar em mais de um sistema é conhecida como *write once, run anywhere* (WORA), sendo uma das principais características da linguagem. Ao escrever um código em Java, o compilador Javac processa o arquivo fonte para um arquivo em *bytecode*, e um interpretador JVM específico para a plataforma se encarrega de processar esse arquivo posteriormente, como mostrado na Figura 1 (BOYARSKY; SELIKOFF, 2015).

Figura 1 – Esquema de execução de um código escrito em Java.



Fonte: Próprio autor.

Além do WORA, outra característica marcante da linguagem é a implementação

do conceito de orientação a objetos. Para implementar tal conceito, o Java faz uso de classes e objetos: enquanto uma classe funciona como uma especificação de uma ideia, um objeto corresponde a uma instância, uma materialização dessa ideia, sendo que uma classe pode possuir vários objetos instanciados (BOYARSKY; SELIKOFF, 2015).

A relação entre classes e objetos dá margem para diversos outros conceitos presentes na linguagem Java. O conceito de encapsulamento, por exemplo, faz uso de modificadores de acesso nos atributos e métodos de cada classe para controlar quais objetos podem acessá-los, enquanto o conceito de herança entre as classes determina uma relação de pai-filho entre elas, tornando possível que uma herde atributos e métodos da outra (BOYARSKY; SELIKOFF, 2015).

Conceitos de abstração, composição e interfaces também estão presentes na linguagem, possibilitando a criação de classes não instanciáveis, classes compostas por diversos objetos, e a garantia que alguns métodos são implementados em determinadas classes, respectivamente. Por fim, um conceito essencial da linguagem é o polimorfismo, que permite que um objeto seja referenciado por diversas maneiras, como por meio das classes que ele herda, ou de interfaces que implementa (BOYARSKY; SELIKOFF, 2015).

Além dos conceitos apresentados, o Java ainda conta com bibliotecas como o *collections framework*, que implementam diferentes estruturas de dados utilizadas comumente na programação de computadores. Estruturas como *hashes*, listas de vetores, pilhas e filas podem ser utilizadas através das interfaces `HashSet`, `ArrayList`, `Stack` e `Queue`, respectivamente. Além de eliminar a necessidade do programador construir cada uma das estruturas, a utilização delas é extremamente comum em aplicações construídas em Java, facilitando o trabalho do desenvolvedor (WATT; BROWN, 2001).

Tanto os elementos presentes no *collections framework* quanto os conceitos apresentados previamente podem ser utilizados para diversos tipos de aplicações em Java (WATT; BROWN, 2001). O gerenciamento de banco de dados relacionais em Java, por exemplo, utiliza uma interface de programação de aplicações (*application programming interface*, API) chamada Java Database Connectivity (JDBC). O JDBC abstrai a implementação de banco de dados específicos, criando uma camada única que contribui para a implementação de métodos para estabelecer conexões, criar *queries* de acesso, e extrair resultados de buscas, por exemplo (REESE, 2000).

O JDBC ainda conta com maneiras para gerenciar múltiplas conexões em um servidor através de um *pool* de conexões, e de centralizar o acesso de dados por meio de objetos de acesso a dados (*data access object*, DAO) (REESE, 2000). A utilização de alguns desses recursos é aprofundada na Seção 4.1 do Capítulo 4, que descreve as atividades realizadas durante a área de Java do Programa de Estágio.

3.2 *User interface e user experience*

A ser escrita.

4 Atividades desenvolvidas

As atividades desenvolvidas durante o período do Estágio Supervisionado são descritas nesta seção. Primeiramente, uma aplicação construída para migrar dados de um banco para outro por meio da linguagem Java é descrita, fazendo uso de conceitos apresentados na Seção 3.1. Posteriormente, um projeto utilizando o *framework* Angular é apresentado, fazendo uso dos conceitos de UI/UX apresentados na Seção 3.2.

4.1 Migração de bancos de dados em Java

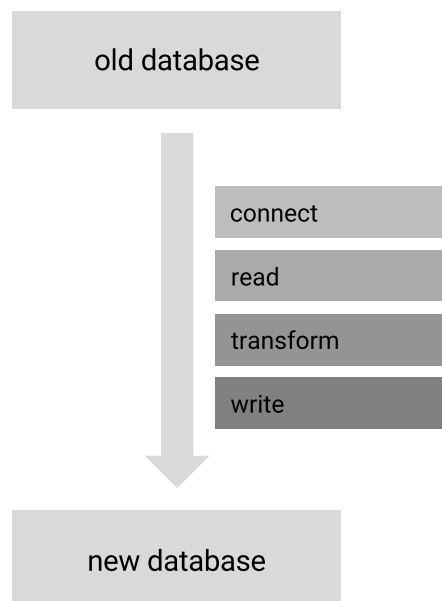
O Avenue Code Events Billing Administration (AC-EBA) é um projeto interno da empresa que visa o gerenciamento de gastos monetários e de pessoal em eventos realizados pela Avenue Code. Originalmente desenvolvido na filial de São Paulo, uma segunda versão do sistema começou a ser projetada a fim de adicionar funcionalidades e melhorar o seu desempenho. A nova versão do AC-EBA começou a ser construída com novas linguagens de *backend* e *frontend* e, portanto, não foi desenvolvida sobre o código da primeira versão. Dessa forma, era necessário migrar os dados da primeira versão para a segunda, e optou-se por construir uma aplicação em Java para auxiliar nesse processo.

O Java Database Connectivity (JDBC) é uma *application programming interface* (API) da linguagem da Java que possibilita gerenciar banco de dados relacionais por meio da abstração da implementação de bancos de dados específicos, criando uma camada intermediária com diversos métodos para o estabelecimento de conexões, leitura, escrita de dados, e extração de resultados a partir de *queries*. Utilizando diversos métodos providos pelo JDBC, uma aplicação para a migração dos dados do banco da primeira versão do AC-EBA foi desenvolvida, e a ela foi dado o nome de AC JDB Migrator (AC-JDBM).

O AC-JDBM foi projetado em uma *pipeline* de quatro etapas diferentes, mostradas na Figura 2. As etapas da *pipeline* dessa aplicação são: (i) conectar-se ao banco de dados do projeto original, (ii) ler os dados presentes nesse banco, (iii) transformar os dados de acordo com as tabelas presentes no novo sistema, e (iv) utilizar esses dados transformados para a escrita das *queries* para popular o banco de dados do novo sistema.

A primeira etapa da aplicação é constituída pela classe `DbConnector`, que busca realizar uma conexão com um banco de dados. Para isso, ela recebe um arquivo de extensão `Property` como parâmetro, que contém as credenciais necessárias para a conexão com um banco. A classe foi feita de modo genérico, sendo que é possível criar um objeto para conectar-se tanto com o banco de dados original quanto o do novo sistema, e métodos de conexão do JDBC foram utilizados em seu desenvolvimento.

Figura 2 – Esquema de execução da aplicação AC JDB Migrator.



Fonte: Próprio autor.

A etapa seguinte corresponde à leitura dos dados do banco original, sendo realizada pela classe `DbReader`. *Queries* para a leitura de todas as tabelas presentes no banco original (por exemplo, `select * from EVENT`) foram codificadas, e os resultados dessas *queries* é obtido por meio de objetos do tipo `ResultSet`. Esses objetos foram então mapeados para objetos do tipo `Table`, que correspondiam às tabelas do antigo banco de dados. A classe `Table` é constituída de um `Set` de objetos do tipo `Row`, e tanto `Table` quanto `Row` foram criados para armazenar os dados lidos de acordo com a tabela correspondente e, posteriormente, passá-los para a próxima etapa da *pipeline*.

A terceira etapa consiste na transformação dos dados lidos para se adequarem às tabelas do banco de dados do novo projeto. O `DbTransformer` foi desenvolvido para receber o conjunto de objetos do tipo `Table` correspondente às tabelas do antigo banco, e transformar os dados recebidos de acordo com as tabelas no novo sistema, que diferiam do sistema original por meio da adição e/ou remoção de algumas colunas, por exemplo. Dessa forma, o `DbTransformer` é constituído de outros objetos, como `EventTransformer` ou `BudgetTransformer`, e cada um deles retorna objetos do tipo `Set` contendo uma interface construída para armazenar os dados transformados, a `SQLWritable`.

Finalmente, a última etapa da *pipeline* de migração do dados é responsável por conectar-se ao banco do novo sistema e escrever as *queries* para a inserção dos dados transformados. Para realizar a conexão, foi utilizada uma instância do `DbConnector` com um arquivo `Property` com os dados do banco do novo sistema, e a interface `SQLWritable`

teve o método `toSQL()` implementado em cada uma de suas instâncias. Esse método era responsável por escrever as *queries* para a inserção dos dados de cada uma das tabelas lidas para as novas tabelas.

No desenvolvimento do AC-JDBM foram utilizados diversos conceitos de orientação a objetos, de padrões de projeto e boas práticas de programação. A criação do projeto como uma *pipeline*, a manipulação de interfaces ao invés de tipos concretos, o encapsulamento dos dados e a separação de responsabilidades pelas classes ao longo do projeto são algumas das características que contribuíram não só para o sucesso da aplicação, mas também para o aprendizado do autor na utilização da linguagem Java aplicada ao *backend* de sistemas.

4.2 Desenvolvimento de interfaces e experiências de usuário

A ser escrita.

5 Conclusão

A ser escrita.

Referências

AVENUE CODE. *Avenue Code*: Trusted advisors for e-commerce. 2017. <<https://www.avenuecode.com/>>. Acesso em: 01/05/2017.

BOYARSKY, J.; SELIKOFF, S. *OCA: Oracle Certified Associate Java SE 8 Programmer I*: Study guide exam 1z0-808. 1. ed. Indianapolis, Indiana (EUA): John Wiley & Sons, Inc., 2015.

FOWLER, M.; HIGHSMITH, J. The agile manifesto. *Software Development*, Miller Freeman, Inc., San Francisco, Califórnia (EUA), v. 9, n. 8, p. 28–35, 2001.

REESE, G. *Database programming with JDBC and JAVA*. 2. ed. Sebastopol, Califórnia (EUA): O'Reilly Media, Inc., 2000.

WATT, D. A.; BROWN, D. *Java collections*: An introduction to abstract data types, data structures and algorithms. 1. ed. Nova York, Nova York (EUA): John Wiley & Sons, Inc., 2001.