

TRABALHO SEMESTRAL – ORIENTAÇÃO A OBJETOS	
Curso: Bacharelado em Ciência da Computação	Período: 3º
Disciplina: Orientação a Objetos	Valor: 10 pontos (peso 0.10)
Professor: Rodrigo Martins Pagliares	Data de entrega: 10/06/2024

Data da última atualização: 24/05/2024

### Instruções:

- Data de entrega: até **10/06/2024 até às 23:59h**.
- Este trabalho poderá ser realizado em grupos de no máximo 5 pessoas.
- Avaliação: Irei avaliar o trabalho com base no código fonte (peso 1) e na apresentação do vídeo (peso 2).

**1. Crie um vídeo de apresentação com sua explicação para a solução deste TRABALHO SEMESTRAL. O vídeo deverá ter entre 15 e 30 minutos e deve estar disponível *online* (por exemplo, no YouTube).**

- O *link* com o vídeo deverá ser enviado para [pagliares@bcc.unifal-mg.edu.br](mailto:pagliares@bcc.unifal-mg.edu.br)
- No *e-mail* com o *link* do vídeo deverá ser incluído também dois *links*: um para o código fonte da **solução do exercício 2** e um segundo *link* para o código fonte da **solução do exercício 3**.

**2. Neste exercício, os discentes terão a oportunidade utilizar os conhecimentos de programação orientada a objetos em um projeto que inclui interface gráfica com o usuário (GUI) e persistência de dados em um banco de dados relacional.**

### História de usuário:

US-1050: Create product
<p>As a Administrator</p> <p>I want to create a new Product</p> <p>In order to increase the variety of products we sell at BlueVelvet Music Store</p>
Acceptance criteria
<ul style="list-style-type: none"> <li>- A product has a name, short description, brand, category, list price, cost.</li> <li>- The fields name and list prices must not be empty</li> </ul>

US-0809: Edit product information
<p>As an Administrator</p> <p>I want to edit the data of an exiting product</p> <p>to improve/fix the information of the products sold by Blue Velvet Music Store.</p>
Acceptance criteria
<ul style="list-style-type: none"> <li>- All fields of a product may be updated.</li> <li>- See US-1050 for the product fields.</li> <li>- Test retrieving a product</li> <li>- Test Updating some product fields:</li> </ul>

US-2032: List products
<p>As an Administrator</p> <p>I want to list all of products</p> <p>in order to have a global view of all products sold on Blue Velvet Music store.</p>
Acceptance criteria
<ul style="list-style-type: none"> <li>- The list of products may be too long. Provide a way for navigation (e.g. pagination for the products list)</li> <li>- By default, it must be possible to see 10 products at once (pagination)</li> <li>- Pagination is based on product names</li> <li>- Pagination should work well with sorting</li> <li>- Sort by product name, ID, brand, and category</li> <li>- Default sorting by name, ascending order</li> <li>- Filter: search by product name</li> </ul>

US-1627: Delete product
<p>As an Administrator</p> <p>I want to delete a product</p> <p>in order to have only products that are sold by Blue Velvet Music Store</p>
Acceptance criteria
<ul style="list-style-type: none"> <li>- Test Deleting a product</li> </ul>

Protótipo de baixa fidelidade:

Product Management

Name

Radiohead T-shirt

Product name

Search

Short description

Brand

Hering

Category

T-Shirt

List price

25.00

Cost

10.00

Create

Update

Delete

Name	Short description	Brand	Category	List price	Cost (US\$)
Radiohead T-Shirt	Black T-Shirt of the album In Rainbows	Hering	T-shirt	25.00	10.00
Wilco CD	CD of the Album A ghost is Born	Virgin Records	CD	15.50	8.00

<<

1

2

3

>>

Figura 1 – Protótipo da GUI a ser implementada usando a biblioteca JavaFX.

## Tecnologias a serem utilizadas:

- **Java SE**
  - Eu estou usando OpenJDK 21.
- **JavaFX**
  - **Sugestão:** Use o assistente específico para criação de projetos **JavaFX** no **IntelliJ** para evitar ter que incluir manualmente as dependências do **JavaFX** no arquivo pom.xml).
  - **Observação:** A solução deverá **OBRIGATORIAMENTE** usar a **API Java FX**.

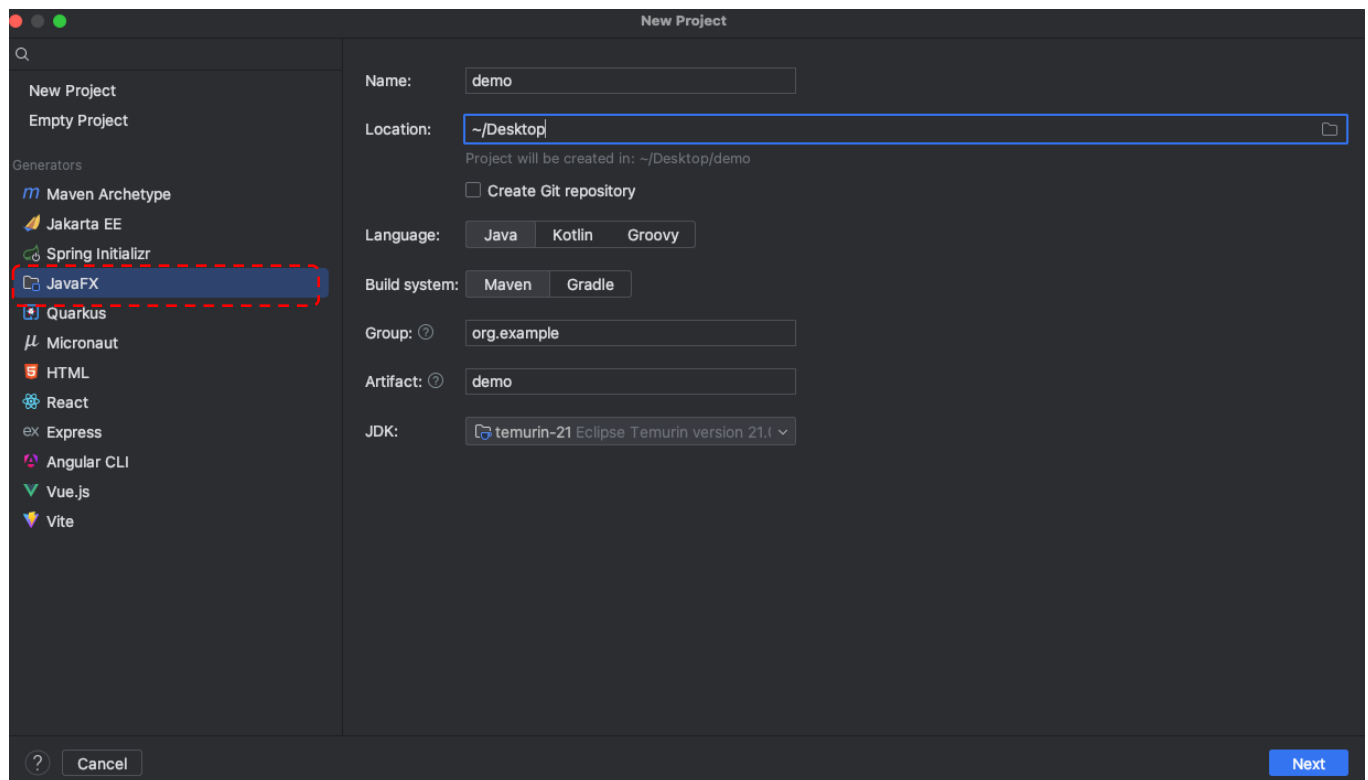


Figura 2 – Assistente para criação de um projeto Java SE com JavaFX no IntelliJ.

- **MySQL**
  - Os dados da aplicação deverão ser persistidos em um **banco de dados relacional MySQL**. Já que vários discentes ainda não fizeram a disciplina de banco de dados, optei por um exemplo simples com apenas uma tabela.
  - Na **Seção Trabalho do Semestre** na página principal da disciplina no **Moodle**, estão presentes alguns artefatos (Slides e Scripts SQL) com um passo a passo com instruções e comandos MySQL para aqueles que ainda não tiveram contato com Sistemas Gerenciadores de Banco de Dados.
    - Também está disponível o endereço de um repositório no **GitHub** com vários exemplos de como conectar um programa em Java com o **MySQL** usando a **API JDBC do Java SE**.

**4. Neste exercício, o discente deverá implementar a solução do Exercício 2 (incluindo interface gráfica e banco de dados) em uma segunda programação Orientada a Objetos (por exemplo, Ruby, C++, C#, JavaScript, Node.JS ou Python)**

**Sugestão:** Use o **ChatGPT** ou **Gemini** como assistente na realização deste exercício, solicitando auxílio para mapeamento do código feito no **exercício 2** para a linguagem de programação escolhida.

**Sugestão:** Se você preferir, sinta-se à vontade em manter neste exercício, o banco de dados criado no MySQL para o exercício 2. Caso prefira, sinta-se à vontade em trocar também de Sistema Gerenciador de Banco de Dados (por exemplo, usando PostGreSQL).

**Sugestão 3:** Caso você opte por utilizar a linguagem JavaScript ou Node.JS, a interface gráfica com o usuário deverá ser implementada usando as tecnologias HTML, CSS e JavaScript para serem executadas dentro de um navegador *web*.

Bons códigos!

Rodrigo Martins Pagliares