

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
ESTADO DE SÃO PAULO
CÂMPUS CAMPINAS

Gustavo Souza da Mota - CP1854194

Mike Jones Cabrera da Silva - CP1854119

Tiago de Freitas Faustino - CP1854127

Sistema de gerenciamento de compra e venda de livros

Campinas

2019

Gustavo Souza da Mota - CP1854194

Mike Jones Cabrera da Silva - CP1854119

Tiago de Freitas Faustino - CP1854127

Relatório final de atividades apresentado ao curso Técnico em Desenvolvimento de Sistemas, do Instituto Federal de Educação, Ciência e Tecnologia do Estado de São Paulo, Câmpus Campinas, como parte das exigências para avaliação de trabalho interdisciplinar desenvolvido em equipe, no terceiro semestre de curso.

Prof. André Luis Bordignon

Programação Avançada I

Profa. Joice Barbosa Mendes

Programação Orientada a Objetos

Prof. Pedro A. Pinheiro Fantinatti

Projeto e Desenvolvimento de Sistemas

Prof. Rafael da Silva Muniz

Programação para Web

Campinas

2019

Agradecimentos

Agradecemos aos professores envolvidos neste projeto pela atenção dedicada aos alunos, orientações e, principalmente, motivação cotidiana para o desenvolvimento de um ambiente favorável à busca, aplicação e compartilhamento do conhecimento adquirido.

Lista de ilustrações

Imagem 1 - Visão geral da dinâmica da metodologia SCRUM

Imagem 2 - Sinalizador visual Kanban com o software Microsoft Planner

Imagem 3 - Spectrum Ranking: The Top Programming Languages 2018

Imagem 4 - Banco de dados - Google Cloud Platform

Imagem 5 - Tela do projeto no software Microsoft Project Professional 2016

Imagem 6 - Quadros no software Microsoft Planner

Imagem 7 - Gráficos no software Microsoft Planner

Imagem 8 - Agenda no software Microsoft Planner

Imagem 9 - Código hospedado na plataforma GitHub

Imagem 10 - IDE Eclipse

Prefácio

Desde a entrega do Relatório de Atividades em 28 de junho de 2019, houveram algumas mudanças importantes no projeto quanto às linguagens de programação utilizadas (JAVA e PHP) e metodologias empregadas (Scrum, Kanban e Scrumban). Deste modo, faz-se necessário situar algumas considerações importantes. A Introdução foi integralmente mantida, sem alterações. Já o Desenvolvimento apresentou diversas modificações, mantendo-se inalterados, apenas, os Objetivos e a Justificativa. A Conclusão, também foi modificada. Os demais tópicos foram atualizados.

Dessarte, este Relatório, em sua versão final, compreende as seções de Introdução, Desenvolvimento, Conclusão, Referências bibliográficas e Apêndices.

A seção 1 (Introdução) apresenta o tema e o contexto objetos de estudo, trazendo ao leitor o panorama do cenário no qual a equipe trabalhou.

Na seção 2 (Desenvolvimento) são apresentados os objetivos gerais e específicos, a justificativa, a equipe, as metodologias, as técnicas e ferramentas, as linguagens de programação (PHP e JAVA), os softwares de gerenciamento de projetos, a plataforma de hospedagem do código desenvolvido, a IDE utilizada e os resultados obtidos.

Em continuidade, a seção 3 (Conclusão) retrata as percepções conclusivas da equipe.

Na seção 4 (Referências bibliográficas) são exibidos os materiais utilizados para o embasamento deste trabalho.

Por fim, a seção 5 (Apêndices) é dedicada aos materiais desenvolvidos durante o projeto, contendo o código do sistema em PHP e Java, assim como imagens do sistema em ambas as linguagens.

Este Relatório foi constantemente revisado e atualizado. Fazemos votos de que ele seja informativo e agregador de conhecimentos úteis dos principais conteúdos teóricos, técnicos e práticos abordados durante o semestre letivo, nas disciplinas de Programação Avançada I, Programação Orientada a Objetos, Projeto e Desenvolvimento de Sistemas e Programação para Web, assim como de conhecimentos adquiridos extra-classe.

Sumário

1. Introdução	8
1.1. Tema / Título	8
1.2. Contexto	8
2. Desenvolvimento	9
2.1. Objetivos	9
2.1.1. Objetivo geral	9
2.1.2. Objetivos específicos	9
2.2. Justificativa	100
2.3. Equipe	111
2.4. Metodologia Scrum	111
2.4.1. Apresentação	111
2.4.2. Papéis	111
2.4.3. Backlog do produto	122
2.4.4. Sprint e principais eventos	133
2.5. Kanban	175
2.6. Scrumban	15
2.7. Técnicas e ferramentas	16
2.7.1. Brainstorming	16
2.7.2. Diagrama de Ishikawa	16
2.7.3. Ciclo PDCA	16
2.7.4. MASP (Método de Análise e Solução de problemas)	16
2.8. Linguagens de programação	17
2.9. Softwares de gerenciamento de projetos	18
2.10. Plataforma de hospedagem do código desenvolvido	20
2.11. IDE Eclipse	21
2.12. Resultados	22
2.12.1. Resultados esperados	232

2.12.2. Resultados atingidos	232
3. Conclusão	23
4. Referências bibliográficas	24
5. Apêndices	28
5.1. Apêndice A - Código do sistema Web em PHP	28
5.2. Apêndice B - Tela do sistema Web em PHP	76
5.3. Apêndice C - Código do software em Java	77
5.4. Apêndice D - Tela de apresentação do sistema em Java	168

1. Introdução

1.1. Tema / Título

Sistema de gerenciamento de compra e venda de livros.

1.2. Contexto

Segundo a Pesquisa de Mercado¹ realizada pela Câmara Brasileira do Livro em parceria com o Sindicato Nacional de Editores de Livros e a Fundação Instituto de Pesquisas Econômicas, os livros dos subsetores Didáticos; Científicos, Técnicos e Profissionais; e, Obras Gerais, somados, corresponderam, em 2017, pela venda de 293.764.377 exemplares, aproximadamente, 82,76% da venda total de livros para o Mercado e o Governo. Apesar da expressividade, esse percentual poderia ser ainda maior caso houvesse uma maneira de absorver os livros publicados por autores autônomos que, geralmente, não encontram espaço em livrarias comerciais de médio e grande porte. Diante desse panorama, é pertinente o estudo e compreensão desse problema para o desenvolvimento de produtos sob encomenda para pequenos livreiros. “Estes são os sistemas encomendados por um cliente em particular.” (SOMMERVILLE, 2011, p.3). O intento é atender essa lacuna e, ao mesmo tempo, automatizar processos manuais com um sistema que ofereça usabilidade que é “medida qualitativa da facilidade e eficiência com a qual um ser humano consegue empregar as funções e os recursos oferecidos pelo produto de alta tecnologia” (PRESSMAN, 2011, p. 290) e interface consistente, que segundo PRESSMAN (2011, p. 290), é estabelecida da seguinte maneira:

A interface deve apresentar e obter informações de forma consistente. Isso implica: (1) todas as informações visuais são organizadas de acordo com regras de projeto mantidas ao longo de todas as exibições de tela, (2) mecanismos de entrada são restritos a um conjunto limitado que é usado de forma consistente por toda a aplicação e (3) mecanismos de navegação para passar de uma tarefa a outra são definidos e implementados de maneira consistente.

Pois, geralmente, muito tempo é dedicado para realizar o gerenciamento do estoque e verificar a (in)disponibilidade de livros.

¹ Pesquisa de Mercado - FIPE CBL/SNEL - Pesquisa Produção e Vendas do Setor Editorial Brasileiro, 2017.

2. Desenvolvimento

2.1. Objetivos

A seguir são apresentados o objetivo geral e os objetivos específicos.

2.1.1. Objetivo geral

O presente projeto tem por objetivo desenvolver um sistema que ofereça usabilidade e interface consistente capaz de sistematizar o gerenciamento da compra e venda de livros.

2.1.2. Objetivos específicos

O software deverá ser capaz de:

- Permitir o gerenciamento da compra, venda e estoque de livros;
- Fornecer informações à tomada de decisão;
- Automatizar procedimentos manuais;
- Otimizar os recursos empreendidos nos controles internos; e,
- Ser adaptável para se ajustar às novas mudanças e necessidades.

2.2. Justificativa

Em relação aos aspectos tecnológico e econômico, o desenvolvimento dessa solução é necessário para melhorar o processo de gerenciamento de compra e venda de obras literárias de uma pequena livreira que, atualmente, é feito de forma integralmente manual. Devido à necessidade em responder às demandas do mercado de maneira ágil, dinâmica e assertiva é vital a criação de uma solução tecnológica que contribua com a modernização e competitividade do negócio perante aos concorrentes, mediante a otimização dos recursos empreendidos durante a execução das atividades da empresa. Além disso, sob o prisma social, é importante contribuir com o desenvolvimento de um ambiente que estimule a leitura e a aquisição de conhecimentos.

2.3. Equipe

Após uma sequência de mudanças durante o primeiro e segundo semestre do curso, finalmente, houve a continuidade e consolidação da equipe anterior, constituída pelos seguintes membros: Gustavo Souza da Mota (Cofundador), Mike Jones Cabrera da Silva (Sócio-proprietário) e Tiago de Freitas Faustino (Cofundador).

2.4. Metodologia Scrum

2.4.1. Apresentação

A partir de Fevereiro de 2019, a metodologia adotada neste projeto passou a ser a Scrum, que é a mais utilizada na atualidade. Segundo *The 13th Annual State of Agile Report*², 54% das empresas entrevistadas utilizam o Scrum como metodologia ágil (COLLABNET VERSIONONE, 2019)

“Scrum is the most known Agile Project Management approach adequate for managing software development.” (CARVALHO e FERNANDES, 2018, p.45 apud SUTHERLAND, 2014)

Scrum é relativamente jovem. Segundo PRESSMAN (2011, p.95) o Scrum foi “concebido por Jeff Sutherland e sua equipe de desenvolvimento no início dos anos 1990”. Esse termo nasceu dentro do rugby e consiste numa jogada onde “everyone in the pack acts together with everyone else to move the ball down the field.” (RISING e JANOFF, 2000)

O seu uso justifica-se para atender, adequadamente, a expressiva mutabilidade que o software requer durante o seu desenvolvimento. O uso do “Scrum represents a new, more accurate way of doing software development that is based on the assumption that software is a new product every time that it is written or composed.” (SCHWABER e BEEDLE, 2002, p.viii)

2.4.2. Papéis

Nessa metodologia, os papéis em destaque exercidos pelos envolvidos no projeto são: o Scrum Master, o Product Owner (o dono do produto) e o Scrum Team (time de programadores).

Sucintamente, segundo SCHWABER e BEEDLE (2002, p.31), “the Scrum Master is a new management role introduced by Scrum. The Scrum Master is responsible for ensuring that Scrum values, practices and rules are enacted and enforced.”

Para COHN (2011, p.139) “o Scrum Master existe para ajudar a equipe em seu uso do Scrum”. Esse autor afirma ainda que “embora o Scrum Master não tenha autoridade sobre os membros da equipe Scrum, ele tem autoridade sobre o processo”.

² Segundo o website da COLLABENET VERSIONONE é a maior e mais amplamente citada pesquisa sobre tendências ágeis, melhores práticas e lições aprendidas.

Ainda segundo SCHWABER e BEEDLE (2002, p.31) “the Scrum Master is responsible for the success of Scrum.”

O Product Owner é “[...] the person who is officially responsible for the project. This person maintains the Product Backlog and ensures that it is visible to everyone.”(SCHWABER e BEEDLE, 2002, p.34)

SCHWABER e BEEDLE (2002, p.34) também atribuem responsabilidade ao dono do produto ao afirmarem que “The Product Owner is responsible for turning issues into work that the Scrum Team selects for a Sprint.”

COHN (2011, p.147) considera que o dono do produto “como a pessoa que garante que a equipe se dedique ao objetivo correto”, sendo o responsável por garantir a execução do backlog do produto.

“O dono do produto tem autoridade para definir um objetivo e modelar a visão. Ele não é apenas um gerente de produto que agora também redige requisitos e define algumas prioridades.” (COHN, 2011, p. 147 apud PICHLER, 2010)

Outro papel importante é o Scrum Team, que desenvolve seu trabalho com o objetivo de atingir o que foi estabelecido como sprint. Segundo SCHWABER e BEEDLE (2002, p.35) “a team commits to achieving a Sprint goal. The team is accorded full authority to do whatever it decides is necessary to achieve the goal.”

Nesta metodologia, os programadores desenvolvem diversas atividades para atingir os objetivos estabelecidos “[...] Eles programam. Eles analisam. Eles projetam. Eles fazem tudo o que for necessário para ajudar a equipe a concluir o trabalho combinado para uma sprint.” (COHN, 2011, p.168)

2.4.3. Backlog do produto

Após a elicitação de requisitos, é necessária a criação do backlog do produto (ou product backlog, em inglês), “que é uma lista de todas as funcionalidades desejadas que ainda não fazem parte do produto.” (COHN, 2011, p.255)

SCHWABER e BEEDLE (2002, p.32) trazem a seguinte conceituação “Product Backlog is an evolving, prioritized queue of business and technical functionality that needs to be developed into a system”. Os referidos autores também detalham o backlog do produto:

Product Backlog is initially incomplete, just an initial list of things that the product or system needs. The first Product Backlog may be a list of requirements that is gleaned from a vision document, garnered from a brainstorming session, or derived from a marketing requirements document. Sources of Product Backlog are as formal or informal as the hosting organization.”(SCHWABER e BEEDLE, 2002, p.33)

Essa lista (do backlog do produto) “[...] é mantida pelo dono do produto e colocada em ordem prioritária [...]”(COHN, 2011, p.255)

SCHWABER e BEEDLE (2002, p.33) demonstram um entendimento similar:

Product Backlog is sorted in order of priority. Top priority Product Backlog drives immediate development activities. The higher a

backlog item's priority, the more urgent it is, the more it has been thought about, and the more consensus there is regarding its value. Higher priority backlog is clearer and has more detailed specification than lower priority backlog.

Desde o início do projeto, a equipe vem observando que o backlog do produto é algo bastante mutável que, aos poucos, ganha forma. O pensamento de COHN (2011) vai ao encontro dessa observação. Segundo esse autor, “[...] o backlog do produto é altamente dinâmico; itens são adicionados, removidos e repriorizados a cada sprint à medida que conhecemos melhor o produto, os usuários, a equipe e assim por diante.” (COHN, 2011, p.255)

2.4.4. Sprint e principais eventos

Para SOMMERVILLE (2011, p.50) “Um sprint do Scrum é uma unidade de planejamento no qual o trabalho a ser feito é avaliado, os recursos para o desenvolvimento são selecionados e o software é implementado.”

A janela de tempo (*time boxing*) dos “Sprints são de comprimento fixo, normalmente duas a quatro semanas.” (SOMMERVILLE, 2011, p. 50)

SBROCCO e MACEDO (2012, p.162) concordam com esse período, pois afirmam que “[...] cada sprint deve ocorrer em um período de duas a quatro semanas [...]”.

Durante um sprint, ocorrem eventos importantes, dos quais destacam-se o Daily Scrum Meetings, Sprint Planning Meeting e Sprint Review.

Daily Scrum Meetings são reuniões diárias, de aproximadamente 15 minutos. “During the meeting, the team explains what it has accomplished since the last meeting, what it is going to do before the next meeting, and what obstacles are in its way.” (SCHWABER e BEEDLE, 2002, p.40)

“A reunião Scrum, realizada diariamente, ajuda a equipe a revelar problemas potenciais o mais cedo possível. Ela também leva à “socialização do conhecimento” [Bee99] e, portanto, promove uma estrutura de equipe auto-organizada.” (PRESSMAN, 2011, p. 96)

O Sprint Planning Meeting “[...] is a monthly meeting where the Product Owner and Team get together to discuss what will be done for the next Sprint [...]” (CHO, 2008, p.191)

SCHWABER e BEEDLE (2002, p.47) apresentam uma conceituação mais abrangente e detalhada. Para esses autores, “Customer, users, management, the Product Owner and the Scrum Team determine the next Sprint goal and functionality at the Sprint Planning meeting. The team devises the individual tasks that must be performed to build the product increment.”

O Sprint Review é um encontro onde é apresentado o que foi construído durante o sprint. É um evento muito importante onde

Management comes to the Sprint Review to see what the team has been able to build with the resources that it has been given. Customers come to the Sprint Review to see if they like what the

team has built. The Product Owner comes to the Sprint Review to see how much functionality has been built. Other engineers and developers come to the meeting to see what the team was able to do with the technology. Everyone wants to see what the team has built, what the Sprint was like, how the technology worked, what shortcuts had to be taken, what things it was able to add, and its ideas as to what can be done next. (SCHWABER e BEEDLE, 2002, p.55)

A imagem 1 apresenta a visão geral da metodologia Scrum:

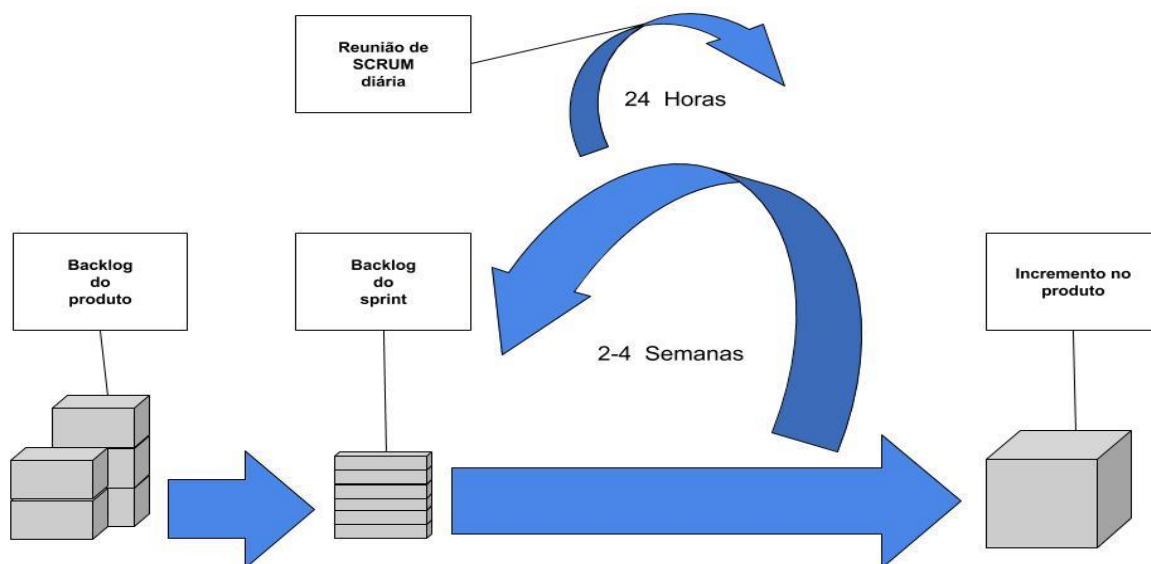


Imagem 1 - Visão geral da dinâmica da metodologia SCRUM
(SBROCCO e MACEDO, 2012, p.162 apud COHN, 2008)

2.5. Kanban

“[..]. podemos dizer que o Kanban é uma abordagem para mudança gerencial do projeto, um conceito para introduzir alterações em um ciclo de desenvolvimento de software ou gerenciamento de projetos.” (MARIOTTI, p. 7).

De acordo com Silva (et al, 2012), Kanban é um termo japonês que significa sinal visual. Uma das grandes características deste método é evidenciar os problemas existentes no processo.

Os métodos ágeis fornecem transparência sobre as atividades em andamento e concluídas, e reportam métricas com velocidade. O Kanban, no entanto, vai um passo além e dá transparência ao processo e seu fluxo, expondo gargalos, filas, variabilidade e desperdícios. Portanto, tudo que impacta no

desempenho da equipe de produção e para entrega de valor, fica explícito no modelo Kanban. (MARIOTTI, p. 7)

No sinalizador Kanban, o cartão é movido de uma fase à outra, até a atividade ser concluída, conforme imagem a seguir:

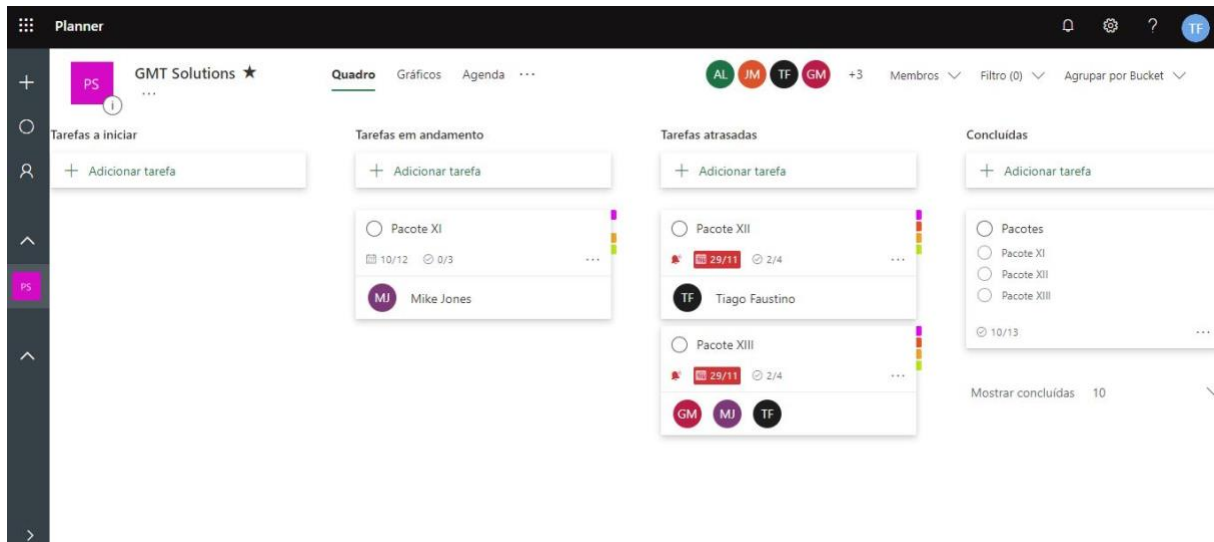


Imagem 2 - Sinalizador visual Kanban com o software Microsoft Planner

O Kanban não é uma metodologia, mas sim um framework para implementar mudanças de forma incremental. Esse é um conceito muito importante para que se entenda o Kanban como um todo já que, quando se fala em metodologias, fala-se em conjuntos de práticas e o Kanban não tem nenhuma prática prescrita. Há, nele, somente propriedades que devem guiar a melhoria no processo atual, não importando quais práticas estejam sendo usadas. (MARIOTTI, p. 13)

Durante a adoção do Kanban no desenvolvimento do projeto, “Observou-se pouca resistência ao processo de implantação, causando um impacto mínimo na forma de trabalhar de cada desenvolvedor.” (SILVA et al, 2012, p. 347). Isso se deve muito ao fato do Kanban possuir poucas prescrições. Segundo Kniberg (2009, apud SILVA et al, 2012, p. 341) o Kanban tem apenas três prescrições: Visualize o fluxo de trabalho atual; Limite o fluxo de trabalho; Acompanhe e gerencie o fluxo de trabalho.

2.6. Scrumban

Scrumban (o Scrum-ban) es una metodología derivada de los enfoques Scrum y Kanban (Pérez, 2011 apud Salvay, 2017, p.98). Esta metodología, por cierto, híbrida, contempla componentes y conceptos de ambas que se complementan entre sí, para lograr una mejor optimización del proceso de desarrollo. El término “Scrumban” fue utilizado por primera vez por Ladas (2008 apud Salvay, 2017, p.98), en su publicación

“Scrumban-Essays on Kanban System for Lean Software Development”.

Para Dal Canto (2015, p. 14) “Scrumban, is a project management agile methodology, which is obtained from a mix of Scrum and Kanban with different aspects from both methodologies (Łukasz Sienkiewicz, 2012 and C. Ladas, 2008).”

Durante o desenvolvimento deste trabalho, de forma mais acentuada na segunda metade do projeto, o Scrumban foi utilizado, combinando práticas do Scrum e do Kanban.

2.7. Técnicas e ferramentas

Durante o semestre letivo, foram utilizadas técnicas que proporcionaram uma melhor performance dos participantes. A seguir, relacionamos aquelas mais utilizadas e sua ideia central:

2.7.1. Brainstorming

Brainstorming“ [...] é uma técnica utilizada para propor soluções a um problema específico. Consiste em uma reunião também chamada de tempestade de ideias, na qual os participantes devem ter liberdade de expor suas sugestões e debater sobre as contribuições dos colegas.” (PATEL)

2.7.2. Diagrama de Ishikawa

Conhecido como espinha de peixe, geralmente é utilizado para solucionar problemas, mediante uma relação de causa-efeito.

“Los Diagramas Causa-Efecto ayudan a los estudiantes a pensar sobre todas las causas reales y potenciales de un suceso o problema, y no solamente en las más obvias o simples. Además, son idóneos para motivar el análisis y la discusión grupal, de manera que cada equipo de trabajo pueda ampliar su comprensión del problema, visualizar las razones, motivos o factores principales y secundarios, identificar posibles soluciones, tomar decisiones y, organizar planes de acción (Ishikawa, *Recuperado el 15*, 1943).

2.7.3. Ciclo PDCA

Trata-se de um ferramenta com alta aplicabilidade, muito utilizada em gestão. Esse ciclo pode ser representado com a seguinte ideia central: planejar (plan), fazer (do), checar (check) e agir (action). É um ciclo criado por

[...] Walter A. Shewart, na década de 20, mas ele se tornou conhecido quando William Edward Deming, um dos gurus da gestão de qualidade, espalhou o conceito pelo mundo. Por esse motivo, o ciclo PDCA ficou conhecido a partir da década de 1950 como ‘Ciclo Deming’. A través dessa teoria, cada processo da empresa passa por quatro fases (GESTÃO DE QUALIDADE)

2.7.4. MASP (Método de Análise e Solução de problemas)

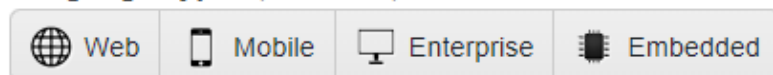
É uma metodologia que geralmente é utilizada para problemas complexos, constituída pelas etapas a saber: 1. Identificação do problema; 2. Observação; 3. Análise; 4. Plano de Ação; 5. Ação; 6. Verificação de resultados; 7. Padronização; e 8. Conclusão.

Oribe (2008, p.32 apud Cruz e Stanzani, p.3) define o MASP como sendo “um método prescritivo, racional, estruturado e sistemático para o desenvolvimento de um processo de melhoria em um ambiente organizacional, visando solução de problemas e obtenção de resultados otimizados”.

2.8. Linguagens de programação e banco de dados

A linguagem utilizada no desenvolvimento do programa neste semestre é o JAVA. Para a criação de uma plataforma web dinâmica utilizamos o PHP. De acordo com o website IEEE Spectrum, considerando os ambientes web, mobile, enterprise e embedded, ambas linguagens estão entre as mais utilizadas no ano de 2018, ocupando, respectivamente, a terceira e sexta posição do referido ranking, conforme a seguir:

Language Types [\(click to hide\)](#)



Language Rank	Types	Spectrum Ranking
1. Python	Web, Enterprise, Embedded	100.0
2. C++	Mobile, Enterprise, Embedded	99.7
3. Java	Web, Mobile, Enterprise	97.5
4. C	Mobile, Enterprise, Embedded	96.7
5. C#	Web, Mobile, Enterprise	89.4
6. PHP	Web	84.9
7. R	Enterprise	82.9
8. JavaScript	Web, Mobile	82.6
9. Go	Web, Enterprise	76.4
10. Assembly	Embedded	74.1

Imagem 3 - Spectrum Ranking: The Top Programming Languages 2018

Ambas as linguagens foram suportadas com dados provenientes de um serviço de banco de dados relacional do Google, o Cloud SQL. Segundo as informações contidas no website da referida empresa,

O Cloud SQL é um serviço de banco de dados totalmente gerenciado que facilita a configuração, a manutenção, o gerenciamento e a administração de bancos de dados relacionais PostgreSQL, MySQL e SQL Server na nuvem. O Cloud SQL oferece altos níveis de desempenho, alta disponibilidade, escalabilidade e conveniência. Criado em uma infraestrutura moderna, com uma rede global privada do Google e segurança de alto nível, o Cloud SQL cuida das tarefas rotineiras do gerenciamento de bancos de dados [...] (Google, 2019)

A imagem a seguir apresenta o ambiente do Google Cloud Platform, onde os bancos de dados encontram-se mantidos:

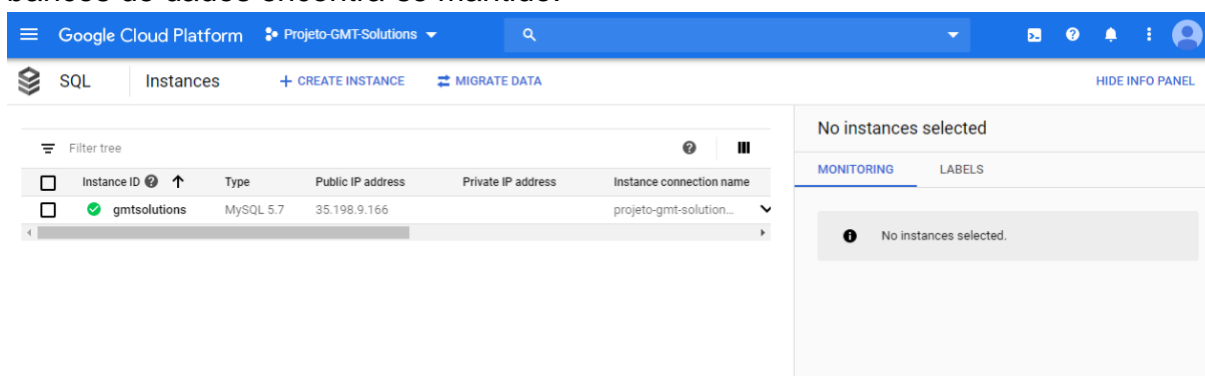


Imagem 4 - Banco de dados - Google Cloud Platform

2.9. Softwares de gerenciamento de projetos

Para o gerenciamento deste projeto foram utilizados dois softwares da Microsoft Corporation, quais sejam: Microsoft Project e Microsoft Planner.

A Microsoft Project Professional 2016 foi a versão utilizada pela equipe. Neste trabalho, esse software de gerenciamento de projetos foi configurado de modo a contemplar o período de duração do projeto, a data de seu início e término, o quantitativo de horas trabalhadas diariamente, os programadores que irão desenvolver o produto, o valor do dia de trabalho de cada programador, as funcionalidades e pacotes a serem realizados, seu período de duração com data de início e término e a vinculação entre as atividades. A imagem a seguir apresenta uma visão do Project:

	i	EDT	Nome da Tarefa	% con	Custo	Duraçã	Início	Término	F	Nomes dos recursos
0	✓	0	PROJETO_ATUALIZADO_EM_10_12_2019	100%	R\$13.493,76	91,62 di	Seg 05/08/19	Ter 10/12/19		
1	✓	1	Pacote I	100%	R\$1.021,44	7 dias	Seg 05/08/19	Ter 13/08/19		
2	✓	1.1	Requisitos PPW	100%	R\$322,56	2 dias	Seg 05/08/19	Ter 06/08/19		Gustavo;Mike;Tiago
3	✓	1.2	Requisitos PA1	100%	R\$322,56	2 dias	Qua 07/08/19	Qui 08/08/19	2	Gustavo;Mike;Tiago
4	✓	1.3	Requisitos POO	100%	R\$322,56	2 dias	Sex 09/08/19	Seg 12/08/19	3	Gustavo;Mike;Tiago
5	✓	1.4	Sprint Review 1	100%	R\$53,76	1 dia	Ter 13/08/19	Ter 13/08/19	4	Gustavo;Mike;Tiago
6	✓	2	Pacote II	100%	R\$1.612,80	11 dias	Qua 14/08/19	Qua 28/08/19		
7	✓	2.1	Diagrama de Classes BDD	100%	R\$483,84	3 dias	Qua 14/08/19	Sex 16/08/19	5	Gustavo;Mike;Tiago
8	✓	2.2	Banco de Dados: MER (Modelo Entidade Relacionamento) BDD	100%	R\$483,84	3 dias	Seg 19/08/19	Qua 21/08/19	7	Gustavo;Mike;Tiago
9	✓	2.3	Banco de Dados: Script BDD	100%	R\$483,84	3 dias	Qui 22/08/19	Seg 26/08/19	8	Mike;Tiago;Gustavo
10	✓	2.4	Pesquisar template da plataforma web PPW	100%	R\$161,28	1 dia	Ter 27/08/19	Ter 27/08/19	9	Gustavo;Mike;Tiago
11	✓	2.5	Sprint Review 2	100%	R\$0,00	1 dia	Qua 28/08/19	Qua 28/08/19	10	
12	✓	3	Pacote III	100%	R\$2.741,76	18 dias	Qui 29/08/19	Seg 23/09/19		
19	✓	4	Pacote IV	100%	R\$1.397,76	26 dias	Ter 24/09/19	Ter 29/10/19		
26	✓	5	Pacote V	100%	R\$1.397,76	26 dias	Ter 24/09/19	Ter 29/10/19		
33	✓	6	Pacote VI	100%	R\$860,16	16 dias	Ter 24/09/19	Ter 15/10/19		
38	✓	7	Pacote VII	100%	R\$322,56	6 dias	Qua 30/10/19	Qua 06/11/19		
41	✓	8	Pacote VIII	100%	R\$860,16	16 dias	Qua 30/10/19	Qui 21/11/19		
46	✓	9	Pacote IX	100%	R\$860,16	16 dias	Qua 16/10/19	Qua 06/11/19		
51	✓	10	Pacote X	100%	R\$322,56	6 dias	Qui 07/11/19	Qui 14/11/19		
54	✓	11	Pacote XI	100%	R\$698,88	13 dias	Sex 22/11/19	Ter 10/12/19		
58	✓	12	Pacote XII	100%	R\$860,16	16 dias	Qui 07/11/19	Sex 29/11/19		
63	✓	13	Pacote XIII	100%	R\$537,60	10 dias	Seg 18/11/19	Sex 29/11/19		

Imagem 5 - Tela do projeto no software Microsoft Project Professional 2016

O Microsoft Planner também é um software de gerenciamento de projetos que visa, dentre outras coisas, organizar melhor o trabalho da equipe, compartilhar arquivos, atribuir tarefas e obter atualizações sobre o andamento das atividades através de quadros, gráficos e de uma agenda.

Sua descoberta se deu mediante uma atividade proposta pelos alunos do curso, no qual cada aluno deveria pesquisar uma ferramenta de gerenciamento de projetos que pudessem auxiliá-los em seu trabalho. Cada aluno pesquisou as ferramentas disponíveis no mercado e as apresentou aos demais colegas do curso. Dentre as opções exibidas, o Microsoft Planner se destacou e, portanto, foi a opção escolhida.

A sua utilização agregou muito valor durante o desenvolvimento deste trabalho, motivando a autonomia dos integrantes da equipe na execução das tarefas. As imagens a seguir apresentam o Planner:

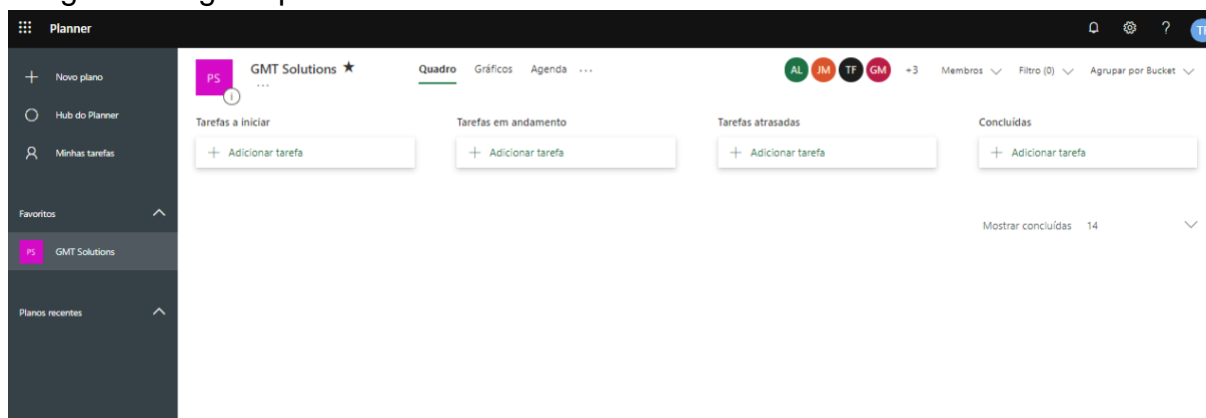


Imagem 6 - Quadros no software Microsoft Planner

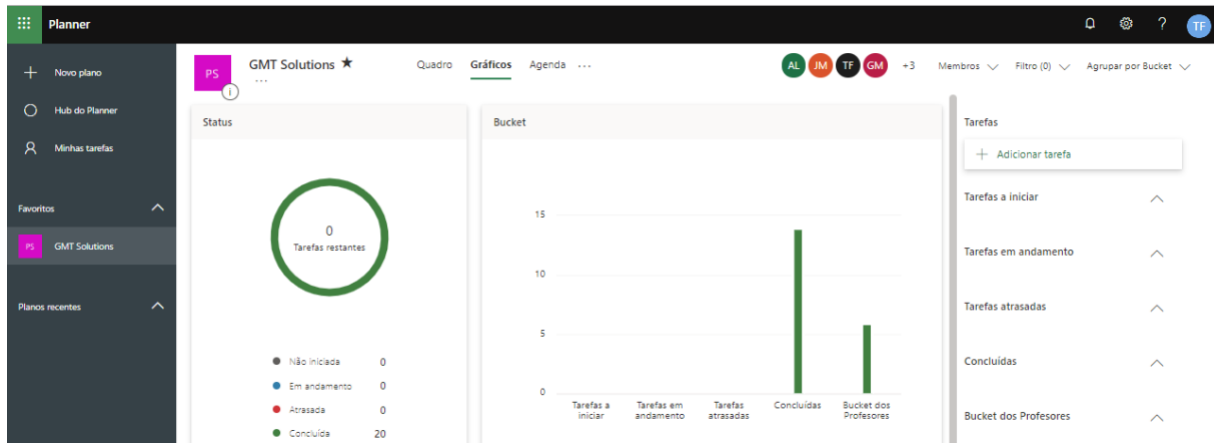


Imagem 7 - Gráficos no software Microsoft Planner

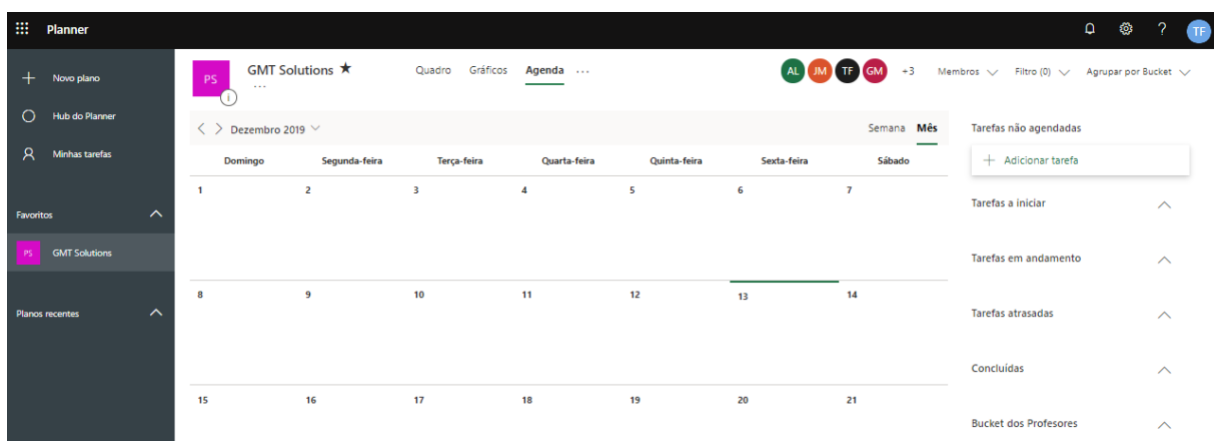


Imagem 8 - Agenda no software Microsoft Planner

2.10. Plataforma de hospedagem do código desenvolvido

A plataforma de hospedagem do código-fonte adotada foi o GitHub, utilizando o Git.

Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou [Open Source](#) de qualquer lugar do mundo. **GitHub** é amplamente utilizado por programadores para divulgação de seus trabalhos ou para que outros programadores contribuam com o projeto, além de promover fácil comunicação através de recursos que relatam problemas ou mesclam [repositórios remotos](#) (*issues, pull request*). (Wikipedia, 2019)

A imagem a seguir exibe o código hospedado pela equipe durante o desenvolvimento do projeto:

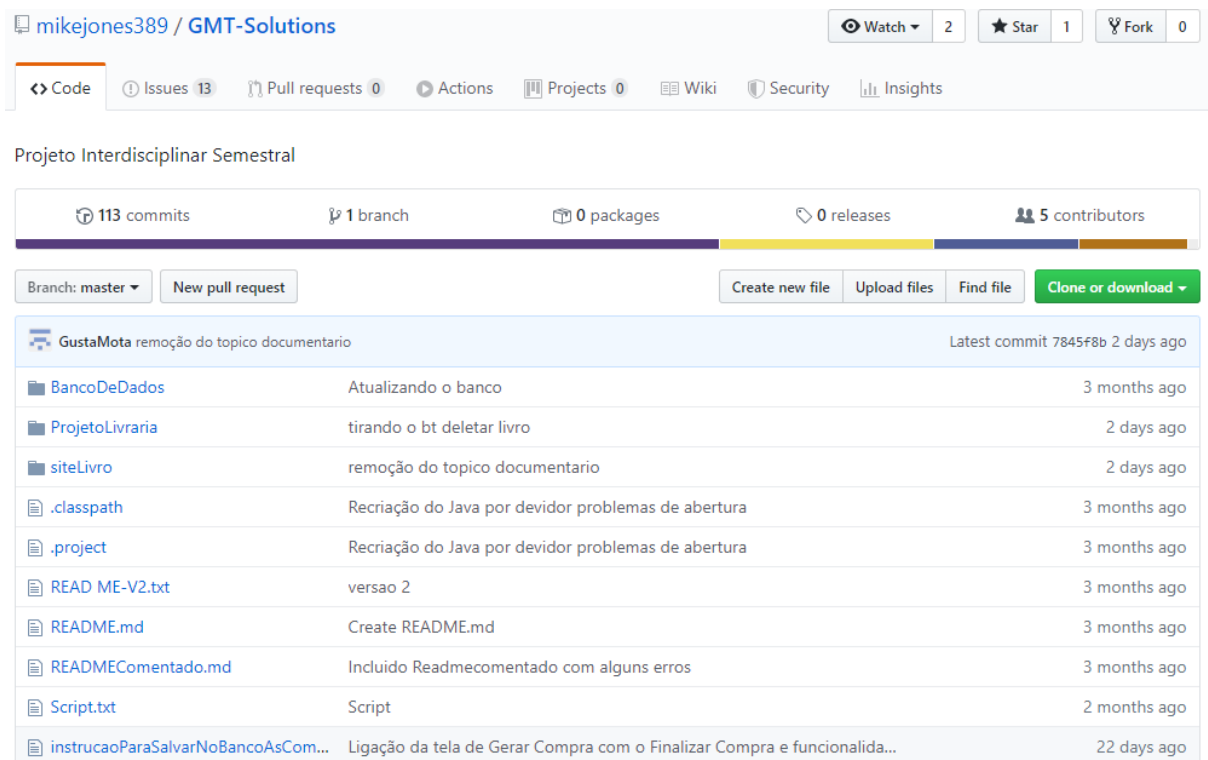


Imagem 9 - Código hospedado na plataforma GitHub

2.11. IDE Eclipse

A IDE utilizada no desenvolvimento do programa foi o Eclipse. Segundo o site Wikipedia (2019)

Eclipse é um IDE para desenvolvimento Java, porém suporta várias outras linguagens a partir de plugins como C/C++,PHP,ColdFusion, Python, Scala e plataforma Android. Ele foi feito em Java e segue o modelo open source de desenvolvimento de software. Atualmente faz parte do kit de desenvolvimento de software recomendado para desenvolvedores Android^[1].

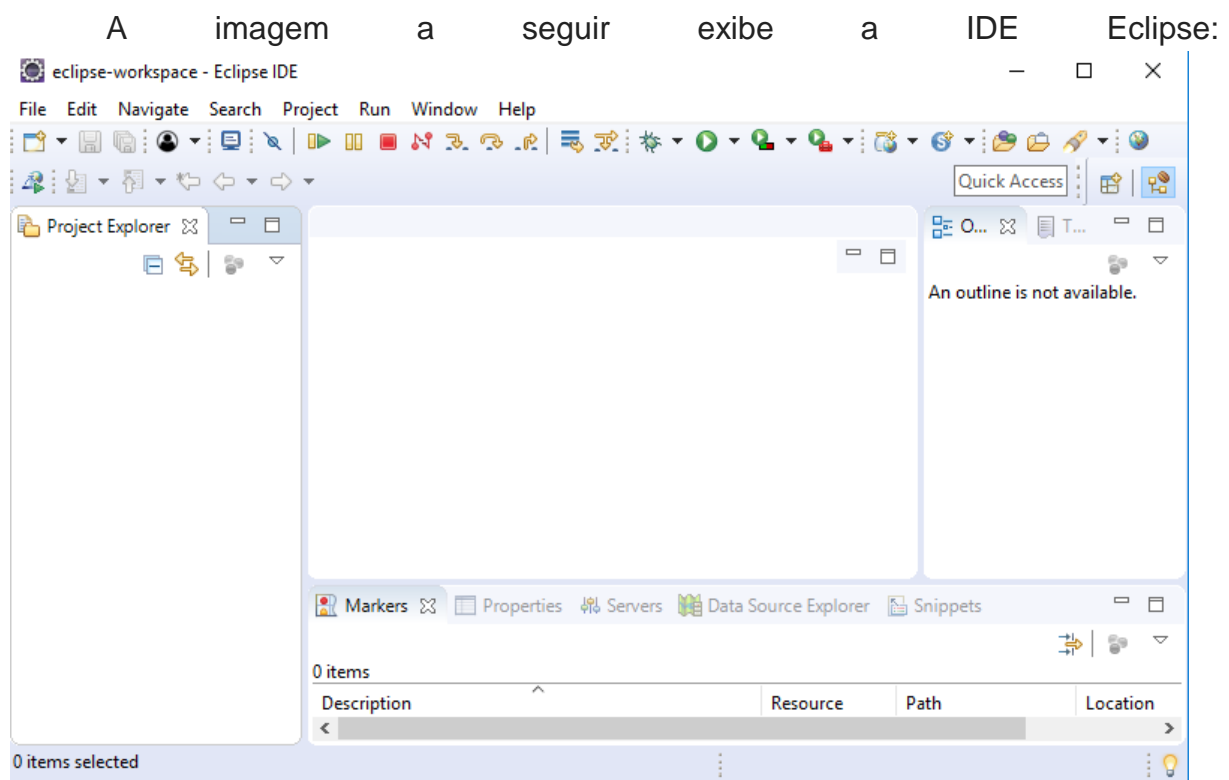


Imagem 10 - IDE Eclipse

2.12. Resultados

2.12.1. Resultados esperados

Ao final do projeto, esperamos apresentar um software sob encomenda, desenvolvido nas linguagens Java e PHP, que atenda ao definido na em “Objetivos”.

2.12.2. Resultados atingidos

Ao final do projeto foram obtidos os seguintes resultados, dentro do prazo estipulado:

- Sistema de gerenciamento de compra e venda de livros, desenvolvido em linguagem de programação Java, com 13 pacotes e 3718 linhas de código;
- Plataforma web, desenvolvida em PHP, com 2811 linhas de código;
- Banco de dados SQL Server no Google Cloud Platform;
- Reduções: do prazo inicial do projeto, em 36 dias; e do valor total do projeto, em R\$ 5967,36,00.

3. Conclusão

Esse semestre esteve cercado por momentos desafiadores, de dificuldades e, também, de muito aprendizado. A equipe lidou com atividades de alta complexidade em diversas situações, tendo que realizar as melhores escolhas dentre as alternativas disponíveis, prevalecendo os interesses da equipe em detrimento dos interesses individuais.

O principal momento de dificuldade da equipe esteve concentrado no primeiro quartil do desenvolvimento do projeto, até meados de setembro. A partir daí houve uma evolução maior, mas nada comparado com a passagem de outubro para novembro em diante. Nesse período, a curva de aprendizagem foi muito significativa. Observamos que isso ocorreu devido à familiarização dos integrantes com as linguagens de programação e o emprego de ferramentas, estratégias e técnicas de gestão de projetos que aumentaram em muito a eficiência produtiva do grupo. A fator tempo também pressionou a equipe para a conclusão do projeto.

Esse feito foi possível devido à especialização dos integrantes com atividades dedicadas e de maior afinidade. Com isso, o volume de entregas foi maior, permitindo a conclusão do projeto dentro do prazo estabelecido, no entanto, houveram, também, algumas desvantagens, uma delas é que o resultado da dedicação exclusiva eleva situações de estresse, pelo peso e responsabilidade na entrega.

Acreditamos que todo o aprendizado adquirido foi possível graças ao ambiente favorável constituído ao exercício da autonomia dos integrantes, facilitando a busca pelo saber e o exercício do protagonismo de um aprendizado ativo na busca de soluções, seja mediante a troca de experiências com a outra equipe, seja por meio das orientações recebidas dos professores.

Enfim, este semestre foi bastante enriquecedor.

4. Referências bibliográficas

PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional**. 7. ed. Porto Alegre: AMGH: Bookman, 2011.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Education do Brasil, 2011.

Câmara Brasileira do Livro - CBL. **Pesquisa de Mercado. Pesquisa Produção e Vendas do Setor Editorial Brasileiro**, 2017. Disponível em: <<http://cbl.org.br/downloads/fipe>> Acesso em 22 de set. 2018, 16:10:05.

SCHWABER, Ken; BEEDLE, Mike. **Agile Software Development with Scrum**. Upper Saddle River, NJ 07458: Prentice Hall, 2002.

COHN, Mike. **Desenvolvimento de software com Scrum: aplicando métodos ágeis com sucesso**. Porto Alegre: Bookman, 2011.

SBROCCO, José Henrique Teixeira de Carvalho; MACEDO, Paulo Cesar de. **Métodos ágeis: engenharia de software sob medida**. 1. ed. São Paulo: Érica, 2012.

VersionOne 13th Annual State of Agile Report. 2019. **The 13th Annual State of Agile Report**. Disponível em

<https://www.stateofagile.com/?_ga=2.152326577.907185782.1575850208-608310929.1575850208#ufh-i-521251909-13th-annual-state-of-agile-report/473508>

Acesso em 09 de dez. 2019, 05:09:08

RISING, Linda; JANOFF, Norman S. IEEE Software. **The scrum software development process for small teams**. 2000. Disponível em: <<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=854065>> Acesso em 23 de mar. 2019, 23:50:18.

CARVALHO, José Dinis; FERNANDES, Sandra. Repositório da Universidade Portucalense. **How to apply SCRUM in PBL teams**. 2018. Disponível em: <http://repositorio.uportu.pt:8080/bitstream/11328/2534/1/2018_PAEE_WorkshopScrum_CarvalhoFernandes.pdf> Acesso em 24 de mar. 2019, 01:38:02.

CHO, Juyun. Colorado State University-Pueblo. **Issues and Challenges of Agile Software Development with Scrum**. 2008. Disponível em: <<https://pdfs.semanticscholar.org/713c/41d6f352f97a94af1af94c56e0ba2584effd.pdf>> Acesso em 24 de mar. 2019, 09:56:14.

Google Patents. Pub. No.: US 2007/0130537 A1. **Single page website interface**. 2007. Disponível em: <<https://patentimages.storage.googleapis.com/ad/78/ae/3e067c1b360f43/US20070130537A1.pdf>> Acesso em 19 de abr. 2019, 16:58:51.

TOLEDO, Cláudio Fabiano Motta. **IDEs e Programação**. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/349201/mod_resource/content/1/%20IDE%20e%20Programa%C3%A7%C3%A3o.pdf> Acesso em 19 de abr. 2019, 19:45:37.

PEREIRA, Silvio do Lago. **Linguagem C**. Disponível em: <<https://www.ime.usp.br/~slago/slago-C.pdf>> Acesso em 19 de abr. 2019, 20:51:16.

BORGES, Eduardo. **Métodos ágeis e documentação de software**. Youtube. 4 nov. 2013. 10min42s. Disponível em: <<https://www.youtube.com/watch?v=3Smbhnmue7Y&t=211s>>. Acesso em: 20 abr. 2019, 01:16:08.

BARCELLOS, Monalessa Perini. Universidade Federal do Espírito Santo, Centro Tecnológico. Departamento de Informática. **Gerência de Projetos**. 2014. Disponível em: <<https://nemo.inf.ufes.br/wp-content/uploads/Monalessa/GEP/Slides-Scrum-VisaoGeral.pdf>> Acesso em 20 de abr. 2019, 02:15:43.

MARQUES, José Roberto. Instituto Brasileiro de Coaching - IBC. **O que é know how?** 2018. Disponível em: <<http://www.ibccoaching.com.br/portal/vida-profissional/o-que-e-know-how/>> Acesso em 02 de mai. 2019, 13:12:23.

GUEDES, Gilleanes T.A. **UML 2: uma abordagem prática**. 2. ed. São Paulo: Novatec Editora, 2011.

MARIOTTI, Flávio S. Engenharia de Software Magazine. **Kanban: o ágil adaptativo**. Edição 45, p. 6-10. Disponível em: <<http://www.garcia.pro.br/EngenhariadeSW/artigosMA/A6%20-%2045-6-%20Kanbam.pdf>> Acesso em 09 de dez. 2019, 05:17:18.

MARIOTTI, Flávio S. Engenharia de Software Magazine. **Kanban no desenvolvimento de projetos de software**. Edição 45, p. 11-16. Disponível em: <<http://www.garcia.pro.br/EngenhariadeSW/artigosMA/A6%20-%2045-6-%20Kanbam.pdf>> Acesso em 09 de dez. 2019, 05:17:18.

SILVA, Diogo Vinícius de S.; SANTOS, F. Alan de O.; NETO, Pedro Santos. VIII Simpósio Brasileiro de Sistemas de Informação (SBSI). **Os benefícios de uso do Kanban na gerência de projetos de manutenção de software**. 2012. Disponível

em: <<http://www.lbd.dcc.ufmg.br/colecoes/sbsi/2012/0035.pdf>> Acesso em 09 de dez. 2019, 05:45:02.

SALVAY, Javier E. Instituto Universitario Aeronautico. **Kanban y Scrumban orientados a Proyectos de Tecnología de la Información**. 2017. Disponível em: <<https://rdu.iaa.edu.ar/bitstream/123456789/880/1/Proyecto%20de%20Grado%20-%20Kanban%20y%20Scrumban%20-%20Javier%20Salvay.pdf>> Acesso em 09 de dez. 2019, 06:29:32.

DAL CANTO, Camilla. Dipartimento di Ingegneria Dell'Energia Dei Sistemi, Del Territorio e Delle Costruzioni. **Using Scrumban Method for Production Planning and Control in Manufacturing Companies**. 2015. Disponível em: <<https://core.ac.uk/download/pdf/79619004.pdf>> Acesso em 09 de dez. 2019, 06:44:52.

PATEL, Neil. **Brainstorming: O Que É, Como Fazer (Passo a Passo)**. Disponível em: <<https://neilpatel.com/br/blog/o-que-e-brainstorming/>> Acesso em 09 de dez. 2019, 20:21:37.

ISHIKAWA, Kaoru. **Diagrama Causa-Efeito**. Recuperado el, v. 15, 1943. Disponível em: <https://scholar.google.com.br/scholar?cluster=7971027371422286382&hl=pt-BR&as_sdt=2005&scioldt=0,5> Acesso em 09 de dez. 2019, 19:52:35.

SANTOS, Virgilio F. M. dos. FMS2. Blog Melhoria de Processos. **O que é método MASP?** 2018. Disponível em: < <https://www.fm2s.com.br/metodo-masp/> > Acesso em 09 de dez. 2019, 21:12:12.

IEEE Spectrum. **Interactive: The Top Programming Languages 2018**. Disponível em:<https://spectrum.ieee.org/ns/IEEE_TPL_2018/index/2018/1/1/1/1/1/50/1/50/1/50/1/30/1/30/1/30/1/20/1/20/1/5/1/5/1/20/1/100/> Acesso em 13 de dez. 2019, 19:27:33.

GITHUB. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2019. Disponível em: <<https://pt.wikipedia.org/w/index.php?title=GitHub&oldid=56762945>>. Acesso em 12 de dez. 2019, 20:31:12.

ECLIPSE (SOFTWARE). In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2019. Disponível em: <[https://pt.wikipedia.org/w/index.php?title=Eclipse_\(software\)&oldid=56092970](https://pt.wikipedia.org/w/index.php?title=Eclipse_(software)&oldid=56092970)>. Acesso em: 13 de dez. 2019, 20:42:01.

Google Cloud. **Cloud SQL**. Disponível em:<<https://cloud.google.com/sql/>> Acesso em 13 de dez. 2019, 21:58:50.

5. Apêndices

5.1. Apêndice A - Código do sistema web em PHP

Arquivo
about.php

```
<?php require_once "i_topo.php"; ?>

<!-- Title page --> <section class="bg-img1 txt-center p-lr-15 p-tb-92" style="background-image: url('images/contato.png');"> <h2 class="ltext-105 cl0 txt-center">

    Sobre </h2> </section> <!-- Content page --> <section class="bg0 p-t-75 p-b-120">

    <div class="container">

        <div class="row p-b-148">

            <div class="col-md-7 col-lg-8">

                <div class="p-t-7 p-r-85 p-r-15-lg p-r-0-md">

                    <h3 class="mtext-111 cl2 p-b-16">

                        Nossa História </h3> <p class="stext-113 cl6 p-b-26">

                            A GMT Group foi fundada em 2018 com um grupo de amigos do IFSP Campinas com o objetivo de impactar o setor livreiro. No início de 2019, houve uma reformulação da composição da equipe, que é atualmente formada pelos Cofundadores Gustavo Souza da Mota, Tiago Freitas Faustino e pelo Sócio-Proprietário Mike Jones Cabrera da Silva. </p> </div> </div>

                <div class="col-11 col-md-5 col-lg-4 m-lr-auto">

                    <div class="how-bor1 ">

                        <div class="hov-img0">

                             </div> </div> </div> </div>

            <div class="row">

                <div class="order-md-2 col-md-7 col-lg-8 p-b-30"> <div class="p-t-7 p-l-85 p-l-15-lg p-l-0-md">
```

```
<h3 class="mtext-111 cl2 p-b-16">
```

```
Nossa Missão </h3> <p class="stext-113 cl6 p-b-26">
```

A GMT Group tem como missão incluir autores autônomos no mercado livreiro, dando visibilidade

para a venda de livros, pois esse mercado é constituído por um oligopólio de poucas editoras

```
que vem ditando as regras do mercado. </p> <div class="bor16 p-l-29 p-b-9 m-t-22">
```

```
<p class="stext-114 cl6 p-r-40 p-b-11">
```

Creativity is just connecting things. When you ask creative people how they did something, they feel a little guilty because they didn't really do it, they just saw something. It seemed obvious to them after a while.

```
</p> <span class="stext-111 cl8">
```

```
- Steve Job's </span> </div> </div> </div> <div class="order-md-1 col-11 col-md-5 col-lg-4 m-lr-auto p-b-30">
```

```
<div class="how-bor2">
```

```
<div class="hov-img0">
```

```
 </div> </div> </div> </div>
```

```
</div> </section> <!-- Footer --> <?php  
require_once "i_footer.php"; ?>
```

```
<!--
```

```
=====
```

```
=== ----->
```

```
<script src="vendor/jquery/jquery-3.2.1.min.js"></script> <!--
```

```
=====
```

```
=== ----->
```

```
<script src="vendor/ansition/js/ansition.min.js"></script> <!--
```

```
=====
```

```
=== ----->
```

```
<script src="vendor/bootstrap/js/popper.js"></script> <script  
src="vendor/bootstrap/js/bootstrap.min.js"></script> <!--
```

```
=====
```

```
=== ----->
```

```
<script src="vendor/select2/select2.min.js"></script> <script>
```

```

        $(".js-select2").each(function(){
            $(this).select2({
minimumResultsForSearch: 20, dropdownParent: $(this).next('.dropDownSelect2')
}); }) </script> <!--
=====
=== ----->

<script src="vendor/MagnificPopup/jquery.magnific-popup.min.js"></script> <!--
=====
=== ----->

<script src="vendor/perfect-scrollbar/perfect-scrollbar.min.js"></script>
<script>
    $(''.js-pscroll').each(function(){
        $(this).css('position','relative');
        $(this).css('overflow','hidden'); var ps = new
        PerfectScrollbar(this, {
            wheelSpeed: 1,
            scrollingThreshold: 1000,
            wheelPropagation: false,
        });

        $(window).on('resize', function(){
            ps.update(); }) }); </script> <!--
=====
=== ----->

<script src="js/main.js"></script>

</body>
</html>

```

Arquivo
cadastroDesejo.php

```

<?php require_once
"conexaoBD.php";

$listaDesejo = 0; $usuario =
$_POST['cd_usuario']; $livro =

```

```
$_POST['cdLivro']; $acao =  
$_POST['acao'];
```

```
if ($acao == "adicionar") { $sql="insert into item_desejado  
(cd_listaDesejo,cd_usuario,cd_livro) values ('$listaDesejo', '$usuario', '$livro')"; }  
else{ $sql="delete from item_desejado where cd_usuario = ".$usuario." and  
cd_livro = ".$livro;
```

```
} if (mysqli_query($db, $sql)) {  
    echo "New record created successfully"; } else  
{  
    echo "Error: " . $sql . "<br>" . mysqli_error($db); }
```

```
header("Location:product.php");
```

```
?> Arquivo conexaoBD.php
```

```
<?php $db = mysqli_connect('35.198.9.166:3306', 'root','@gmtifsp',  
'projeto_livraria'); if (!$db) { echo 'Não deu para conectar ao Banco de Dados'; exit; } ?>
```

Arquivo contact.php

```
<?php require_once "i_topo.php"; ?>
```

```
<!-- Title page --> <section class="bg-img1 txt-center p-lr-15 p-tb-  
92" style="background-image: url('images/contato.png');"> <h2  
class="ltext-105 cl0 txt-center">
```

```
Contato </h2> </section> <!-- Content page
```

```
--> <section class="bg0 p-t-104 p-b-116">
```

```
<div class="container">
```

```
<div class="flex-w flex-tr">
```

```
<div class="size-210 bor10 p-lr-70 p-t-55 p-b-70 p-lr-15-lg w-  
full-md"> <form> <h4 class="mtext-105 cl2 txt-center p-b-30">
```

```
Mande uma Mensagem à nós </h4> <div class="bor8 m-b-20 how-pos4-  
parent">
```

```
<input class="stext-111 cl2 plh3 size-116 p-l-62  
p-r-30" type="text" name="email" placeholder="Seu Endereço de Email">
```

```
<img class="how-pos4 pointer-none"
```


`src="images/icons/icon-email.png" alt="ICON">`

`</div>`

`<div class="bor8 m-b-30">`

`<textarea class="stext-111 cl2 plh3 size-120 p-lr-28 p-tb-25" name="msg" placeholder="Digite seu comentário"></textarea>`

`</div>`

`<button class="flex-c-m stext-101 cl0 size-121 bg3 bor1 hov-btn3 p-lr-15 trans-04 pointer">`

Enviar `</button> </form> </div> <div class="size-210 bor10 flex-w flex-col-m p-lr-93 p-tb-30 p-lr-15-lg w-full-md">`

`<div class="flex-w w-full p-b-42">`

` <div class="size-212 p-t-2">`

``

Endereço ` <p class="stext-115 cl6 size-213 p-t-18">`

Av. Comendador Aladino Selmi, S/N, CTI Renato Archer, Bairro Amarais -
Campinas/SP - CEP: 13069-901 `</p> </div> </div> <div class="flex-w w-full p-b-42">`

``

` <div class="size-212 p-t-2">`

``

Vamos Conversar ` <p class="stext-115 cl1 size-213 p-t-18">`

+55 19 3746-6128 `</p> </div> </div> <div class="flex-w w-full">`

``

` <div class="size-212 p-t-2">`

``

Suporte Comercial ` <p class="stext-115 cl1 size-213 p-t-18">`

gmtstore@gmail.com `</p> </div> </div> </div> </div>`

```

</div> </section>

<!-- Footer --> <?php require_once
    "i_footer.php"; ?>

<!--
=====
=== ----->

<script src="vendor/jquery/jquery-3.2.1.min.js"></script> <!--
=====
=== ----->

<script src="vendor/ansition/js/ansition.min.js"></script> <!--
=====
=== ----->

<script src="vendor/bootstrap/js/popper.js"></script>
<script src="vendor/bootstrap/js/bootstrap.min.js"></script> <!--
=====
=== ----->

<script src="vendor/select2/select2.min.js"></script> <script>
    $(".js-select2").each(function(){
        $(this).select2({
minimumResultsForSearch: 20, dropdownParent: $(this).next('.dropDownSelect2')
}); }) </script> <!--
=====
=== ----->

<script src="vendor/MagnificPopup/jquery.magnific-popup.min.js"></script> <!--
=====
=== ----->

<script src="vendor/perfect-scrollbar/perfect-scrollbar.min.js"></script>
<script>
    $(''.js-pscroll').each(function(){
        $(this).css('position','relative');
        $(this).css('overflow','hidden'); var ps = new
        PerfectScrollbar(this, {
            wheelSpeed: 1,
            scrollingThreshold: 1000,
            wheelPropagation: false, });
    });

```

```

$(window).on('resize', function(){
    ps.update(); }) }); </script> <!--
=====
=== ----->

<script
src="https://maps.googleapis.com/maps/api/js?key=AlzaSyAKFWBqIKAGCeS1rM
VoaNIwyayu0e0YRes"></script>
<script src="js/map-custom.js"></script> <!--
=====
=== ----->

<script src="js/main.js"></script>

</body>
</html>

```

Arquivo
i_footer.php

```

<footer class="bg3 p-t-75 p-b-32">
    <div class="container">
        <div class="row">
            <div class="col-sm-6 col-lg-3 p-b-50"> <p
class="stext-107 cl7 size-201">
                Alguma pergunta? Entre em contato </p> </div> <div
class="col-sm-6 col-lg-3 p-b-50">
                    <h4 class="stext-301 cl0 p-b-30"> </h4> <form>
                        <div class="wrap-input1 w-full p-b-4">
                            <input class="input1 bg-none plh1 stext-107 cl7"
type="text" name="email" placeholder="gmtgroup@gmail.com">
                        </div> </div> </form> </div> </div> <div
class="p-t-40">
                            <div class="flex-c-m flex-w p-b-18"> <a
href="#" class="m-all-1">
                                
                            </a> <a href="#" class="m-all-1">
                                
    </a>
    <a href="#" class="m-all-1">
        
    </a> <a href="#" class="m-all-1">
        
    </a> <a href="#" class="m-all-1">
        
    </a> </div> <p class="stext-107 cl6 txt-center">
        <!-- Link back to Colorlib can't be removed. Template is
licensed under CC BY 3.0. --> Copyright &copy;<script>document.write(new
Date().getFullYear());</script> All rights reserved | This template is made with <i
class="fa fa-heart-o" aria-hidden="true"></i> by <a href="https://colorlib.com"
target="_blank">Colorlib</a> <!-- Link back to Colorlib can't be removed. Template
is licensed under CC BY 3.0. -->
    </p> </div> </div>
    </footer>

    <!-- Back to top --> <div class="btn-back-to-top"
id="myBtn">
        <span class="symbol-btn-back-to-top">
            <i class="zmdi zmdi-chevron-up"></i> </span>
        </div>

```

Arquivo
i_topo.php

```

<?php if(!isset($_SESSION)){
    session_start(); } //echo
    substr(md5("369"),0,15);
//exit; ?>

<!DOCTYPE html> <html
lang="pt-br"> <head> <title>GMT
Store</title>

```

```

<meta charset="UTF-8"> <meta name="viewport" content="width=device-width,
initial-scale=1"> <!--
=====
=== ----->

    <link rel="icon" type="image/png" href="images/icons/favicon.png"/> <!--
=====
=== ----->

    <link rel="stylesheet" type="text/css"
href="vendor/bootstrap/css/bootstrap.min.css"> <!--
=====
=== ----->

    <link rel="stylesheet" type="text/css" href="fonts/font-awesome-4.7.0/css/font-
awesome.min.css"> <!--
=====
=== ----->

    <link rel="stylesheet" type="text/css" href="fonts/ionic/css/material-design-
ionic-font.min.css"> <!--
=====
=== ----->

    <link rel="stylesheet" type="text/css" href="fonts/linearicons-v1.0.0/icon-
font.min.css"> <!--
=====
=== ----->

    <link rel="stylesheet" type="text/css" href="vendor/animate/animate.css"> <!--
=====
=== ----->

    <link rel="stylesheet" type="text/css" href="vendor/css-
hamburgers/hamburgers.min.css"> <!--
=====
=== ----->

    <link rel="stylesheet" type="text/css"
href="vendor/animsition/css/animsition.min.css"> <!--
=====
=== ----->

    <link rel="stylesheet" type="text/css" href="vendor/select2/select2.min.css"> <!--
-
=====
=== ----->

    <link rel="stylesheet" type="text/css"
href="vendor/daterangepicker/daterangepicker.css"> <!--

```

```

=====
== ----->

<link rel="stylesheet" type="text/css" href="vendor/slick/slick.css"> <!--
=====
== ----->

<link rel="stylesheet" type="text/css" href="vendor/MagnificPopup/magnific-
popup.css"> <!--
=====
== ----->

<link rel="stylesheet" type="text/css" href="vendor/perfect-scrollbar/perfect-
scrollbar.css"> <!--
=====
== ----->

<link rel="stylesheet" type="text/css" href="css/util.css"> <link rel="stylesheet"
type="text/css" href="css/main.css"> <!--
=====
== -----> </head> <body class="animsition">

<!-- Header --> <header
class="header-v4">

    <!-- Header desktop --> <div
class="container-menu-desktop">

        <!-- Topbar --> <div
class="top-bar">

            <div class="content-topbar flex-sb-m h-full container">

                <div class="left-top-bar"> </div> <div
class="right-top-bar flex-w h-full">
                    <a target="_blank" href="FAQ.pdf"
class="flex-c-m trans-04 p-lr-25">
                        Ajuda & FAQs </a>

                <?php if (empty($_SESSION)){ ?> <a href="indexLogin.php" class="flex-c-m trans-04 p-lr-25">
                    Minha conta </a> <?php }else { ?> <a href="sairSessao.php" class="flex-c-m trans-04
p-lr-25">
                        Bem vindo, <?php echo
$_SESSION['nm_usuario'];?>. Clique aqui para Sair.
                    </a>

```

```

        <?php } ?> <a href="#" class="flex-c-m trans-04 p-lr-25">
BRL </a> </div> </div> </div> <div class="wrap-
        menu-desktop how-shadow1">
        <nav class="limiter-menu-desktop container">
            <!-- Logo desktop --> <a
            href="#" class="logo">
                
            </a> <!-- Menu desktop -
            ->
            <div class="menu-desktop">
            <ul class="main-menu">
                <li> <a href="product.php?genero=0">Shop</a> </li> <li
                class="label1" data-label1="hot">

                <?php if (empty($_SESSION)){ ?>

                <a href="indexLogin.php"
>Carrinho</a>

                <?php }else{
                ?>

                <a href="shoping-cart.php"
>Carrinho</a>

                <?php } ?> </li> <li> <a href="about.php">Sobre</a> </li> <li> <a
                href="contact.php">Contato</a> </li> </ul> </div> <!-- Icon header -->
                <div class="wrap-icon-header flex-w flex-r-m">
                <?php if (empty($_SESSION)){ ?> <a href="indexLogin.php" class="icon-header-item cl2 hov-cl1 trans-04
                p-l-22 p-r-11 icon-header-noti js-show-cart2" data-notify="0">
                <i class="zmdi zmdi-favorite-outline"></i> </a>

                <?php }else {
                    require_once "conexaoBD.php";

```

```

        $query = "SELECT * FROM item_desejado i
inner join livro l on i.cd_livro=l.cd_livro where
i.cd_usuario='$_SESSION['cd_usuario']";

        $result = mysqli_query($db,$query); $total = 0; $num_results =
mysqli_num_rows($result); ?> <a href="#" class="icon-header-item cl2 hov-cl1 trans-04 p-l-22 p-
r-11 icon-header-noti js-show-cart2" data-notify="<?php echo $num_results;
?>">

<i class="zmdi zmdi-favorite-outline"></i> </a>

        <?ph
        p } ?>

</div> </nav> </div>
</div> </header>

<!-- wishlist --> <div class="wrap-header-cart js-
panel-cart2">

    <div class="s-full js-hide-cart2"></div>
    <div class="header-cart flex-col-l p-l-65 p-r-25">
        <div class="header-cart-title flex-w flex-sb-m p-b-8">
            <span class="mtext-103 cl2">

                <?php if (empty($_SESSION)){

                    ?>

                Usuário não logado <?php }else { ?>

                    Seu carrinho, <?php echo
$_SESSION['nm_usuario']; ?>

                    <?ph
                    p } ?>

                </span>

            <div class="fs-35 lh-10 cl2 p-lr-5 pointer hov-cl1 trans-04 js-
hide-cart2">

                <i class="zmdi zmdi-close"></i> </div> </div> <div class="header-

```



```

        cart-content flex-w js-pscroll">
            <ul class="header-cart-wrapitem w-full">

                <?php require_once "conexaoBD.php";

                $listaltensDesejosSelecionados = array(); $query = "SELECT * FROM
                item_desejado i inner join livro l on i.cd_livro=l.cd_livro where
                i.cd_usuario=".$_SESSION['cd_usuario'];

                $result = mysqli_query($db,$query); $total = 0;
                $num_results = mysqli_num_rows($result); for
                ($i=0; $i <$num_results; $i++)

                {

                    $row = mysqli_fetch_array($result);
                    array_push($listaltensDesejosSelecionados,
                    $row['cd_livro']);

                ?> <li class="header-cart-item flex-w flex-t m-b-12">

                    <div class="header-cart-item-img">

                    </div> <div class="header-cart-item-txt p-t-8">

                        <a href="#" class="header-cart-item-name m-b-
                        18 hov-cl1 trans-04">

                            <?php echo $row['nm_livro']; ?> </a> <span class="header-cart-item-
                            info"> <?php echo $row['preco_venda'];

                            $total += $row['preco_venda']; ?> </span> </div> </li>

                                <?ph
                                p } ?>

                    </ul> <div class="w-
                    full">

                        <div class="header-cart-total w-full p-tb-40">

                            Total: <?php echo $total; ?> </div> <!--<div class="header-cart-buttons flex-w
                            w-full">

                                <a href="shopping-cart.php" class="flex-c-m stext-
                                101 cl0 size-107 bg3 bor2 hov-btn3 p-lr-15 trans-04 m-r-8 m-b-10">

```

```

101 cl0 size-107 bg3 bor2 hov-btn3 p-lr-15 trans-04 m-b-10">
    Check Out </a> </div>--> </div> </div>
</div> </div>

```

```
$query = "select * from livro where genero like '%Fic%'";
```

```

",
,

        } else if ($genero == 2){

            $query = "select * from livro where genero like

'%Romance%' ";

        } else if ($genero == 3){

            $query = "select * from livro where genero like

'%Docu%' ";

        } else if ($genero == 4){

            $query = "select * from livro where genero like

'%Literatu%' ";

        }

        $result = mysqli_query($db,$query); $num_results =
        mysqli_num_rows($result); for ($i=0; $i
        <$num_results; $i++) {

            $row = mysqli_fetch_array($result); ?>

            <div class="col-sm-6 col-md-4 col-lg-3 p-b-35
isotope-item women"> <!-- Homepage -->

                <div class="block2">

                    <div class="block2-pic hov-img0">
                        

                        <?php if (empty($_SESSION)){ ?>

                            <a href="indexLogin.php" class="block2-
btn flex-c-m stext-103 cl2 size-102 bg0 bor2 hov-btn1 p-lr-15 trans-04 js-
addcart-detail">Carrinho</a>

                            <?php }else{ ?> <a href="montarCarrinho.php?livro=<?php echo $row['cd_livro'];?>"
class="block2-btn flex-c-m stext-103 cl2 size-102 bg0 bor2 hov-btn1 p-lr-15
trans-04 js-addcart-detail">

                                Carrinho </a> <?php } ?> </div> <div class="block2-txt flex-w flex-t p-t-14">

                                    <div class="block2-txt-child1 flex-col-l ">

                                        <a class="stext-104 cl4 hov-cl1
trans-04 js-name-b2 p-b-6">

                                            <?php echo $row['nm_livro']; ?> </a> <span class="stext-105 cl3">

                                                <?php echo $row['preco_venda']; ?> </span> </div> <div class="block2-txt-

```

```

child2 flex-r p-t-3">

    <form method="post" action="cadastroDesejo.php" name="frmListaDesejo"
        id="frmListaDesejo">

        <input type="hidden" name="cdLivro"
id="cdLivro" value="<?php echo $row['cd_livro'];?>">

        <input type="hidden"
name="cd_usuario" id="cd_usuario" value="<?php echo
$_SESSION['cd_usuario'];?>">

        <?ph
        p

        if(in_array($row['cd_livro'],$listaltensDesejosSelecionados)){

            ?>

            <input type="hidden"
name="acao" id="acao" value="deletar">

            <button type="submit"
name="enviar" > </button>

            <?php }else{
            ?>

            <input type="hidden"
name="acao" id="acao" value="adicionar">

            <button type="submit"
name="enviar" > </button>

            <?php } ?> </form> </div>
            </div> </div> </div>

            <?php } mysqli_close($db); ?>

            </div> </div> </div> <!-- Footer -->
            <?php require_once "i_footer.php"; ?>

<!--
=====
=== ----->

    <script src="vendor/jquery/jquery-3.2.1.min.js"></script>
<!--
=====

```

```

=====-->

<script src="vendor/animations/js/animations.min.js"></script> <!--
=====
=====-->

<script src="vendor/bootstrap/js/popper.js"></script> <script
src="vendor/bootstrap/js/bootstrap.min.js"></script> <!--
=====
=====-->

<script src="vendor/select2/select2.min.js"></script> <script>

    $(".js-select2").each(function(){

        $(this).select2({

minimumResultsForSearch: 20, dropdownParent: $(this).next('.dropDownSelect2')
}); }) </script> <!--
=====
=====-->

<script src="vendor/daterangepicker/moment.min.js"></script> <script
src="vendor/daterangepicker/daterangepicker.js"></script> <!--
=====
=====-->

<script src="vendor/slick/slick.min.js"></script> <script src="js/slick-
custom.js"></script> <!--
=====
=====-->

<script src="vendor/parallax100/parallax100.js"></script> <script>

    $('.parallax100').parallax100(); </script> <!--
=====
=====-->

<script src="vendor/MagnificPopup/jquery.magnific-
popup.min.js"></script>

<script>

    $('.gallery-lb').each(function() { // the containers for all your galleries
$(this).magnificPopup({
        delegate: 'a', // the selector for gallery item type: 'image',
        gallery: {

enabled:true }, mainClass: 'mfp-fade' }); }); </script> <!--
=====
=====-->

```

```

<script src="vendor/isotope/isotope.pkgd.min.js"></script> <!--
=====
=== ----->

<script src="vendor/sweetalert/sweetalert.min.js"></script> <script>

    $(' .js-addwish-b2, .js-addwish-detail').on('click', function(e){
e.preventDefault();

    });

    $(' .js-addwish-b2').each(function(){

var nameProduct = $(this).parent().parent().find(' .js-name-
b2').html();

    $(this).on('click', function(){

    swal(nameProduct, "is added to wishlist !", "success"); $(this).addClass('js-
addedwish-b2'); $(this).off('click'); }); }); $(' .js-addwish-
detail').each(function(){

    var nameProduct = $(this).parent().parent().parent().find(' .js-
name-detail').html();

    $(this).on('click', function(){

    swal(nameProduct, "is added to wishlist !", "success"); $(this).addClass('js-
addedwish-detail'); $(this).off('click'); }); }); /*-----
----*/
    $(' .js-addcart-detail').each(function(){

    var nameProduct = $(this).parent().parent().parent().parent().find(' .js-
name-detail').html();

    $(this).on('click', function(){

    swal(nameProduct, "is added to cart !", "success"); }); }); </script> <!--
=====
=== ----->

<script src="vendor/perfect-scrollbar/perfect-scrollbar.min.js"></script>

<script>

    $(' .js-pscroll').each(function(){

    $(this).css('position','relative');
    $(this).css('overflow','hidden'); var ps = new
PerfectScrollbar(this, {

    wheelSpeed: 1, scrollingThreshold:
1000, wheelPropagation: false, });


```

```
$(window).on('resize', function(){
    ps.update(); }) }); </script> <!--
=====
== <!-->

<script src="js/main.js"></script>
</body> </html>
```

Arquivo
product.php[illegible]

```

    } else if ($genero == 2){
        $query = "select * from livro where genero like
'%Romance%' ";
    } else if ($genero == 3){
        $query = "select * from livro where genero like
'%Docu%' ";
    } else if ($genero == 4){
        $query = "select * from livro where genero like
'%Literatu%' ";
    }
    $result = mysqli_query($db,$query); $num_results =
    mysqli_num_rows($result); for ($i=0; $i
    <$num_results; $i++) {
        $row = mysqli_fetch_array($result); ?>

        <div class="col-sm-6 col-md-4 col-lg-3 p-b-35
isotope-item women"> <!-- Homepage -->
        <div class="block2">
            <div class="block2-pic hov-img0">
                
                <?php if (empty($_SESSION)){ ?>
                    <a href="indexLogin.php" class="block2-
btn flex-c-m stext-103 cl2 size-102 bg0 bor2 hov-btn1 p-lr-15 trans-04 js-
addcart-detail">Carrinho</a>
                    <?php }else{ ?> <a href="montarCarrinho.php?livro=<?php echo $row['cd_livro']; ?>"
class="block2-btn flex-c-m stext-103 cl2 size-102 bg0 bor2 hov-btn1 p-lr-15
trans-04 js-addcart-detail">
                        Carrinho
                    </a> <?php } ?> </div> <div class="block2-txt flex-w
flex-t p-t-14">
                        <div class="block2-txt-child1 flex-col-l ">
                            <a class="stext-104 cl4 hov-cl1
trans-04 js-name-b2 p-b-6">
                                <?php echo $row['nm_livro']; ?> </a> <span class="stext-105 cl3">

```



```

        <?php echo $row['preco_venda']; ?> </span> </div> <div class="block2-txt-
child2 flex-r p-t-3">

```

```

        <form method="post" action="cadastroDesejo.php" name="frmListaDesejo"
            id="frmListaDesejo">

```

```

                <input type="hidden" name="cdLivro"
id="cdLivro" value="<?php echo $row['cd_livro']; ?>">

```

```

                <input type="hidden"
name="cd_usuario" id="cd_usuario" value="<?php echo
$_SESSION['cd_usuario']; ?>">

```

```

                <?ph
p

```

```

if(in_array($row['cd_livro'],$listaltensDesejosSelecionados)){
    ?>

```

```

                <input type="hidden"
name="acao" id="acao" value="deletar">

```

```

                <button type="submit"
name="enviar" > </button>

```

```

                <?php }else{
                ?>

```

```

                <input type="hidden"
name="acao" id="acao" value="adicionar">

```

```

                <button type="submit"
name="enviar" > </button>

```

```

                <?ph
p
            } ?> </form> </div> </div>
        </div> </div>

```

```

        <?php } mysqli_close($db); ?>

```

```

    </div> </div> </div> <!-- Footer -->
    <?php require_once "i_footer.php"; ?>

```

```

<!--
=====
=====-->

```

```

<script src="vendor/jquery/jquery-3.2.1.min.js"></script> <!--
=====
=== ----->

<script src="vendor/animation/js/animation.min.js"></script> <!--
=====
=== ----->

<script src="vendor/bootstrap/js/popper.js"></script> <script
src="vendor/bootstrap/js/bootstrap.min.js"></script> <!--
=====
=== ----->

<script src="vendor/select2/select2.min.js"></script> <script>
    $(".js-select2").each(function(){
        $(this).select2({
            minimumResultsForSearch: 20, dropdownParent:
            $(this).next('.dropDownSelect2') });
    }) </script> <!--
=====
=== ----->

<script src="vendor/daterangepicker/moment.min.js"></script> <script
src="vendor/daterangepicker/daterangepicker.js"></script> <!--
=====
=== ----->

<script src="vendor/slick/slick.min.js"></script> <script src="js/slick-
custom.js"></script> <!--
=====
=== ----->

<script src="vendor/parallax100/parallax100.js"></script> <script>
    $('.parallax100').parallax100(); </script> <!--
=====
=== ----->

<script src="vendor/MagnificPopup/jquery.magnific-
popup.min.js"></script>

<script>
    $('.gallery-lb').each(function() { // the containers for all your galleries
        $(this).magnificPopup({

            delegate: 'a', // the selector for gallery item type: 'image',
            gallery: {

```

```

        enabled:true }, mainClass: 'mfp-fade' }); }); </script> <!--
=====
=== ----->

<script src="vendor/isotope/isotope.pkgd.min.js"></script> <!--
=====
=== ----->

<script src="vendor/sweetalert/sweetalert.min.js"></script> <script>
    $(' .js-addwish-b2, .js-addwish-detail').on('click', function(e){
e.preventDefault();

    });

    $(' .js-addwish-b2').each(function(){
var nameProduct = $(this).parent().parent().find(' .js-name-
        b2').html();

        $(this).on('click', function(){

            swal(nameProduct, "is added to wishlist !", "success"); $(this).addClass('js-
            addedwish-b2'); $(this).off('click'); }); }); $(' .js-addwish-
            detail').each(function(){

                var nameProduct = $(this).parent().parent().parent().find(' .js-
            name-detail').html();

                $(this).on('click', function(){

                    swal(nameProduct, "is added to wishlist !", "success"); $(this).addClass('js-
            addedwish-detail'); $(this).off('click'); }); }); /*-----
            ----*/ $(' .js-addcart-detail').each(function(){

                var nameProduct = $(this).parent().parent().parent().parent().find(' .js-
            name-detail').html();

                $(this).on('click', function(){

                    swal(nameProduct, "is added to cart !", "success"); }); }); </script> <!--
=====
=== ----->

<script src="vendor/perfect-scrollbar/perfect-scrollbar.min.js"></script>
<script>
    $(' .js-pscroll').each(function(){

        $(this).css('position','relative');
        $(this).css('overflow','hidden'); var ps = new
        PerfectScrollbar(this, {

```

```

wheelSpeed: 1, scrollingThreshold:
1000, wheelPropagation: false, });
$(window).on('resize', function(){
ps.update(); }) }); </script> <!--
=====
=====-->

<script src="js/main.js"></script>
</body> </html>

```

```
<?php $busca = $_GET['search'];
```

```

        if($busca == null){

$query = "select * from livro "; } else{ $query = "select * from livro where nm_livro
like '%" . $busca . "%' ";

} $result = mysqli_query($db,$query); $num_results =
mysqli_num_rows($result);

if($num_results == 0){

$query = "select * from livro ";

$result = mysqli_query($db,$query); $num_results =
mysqli_num_rows($result); } for ($i=0; $i <$num_results; $i++) {
$row = mysqli_fetch_array($result);

?>

```

```

<div class="col-sm-6 col-md-4 col-lg-3 p-b-35
isotope-item women">
    <!-- Block2 --> <div
    class="block2">

        <div class="block2-pic hov-img0">

            <?php if (empty($_SESSION)){ ?>

                <a href="indexLogin.php" class="block2-
btn flex-c-m stext-103 cl2 size-102 bg0 bor2 hov-btn1 p-lr-15 trans-04 js-
addcart-detail">Carrinho</a>

                <?php }else{ ?> <a href="montarCarrinho.php?livro=<?php echo $row['cd_livro']; ?>"
class="block2-btn flex-c-m stext-103 cl2 size-102 bg0 bor2 hov-btn1 p-lr-15
trans-04 js-addcart-detail">

                    Carrinho </a> <?php } ?> </div> <div class="block2-txt flex-w flex-t p-t-14">

                        <div class="block2-txt-child1 flex-col-l ">

                            <a class="stext-104 cl4 hov-cl1 trans-04 js-
name-b2 p-b-6">

                                <?php echo $row['nm_livro']; ?> </a> <span class="stext-105 cl3">

                                    <?php echo $row['preco_venda']; ?> </span> </div> <div class="block2-txt-
child2 flex-r p-t-3">

                                <form method="post" action="cadastroDesejo.php" name="frmListaDesejo"

```

```

        id="frmListaDesejo"> <input type="hidden" name="cdLivro" id="cdLivro"
                                value="<?php echo $row['cd_livro'];?>">
                                <input type="hidden"
name="cd_usuario" id="cd_usuario" value="<?php echo
$_SESSION['cd_usuario'];?>">

                                <?ph
                                p

if(in_array($row['cd_livro'],$listaltensDesejosSelecionados)){

                                ?> <input type="hidden"
name="acao" id="acao" value="deletar">

                                <button type="submit" name="enviar" > </button> <!--<button type="submit" name="enviar" > </button>-->

                                <?php }else{ ?> <input type="hidden" name="acao" id="acao" value="adicionar">

                                <button type="submit"
name="enviar" > </button>

                                <?php } ?> </form> </div> </div>
                                </div> </div>

<?ph
p }

                                mysqli_close($db);

?>

</div> </div> </div> <?php require_once
                                "i_footer.php"; ?>

<!--
=====
=== <!-->

                                <script src="vendor/jquery/jquery-3.2.1.min.js"></script> <!--
=====
=== <!-->

                                <script src="vendor/animation/js/animation.min.js"></script> <!--
=====
=== <!-->

                                <script src="vendor/bootstrap/js/popper.js"></script> <script

```

```

src="vendor/bootstrap/js/bootstrap.min.js"></script> <!--
=====
=== ----->

<script src="vendor/select2/select2.min.js"></script> <script>
    $(".js-select2").each(function(){
        $(this).select2({
minimumResultsForSearch: 20, dropdownParent: $(this).next('.dropDownSelect2')
}); }) </script> <!--
=====
=== ----->

<script src="vendor/daterangepicker/moment.min.js"></script> <script
src="vendor/daterangepicker/daterangepicker.js"></script> <!--
=====
=== ----->

<script src="vendor/slick/slick.min.js"></script> <script src="js/slick-
custom.js"></script> <!--
=====
=== ----->

<script src="vendor/parallax100/parallax100.js"></script> <script>
    $('parallax100').parallax100(); </script> <!--
=====
=== ----->

<script src="vendor/MagnificPopup/jquery.magnific-
popup.min.js"></script>
<script>
    $('gallery-lb').each(function() { // the containers for all your galleries
$(this).magnificPopup({
        delegate: 'a', // the selector for gallery item type: 'image',
        gallery: {
enabled:true }, mainClass: 'mfp-fade' }); }); </script> <!--
=====
=== ----->

<script src="vendor/isotope/isotope.pkgd.min.js"></script> <!--
=====
=== ----->

<script src="vendor/sweetalert/sweetalert.min.js"></script> <script>
    $('js-addwish-b2, .js-addwish-detail').on('click', function(e){

```

```

e.preventDefault();

});

$('.js-addwish-b2').each(function(){
var nameProduct = $(this).parent().parent().find('.js-name-
b2').html();

$(this).on('click', function(){

swal(nameProduct, "is added to wishlist !", "success"); $(this).addClass('js-
addedwish-b2'); $(this).off('click'); }); });

$('.js-addwish-detail').each(function(){

var nameProduct = $(this).parent().parent().parent().find('.js-
name-detail').html();

$(this).on('click', function(){
swal(nameProduct, "is added to wishlist !", "success");

$(this).addClass('js-addedwish-detail'); $(this).off('click');
}); });

/*-----*/

$('.js-addcart-detail').each(function(){

var nameProduct = $(this).parent().parent().parent().parent().find('.js-
name-detail').html();

$(this).on('click', function(){

swal(nameProduct, "is added to cart !", "success"); }); });

</script> <!--
=====
== <----->

<script src="vendor/perfect-scrollbar/perfect-scrollbar.min.js"></script>
<script>

$('.js-pscroll').each(function(){

$(this).css('position','relative');

```



```

        $(this).css('overflow','hidden'); var ps = new
        PerfectScrollbar(this, {
            wheelSpeed: 1,
            scrollingThreshold: 1000,
            wheelPropagation: false, });

        $(window).on('resize', function(){
            ps.update(); }) }); </script>

<!--
=====
=====-->

<script src="js/main.js"></script>

</body>
</html>

```

Arquivo
sairSessao.php

```

<?php session_start();

    session_start(); session_destroy();
    header("Location: product.php");

?> Arquivo shopping-cart.php

<?php require_once "i_topo.php";

$_SESSION['sessaoTotal'] = 0; $lista = array(); if
(empty($_SESSION['listaltensCarrinhoSelecionados'])) { $lista =
$_SESSION['listaltensCarrinhoSelecionados']; } ?>

<!-- breadcrumb --> <div
class="container">

    <div class="bread-crumbs flex-w p-l-25 p-r-15 p-t-30 p-lr-0-lg"> <a
href="index.php" class="stext-109 cl8 hov-cl1 trans-04">

Home <i class="fa fa-angle-right m-l-9 m-r-10" aria-
hidden="true"></i>

    </a> <span class="stext-109
cl4">

```

```

Carrinho de Compra </span>
</div> </div>
<!-- Shopping Cart --> <form
class="bg0 p-t-75 p-b-85">

    <div class="container">

        <div class="row">

            <div class="col-lg-10 col-xl-7 m-lr-auto m-b-50">

                <div class="m-l-25 m-r--38 m-lr-0-xl">

                    <div class="wrap-table-shopping-cart">

                        <table class="table-shopping-cart">

                            <tr class="table_head">

                                <th class="column-1">Produtos</th> <th class="column-2"></th> <!--<th
                                    class="column-3">Price</th>-->

                                <th
class="column-4">Quantidade</th>

                                <th class="column-5">Total</th> </tr> <?php

$codigos = ""; foreach ($lista as
    $item) {

$codigos = $codigos . $item. ", "; }

$codigos = substr($codigos,0, -2); if
(empty($codigos) ){

$codigos=0; }

$query = "select * from livro where cd_livro in
(".$codigos.)";

$result = mysqli_query($db,$query); $num_results =
mysqli_num_rows($result); for ($i=0; $i
<$num_results; $i++)

    { $row = mysqli_fetch_array($result); ?>

        <tr class="table_row">

            <td class="column-1">

                <div class="how-itemcart1">
                    

```

```

</div> </td> <td class="column-2"><?php echo $row['nm_livro']; ?></td>
<!--<td class="column-3">?php echo
$row['preco_venda']; ?></td>-->

<td class="column-4">
<div class="wrap-num-product
flex-w m-l-auto m-r-0">

<div class="btn-num-product-
down cl8 hov-btn3 trans-04 flex-c-m" onclick="carrinhoSubValor(<?php echo
$row['cd_livro']; ?>,<?php echo $row['preco_venda']; ?>);">

<i class="fs-16 zmdi
zmdi-minus"></i>

</div> <input class="mtext-104 cl3 txt-center num-product" type="number"
name="num-product<?php echo $row['cd_livro']; ?>" id="num-product<?php echo
$row['cd_livro']; ?>" value="0">

<?ph
p

$codigoFinal="valor".$row['cd_livro'];

?> <div class="btn-num-product-up cl8 hov-btn3 trans-04 flex-c-m"
onclick="carrinhoAdValor(<?php echo $row['cd_livro']; ?>,<?php echo
$row['preco_venda']; ?>);">

<i class="fs-16 zmdi
zmdi-plus"></i>

</div> </div> </td> <td class="column-5" id="valor<?php echo $row['cd_livro'];
?>"></td> </tr>

<?ph
p
} ?>

</table> </div> <div class="flex-w flex-sb-m bor15 p-t-18 p-b-15 p-lr-40 p-lr-
15-sm">

<div class="flex-w flex-m m-r-20 m-tb-5">

<input class="stext-104 cl2 plh4 size-
117 bor13 p-lr-20 m-r-10 m-tb-5" type="text" name="coupon"
placeholder="Código do Cupom">

<div class="flex-c-m stext-101 cl2 size-

```

```

118 bg8 bor13 hov-btn3 p-lr-15 trans-04 pointer m-tb-5">
    Aplicar </div> </div> </div> </div> </div> <div class="col-sm-10 col-lg-7
col-xl-5 m-lr-auto m-b-50">
        <div class="bor10 p-lr-40 p-t-30 p-b-40 m-l-63 m-r-40
m-lr-0-xl p-lr-15-sm">
            <h4 class="mtext-109 cl2 p-b-30">
Total </h4> <div class="flex-w flex-t bor12 p-b-13">
            <div class="size-208">
                <span class="stext-110 cl2">
Subtotal: </span> </div> <div class="size-209">
                <span class="mtext-110 cl2"> <input type="text" readonly="readonly"
id="subTotal" value="0">
            </span> </div> </div> <div class="p-t-15">
                <div class="bor8 bg0 m-b-22">
                    <input class="stext-111 cl8 plh3 size-111
p-lr-15" type="text" id="cep" name="cep" placeholder="CEP">
                </div> <div class="size-
208">
                    <span class="stext-110 cl2">
                        Frete: <input type="text"
readonly="readonly" id="frete" name="frete" value="0" >
                    </span> </div> <div class="size-208">
                        <span class="stext-110 cl2"> </span>
                    </div> <div class="size-209">
                        <span class="mtext-110 cl2">
                            &nbsp; </span> </div> </div> <div class="flex-
w">
                            <div class="flex-c-m stext-101 cl2 size-115 bg8
bor13 hov-btn3 p-lr-15 trans-04 pointer">
                                <a href="#" onClick="calcularFrete();">
Calcular Frete</a>
                            </div> </div> <div class="size-209">
                                <span class="mtext-110 cl2">
                                    &nbsp; </span> </div> <div class="panel panel-info"> <h4

```

```

        class="mtext-109 cl2 p-b-30">
Pagamento </h4> <div class="panel-body">
    <div class="form-group">
<div class="col-md-12"><strong>Cartão:</strong></div>
    <div class="col-md-12">
        <select id="CreditCardType"
name="CreditCardType" class="form-control">
            <option
value="5">Crédito</option>
            <option
value="6">Débito</option>
        </select> </div> </div> <div class="form-group">
        <div class="col-md-12"><strong>Número do
Cartão de Crédito/Débito:</strong></div>
        <div class="col-md-12"><input type="tel"
class="form-control" id="cc" name="cc" maxlength="19" /></div>
        </div> <div class="form-
group">
        <div class="col-md-12"><strong>Código de
Segurança:</strong></div>
        <div class="col-md-12"><input type="text" class="form-control" name="car_code"
maxlength="3" /></div>
        </div> <div class="form-
group">
        <div class="col-md-12">
            <strong>Data de Expiração</strong> </div> <div class="col-lg-6 col-md-6
col-sm-6 col-xs-12">
                <select class="form-control"
name="">
                    <option value="">Mês</option>
                    <option value="01">01</option>
                    <option value="02">02</option>
                    <option value="03">03</option>
                    <option value="04">04</option>
                    <option value="05">05</option>
                    <option value="06">06</option>

```

```

                                <option value="07">07</option>
                                <option value="08">08</option>
                                <option value="09">09</option>
<option value="10">10</option> <option value="11">11</option> <option
value="12">12</option> </select> </div> <div class="col-lg-6 col-md-6 col-sm-6
col-xs-12">

                                <select class="form-control"
name="">

                                <option value="">Ano</option> <option value="2019">2019</option>

                                <option
value="2020">2020</option>

                                <option
value="2021">2021</option>

                                <option
value="2022">2022</option>

                                <option
value="2023">2023</option>

                                <option
value="2024">2024</option>

                                <option
value="2025">2025</option>

                                </select> </div> </div> <div class="form-group">

                                <div class="col-md-12">

                                <span>Pague seguro usando seu
cartão.</span>

                                </div> <div class="col-md-
12">

                                <ul class="cards">

                                <li > <a href="#"

class="m-all-1">

                                </a>

                                Visa </li> <li class="mastercard hand">

                                <a href="#"

class="m-all-1">

```

```


    </a> MasterCard </li> <li class="amex hand">
<a href="#"
class="m-all-1">

    </a> PayPal </li> </ul> <div class="clearfix"></div> </div> </div> </div>
    </div> <div class="flex-w flex-t p-t-27 p-b-33">
        <div class="size-208">
            <span class="mtext-101 cl2">
Total: </span> </div> <div class="size-209 p-t-1">
            <input type="text" readonly="readonly"
id="somaTotal" value="0">
</div> </div> <button class="flex-c-m stext-101 cl0 size-116 bg3 bor14 hov-btn3 p-
lr-15 trans-04 pointer">
    Finalizar Compra </button> </div>
</div> </div> </div> </form> </div> <?php
    require_once "i_footer.php";
?>

<!--
=====
=== ----->
    <script src="vendor/jquery/jquery-3.2.1.min.js"></script> <!--
=====
=== ----->
    <script src="vendor/animation/js/animation.min.js"></script> <!--
=====
=== ----->
<script src="vendor/bootstrap/js/popper.js"></script> <script
src="vendor/bootstrap/js/bootstrap.min.js"></script> <!--
=====
=== ----->
    <script src="vendor/select2/select2.min.js"></script> <script>
        $(".js-select2").each(function(){

```

```

        $(this).select2({
minimumResultsForSearch: 20, dropdownParent: $(this).next('.dropDownSelect2')
}); }) </script> <!--
=====
=== ----->

<script src="vendor/MagnificPopup/jquery.magnific-popup.min.js"></script> <!--
=====
=== ----->

<script src="vendor/perfect-scrollbar/perfect-scrollbar.min.js"></script>
<script>
    $(''.js-pscroll').each(function(){
        $(this).css('position','relative');
        $(this).css('overflow','hidden'); var ps = new
        PerfectScrollbar(this, {
            wheelSpeed: 1,
            scrollingThreshold: 1000,
            wheelPropagation: false, });

        $(window).on('resize', function(){
            ps.update(); }) }); </script> <!--
=====
=== ----->

<script src="js/main.js"></script>

<script>

    function calculaFrete(){

        cep=document.getElementById("cep").value;

        valor = 1; if(cep ==
        "13142134"){
            valor = "10.99"; }else if(cep ==
            "13068211") {
                valor = "15.99"; }else{ valor = "20.00"; } document.getElementById("frete").value=valor;
        calculaValorTotal(valor); } function calculaValorTotal(valor){

```



```

subTotal=document.getElementById("subTotal").value;
total  =  parseFloat(subTotal)  +  parseFloat(valor);
document.getElementById("somaTotal").value=total; }

function carrinhoAdValor(codigo,valor){

    input= "num-product"+codigo;
    qtd=document.getElementById(input).value;

    qtd++;

    soma= valor*qtd; codigoSoma= "valor"+codigo;
    document.getElementById(codigoSoma).innerHTML=soma;
    calculaAdTotal(valor,qtd, soma); } function carrinhoSubValor(codigo,valor){

    input= "num-product"+codigo;
    qtd=document.getElementById(input).value;

    if (qtd > 0) { qtd--
    ; }

    soma= valor*qtd; codigoSoma= "valor"+codigo;
    document.getElementById(codigoSoma).innerHTML=soma;
    calculaSubTotal(valor,qtd, soma); } function calculaAdTotal(valor,qtd, soma){

    temp = document.getElementById("subTotal").value;

    if (qtd > 0){
    qtdLivroAnterior = (qtd - 1) * valor; }else{ qtdLivroAnterior = 0; }
    total = (parseFloat(soma) - qtdLivroAnterior) +
parseFloat(temp) ;

    document.getElementById("subTotal").value=total; } function
calculaSubTotal(valor,qtd, soma){

    temp = document.getElementById("subTotal").value;

    if (qtd > 0){
    qtdLivroAnterior = (qtd - 1) * valor; }else{ qtdLivroAnterior = 0; }

```

```

        total = parseFloat(temp) - (parseFloat(soma) -
qtdLivroAnterior) ; document.getElementById("subTotal").value=total;

    } function mascara(o,f){

v_obj=o v_fun=f setTimeout("execmascara()",1) }
function execmascara(){

v_obj.value=v_fun(v_obj.value) } function mcc(v){

v = v.replace(/D/g, ""); // Permite apenas dígitos v = v.replace(/(\d{4})/g, "$1."); //
Coloca um ponto a cada 4 caracteres v = v.replace(/\.$/, ""); // Remove o ponto se estiver
sobrando

v = v.substring(0, 19)// Limita o tamanho return
v;
} function id( el ){

return document.getElementById( el ); } window.onload =
function(){

id('cc').onkeypress = function(){ mascara( this,
mcc ); } } </script>

</body>
</html>

```

Arquivo
validarLogin.php

```

<?ph
p

```

```

    if(!isset($_SESSION)){
        session_start(); } $email =
$_POST['username']; $senha =
    $_POST['pass'];

require_once "conexaoBD.php";

$query = "select * from usuario where login = '". $email. "'"; $result =

```

```
mysql_query($db,$query); $num_results = mysqli_num_rows($result);
$row = mysqli_fetch_array($result); //echo stripslashes($row['senha']);
```

```
if (substr(md5($senha),0,15) == $row['senha']){
```

```
    $_SESSION['cd_usuario'] = $row['cd_usuario'];
    $_SESSION['nm_usuario'] = $row['nm_usuario'];
    header("Location: product.php");
```

```
}else{
```

```
    header("Location: indexLogin.php?erro=1");
```

```
} mysqli_close($db); ?> Arquivo
```

indexCadastro.php

```
<!DOCTYPE html> <html lang="pt-br">
```

```
<head> <title>Cadastro GMT Store</title>
```

```
<meta charset="UTF-8"> <meta name="viewport" content="width=device-width,
initial-scale=1"> <!--
```

```
=====
=== ----->
```

```
<link rel="icon" type="image/png" href="images/icons/favicon.ico"/> <!--
```

```
=====
=== ----->
```

```
<link rel="stylesheet" type="text/css"
href="vendor/bootstrap/css/bootstrap.min.css"> <!--
```

```
=====
=== ----->
```

```
<link rel="stylesheet" type="text/css" href="fonts/font-awesome-4.7.0/css/font-
awesome.min.css"> <!--
```

```
=====
=== ----->
```

```
<link rel="stylesheet" type="text/css" href="fonts/Linearicons-Free-v1.0.0/icon-
font.min.css"> <!--
```

```
=====
=== ----->
```

```
<link rel="stylesheet" type="text/css" href="vendor/animate/animate.css"> <!--
```

```
=====
```

```

=====-->
    <link rel="stylesheet" type="text/css" href="vendor/css-
hamburgers/hamburgers.min.css"> <!--
=====
=====-->

    <link rel="stylesheet" type="text/css"
href="vendor/animation/css/animation.min.css"> <!--
=====
=====-->

    <link rel="stylesheet" type="text/css" href="vendor/select2/select2.min.css"> <!--
-
=====
=====-->

    <link rel="stylesheet" type="text/css"
href="vendor/daterangepicker/daterangepicker.css"> <!--
=====
=====-->

<link rel="stylesheet" type="text/css" href="cssLogin/util.css"> <link rel="stylesheet"
type="text/css" href="cssLogin/main.css"> <!--
=====
=====--> </head> <body> <div class="limiter">

    <div class="container-login100">

        <div class="wrap-login100 p-l-85 p-r-85 p-t-55 p-b-55">

            <form class="login100-form validate-form flex-sb flex-w">

                <span class="login100-form-title p-b-32">

Novo Cadastro </span> <span class="txt1 p-b-
11">

Nome Completo </span> <div class="wrap-input100 validate-input m-b-36"
data-validate = "Nome
is required">

                <input class="input100" type="text" name="nome" >

                <span class="focus-input100"></span>

            </div> <span class="txt1 p-b-
11">

CPF </span> <div class="wrap-input100 validate-input m-b-36" data-
validate = "Nome
is required">

```

```

        <input class="input100" type="text" name="nome" >
    <span class="focus-input100"></span> </div> <span class="txt1
        p-b-11">
    Genero </span> <div class="wrap-input100 validate-input m-b-36" style=
        "border:0px;" data-validate = "Nome is required">
        <input class="input50" type="radio"
name="genero" id="masculino" >
        <span class="txt1 p-b-11"> Masculino </span> <br> <input
        class="input50" type="radio" name="genero" id="feminino" >
    <span class="txt1 p-b-11"> Feminino </span> <span class="focus-
    input100"></span> </div> <span class="txt1 p-b-11"> Data de Nascimento
    </span> <div class="wrap-input100 validate-input m-b-36" data-validate = "Nome
    is required">
        <input class="input100" type="date" name="nome" >
    <span class="focus-input100"></span> </div> <span class="txt1
        p-b-11">
        E-mail
    </span> <div class="wrap-input100 validate-input m-b-36" data-validate = "Nome
    is required">
        <input class="input100" type="text" name="nome" >
    <span class="focus-input100"></span> </div> <span class="txt1
        p-b-11">
    Celular </span> <div class="wrap-input100 validate-input m-b-36" data-
    validate = "Nome is required">
        <input class="input100" type="text" name="nome" >
    <span class="focus-input100"></span> </div> <span class="txt1
        p-b-11">
    Login </span> <div class="wrap-input100 validate-input m-b-36" data-
    validate = "Nome is required">
        <input class="input100" type="text" name="nome" >
    <span class="focus-input100"></span> </div> <span class="txt1
        p-b-11">
    Senha </span> <div class="wrap-input100 validate-input m-b-12" data-validate =
        "Password is required">
        <span class="btn-show-pass">
    <i class="fa fa-eye"></i> </span> <input class="input100"

```

```

type="password" name="pass" >
    <span class="focus-input100"></span> </div> <div class="flex-sb-
        m w-full p-b-48">
            <div class="contact100-form-checkbox">
                <input class="input-checkbox100" id="ckb1"
type="checkbox" name="remember-me">
            </div> </div> <div class="container-login100-form-btn">
                <button class="login100-form-btn">
                    Cadastrar </button> </div> </form>
        </div> </div> </div> <div
        id="dropDownSelect1"></div>

<!--
=====
=== ----->

    <script src="vendor/jquery/jquery-3.2.1.min.js"></script> <!--
=====
=== ----->

    <script src="vendor/animstion/js/animstion.min.js"></script> <!--
=====
=== ----->

<script src="vendor/bootstrap/js/popper.js"></script> <script
src="vendor/bootstrap/js/bootstrap.min.js"></script> <!--
=====
=== ----->

    <script src="vendor/select2/select2.min.js"></script> <!--
=====
=== ----->

<script src="vendor/daterangepicker/moment.min.js"></script> <script
src="vendor/daterangepicker/daterangepicker.js"></script> <!--
=====
=== ----->

    <script src="vendor/countdowntime/countdowntime.js"></script> <!--
=====
=== ----->

    <script src="js/main.js"></script>

</body>

```

</html>

Arquivo
indexLogin.php

```
<!DOCTYPE html> <html lang="pt-br">
<head> <title>Login GMT Store</title>

<meta charset="UTF-8"> <meta name="viewport" content="width=device-width,
initial-scale=1"> <!--
=====
=== ----->

    <link rel="icon" type="image/png" href="images/icons/favicon.ico"/> <!--
=====
=== ----->

    <link rel="stylesheet" type="text/css"
href="vendor/bootstrap/css/bootstrap.min.css"> <!--
=====
=== ----->

    <link rel="stylesheet" type="text/css" href="fonts/font-awesome-4.7.0/css/font-
awesome.min.css"> <!--
=====
=== ----->

    <link rel="stylesheet" type="text/css" href="fonts/Linearicons-Free-v1.0.0/icon-
font.min.css"> <!--
=====
=== ----->

    <link rel="stylesheet" type="text/css" href="vendor/animate/animate.css"> <!--
=====
=== ----->

    <link rel="stylesheet" type="text/css" href="vendor/css-
hamburgers/hamburgers.min.css"> <!--
=====
=== ----->

    <link rel="stylesheet" type="text/css"
href="vendor/animation/css/animation.min.css">
<!--
=====
=== ----->
```

```

<link rel="stylesheet" type="text/css" href="vendor/select2/select2.min.css"> <!--
-
=====
=== ----->

<link rel="stylesheet" type="text/css"
href="vendor/daterangepicker/daterangepicker.css"> <!--
=====
=== ----->

<link rel="stylesheet" type="text/css" href="cssLogin/util.css"> <link rel="stylesheet"
type="text/css" href="cssLogin/main.css"> <!--
=====
=== -----> </head> <body> <div class="limiter">

    <div class="container-login100">

        <div class="wrap-login100 p-l-85 p-r-85 p-t-55 p-b-55">

            <form action="validarLogin.php" class="login100-form
validate-form flex-sb flex-w" method="post">

                <span class="login100-form-title p-b-32">

                    Login </span> <span class="txt1 p-b-11">

                        email </span> <div class="wrap-input100 validate-input m-b-36" data-validate =
                            "Username is required">

                                <input class="input100" type="text"
name="username" >

                                    <span class="focus-input100"></span> </div> <span class="txt1
p-b-11">

                                        Senha </span> <div class="wrap-input100 validate-input m-b-12" data-validate =
                                            "Password is required">

                                                <span class="btn-show-pass">

                                                    <i class="fa fa-eye"></i>

                                                        </span> <input class="input100" type="password" name="pass" >

                                                            <span class="focus-input100"></span> </div> <div class="flex-sb-
m w-full p-b-48">

                                                                <div class="contact100-form-checkbox">

                                                                    <input class="input-checkbox100" id="ckb1"
type="checkbox" name="remember-me">

                                                                        <label class="label-checkbox100"
for="ckb1">

```



```

lembra-me </label> </div> <div> <a href="#" class="txt3">
Esqueceu a Senha? </a> </div> </div> <div class="container-
login100-form-btn">
<button class="login100-form-btn">
Login </button>
<br>
<div class="form-group">
<input type="text" class="input100" value="Nome de usuário">
<input type="password" class="input100" value="Senha">
<button type="submit"
class="login100-form-btn">
Cadastre-se Aqui! </button> <br> <br> <?php
<div class="form-group">
<div class="text">
if(!empty($_GET['erro'])){
echo "<span style='color:red;'> Usuário ou senha
invalidos. </span>";
}
?>
</div> </form> </div> </div> </div>
<div id="dropDownSelect1"></div>
<!--
=====
== <script src="vendor/jquery/jquery-3.2.1.min.js"></script> <!--
=====
== <script src="vendor/animstion/js/animstion.min.js"></script> <!--
=====
== <script src="vendor/bootstrap/js/popper.js"></script> <script
src="vendor/bootstrap/js/bootstrap.min.js"></script> <!--
=====
== <script src="vendor/select2/select2.min.js"></script> <!--
=====
== >

```

```

<script src="vendor/daterangepicker/moment.min.js"></script> <script
src="vendor/daterangepicker/daterangepicker.js"></script> <!--
=====
=== ----->

    <script src="vendor/countdowntime/countdowntime.js"></script> <!--
=====
=== ----->

    <script src="js/main.js"></script>

</body>
</html>

```

Arquivo
montarCarrinho.php

```

<?ph
p
    if(!isset($_SESSION)){
        session_start(); }

    $livro = $_GET['livro']; //$listaltensCarrinhoSelecionados = array(); if
    (empty($_SESSION['listaltensCarrinhoSelecionados']) ){

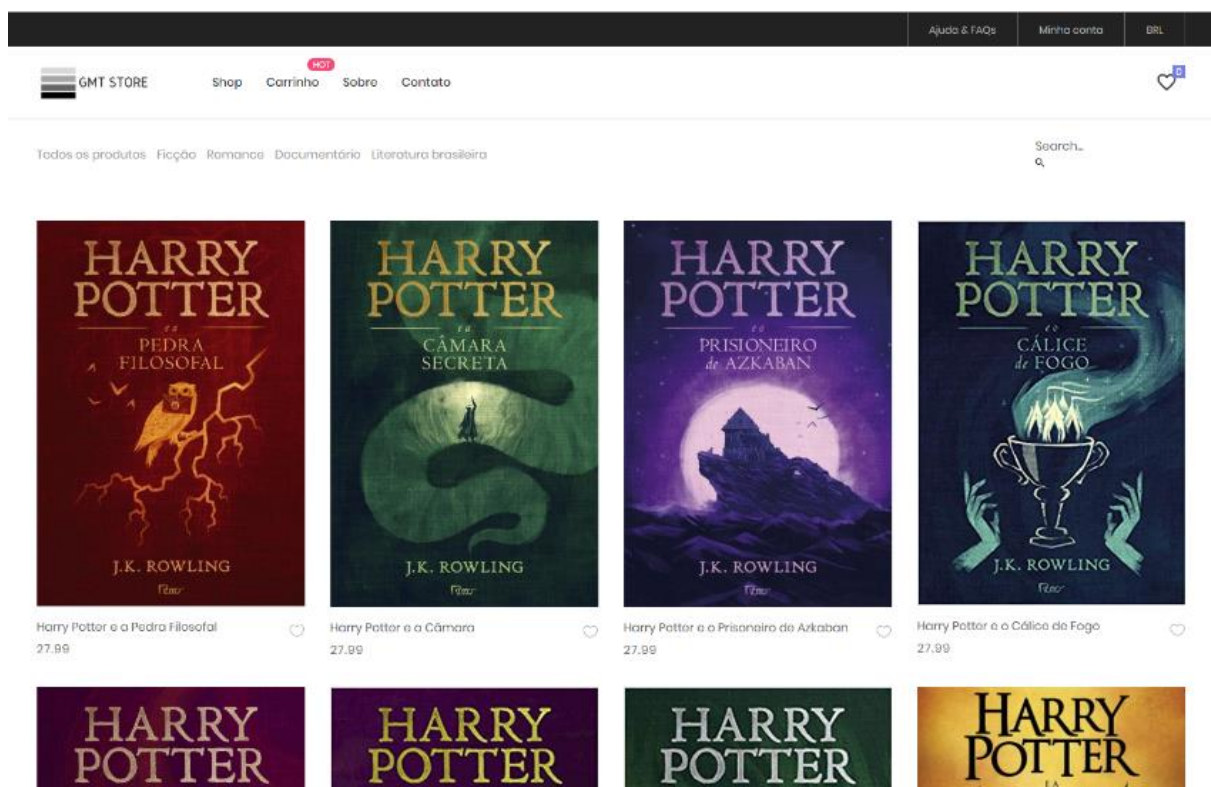
        $_SESSION['listaltensCarrinhoSelecionados'] = array(); } if
    (!in_array($_GET['livro'] , $_SESSION['listaltensCarrinhoSelecionados'])){

        array_push($_SESSION['listaltensCarrinhoSelecionados'] ,
$_GET['livro']);

        } header("Location: shoping-cart.php"); ?>

```

5.2. Apêndice B - Tela do sistema Web em PHP



5.3. Apêndice C - Código do software em Java

```
//Projeto>src>principal: pacote responsável pela inicialização do sistema;
//principal> Principal.java: classe que realiza inicialização do sistema;
package principal;

import javax.swing.JFrame;
import view.CadastroFornecedorView;
import view.CadastroLivroView;
import view.MenuView;

public class Principal extends JFrame{

    public static void main(String[] args) {
        MenuView menu = new MenuView();

        menu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        menu.setSize(800,500);
        menu.setResizable(false);
        menu.setVisible(true);
    }
}

//Projeto>src>bdd: pacote responsável pela conexão com o banco de dados;
//bdd> BancoDeDados.java: classe que realiza a conexão;
package bdd;

import java.sql.DriverManager;
import java.sql.ResultSet;

import javax.swing.JOptionPane;
```

```

import com.mysql.jdbc.Connection;
import com.mysql.jdbc.PreparedStatement;

public class BancoDeDados {
    private static Connection connection = null;

    public static Connection conectar(){
        String servidor =
"jdbc:mysql://35.198.9.166:3306/projeto_livraria";
        String usuario = "root";
        String senha = "@gmtifsp";
        String driver = "com.mysql.jdbc.Driver";
        try {
            Class.forName(driver);
            connection = (Connection)
DriverManager.getConnection(servidor, usuario, senha);
            return connection;
        }catch (Exception e) {
            JOptionPane.showMessageDialog(null, " FALHA AO
CONECTAR O BANCO DE DADOS (ERRO: " + e.getMessage()+")");
        }
        return connection;
    }

    public static boolean estaConectado() {
        if(connection != null) {
            return true;
        }else {
            return false;
        }
    }
}

```

//Projeto>src>controller: pacote responsável pelo recebimento e envio de informações a seus respectivos destinos;

//controller> CompraController.java: classe que recebe as informações relativas a classe Compra;

```

package controller;

```

```

import java.sql.SQLException;
import dao.CompraDAO;
import model.Compra;

```

```

public class CompraController {

    public int cadastro(Compra compra) {
        int idCompra = 0;
        CompraDAO compraDAO = new CompraDAO();
        try {
            idCompra = compraDAO.inserirCompra(compra);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return idCompra;
    }
}

```

//controller>FornecedorController.java: classe que recebe as informações relativas a classe Fornecedor;

```

package controller;

import java.sql.SQLException;
import dao.FornecedorDAO;
import dao.LivroDAO;
import model.Fornecedor;

public class FornecedorController {

    public boolean cadastro(Fornecedor fornecedor) throws
    SQLException {
        FornecedorDAO fornecedorDAO = new
    FornecedorDAO();
        fornecedorDAO.inserir(fornecedor);

        return true;
    }

    public void ListarFornecedor(){
        FornecedorDAO fornecedorDAO = new
    FornecedorDAO();
        fornecedorDAO.Listar();
    }

    public boolean atualizar(Fornecedor fornecedor, int id)
    throws SQLException{
        FornecedorDAO fd = new FornecedorDAO();
        fd.atualizar(fornecedor, id);
        return true;
    }
}

```

```

    }
}
//controller>LivroController.java: classe que recebe as informações relativas a
classe Livro;
package controller;

import java.sql.SQLException;
import dao.LivroDAO;
import model.Livro;

public class LivroController {

    public boolean cadastro(Livro livro) throws SQLException {
        LivroDAO livroDAO = new LivroDAO();
        livroDAO.inserir(livro);
        return true;
    }

    public void ListarLivros(){
        LivroDAO livroDAO = new LivroDAO();
        livroDAO.Listar();
    }

    public boolean atualizar(Livro livro, int id) throws SQLException{
        LivroDAO Id = new LivroDAO();
        Id.atualizar(livro, id);
        return true;
    }
}

```

//Projeto>src>dao: pacote responsável pela manipulação de dados referente a todas as ações que se conectam com o banco de dados;

//dao> CompraDAO.java: classe que manipula todas as informações relativas a classe Compra que se conectam ao banco de dados;

```

package dao;

import java.io.FileWriter;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;

```

```

import java.util.ArrayList;
import java.util.List;
import com.mysql.jdbc.PreparedStatement;
import com.mysql.jdbc.Statement;
import bdd.BancoDeDados;
import model.Compra;
import model.Fornecedor;
import model.Livro;

public class CompraDAO{
    private static PreparedStatement st = null;
    private static ResultSet rs = null;
    List<Compra> compras;

    public int inserirCompra(Compra compra) throws SQLException{
        Connection bdd = BancoDeDados.conectar();
        SimpleDateFormat df = new SimpleDateFormat("dd/MM/yyyy");
        java.util.Date dataUtil = new java.util.Date();
        try {
            dataUtil = df.parse( compra.getDtCompra());
        }catch (ParseException ex) {
        }
        java.sql.Date dataSql = new java.sql.Date(dataUtil.getTime());
        SimpleDateFormat df2 = new SimpleDateFormat("dd/MM/yyyy");
        java.util.Date dataUtil2 = new java.util.Date();
        try {
            dataUtil = df.parse( compra.getDtEntrega());
        } catch (ParseException ex) {
        }
        java.sql.Date dataSql2 = new java.sql.Date(dataUtil.getTime());
        String sql1 = "insert into compra (dt_compra, dt_entrega,
cd_fornecedor) values (?, ?, ?)";
        final PreparedStatement smt = (PreparedStatement)
bdd.prepareStatement(sql1, Statement.RETURN_GENERATED_KEYS);
        smt.setDate(1, dataSql);
        smt.setDate(2, dataSql2);
        smt.setInt(3, compra.getFornecedor().getCodigo());
        smt.executeUpdate();
        final ResultSet rs1 = smt.getGeneratedKeys();
        int lastId = 0;
        if (rs1.next()) {
            lastId = rs1.getInt(1);
        }
    }
}

```



```

        String sql2 = "insert into item_compra (cd_livro, cd_compra,
quantidade, preco_unitario) values (?, ?, ?, ?)";
        PreparedStatement smt2 = (PreparedStatement)
bdd.prepareStatement(sql2);
        smt2.setInt(1, compra.getLivro().getCdLivro());
        smt2.setInt(2, lastId);
        smt2.setInt(3, compra.getQuantidade());
        smt2.setDouble(4, compra.getPreco());
        smt2.executeUpdate();
        return lastId;
    }

    public ArrayList<Compra> Listar(){
        Connection bdd = BancoDeDados.conectar();
        compras = new ArrayList<Compra>();
        int linha = 0;
        String texto = null;
        try {
            String sql = "SELECT * FROM compra as c JOIN
item_compra as ic JOIN livro as l JOIN fornecedor as f ON f.cd_fornecedor =
c.cd_fornecedor and ic.cd_compra = c.cd_compra and l.cd_livro = ic.cd_livro
ORDER BY c.cd_compra;";
            PreparedStatement smt = (PreparedStatement)
bdd.prepareStatement(sql);
            ResultSet rs = smt.executeQuery();
            while(rs.next()) {
                Compra compra = new Compra();
                compra.setCdCompra(rs.getInt("cd_Compra"));
                compra.setDtCompra(rs.getString("dt_Compra"));
                compra.setDtEntrega(rs.getString("dt_Entrega"));

                compra.getLivro().setNomeLivro(rs.getString("nm_Livro"));

                compra.getFornecedor().setNmFornecedor(rs.getString("nm_Forneced
or"));

                compra.setPreco(rs.getDouble("preco_unitario"));
                compra.setQuantidade(rs.getInt("quantidade"));
                compras.add(compra);
            }
        } catch (Exception e) {
            System.out.println("ERRO: " + e.getMessage());
        }
        return (ArrayList<Compra>)compras;
    }
}

```

```

public boolean gerarArq(String caminho, List<Compra> compras) {
    this.compras= compras;
    Date data = new Date(System.currentTimeMillis());
    try {
        FileWriter arq = new FileWriter(caminho);
        PrintWriter gravarArq = new PrintWriter(arq);
        gravarArq.println("# RELATÓRIO DO HISTÓRICO DE
COMPRAS GERADO EM "+data+" #");
        for(int i = 0; i<compras.size(); i++) {
            gravarArq.print("ID: ");
            gravarArq.println(((Compra)
compras.get(i)).getCdCompra());
            gravarArq.print("Data de Compra: ");
            gravarArq.println(((Compra)
compras.get(i)).getDtCompra());
            gravarArq.print("Data de Entrega: ");
            gravarArq.println(((Compra)
compras.get(i)).getDtEntrega());
            gravarArq.print("Título do Livro: ");
            gravarArq.println(((Compra)
compras.get(i)).getLivro().getNomeLivro());
            gravarArq.print("Nome do Fornecedor: ");
            gravarArq.println(((Compra)
compras.get(i)).getFornecedor().getNmFornecedor());
            gravarArq.print("Preço Unitário: ");
            gravarArq.println(((Compra)
compras.get(i)).getPreco());
            gravarArq.print("Quantidade: ");
            gravarArq.println(((Compra)
compras.get(i)).getQuantidade());
            gravarArq.println("- - - - -");
            gravarArq.println("- - - - -");
        }
        gravarArq.close();
        return true;

    }catch(Exception e){
        System.out.println("ERRO: "+ e.getMessage());
        return false;
    }
}

public boolean gerarNota(String caminho, Compra compras, int id) {

```

```

Date data = new Date(System.currentTimeMillis());
LivroDAO ld = new LivroDAO();
FornecedorDAO fd = new FornecedorDAO();
try {
    FileWriter arq = new FileWriter(caminho);
    PrintWriter gravarArq = new PrintWriter(arq);
    gravarArq.println("          #NOTA FISCAL - DATA:
"+data+" #          ");
    gravarArq.println(" ESSA NOTA PODE SER USADA
COMO UM COMPROVANTE DE COMPRA,");
    gravarArq.println("JA QUE NELA ESTÃO PRESENTES
INFORMAÇÕES SOBRE O PRODUTO ");
    gravarArq.println("E SOBRE O FORNECEDOR.
");
    gravarArq.println("          #INFORMAÇÕES GERAIS#
");

    gravarArq.print("ID: ");
    gravarArq.println(id);
    gravarArq.print("Data de Compra: ");
    gravarArq.println(compras.getDtCompra());
    gravarArq.print("Data de Entrega: ");
    gravarArq.println(compras.getDtEntrega());
    gravarArq.print("Título do Livro: ");
    int cdLivro = compras.getIdLivro();
    Livro livro = new Livro();
    livro = ld.getLivro(cdLivro);
    gravarArq.println(livro.getNomeLivro());
    gravarArq.print("Nome do Fornecedor: ");
    int cdFornecedor = compras.getIdFornecedor();
    Fornecedor fornecedor = new Fornecedor();
    fornecedor = fd.getFornecedor(cdFornecedor);
    gravarArq.println(fornecedor.getNmFornecedor());
    gravarArq.print("Preço Unitário: ");
    gravarArq.println(compras.getPreco());
    gravarArq.print("Quantidade: ");
    gravarArq.println(compras.getQuantidade());
    gravarArq.println("- - - - -");
");
    gravarArq.println("          VALOR TOTAL:
R$"+compras.getPreco()*compras.getQuantidade());
    gravarArq.println("- - - - -");
");
    gravarArq.println("- - - - -");
");

```

```

        gravarArq.println(" DESTAQUE AQUI ..... DESTAQUE
AQUI ..... DESTAQUE AQUI ");
        gravarArq.println("- - - - -");
    );
        gravarArq.println("          #NOTA FISCAL - DATA:
"+data+" #          ");
        gravarArq.println(" ESSA NOTA PODE SER USADA
COMO UM COMPROVANTE DE COMPRA,");
        gravarArq.println("JA QUE NELA ESTÃO PRESENTES
INFORMAÇÕES SOBRE O PRODUTO ");
        gravarArq.println("E SOBRE O FORNECEDOR.
");
        gravarArq.println("          #INFORMAÇÕES GERAIS#
");

        gravarArq.print("ID: ");
        gravarArq.println(id);
        gravarArq.print("Data de Compra: ");
        gravarArq.println(compras.getDtCompra());
        gravarArq.print("Data de Entrega: ");
        gravarArq.println(compras.getDtEntrega());
        gravarArq.print("Título do Livro: ");
        livro = ld.getLivro(cdLivro);
        gravarArq.println(livro.getNomeLivro());
        gravarArq.print("Nome do Fornecedor: ");
        fornecedor = fd.getFornecedor(cdFornecedor);
        gravarArq.println(fornecedor.getNmFornecedor());
        gravarArq.print("Preço Unitário: ");
        gravarArq.println(compras.getPreco());
        gravarArq.print("Quantidade: ");
        gravarArq.println(compras.getQuantidade());
        gravarArq.println("- - - - -");
    );
        gravarArq.println("          VALOR TOTAL:
R$"+compras.getPreco()*compras.getQuantidade());
        gravarArq.println("- - - - -");
    );

        gravarArq.close();
        return true;
    }catch(Exception e){
        System.out.println("ERRO: "+ e.getMessage());
        return false;
    }
}
}
}

```

//dao> FornecedorDAO.java: classe que manipula todas as informações relativas a classe Fornecedor que se conectam ao banco de dados;

```
package dao;

import java.io.FileWriter;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import com.mysql.jdbc.PreparedStatement;
import bdd.BancoDeDados;
import model.Fornecedor;

public class FornecedorDAO {
    private static PreparedStatement statement = null;
    private static ResultSet resultSet = null;
    int cd;
    List<Fornecedor> fornecedores;

    public void inserir(Fornecedor fornecedor) throws SQLException
    {
        Connection bdd = BancoDeDados.conectar();
        String sql = "INSERT INTO fornecedor(nm_fornecedor,
nm_fantasia, rz_social, cnpj, email, telefone, celular) VALUES
(?,?,?,?,?,?,?);";
        PreparedStatement smt = (PreparedStatement)
bdd.prepareStatement(sql);
        smt.setString(1, fornecedor.getNmFornecedor());
        smt.setString(2, fornecedor.getNmFantasia());
        smt.setString(3, fornecedor.getRzSocial());
        smt.setInt(4, fornecedor.getCnpj());
        smt.setString(5, fornecedor.getEmail());
        smt.setInt(6, fornecedor.getTelefone());
        smt.setInt(7, fornecedor.getCelular());
        smt.executeUpdate();
    }

    public ArrayList<Fornecedor> Listar(){
        Connection bdd = BancoDeDados.conectar();
        fornecedores = new ArrayList<Fornecedor>();
    }
}
```

```

        int linha = 0;
        String texto = null;
        try {
            String sql = "SELECT * FROM fornecedor ORDER
BY cd_fornecedor";
            PreparedStatement smt = (PreparedStatement)
bdd.prepareStatement(sql);
            ResultSet rs = smt.executeQuery();
            while(rs.next()) {
                Fornecedor fornecedor = new Fornecedor();

                fornecedor.setCodigo(rs.getInt("cd_fornecedor"));

                fornecedor.setNmFornecedor(rs.getString("nm_fornecedor"));

                fornecedor.setNmFantasia(rs.getString("nm_fantasia"));

                fornecedor.setRzSocial(rs.getString("rz_social"));
                fornecedor.setCnpj(rs.getInt("cnpj"));
                fornecedor.setEmail(rs.getString("email"));
                fornecedor.setTelefone(rs.getInt("telefone"));
                fornecedor.setCelular(rs.getInt("celular"));
                fornecedores.add(fornecedor);
            }
        } catch (Exception e) {
            System.out.println("ERRO: " + e.getMessage());
        }
        return (ArrayList<Fornecedor>) fornecedores;
    }

    public void deletar(int cd) {
        Connection bdd = BancoDeDados.conectar();
        this.cd = cd;
        try {
            String sql = "delete from fornecedor where
cd_fornecedor = " + cd + " ";
            PreparedStatement smt = (PreparedStatement)
bdd.prepareStatement(sql);
            smt.executeUpdate();
        } catch (Exception e) {
            System.out.println("ERRO: " + e.getMessage());
        }
    }
}

```

```

        public boolean gerarArq(String caminho, List<Fornecedor>
fornecedores) {
            this.fornecedores= fornecedores;
            Date data = new Date(System.currentTimeMillis());
            try {
                FileWriter arq = new FileWriter(caminho);
                PrintWriter gravarArq = new PrintWriter(arq);
                gravarArq.println("# RELATÓRIO DOS
FORNECEDORES CADASTRADOS GERADO EM "+ data + " #");
                for(int i = 0; i<fornecedores.size(); i++) {
                    gravarArq.print("ID: ");
                    gravarArq.println(((Fornecedor)
fornecedores.get(i)).getCodigo());
                    gravarArq.print("Nome: ");
                    gravarArq.println(((Fornecedor)
fornecedores.get(i)).getNmFornecedor());
                    gravarArq.print("Nome Fantasia: ");
                    gravarArq.println(((Fornecedor)
fornecedores.get(i)).getNmFantasia());
                    gravarArq.print("Razão Social: ");
                    gravarArq.println(((Fornecedor)
fornecedores.get(i)).getRzSocial());
                    gravarArq.print("CNPJ: ");
                    gravarArq.println(((Fornecedor)
fornecedores.get(i)).getCnpj());
                    gravarArq.print("E-mail: ");
                    gravarArq.println(((Fornecedor)
fornecedores.get(i)).getEmail());
                    gravarArq.print("Telefone: ");
                    gravarArq.println(((Fornecedor)
fornecedores.get(i)).getTelefone());
                    gravarArq.print("Celular: ");
                    gravarArq.println(((Fornecedor)
fornecedores.get(i)).getCelular());
                    gravarArq.println("- - - - -");
                }
                gravarArq.close();
                return true;
            }catch(Exception e) {
                System.out.println("ERRO: "+ e.getMessage());
                return false;
            }
        }
    }

```

```

public Fornecedor getFornecedor(int cd) {
    Connection bdd = BancoDeDados.conectar();
    fornecedores = new ArrayList<Fornecedor>();
    Fornecedor fornecedor = null;
    String texto = null;
    try {
        String sql = "SELECT * FROM fornecedor WHERE
cd_fornecedor = "+cd+" ";
        PreparedStatement smt = (PreparedStatement)
bdd.prepareStatement(sql);
        ResultSet rs = smt.executeQuery();
        while(rs.next()) {
            fornecedor = new Fornecedor();

fornecedor.setNmFornecedor(rs.getString("nm_fornecedor"));
        }
    }catch(Exception e) {
        System.out.println("ERRO: "+ e.getMessage());
    }
    return fornecedor;
}

public ArrayList<Fornecedor> buscar(int cd){
    Connection bdd = BancoDeDados.conectar();
    fornecedores = new ArrayList<Fornecedor>();
    try {
        String sql = "SELECT * FROM fornecedor WHERE
cd_fornecedor = "+cd+" ";
        PreparedStatement smt = (PreparedStatement)
bdd.prepareStatement(sql);
        ResultSet rs = smt.executeQuery();
        while(rs.next()) {
            Fornecedor f = new Fornecedor();
            f.setCodigo(rs.getInt("cd_fornecedor"));

f.setNmFornecedor(rs.getString("nm_fornecedor"));

f.setNmFantasia(rs.getString("nm_fantasia"));
            f.setRzSocial(rs.getString("rz_social"));
            f.setCnpj(rs.getInt("cnpj"));
            f.setEmail(rs.getString("email"));
            f.setTelefone(rs.getInt("telefone"));
            f.setCelular(rs.getInt("celular"));
        }
    }
}

```



```

        fornecedores.add(f);
    }
} catch (Exception e) {
    System.out.println("ERRO: "+e.getMessage());
}
return (ArrayList<Fornecedor>) fornecedores;
}

public void atualizar(Fornecedor fornecedor, int id) throws
SQLException{
    Connection bdd = BancoDeDados.conectar();
    try {
        String sql = "UPDATE fornecedor SET
nm_fornecedor = ?, nm_fantasia = ?, rz_social = ?, cnpj = ?, email = ?,
telefone = ?, celular = ? WHERE cd_fornecedor = "+id+"";
        PreparedStatement smt = (PreparedStatement)
bdd.prepareStatement(sql);
        smt.setString(1, fornecedor.getNmFornecedor());
        smt.setString(2, fornecedor.getNmFantasia());
        smt.setString(3, fornecedor.getRzSocial());
        smt.setInt(4, fornecedor.getCnpj());
        smt.setString(5, fornecedor.getEmail());
        smt.setInt(6, fornecedor.getTelefone());
        smt.setInt(7, fornecedor.getCelular());
        smt.executeUpdate();
    } catch (Exception e) {
        System.out.println("ERRO: "+e.getMessage());
    }
    bdd.close();
}
}

```

//dao> LivroDAO.java: classe que manipula todas as informações relativas a classe Livro que se conectam ao banco de dados;

```

package dao;

import java.io.FileWriter;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

```

```

import com.mysql.jdbc.PreparedStatement;
import bdd.BancoDeDados;
import model.Livro;

public class LivroDAO {
    private static PreparedStatement statement = null;
    private static ResultSet resultSet = null;
    List<Livro> livros;
    int linha = 0;
    int cd;

    public void inserir(Livro livro) throws SQLException {
        Connection bdd = BancoDeDados.conectar();
        String sql = "INSERT INTO livro(nm_livro, autor, editora,
genero, ano_livro, edicao, preco_venda, qnt_livro, cd_fornecedor,
link_img) VALUES (?,?,?,?,?,?,?,?,?,?)";
        PreparedStatement smt = (PreparedStatement)
bdd.prepareStatement(sql);
        smt.setString(1, livro.getNomeLivro());
        smt.setString(2, livro.getAutorLivro());
        smt.setString(3, livro.getEditoraLivro());
        smt.setString(4, livro.getGeneroLivro());
        smt.setInt(5, livro.getAnoLivro());
        smt.setInt(6, livro.getEdicaoLivro());
        smt.setDouble(7, livro.getPrecoVenda());
        smt.setInt(8, livro.getQntLivro());
        smt.setInt(9, livro.getCdFornecedor());
        smt.setString(10, livro.getLinkImg());
        smt.executeUpdate();
        bdd.close();
    }

    public ArrayList<Livro> buscar(int cd){
        Connection bdd = BancoDeDados.conectar();
        livros = new ArrayList<Livro>();
        try {
            String sql = "SELECT * FROM livro WHERE
cd_livro = "+cd+"";
            PreparedStatement smt = (PreparedStatement)
bdd.prepareStatement(sql);
            ResultSet rs = smt.executeQuery();
            while(rs.next()) {
                Livro livro = new Livro();
                livro.setCdLivro(rs.getInt("cd_Livro"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return livros;
    }
}

```

```

        livro.setNomeLivro(rs.getString("nm_Livro"));
        livro.setAutorLivro(rs.getString("autor"));
        livro.setEditoraLivro(rs.getString("editora"));
        livro.setGeneroLivro(rs.getString("genero"));
        livro.setAnoLivro(rs.getInt("ano_livro"));
        livro.setEdicaoLivro(rs.getInt("edicao"));

        livro.setPrecoVenda(rs.getDouble("preco_venda"));
        livro.setQntLivro(rs.getInt("qnt_livro"));

        livro.setCdFornecedor(rs.getInt("cd_fornecedor"));
        livro.setLinkImg(rs.getString("link_img"));
        livros.add(livro);
    }
} catch (Exception e) {
    System.out.println("ERRO: "+e.getMessage());
}
return (ArrayList<Livro>) livros;
}

public ArrayList<Livro> Listar(){
    Connection bdd = BancoDeDados.conectar();
    livros = new ArrayList<Livro>();
    String texto = null;
    try {
        String sql = "SELECT * FROM livro ORDER BY
cd_livro";
        PreparedStatement smt = (PreparedStatement)
bdd.prepareStatement(sql);
        ResultSet rs = smt.executeQuery();
        while(rs.next()) {
            Livro livro = new Livro();
            livro.setCdLivro(rs.getInt("cd_Livro"));

            livro.setNomeLivro(rs.getString("nm_Livro"));
            livro.setAutorLivro(rs.getString("autor"));
            livro.setEditoraLivro(rs.getString("editora"));
            livro.setGeneroLivro(rs.getString("genero"));
            livro.setAnoLivro(rs.getInt("ano_livro"));
            livro.setEdicaoLivro(rs.getInt("edicao"));

            livro.setPrecoVenda(rs.getDouble("preco_venda"));
            livro.setQntLivro(rs.getInt("qnt_livro"));

```

```

        livro.setCdFornecedor(rs.getInt("cd_fornecedor"));
        livro.setLinkImg(rs.getString("link_img"));
        livros.add(livro);
    }
} catch (Exception e) {
    System.out.println("ERRO: " + e.getMessage());
}
return (ArrayList<Livro>) livros;
}

public void deletar(int cd) {
    Connection bdd = BancoDeDados.conectar();
    this.cd = cd;
    try {
        String sql = "delete from livro where cd_livro = "+
cd + " ";
        PreparedStatement smt = (PreparedStatement)
bdd.prepareStatement(sql);
        smt.executeUpdate();
    } catch (Exception e) {
        System.out.println("ERRO: " + e.getMessage());
    }
}

public boolean gerarArq( String caminho, List<Livro> livros) {
    this.livros = livros;
    Date data = new Date(System.currentTimeMillis());
    try {
        FileWriter arq = new FileWriter(caminho);
        PrintWriter gravarArq = new PrintWriter(arq);
        gravarArq.println("#RELATÓRIO DOS LIVROS
CADASTRADOS GERADO EM "+ data + "#");
        for (int i = 0; i < livros.size(); i++) {
            gravarArq.print("ID: ");
            gravarArq.println(((Livro)
livros.get(i)).getCdLivro());
            gravarArq.print("Título: ");
            gravarArq.println(((Livro)
livros.get(i)).getNomeLivro());
            gravarArq.print("Autor: ");
            gravarArq.println(((Livro)
livros.get(i)).getAutorLivro());
            gravarArq.print("Editora: ");

```

```

                                gravarArq.println(((Livro)
livros.get(i)).getEditoraLivro());
                                gravarArq.print("Gênero: ");
                                gravarArq.println(((Livro)
livros.get(i)).getGeneroLivro());
                                gravarArq.print("Ano: ");
                                gravarArq.println(((Livro)
livros.get(i)).getAnoLivro());
                                gravarArq.print("Edição: ");
                                gravarArq.println(((Livro)
livros.get(i)).getEdicaoLivro());
                                gravarArq.print("Preço de Venda");
                                gravarArq.println(((Livro)
livros.get(i)).getPrecoVenda());
                                gravarArq.print("Quantidade: ");
                                gravarArq.println(((Livro)
livros.get(i)).getQntLivro());
                                gravarArq.print("Código do Fornecedor: ");
                                gravarArq.println(((Livro)
livros.get(i)).getCdFornecedor());
                                gravarArq.print("Link Imagem");
                                gravarArq.println(((Livro)
livros.get(i)).getLinkImg());
                                gravarArq.println("- - - - -");
- - - - -");
                                }
                                gravarArq.close();
                                return true;
                                }catch(Exception e) {
                                    System.out.println("ERRO: "+ e.getMessage());
                                    return false;
                                }
                            }
                        }

public Livro getLivro (int cd){
    Connection bdd = BancoDeDados.conectar();
    livros = new ArrayList<Livro>();
    Livro livro= null;
    String texto = null;
    try {
        String sql = "SELECT * FROM livro WHERE
cd_livro = "+cd+"";
        PreparedStatement smt = (PreparedStatement)
        bdd.prepareStatement(sql);

```

```

        ResultSet rs = smt.executeQuery();
        while(rs.next()) {
            System.out.println("ENTROU3");
            livro = new Livro();

            livro.setNomeLivro(rs.getString("nm_Livro"));

        }
    } catch (Exception e) {
        System.out.println("ERRO: "+e.getMessage());
    }
    return livro;
}

public void atualizar(Livro livro, int id) throws SQLException{
    Connection bdd = BancoDeDados.conectar();
    try {
        String sql = "UPDATE livro SET nm_livro = ?, autor
= ?, editora = ?,genero = ? , ano_livro = ?, edicao = ?,preco_venda =
?,qnt_livro = ?,cd_fornecedor = ?,link_img = ? WHERE cd_livro =
"+id+";";

        PreparedStatement smt = (PreparedStatement)
bdd.prepareStatement(sql);
        smt.setString(1, livro.getNomeLivro());
        smt.setString(2, livro.getAutorLivro());
        smt.setString(3, livro.getEditoraLivro());
        smt.setString(4, livro.getGeneroLivro());
        smt.setInt(5, livro.getAnoLivro());
        smt.setInt(6, livro.getEdicaoLivro());
        smt.setDouble(7, livro.getPrecoVenda());
        smt.setInt(8, livro.getQntLivro());
        smt.setInt(9, livro.getCdFornecedor());
        smt.setString(10, livro.getLinkImg());
        smt.executeUpdate();
    } catch (Exception e) {
        System.out.println("ERRO: "+e.getMessage());
    }
    bdd.close();
}
}

```

//dao> SistemaDAO.java: classe que manipula todas as informações relativas a classe Sistema que se conectam ao banco de dados;

```
package dao;
```

```

import java.io.FileWriter;
import java.io.PrintWriter;
import javax.swing.JOptionPane;

public class SistemaDAO {
    String caminho;
    String texto;

    public boolean gerarArq(String caminho, String texto) {
        try {
            FileWriter arq = new FileWriter(caminho);
            PrintWriter gravarArq = new PrintWriter(arq);
            gravarArq.println("#                TEXTO
DE AJUDA                #");
            gravarArq.println("#
#");
            gravarArq.println("# ESSE PROGRAMA TEM A
FUNÇÃO DE DISPONIBILIZAR ACESSO AO BANCO DE DADOS E
SUAS #");
            gravarArq.println("# POSSÍVEIS ALTERAÇÕES.
BASEADO NA LINGUAGEM DE PROGRAMAÇÃO JAVA, O SISTEMA
TEM #");
            gravarArq.println("# FUNCIONALIDADES
ACESSÍVEIS POR MEIO DA BARRA DE TAREFAS LOCALIZADO NA
REGIÃO #");
            gravarArq.println("# SUPERIOR DA TELA.
#");
            gravarArq.println("#
#");
            gravarArq.println("# O MENU:
#");
            gravarArq.println("# NO MENU PODEMOS
LOCALIZAR QUATRO TÓPICOS DE TRABALHOS, NA QUAL CADA
UM DELES #");
            gravarArq.println("# POSSUI SUA DETERMINADA
FUNÇÃO. A SEGUIR EXPLICAREMOS A FUNÇÃO DOS TÓPICOS,
NA #");
            gravarArq.println("# QUAL ESTÃO DIVIDIDOS EM:
#");
            gravarArq.println("#
#");
            gravarArq.println("# 1- CADASTRO
#");

```

```

        gravarArq.println("# O CADASTRO É UM TÓPICO
ESPECÍFICO PARA REALIZAR QUALQUER TIPO DE CADASTRO
NO #");

        gravarArq.println("# BANCO DE DADOS. ESTE
TÓPICO POSSUI SUBTÓPICOS, SENDO ESTES:      #");
        gravarArq.println("#
#");

        gravarArq.println("# 1.1- CADASTRAR LIVRO
#");

        gravarArq.println("# ESSE TÓPICO REALIZA A
FUNÇÃO DE CADASTRAR UM LIVRO NO BANCO DE DADOS.
BASTA #");

        gravarArq.println("# PREENCHER TODOS OS
CAMPOS CORRETAMENTE E CLICAR EM CONCLUIR QUE O
LIVRO SERÁ #");

        gravarArq.println("# INCLUSO NO BANCO DE
DADOS. VALE RESSALTAR QUE PARA O CADASTRO DO LIVRO É
#");

        gravarArq.println("# OBRIGATÓRIO A SELEÇÃO
DE UM FORNECEDOR PARA QUE O CADASTRO SEJA
CONCLUÍDO COM #");
        gravarArq.println("# PERFEIÇÃO.
#");

        gravarArq.println("#
#");

        gravarArq.println("# 1.2- CADASTRAR
FORNECEDOR      #");

        gravarArq.println("# ESSE TÓPICO REALIZA A
FUNÇÃO DE CADASTRAR UM FORNECEDOR NO BANCO DE
DADOS. É #");

        gravarArq.println("# SÓ PREENCHER TODOS OS
CAMPOS CORRETAMENTE E CLICAR EM CONCLUIR QUE O
TODOS OS #");

        gravarArq.println("# DADOS DO FORNECEDOR
SERÃO SALVOS.      #");
        gravarArq.println("#
#");

        gravarArq.println("# 2- COMPRAS
#");

        gravarArq.println("# NESSE TÓPICO, POSSUI
APENAS UM SUBTÓPICO, SENDO ELE O GERAR COMPRA, E
ESTE, #");

```



```

                                gravarArq.println("# TEM A FUNÇÃO DE INSERIR
NO BANCO DE DADOS UMA COMPRA REALIZADA. PARA TAL
AÇÃO #");
                                gravarArq.println("# BASTA QUE SEJA
SELECIONADO UM FORNECEDOR EXISTENTE E UM LIVRO
EXISTENTE. CASO #");
                                gravarArq.println("# NÃO EXISTA NENHUM
LIVRO OU FORNECEDOR CADASTRADO NO BANCO,
OBRIGATORIAMENTE #");
                                gravarArq.println("# EXIGE-SE QUE SEJA
CADASTRADO UM FORNECEDOR E UM LIVRO
RESPECTIVAMENTE, E POR #");
                                gravarArq.println("# FIM, SEJA GERADO UMA
COMPRA.                                #");
                                gravarArq.println("# NO FINAL DE UMA
COMPRA, É EMITIDO UMA NOTA FISCAL, E ESTE ARQUIVO É
SALVO NA #");
                                gravarArq.println("# PRÓPRIA PASTA DO
PROGRAMA.                                #");
                                gravarArq.println("#
#");
                                gravarArq.println("# 3-LISTAR
#");
                                gravarArq.println("# ESSE TÓPICO TEM A
FUNÇÃO DE APRESENTAR EM UMA TABELA ALGUMAS
INFORMAÇÕES QUE #");
                                gravarArq.println("# SÃO SALVAS NO BANCO
DE DADOS REFERENTES A CADA SUBTOPICO, SENDO ELES:
#");
                                gravarArq.println("# - LISTAR LIVROS
#");
                                gravarArq.println("# LISTA TODOS OS LIVROS
CADASTRADOS NO BANCO DE DADOS, ORDENADOS PELO
CÓDIGO #");
                                gravarArq.println("# DE IDENTIFICAÇÃO. AS
COLUNAS DA TABELA SÃO: CÓDIGO; TÍTULO; QUANTIDADE.
#");
                                gravarArq.println("# ESSA TELA POSSUI TRÊS
BOTÕES: DELETAR(APAGA O LIVRO SELECIONADO NA TABELA
#");
                                gravarArq.println("# DO BANCO DE DADOS);
ATUALIZAR(ATUALIZA O LIVRO SELECIONADO NA TABELA DO
#");

```

```

        gravarArq.println("# BANCO DE DADOS); E
GERAR ARQUIVO(GERA UM ARQUIVO TEXTO, SALVO NA
PRÓPRIA  #");
        gravarArq.println("# PASTA DO PROGRAMA,
COM TODAS AS INFORMAÇÕES DE TODOS OS LIVROS
CADASTRADOS  #");
        gravarArq.println("# NO BANCO DE DADOS.
#");
        gravarArq.println("#
#");
        gravarArq.println("# - LISTAR FORNECEDOR
#");
        gravarArq.println("# LISTA TODOS OS
FORNECEDORES CADASTRADOS NO BANCO DE DADOS,
ORDENADOS PELO  #");
        gravarArq.println("# CÓDIGO DE
IDENTIFICAÇÃO. AS COLUNAS DA TABELA SÃO: ID; NOME
FORNECEDOR;  #");
        gravarArq.println("# TELEFONE; E-MAIL.
#");
        gravarArq.println("# ESSA TELA POSSUI TRÊS
BOTÕES: DELETAR(APAGA O FORNECEDOR SELECIONADO NA
#");
        gravarArq.println("# TABELA DO BANCO DE
DADOS); ATUALIZAR(ATUALIZA O FORNECEDOR SELECIONADO
NA  #");
        gravarArq.println("# TABELA DO BANCO DE
DADOS COM AS NOVAS INFORMAÇÕES PASSADAS); E GERAR
ARQUIVO #");
        gravarArq.println("# (GERA UM ARQUIVO
TEXTO, SALVO NA PRÓPRIA PASTA DO PROGRAMA, COM TODAS
AS  #");
        gravarArq.println("# INFORMAÇÕES DE TODOS
OS FORNECEDORES CADASTRADOS NO BANCO DE DADOS.
#");
        gravarArq.println("#
#");
        gravarArq.println("# - LISTAR CLIENTES
#");
        gravarArq.println("# LISTA TODOS OS
CLIENTES CADASTRADOS NO BANCO DE DADOS, ORDENADOS
PELO  #");

```

```

        gravarArq.println("# CÓDIGO DE
IDENTIFICAÇÃO. AS COLUNAS DA TABELA SÃO: ID; NOME; CPF;
CELULAR. #");
        gravarArq.println("# ESSA TELA POSSUI DOIS
BOTÕES: DELETAR(APAGA O CLIENTE SELECIONADO NA
TABELA #");
        gravarArq.println("# DO BANCO DE DADOS) E
GERAR ARQUIVO(GERA UM ARQUIVO TEXTO, SALVO NA
PRÓPRIA #");
        gravarArq.println("# PASTA DO PROGRAMA,
COM TODAS AS INFORMAÇÕES DE TODOS OS CLIENTES
CADASTRADOS #");
        gravarArq.println("# NO BANCO DE DADOS.
#");
        gravarArq.println("#
#");
        gravarArq.println("# - LISTAR COMPRAS
#");
        gravarArq.println("# LISTA TODOS AS
COMPRAS CADASTRADAS NO BANCO DE DADOS, ORDENADAS
PELO CÓDIGO #");
        gravarArq.println("# DE IDENTIFICAÇÃO. AS
COLUNAS DA TABELA SÃO: ID DA COMPRA; DT DE COMPRA;
#");
        gravarArq.println("# DT DE ENTREGA; LIVRO;
FORNECEDOR; PREÇO UNITÁRIO. #");
        gravarArq.println("# ESSA TELA POSSUI
APENAS UM BOTÃO, O GERAR ARQUIVO CUJA FUNÇÃO É
GERAR UM #");
        gravarArq.println("# ARQUIVO TEXTO, SALVO
NA PRÓPRIA PASTA DO PROGRAMA, COM TODAS AS
INFORMAÇÕES #");
        gravarArq.println("# DE TODAS AS COMPRAS
CADASTRADAS NO BANCO DE DADOS. #");
        gravarArq.println("#
#");
        gravarArq.close();
        return true;
    }catch(Exception e) {
        JOptionPane.showMessageDialog(null,
"ERRO:"+e.getMessage());
        return false;
    }
}

```

```
}
```

//dao> UsuarioDAO.java: classe que manipula todas as informações relativas a classe Usuario que se conectam ao banco de dados;

```
package dao;

import java.io.FileWriter;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.Date;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;
import com.mysql.jdbc.PreparedStatement;
import bdd.BancoDeDados;
import model.Usuario;

public class UsuarioDAO {
    int cd;
    List<Usuario> usuarios;
    public ArrayList<Usuario> Listar(){
        Connection bdd = BancoDeDados.conectar();
        usuarios = new ArrayList<Usuario>();
        int linha = 0;
        String texto = null;
        try {
            String sql = "SELECT * FROM usuario ORDER BY
cd_usuario";
            PreparedStatement smt = (PreparedStatement)
bdd.prepareStatement(sql);
            ResultSet rs = smt.executeQuery();
            while(rs.next()) {
                Usuario usuario = new Usuario();

                usuario.setCdUsuario(rs.getInt("cd_usuario"));

                usuario.setNmUsuario(rs.getString("nm_usuario"));

                usuario.setCpfUsuario(rs.getInt("cpf_usuario"));
                usuario.setSexo(rs.getString("sexo"));

                usuario.setDtNascimento(rs.getString("data_nascimento"));
                usuario.setEmail(rs.getString("email"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

        usuario.setCelular(rs.getInt("celular"));
        usuario.setPerfil(rs.getString("perfil"));
        usuario.setLogin(rs.getString("login"));
        usuario.setSenha(rs.getString("senha"));
        usuarios.add(usuario);
    }
} catch (Exception e) {
    System.out.println("ERRO: " + e.getMessage());
}

return (ArrayList<Usuario>) usuarios;
}

public void deletar(int cd) {
    Connection bdd = BancoDeDados.conectar();
    this.cd = cd;
    try {
        String sql = "delete from usuario where cd_usuario
= "+ cd + " ";
        PreparedStatement smt = (PreparedStatement)
        bdd.prepareStatement(sql);
        smt.executeUpdate();
    } catch (Exception e) {
        System.out.println("ERRO: "+ e.getMessage());
    }
}

public boolean gerarArq(String caminho, List<Usuario> usuarios)
{
    this.usuarios=usuarios;
    Date data = new Date(System.currentTimeMillis());
    try {
        FileWriter arq = new FileWriter(caminho);
        PrintWriter gravarArq = new PrintWriter(arq);
        gravarArq.println("# RELATÓRIO DOS CLIENTES
CADASTRADOS GERADO EM "+data+" #");
        for(int i = 0; i<usuarios.size(); i++) {
            gravarArq.print("ID: ");
            gravarArq.println(((Usuario)
usuarios.get(i)).getCdUsuario());
            gravarArq.print("Nome: ");
            gravarArq.println(((Usuario)
usuarios.get(i)).getNmUsuario());
            gravarArq.print("CPF: ");

```

```

        usuarios.get(i).getCpfUsuario());
        gravarArq.println(((Usuario)
        usuarios.get(i).getCelular());
        gravarArq.print("Celular: ");
        gravarArq.println(((Usuario)
        usuarios.get(i).getSexo());
        gravarArq.print("Sexo: ");
        gravarArq.println(((Usuario)
        usuarios.get(i).getDtNascimento());
        gravarArq.print("Data de Nascimento: ");
        gravarArq.println(((Usuario)
        usuarios.get(i).getEmail());
        gravarArq.print("E-mail: ");
        gravarArq.println(((Usuario)
        usuarios.get(i).getPerfil());
        gravarArq.print("Perfil: ");
        gravarArq.println(((Usuario)
        usuarios.get(i).getLogin());
        gravarArq.print("Login: ");
        gravarArq.println(((Usuario)
        usuarios.get(i).getSenha());
        gravarArq.print("Senha: ");
        gravarArq.println(((Usuario)
        -----
        -----");
    }
    gravarArq.close();
    return true;
} catch (Exception e) {
    System.out.println("ERRO: "+ e.getMessage());
    return false;
}
}
}
}

```

```

//Projeto>src>model: pacote responsável pela modelação das classes;
//model> Compra: classe responsável pela modelação da classe Compra;
package model;

public class Compra {
    private int cdCompra;
    private String dtCompra;
    private String dtEntrega;
    private Fornecedor fornecedor;

```

```

private Livro livro;
private int quantidade;
private double preco;

public Compra(){
    livro = new Livro();
    fornecedor = new Fornecedor();
}

public int getCdCompra() {
    return cdCompra;
}

public void setCdCompra(int cdCompra) {
    this.cdCompra = cdCompra;
}

public String getDtCompra() {
    return dtCompra;
}

public void setDtCompra(String dtCompra) {
    this.dtCompra = dtCompra;
}

public String getDtEntrega() {
    return dtEntrega;
}

public void setDtEntrega(String dtEntrega) {
    this.dtEntrega = dtEntrega;
}

public Fornecedor getFornecedor() {
    return fornecedor;
}

public void setFornecedor(Fornecedor fornecedor) {
    this.fornecedor = fornecedor;
}

public Livro getLivro() {
    return livro;
}

```

```

    public void setLivro(Livro cdLivro) {
        this.livro = cdLivro;
    }

    public int getQuantidade() {
        return quantidade;
    }

    public void setQuantidade(int quantidade) {
        this.quantidade = quantidade;
    }

    public double getPreco() {
        return preco;
    }

    public void setPreco(double preco) {
        this.preco = preco;
    }

    public int getIdLivro() {
        return livro.getCdLivro();
    }

    public int getIdFornecedor() {
        return fornecedor.getCodigo();
    }

    @Override
    public String toString() {
        return "Compra [cdCompra=" + cdCompra + ",
dtCompra=" + dtCompra + ", dtEntrega=" + dtEntrega
        + ", cdFornecedor=" +
fornecedor.getCodigo()+ ", cdLivro=" + livro.getCdLivro() + ",
quantidade=" + quantidade + ", preco="
        + preco + "]";
    }
}

```

//model> CompraTableModel: classe responsável pela modelação da tabela que apresenta os dados da classe Compra;

```
package model;
```



```

import java.util.ArrayList;
import java.util.List;
import javax.swing.table.AbstractTableModel;

public class CompraTableModel extends AbstractTableModel{
    private List<Compra> dados = new ArrayList<Compra>();
    private String[] colunas = {"ID da Compra", "Dt de Compra", "Dt
de Entrega", "Livro", "Fornecedor", "Preço Unitário"};

    @Override
    public String getColumnName(int column) {
        return colunas[column];
    }

    @Override
    public int getColumnCount() {
        return colunas.length;
    }

    @Override
    public int getRowCount() {
        return dados.size();
    }

    @Override
    public Object getValueAt(int linha, int coluna) {
        switch(coluna) {
            case 0:
                return dados.get(linha).getCdCompra();
            case 1:
                return dados.get(linha).getDtCompra();
            case 2:
                return dados.get(linha).getDtEntrega();
            case 3:
                return dados.get(linha).getLivro().getNomeLivro();
            case 4:
                return
dados.get(linha).getFornecedor().getNmFornecedor();
            case 5:
                return dados.get(linha).getPreco();
            case 6:
                return dados.get(linha).getQuantidade();
        }
        return null;
    }
}

```

```

    }

    public void addRow(Compra c) {
        this.dados.add(c);
    }

    public int getSelectRow(int linha) {
        return dados.get(linha).getCdCompra();
    }

    public void removeAll() {
        this.dados.clear();
    }
}

```

//model> Fornecedor: classe responsável pela modelação da classe Fornecedor;

```

package model;

public class Fornecedor {
    private int codigo;
    private String nmFornecedor;
    private String nmFantasia;
    private String rzSocial;
    private int cnpj;
    private String email;
    private int telefone;
    private int celular;

    public Fornecedor(){
    }

    public String getNmFornecedor() {
        return nmFornecedor;
    }

    public void setNmFornecedor(String nmFornecedor) {
        this.nmFornecedor=nmFornecedor;
    }

    public String getNmFantasia() {
        return nmFantasia;
    }
}

```

```

    }

    public void setNmFantasia(String nmFantasia) {
        this.nmFantasia=nmFantasia;
    }

    public String getRzSocial() {
        return rzSocial;
    }

    public void setRzSocial(String rzSocial) {
        this.rzSocial=rzSocial;
    }

    public int getCnpj() {
        return cnpj;
    }

    public void setCnpj(int cnpj) {
        this.cnpj=cnpj;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email=email;
    }

    public int getTelefone() {
        return telefone;
    }

    public void setTelefone(int telefone) {
        this.telefone=telefone;
    }

    public int getCelular() {
        return celular;
    }

    public void setCelular(int celular) {
        this.celular=celular;
    }

```

```

    }

    public int getCodigo() {
        return codigo;
    }

    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }
}

```

//model> FornecedorTableModel: classe responsável pela modelação da tabela que apresenta os dados da classe Fornecedor;

```
package model;
```

```

import java.util.ArrayList;
import java.util.List;
import javax.swing.JTable;
import javax.swing.table.AbstractTableModel;

```

```

public class FornecedorTableModel extends AbstractTableModel{
    private List<Fornecedor> dados = new ArrayList<Fornecedor>();
    private String[] colunas = {"ID", "Nome Fornecedor", "Telefone",
    "E-mail"};

```

```

    @Override
    public String getColumnName(int column) {
        return colunas[column];
    }

```

```

    @Override
    public int getColumnCount() {
        return colunas.length;
    }

```

```

    @Override
    public int getRowCount() {
        return dados.size();
    }

```

```

    @Override
    public Object getValueAt(int linha, int coluna) {
        switch(coluna) {
            case 0:

```

```

        return dados.get(linha).getCodigo();
    case 1:
        return dados.get(linha).getNmFornecedor();
    case 2:
        return dados.get(linha).getTelefone();
    case 3:
        return dados.get(linha).getEmail();
    case 4:
        return dados.get(linha).getNmFantasia();
    case 5:
        return dados.get(linha).getRzSocial();
    case 6:
        return dados.get(linha).getCnpj();
    case 7:
        return dados.get(linha).getCelular();
    }
    return null;
}

public void addROw(Fornecedor f) {
    this.dados.add(f);
}

public int getSelectRow(int linha) {
    return dados.get(linha).getCodigo();
}

public void removeAll() {
    this.dados.clear();
}
}

```

//model> Livro: classe responsável pela modelação da classe Livro;
package model;

```

import java.awt.Component;

public class Livro {
    private int cdLivro;
    private String nmLivro;
    private String autorLivro;
    private String editoraLivro;
    private String generoLivro;
    private int anoLivro;

```

```

private int edicaoLivro;
private double precoVenda;
private int qntLivro;
private int cdFornecedor;
private String linkImg;

public Livro(){
}

public int getCdLivro() {
    return cdLivro;
}

public void setCdLivro(int cdLivro) {
    this.cdLivro = cdLivro;
}

public String getNomeLivro() {
    return nmLivro;
}

public void setNomeLivro(String nome) {
    this.nmLivro = nome;
}

public String getAutorLivro() {
    return autorLivro;
}

public void setAutorLivro(String autorLivro) {
    this.autorLivro = autorLivro;
}

public String getEditoraLivro() {
    return editoraLivro;
}

public void setEditoraLivro(String editoraLivro) {
    this.editoraLivro = editoraLivro;
}

public String getGeneroLivro(){
    return generoLivro;
}

```

```

public void setGeneroLivro(String generoLivro) {
    this.generoLivro = generoLivro;
}

public int getAnoLivro() {
    return anoLivro;
}

public void setAnoLivro(int anoLivro) {
    this.anoLivro = anoLivro;
}

public int getEdicaoLivro() {
    return edicaoLivro;
}

public void setEdicaoLivro(int edicaoLivro) {
    this.edicaoLivro = edicaoLivro;
}

public double getPrecoVenda() {
    return precoVenda;
}

public void setPrecoVenda(double precoLivro) {
    this.precoVenda=precoLivro;
}

public int getQntLivro() {
    return qntLivro;
}

public void setQntLivro(int qntLivro) {
    this.qntLivro=qntLivro;
}

public int getCdFornecedor() {
    return cdFornecedor;
}

public void setCdFornecedor(int cdFornecedor) {
    this.cdFornecedor=cdFornecedor;
}

```

```

        public String getLinkImg() {
            return linkImg;
        }

        public void setLinkImg(String linkImg) {
            this.linkImg=linkImg;
        }
    }

```

//model> LivroTableModel: classe responsável pela modelação da tabela que apresenta os dados da classe Livro;

```

package model;

import java.util.ArrayList;
import java.util.List;
import javax.swing.table.AbstractTableModel;

public class LivroTableModel extends AbstractTableModel{
    private List<Livro> dados = new ArrayList<Livro>();
    private String[] colunas = {"Código", "Título", "Quantidade"};

    @Override
    public String getColumnName(int column) {
        return colunas[column];
    }

    @Override
    public int getColumnCount() {
        return colunas.length;
    }

    @Override
    public int getRowCount() {
        return dados.size();
    }

    @Override
    public Object getValueAt(int linha, int coluna) {
        switch(coluna) {
            case 0:
                return dados.get(linha).getCdLivro();
            case 1:
                return dados.get(linha).getNomeLivro();

```



```

        case 2:
            return dados.get(linha).getQntLivro();
        case 3:
            return dados.get(linha).getPrecoVenda();
        case 4:
            return dados.get(linha).getAnoLivro();
        case 5:
            return dados.get(linha).getAutorLivro();
        case 6:
            return dados.get(linha).getCdFornecedor();
        case 7:
            return dados.get(linha).getEdicaoLivro();
        case 8:
            return dados.get(linha).getEditoraLivro();
        case 9:
            return dados.get(linha).getGeneroLivro();
        case 10:
            return dados.get(linha).getLinkImg();
    }
    return null;
}

public void addRow(Livro l) {
    this.dados.add(l);
}

public int getSelectRow(int linha) {
    return dados.get(linha).getCdLivro();
}

public void removeAll() {
    this.dados.clear();
}
}

```

//model> Usuario: classe responsável pela modelação da classe Usuario;
package model;

```

public class Usuario {
    private String nmUsuario;
    private int cpfUsuario;
    private String sexo;
    private String dtNascimento;
    private String email;
}

```

```

private int celular;
private String perfil;
private String login;
private String senha;
private int cdUsuario;

public Usuario() {
}

public int getCdUsuario() {
    return cdUsuario;
}

public void setCdUsuario(int cdUsuario) {
    this.cdUsuario=cdUsuario;
}

public String getNmUsuario(){
    return nmUsuario;
}

public void setNmUsuario(String nmUsuario) {
    this.nmUsuario=nmUsuario;
}

public int getCpfUsuario() {
    return cpfUsuario;
}

public void setCpfUsuario(int cpfUsuario) {
    this.cpfUsuario=cpfUsuario;
}

public String getSexo() {
    return sexo;
}

public void setSexo(String sexo) {
    this.sexo=sexo;
}

public String getDtNascimento(){
    return dtNascimento;
}

```

```
public void setDtNascimento(String dtNascimento) {
    this.dtNascimento= dtNascimento;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email=email;
}

public int getCelular() {
    return celular;
}

public void setCelular(int celular) {
    this.celular=celular;
}

public String getPerfil() {
    return perfil;
}

public void setPerfil(String perfil) {
    this.perfil=perfil;
}

public String getLogin() {
    return login;
}

public void setLogin(String login) {
    this.login=login;
}

public String getSenha() {
    return senha;
}

public void setSenha(String senha) {
    this.senha=senha;
}
```

```
    }  
}
```

//model> UsuarioTableModel: classe responsável pela modelação da tabela que apresenta os dados da classe Usuario;

```
package model;  
  
import java.util.ArrayList;  
import java.util.List;  
import javax.swing.table.AbstractTableModel;  
  
public class UsuarioTableModel extends AbstractTableModel{  
    private List<Usuario> dados = new ArrayList<Usuario>();  
    private String[] colunas = {"ID", "Nome", "CPF", "Celular"};  
  
    @Override  
    public String getColumnName(int column) {  
        return colunas[column];  
    }  
  
    @Override  
    public int getColumnCount() {  
        return colunas.length;  
    }  
  
    @Override  
    public int getRowCount() {  
        return dados.size();  
    }  
  
    @Override  
    public Object getValueAt(int linha, int coluna) {  
        switch(coluna) {  
            case 0:  
                return dados.get(linha).getCdUsuario();  
            case 1:  
                return dados.get(linha).getNmUsuario();  
            case 2:  
                return dados.get(linha).getCpfUsuario();  
            case 3:  
                return dados.get(linha).getCelular();  
            case 4:  
                return dados.get(linha).getSexo();  
            case 5:  
                return dados.get(linha).getEndereco();  
        }  
    }  
}
```

```

        return dados.get(linha).getDtNascimento();
    case 6:
        return dados.get(linha).getEmail();
    case 7:
        return dados.get(linha).getPerfil();
    case 8:
        return dados.get(linha).getLogin();
    case 9:
        return dados.get(linha).getSenha();
    }
    return null;
}

public void addRow(Usuario u) {
    this.dados.add(u);
}

public int getSelectRow(int linha) {
    return dados.get(linha).getCdUsuario();
}

public void removeAll() {
    this.dados.clear();
}
}

```

//Projeto>src>view: pacote responsável pela apresentação visual;
 //view> AtualizarFornecedorView: classe responsável pela apresentação visual da tela de atualização do Fornecedor;
 package view;

```

import java.awt.Font;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.HeadlessException;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;

```

```

import javax.swing.JOptionPane;
import javax.swing.JTextField;

import controller.FornecedorController;
import dao.FornecedorDAO;
import model.Fornecedor;

public class AtualizarFornecedorView extends JFrame implements
ActionListener{
    private JTextField nmFornecedorField;
    private JTextField nmFantasiaField;
    private JTextField rzSocialField;
    private JTextField cnpjField;
    private JTextField emailField;
    private JTextField telefoneField;
    private JTextField celularField;
    private JTextField cdFornecedorField;
    JFrame frameList;
    private int id;
    private Fornecedor fornecedor;
    ArrayList<Fornecedor> fornecedores;

    public AtualizarFornecedorView(int id) {
        this.setSize(800,500);
        this.setResizable(false);
        this.setVisible(true);
        this.id = id;
        criarFormulario();
    }

    private void criarFormulario() {
        FornecedorDAO fd = new FornecedorDAO();
        fd.buscar(id);
        fornecedores = (ArrayList<Fornecedor>) fd.buscar(id);
        setLayout(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.weightx=0.1;
        gbc.weighty=0.1;
        JLabel titulo = new JLabel("Cadastro de Fornecedor");
        titulo.setFont(new Font("Arial",
Font.CENTER_BASELINE, 40));
        gbc.gridx=0;
        gbc.gridy=0;
        gbc.gridwidth = GridBagConstraints.REMAINDER;

```

```

        gbc.fill = GridBagConstraints.LINE_START;
        gbc.anchor = GridBagConstraints.NORTH;
        add(titulo,gbc);
        JLabel nmFornecedorLabel = new JLabel ("Nome do
Fornecedor");
        nmFornecedorLabel.setFont(new Font("Arial",
Font.BOLD, 16));
        gbc.gridx=0;
        gbc.gridy=1;
        gbc.insets= new Insets(10,0,0,5);
        gbc.gridwidth = GridBagConstraints.RELATIVE;
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.LINE_END;
        add(nmFornecedorLabel, gbc);
        nmFornecedorField = new JTextField(30);

        nmFornecedorField.setText(fornecedores.get(0).getNmForneced
or());
        gbc.gridx=1;
        gbc.insets= new Insets(10,0,0,5);
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.LINE_START;
        add(nmFornecedorField, gbc);
        JLabel nmFantasiaLabel = new JLabel ("Nome
Fantasia");
        nmFantasiaLabel.setFont(new Font("Arial", Font.BOLD,
16));
        gbc.gridx=0;
        gbc.gridy=2;
        gbc.insets= new Insets(10,0,0,5);
        gbc.gridwidth = GridBagConstraints.RELATIVE;
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.LINE_END;
        add(nmFantasiaLabel, gbc);
        nmFantasiaField = new JTextField(30);

        nmFantasiaField.setText(fornecedores.get(0).getNmFantasia());
        gbc.gridx=1;
        gbc.insets= new Insets(10,0,0,5);
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.LINE_START;
        add(nmFantasiaField, gbc);
        JLabel rzSocialLabel = new JLabel ("razão Social");
        rzSocialLabel.setFont(new Font("Arial", Font.BOLD, 16));

```

```

gbc.gridx=0;
gbc.gridy=3;
gbc.insets= new Insets(10,0,0,5);
gbc.gridwidth = GridBagConstraints.RELATIVE;
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(rzSocialLabel, gbc);
rzSocialField = new JTextField(30);
rzSocialField.setText(fornecedores.get(0).getRzSocial());
gbc.gridx=1;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(rzSocialField, gbc);
JLabel cnpjLabel = new JLabel ("CNPJ");
cnpjLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx=0;
gbc.gridy=4;
gbc.insets= new Insets(10,0,0,5);
gbc.gridwidth = GridBagConstraints.RELATIVE;
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(cnpjLabel, gbc);
cnpjField = new JTextField(30);
String cnpj =
String.valueOf(fornecedores.get(0).getCnpj());
cnpjField.setText(cnpj);
gbc.gridx=1;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(cnpjField, gbc);
JLabel emailLabel = new JLabel ("Email");
emailLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx=0;
gbc.gridy=5;
gbc.insets= new Insets(10,0,0,5);
gbc.gridwidth = GridBagConstraints.RELATIVE;
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(emailLabel, gbc);
emailField = new JTextField(30);
emailField.setText(fornecedores.get(0).getEmail());
gbc.gridx=1;

```



```

gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(emailField, gbc);
JLabel telefoneLabel = new JLabel ("Telefone");
telefoneLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx=0;
gbc.gridy=6;
gbc.insets= new Insets(10,0,0,5);
gbc.gridwidth = GridBagConstraints.RELATIVE;
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(telefoneLabel, gbc);
telefoneField = new JTextField(30);
String telefone =
String.valueOf(fornecedores.get(0).getTelefone());
telefoneField.setText(telefone);
gbc.gridx=1;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(telefoneField, gbc);
JLabel celularLabel = new JLabel ("Celular");
celularLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx=0;
gbc.gridy=7;
gbc.insets= new Insets(10,0,0,5);
gbc.gridwidth = GridBagConstraints.RELATIVE;
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(celularLabel, gbc);
celularField = new JTextField(30);
String celular =
String.valueOf(fornecedores.get(0).getCelular());
gbc.gridx=1;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(celularField, gbc);
JButton botaoSalvar = new JButton("Salvar");
botaoSalvar.addActionListener(this);
botaoSalvar.setActionCommand("salvar");
JButton botaoSair = new JButton("Sair");
botaoSair.addActionListener(this);

```

```

        botaoSair.setActionCommand("sair");
        gbc.gridx=1;
        gbc.gridy=12;
        gbc.fill = GridBagConstraints.NONE;
        gbc.insets=new Insets(10,0,0,5);
        gbc.anchor = GridBagConstraints.LINE_START;
        add(botaoSalvar, gbc);
        gbc.gridx=1;
        gbc.gridy=12;
        gbc.fill = GridBagConstraints.NONE;
        gbc.insets=new Insets(10,135,0,5);
        gbc.anchor = GridBagConstraints.LINE_START;
        add(botaoSair, gbc);
    }
    public void sair() {
        this.dispose();
        this.setVisible(false);
    }

    public void salvar() {
        Fornecedor fornecedor = new Fornecedor();

        fornecedor.setNmFornecedor(nmFornecedorField.getText());
        fornecedor.setNmFantasia(nmFantasiaField.getText());
        fornecedor.setRzSocial(rzSocialField.getText());
        fornecedor.setCnpj(Integer.parseInt(cnpjField.getText()));
        fornecedor.setEmail(emailField.getText());

        fornecedor.setTelefone(Integer.parseInt(telefoneField.getText()))
;

        fornecedor.setCelular(Integer.parseInt(celularField.getText()));
        FornecedorController fornecedorController = new
FornecedorController();
        try {
            if(fornecedorController.atualizar(fornecedor, id)) {
                JOptionPane.showMessageDialog(null,
"Cadastro do fornecedor realizado com sucesso");
                this.dispose();
                this.setVisible(false);
            }
            else {
                JOptionPane.showMessageDialog(null,
"Problema ao atualizar fornecedor!");
            }
        }
    }
}

```

```

    }
    } catch (HeadlessException | SQLException e) {
        e.printStackTrace();
    }
}

@Override
public void actionPerformed(ActionEvent e) {
    if(e.getActionCommand().equals("sair")) {
        sair();
    }
    else if(e.getActionCommand().equals("salvar")) {
        salvar();
    }
}
}
}

```

//view> AtualizarLivroView: classe responsável pela apresentação visual da tela de atualização do Livro;

```

package view;

import java.awt.Font;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.HeadlessException;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.SwingConstants;

```

```

import javax.swing.event.TableModelListener;
import javax.swing.table.DefaultTableModel;
import controller.LivroController;
import dao.FornecedorDAO;
import dao.LivroDAO;
import model.Fornecedor;
import model.FornecedorTableModel;
import model.Livro;

public class AtualizarLivroView extends JFrame implements
ActionListener {
    private JTextField nomeLivroField;
    private JTextField nomeAutorField;
    private JTextField editoraField;
    private JTextField generoField;
    private JTextField anoField;
    private JTextField edicaoField;
    private JTextField precoVendaField;
    private JTextField cdFornecedorField;
    private JTextField psEstoqueField;
    private JTextField qntLivroField;
    private JTextField imgLinkField;
    JFrame frameList;
    private Livro livro;
    ImagemIcon imagem1 = new
ImagemIcon(getClass().getResource("teste.png"));
    JLabel imagem = new JLabel(imagem1);
    FornecedorTableModel tableModel = new
FornecedorTableModel();
    JTable jtFornecedor;
    private int id;
    ArrayList<Livro> livros;

    public AtualizarLivroView(int id) {
        this.setSize(800,500);
        this.setResizable(false);
        this.setVisible(true);
        this.id=id;
        construir();
    }

    public void construir() {
        LivroDAO ld = new LivroDAO();
        ld.buscar(id);
    }

```

```

        livros = (ArrayList<Livro>) Id.buscar(id);
        setLayout(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.weightx=0.1;
        gbc.weighty=0.1;
        JLabel titulo = new JLabel("Cadastro de Livros");
        titulo.setFont(new Font("Arial",
Font.CENTER_BASELINE, 40));
        gbc.gridx=0;
        gbc.gridy=0;
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        gbc.fill = GridBagConstraints.LINE_START;
        gbc.anchor = GridBagConstraints.NORTH;
        add(titulo,gbc);
        JLabel cdFornecedorLabel = new JLabel ("CÃ³digo do
Fornecedor");
        cdFornecedorLabel.setFont(new Font("Arial", Font.BOLD,
16));

        gbc.gridx=0;
        gbc.gridy=1;
        gbc.insets= new Insets(10,0,0,5);
        gbc.gridwidth = GridBagConstraints.RELATIVE;
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.LINE_END;
        add(cdFornecedorLabel, gbc);
        cdFornecedorField = new JTextField(30);
        cdFornecedorField.setEditable(false);
        String cdFornecedor =
String.valueOf(livros.get(0).getCdFornecedor());
        cdFornecedorField.setText(cdFornecedor);
        cdFornecedorField.setEditable(false);
        gbc.gridx=1;
        gbc.insets= new Insets(10,0,0,5);
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.LINE_START;
        add(cdFornecedorField, gbc);
        JLabel nomeLivroLabel = new JLabel("Nome ");
        nomeLivroLabel.setFont(new Font("Arial", Font.BOLD,
16));

        gbc.gridx=0;
        gbc.gridy=2;
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor=GridBagConstraints.LINE_END;
        gbc.insets= new Insets(5,0,0,5);

```

```

add(nomeLivroLabel, gbc);
nomeLivroField = new JTextField(30);
nomeLivroField.setText(livros.get(0).getNomeLivro());
gbc.gridx=1;
gbc.gridy=2;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(nomeLivroField, gbc);
JLabel nomeAutorLabel = new JLabel("Autor ");
nomeAutorLabel.setFont(new Font("Arial", Font.BOLD,
16));

gbc.gridx=0;
gbc.gridy=3;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(nomeAutorLabel, gbc);
nomeAutorField = new JTextField(30);
nomeAutorField.setText(livros.get(0).getAutorLivro());
gbc.gridx=1;
gbc.gridy=3;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(nomeAutorField, gbc);
JLabel editoraLabel = new JLabel ("Editora");
editoraLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx=0;
gbc.gridy=4;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(editoraLabel, gbc);
editoraField = new JTextField(30);
editoraField.setText(livros.get(0).getEditoraLivro());
gbc.gridx=1;
gbc.gridy=4;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(editoraField, gbc);
JLabel generoLabel = new JLabel ("Gênero");
generoLabel.setFont(new Font("Arial", Font.BOLD, 16));

```

```

gbc.gridx=0;
gbc.gridy=5;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(generoLabel, gbc);
generoField = new JTextField(30);
generoField.setText(livros.get(0).getGeneroLivro());
gbc.gridx=1;
gbc.gridy=5;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(generoField, gbc);
JLabel anoLabel = new JLabel ("Ano de Lançamento");
anoLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx=0;
gbc.gridy=6;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(anoLabel, gbc);
anoField = new JTextField(30);
String ano = String.valueOf(livros.get(0).getAnoLivro());
anoField.setText(ano);
gbc.gridx=1;
gbc.gridy=6;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(anoField, gbc);
JLabel edicaoLabel = new JLabel ("Edição");
edicaoLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx=0;
gbc.gridy=7;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(edicaoLabel, gbc);
edicaoField = new JTextField(30);
String edicao =
String.valueOf(livros.get(0).getEdicaoLivro());
edicaoField.setText(edicao);
gbc.gridx=1;

```

```

gbc.gridy=7;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(edicaoField, gbc);
JLabel precoVendaLabel = new JLabel ("Preço de
Venda");
precoVendaLabel.setFont(new Font("Arial", Font.BOLD,
16));

gbc.gridx=0;
gbc.gridy=8;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(precoVendaLabel, gbc);
precoVendaField = new JTextField(30);
String preco =
String.valueOf(livros.get(0).getPrecoVenda());
precoVendaField.setText(preco);
gbc.gridx=1;
gbc.gridy=8;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(precoVendaField, gbc);
JLabel qntLivroLabel = new JLabel ("Quantidade");
qntLivroLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx=0;
gbc.gridy=9;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(qntLivroLabel, gbc);
qntLivroField = new JTextField(30);
String qnt = String.valueOf(livros.get(0).getQntLivro());
qntLivroField.setText(qnt);
gbc.gridx=1;
gbc.gridy=9;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(qntLivroField, gbc);
JLabel imgLinkLabel = new JLabel ("Link Imagem");
imgLinkLabel.setFont(new Font("Arial", Font.BOLD, 16));

```



```

        gbc.gridx=0;
        gbc.gridy=10;
        gbc.insets= new Insets(10,0,0,5);
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.LINE_END;
        add(imgLinkLabel, gbc);
        imgLinkField = new JTextField(30);
        imgLinkField.setText(livros.get(0).getLinkImg());
        gbc.gridx=1;
        gbc.gridy=10;
        gbc.insets= new Insets(10,0,0,5);
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.LINE_START;
        add(imgLinkField, gbc);
        JButton botaoSalvar = new JButton("Salvar");
        botaoSalvar.addActionListener(this);
        botaoSalvar.setActionCommand("salvar");
        JButton botaoSair = new JButton("Sair");
        botaoSair.addActionListener(this);
        botaoSair.setActionCommand("sair");
        gbc.gridx=1;
        gbc.gridy=12;
        gbc.fill = GridBagConstraints.NONE;
        gbc.insets=new Insets(10,0,0,5);
        gbc.anchor = GridBagConstraints.LINE_START;
        add(botaoSalvar, gbc);
        gbc.gridx=1;
        gbc.gridy=12;
        gbc.fill = GridBagConstraints.NONE;
        gbc.insets=new Insets(10,135,0,5);
        gbc.anchor = GridBagConstraints.LINE_START;
        add(botaoSair, gbc);
    }

    public void sair() {
        this.dispose();
        this.setVisible(false);
    }

    public void salvar() {
        Livro livro = new Livro();
        livro.setNomeLivro(nomeLivroField.getText());
        livro.setAutorLivro(nomeAutorField.getText());
        livro.setEditoraLivro(editoraField.getText());
    }

```

```

        livro.setGeneroLivro(generoField.getText());
        livro.setAnoLivro(Integer.parseInt(anoField.getText()));

        livro.setEdicaoLivro(Integer.parseInt(edicaoField.getText()));

        livro.setPrecoVenda(Double.parseDouble(precoVendaField.getText()));

        livro.setCdFornecedor(Integer.parseInt(cdFornecedorField.getText()));

        livro.setQntLivro(Integer.parseInt(qntLivroField.getText()));
        livro.setLinkImg(imgLinkField.getText());
        LivroController livroController = new LivroController();
        try {
            if(livroController.atualizar(livro,id)) {
                JOptionPane.showMessageDialog(null,
"Cadastro de livro realizado com sucesso");
                this.dispose();
                this.setVisible(false);
                ListarLivroView llv = new ListarLivroView();
                llv.revalidate();
                llv.repaint();
            }
            else {
                JOptionPane.showMessageDialog(null,
"Problema ao realizar cadastro de livro!");
            }
        } catch (HeadlessException | SQLException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getActionCommand().equals("sair")) {
            sair();
        }
        else if(e.getActionCommand().equals("salvar")) {
            salvar();
        }
    }

    private void
tableModelMouseClicked(java.awt.event.MouseEvent evt) {
        int i = jtFornecedor.getSelectedRow();

```

```

        TableModelListener[] model =
tableModel.getTableModelListeners();
        cdFornecedorField.setText(jtFornecedor.getValueAt(i,
0).toString());
    }
}

//view> CadastroFornecedorView: classe responsável pela apresentação
visual da tela de cadastro do fornecedor;
package view;

import java.awt.Font;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.HeadlessException;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import controller.FornecedorController;
import model.Fornecedor;

public class CadastroFornecedorView extends JPanel{
    private JTextField nmFornecedorField;
    private JTextField nmFantasiaField;
    private JTextField rzSocialField;
    private JTextField cnpjField;
    private JTextField emailField;
    private JTextField telefoneField;
    private JTextField celularField;
    private JTextField cdFornecedorField;
    private Fornecedor fornecedor;

    public CadastroFornecedorView() {
        criarFormulario();
    }

    private void criarFormulario() {
        setLayout(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();

```

```

        gbc.weightx=0.1;
        gbc.weighty=0.1;
        JLabel titulo = new JLabel("Cadastro de Fornecedor");
        titulo.setFont(new Font("Arial",
Font.CENTER_BASELINE, 40));
        gbc.gridx=0;
        gbc.gridy=0;
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        gbc.fill = GridBagConstraints.LINE_START;
        gbc.anchor = GridBagConstraints.NORTH;
        add(titulo,gbc);
        JLabel nmFornecedorLabel = new JLabel ("Nome do
Fornecedor");
        nmFornecedorLabel.setFont(new Font("Arial",
Font.BOLD, 16));
        gbc.gridx=0;
        gbc.gridy=1;
        gbc.insets= new Insets(10,0,0,5);
        gbc.gridwidth = GridBagConstraints.RELATIVE;
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.LINE_END;
        add(nmFornecedorLabel, gbc);
        nmFornecedorField = new JTextField(30);
        gbc.gridx=1;
        gbc.insets= new Insets(10,0,0,5);
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.LINE_START;
        add(nmFornecedorField, gbc);
        JLabel nmFantasiaLabel = new JLabel ("Nome
Fantasia");
        nmFantasiaLabel.setFont(new Font("Arial", Font.BOLD,
16));
        gbc.gridx=0;
        gbc.gridy=2;
        gbc.insets= new Insets(10,0,0,5);
        gbc.gridwidth = GridBagConstraints.RELATIVE;
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.LINE_END;
        add(nmFantasiaLabel, gbc);
        nmFantasiaField = new JTextField(30);
        gbc.gridx=1;
        gbc.insets= new Insets(10,0,0,5);
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.LINE_START;

```

```

add(nmFantasiaField, gbc);
JLabel rzSocialLabel = new JLabel ("razão Social");
rzSocialLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx=0;
gbc.gridy=3;
gbc.insets= new Insets(10,0,0,5);
gbc.gridwidth = GridBagConstraints.RELATIVE;
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(rzSocialLabel, gbc);
rzSocialField = new JTextField(30);
gbc.gridx=1;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(rzSocialField, gbc);
JLabel cnpjLabel = new JLabel ("CNPJ");
cnpjLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx=0;
gbc.gridy=4;
gbc.insets= new Insets(10,0,0,5);
gbc.gridwidth = GridBagConstraints.RELATIVE;
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(cnpjLabel, gbc);
cnpjField = new JTextField(30);
gbc.gridx=1;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(cnpjField, gbc);
JLabel emailLabel = new JLabel ("Email");
emailLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx=0;
gbc.gridy=5;
gbc.insets= new Insets(10,0,0,5);
gbc.gridwidth = GridBagConstraints.RELATIVE;
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(emailLabel, gbc);
emailField = new JTextField(30);
gbc.gridx=1;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;

```

```

gbc.anchor = GridBagConstraints.LINE_START;
add(emailField, gbc);
JLabel telefoneLabel = new JLabel ("Telefone");
telefoneLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx=0;
gbc.gridy=6;
gbc.insets= new Insets(10,0,0,5);
gbc.gridwidth = GridBagConstraints.RELATIVE;
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(telefoneLabel, gbc);
telefoneField = new JTextField(30);
gbc.gridx=1;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(telefoneField, gbc);
JLabel celularLabel = new JLabel ("Celular");
celularLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx=0;
gbc.gridy=7;
gbc.insets= new Insets(10,0,0,5);
gbc.gridwidth = GridBagConstraints.RELATIVE;
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(celularLabel, gbc);
celularField = new JTextField(30);
gbc.gridx=1;
gbc.insets= new Insets(10,0,0,5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(celularField, gbc);
ActionSalvar actionSalvar = new ActionSalvar();
ActionSair actionSair = new ActionSair();
JButton botaoSalvar = new JButton("Salvar");
botaoSalvar.addActionListener(actionSalvar);
JButton botaoSair = new JButton("Sair");
botaoSair.addActionListener(actionSair);
gbc.gridx=1;
gbc.gridy=12;
gbc.fill = GridBagConstraints.NONE;
gbc.insets=new Insets(10,0,0,5);
gbc.anchor = GridBagConstraints.LINE_START;
add(botaoSalvar, gbc);

```

```

        gbc.gridx=1;
        gbc.gridy=12;
        gbc.fill = GridBagConstraints.NONE;
        gbc.insets=new Insets(10,135,0,5);
        gbc.anchor = GridBagConstraints.LINE_START;
        add(botaoSair, gbc);
    }

    private class ActionSalvar implements ActionListener{

        public void actionPerformed(ActionEvent Event) {
            System.out.println(" Cliquei no botão salvar
{view}");

            Fornecedor fornecedor = new Fornecedor();

            fornecedor.setNmFornecedor(nmFornecedorField.getText());

            fornecedor.setNmFantasia(nmFantasiaField.getText());
            fornecedor.setRzSocial(rzSocialField.getText());

            fornecedor.setCnpj(Integer.parseInt(cnpjField.getText()));
            fornecedor.setEmail(emailField.getText());

            fornecedor.setTelefone(Integer.parseInt(telefoneField.getText()));

            fornecedor.setCelular(Integer.parseInt(celularField.getText()));
            FornecedorController fornecedorController = new
FornecedorController();
            try {

                if(fornecedorController.cadastro(fornecedor)) {

                    JOptionPane.showMessageDialog(null, "Cadastro do fornecedor
realizado com sucesso");

                    nmFornecedorField.setText(" ");
                    nmFantasiaField.setText(" ");
                    rzSocialField.setText(" ");
                    cnpjField.setText(" ");
                    emailField.setText(" ");
                    telefoneField.setText(" ");
                    celularField.setText(" ");
                    cdFornecedorField.setText(" ");
                }
            } else {

```

```
JOptionPane.showMessageDialog(null, "Problema ao realizar cadastro  
do fornecedor!");
```

```
    }  
    } catch (HeadlessException e) {  
        e.printStackTrace();  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
    }  
}  
private class ActionSair implements ActionListener{  
  
    public void actionPerformed(ActionEvent arg0) {  
        System.exit(0);  
    }  
}  
}
```

//view> CadastroLivroView: classe responsável pela apresentação visual da
tela de cadastro do Livro;

```
package view;  
  
import java.awt.Color;  
import java.awt.Font;  
import java.awt.GridBagConstraints;  
import java.awt.GridBagLayout;  
import java.awt.GridLayout;  
import java.awt.HeadlessException;  
import java.awt.Insets;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.awt.event.MouseAdapter;  
import java.awt.event.MouseEvent;  
import java.sql.SQLException;  
import java.util.ArrayList;  
import java.util.List;  
  
import javax.swing.ImageIcon;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;
```



```

import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.SwingConstants;
import javax.swing.event.TableModelListener;
import javax.swing.table.DefaultTableModel;

import controller.LivroController;
import dao.FornecedorDAO;
import model.Fornecedor;
import model.FornecedorTableModel;
import model.Livro;

public class CadastroLivroView extends JPanel {
    private JTextField nomeLivroField;
    private JTextField nomeAutorField;
    private JTextField editoraField;
    private JTextField generoField;
    private JTextField anoField;
    private JTextField edicaoField;
    private JTextField precoVendaField;
    private JTextField cdFornecedorField;
    private JTextField psEstoqueField;
    private JTextField qntLivroField;
    private JTextField imgLinkField;
    JFrame frameList;
    private Livro livro;
    ImageIcon imagem1 = new
ImagemIcon(getClass().getResource("teste.png"));
    JLabel imagem = new JLabel(imagem1);
    FornecedorTableModel tableModel = new
FornecedorTableModel();

    public CadastroLivroView() {
        criarFormulario();
    }

    private void criarFormulario() {
        setLayout(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.weightx = 0.1;
        gbc.weighty = 0.1;
        JLabel titulo = new JLabel("Cadastro de Livros");

```

```

        titulo.setFont(new Font("Arial",
Font.CENTER_BASELINE, 40));
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.gridwidth = GridBagConstraints.REMAINDER;
        gbc.fill = GridBagConstraints.LINE_START;
        gbc.anchor = GridBagConstraints.NORTH;
        add(titulo, gbc);
        JLabel cdFornecedorLabel = new JLabel("CÃ³digo do
Fornecedor");
        cdFornecedorLabel.setFont(new Font("Arial", Font.BOLD,
16));

        gbc.gridx = 0;
        gbc.gridy = 1;
        gbc.insets = new Insets(10, 0, 0, 5);
        gbc.gridwidth = GridBagConstraints.RELATIVE;
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.LINE_END;
        add(cdFornecedorLabel, gbc);
        cdFornecedorField = new JTextField(30);
        cdFornecedorField.setEditable(false);
        gbc.gridx = 1;
        gbc.insets = new Insets(10, 0, 0, 5);
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.LINE_START;
        add(cdFornecedorField, gbc);
        ActionListener actionListar = new ActionListener();
        JButton botaoFornecedor = new JButton("Selecionar");
        botaoFornecedor.addActionListener(actionListar);
        gbc.gridx = 1;
        gbc.gridy = 1;
        gbc.fill = GridBagConstraints.RELATIVE;
        gbc.insets = new Insets(10, 0, 0, 45);
        gbc.anchor = GridBagConstraints.LINE_END;
        add(botaoFornecedor, gbc);
        JLabel nomeLivroLabel = new JLabel("Nome ");
        nomeLivroLabel.setFont(new Font("Arial", Font.BOLD,
16));

        gbc.gridx = 0;
        gbc.gridy = 2;
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.LINE_END;
        gbc.insets = new Insets(5, 0, 0, 5);
        add(nomeLivroLabel, gbc);

```

```

nomeLivroField = new JTextField(30);
gbc.gridx = 1;
gbc.gridy = 2;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(nomeLivroField, gbc);
JLabel nomeAutorLabel = new JLabel("Autor ");
nomeAutorLabel.setFont(new Font("Arial", Font.BOLD,
16));

gbc.gridx = 0;
gbc.gridy = 3;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(nomeAutorLabel, gbc);
nomeAutorField = new JTextField(30);
gbc.gridx = 1;
gbc.gridy = 3;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(nomeAutorField, gbc);
JLabel editoraLabel = new JLabel("Editora");
editoraLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx = 0;
gbc.gridy = 4;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(editoraLabel, gbc);
editoraField = new JTextField(30);
gbc.gridx = 1;
gbc.gridy = 4;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(editoraField, gbc);
JLabel generoLabel = new JLabel("Gênero");
generoLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx = 0;
gbc.gridy = 5;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;

```

```

gbc.anchor = GridBagConstraints.LINE_END;
add(generoLabel, gbc);
generoField = new JTextField(30);
gbc.gridx = 1;
gbc.gridy = 5;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(generoField, gbc);
JLabel anoLabel = new JLabel("Ano de Lançamento");
anoLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx = 0;
gbc.gridy = 6;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(anoLabel, gbc);
anoField = new JTextField(30);
gbc.gridx = 1;
gbc.gridy = 6;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(anoField, gbc);
JLabel edicaoLabel = new JLabel("Edição");
edicaoLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx = 0;
gbc.gridy = 7;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(edicaoLabel, gbc);
edicaoField = new JTextField(30);
gbc.gridx = 1;
gbc.gridy = 7;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(edicaoField, gbc);
JLabel precoVendaLabel = new JLabel("Preço de
Venda");
precoVendaLabel.setFont(new Font("Arial", Font.BOLD,
16));
gbc.gridx = 0;

```

```

gbc.gridy = 8;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(precoVendaLabel, gbc);
precoVendaField = new JTextField(30);
gbc.gridx = 1;
gbc.gridy = 8;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(precoVendaField, gbc);
JLabel qntLivroLabel = new JLabel("Quantidade");
qntLivroLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx = 0;
gbc.gridy = 9;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(qntLivroLabel, gbc);
qntLivroField = new JTextField(30);
gbc.gridx = 1;
gbc.gridy = 9;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(qntLivroField, gbc);
JLabel imgLinkLabel = new JLabel("Link Imagem");
imgLinkLabel.setFont(new Font("Arial", Font.BOLD, 16));
gbc.gridx = 0;
gbc.gridy = 10;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_END;
add(imgLinkLabel, gbc);
imgLinkField = new JTextField(30);
gbc.gridx = 1;
gbc.gridy = 10;
gbc.insets = new Insets(10, 0, 0, 5);
gbc.fill = GridBagConstraints.NONE;
gbc.anchor = GridBagConstraints.LINE_START;
add(imgLinkField, gbc);
ActionSalvar actionSalvar = new ActionSalvar();
ActionSair actionSair = new ActionSair();

```

```

        JButton botaoSalvar = new JButton("Salvar");
        botaoSalvar.addActionListener(actionSalvar);
        JButton botaoSair = new JButton("Sair");
        botaoSair.addActionListener(actionSair);
        gbc.gridx = 1;
        gbc.gridy = 12;
        gbc.fill = GridBagConstraints.NONE;
        gbc.insets = new Insets(10, 0, 0, 5);
        gbc.anchor = GridBagConstraints.LINE_START;
        add(botaoSalvar, gbc);
        gbc.gridx = 1;
        gbc.gridy = 12;
        gbc.fill = GridBagConstraints.NONE;
        gbc.insets = new Insets(10, 135, 0, 5);
        gbc.anchor = GridBagConstraints.LINE_START;
        add(botaoSair, gbc);
    }

    private class ActionSalvar implements ActionListener {

        public void actionPerformed(ActionEvent Event) {
            Livro livro = new Livro();
            livro.setNomeLivro(nomeLivroField.getText());
            livro.setAutorLivro(nomeAutorField.getText());
            livro.setEditoraLivro(editoraField.getText());
            livro.setGeneroLivro(generoField.getText());

            livro.setAnoLivro(Integer.parseInt(anoField.getText()));

            livro.setEdicaoLivro(Integer.parseInt(edicaoField.getText()));

            livro.setPrecoVenda(Double.parseDouble(precoVendaField.getText()));

            livro.setCdFornecedor(Integer.parseInt(cdFornecedorField.getText()));

            livro.setQntLivro(Integer.parseInt(qntLivroField.getText()));
            livro.setLinkImg(imgLinkField.getText());
            LivroController livroController = new
LivroController();
            try {
                if (livroController.cadastro(livro)) {

                    JOptionPane.showMessageDialog(null, "Cadastro de livro realizado
com sucesso");

```

```

nomeLivroField.setText(" ");
nomeAutorField.setText(" ");
editoraField.setText(" ");
generoField.setText(" ");
anoField.setText(" ");
edicaoField.setText(" ");
precoVendaField.setText(" ");
cdFornecedorField.setText(" ");
    } else {

JOptionPane.showMessageDialog(null, "Problema ao realizar cadastro
de livro!");

    }
    } catch (HeadlessException | SQLException e) {
        e.printStackTrace();
    }
}

private class ActionSair implements ActionListener {

    public void actionPerformed(ActionEvent arg0) {
        System.exit(0);
    }
}

private class ActionListar implements ActionListener {
    int rows;
    int headers;
    JTable jtFornecedor = new JTable(new
DefaultTableModel(rows, headers));

    @Override
    public void actionPerformed(ActionEvent e) {
        jtFornecedor.setModel(tableModel);
        frameList = new JFrame();
        frameList.setVisible(true);
        frameList.setSize(300, 450);
        frameList.setResizable(false);

frameList.setLocationRelativeTo(cdFornecedorField);
        frameList.setLayout(new GridLayout(3, 1));
        JPanel panelTitulo = new JPanel();
        panelTitulo.setLayout(new GridLayout(1, 1));

```

```

        JLabel titulo = new JLabel("Selezione o
Fornecedor", SwingConstants.CENTER);
        titulo.setFont(new Font("Arial", Font.BOLD, 18));
        panelTitulo.add(titulo);
        frameList.add(panelTitulo);
        List<Fornecedor> fornecedores = new
ArrayList<Fornecedor>();
        FornecedorDAO fd = new FornecedorDAO();
        fornecedores = (List<Fornecedor>) fd.Listar();
        tableModel.removeAll();
        for (int i = 0; i < fornecedores.size(); i++) {
            tableModel.addRow(fornecedores.get(i));
        }
        jtFornecedor.addMouseListener(new
MouseAdapter() {
            @Override
            public void mousePressed(MouseEvent e) {
                int row =
jtFornecedor.rowAtPoint(e.getPoint());
                row =
jtFornecedor.getSelectedRow();

                jtFornecedor.getSelectionModel().setSelectionInterval(row, row);
                cdFornecedorField.setText("" +
tableModel.getSelectRow(row));

                cdFornecedorField.setEditable(false);
                if (e.getButton() ==
MouseEvent.BUTTON3) {
                    }
                }
            });
        JScrollPane scrollPane = new
JScrollPane(jtFornecedor);
        frameList.add(scrollPane);
        JPanel panelInferior = new JPanel();
        panelInferior.setLayout(new GridBagLayout());
        GridBagConstraints gbc = new
GridBagConstraints();
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.insets = new Insets(10, 0, 10, 10);
        ActionConcluir actionConcluir = new
ActionConcluir();
        JButton jbConcluir = new JButton("Concluir");

```



```

        jbConcluir.addActionListener(actionConcluir);
        ActionCancelar actionCancelar = new
ActionCancelar();

        JButton jbCancelar = new JButton("Cancelar");
        jbCancelar.addActionListener(actionCancelar);
        panelInferior.add(jbConcluir, gbc);
        gbc.gridx = 1;
        panelInferior.add(jbCancelar, gbc);
        frameList.add(panelInferior);
        MouseEvent evt = null;
        tableModelMouseClicked(evt);
    }

    public void
jTFornecedoresMouseClicked(java.awt.event.MouseEvent evt) {
        if (jtFornecedor.getSelectedRow() != -1) {

            cdFornecedorField.setText(jtFornecedor.getValueAt(jtFornecedor.getSe
electedRow(), 0).toString());
        }
    }

    private void
tableModelMouseClicked(java.awt.event.MouseEvent evt) {
        int i = jtFornecedor.getSelectedRow();
        TableModelListener[] model =
tableModel.getTableModelListeners();

        cdFornecedorField.setText(jtFornecedor.getValueAt(i, 0).toString());
    }

    private class ActionConcluir implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent e) {
            frameList.setVisible(false);
            frameList.dispose();
        }
    }

    private class ActionCancelar implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent e) {

```

```

cdFornecedorField.setText(" ");
frameList.setVisible(false);
frameList.dispose();
    }
}
}
}

```

//view> FinalizarCompraView: classe responsável pela apresentação visual da tela de finalização de compra;

```

package view;

import java.awt.BorderLayout;
import java.awt.Font;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.HeadlessException;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Date;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.SwingConstants;
import controller.CompraController;
import dao.CompraDAO;
import model.Compra;

public class FinalizarCompraView extends JPanel implements
ActionListener {
    private JFrame frame;
    private JTextField quantidadeField;
    private JTextField precoField;
    private Compra compra;
    int n;

```

```

public FinalizarCompraView(Compra c) {
    this.compra = c;
    criarFormulario();
}

public void criarFormulario() {
    this.setLayout(new BorderLayout());
    JLabel espacoLabel = new JLabel("");
    JLabel espacoLabel2 = new JLabel("");
    JLabel tituloLabel = new JLabel("Compra",
SwingConstants.CENTER);
    tituloLabel.setFont(new Font("Arial", Font.BOLD, 40));
    JLabel livroLabel = new JLabel("Livro Selecionado");
    livroLabel.setFont(new Font("Arial", Font.BOLD, 16));
    JTextField livroField = new JTextField(20);
    livroField.setEditable(false);
    String codLivro = GerarCompraView.codLivro;
    livroField.setText(codLivro);
    JLabel quantidadeLabel = new JLabel("Quantidade");
    quantidadeLabel.setFont(new Font("Arial", Font.BOLD,
16));

    quantidadeField = new JTextField(20);
    JLabel precoLabel = new JLabel("Preço Unitário");
    precoLabel.setFont(new Font("Arial", Font.BOLD, 16));
    precoField = new JTextField(20);
    JButton concluir = new JButton("Concluir");
    concluir.addActionListener(this);
    concluir.setActionCommand("concluir");
    JButton cancelar = new JButton("Cancelar");
    cancelar.addActionListener(this);
    cancelar.setActionCommand("cancelar");
    JPanel panelCentral = new JPanel();
    panelCentral.setLayout(new GridBagLayout());
    GridBagConstraints gbc = new GridBagConstraints();
    gbc.weightx = 0.1;
    gbc.weighty = 0.1;
    gbc.insets = new Insets(10, 0, 0, 5);
    this.add(tituloLabel, BorderLayout.NORTH);
    gbc.gridx = 0;
    gbc.gridy = 0;
    panelCentral.add(espacoLabel, gbc);
    gbc.gridx = 1;
    panelCentral.add(livroLabel, gbc);
    gbc.gridx = 2;

```

```

        panelCentral.add(livroField, gbc);
        gbc.gridy = 1;
        gbc.gridx = 1;
        panelCentral.add(quantidadeLabel, gbc);
        gbc.gridx = 2;
        panelCentral.add(quantidadeField, gbc);
        gbc.gridy = 2;
        gbc.gridx = 1;
        panelCentral.add(precoLabel, gbc);
        gbc.gridx = 2;
        panelCentral.add(precoField, gbc);
        gbc.gridx = 3;
        panelCentral.add(espacoLabel2, gbc);
        gbc.gridy = 3;
        gbc.gridx = 1;
        panelCentral.add(concluir, gbc);
        gbc.gridx = 2;
        panelCentral.add(cancelar, gbc);
        this.add(panelCentral, BorderLayout.CENTER);
    }

    public void acaoConcluir(){

        compra.setQuantidade(Integer.parseInt(quantidadeField.getText
()));

        compra.setPreco(Double.parseDouble(precoField.getText()));
        int idCompra = 0;
        CompraController compraController = new
CompraController();
        try {
            idCompra = compraController.cadastro(compra);
        } catch (HeadlessException e) {
        }
        gerarNota(idCompra);
    }

    public void gerarNota(int idCompra) {
        CompraDAO cd = new CompraDAO();
        n++;
        Date data = new Date(System.currentTimeMillis());
        String arq = ""+n+"NotaFiscal-"+data+".txt";
        Compra texto = compra;
        if(cd.gerarNota(arq, texto, idCompra)) {

```

```

        JOptionPane.showMessageDialog(null, "Nota fiscal
gerada com sucesso");
    }
    else {
        JOptionPane.showMessageDialog(null, "Falha ao
gerar nota fiscal");
    }
}

public void acaoCancelar() {
    JOptionPane.showMessageDialog(null, "Imprementar
acao para voltar ao GerarCompraView");
}

public void actionPerformed(ActionEvent e) {
    if (e.getActionCommand().equals("concluir")) {
        acaoConcluir();
    } else if (e.getActionCommand().equals("cancelar")) {
        acaoCancelar();
    }
}
}
}

```

//view> GerarCompraView: classe responsável pela apresentação visual da tela de gerar compra;

```

package view;

import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.text.DateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import javax.swing.AbstractButton;

```

```

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.SwingConstants;
import javax.swing.event.TableModelListener;
import javax.swing.table.DefaultTableModel;
import com.toedter.calendar.JDateChooser;
import dao.FornecedorDAO;
import dao.LivroDAO;
import model.Compra;
import model.Fornecedor;
import model.FornecedorTableModel;
import model.Livro;
import model.LivroTableModel;

public class GerarCompraView extends JPanel implements
ActionListener {
    private JTextField dtCompraField;
    private JTextField dtEntregaField;
    private JTextField cdFornecedorField;
    private JTextField cdLivroField;
    private JFrame frame;
    public static String codLivro;
    public static String codFornecedor;
    public static String dtCompra;
    public static String dtEntrega;
    JDateChooser dataCompra;
    JDateChooser dataEntrega;
    JFrame frameList = new JFrame();
    LivroTableModel tableModelLivro = new LivroTableModel();
    FornecedorTableModel tableModelFornecedor = new
FornecedorTableModel();
    int rows;
    int headers;
    private Compra compra;
    JTable jtFornecedor = new JTable(new DefaultTableModel(rows,
headers));
    JTable jtLivro = new JTable(new DefaultTableModel(rows,
headers));

```

```

public GerarCompraView(JFrame principal) {
    this.frame = principal;
    criarFormulario();
}

public void criarFormulario() {
    this.setLayout(new BorderLayout());
    JLabel titulo = new JLabel("Compra",
SwingConstants.CENTER);
    titulo.setFont(new Font("Arial",
Font.CENTER_BASELINE, 40));
    this.add(titulo, BorderLayout.NORTH);
    JPanel panelCentral = new JPanel();
    panelCentral.setLayout(new GridBagLayout());
    GridBagConstraints gbc = new GridBagConstraints();
    gbc.weightx = 0.1;
    gbc.weighty = 0.1;
    gbc.insets = new Insets(10, 0, 0, 5);
    gbc.anchor = GridBagConstraints.LINE_END;
    JLabel espaco1 = new JLabel("");
    espaco1.setFont(new Font("Arial", Font.BOLD, 16));
    gbc.gridx = 0;
    gbc.gridy = 0;
    panelCentral.add(espaco1, gbc);
    JLabel cdFornecedor = new JLabel("Código do
Fornecedor");
    cdFornecedor.setFont(new Font("Arial", Font.BOLD, 16));
    gbc.gridx = 1;
    gbc.gridy = 0;
    panelCentral.add(cdFornecedor, gbc);
    cdFornecedorField = new JTextField(20);
    cdFornecedorField.setEditable(false);
    gbc.gridx = 2;
    gbc.anchor = GridBagConstraints.CENTER;
    panelCentral.add(cdFornecedorField, gbc);
    JButton botaoFornecedor = new JButton("Selecionar");
    botaoFornecedor.addActionListener(this);
    botaoFornecedor.setActionCommand("listarFornecedor");
    gbc.gridx = 3;
    gbc.anchor = GridBagConstraints.LINE_START;
    panelCentral.add(botaoFornecedor, gbc);
    JLabel cdLivro = new JLabel("Código do Livro");
    cdLivro.setFont(new Font("Arial", Font.BOLD, 16));
    gbc.gridx = 1;

```

```

gbc.gridy = 1;
gbc.anchor = GridBagConstraints.LINE_END;
panelCentral.add(cdLivro, gbc);
cdLivroField = new JTextField(20);
cdLivroField.setEditable(false);
gbc.gridx = 2;
gbc.anchor = GridBagConstraints.CENTER;
panelCentral.add(cdLivroField, gbc);
JButton botaoLivro = new JButton("Selecionar");
botaoLivro.addActionListener(this);
botaoLivro.setActionCommand("listarLivro");
gbc.gridx = 3;
gbc.anchor = GridBagConstraints.LINE_START;
panelCentral.add(botaoLivro, gbc);
gbc.gridx = 1;
gbc.gridy = 2;
gbc.anchor = GridBagConstraints.LINE_END;
JLabel dtCompraLabel = new JLabel("Data da Compra");
dtCompraLabel.setFont(new Font("Arial", Font.BOLD,
16));
panelCentral.add(dtCompraLabel, gbc);
gbc.gridx = 2;
dataCompra = new JDateChooser("dd/MM/yyyy",
"###/##/####", '_');
dataCompra.setFont(new Font("Arial", Font.BOLD, 16));
dataCompra.setPreferredSize(new Dimension(200, 20));
dataCompra.setDate(new Date());
gbc.anchor = GridBagConstraints.CENTER;
panelCentral.add(dataCompra, gbc);
JLabel dtEntregaLabel = new JLabel("Data de Entrega");
dtEntregaLabel.setFont(new Font("Arial", Font.BOLD,
16));
gbc.gridx = 1;
gbc.gridy = 3;
gbc.anchor = GridBagConstraints.LINE_END;
panelCentral.add(dtEntregaLabel, gbc);
dataEntrega = new JDateChooser("dd/MM/yyyy",
"###/##/####", '_');
dataEntrega.setFont(new Font("Arial", Font.BOLD, 16));
dataEntrega.setPreferredSize(new Dimension(200, 20));
dataEntrega.setDate(new Date());
gbc.gridx = 2;
gbc.anchor = GridBagConstraints.CENTER;
panelCentral.add(dataEntrega, gbc);

```



```

        this.add(panelCentral, BorderLayout.CENTER);
        JButton botaoSalvar = new JButton("Salvar");
        botaoSalvar.addActionListener(this);
        botaoSalvar.setActionCommand("salvar");
        gbc.gridx = 1;
        gbc.gridy = 4;
        gbc.anchor = GridBagConstraints.LINE_END;
        panelCentral.add(botaoSalvar, gbc);
        JButton botaoSair = new JButton("Sair");
        botaoSair.addActionListener(this);
        botaoSair.setActionCommand("sair");
        gbc.gridx = 2;
        gbc.anchor = GridBagConstraints.LINE_END;
        panelCentral.add(botaoSair, gbc);
    }

    public void acaoSair() {
    }

    public void acaoSalvar() {
        codLivre = cdLivreField.getText();
        compra = new Compra();

        compra.getFornecedor().setCodigo(Integer.parseInt(cdFornecedorField.getText()));

        compra.getLivre().setCdLivre(Integer.parseInt(cdLivreField.getText()));

        Date dtCompra = dataCompra.getDate();
        String dataCompra =
        DateFormat.getDateInstance().format(dtCompra);
        compra.setDtCompra(dataCompra);
        Date dtEntrega = dataEntrega.getDate();
        String dataEntrega =
        DateFormat.getDateInstance().format(dtEntrega);
        compra.setDtEntrega(dataEntrega);
        FinalizarCompraView fcv = new
        FinalizarCompraView(compra);
        fcv.criarFormulario();
        fcv.setVisible(true);
        this.frame.getContentPane().removeAll();
        this.frame.getContentPane().add(fcv);
        this.frame.revalidate();
        this.frame.repaint();
    }

```

```

    }

    public void acaoListarFornecedor() {
        jtFornecedor.setModel(tableModelFornecedor);
        frameList = new JFrame();

        frameList.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frameList.setVisible(true);
        frameList.setSize(300, 450);
        frameList.setResizable(false);
        frameList.setLocationRelativeTo(cdFornecedorField);
        frameList.setLayout(new GridLayout(3, 1));
        JPanel panelTitulo = new JPanel();
        panelTitulo.setLayout(new GridLayout(1, 1));
        JLabel titulo = new JLabel("Selecione o Fornecedor",
SwingConstants.CENTER);
        titulo.setFont(new Font("Arial", Font.BOLD, 18));
        panelTitulo.add(titulo);
        frameList.add(panelTitulo);
        List<Fornecedor> fornecedores = new
ArrayList<Fornecedor>();
        FornecedorDAO fd = new FornecedorDAO();
        fornecedores = (List<Fornecedor>) fd.Listar();
        tableModelFornecedor.removeAll();
        for (int i = 0; i < fornecedores.size(); i++) {

            tableModelFornecedor.addRow(fornecedores.get(i));
        }
        jtFornecedor.addMouseListener(new MouseAdapter() {

            @Override
            public void mousePressed(MouseEvent e) {
                int row =
jtFornecedor.rowAtPoint(e.getPoint());
                row = jtFornecedor.getSelectedRow();

                jtFornecedor.getSelectionModel().setSelectionInterval(row, row);
                cdFornecedorField.setText("" +
tableModelFornecedor.getSelectRow(row));
                cdFornecedorField.setEditable(false);
                if (e.getButton() == MouseEvent.BUTTON3)
            {

            }
        }
    }
}

```

```

});
JScrollPane scrollPane = new JScrollPane(jtFornecedor);
frameList.add(scrollPane);
JPanel panelInferior = new JPanel();
panelInferior.setLayout(new GridBagLayout());
GridBagConstraints gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 0;
gbc.insets = new Insets(10, 0, 10, 10);
JButton jbConcluir = new JButton("Concluir");
jbConcluir.addActionListener(this);
jbConcluir.setActionCommand("concluir");
JButton jbCancelar = new JButton("Cancelar");
jbCancelar.addActionListener(this);
jbCancelar.setActionCommand("cancelarFornecedor");
panelInferior.add(jbConcluir, gbc);
gbc.gridx = 1;
panelInferior.add(jbCancelar, gbc);
frameList.add(panelInferior);
MouseEvent evt = null;
tableModelMouseClickedFornecedor(evt);
}

public void acaoConcluir(){
    frameList.setVisible(false);
    frameList.dispose();
}

public void acaoCancelarFornecedor() {
    cdFornecedorField.setText("");
    frameList.setVisible(false);
    frameList.dispose();
}

public void acaoCancelarLivro() {
    cdLivroField.setText("");
    frameList.setVisible(false);
    frameList.dispose();
}

public void acaoListarLivro(){
    jtLivro.setModel(tableModelLivro);
    frameList = new JFrame();
    frameList.setVisible(true);
}

```

```

frameList.setSize(300, 450);
frameList.setResizable(false);
frameList.setLocationRelativeTo(cdLivroField);
frameList.setLayout(new GridLayout(3, 1));
JPanel panelTitulo = new JPanel();
panelTitulo.setLayout(new GridLayout(1, 1));
JLabel titulo = new JLabel("Selecione o Livro",
SwingConstants.CENTER);
titulo.setFont(new Font("Arial", Font.BOLD, 18));
panelTitulo.add(titulo);
frameList.add(panelTitulo);
List<Livro> livros = new ArrayList<Livro>();
LivroDAO ld = new LivroDAO();
livros = (List<Livro>) ld.Listar();
tableModelLivro.removeAll();
for (int i = 0; i < livros.size(); i++) {
    tableModelLivro.addRow(livros.get(i));
}
jtLivro.addMouseListener(new MouseAdapter() {

    @Override
    public void mousePressed(MouseEvent e) {
        int row = jtLivro.rowAtPoint(e.getPoint());
        row = jtLivro.getSelectedRow();

        jtLivro.getSelectionModel().setSelectionInterval(row, row);
        cdLivroField.setText("" +
tableModelLivro.getSelectedRow(row));
        cdLivroField.setEditable(false);
        if (e.getButton() == MouseEvent.BUTTON3)
        {

            }

        });
JScrollPane scrollPane = new JScrollPane(jtLivro);
frameList.add(scrollPane);
JPanel panelInferior = new JPanel();
panelInferior.setLayout(new GridBagLayout());
GridBagConstraints gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 0;
gbc.insets = new Insets(10, 0, 10, 10);
JButton jbConcluir = new JButton("Concluir");
jbConcluir.addActionListener(this);

```

```

        jbConcluir.setActionCommand("concluir");
        JButton jbCancelar = new JButton("Cancelar");
        jbCancelar.addActionListener(this);
        jbCancelar.setActionCommand("cancelarLivro");
        painelInferior.add(jbConcluir, gbc);
        gbc.gridx = 1;
        painelInferior.add(jbCancelar, gbc);
        frameList.add(painelInferior);
        MouseEvent evt = null;
        tableModelMouseClickedLivro(evt);
    }

    private void
    tableModelMouseClickedFornecedor(java.awt.event.MouseEvent evt) {
        int i = jtFornecedor.getSelectedRow();
        TableModelListener[] model =
        tableModelFornecedor.getTableModelListeners();
        cdFornecedorField.setText(jtFornecedor.getValueAt(i,
0).toString());
    }

    private void
    tableModelMouseClickedLivro(java.awt.event.MouseEvent evt) {
        int i = jtLivro.getSelectedRow();
        TableModelListener[] model =
        tableModelLivro.getTableModelListeners();
        cdLivroField.setText(jtLivro.getValueAt(i, 0).toString());
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getActionCommand().equals("salvar")) {
            acaoSalvar();
        }
        else if (e.getActionCommand().equals("listarLivro")) {
            acaoListarLivro();
        }
        else if (e.getActionCommand().equals("listarFornecedor"))
        {
            acaoListarFornecedor();
        }
        else if (e.getActionCommand().equals("sair")) {
            acaoSair();
        }
    }

```

```

        else if (e.getActionCommand().equals("concluir")) {
            acaoConcluir();
        }
        else if
(e.getActionCommand().equals("cancelarFornecedor")) {
            acaoCancelarFornecedor();
        }
        else if (e.getActionCommand().equals("cancelarLivro")) {
            acaoCancelarLivro();
        }
    }
}

```

//view> ListarClienteView: classe responsável pela apresentação visual da tela de listagem de clientes;

```

package view;

import java.awt.BorderLayout;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.sql.Date;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingConstants;
import javax.swing.event.TableModelListener;
import javax.swing.table.DefaultTableModel;
import dao.UsuarioDAO;
import model.Usuario;
import model.UsuarioTableModel;

public class ListarClienteView extends JPanel implements
ActionListener {
    String rows[][] = {};
    String headers[] = {};
    UsuarioTableModel tableModel = new UsuarioTableModel();

```

```

        jTableUsuarios = new.JTable(new DefaultTableModel(rows,
headers));
        int cdCliente;
        List<Usuario> usuarios;
        int n;

        public ListarClienteView() {
            jTableUsuarios.setModel(tableModel);
            criarFormulario();
        }

        public void criarFormulario() {
            this.setLayout(new BorderLayout());
            JLabel titulo = new JLabel("Clientes Cadastrados",
SwingConstants.CENTER);
            titulo.setFont(new Font("Arial", Font.BOLD, 40));
            JPanel panelSuperior = new JPanel();
            JPanel panelCentral = new JPanel();
            JScrollPane scrollPane = new JScrollPane(jTableUsuarios);
            UsuarioDAO ud = new UsuarioDAO();
            usuarios = (ArrayList<Usuario>) ud.Listar();
            for (int i = 0; i < usuarios.size(); i++) {
                tableModel.addRow(usuarios.get(i));
            }
            JButton jbDeletar = new JButton("Deletar");
            jbDeletar.addActionListener(this);
            jbDeletar.setActionCommand("deletar");
            JButton jbGerarArq = new JButton("Gerar Arquivo");
            jbGerarArq.addActionListener(this);
            jbGerarArq.setActionCommand("gerar arquivo");
            panelCentral.add(scrollPane);
            panelSuperior.add(titulo);
            this.add(jbDeletar, BorderLayout.WEST);
            this.add(panelSuperior, BorderLayout.NORTH);
            this.add(panelCentral, BorderLayout.CENTER);
            this.add(jbGerarArq, BorderLayout.EAST);
        }

        public void deletar() {
            MouseEvent evt = null;
            tableModelMouseClicked(evt);
            int cd =
usuarios.get(jTableUsuarios.getSelectedRow()).getCdUsuario();
            UsuarioDAO ud = new UsuarioDAO();

```

```

        ud.deletar(cd);
    }

    public void gerarArq() {
        UsuarioDAO ud = new UsuarioDAO();
        n++;
        Date data = new Date(System.currentTimeMillis());
        String arq = "" + n + "RelatórioDeClientes-" + data + ".txt";
        ArrayList texto = (ArrayList) usuarios;
        if (ud.gerarArq(arq, texto)) {
            JOptionPane.showMessageDialog(null, "Arquivo
gerado com Sucesso");
        } else {
            JOptionPane.showMessageDialog(null, "Falha ao
gerar Arquivo");
        }
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getActionCommand().equals("deletar")) {
            deletar();
        }
        if (e.getActionCommand().equals("gerar arquivo")) {
            gerarArq();
        }
    }

    private void
tableModelMouseClicked(java.awt.event.MouseEvent evt) {
        cdCliente = jtUsuarios.getSelectedRow();
        TableModelListener[] model =
tableModel.getTableModelListeners();
    }
}

```

//view> ListarCompraView: classe responsável pela apresentação visual da tela de listagem de compras;

```

package view;

```

```

import java.awt.BorderLayout;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

```



```

import java.sql.Date;
import java.util.ArrayList;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingConstants;
import javax.swing.table.DefaultTableModel;
import dao.CompraDAO;
import model.Compra;
import model.CompraTableModel;

public class ListarCompraView extends JPanel implements
ActionListener{
    CompraTableModel tableModel = new CompraTableModel();
    CompraDAO cd = new CompraDAO();
    String rows[][] = {};
    String headers[] = {};
    JTable jtCompras = new JTable(new DefaultTableModel(rows,
headers));
    int n;
    ArrayList<Compra> compras;

    public ListarCompraView() {
        jtCompras.setModel(tableModel);
        inicializar();
    }
    public void inicializar() {
        JPanel panelSuperior = new JPanel();
        JPanel panelCentral = new JPanel();
        JLabel titulo = new JLabel("Histórico de Compras",
SwingConstants.CENTER);
        titulo.setFont(new Font("Arial", Font.BOLD, 40));
        compras = new ArrayList<Compra>();
        compras = (ArrayList<Compra>) cd.Listar();
        for (int i = 0; i < compras.size(); i++) {
            tableModel.addRow(compras.get(i));
        }
        JScrollPane scrollPane = new JScrollPane(jtCompras);
        JButton jbGerarArq = new JButton("Gerar Arquivo");
        jbGerarArq.addActionListener(this);
        jbGerarArq.setActionCommand("gerar arquivo");
    }
}

```

```

        this.setLayout(new BorderLayout());
        panelSuperior.add(titulo);
        panelCentral.add(scrollPane, BorderLayout.NORTH);
        this.add(panelSuperior, BorderLayout.NORTH);
        this.add(panelCentral, BorderLayout.CENTER);
        this.add(jbGerarArq, BorderLayout.EAST);
    }

    public void gerarArq() {
        CompraDAO cd = new CompraDAO();
        n++;
        Date data = new Date(System.currentTimeMillis());
        String arq = ""+n+"RelatórioDeCompra-"+data+".txt";
        ArrayList texto = (ArrayList) compras;
        if(cd.gerarArq(arq, texto)) {
            JOptionPane.showMessageDialog(null, "Arquivo
gerado com sucesso");
        }
        else {
            JOptionPane.showMessageDialog(null, "Falha ao
gerar arquivo");
        }
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getActionCommand().equals("gerar arquivo")) {
            gerarArq();
        }
    }
}

```

//view> ListarFornecedorView: classe responsável pela apresentação visual da tela de listagem de fornecedores;

```

package view;

import java.awt.BorderLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.sql.Date;
import java.util.ArrayList;

```

```

import java.util.List;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingConstants;
import javax.swing.event.TableModelListener;
import javax.swing.table.DefaultTableModel;
import dao.FornecedorDAO;
import model.Fornecedor;
import model.FornecedorTableModel;

public class ListarFornecedorView extends JPanel implements
ActionListener {

    private static final long serialVersionUID = 1L;
    FornecedorTableModel tableModel = new
FornecedorTableModel();
    FornecedorDAO fd = new FornecedorDAO();
    String rows[][] = {};
    String headers[] = {};
    JTable jtFornecedor = new JTable(new DefaultTableModel(rows,
headers));
    int cdFornecedor;
    List<Fornecedor> fornecedores;
    int n;

    public ListarFornecedorView() {
        jtFornecedor.setModel(tableModel);
        janelaPrincipal();
    }

    public void janelaPrincipal() {
        FornecedorDAO fd = new FornecedorDAO();
        fornecedores = (List<Fornecedor>) fd.Listar();
        Fornecedor f = new Fornecedor();
        for (int i = 0; i < fornecedores.size(); i++) {
            tableModel.addRow(fornecedores.get(i));
        }

        JPanel panelSuperior = new JPanel();

```

```

        JLabel titulo = new JLabel("Fornecedores Cadastrados",
SwingConstants.CENTER);
        titulo.setFont(new Font("Arial", Font.BOLD, 40));
        panelSuperior.add(titulo);
        JPanel panelCentral = new JPanel();
        JLabel cdLivro = new JLabel("Código",
SwingConstants.CENTER);
        cdLivro.setFont(new Font("Arial", Font.BOLD, 30));
        JLabel nmLivro = new JLabel("Nome",
SwingConstants.HORIZONTAL);
        nmLivro.setFont(new Font("Arial", Font.BOLD, 30));
        JScrollPane scrollPane = new JScrollPane(jtFornecedor);
        JButton jbDeletar = new JButton("Deletar");
        jbDeletar.addActionListener(this);
        jbDeletar.setActionCommand("deletar");
        JButton jbGerarArq = new JButton("Gerar Arquivo");
        jbGerarArq.addActionListener(this);
        jbGerarArq.setActionCommand("gerar arquivo");
        JButton jbAtualizar = new JButton("Atualizar");
        jbAtualizar.addActionListener(this);
        jbAtualizar.setActionCommand("atualizar");
        JPanel panelWest = new JPanel();
        panelWest.setLayout(new GridLayout(2, 1));
        this.setLayout(new BorderLayout());
        this.add(panelSuperior, BorderLayout.NORTH);
        panelCentral.add(scrollPane, BorderLayout.NORTH);
        this.add(panelCentral, BorderLayout.CENTER);
        panelWest.add(jbDeletar);
        panelWest.add(jbAtualizar);
        this.add(panelWest, BorderLayout.WEST);
        this.add(jbGerarArq, BorderLayout.EAST);
    }

    public void deletar() {
        MouseEvent evt = null;
        tableModelMouseClicked(evt);
        int cd =
fornecedores.get(jtFornecedor.getSelectedRow()).getCodigo();
        FornecedorDAO fd = new FornecedorDAO();
        fd.deletar(cd);
    }

    public void gerarArq() {
        FornecedorDAO fd = new FornecedorDAO();

```

```

        n++;
        Date data = new Date(System.currentTimeMillis());
        String arq = "" + n + "RelatórioDeFornecedores-" + data +
".txt";

        ArrayList texto = (ArrayList) fornecedores;
        if (fd.gerarArq(arq, texto)) {
            JOptionPane.showMessageDialog(null, "Arquivo
gerado com Sucesso");
        } else {
            JOptionPane.showMessageDialog(null, "Falha ao
gerar Arquivo");
        }
    }

    public void atualizar() {
        if (jtFornecedor.getSelectedRow() != -1) {
            MouseEvent evt = null;
            tableModelMouseClicked(evt);
            int id =
fornecedores.get(jtFornecedor.getSelectedRow()).getCodigo();
            AtualizarFornecedorView afv = new
AtualizarFornecedorView(id);
            this.add(afv);
            this.revalidate();
            this.repaint();
        } else {
            JOptionPane.showMessageDialog(null, "Selecione
algum fornecedor da tabela");
        }
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        if (e.getActionCommand().equals("deletar")) {
            deletar();
            ListarFornecedorView lfv = new
ListarFornecedorView();
            lfv.repaint();
            lfv.validate();
        } else if (e.getActionCommand().equals("gerar arquivo")) {
            gerarArq();
        } else if (e.getActionCommand().equals("atualizar")) {
            atualizar();
        }
    }

```

```

        }
    }

    private void
    tableModelMouseClicked(java.awt.event.MouseEvent evt) {
        cdFornecedor = jtFornecedor.getSelectedRow();
        TableModelListener[] model =
        tableModel.getTableModelListeners();
    }
}

```

//view> ListarLivroView: classe responsável pela apresentação visual da tela de listagem de livros;

```

package view;

import java.awt.BorderLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.sql.Date;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingConstants;
import javax.swing.event.TableModelListener;
import javax.swing.table.DefaultTableModel;
import dao.LivroDAO;
import model.Livro;
import model.LivroTableModel;

public class ListarLivroView extends JPanel implements ActionListener{
    LivroTableModel tableModel = new LivroTableModel();
    LivroDAO ld = new LivroDAO();
    String rows[][]= {};
    String headers[] = {};
    JTable jtLivros = new JTable(new DefaultTableModel(rows,
headers));
}

```

```

int cdLivro;
Livro l;
List<Livro> livros;
int n;

public ListarLivroView() {
    jtLivros.setModel(tableModel);
    janelaPrincipal();
}

public void janelaPrincipal(){
    livros = (ArrayList<Livro>) ld.Listar();
    for (int i = 0; i < livros.size(); i++) {
        tableModel.addRow(livros.get(i));
    }
    JPanel panelSuperior = new JPanel();
    JLabel titulo = new JLabel("Livros
Cadastrados",SwingConstants.CENTER);
    titulo.setFont(new Font("Arial", Font.BOLD, 40));
    panelSuperior.add(titulo);
    JPanel panelCentral = new JPanel();
    JLabel cdLivro = new JLabel("Código",
SwingConstants.CENTER);
    cdLivro.setFont(new Font("Arial", Font.BOLD, 30));
    JLabel nmLivro = new JLabel("Título",
SwingConstants.HORIZONTAL);
    nmLivro.setFont(new Font("Arial", Font.BOLD, 30));
    JButton btDeletar= new JButton("Deletar");
    btDeletar.addActionListener(this);
    btDeletar.setActionCommand("deletar");
    JButton btGerarArq = new JButton("Gerar Arquivo");
    btGerarArq.addActionListener(this);
    btGerarArq.setActionCommand("gerar arquivo");
    JButton btAtualizar = new JButton("Atualizar");
    btAtualizar.addActionListener(this);
    btAtualizar.setActionCommand("atualizar");
    JScrollPane scrollPane = new JScrollPane(jtLivros);
    JPanel panelWest = new JPanel();
    panelWest.setLayout(new GridLayout(2,1));
    this.setLayout(new BorderLayout());
    this.add(panelSuperior, BorderLayout.NORTH);
    panelCentral.add(scrollPane,BorderLayout.NORTH);
    this.add(panelCentral, BorderLayout.CENTER);
    panelWest.add(btDeletar);

```

```

        panelWest.add(btAtualizar);
        this.add(panelWest, BorderLayout.WEST);
        this.add(btGerarArq, BorderLayout.EAST);
    }

    public void deletar() {
        MouseEvent evt = null;
        tableModelMouseClicked(evt);
        int cd = livros.get(jtLivros.getSelectedRow()).getCdLivro();
        LivroDAO ld = new LivroDAO();
        ld.deletar(cd);
    }

    public void gerarArq() {
        LivroDAO ld = new LivroDAO();
        n++;
        Date data = new Date(System.currentTimeMillis());
        String arq = ""+n+"RelatorioDeLivros-"+data+".txt";
        ArrayList texto = (ArrayList) livros;
        if(ld.gerarArq(arq, texto)) {
            JOptionPane.showMessageDialog(null, "Arquivo
gerado com Sucesso");
        }
        else {
            JOptionPane.showMessageDialog(null, "Falha ao
gerar Arquivo");
        }
    }

    public void atualizar() {
        if (jtLivros.getSelectedRow() != -1) {
            MouseEvent evt = null;
            tableModelMouseClicked(evt);
            int id =
livros.get(jtLivros.getSelectedRow()).getCdLivro();
            AtualizarLivroView alv = new
AtualizarLivroView(id);
            this.add(alv);
            this.revalidate();
            this.repaint();
        }
        else {
            JOptionPane.showMessageDialog(null, "Selecione
algum livro da tabela");
        }
    }

```



```

    }
}

@Override
public void actionPerformed(ActionEvent e) {
    if(e.getActionCommand().equals("deletar")) {
        deletar();
    }
    else if(e.getActionCommand().equals("gerar arquivo")) {
        gerarArq();
    }
    else if(e.getActionCommand().equals("atualizar")) {
        atualizar();
    }
}

private void
tableModelMouseClicked(java.awt.event.MouseEvent evt) {
    cdLivro = jtLivros.getSelectedRow();
    TableModelListener[] model =
tableModel.getTableModelListeners();
}
}

```

//view> MenuView: classe responsável pela apresentação visual da tela inicial do sistema e pela construção da barra de tarefas;

```

package view;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Date;
import java.util.ArrayList;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import dao.LivroDAO;
import dao.SistemaDAO;

public class MenuView extends JFrame implements ActionListener {
    JMenuBar menuBar;

```

```

        JMenu menuCadastro, menuCompra, menuListar, menuOpcoes;
        JMenuItem itemLivro, itemFornecedor, itemCompra,
        itemListarLivro, itemListarFornecedor, itemSair, itemListarCliente,
            itemListarCompra, itemAjuda;
        ImageIcon imagem1 = new
        ImageIcon(getClass().getResource("Logo2.png"));
        JLabel imagem = new JLabel(imagem1);

        public MenuView() {
            this.inicializar();
            this.construir();
            add(imagem);
            this.setJMenuBar(menuBar);

            this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            this.setSize(800, 600);
            this.setVisible(true);
        }

        public void inicializar() {
            menuBar = new JMenuBar();
            menuCadastro = new JMenu("Cadastro");
            menuCompra = new JMenu("Compras");
            menuListar = new JMenu("Listar");
            menuOpcoes = new JMenu("Opções");
            itemLivro = new JMenuItem("Cadastrar Livro");
            itemLivro.addActionListener(this);
            itemLivro.setActionCommand("cadastrarLivro");
            itemFornecedor = new JMenuItem("Cadastrar
Fornecedor");
            itemFornecedor.addActionListener(this);

            itemFornecedor.setActionCommand("cadastrarFornecedor");
            itemCompra = new JMenuItem("Gerar Compra");
            itemCompra.addActionListener(this);
            itemCompra.setActionCommand("gerarCompra");
            itemListarLivro = new JMenuItem("Listar Livros");
            itemListarLivro.addActionListener(this);
            itemListarLivro.setActionCommand("listarLivro");
            itemListarFornecedor = new JMenuItem("Listar
Fornecedor");
            itemListarFornecedor.addActionListener(this);

            itemListarFornecedor.setActionCommand("listarFornecedor");

```

```

        itemListarCliente = new JMenuItem("Listar Clientes");
        itemListarCliente.addActionListener(this);
        itemListarCliente.setActionCommand("listarCliente");
        itemListarCompra = new JMenuItem("Listar Compra");
        itemListarCompra.addActionListener(this);
        itemListarCompra.setActionCommand("listarCompra");
        itemSair = new JMenuItem("Sair");
        itemSair.addActionListener(this);
        itemSair.setActionCommand("sair");
        itemAjuda = new JMenuItem("Ajuda");
        itemAjuda.addActionListener(this);
        itemAjuda.setActionCommand("ajuda");
    }

```

```

    public void construir() {
        menuCadastro.add(itemLivro);
        menuCadastro.add(itemFornecedor);
        menuBar.add(menuCadastro);
        menuCompra.add(itemCompra);
        menuBar.add(menuCompra);
        menuListar.add(itemListarLivro);
        menuListar.add(itemListarFornecedor);
        menuListar.add(itemListarCliente);
        menuListar.add(itemListarCompra);
        menuBar.add(menuListar);
        menuOpcoes.add(itemSair);
        menuOpcoes.add(itemAjuda);
        menuBar.add(menuOpcoes);
    }

```

@Override

```

    public void actionPerformed(ActionEvent e) {
        if (e.getActionCommand().equals("cadastrarLivro")) {
            CadastroLivroView clv = new CadastroLivroView();
            clv.setVisible(true);
            this.getContentPane().removeAll();
            this.getContentPane().add(clv);
            this.revalidate();
            this.repaint();
        } else if
(e.getActionCommand().equals("cadastrarFornecedor")) {
            CadastroFornecedorView cfv = new
CadastroFornecedorView();
            cfv.setVisible(true);

```

```

        this.getContentPane().removeAll();
        this.getContentPane().add(cfv);
        this.revalidate();
        this.repaint();
    }

    else if (e.getActionCommand().equals("gerarCompra")) {
        GerarCompraView gcv = new
GerarCompraView(this);
        gcv.setVisible(true);
        this.getContentPane().removeAll();
        this.getContentPane().add(gcv);
        this.revalidate();
        this.repaint();
    }

    else if (e.getActionCommand().equals("listarLivro")) {
        ListarLivroView llv = new ListarLivroView();
        llv.setVisible(true);
        this.getContentPane().removeAll();
        this.getContentPane().add(llv);
        this.revalidate();
        this.repaint();
    }

    else if (e.getActionCommand().equals("listarFornecedor"))
{
        ListarFornecedorView lfv = new
ListarFornecedorView();
        lfv.setVisible(true);
        this.getContentPane().removeAll();
        this.getContentPane().add(lfv);
        this.revalidate();
        this.repaint();
    }

    else if (e.getActionCommand().equals("listarCliente")) {
        ListarClienteView lcv = new ListarClienteView();
        lcv.setVisible(true);
        this.getContentPane().removeAll();
        this.getContentPane().add(lcv);
        this.revalidate();
        this.repaint();
    } else if (e.getActionCommand().equals("listarCompra")) {

```

```

        ListarCompraView lcv = new ListarCompraView();
        lcv.setVisible(true);
        this.getContentPane().removeAll();
        this.getContentPane().add(lcv);
        this.revalidate();
        this.repaint();
    } else if (e.getActionCommand().equals("sair")) {
        JOptionPane.showMessageDialog(null, "Clique em
'OK' para finalizar o programa");
        System.exit(0);
    } else if (e.getActionCommand().equals("ajuda")) {
        SistemaDAO sd = new SistemaDAO();
        String arq = "Texto_de_Ajuda.txt";
        String texto = " ";
        if (sd.gerarArq(arq, texto)) {
            JOptionPane.showMessageDialog(null,
"Arquivo gerado com Sucesso");
        } else {
            JOptionPane.showMessageDialog(null,
"Falha ao gerar Arquivo");
        }
    }
}
}
}

```

5.4. Apêndice D - Tela de apresentação do sistema em Java

