



### Sistema para loja de produtos de jardinagem - AgroPopShop

<https://github.com/algaeworks/curso-especialista-jpa/tree/master/diagrama-domain-model-especialista-jpa>  
<https://sites.google.com/a/cin.ufpe.br/if718/home?authuser=0>

Neste semestre, será projetado e desenvolvido um sistema de informação online para uma loja de produtos de jardinagem. A implementação será feita de forma incremental. Há 2 *milestones*<sup>1</sup>, e em cada *milestone* novas funcionalidades serão adicionadas ao sistema. Em cada *milestone* há várias *userstories*.

## 1. DESCRIÇÃO

O propósito é construir um sistema de informação para cadastro de clientes, de produtos e de pedidos de utensílios e insumos para uma única loja. Em sistemas maiores, seriam acrescentadas funcionalidades de gestão de pedidos, gestão de estoque, gestão financeira, gestão de funcionários, geração de relatórios administrativos, ferramentas de marketing. Mas esse escopo é  **muito grande**  para 2 meses! Então, o professor simplificou o projeto para a época de pandemia e ensino remoto.

O sistema conterà cadastro os clientes, cadastro de produtos, registro de pedidos com vários itens, registro do tipo e o valor do pagamento efetuado, entre outras funcionalidades. Além disso, o sistema é responsável pela emissão de alguns relatórios gerenciais, contendo quantidade de pedidos, faturamento e lucro da loja em um período de tempo determinado.

### 1.1. Os Milestones

**Milestone 1** - Entregas de 13/04/2021 até 10/05/2021 - Implemente as *userstories* 1, 2, 3, 4.

**Milestone 2** - Entregas de 11/05/2021 até 05/06/2021 – Escolha e implemente cinco *userstories*.

### 1.2. As UserStories

As *user stories* levantadas para o sistema estão mostradas a seguir. *User Stories* são uma forma de expressar **requisitos funcionais** desejados para o sistema (o que o sistema faz, e não como ele faz). Observe que, no mundo real, o cliente poderia mudar de ideia com respeito a esses requisitos funcionais. As *user stories* foram priorizadas pelo cliente na ordem que segue.

UserStory 1	Manter e exibir informações de um cliente (1,0 ponto)
- Adicione um novo cliente ao sistema. Manter as seguintes informações: código (único, autogerado), nome completo, gênero (masculino, feminino, não informar), endereço, e-Mail, CEP, e mais um campo que o professor te dirá. Tudo num <b>CRUD</b> : <i>Create, Read, Update, Delete</i> .	

UserStory 2	Manter e exibir informações de produtos (1,5 ponto)
- Adicione um novo produto ao sistema. Manter as seguintes informações: código (único, autogerado), nome do produto, marca, altura, largura, profundidade, peso, preço. Tudo num <b>CRUD</b> : <i>Create, Read, Update, Delete</i> .	

<sup>1</sup> *Milestone* ou marco é uma técnica para gerenciamento de projetos que permite o teste da funcionalidade de um novo produto ao longo do projeto. O termo é uma expressão inglesa (referente a um marco quilométrico) utilizada como designação de um ponto de controle em um cronograma, pela definição de pontos de checagem ou marcos de desenvolvimento. Representa a conclusão de um conjunto de tarefas ou fase, passiva de aprovação e formalização por parte do cliente (Fonte: <https://pt.wikipedia.org/wiki/Milestone>)

<b>UserStory 3</b>	<b>Manter e exibir informação de dependentes de clientes (0,5 ponto)</b>
<ul style="list-style-type: none"> <li>- Adicione um novo dependente de um cliente ao sistema. Após o cadastro do cliente, será possível vincular filho(s)/dependente(s). Manter as seguintes informações: código (único, autogerado), nome completo, data de nascimento, gênero (masculino, feminino, não informado). Tudo num <b>CRUD</b>: <i>Create, Read, Update, Delete</i>. Ao excluir um cliente, os dependentes são excluídos em cascata.</li> </ul>	

<b>UserStory 4</b>	<b>Manter e exibir informação de pedidos (2,0 pontos)</b>
<ul style="list-style-type: none"> <li>- Ao cadastrar um pedido, todos os produtos cadastrados são exibidos em uma listagem, ordenados alfabeticamente pelo nome. O cliente deve selecionar os itens com a quantidade desejada. Ao cadastrar o pedido, o cliente deve selecionar a forma de pagamento (dinheiro ou cartão de crédito). Guarde os dados pertinentes para a forma de pagamento (número do cartão ou valor pago).</li> <li>- Ao cadastrar a venda, os seguintes dados devem ser guardados: data da venda, os itens que foram vendidos (quantidade de produtos e valor unitário). Não há controle de estoque, a quantidade de produtos disponível é infinita.</li> <li>- Ao listar os pedidos, será possível visualizar o nome do cliente, o nome do produto, a quantidade comprada, o valor unitário, o valor total dos produtos, o total do pedido, produtos com os valores e quantidades, o cálculo de imposto (fixo em 22,5%) e a forma de pagamento.</li> <li>- Ao excluir a venda, o registro de produtos e de clientes não é afetado.</li> <li>- Ao atualizar a venda, é um caos.</li> <li>- Tudo num <b>CRUD</b>: <i>Create, Read, Update, Delete</i>.</li> </ul>	

<b>UserStory 5</b>	<b>Faturamento (1,0 ponto)</b>
<ul style="list-style-type: none"> <li>- Exiba, em uma tabela com 12 linhas, o faturamento de cada mês nos últimos 12 meses, a contar de uma data informada pelo usuário. Ao final do relatório deve ser exibido, em outra tabela, o valor do faturamento total da loja, o valor total do imposto pago e a soma dos valores (faturamento + imposto fixo em 12% sobre o faturamento).</li> </ul>	

<b>UserStory 6</b>	<b>Buscar pelo nome dos produtos (1,0 ponto)</b>
<ul style="list-style-type: none"> <li>- Permita que o usuário do sistema busque os produtos pelo nome.</li> </ul>	

<b>UserStory 7</b>	<b>Buscar produtos (2,0 pontos)</b>
<ul style="list-style-type: none"> <li>- Permita que o usuário do sistema busque os produtos por qualquer campo.</li> </ul>	

<b>UserStory 8</b>	<b>Iniciar contato pelo WhatsApp (0,5 ponto)</b>
<ul style="list-style-type: none"> <li>- Faça uma página de contato e insira um botão para iniciar uma conversa no WhatsApp com um número de telefone da loja.</li> </ul>	

<b>UserStory 9</b>	<b>e-Mail promocional (1,5 ponto)</b>
<ul style="list-style-type: none"> <li>- Faça uma funcionalidade que envie um e-mail para o cliente com a promoção de 5% de desconto, após 1 mês da compra, no item mais barato que o cliente comprou no último pedido.</li> </ul>	

<b>UserStory 10</b>	<b>Previsão do tempo (1,5 ponto)</b>
<ul style="list-style-type: none"> <li>- Faça uma funcionalidade que mostre na tela de listagem dos clientes a previsão do tempo dos próximos 2 dias de acordo com o CEP cadastrado. Obtenha a previsão do tempo via um serviço externo.</li> </ul>	

<b>UserStory 11</b>	<b>Preenchimento do endereço (1,5 ponto)</b>
<ul style="list-style-type: none"> <li>- No ato do cadastro do cliente, preencha o endereço de forma automática após o cliente digitar o CEP. Obtenha o endereço via um serviço externo.</li> </ul>	

<b>UserStory 12</b>	<b>Aniversariantes do mês (1,0 ponto)</b>
<ul style="list-style-type: none"> <li>- Exiba, em uma tabela, o nome completo, a idade e a data de nascimento dos clientes e dos dependentes que são aniversariantes em um determinado mês. O mês deverá ser informado pelo usuário do sistema.</li> </ul>	

<b>UserStory 13</b>	<b>Gerar boleto do pedido (2,0 pontos)</b>
- Adicione o método de pagamento por boleto bancário. Use uma biblioteca externa para gerar o boleto completo com o código de barras.	

<b>UserStory 14</b>	<b>Foto do produto (2,0 pontos)</b>
- Modifique o CRUD de produto, acrescentando a foto do produto. O arquivo da foto deve ser guardado no banco de dados.	

<b>UserStory 15</b>	<b>Frete para a Marte (1,0 ponto)</b>
- Exiba, em uma tabela, o código, o nome, o preço, o peso e o volume de todos os produtos cadastrados na loja, ordenados pelo volume do produto, do mais volumoso para o menos volumoso. Para cada item, exiba o valor do frete estimado do Brasil para Marte (R\$ 123456,00 por grama).	

<b>UserStory 16</b>	<b>Desconto (1,0 ponto)</b>
- Exiba, em uma tabela, o nome, o preço original e o preço com desconto de 3,018735% dos 4 produtos mais novos cadastrados na loja, desde o início dos registros, ordenado pelo preço com desconto (do maior preço com desconto para o menor preço com desconto).	

<b>UserStory 17</b>	<b>Volume do produto (1,0 ponto)</b>
- Exiba o volume de cada produto na listagem de produtos. O volume deve ser calculado com base na altura*profundidade*largura.	

<b>UserStory 18</b>	<b>Novatos (1,0 ponto)</b>
- Exiba, em uma tabela, informações sobre novos clientes cadastrados na loja em um determinado mês e ano. O mês e o ano deverão ser informados pelo usuário do sistema. O relatório deverá exibir, em uma tabela, o código, o nome, a data de nascimento e a data de cadastro do(s) cliente(s).	

### 1.3. O que entregar:

- Entregar o código-fonte no seu repositório da disciplina no GitHub.

## 2. CORREÇÃO E AVALIAÇÃO

### 2.1 Pontos a serem observados na correção

<b>Compilação</b>	Se não compilar, a nota é zero.
<b>Qualidade do código</b>	Será avaliada a utilização de endentação de código, nome das classes, nome das variáveis, nome dos métodos, organização dos pacotes, ocorrência de loops desnecessários, uso de "gambiaras", se há repetição de código, uso de números mágicos, nomes de símbolos mal escolhidos, modularização, cascatas de <i>ifs</i> ou endentação profunda de <i>loops/testes</i> , consistência na codificação, tratamento de exceções, consistências nos nomes dos arquivos, estilo de codificação homogêneo, organização das pastas e dos pacotes, respeito ao padrão MVC, respeito a utilização do Thymeleaf na camada view.
<b>Compleitude</b>	Para cada <i>milestone</i> haverá o teste manual das funcionalidades de cada uma das <i>userstories</i> . Será avaliado se todas as <i>userstories</i> do <i>milestone</i> estão funcionais. Para cada <i>userstory</i> há uma nota. Exemplo: <b>CRUD</b> de cliente = 1,0 ponto / 4 funcionalidades = 0,25 por funcionalidade. Se o aluno não fizer uma funcionalidade de algum CRUD, o professor pode dar mais pontos se outras tecnologias forem usadas (bootstrap, materialize, heroku, Ci/Cd, spring native). <b>Não será avaliada a estética da camada view.</b> <i>A qualquer momento o professor poderá solicitar alteração ou inclusão de um campo em quaisquer dos CRUD ou alteração de consultas/relatórios presentes no projeto. Se não houver alteração em tempo hábil (cerca de 2 horas), conforme solicitado pelo professor, a nota será prejudicada.</i>
<b>Atraso</b>	Não tolero atraso. O primeiro dia de atraso resultará num desconto de 4,0 pontos na nota total da unidade. Os dias subsequentes terão desconto de 0,5 ponto por dia. Ao final de 12 dias corridos de atraso, a nota será 0,0. Agende um horário com o professor ou monitor, se necessário.

## 2.2. Dicas e considerações

1 - Não resolva um problema que você ainda não tem!

2 - Veja o manifesto ágil:

<http://agilemanifesto.org/iso/ptbr/manifesto.html>

<https://robsoncamargo.com.br/blog/Manifesto-Agil-entenda-como-surgiu-e-conheca-os-12-principios>

2.1 - Lembre-se: Indivíduos e interações mais que processos e ferramentas.

2.2 - Lembre-se: Software em funcionamento mais que documentação abrangente.

5 - Ao apagar um cliente (CRUD - **Delete**), apague todos os registros em cascata. Na vida real isso não é feito, mas facilita o seu trabalho. Não modifique o create, o read ou o update, e não faça do update um insert.

6 - Qualquer campo que tenha o valor calculado deve ser *calculado*. Não coloque no banco de dados valores que podem ser obtidos com código/programação.

7 - Na vida real, deveria ser feita uma tabela de notasFiscais, comandasPagas, paraFaturamento ou algo assim. Essa tabela é da *user story* do pedido. Fazendo assim, teríamos o registro de todo o faturamento sem mexer em pedido, sem mexer em produto, sem mexer no cliente. Na US04, deveria haver um *insert* toda vez que houver uma venda. Assim que finalizar a venda, seriam registrados todos os itens do pedido com os valores atuais, conteria o histórico de vendas a ser usado no relatório gerencial. Deveria...

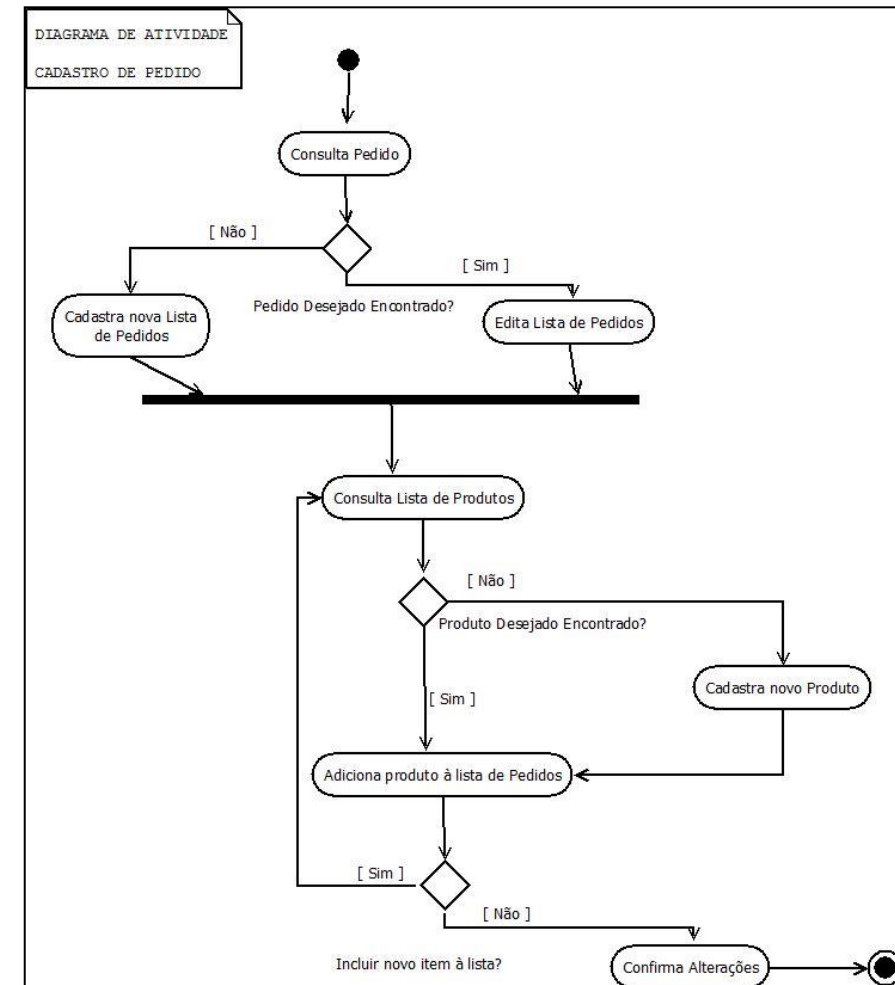
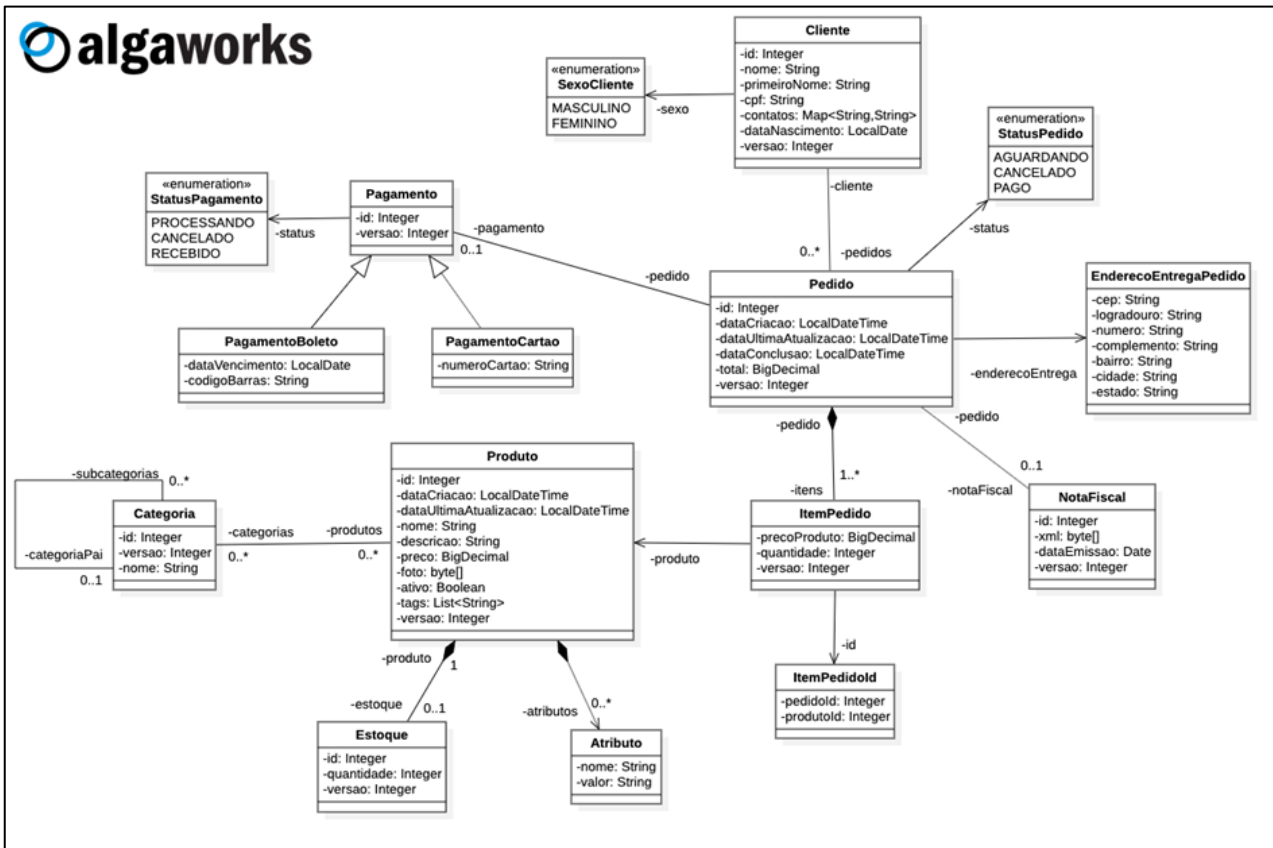
## 3.0. DESAFIOS DESAFIADORES DIFÍCIES

Desenvolver a mesma aplicação:

- a) Utilizar *Bootstrap*
- b) Utilizar *Vaadin*
- c) Utilizar testes de unidade com *jUnit*
- d) Criar uma *single page application* utilizando *react*
- e) Publicar o projeto no *heroku*
- f) Criar aplicativo mobile utilizando *Flutter*
- g) Utilizar o SGBD *Postgres* ao invés do *H2*.

#### 4.0. Modelo Conceitual + Diagrama de Atividades

A seguir é apresentado o diagrama de classes e o diagrama de atividade em UML, de uma loja genérica. Esse modelo poderá ser implementado em software, mas não precisa seguir o modelo conceitual entidade-por-entidade. Leia para entender que as *userstories* são derivadas desse modelo.



<https://github.com/algaworks/curso-especialista-jpa/tree/master/diagrama-domain-model-especialista-jpa>