

## Especificações dos casos de uso do projeto simplificado de microsserviços

### 1. Introdução

As especificações dos casos de uso a seguir trata do desenvolvimento de um sistema dividido em três microsserviços: o microsserviço cliente; produto e login. No microsserviço cliente teremos o cadastro de cliente e dos produtos que ele tem assinatura. O microsserviço produto vai conter o cadastro de produto. Já o microsserviço login vai conter o cadastro de usuário e a conexão com os dois microsserviços anteriores. O programa deverá ser desenvolvido seguindo as especificações a seguir: no *backend* se utilizará a linguagem de programação Java. No *frontend* será utilizado o *framework* SpringBoot do Java (para web ou intranet). Para o banco de dados pode ser utilizado uma das seguintes tecnologias: Sqlite ou MySQL.

### 2. Levantamento de requisitos

Um requisito pode ser definido como uma descrição dos serviços fornecidos pelo sistema e as suas restrições operacionais (SOMMERVILLE, 2007). Tradicionalmente, os requisitos são divididos em dois tipos: requisitos funcionais e requisitos não funcionais.

#### 2.1. Requisitos funcionais

Os requisitos funcionais descrevem o que o sistema deve fazer, isto é, definem a funcionalidade desejada do software (SOMMERVILLE, 2007). A Tabela 1 apresenta os requisitos funcionais deste projeto.

Tabela 1: Requisitos funcionais

Identificação	Requisito funcional	Prioridade
RF001	Manter cliente (microsserviço cliente)	Essencial
RF002	Manter produto (microsserviço produto)	Essencial
RF003	Manter os produtos vinculados ao cliente (microsserviço cliente)	Essencial
RF004	Manter usuário (microsserviço login)	Essencial

Fonte: Acervo do autor, 2024

##### 2.1.1. Descrição dos requisitos funcionais

Nesta sessão estão descritas as especificações dos requisitos funcionais previstos na tabela 1.

##### 2.1.1.1. Manter cliente

Este requisito funcional descreve os casos de uso relacionados a inserção, alteração, exclusão e busca do cliente no banco de dados.

Tabela 2: Cadastrar cliente

RF001	
Nome:	Cadastrar cliente
Descrição:	O programa deve inserir um novo cliente no banco de dados.

Plataforma:	Web (Java SpringBoot)
Atores:	Administrador
Prioridade:	Essencial
Requisito não funcional associado:	RNF002; RNF003; RNF004
Entradas e restrições:	<ul style="list-style-type: none"> <li>• Código do cliente (Integer, not null, pk);</li> <li>• Nome (String, not null, lenght 150);</li> <li>• Sexo (String, not null, lenght 1);</li> <li>• Data de nascimento (Date, not null);</li> <li>• CPF (String, not null, lenght 14);</li> <li>• E-mail (String, not null, lenght 150);</li> <li>• Celular (String, not null, lenght 15);</li> <li>• Ativo** (Integer not null, default 1);</li> </ul>
Fluxo de eventos	
Fluxo principal:	Os atores deverão informar todas as entradas com especificação "not null". Após essa etapa os atores terão cadastrado um novo cliente no banco de dados.
Fluxo secundário 1:	Caso alguma informação "not null" seja inválida (não preenchida), uma janela será apresentada informando que os dados solicitados para cadastro estão incompletos.
Fluxo secundário 2:	Caso o CPF já esteja cadastrado no banco de dados, uma janela será apresentada informando que o CPF já está cadastrado no banco de dados.
Fluxo secundário 3:	Caso o E-mail já esteja cadastrado no banco de dados, uma janela será apresentada informando que o E-mail já está cadastrado no banco de dados.
Fluxo secundário 4:	Caso o Celular já esteja cadastrado no banco de dados, uma janela será apresentada informando que o número de celular já está cadastrado no banco de dados.

Fonte: Acervo do autor, 2024

A coluna Ativo\*\* pode receber somente um dos seguintes valores: 1 caso o cliente esteja ativo e 0 caso o cliente esteja inativo.

Tabela 3: Alterar cliente

<b>RF001</b>	
Nome:	Alterar cliente
Descrição:	O programa deve alterar os dados do cliente no banco de dados.
Plataforma:	Web (Java SpringBoot)
Atores:	Administrador e Cliente
Prioridade:	Desejável
Requisito não funcional associado:	RNF002; RNF003; RNF004
Entradas e restrições:	<ul style="list-style-type: none"> <li>• Código do cliente (Integer, not null, pk);</li> <li>• Nome (String, not null, lenght 150);</li> <li>• Sexo (String, not null, lenght 1);</li> <li>• Data de nascimento (Date, not null);</li> <li>• CPF (String, not null, lenght 14);</li> </ul>

	<ul style="list-style-type: none"> <li>E-mail (String, not null, lenght 150);</li> <li>Celular (String, not null, lenght 15);</li> <li>Ativo** (Integer not null, default 1);</li> </ul>
Fluxo de eventos	
Fluxo principal:	Os atores deverão informar todas as entradas com especificação "not null". Após essa etapa os atores terão alterado os dados do cliente no banco de dados.
Fluxo secundário 1:	Caso alguma informação "not null" seja inválida (não preenchida), uma janela será apresentada informando que os dados solicitados para a alteração do cadastro estão incompletos.
Fluxo secundário 2:	Caso o CPF já esteja cadastrado no banco de dados, uma janela será apresentada informando que o CPF já está cadastrado no banco de dados.
Fluxo secundário 3:	Caso o E-mail já esteja cadastrado no banco de dados, uma janela será apresentada informando que o E-mail já está cadastrado no banco de dados.
Fluxo secundário 3:	Caso o Celular já esteja cadastrado no banco de dados, uma janela será apresentada informando que o número de celular já está cadastrado no banco de dados.

Fonte: Acervo do autor, 2024

A coluna Ativo\*\* pode receber somente um dos seguintes valores: 1 caso o cliente esteja ativo e 0 caso o cliente esteja inativo.

Tabela 4: Excluir cliente

<b>RF001</b>	
Nome:	Excluir cliente
Descrição:	O programa deve excluir os dados do cliente no banco de dados.
Plataforma:	Web (Java SpringBoot)
Atores:	Administrador
Prioridade:	Desejável
Requisito não funcional associado:	RNF002; RNF003; RNF004
Entradas e restrições:	<ul style="list-style-type: none"> <li>Código (Integer, not null, primary key);</li> </ul>
Fluxo de eventos	
Fluxo principal:	Os atores deverão informar o código do cliente que deseja excluir. Após essa etapa os atores terão excluído os dados do cliente no banco de dados.
Fluxo secundário 1:	Caso o código passado seja inválido (código inexistente), uma janela será apresentada informando que o cliente não se encontra no sistema.
Fluxo secundário 2:	Caso o cliente tenha um ou mais produtos associados a ele, uma janela será apresentada para alertar sobre a existência de produto/s associado/s ao cliente com duas alternativas de botão: "Excluir"

	ou “Cancelar”.
Fluxo secundário 3:	Caso a escolha do botão no Fluxo secundário 2 for “Excluir”, o/s produto/s também será/ão excluído/s do banco de dados. Caso a escolha for “Cancelar”, o cliente e o/s produto/s não serão excluídos do banco de dados, mas sim inativados.

Fonte: Acervo do autor, 2024

Tabela 5: Listar cliente

RF001	
Nome:	Listar cliente
Descrição:	O programa deve listar os dados dos clientes cadastrados no banco de dados.
Plataforma:	Web (Java SpringBoot)
Atores:	Administrador e Cliente
Prioridade:	Desejável
Requisito não funcional associado:	RNF002; RNF003; RNF004
Saídas:	<ul style="list-style-type: none"> <li>• Código (Integer, not null, primary key);</li> <li>• Nome;</li> <li>• Sexo;</li> <li>• E-mail;</li> <li>• Ativo;</li> </ul>
Fluxo de eventos	
Fluxo principal:	Os atores poderão informar o código e/ou o nome e/ou o sexo e/ou e-mail do cliente para realizar a busca. Após essa etapa os atores terão a listagem dos dados dos clientes cadastrados no banco de dados.
Fluxo secundário 1:	Caso os parâmetros passados não encontrem nenhuma informação no banco de dados, uma janela será apresentada uma janela informando que o não existe nenhum cliente cadastrado no sistema com as informações passadas.

Fonte: Acervo do autor, 2024

### 2.1.1.2. Manter produto

Este requisito funcional descreve os casos de uso relacionados a inserção e busca do produto no banco de dados.

Tabela 6: Cadastrar produto

RF002	
Nome:	Cadastrar produto
Descrição:	O programa deve inserir um novo produto no banco de dados.
Plataforma:	Web (Java SpringBoot)
Atores:	Administrador
Prioridade:	Essencial

Requisito não funcional associado:	RNF002; RNF003; RNF004
Entradas e restrições:	<ul style="list-style-type: none"> <li>Código do produto (Integer, not null, pk);</li> <li>Título do produto (String, not null, length 150);</li> <li>Descrição do produto (String, not null, length 1000);</li> <li>Ativo** (Integer, not null, default 1);</li> </ul>
<b>Fluxo de eventos</b>	
Fluxo principal:	Os atores deverão informar todas as entradas com especificação "not null". Após essa etapa o administrador terá cadastrado novo produto no banco de dados.
Fluxo secundário 1:	Caso alguma informação "not null" seja inválida (não preenchida), uma janela será apresentada informando que os dados solicitados para cadastro estão incompletos.

Fonte: Acervo do autor, 2024

A coluna Ativo\*\* pode receber somente um dos seguintes valores: 1 caso o produto esteja ativo e 0 caso o produto esteja inativo.

Tabela 7: Alterar produto

<b>RF002</b>	
Nome:	Alterar produto
Descrição:	O programa deve alterar os dados do produto no banco de dados.
Plataforma:	Web (Java SpringBoot)
Atores:	Administrador
Prioridade:	Desejável
Requisito não funcional associado:	RNF002; RNF003; RNF004
Entradas e restrições:	<ul style="list-style-type: none"> <li>Código do produto (Integer, not null, pk);</li> <li>Título do produto (String, not null, length 150);</li> <li>Descrição do produto (String, not null, length 1000);</li> <li>Ativo** (Integer, not null, default 1);</li> </ul>
<b>Fluxo de eventos</b>	
Fluxo principal:	Os atores deverão informar todas as entradas com especificação "not null". Após essa etapa os atores terão alterado os dados do produto no banco de dados.
Fluxo secundário 1:	Caso alguma informação "not null" seja inválida (não preenchida), uma janela será apresentada informando que os dados solicitados para alteração estão incompletos.

Fonte: Acervo do autor, 2024

A coluna Ativo\*\* pode receber somente um dos seguintes valores: 1 caso o produto esteja ativo e 0 caso o produto esteja inativo.

Tabela 8: Excluir produto

RF002	
Nome:	Excluir produto
Descrição:	O programa deve excluir os dados do produto no banco de dados.
Plataforma:	Web (Java SpringBoot)
Atores:	Administrador
Prioridade:	Desejável
Requisito não funcional associado:	RNF002; RNF003; RNF004
Entradas e restrições:	<ul style="list-style-type: none"> <li>Código do produto (Integer, not null, primary key);</li> </ul>
Fluxo de eventos	
Fluxo principal:	Os atores deverão informar o código do produto a ser excluído no banco de dados. Após essa etapa o produto será excluído no banco de dados.

Fonte: Acervo do autor, 2024

Tabela 9: Listar produto

RF002	
Nome:	Listar produto
Descrição:	O programa deve listar os dados do produto cadastrado no banco de dados.
Plataforma:	Web (Java SpringBoot)
Atores:	Cliente e Administrador
Prioridade:	Desejável
Requisito não funcional associado:	RNF002; RNF003; RNF004
Saídas:	<ul style="list-style-type: none"> <li>Código do produto;</li> <li>Título do produto;</li> <li>Descrição do produto;</li> <li>Ativo;</li> </ul>
Fluxo de eventos	
Fluxo principal:	Os atores podem informar o nome do produto para realizar a busca. Após essa etapa os atores terão a lista dos produtos cadastrados no banco de dados.
Fluxo secundário 1:	Caso os parâmetros passados não encontrem nenhuma informação no banco de dados, uma janela será apresentada informando que o não existe nenhum produto cadastrado no banco de dados com as informações passadas.

Fonte: Acervo do autor, 2024

### 2.1.1.3. Manter produto vinculado ao cliente

Este requisito funcional descreve os casos de uso relacionados a inserção e busca da viagem no banco de dados.

Tabela 10: Cadastrar produto do cliente

RF003	
Nome:	Cadastrar produto do cliente
Descrição:	O programa deve inserir um novo vínculo de produto cliente no banco de dados.
Plataforma:	Web (Java SpringBoot)
Atores:	Administrador
Prioridade:	Essencial
Requisito não funcional associado:	RNF002; RNF003; RNF004
Entradas e restrições:	<ul style="list-style-type: none"> <li>• Código (Integer, not null, pk);</li> <li>• Código do produto (Integer, not null, foreign key);</li> <li>• Código do cliente (Integer, not null, foreign key);</li> <li>• Data da ativação (Date, not null);</li> <li>• Data da inativação (Date, not null);</li> <li>• Valor do produto (Double, not null);</li> <li>• Desconto (Double, not null, default 0.00);</li> <li>• Ativo** (Integer, not null, default 1);</li> </ul>
Fluxo de eventos	
Fluxo principal:	Os atores deverão informar todas as entradas com especificação "not null". Após essa etapa o administrador terá cadastrado um novo vínculo do produto cliente no banco de dados.
Fluxo secundário 1:	Caso alguma informação "not null" seja inválida (não preenchida), uma janela será apresentada informando que os dados solicitados para cadastro estão incompletos.

Fonte: Acervo do autor, 2024

A coluna Ativo\*\* pode receber somente um dos seguintes valores: 1 caso o vínculo do produto ao cliente esteja ativo e 0 caso o vínculo do produto ao cliente esteja inativo.

Tabela 11: Alterar produto do cliente

RF003	
Nome:	Alterar produto do cliente
Descrição:	O programa deve alterar os dados do vínculo do produto do cliente no banco de dados.
Plataforma:	Web (SpringBoot)
Atores:	Administrador
Prioridade:	Desejável
Requisito não funcional associado:	RNF002; RNF003; RNF004
Entradas e restrições:	<ul style="list-style-type: none"> <li>• Código (Integer, not null, pk);</li> <li>• Código do produto (Integer, not null, foreign key);</li> <li>• Código do cliente (Integer, not null, foreign key);</li> <li>• Data da ativação (Date, not null);</li> <li>• Data da inativação (Date, not null);</li> <li>• Valor do produto (Double, not null);</li> <li>• Desconto (Double, not null, default 0.00);</li> <li>• Ativo** (Integer, not null);</li> </ul>
Fluxo de eventos	

Fluxo principal:	Os atores deverão informar todas as entradas com especificação “not null”. Após essa etapa os atores terão alterado os dados da viagem no banco de dados.
Fluxo secundário 1:	Caso alguma informação “not null” seja inválida (não preenchida), uma janela será apresentada informando que os dados solicitados para alteração estão incompletos.

Fonte: Acervo do autor, 2024

A coluna Ativo\*\* pode receber somente um dos seguintes valores: 1 caso o vínculo do produto ao cliente esteja ativo e 0 caso o vínculo do produto ao cliente esteja inativo.

Tabela 12: Excluir produto do cliente

RF002	
Nome:	Excluir produto do cliente
Descrição:	O programa deve excluir os dados do vínculo do produto do cliente no banco de dados.
Plataforma:	Web (SpringBoot)
Atores:	Administrador
Prioridade:	Desejável
Requisito não funcional associado:	RNF002; RNF003; RNF004
Entradas e restrições:	<ul style="list-style-type: none"> <li>Código(Integer, not null, primary key);</li> </ul>
Fluxo de eventos	
Fluxo principal:	Os atores deverão informar o código do vínculo do produto do cliente a ser excluído no banco de dados. Após essa etapa o vínculo do produto do cliente será excluído no banco de dados.

Fonte: Acervo do autor, 2024

Tabela 13: Listar produto do cliente

RF003	
Nome:	Listar produto do cliente
Descrição:	O programa deve listar os dados do produto vinculado ao cliente cadastrado no banco de dados.
Plataforma:	Web (SpringBoot)
Atores:	Cliente e Administrador
Prioridade:	Desejável
Requisito não funcional associado:	RNF002; RNF003; RNF004
Saídas:	<ul style="list-style-type: none"> <li>Código do vínculo;</li> <li>Nome do produto;</li> <li>Nome do cliente;</li> <li>Data da ativação;</li> <li>Data da inativação;</li> <li>Valor do produto;</li> <li>Desconto (Double, not null, default 0.00;</li> <li>Ativo;</li> </ul>



Fluxo de eventos	
Fluxo principal:	Os atores podem informar o nome do produto para realizar a busca. Após essa etapa os atores terão a lista do produto do cliente cadastrado no banco de dados.
Fluxo secundário 1:	Caso os parâmetros passados não encontrem nenhuma informação no banco de dados, uma janela será apresentada informando que o não existe nenhum produto vinculado ao cliente cadastrado no banco de dados com as informações passadas.

Fonte: Acervo do autor, 2024

#### 2.1.1.4. Manter usuário

Este requisito funcional descreve os casos de uso relacionados a inserção, alteração, exclusão e busca de um ou mais usuários do sistema no banco de dados.

Tabela 14: Cadastrar usuário

RF004	
Nome:	Cadastrar usuário
Descrição:	O programa deve inserir um novo usuário no banco de dados.
Plataforma:	Web (SpringBoot)
Atores:	Administrador
Prioridade:	Essencial
Requisito não funcional associado:	RNF002; RNF003; RNF004
Entradas e restrições:	<ul style="list-style-type: none"> <li>• Código (Integer, not null, pk);</li> <li>• Nome (String, not null, lenght 150);</li> <li>• Sexo (String, not null, lenght 1);</li> <li>• E-mail (String, not null, lenght 150, unique);</li> <li>• Senha (String, not null, lenght 10);</li> <li>• Grupo** (String, not null, lenght 15);</li> <li>• Ativo** (Integer, not null, default 1);</li> </ul>
Fluxo de eventos	
Fluxo principal:	Os atores deverão informar todas as entradas com especificação "not null". Após essa etapa será cadastrado um novo usuário de sistema no banco de dados.
Fluxo secundário 1:	Caso alguma informação "not null" seja inválida (não preenchida), uma janela será apresentada informando que os dados solicitados para cadastro estão incompletos.

Fonte: Acervo do autor, 2024

A coluna Ativo\*\* pode receber somente um dos seguintes valores: 1 caso o usuário esteja ativo e 0 caso o usuário esteja inativo.

A coluna Grupo\*\* pode receber os seguintes valores: "Administrador"; "Cliente".

Tabela 15: Alterar usuário

RF004	
Nome:	Alterar usuário
Descrição:	O programa deve alterar os dados do usuário no banco de dados.
Plataforma:	Web (SpringBoot)
Atores:	Administrador, cliente
Prioridade:	Desejável
Requisito não funcional associado:	RNF002; RNF003; RNF004
Entradas e restrições:	<ul style="list-style-type: none"> <li>• Código (Integer, not null, pk);</li> <li>• Nome (String, not null, lenght 150);</li> <li>• Sexo (String, not null, lenght 1);</li> <li>• E-mail (String, not null, lenght 150, unique);</li> <li>• Senha (String, not null, lenght 10);</li> <li>• Grupo** (String, not null, lenght 15);</li> <li>• Ativo** (Integer, not null, default 1);</li> </ul>
Fluxo de eventos	
Fluxo principal:	Os atores deverão informar todas as entradas com especificação "not null". Após essa etapa os atores terão alterado os dados do usuário no banco de dados.
Fluxo secundário 1:	Caso alguma informação "not null" seja inválida (não preenchida), uma janela será apresentada informando que os dados solicitados para alteração estão incompletos.

Fonte: Acervo do autor, 2024

A coluna Ativo\*\* pode receber somente um dos seguintes valores: 1 caso o usuário esteja ativo e 0 caso o usuário esteja inativo.

A coluna Grupo\*\* pode receber os seguintes valores: "Administrador"; "Cliente".

Tabela 16: Excluir usuário

RF004	
Nome:	Excluir usuário
Descrição:	O programa deve excluir os dados do usuário no banco de dados.
Plataforma:	Web (SpringBoot)
Atores:	Administrador
Prioridade:	Desejável
Requisito não funcional associado:	RNF002; RNF003; RNF004
Entradas e restrições:	<ul style="list-style-type: none"> <li>• Código(Integer, not null, primary key);</li> </ul>
Fluxo de eventos	
Fluxo principal:	Os atores deverão informar o código do usuário a ser excluído do banco de dados. Após essa etapa o usuário será excluído no banco de dados.

Fonte: Acervo do autor, 2024

Tabela 17: Listar usuário

RF004	
Nome:	Listar usuário
Descrição:	O programa deve listar os usuários cadastrados no banco de dados.
Plataforma:	Web (SpringBoot)
Atores:	Administrador, Cliente
Prioridade:	Desejável
Requisito não funcional associado:	RNF002; RNF003; RNF004
Saídas:	<ul style="list-style-type: none"> <li>• Código(Integer, not null, primary key);</li> <li>• Nome;</li> <li>• E-mail;</li> <li>• Grupo;</li> <li>• Ativo;</li> </ul>
Fluxo de eventos	
Fluxo principal:	Os atores poderão informar o nome e/ou e-mail do usuário para realizar a busca. Após essa etapa os atores terão a listagem dos dados dos usuários cadastrados no banco de dados.
Fluxo secundário 1:	Caso os parâmetros passados não encontrem nenhuma informação no banco de dados, uma janela será apresentada informando que o não existe nenhum usuário cadastrado no banco de dados com as informações passadas.

Fonte: Acervo do autor, 2024

## 2.2. Requisitos não funcionais

“Os requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema” (SOMMERVILLE, 2007). A Tabela 18 apresenta os requisitos não funcionais deste projeto.

Tabela 18: Requisitos não funcionais

Identificação	Requisito não funcional	Categoria	Prioridade
RNF001	Sistema para Web	Acesso pelo navegador (Chrome, FireFox, Edge)	Essencial
RNF002	Arquitetura MVC	Padrões	Essencial
RNF003	Banco de dados	MongoDB ou Sqlite ou MySQL	Essencial
RNF004	Frontend	Html, Javascript, css.	Essencial
RNF005	Backend	Linguagem de programação Java com SpringBoot	Essencial

Fonte: Acervo do autor, 2024

### 2.2.1 Descrição dos requisitos não funcionais

Os requisitos não funcionais envolvem a escolha da arquitetura do sistema, a plataforma que o programa será implantado (*desktop*, *intranet*, *web*, etc), a linguagem de programação que será utilizada, o banco de dados apropriado para a aplicação, entre outras escolhas.

### 2.2.1.1. Sistema (RNF001)

O sistema deverá ser instalado no servidor de aplicação web e ficar disponível 24 horas por dia os sete dias da semana.

### 2.2.1.2. Arquitetura MVC (RNF002)

A arquitetura MVC (model, view, controller) é um padrão sólido para o desenvolvimento de software, proporcionando uma estrutura organizada que promove a manutenibilidade, a reutilização de código e a testabilidade. Ao entender e aplicar adequadamente os conceitos do MVC, os desenvolvedores podem criar aplicativos mais robustos e escaláveis.

### 2.2.1.3. Banco de dados (RNF003)

Existe a possibilidade de se trabalhar com um dos seguintes bancos de dados: MongoDB; SQLite ou MySQL. Esta escolha fica a critério do desenvolvedor.

O uso do SQLite é recomendado quando a simplicidade da administração, implementação e manutenção. O MySQL é um SGBD direcionado para aplicações robustas, com grandes volumes de dados e complexas. As situações onde a simplicidade é a melhor escolha, então o MongoDB pode ser uma saída para armazenar os dados.

### 2.2.1.4. Frontend (RNF004)

As linguagens de programação *frontend* lidam com as interfaces da aplicação com os usuários. Para uma plataforma na intranet ou web o ideal seria utilizar as páginas Htmls, Javascript e css associados a um *framework* como *BootStrap*, *Angular*, *React*, entre outros.

### 2.2.1.5. Backend (RNF005)

As linguagens de programação *backend* lidam com os bastidores das funcionalidades das aplicações. É o código que conecta a aplicação com o banco de dados, gerencia as conexões dos usuários e alimenta a aplicação. O *backend* trabalha em conjunto com o *frontend* para entregar o produto para o usuário final.

A linguagem de programação Java, com o *framework SpringBoot*, foi escolhida para ser utilizada nesta aplicação para o *backend*.