
Documentação de Projeto

para o sistema

Desenvolvimento automatizado de aplicações *Back-end for Front-end* (BFF Squared)

Versão 1.3

Projeto de sistema elaborado pelo aluno Marlon Henrique da Silva e apresentado ao curso de **Engenharia de Software** da **PUC Minas** como parte do Trabalho de Conclusão de Curso (TCC) sob orientação de conteúdo do professor Marco Rodrigo Costa, orientação acadêmica do professor José Laerte Xavier e orientação de TCC II do professor (a ser definido no próximo semestre).

08/03/2022

Tabela de Conteúdo

1. Introdução	1
2. Modelos de Usuário e Requisitos	1
2.1 Descrição de Atores	1
2.2 Modelos de Usuários	1
2.3 Modelo de Casos de Uso e Histórias de Usuários	4
2.4 Diagrama de Sequência do Sistema e Contrato de Operações	6
3. Modelos de Projeto	6
3.1 Diagrama de Classes	6
3.2 Diagramas de Sequência	6
3.3 Diagramas de Comunicação	6
3.4 Arquitetura	6
3.5 Diagramas de Estados	6
3.6 Diagrama de Componentes e Implantação.	6
4. Projeto de Interface com Usuário	7
5. Glossário e Modelos de Dados	11
6. Casos de Teste	11
7. Cronograma e Processo de Implementação	11

Histórico de Revisões

Nome	Data	Razões para Mudança	Versão
Criação	08/03/2022	Criação do Documento	1.0
Modificações	17/03/2022	Adição do diagrama de caso de uso e histórias de usuários	1.1
Modificações	22/03/2022	Adição dos modelos de usuário e um projeto de interface	1.2
Modificações	27/03/2022	Adição dos projetos de interface completos	1.3
Modificações	10/04/2022	Adição dos diagramas de classe, sequência e comunicação	1.4

1. Introdução

Este documento agrega: 1) a elaboração e revisão de modelos de domínio e 2) modelos de projeto para o sistema BFF Squared. A referência principal para a descrição geral do problema, domínio e requisitos do sistema é o documento de especificação que descreve a visão de domínio do sistema. Tal especificação acompanha este documento. Anexo a este documento também se encontra o Glossário.

2. Modelos de Usuário e Requisitos

Nesta seção são apresentados os modelos de usuários e requisitos para o sistema. Tal apresentação é feita pela descrição dos atores (Seção 2.1), descrição dos modelos de usuários (Seção 2.2), descrição de casos de uso de histórias dos usuários (Seção 2.3) e descrição do diagrama de sequências e contrato de operações para o sistema (seção 2.4).

2.1 Descrição de Atores

Os atores identificados para a utilização do sistema são pessoas desenvolvedoras de software, possivelmente com qualificações em *back-end*, *front-end* e *mobile*. As qualificações das pessoas desenvolvedoras podem ser diversas, pois as ações não demandam conhecimento específico e aprofundado em uma determinada tecnologia, mas é necessário que entendam da ferramenta e das suas próprias necessidades. O artefato produzido pelas interações dos usuários com o sistema é utilizado como insumo para que o sistema habilite suas funcionalidades e gere uma aplicação BFF.

2.2 Modelos de Usuários

Para este projeto, seus usuários foram modelados por meio de personas, de acordo com a definição de Courage e Baxter (2005), abordando os seguintes tópicos: identidade, status, objetivos, habilidades e tarefas. As personas exemplificadas nas Tabelas 1, 2 e 3 foram criadas com base em conversas com pessoas desenvolvedoras que demonstraram interesse na proposta do sistema. As características delas foram mescladas com personas fictícias a fim de exemplificar a abrangência do público alvo e de possíveis usuários deste sistema.




 <p>Fonte: freepik. Imagem livre de direitos autorais</p>	<p>Camila Alves, 30 anos, desenvolvedora <i>back-end</i>. Ela trabalha desenvolvendo e dando manutenção em microsserviços de uma empresa de serviços digitais presente em diversas plataformas.</p>
<p>Status</p>	<p>Primária</p>
<p>Objetivos</p>	<p>Aprimorar seus conhecimentos sobre sistemas distribuídos, escalabilidade e arquitetura de software para contribuir mais em seu trabalho e assumir uma posição de liderança técnica.</p>
<p>Habilidades</p>	<p>Construção de APIs (<i>Application Programming Interface</i>), REST (<i>Representational State Transfer</i>), WebSockets, Kafka, GraphQL, bancos de dados, monitoramento de sistemas.</p>
<p>Tarefas</p>	<p>Desenvolver novas funcionalidades para as aplicações já existentes em seu trabalho; revisão e refatoração de códigos; programação em pares; auxiliar novos membros da equipe; sugerir soluções e melhorias para as aplicações do seu time.</p>

Tabela 1. Primeira persona identificada.

 <p>Fonte: freepik. Imagem livre de direitos autorais</p>	<p>Murilo Santos, 28 anos, desenvolvedor <i>front-end</i>. Ele trabalha com desenvolvimento <i>front-end</i> há 10 anos. Se formou em um curso técnico de TI enquanto cursava o ensino médio e, em seguida, ingressou em um curso de graduação em Sistemas de Informação.</p>
<p>Status</p>	<p>Primário</p>
<p>Objetivos</p>	<p>Encontrar e/ou criar ferramentas para facilitar o desenvolvimento <i>front-end</i>; descobrir novas</p>

	maneiras de resolver problemas comuns no desenvolvimento Web; aprender mais sobre sistemas distribuídos e seus impactos em aplicações Web complexas.
Habilidades	Construção de aplicações <i>front-end</i> ; HTML (<i>HyperText Markup Language</i>); CSS (<i>Cascading Style Sheet</i>); JavaScript; integração com APIs REST e GraphQL; otimização de aplicações Web.
Tarefas	Liderança técnica de <i>front-end</i> Web; revisão e refatoração de código; mentoria de novos colaboradores; pesquisa de novas tecnologias e ferramentas para serem integradas ao processo de desenvolvimento das equipes.

Tabela 2. Segunda persona identificada.

 <p>Fonte: freepik. Imagem livre de direitos autorais</p>	Juliana Ferreira, 23 anos, desenvolvedora <i>mobile</i> . Recém-formada em um curso técnico de programação e desenvolvimento de aplicações móveis. Atua no mercado de trabalho há 3 anos fazendo estágios remunerados, mas foi contratada há pouco mais de 8 meses pela empresa atual.
Status	Secundária
Objetivos	Expandir e aprofundar seus conhecimentos sobre desenvolvimento <i>mobile</i> , Interface de Usuário/Experiência de Usuário (UI/UX, do inglês User Interface/User Experience), consumo de APIs (<i>Application Programming Interface</i>) com REST (<i>Representational State Transfer</i>) e GraphQL.
Habilidades	Desenvolvimento de aplicativos para multiplataformas com React Native e Flutter; desenvolvimento Web com React JS; design e prototipação de interfaces com Figma.

Tarefas	Implementar novas funcionalidades e aprimorar a usabilidade do aplicativo da empresa em que trabalha, disponível para Android e iOS. Uma nova versão deve ser lançada semanalmente com correções e/ou novidades.
---------	--

Tabela 3. Terceira persona identificada.

2.3 Modelo de Casos de Uso e Histórias de Usuários

Nesta seção estão representadas as histórias de usuário definidas para o sistema e na Figura 1 estão ilustrados os casos de uso que representam as ações dos usuários durante a sua utilização. Ambos são a materialização do escopo da aplicação. Para fins de organização, as histórias se encontram numeradas de 1 a 10, rotuladas com a letra H (de história), enquanto os casos de uso se encontram numerados de 1 a 8, rotulados com a letra C (de caso). Em ambos, os números não significam ordem de prioridade e/ou de importância.

1. H02: Camila, como desenvolvedora *back-end*, deseja abstrair o versionamentos das APIs para os clientes;
2. H03: Pedro, como desenvolvedor *front-end*, deseja abstrair as fontes de dados da aplicação;
3. H04: Murilo, como desenvolvedor *full stack*, deseja uma maneira simples de agregar os dados de diversas fontes para usos específicos do sistema;
4. H05: Paulo, como desenvolvedor *mobile*, deseja uma maneira simples de consumir dados por demanda para um aplicativo;
5. H06: Juliana, como desenvolvedora *back-end*, deseja uma maneira simples de criar uma API GraphQL;
6. H07: Henrique, como arquiteto de software, deseja uma maneira simples de criar aplicações BFFs para gerenciar o fluxo de dados entre microserviços e aplicações clientes;
7. H08: Daniel, como desenvolvedor, deseja automatizar e padronizar a criação de aplicações para o mesmo propósito, mas com diferentes contextos;
8. H09: Giovana, como desenvolvedora, deseja definir novos recursos em uma API apenas especificando os dados;
9. H10: Walter, como desenvolvedor *back-end*, deseja gerenciar e testar APIs de forma visual, utilizando interfaces dedicadas.

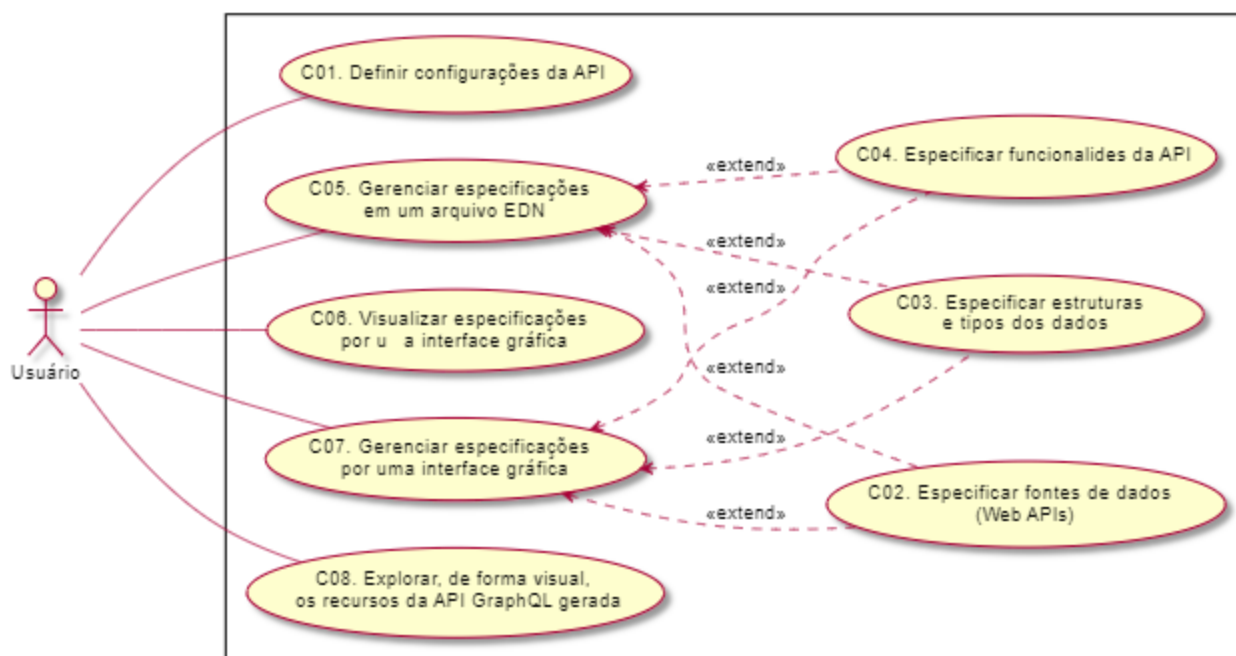


Figura 1. Diagrama de casos de uso do sistema.

2.4 Diagrama de Sequência do Sistema e Contrato de Operações

Nesta seção, são apresentados os diagramas de sequência do sistema com seus respectivos contratos de operações, baseados no diagrama de casos de uso apresentado anteriormente. Na Figura 2 está representado o fluxo de criação de componentes, que são todas as entidades do sistema gerenciadas pelo mesmo. É necessário apenas especificar os detalhes de cada uma. O gerenciamento desses componentes é padronizado para todos os diferentes tipos, mudando apenas os atributos que cada um possui. Todo o gerenciamento ocorre por meio da interface gráfica do sistema e as informações são persistidas em um arquivo EDN. Também há a possibilidade de gerenciar esses componentes diretamente pelo arquivo EDN.

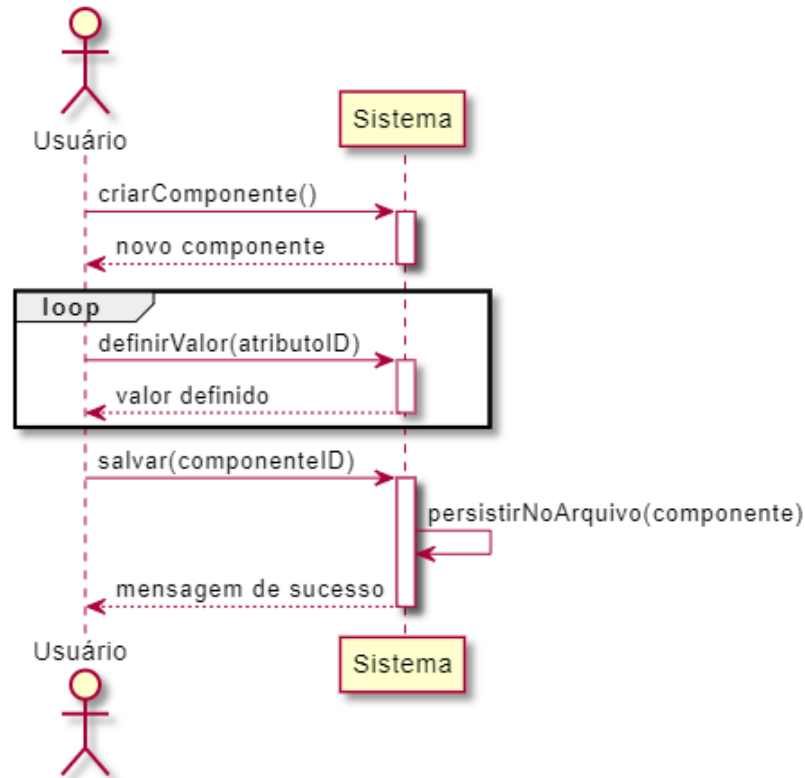


Figura 2. Diagrama de sequência do sistema para criação de componentes.

Na Figura 3 está representado o fluxo de listagem de componentes que ocorre por meio do sistema, carregando os dados do arquivo EDN para serem utilizados pela interface gráfica e pela aplicação gerada.

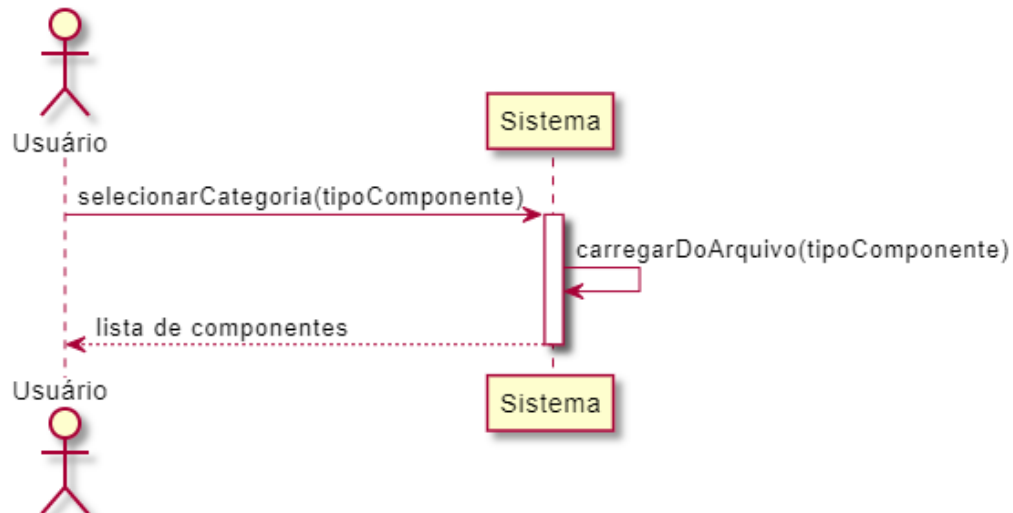


Figura 3. Diagrama de sequência do sistema para listagem de componentes.

Na Figura 4 está representado o fluxo de alteração de componentes que ocorre por meio do sistema, alterando os dados carregados do arquivo EDN e persistindo as novas informações.

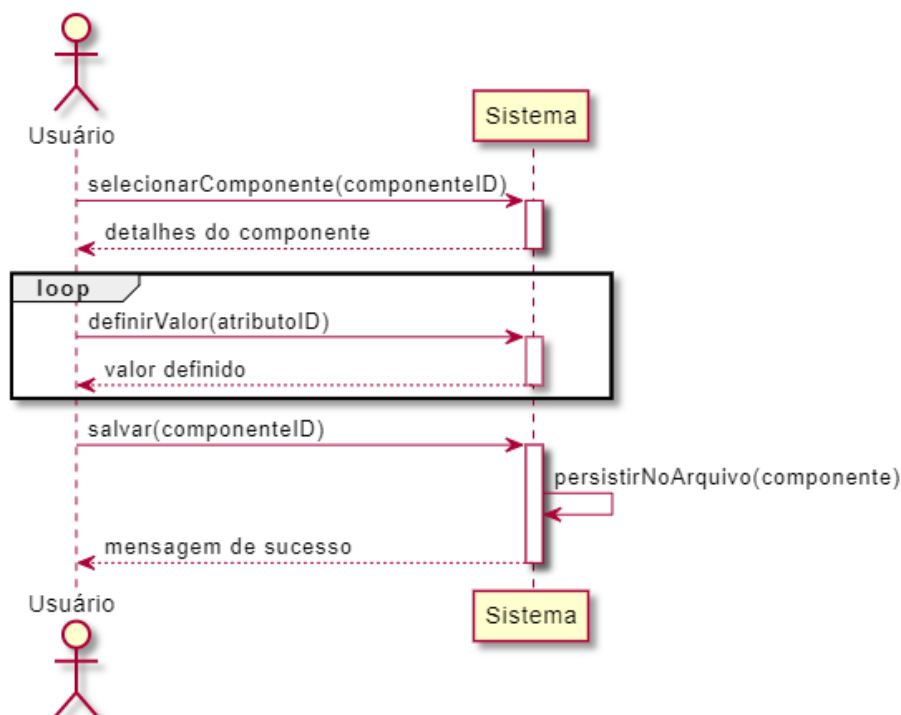


Figura 4. Diagrama de sequência do sistema para alteração de componentes.

Na Figura 5 está representado o fluxo de exclusão de componentes que ocorre por meio do sistema, removendo uma parte dos dados carregados do arquivo EDN e persistindo as informações restantes.

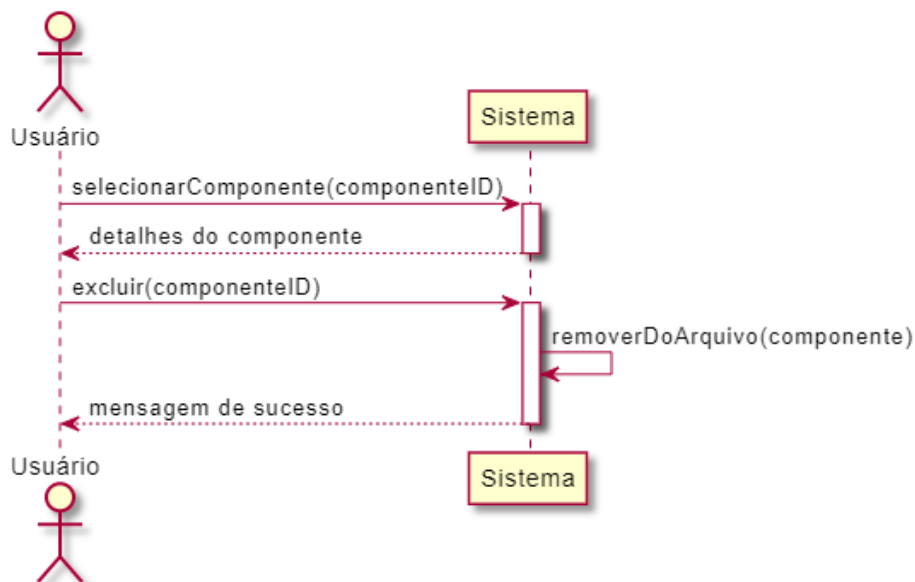


Figura 5. Diagrama de sequência do sistema para exclusão de componentes.

Na Figura 6 está representado o fluxo de gerenciamento de configurações que ocorre por meio do sistema. Os dados são carregados e atualizados no arquivo EDN e também podem ser gerenciados diretamente por ele. As configurações especificadas são referentes à aplicação que será gerada.

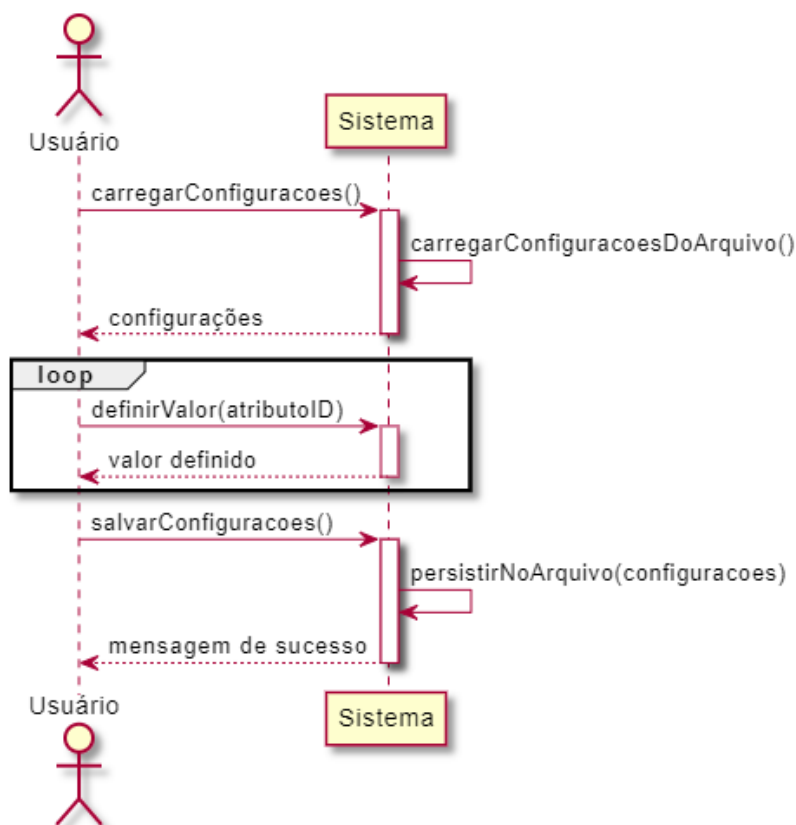


Figura 6. Diagrama de sequência do sistema para definição de configurações.

3. Modelos de Projeto

Nesta seção são apresentados os modelos de projeto do sistema proposto. A seção é composta pelo diagrama de Classes (Seção 3.1), diagramas de sequência (Seção 3.2), diagramas de comunicação (Seção 3.3), arquitetura lógica (Seção 3.4), diagramas de estado (Seção 3.5) e pelos diagramas de componentes e implantação (Seção 3.6).

3.1 Diagrama de Classes

Diagrama de classes do sistema com todas as entidades que são gerenciadas pelo mesmo. Os componentes do sistema estão representados na Figura 7 e são eles: Type, Input, Interface, Enum, Query, Mutation, API. Alguns desses componentes possuem estruturas compostas e por isso surgem as entidades Arg, Field e Value.

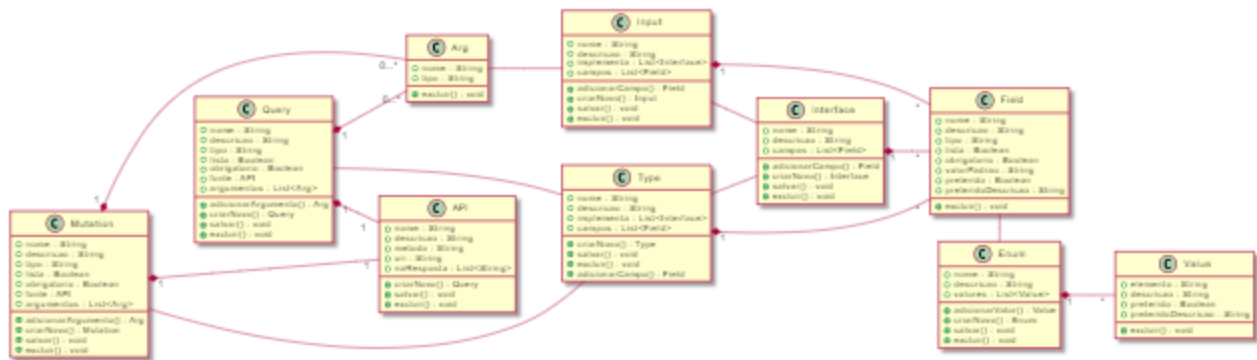


Figura 7. Diagrama de classes do sistema.

3.2 Diagramas de Sequência

Nesta seção estão representados os diagramas de sequência, baseados nos casos de uso do sistema. As interações ocorrem, inicialmente, na Interface Gráfica, fazendo o papel de fronteira do sistema.

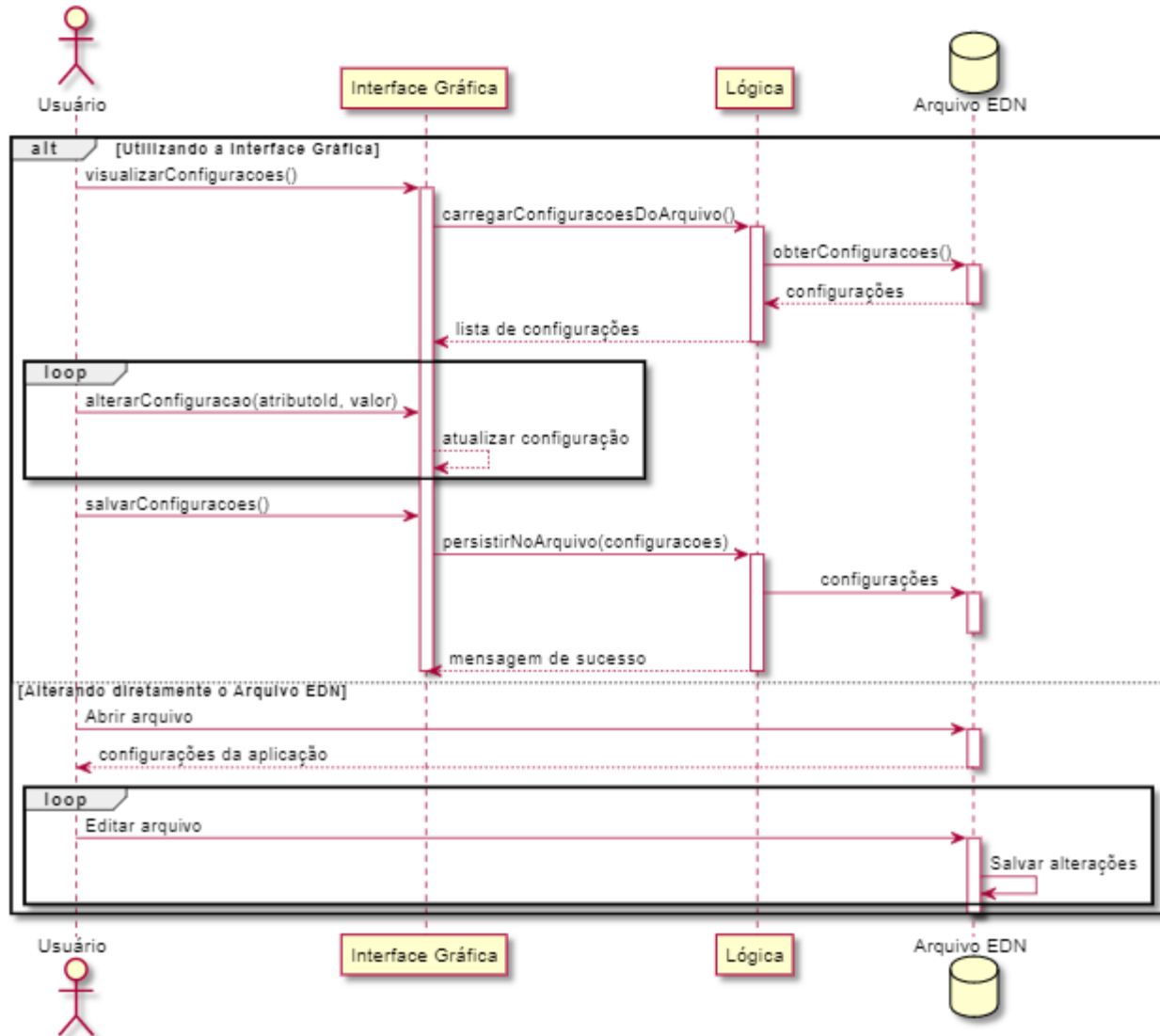


Figura 8. Diagrama de sequência do sistema para o caso de uso C01.

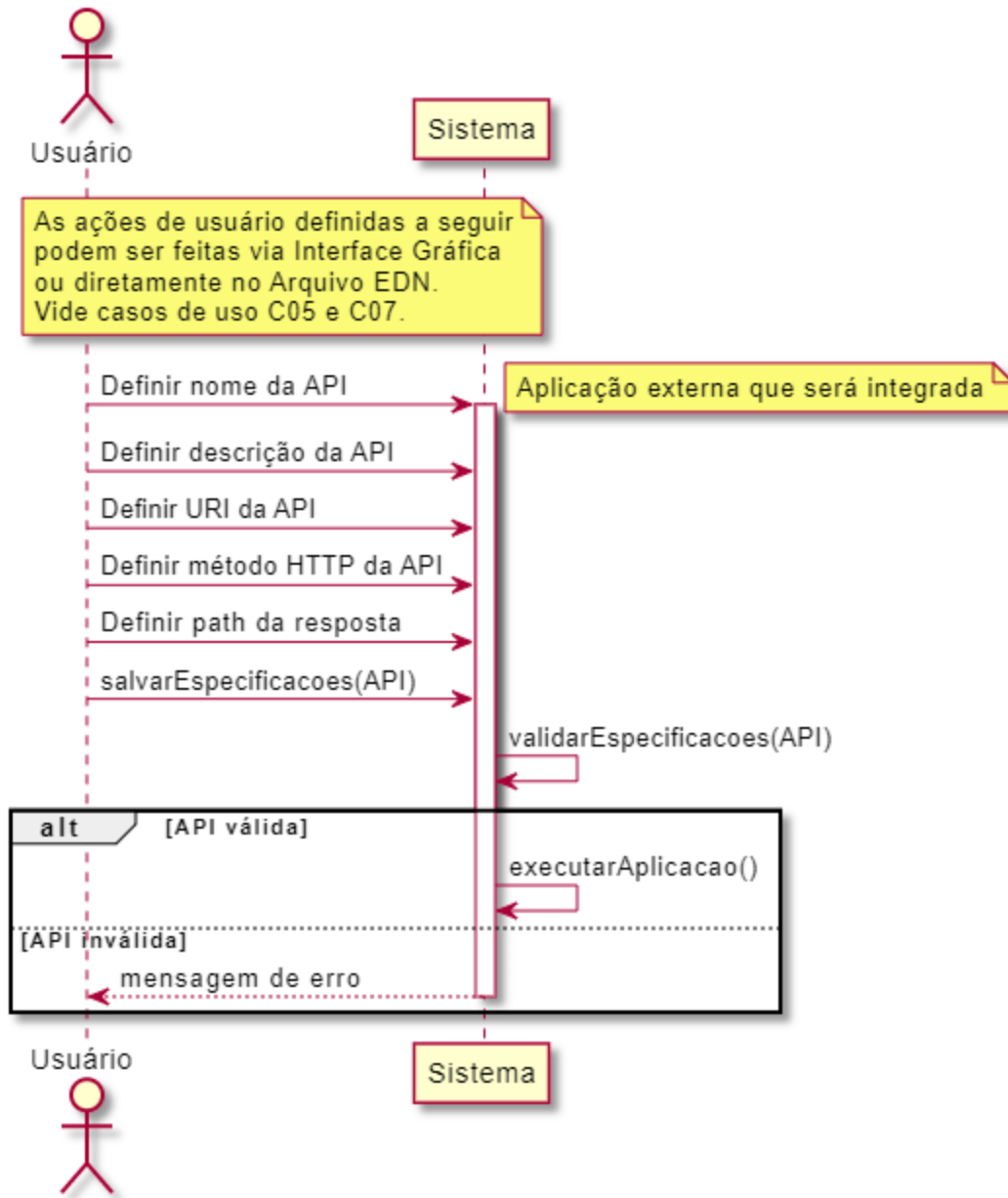


Figura 9. Diagrama de sequência do sistema para o caso de uso C02.

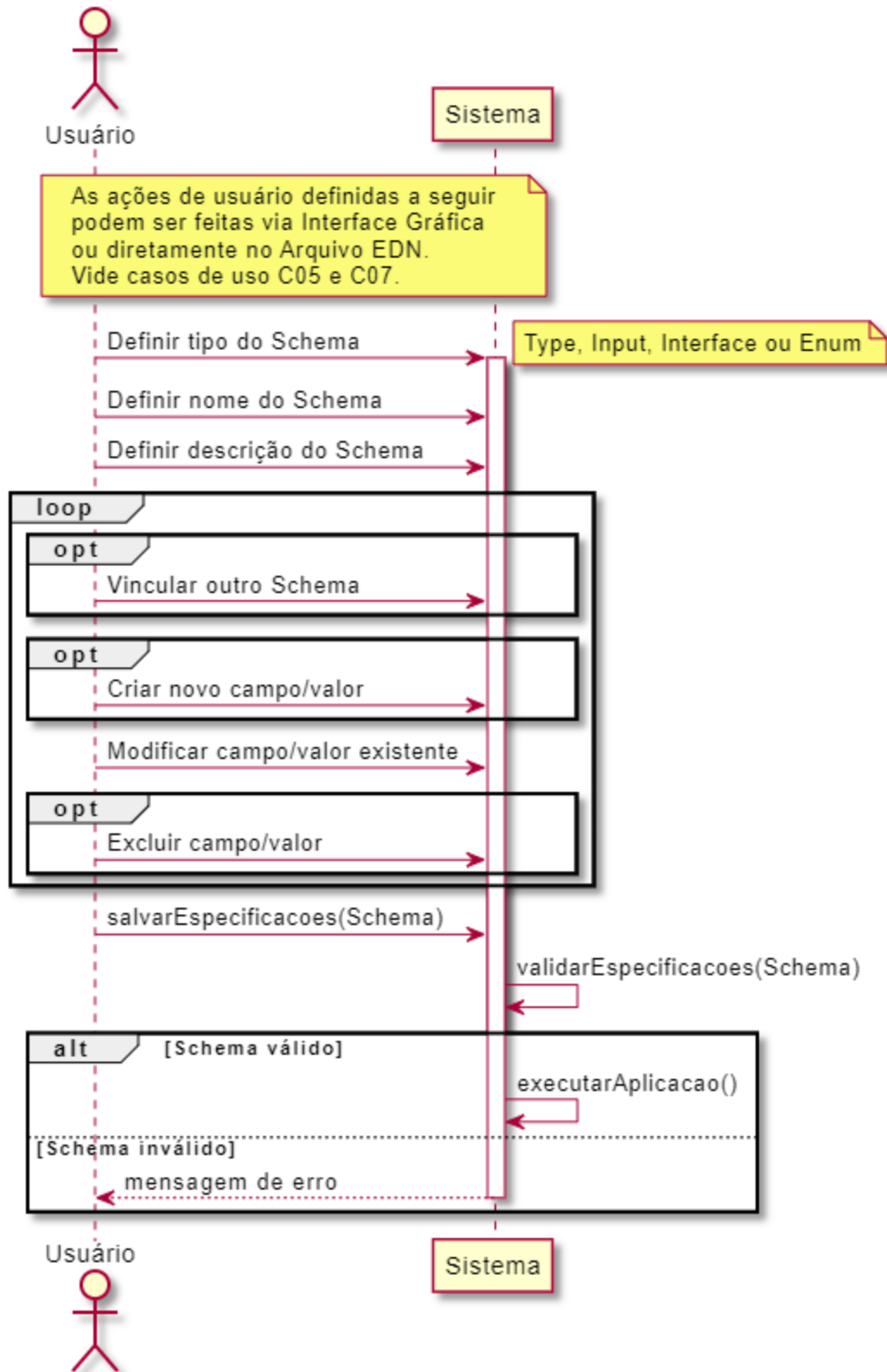


Figura 10. Diagrama de sequência do sistema para o caso de uso C03.

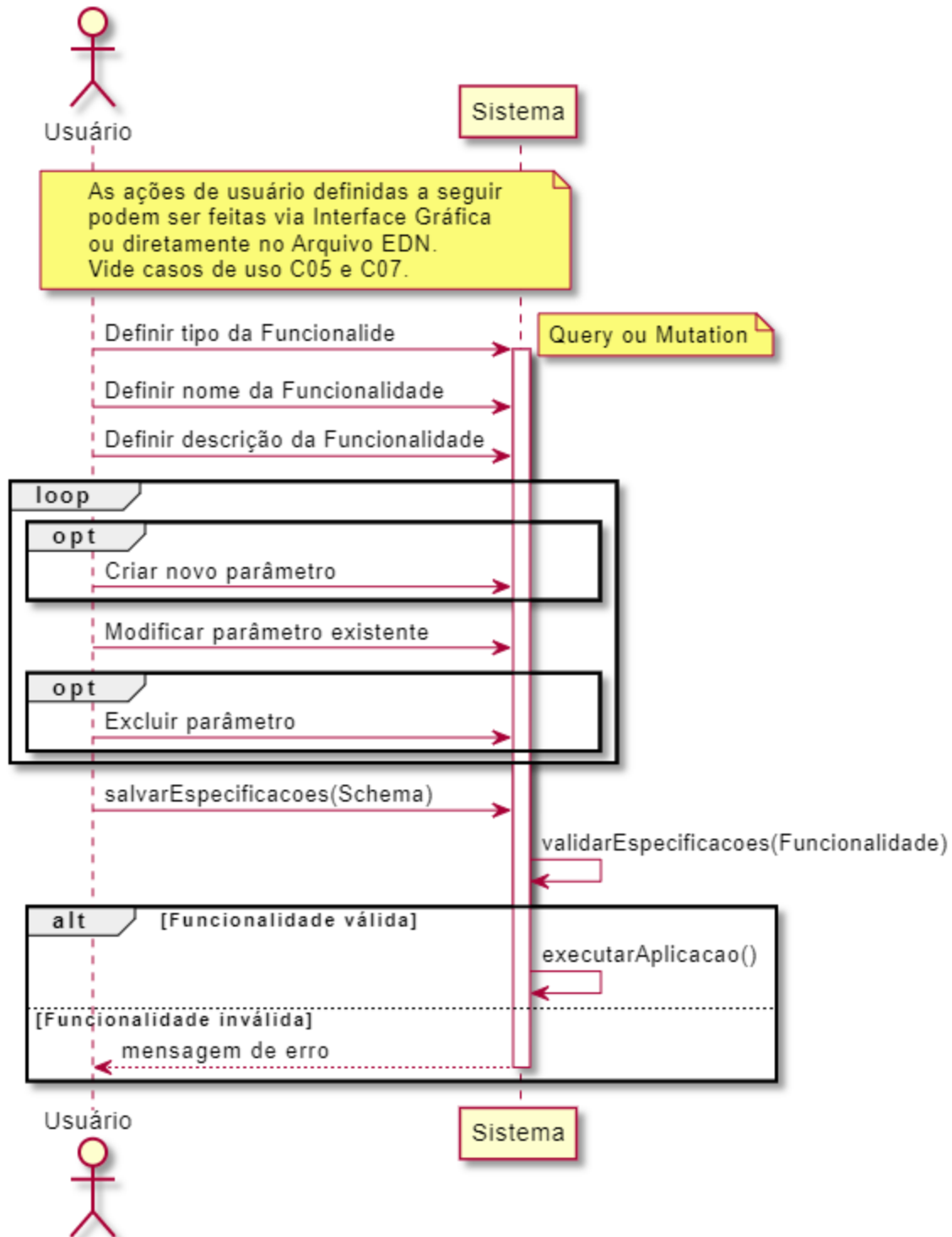


Figura 11. Diagrama de sequência do sistema para o caso de uso C04.

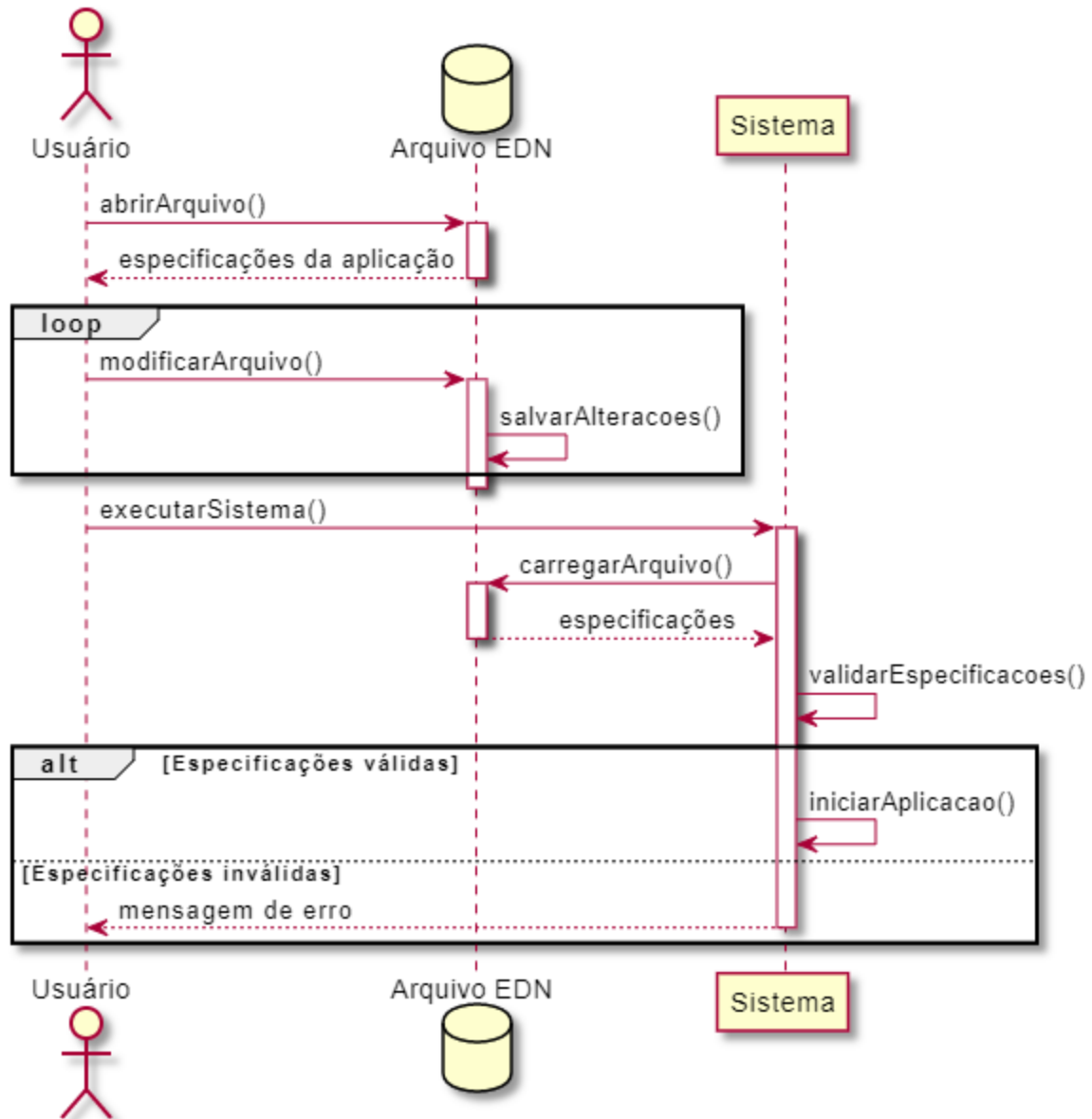


Figura 12. Diagrama de sequência do sistema para o caso de uso C05.

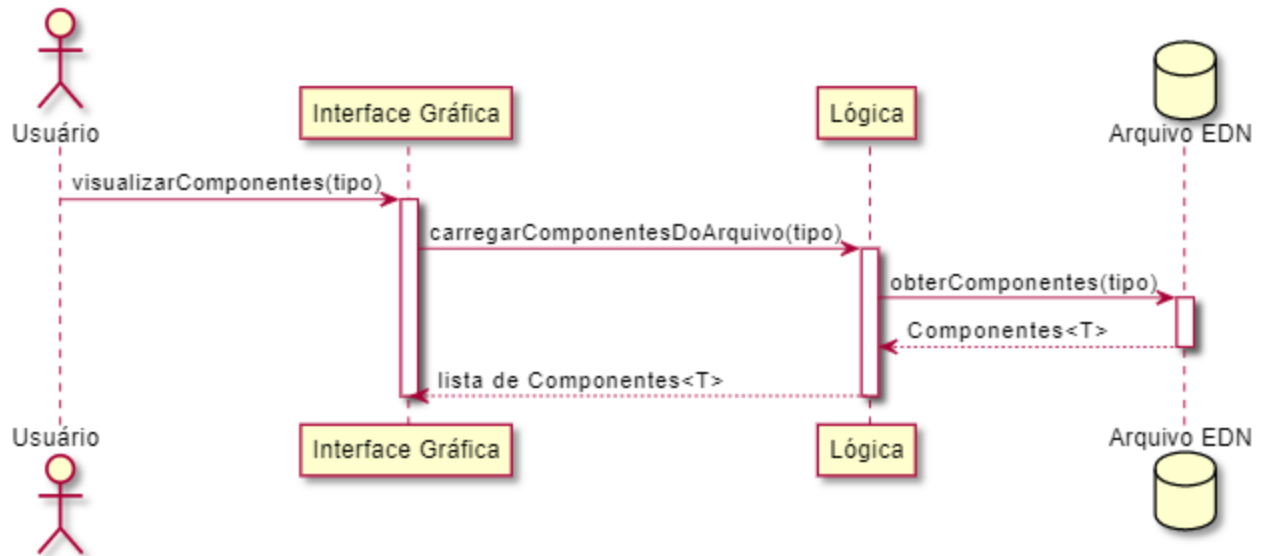


Figura 13. Diagrama de sequência do sistema para o caso de uso C06.

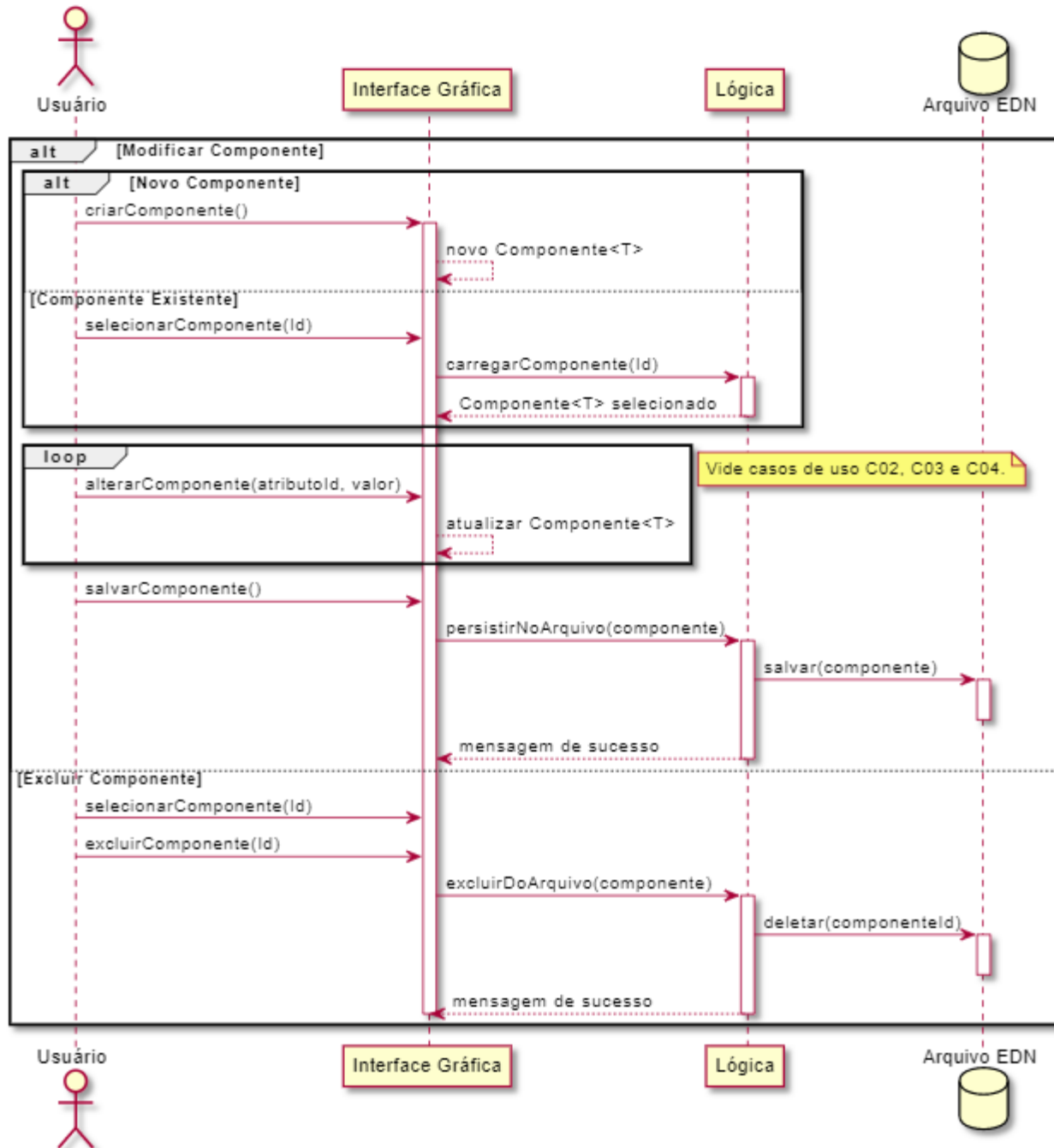


Figura 14. Diagrama de sequência do sistema para o caso de uso C07.

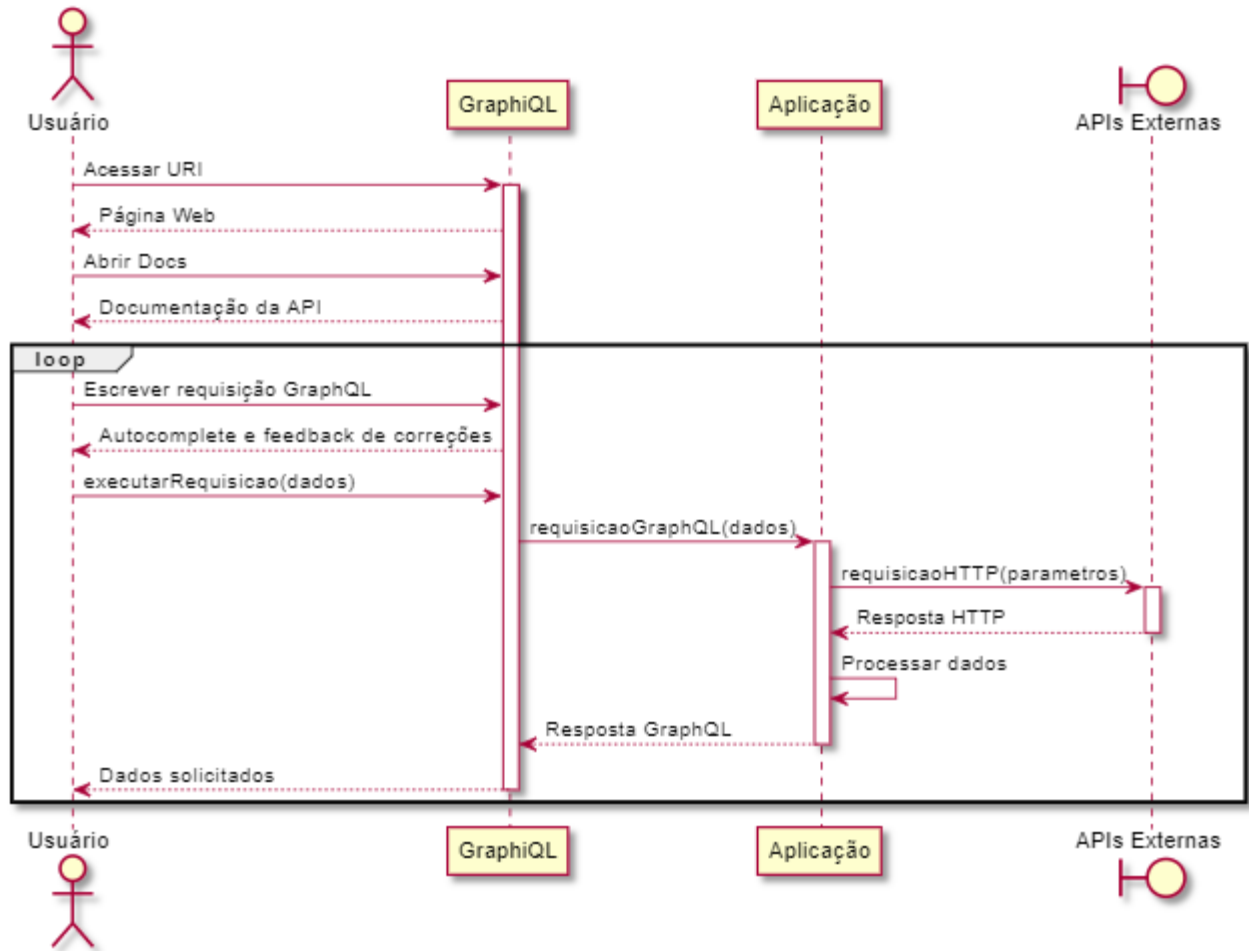


Figura 15. Diagrama de sequência do sistema para o caso de uso C08.

3.3 Diagramas de Comunicação

Nesta seção estão representados os diagramas de comunicação, baseados nos diagramas de sequência apresentados na seção anterior. As mensagens são, inicialmente, encaminhadas para a Interface Gráfica, fazendo o papel de fronteira do sistema.

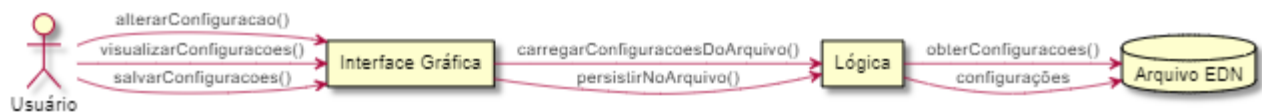


Figura 16. Diagrama de comunicação do sistema para o caso de uso C01.

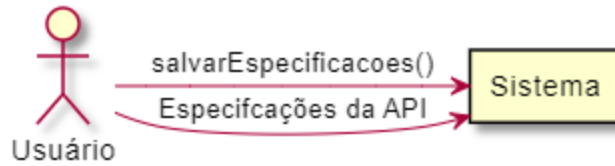


Figura 17. Diagrama de comunicação do sistema para o caso de uso C02.

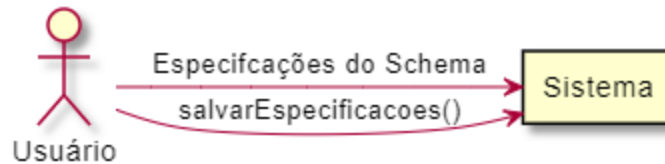


Figura 18. Diagrama de comunicação do sistema para o caso de uso C03.

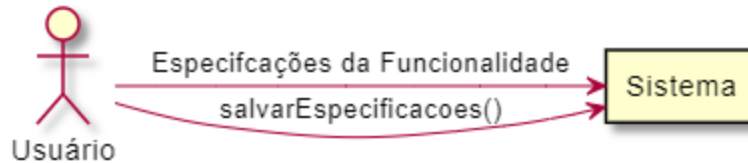


Figura 19. Diagrama de comunicação do sistema para o caso de uso C04.

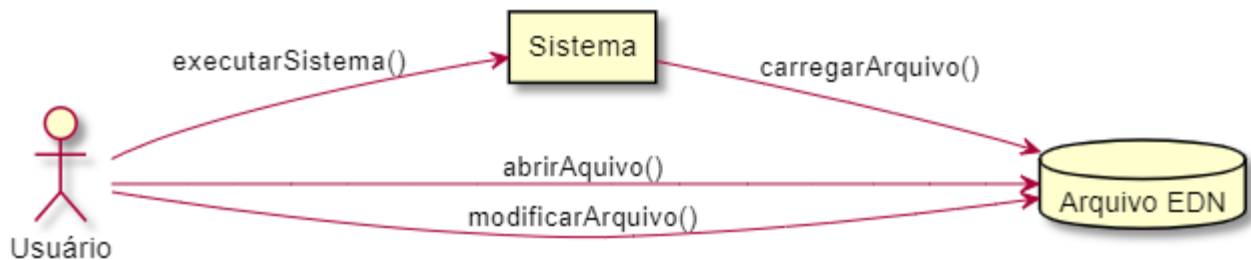


Figura 20. Diagrama de comunicação do sistema para o caso de uso C05.



Figura 21. Diagrama de sequência do sistema para o caso de uso C06.

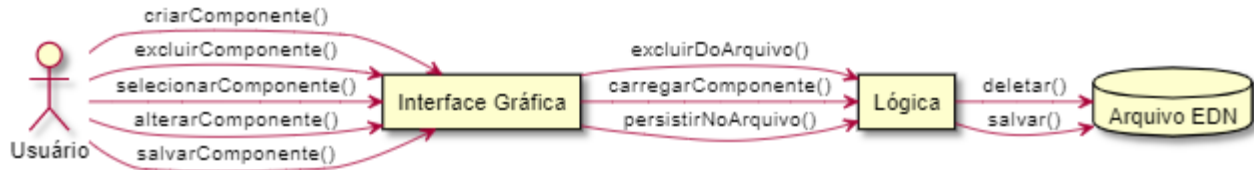


Figura 22. Diagrama de comunicação do sistema para o caso de uso C07.

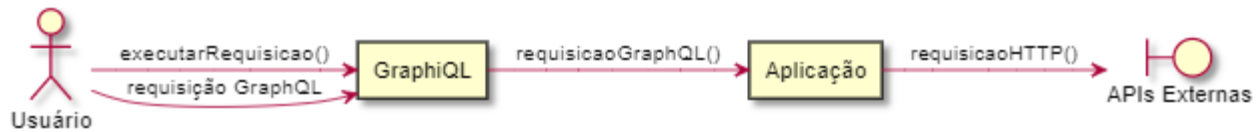


Figura 23. Diagrama de comunicação do sistema para o caso de uso C08.

3.4 Arquitetura

Pode ser descrita com um diagrama apropriado da UML ou C4 Model

3.5 Diagramas de Estados

Diagramas de estados do sistema.

3.6 Diagrama de Componentes e Implantação.

Diagramas de componentes do sistema. Diagrama de implantação mostrando onde os componentes estarão alocados para a execução.

4. Projeto de Interface com Usuário

A seguir estão representados alguns *wireframes* (protótipos) das interfaces gráficas do sistema que serão construídas para a interação visual dos usuários com o sistema proposto. Na Figura 2 está representada a tela principal do sistema, que será a tela de gerenciamento de *schemas*, na qual serão listados, criados, editados e excluídos todos os modelos de dados que serão tratados pelo sistema. Como todos os outros elementos do sistema a serem definidos pelo usuário, os *schemas* são apenas especificações de dados, com seus atributos e seus respectivos valores. Essa tela corresponde aos casos de uso C03, C06 e C07.

Header

Schemas

APIs

Queries

Mutations

Settings

Enums	Interfaces	Types	Inputs
Type 1 >			
Type 2 >			
Type 3 >			
Type 4 >			
Type 5 >			
Type 6 >			
...			

Create new

Delete

Name:	Input	Kind	v
Description:	Input		
Implements: 1 Values:	Input		
Fields:			Add
Name:	Input		
Description:	Input		
Type:	Select v	<input checked="" type="radio"/> List	<input type="radio"/> Required
Default value:	Input		
Deprecated? <input type="checkbox"/>	Input	Delete	
Name:	Input		
Description:	Input		
Type:	Select v	<input checked="" type="radio"/> List	<input type="radio"/> Required
Default value:	Input		
<div style="border: 1px solid black; padding: 5px 20px; border-radius: 5px; display: inline-block;">Save</div>			

Figura 2. Wireframe da aba de *schemas* do sistema.

Na Figura 3 está representada a tela de gerenciamento de *APIs*, na qual serão listadas, criadas, editadas e excluídas todas as fontes de dados externas que serão tratados pelo sistema. Neste caso, cada fonte de dados é um *endpoint* de uma outra API que será consumida para obter e/ou modificar dados, a depender do funcionamento de cada recurso. Nessa tela são definidas apenas as informações necessárias para realizar uma determinada requisição, a especificação dos dados recebidos e enviados, assim como os parâmetros suportados, é feita diretamente nas abas de *Queries* e *Mutations*. Essa tela corresponde aos casos de uso C02, C06 e C07.

Header													
<div>Schemas APIs Queries Mutations Settings</div>													
<div>Endpoint 1 ></div> <div>Endpoint 2 ></div> <div>Endpoint 3 ></div> <div>Endpoint 4 ></div> <div>Endpoint 5 ></div> <div>Endpoint 6 ></div> <div>...</div>	<div>Create new Delete</div> <table border="1"><tr><td>Name:</td><td>Input</td><td>Method ></td></tr><tr><td>Description:</td><td colspan="2">Input</td></tr><tr><td>URI:</td><td colspan="2">Input</td></tr><tr><td>Into response:</td><td colspan="2">Input</td></tr></table> <div>Save</div>	Name:	Input	Method >	Description:	Input		URI:	Input		Into response:	Input	
Name:	Input	Method >											
Description:	Input												
URI:	Input												
Into response:	Input												

Figura 3. Wireframe da aba de *APIs* do sistema.

Na Figura 4 está representada a tela de gerenciamento de *Queries*, na qual serão listadas, criadas, editadas e excluídas todas as consultas de dados externas que serão tratados pelo sistema. Neste caso, cada consulta é um recurso da aplicação que será fornecido para obter dados de uma determinada *API*. Nessa tela são definidas apenas as informações necessárias para realizar uma determinada consulta, utilizando modelos de dados e uma *API* previamente definidos. Além disso, também é feita a especificação dos dados recebidos e enviados, assim como os parâmetros suportados. Essa tela corresponde aos casos de uso C04, C06 e C07.

Header

Schemas APIs Queries Mutations Settings

Query 1 >

Query 2 >

Query 3 >

Query 4 >

Query 5 >

Query 6 >

...

Create new Delete

Name: Input

Type: Select ☐ List ☐ Required

Source: Select

Args: Add

Name:	Select	V
Type:	Select	V Delete
Name:	Select	V
Type:	Select	V Delete
Name:	Select	V
Type:	Select	V Delete

Save

Figura 4. Wireframe da aba de *Queries* do sistema.

Na Figura 5 está representada a tela de gerenciamento de *Mutations* (que é igual à tela de gerenciamento de *Queries*), na qual serão listadas, criadas, editadas e excluídas todas as modificações de dados externos que serão tratados pelo sistema. Neste caso, cada modificação é um recurso da aplicação que será fornecido para enviar dados para uma determinada *API*. Nessa tela são definidas apenas as informações necessárias para realizar uma determinada modificação, utilizando modelos de dados e uma *API* previamente definidos. Além disso, também é feita a especificação dos dados enviados e recebidos, assim como os parâmetros suportados. Essa tela corresponde aos casos de uso C04, C06 e C07.

Header																																																							
<div style="display: flex; justify-content: space-between; margin-bottom: 10px;"> Schemas APIs Queries Mutations Settings </div> <div style="display: flex;"> <div style="flex: 1;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">Mutation 1</td><td style="text-align: right;">▼</td></tr> <tr><td style="text-align: center;">Mutation 2</td><td style="text-align: right;">▼</td></tr> <tr><td style="text-align: center;">Mutation 3</td><td style="text-align: right;">▼</td></tr> <tr><td style="text-align: center;">Mutation 4</td><td style="text-align: right;">▼</td></tr> <tr><td style="text-align: center;">Mutation 5</td><td style="text-align: right;">▼</td></tr> <tr><td style="text-align: center;">Mutation 6</td><td style="text-align: right;">▼</td></tr> <tr><td colspan="2" style="text-align: center;">...</td></tr> </table> </div> <div style="flex: 1; padding-left: 10px;"> <div style="display: flex; justify-content: space-between; margin-bottom: 10px;"> Create new Delete </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 20%;">Name:</td><td colspan="3">Input</td></tr> <tr> <td>Type:</td> <td>Select</td> <td>V</td> <td>● List ● Required</td> </tr> <tr> <td>Source:</td> <td colspan="3">Select V</td> </tr> <tr> <td>Args:</td> <td colspan="3" style="text-align: right;">Add</td> </tr> <tr> <td>Name:</td> <td colspan="3">Select V</td> </tr> <tr> <td>Type:</td> <td>Select</td> <td>V</td> <td>Delete</td> </tr> <tr> <td>Name:</td> <td colspan="3">Select V</td> </tr> <tr> <td>Type:</td> <td>Select</td> <td>V</td> <td>Delete</td> </tr> <tr> <td>Name:</td> <td colspan="3">Select V</td> </tr> <tr> <td>Type:</td> <td>Select</td> <td>V</td> <td>Delete</td> </tr> </table> <div style="text-align: center; margin-top: 10px;">Save</div> </div> </div>		Mutation 1	▼	Mutation 2	▼	Mutation 3	▼	Mutation 4	▼	Mutation 5	▼	Mutation 6	▼	...		Name:	Input			Type:	Select	V	● List ● Required	Source:	Select V			Args:	Add			Name:	Select V			Type:	Select	V	Delete	Name:	Select V			Type:	Select	V	Delete	Name:	Select V			Type:	Select	V	Delete
Mutation 1	▼																																																						
Mutation 2	▼																																																						
Mutation 3	▼																																																						
Mutation 4	▼																																																						
Mutation 5	▼																																																						
Mutation 6	▼																																																						
...																																																							
Name:	Input																																																						
Type:	Select	V	● List ● Required																																																				
Source:	Select V																																																						
Args:	Add																																																						
Name:	Select V																																																						
Type:	Select	V	Delete																																																				
Name:	Select V																																																						
Type:	Select	V	Delete																																																				
Name:	Select V																																																						
Type:	Select	V	Delete																																																				

Figura 5. Wireframe da aba de *Mutations* do sistema.

Na Figura 6 está representada a tela de *Settings*, na qual serão definidas todas as configurações de servidor necessárias para executar a aplicação nos ambientes desejados, expondo uma *API GraphQL* com os recursos definidos nas outras telas. Essa tela corresponde aos casos de uso C01, C06 e C07.

Header																			
<div style="display: flex; justify-content: space-between; margin-bottom: 10px;"> Schemas APIs Queries Mutations Settings </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">Name:</td> <td style="width: 30%;">Input</td> <td style="width: 20%;">Env:</td> <td style="width: 30%;">Select V</td> </tr> <tr> <td colspan="2">Description:</td> <td colspan="2">Host:</td> </tr> <tr> <td colspan="2">API Path:</td> <td colspan="2">Port:</td> </tr> <tr> <td colspan="2">IDE Path:</td> <td colspan="2">CORS:</td> </tr> </table> <div style="text-align: right; margin-top: 20px;"> <div style="border: 1px solid black; padding: 5px 10px;">Save</div> </div>				Name:	Input	Env:	Select V	Description:		Host:		API Path:		Port:		IDE Path:		CORS:	
Name:	Input	Env:	Select V																
Description:		Host:																	
API Path:		Port:																	
IDE Path:		CORS:																	

Figura 6. Wireframe da aba de *Settings* do sistema.

Na Figura 7 está representada a tela do GraphiQL (uma ferramenta para testes e exploração de APIs GraphQL), que estará incorporada no sistema como parte da aplicação gerada. Essa ferramenta é open source, geralmente disponibilizada com APIs públicas ou como complemento de frameworks e bibliotecas de GraphQL para auxiliar no desenvolvimento. Essa tela corresponde ao caso de uso C08.

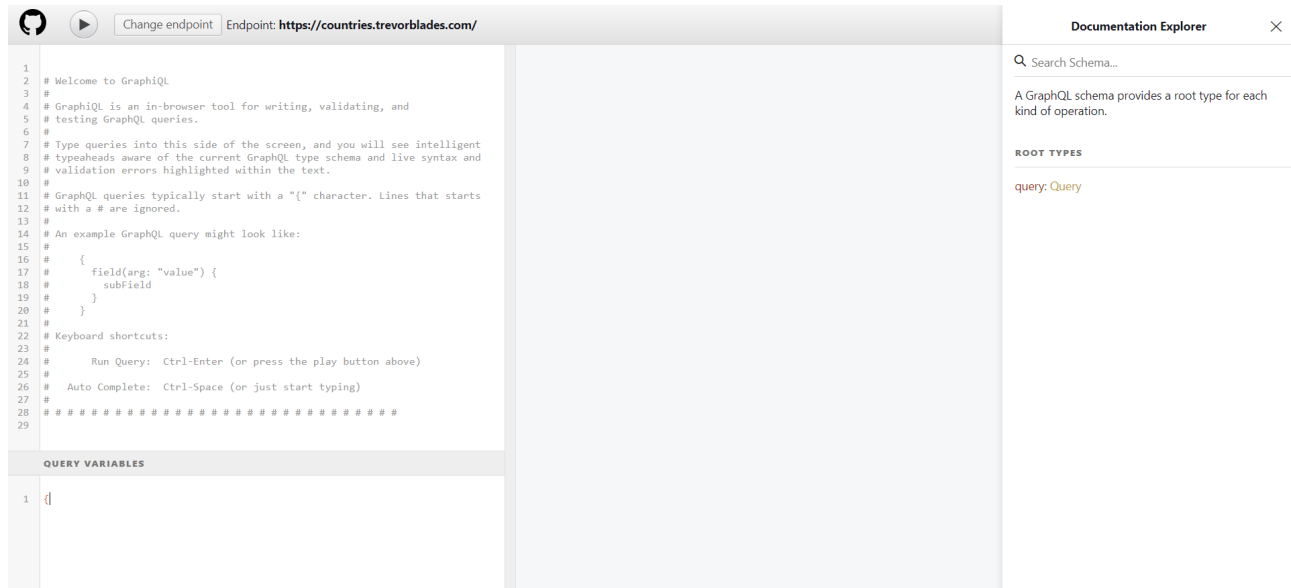


Figura 7. Interface da ferramenta GraphiQL incorporada no sistema.

5. Glossário e Modelos de Dados

Deve-se apresentar o glossário para o sistema. Também apresente esquemas de banco de dados e as estratégias de mapeamento entre as representações de objetos e não-objetos.

6. Casos de Teste

Uma descrição de casos de teste para validação do sistema.

7. Cronograma e Processo de Implementação

Uma descrição do cronograma para implementação do sistema e do processo que será seguido durante a implementação.