

Belo Horizonte, 2022

# Trabalho de Conclusão de Curso I

**Aluno**

**Marlon Henrique da Silva**

**Orientadores**

**José Laerte Xavier**

**Marco Rodrigo Costa**

**PUC Minas**  
**Engenharia de Software**

FERRAMENTA PARA ENGENHARIA DE SOFTWARE

# BFF Squared

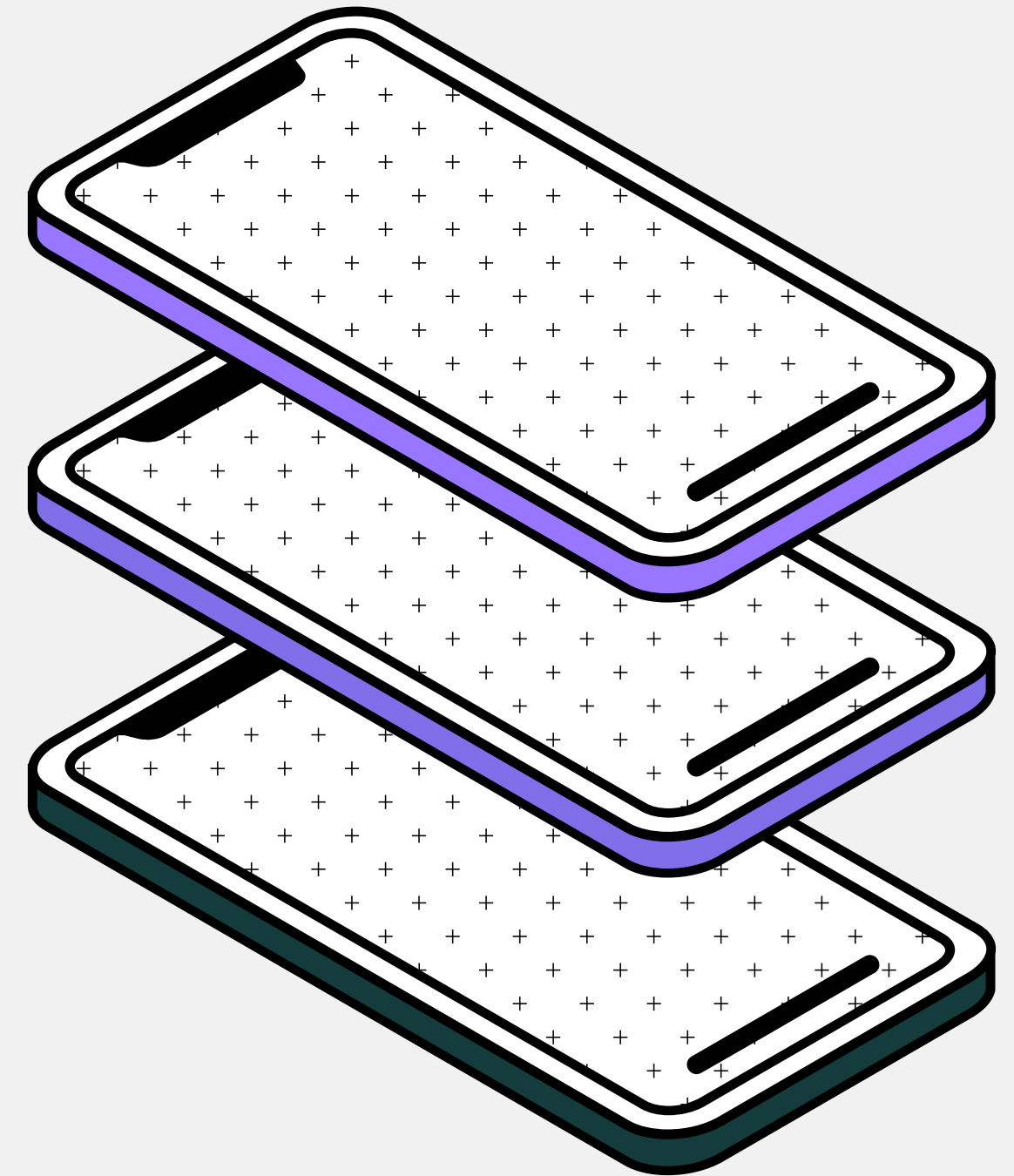
Um *framework* para a construção de BFFs



BFF

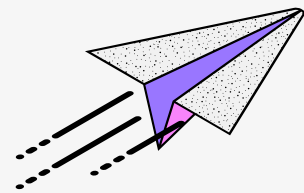
# *Back-end for Front-end*

Um padrão de projeto criado pensando na experiência do desenvolvedor e do usuário.



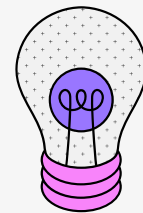
# Objetivo do Software

Automatizar a construção de aplicações BFF baseadas em especificações de dados para integrar diferentes aplicações de *back-end*.



## GraphQL

Linguagem de consulta e *runtime* para APIs



## EDN

*Extensible Data Notation*  
(Notação de Dados Extensível)



## JVM

*Java Virtual Machine*  
(Máquina Virtual Java)

# Dentro do Escopo

*Queries e Mutations GraphQL*

Requisições HTTP

Padrão REST

# Fora do Escopo

*Subscriptions GraphQL*

Mensageria

*WebSockets*

# Principais necessidades e funcionalidades

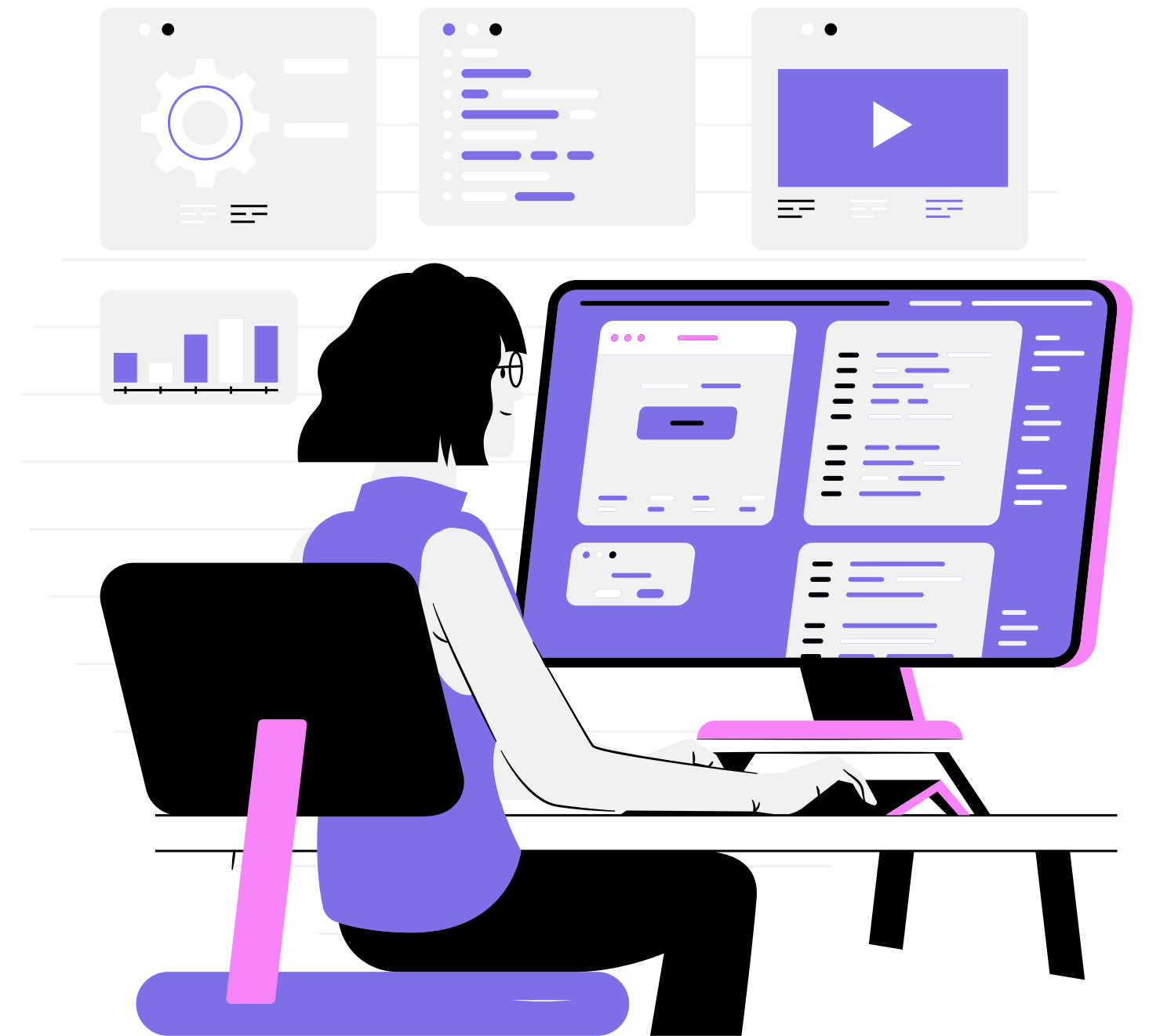
**Integração** entre diversos microsserviços e diferentes aplicações clientes

Controle e **flexibilidade** sobre os dados das requisições

**Automação** para criação de Web APIs

Facilitação para **implementação** de GraphQL

**Facilitação** para implementação de uma aplicação BFF



# Atores

## Personas



## Pessoas desenvolvedoras de software

Conhecimento em **back-end**

Conhecimento em **front-end**

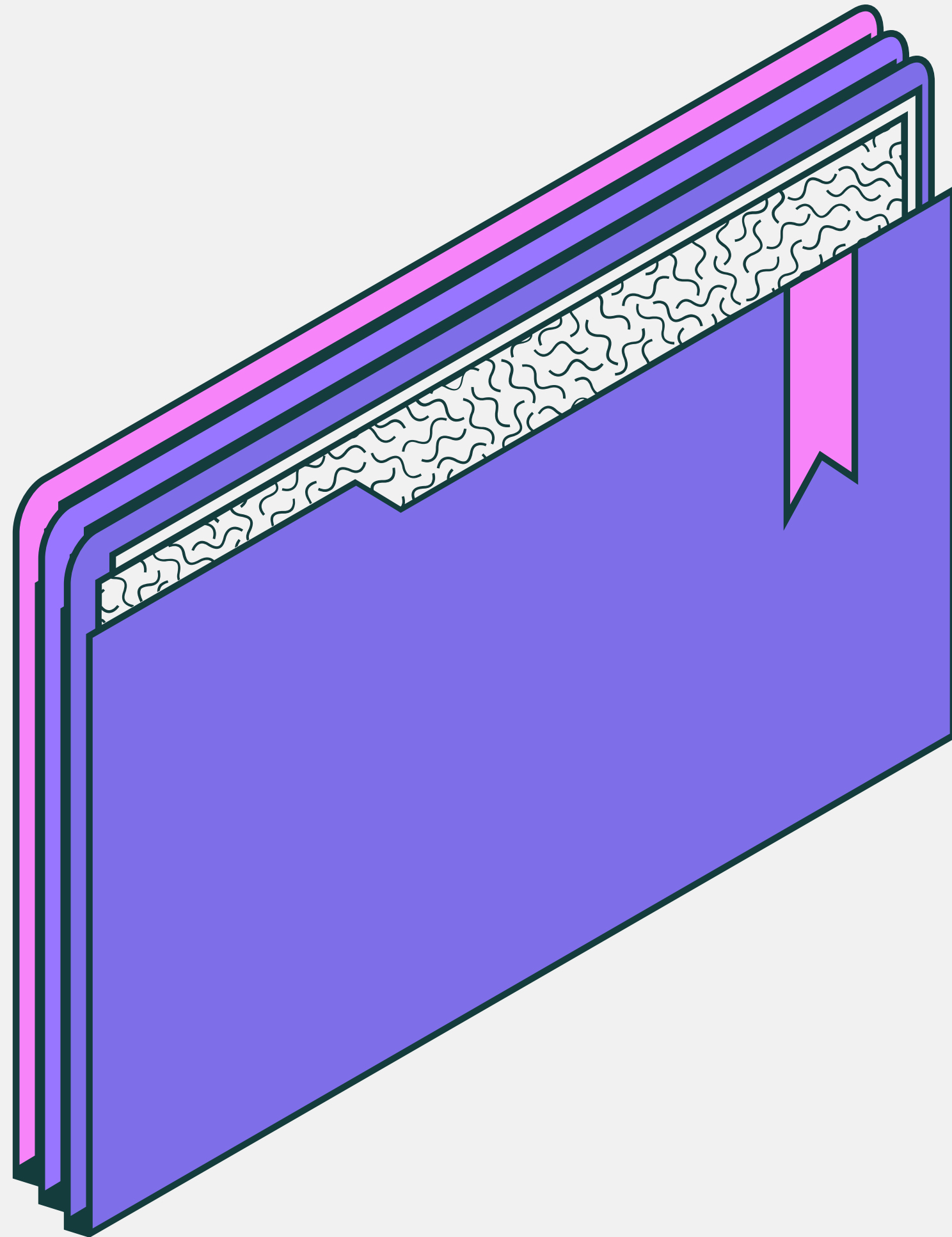
Conhecimento em **aplicações móveis**

Conhecimento em **sistemas distribuídos**

Diferentes níveis de experiência  
e **senioridade**

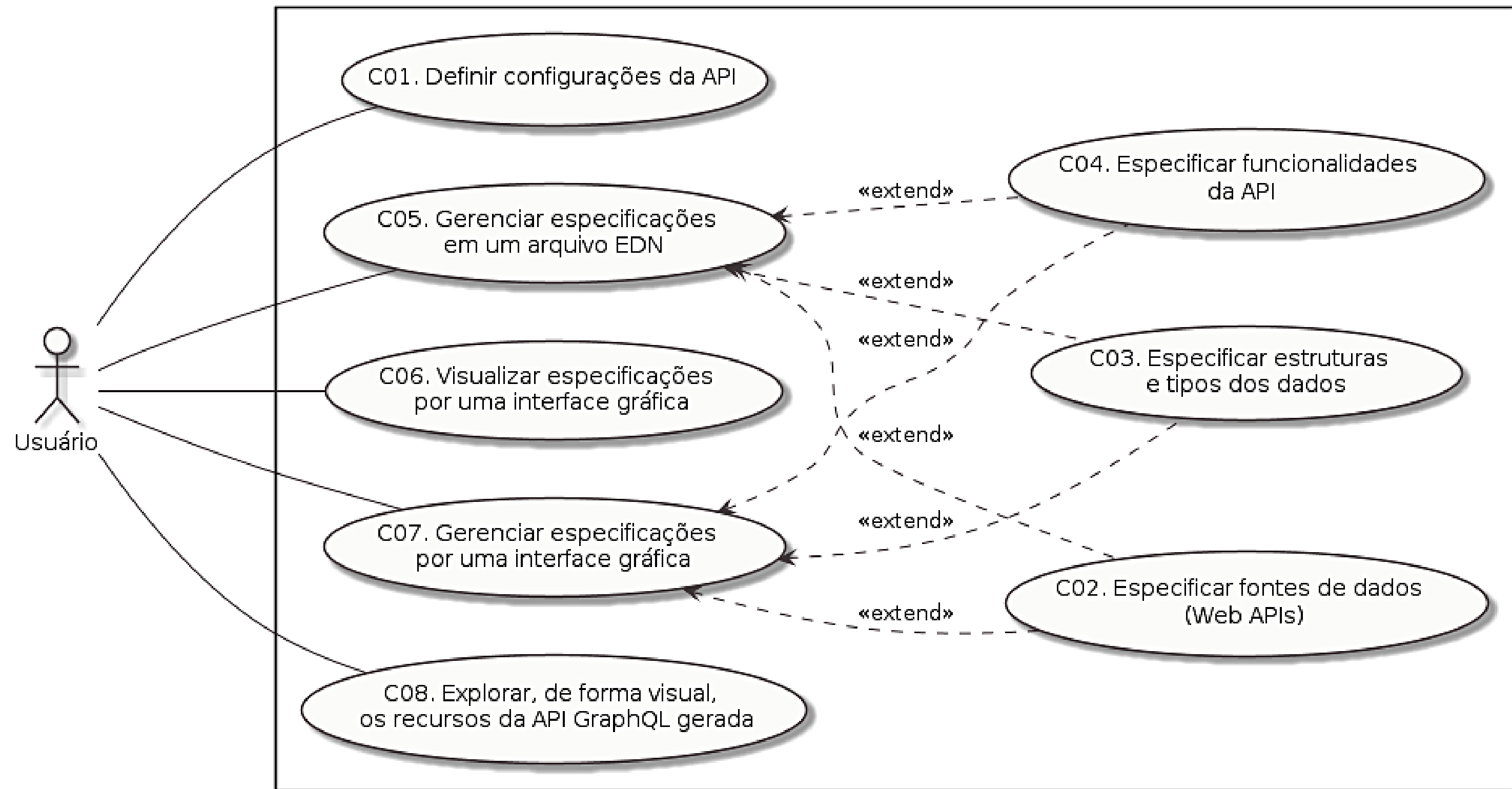
# Modelagem do Sistema

Diagramas

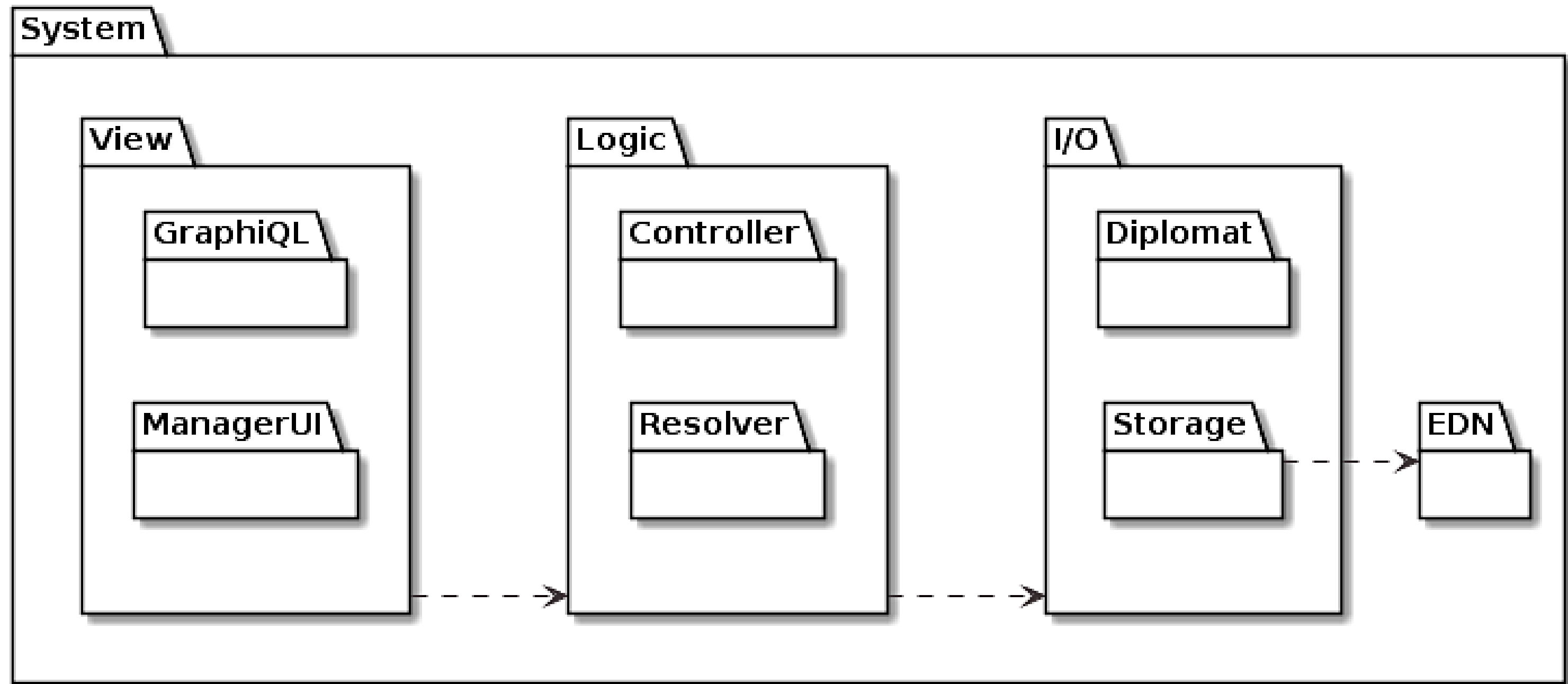




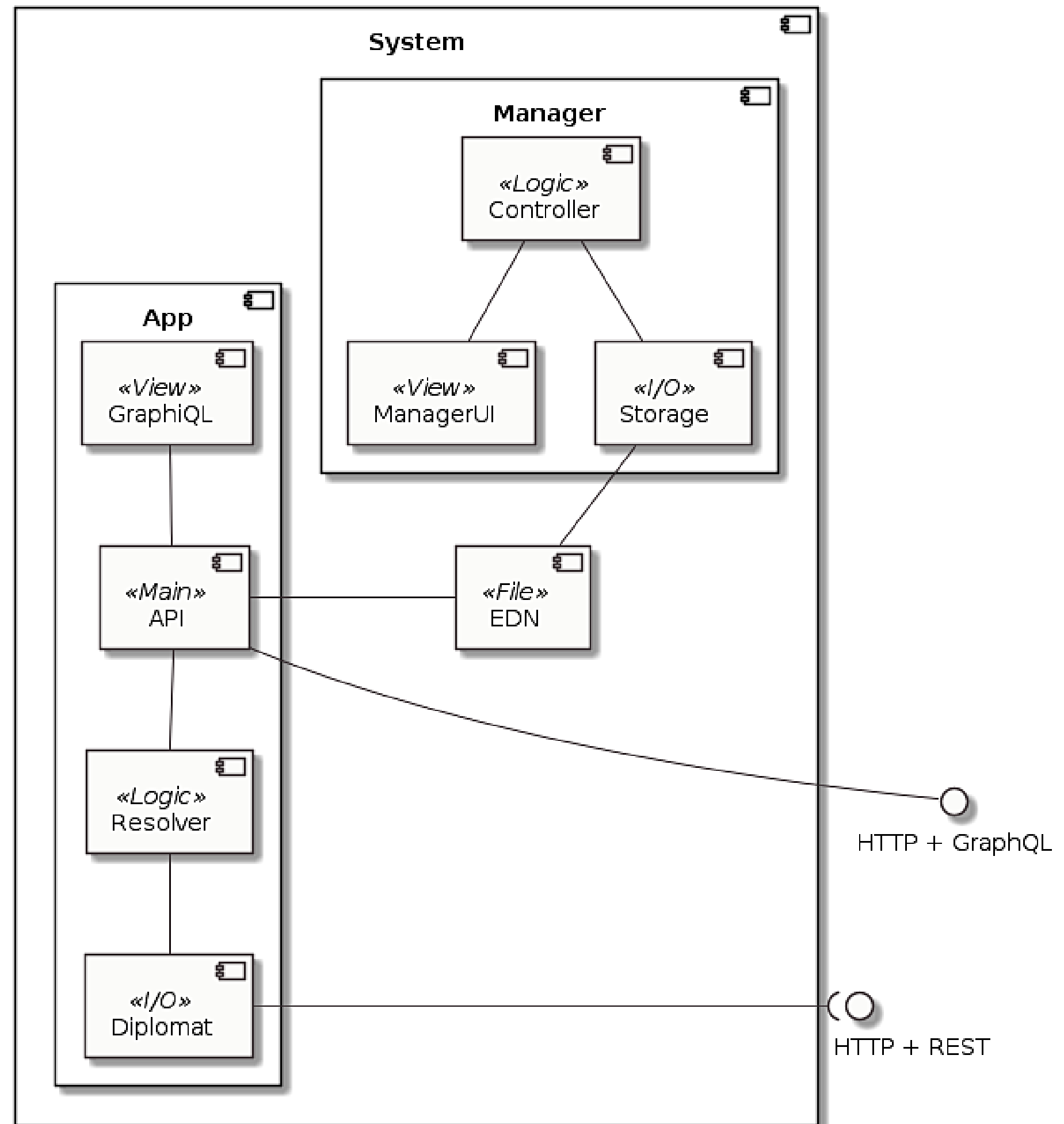
# Diagrama de Caso de Uso



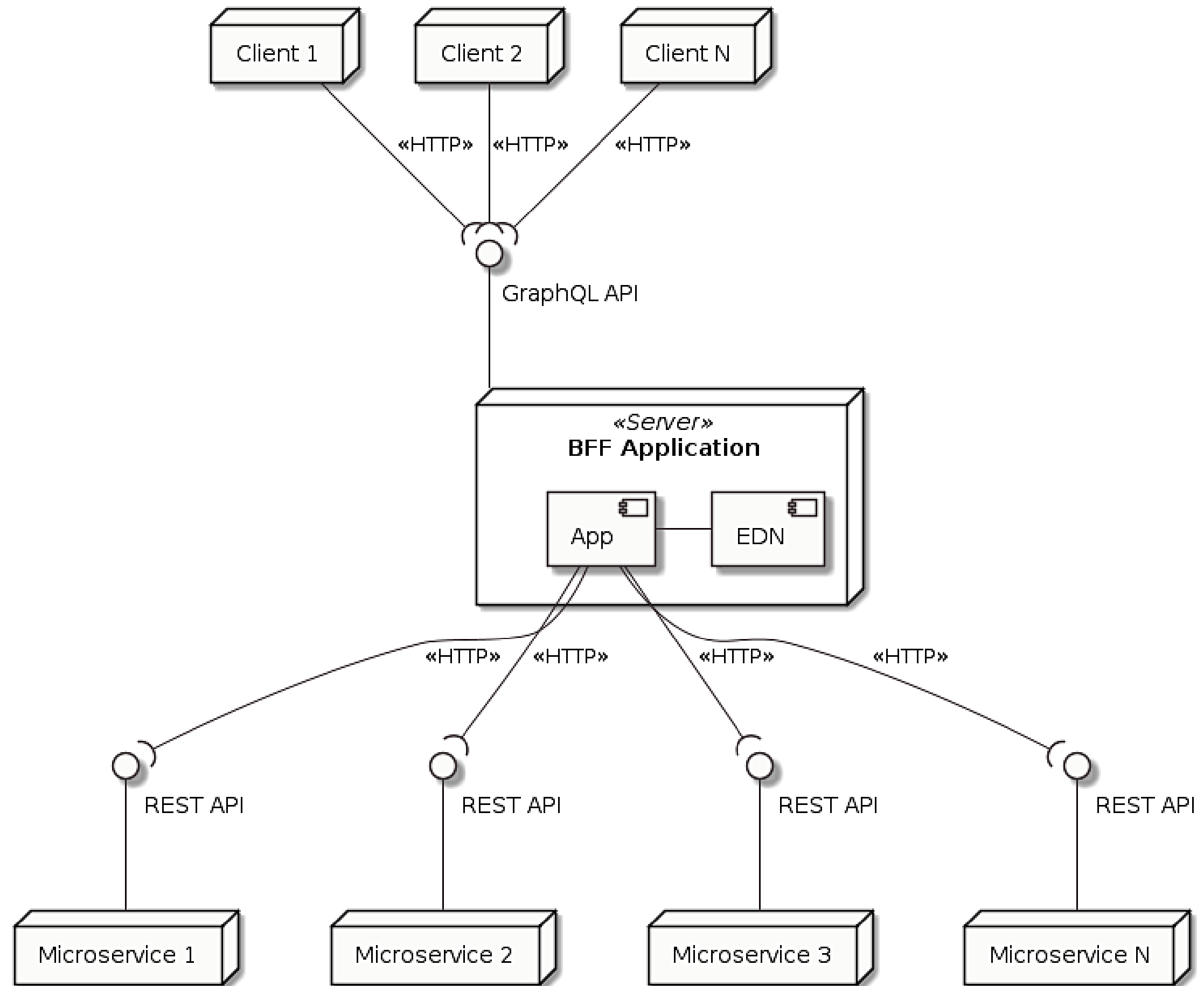
# Diagrama de **Arquitectura Lógica**



# Diagrama de Componentes

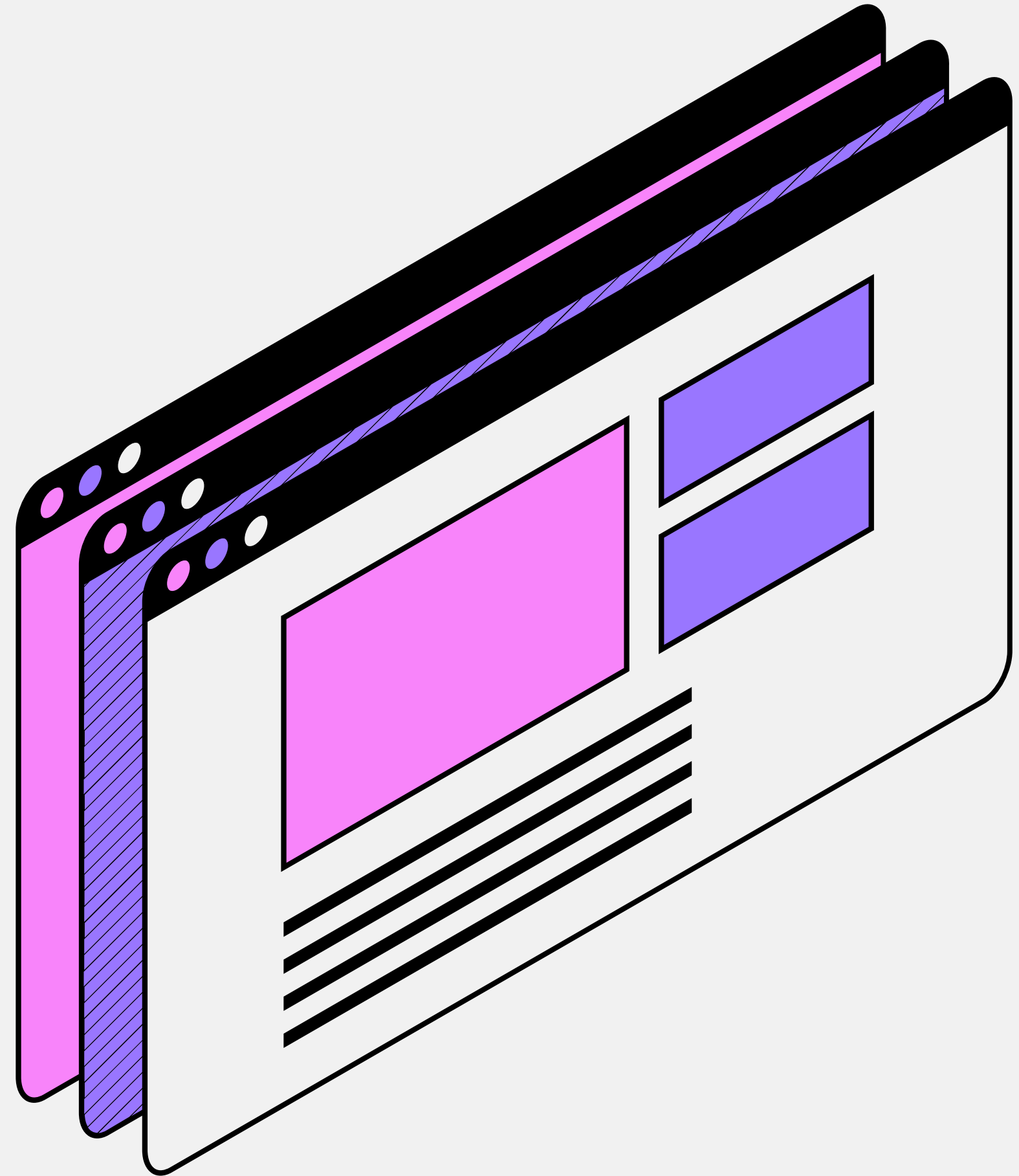


# Diagrama de Implantação



# Projeto de Interfaces de Usuário

Telas do Sistema



# Projeto de Interface

Aba de gerenciamento de Schemas (tela principal)

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:

Select

v

Fields / Values:

Add

Name:

Input

Description:

Input

Type:

Select

v

List

Required

Default value:

Input

Deprecated?

☐

Input

Delete

Name:

Input

Description:

Input

Type:

Select

v

List

Required

Default value:

Input

Save

# Projeto de Interface

## Aba de gerenciamento de APIs

Header

Schemas

APIs

Queries

Mutations

Settings

Endpoint 1

▼

Endpoint 2

▼

Endpoint 3

▼

Endpoint 4

▼

Endpoint 5

▼

Endpoint 6

▼

...

Create new

Delete

Name:

Input

Method

▼

Description:

Input

URI:

Input

Into response:

Input

Save

# Projeto de Interface

## Aba de gerenciamento de Queries

Header

Schemas

APIs

Queries

Mutations

Settings

Query 1

>

Query 2

>

Query 3

>

Query 4

>

Query 5

>

Query 6

>

...

Create new

Delete

Name:

Input

Description:

Input

Type:

Select

>

List

Required

Source:

Select

>

Args:

Add

Name:

Select

>

Type:

Select

>

Delete

Name:

Select

>

Type:

Select

>

Delete

Name:

Select

>

Type:

Select

>

Delete

Save



# Projeto de Interface

Aba de gerenciamento de *Mutations*

Header

Schemas

APIs

Queries

Mutations

Settings

Mutation 1

▼

Mutation 2

▼

Mutation 3

▼

Mutation 4

▼

Mutation 5

▼

Mutation 6

▼

...

Create new

Delete

Name:

Input

Description:

Input

Type:

Select

▼

List

Required

Source:

Select

▼

Args:

Add

Name:

Select

▼

Type:

Select

▼

Delete

Name:

Select

▼

Type:

Select

▼

Delete

Name:

Select

▼

Type:

Select

▼

Delete

Save

# Projeto de Interface

Aba de gerenciamento de *Settings* (configurações)

Header

Schemas

APIs

Queries

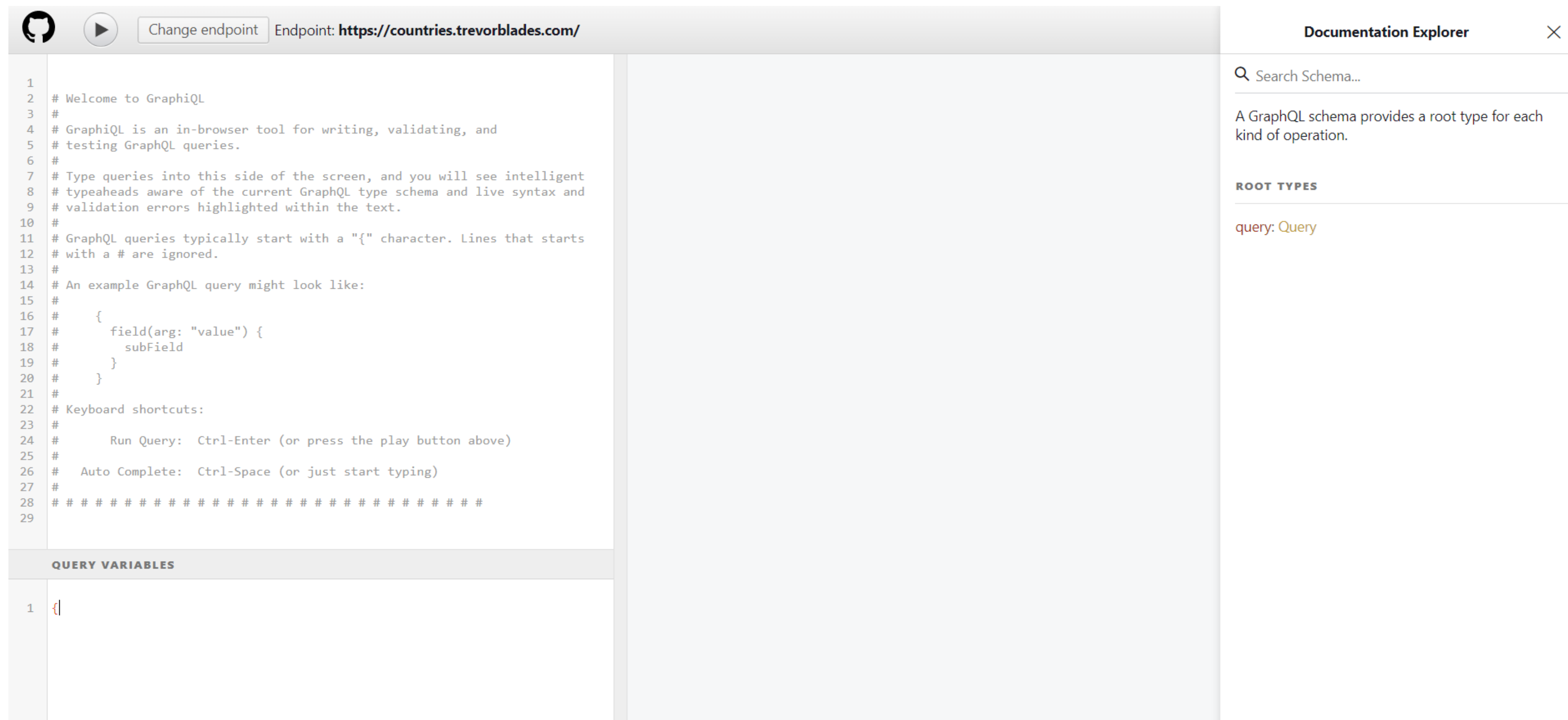
Mutations

Settings

Name:	Input	Env:	Select	v
Description:	Input	Host:	Input	
API Path:	Input	Port:	Input	
IDE Path:	Input	CORS:	Input	

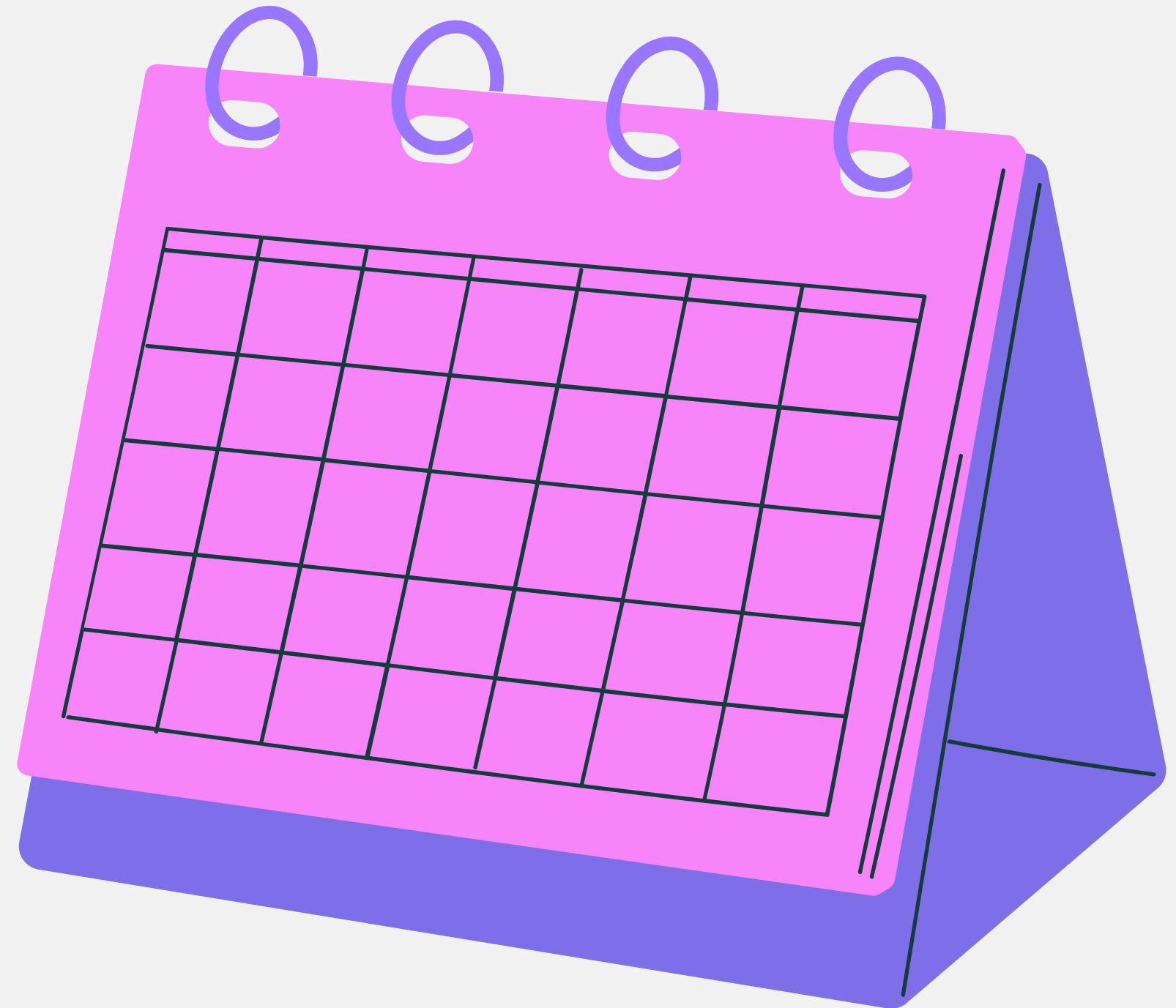
Save

## Tela de teste e exploração da API (GraphQL)



# Cronograma de Desenvolvimento

TCC II



# Tarefas

1. Implementar componente *Storage*

2. Implementar componente *Diplomat*

3. Implementar componente *Resolver*

4. Implementar componente *API (Main)* com *GraphiQL*

5. Implementar componente *ManagerUI*

6. Implementar componente *Controller*

7. Implementar tela de gerenciamento de *Schemas*

8. Implementar tela de gerenciamento de *APIs*

9. Implementar tela de gerenciamento de *Queries*

10. Implementar tela de gerenciamento de *Mutations*

11. Implementar tela de gerenciamento de *Settings*

12. Implementar *script* de *construção (build)* da aplicação

13. Abstrair e empacotar o *código base* do sistema

14. Implementar *script* de *geração* do projeto

# Cronograma (parte 1)

## Tarefas por *Sprint*

Número da Tarefa	2022						
	Agosto		Setembro		Outubro		Novembro
	<i>Sprint 1</i>	<i>Sprint 2</i>	<i>Sprint 3</i>	<i>Sprint 4</i>	<i>Sprint 5</i>	<i>Sprint 6</i>	<i>Sprint 7</i>
<b>1</b>	X						
<b>2</b>	X						
<b>3</b>		X					
<b>4</b>		X					
<b>5</b>			X				
<b>6</b>			X				
<b>7</b>				X			

# Cronograma (parte 2)

## Tarefas por *Sprint*

Número da Tarefa	2022						
	Agosto		Setembro		Outubro		Novembro
	<i>Sprint 1</i>	<i>Sprint 2</i>	<i>Sprint 3</i>	<i>Sprint 4</i>	<i>Sprint 5</i>	<i>Sprint 6</i>	<i>Sprint 7</i>
8				X			
9					X		
10					X		
11						X	
12						X	
13							X
14							X



**PUC Minas**

**Obrigado pela atenção!**

**Trabalho de Conclusão de Curso I  
Engenharia de Software  
PUC Minas**

**Marlon Henrique da Silva**