

GraphTest

Lucas Cabral Soares

Maria Eduarda Amaral Muniz

Uma Ferramenta para gerar
grafos de teste estrutural e
funcional

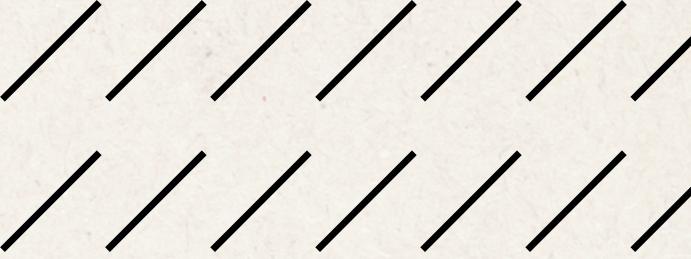
Orientadores:

Cleiton Silva Tavares

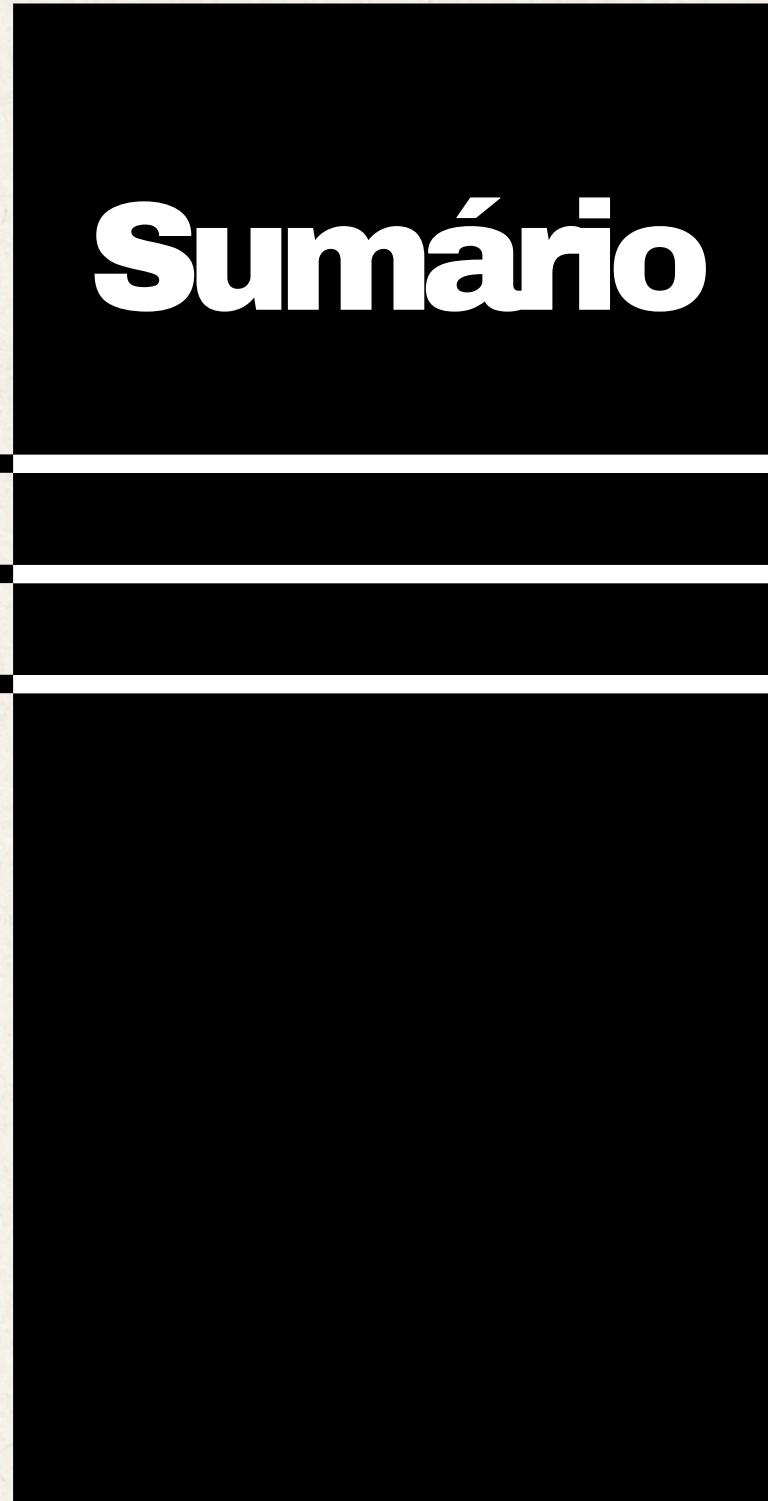
Danilo de Quadros Maia Filho

Leonardo Vilela Cardoso

Raphael Ramos Dias Costa



Sumário



01	Objetivo do Software
02	Escopo do Software
03	Principais Necessidades e Funcionalidades
04	Descrição dos Atores
05	Diagrama de Caso de Uso
06	Arquitetura do Software
07	Modelagem dos Principais Componentes
08	Interfaces com o Usuário
09	Cronograma do TCC II

01 Objetivo do Software

Contextualização

- A ferramenta surgiu da necessidade de oferecer um recurso acessível, interativo e web para apoiar estudantes, professores e profissionais na derivação de casos de teste.
- O GraphTest é uma plataforma web criada para facilitar o ensino e a prática de Teste de Software, reunindo técnicas de caixa branca e caixa preta em um único ambiente.
- O sistema permite gerar e manipular grafos de fluxo de controle e grafos de causa-efeito de forma visual, reduzindo a complexidade de processos manuais.

Problema

- Há escassez de ferramentas web que integrem, em um mesmo ambiente, a modelagem completa de GFC e GCE de forma intuitiva.
- A ausência dessas soluções dificulta a prática dessas estratégias de teste e limita a visualização clara de critérios de cobertura.
- Processos manuais tornam a geração de grafos e casos de teste trabalhosa, suscetível a erros e pouco acessível para uso acadêmico e profissional.

02 Escopo do Software

Dentro do Escopo

- A plataforma integra, em um único ambiente, recursos para testes estruturais e funcionais.
- No teste estrutural, o usuário importa código Java e o sistema gera automaticamente o Grafo de Fluxo de Controle, calcula a complexidade ciclomática e produz o esqueleto do teste.
- No teste funcional, o usuário modela causas e efeitos para formar o GCE, converte o grafo em tabela de decisão e obtém o esqueleto do teste funcional. A solução unifica duas perspectivas de teste geralmente separadas, oferecendo uma ferramenta completa.

Fora do Escopo

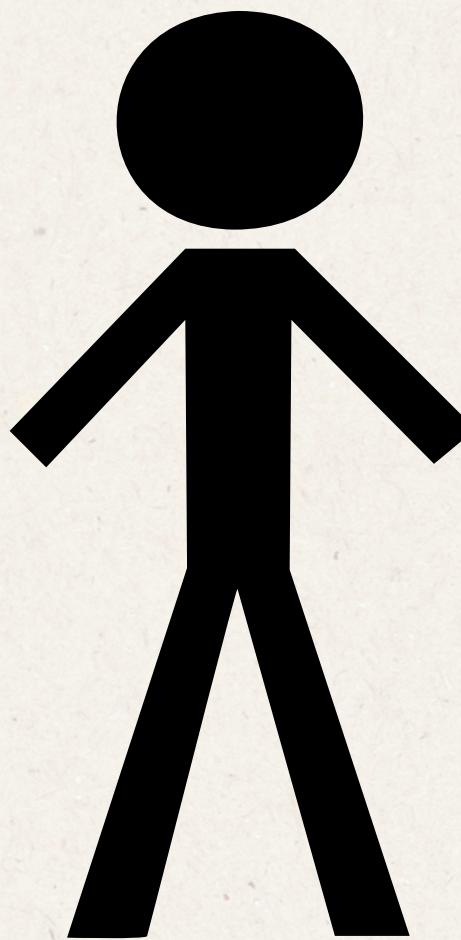
- A plataforma não terá como objetivo executar testes automatizados diretamente em frameworks como JUnit, PyTest ou Selenium.
- Não será oferecido suporte a outras linguagens de programação além de Java.
- Não haverá integração direta com ambientes de desenvolvimento integrados (IDEs) como Eclipse ou IntelliJ IDE, considerando que a interação do usuário com o sistema ocorrerá exclusivamente por meio da interface web da plataforma.

03 Principais Necessidades e Funcionalidades

Necessidades	Funcionalidades
Automatização da geração de grafos de fluxo de controle	<ul style="list-style-type: none">• Importar código-fonte• Gerar GFC a partir de um método pré-selecionado• Calcular automaticamente a complexidade ciclomática do método a partir do GFC gerado• Gerar esqueleto do método de teste
Derivação de casos de teste funcionais a partir de especificações	<ul style="list-style-type: none">• Modelar causas, efeitos, operadores lógicos e restrições por meio de interface gráfica• Converter o grafo de causa-efeito em tabela de decisão• Validar consistência do modelo• Gerar esqueleto do método de teste
Visualização interativa e didática das estruturas de teste	<ul style="list-style-type: none">• Fornecer layouts automáticos de grafos• Adicionar anotações/legendas automáticas

04 Descrição dos Atores

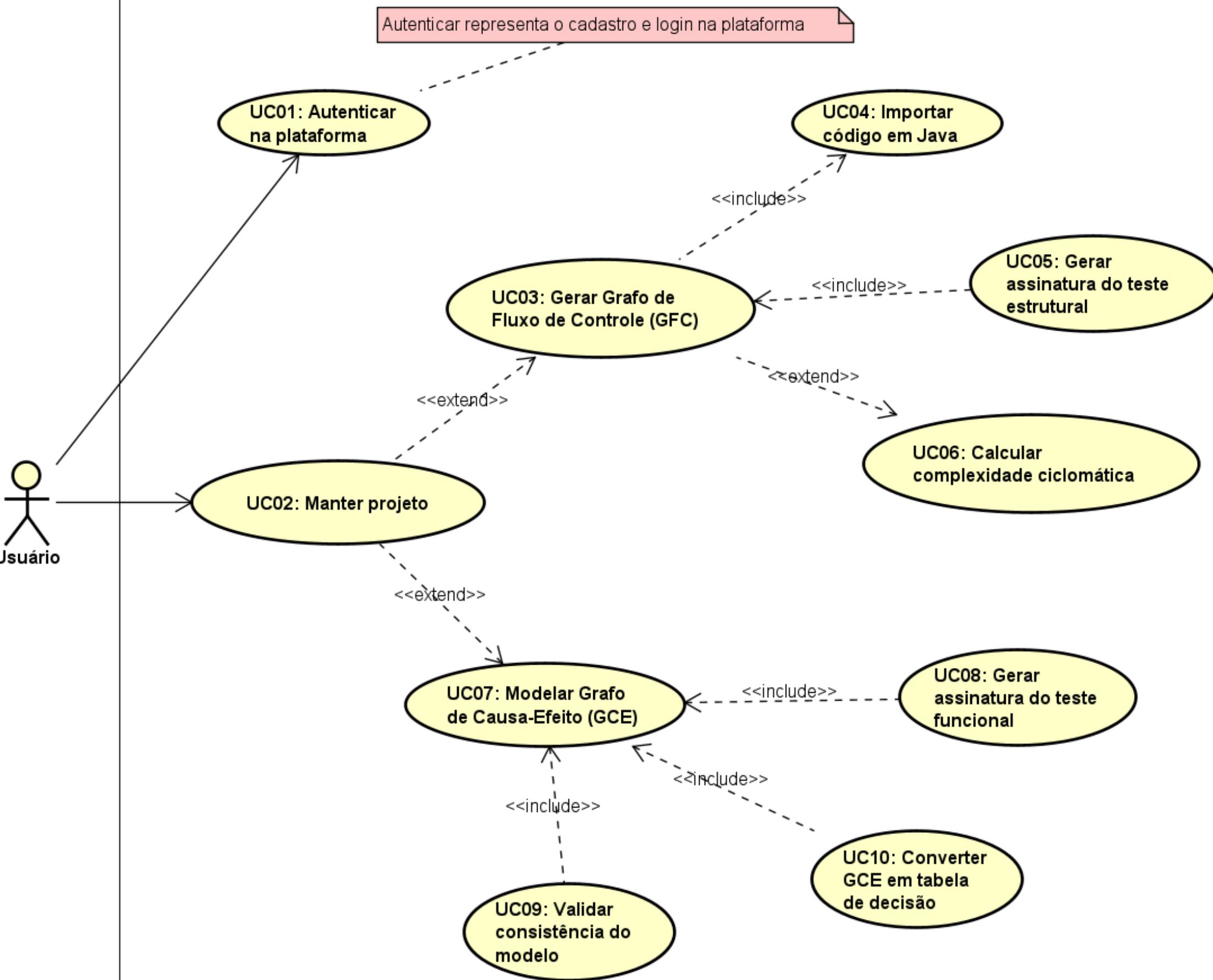
- O sistema possui um único ator: o Usuário. Ele representa qualquer pessoa que utiliza a plataforma, unificando todos os perfis em um mesmo papel.
- O Usuário interage com o sistema para gerar e analisar grafos de teste, criar modelos funcionais e estruturar casos de teste. Entre suas ações estão importar código para produzir o GFC, modelar causas e efeitos para formar o GCE, derivar tabelas de decisão e validar modelos.



05

Modelagem do diagrama de caso de uso

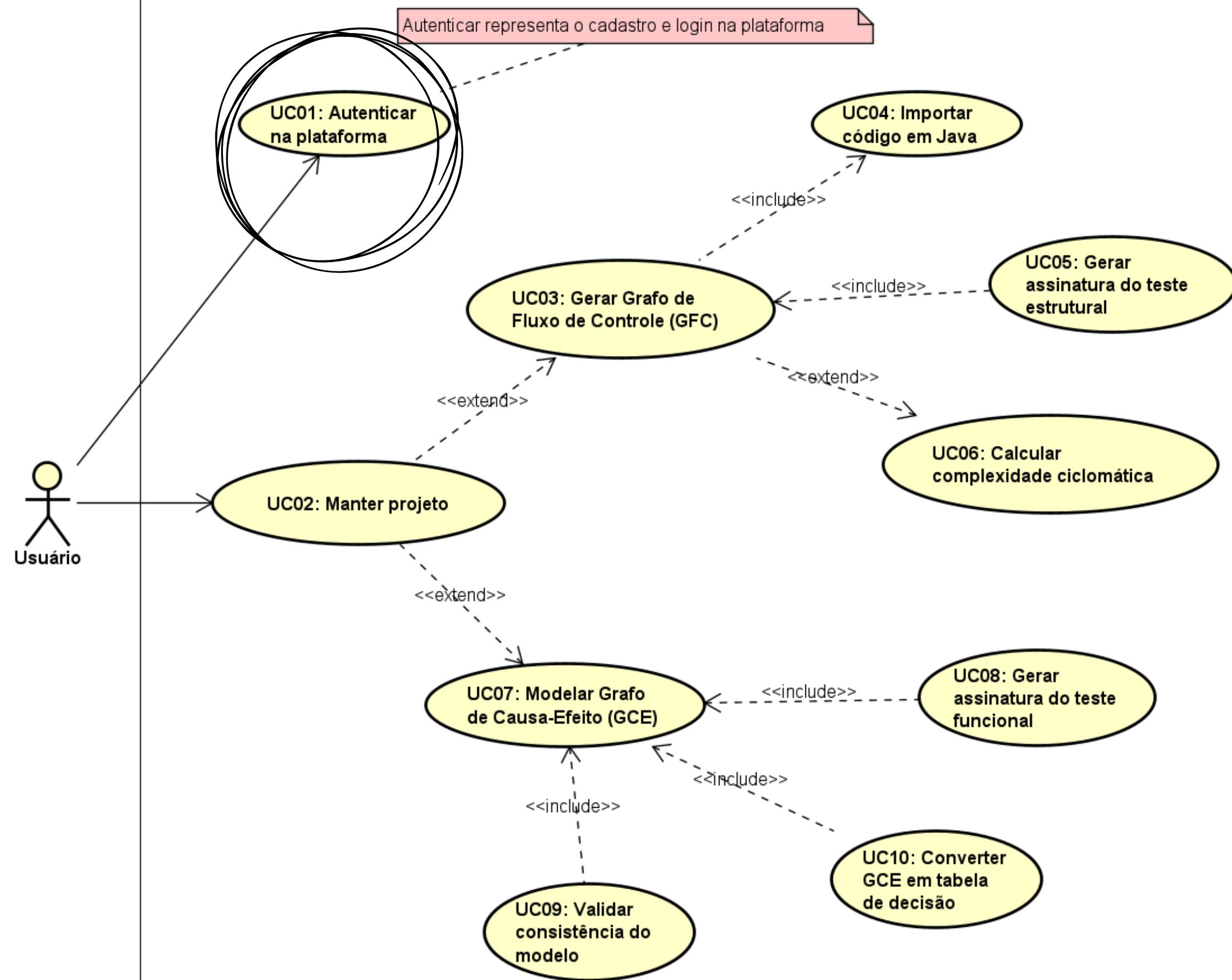
Plataforma de geração de grafos de teste



Modelagem do diagrama de caso de uso

- O sistema possui um único ator primário
 - Usuário
- Ao todo são 10 casos de uso

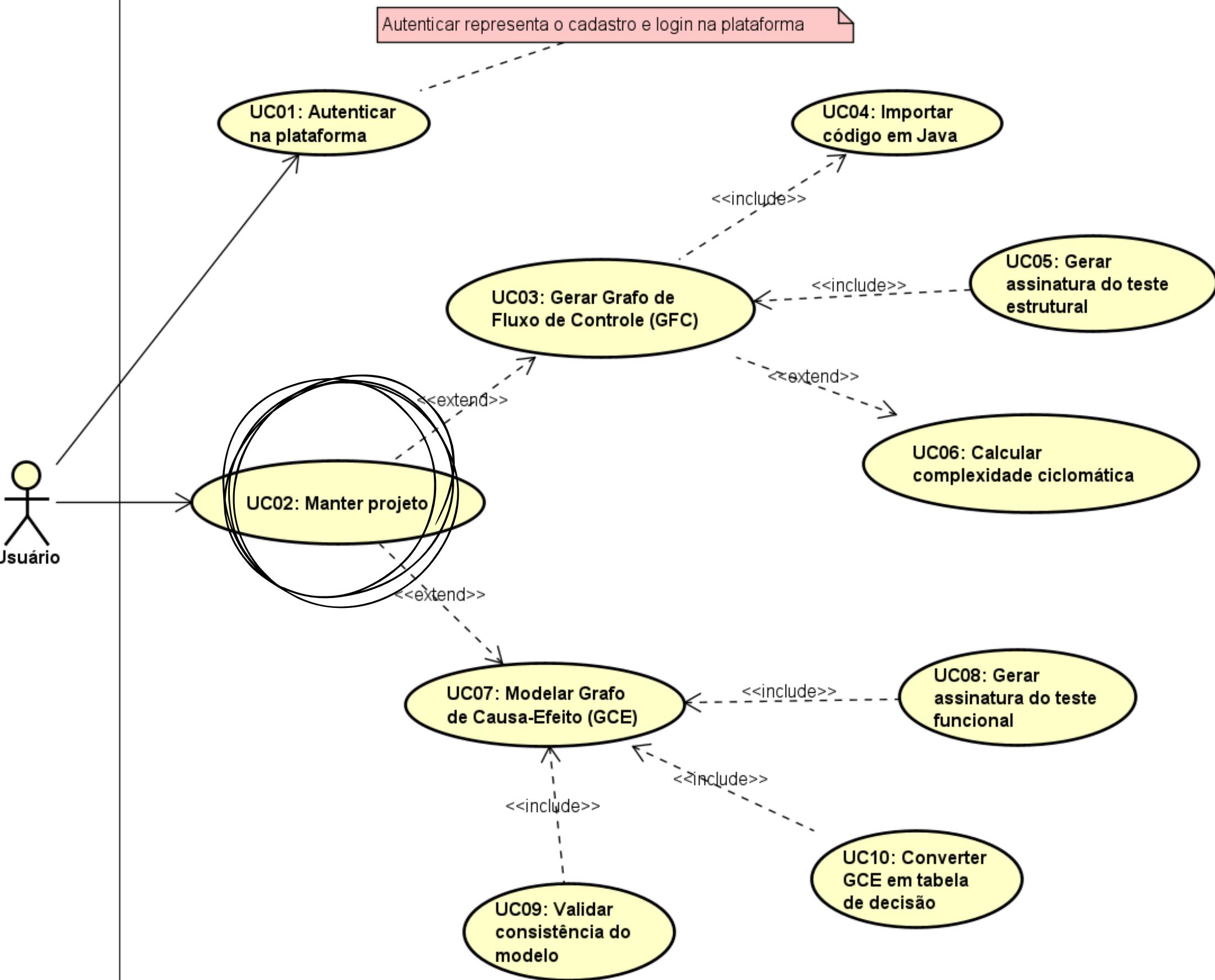
Plataforma de geração de grafos de teste



Modelagem do diagrama de caso de uso

- O sistema possui um único ator primário
 - Usuário
- Ao todo são 10 casos de uso

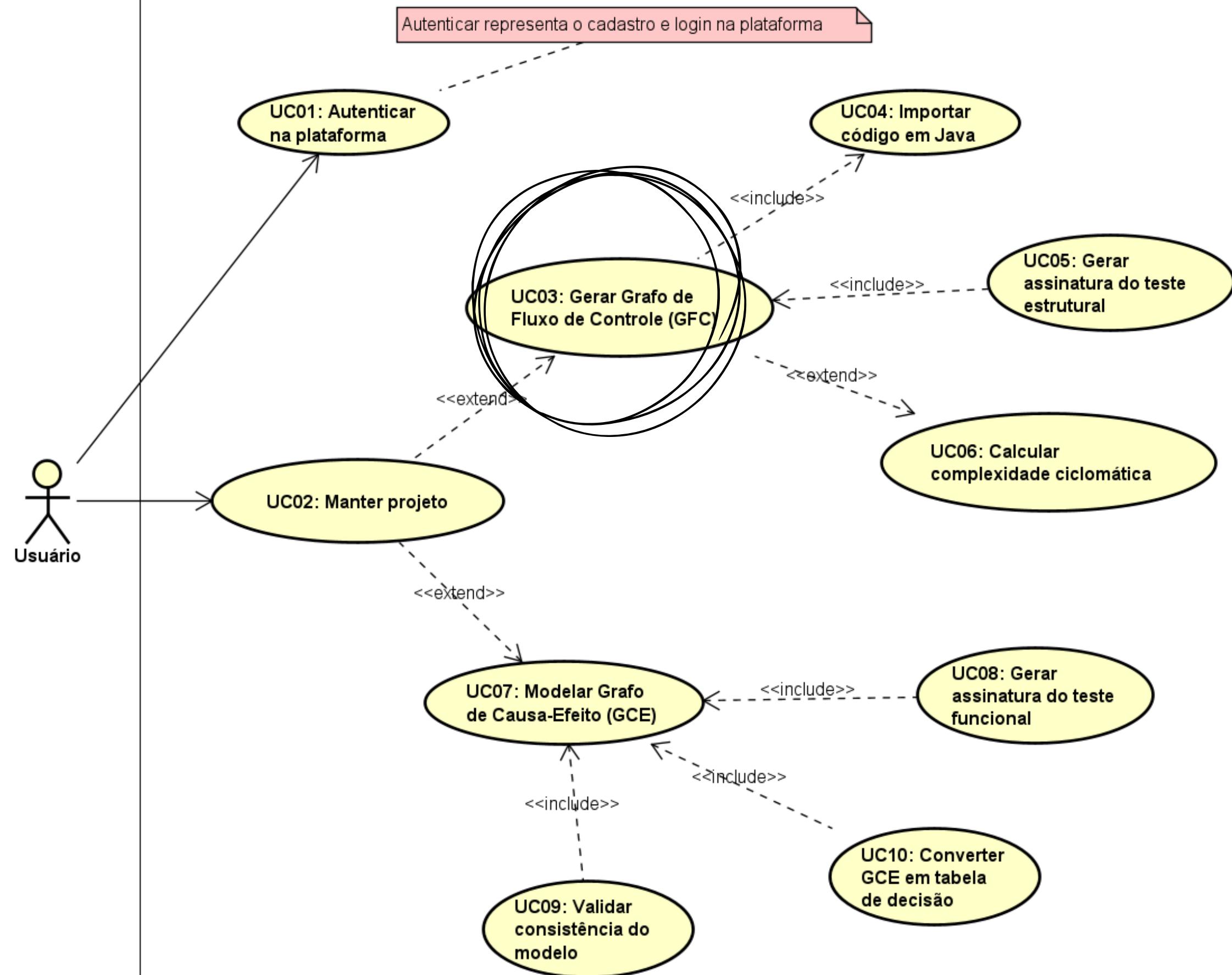
Plataforma de geração de grafos de teste



Modelagem do diagrama de caso de uso

- O sistema possui um único ator primário
 - Usuário
- Ao todo são 10 casos de uso

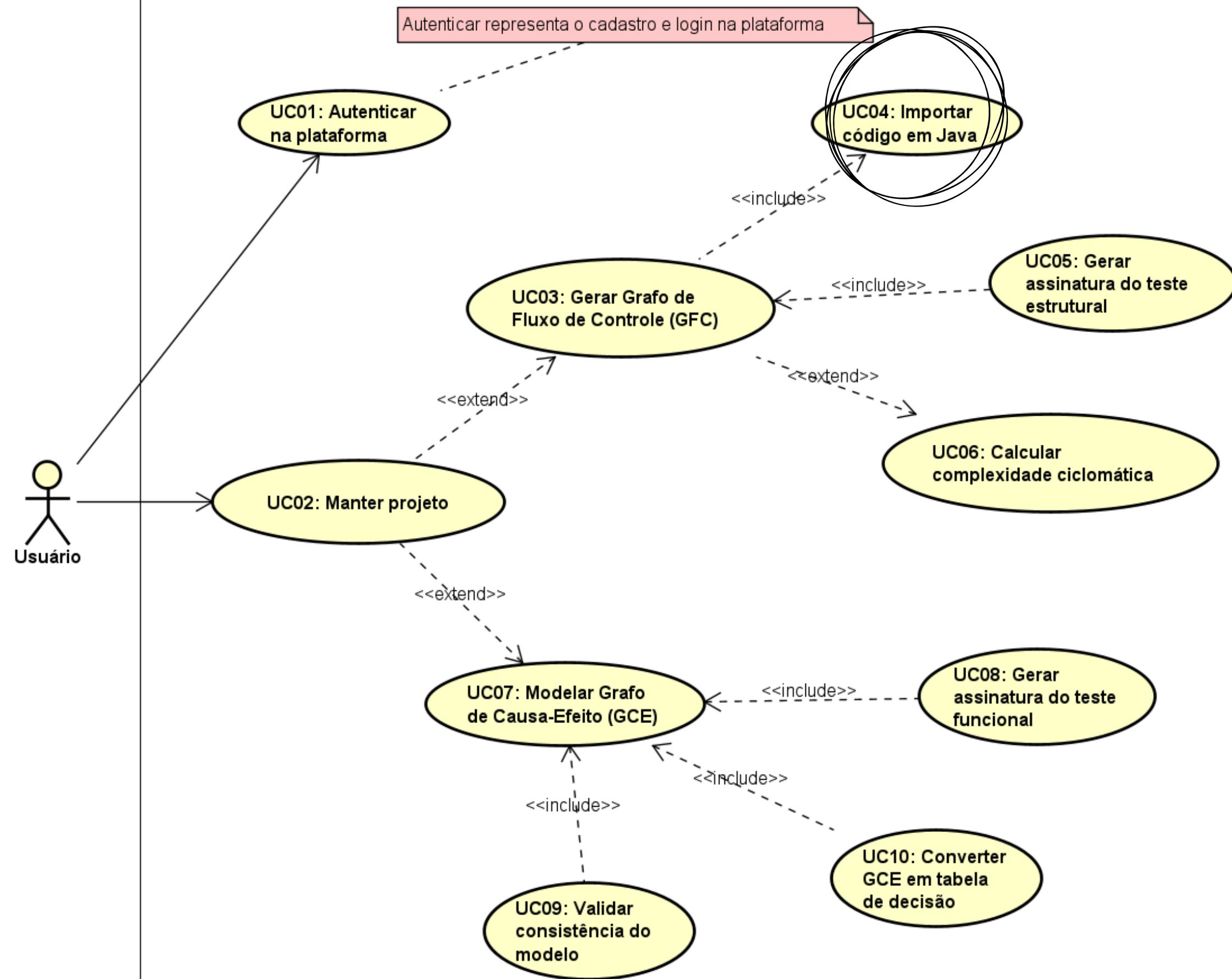
Plataforma de geração de grafos de teste



Modelagem do diagrama de caso de uso

- O sistema possui um único ator primário
 - Usuário
- Ao todo são 10 casos de uso

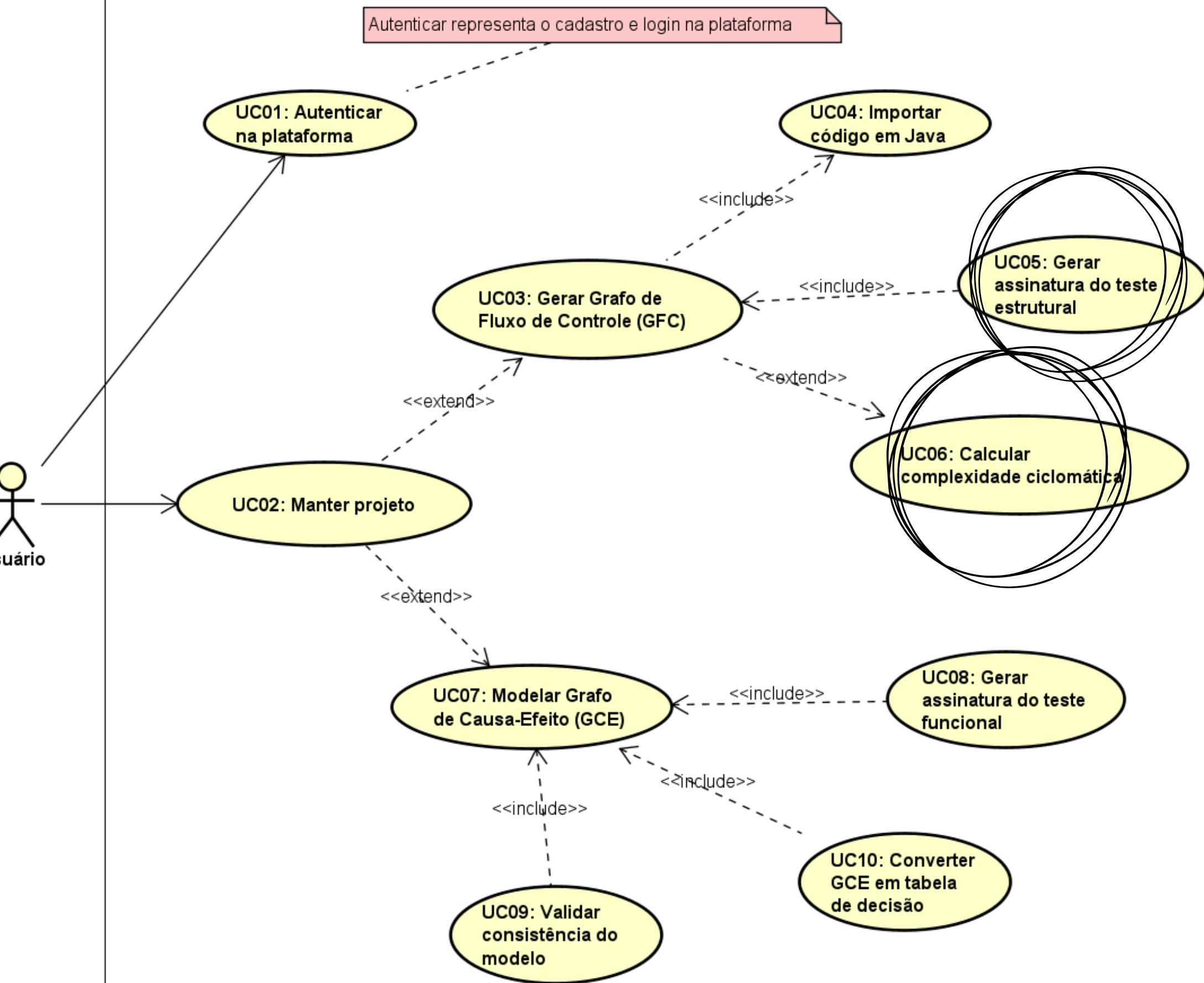
Plataforma de geração de grafos de teste



Modelagem do diagrama de caso de uso

- O sistema possui um único ator primário
 - Usuário
- Ao todo são 10 casos de uso

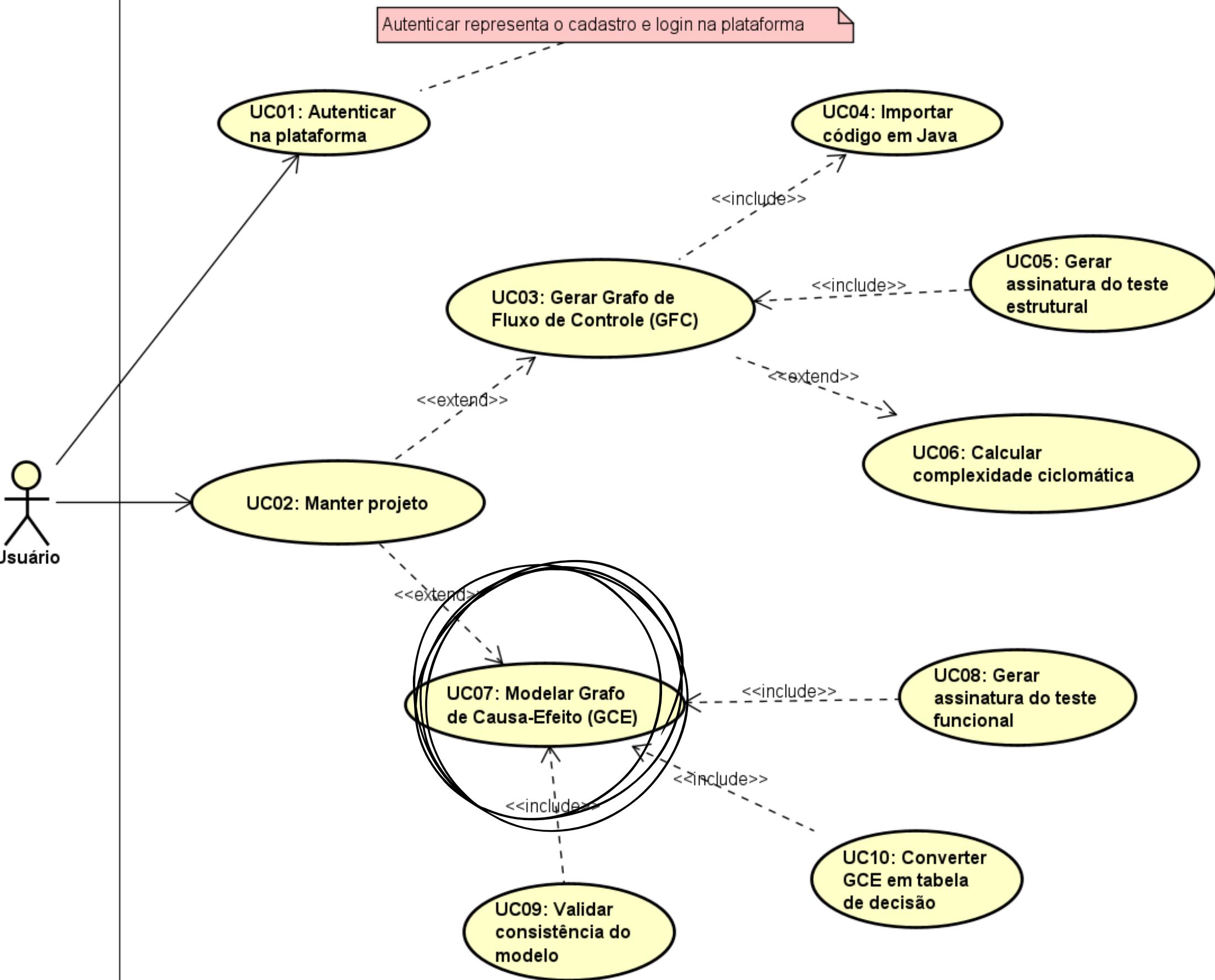
Plataforma de geração de grafos de teste



Modelagem do diagrama de caso de uso

- O sistema possui um único ator primário
 - Usuário
- Ao todo são 10 casos de uso

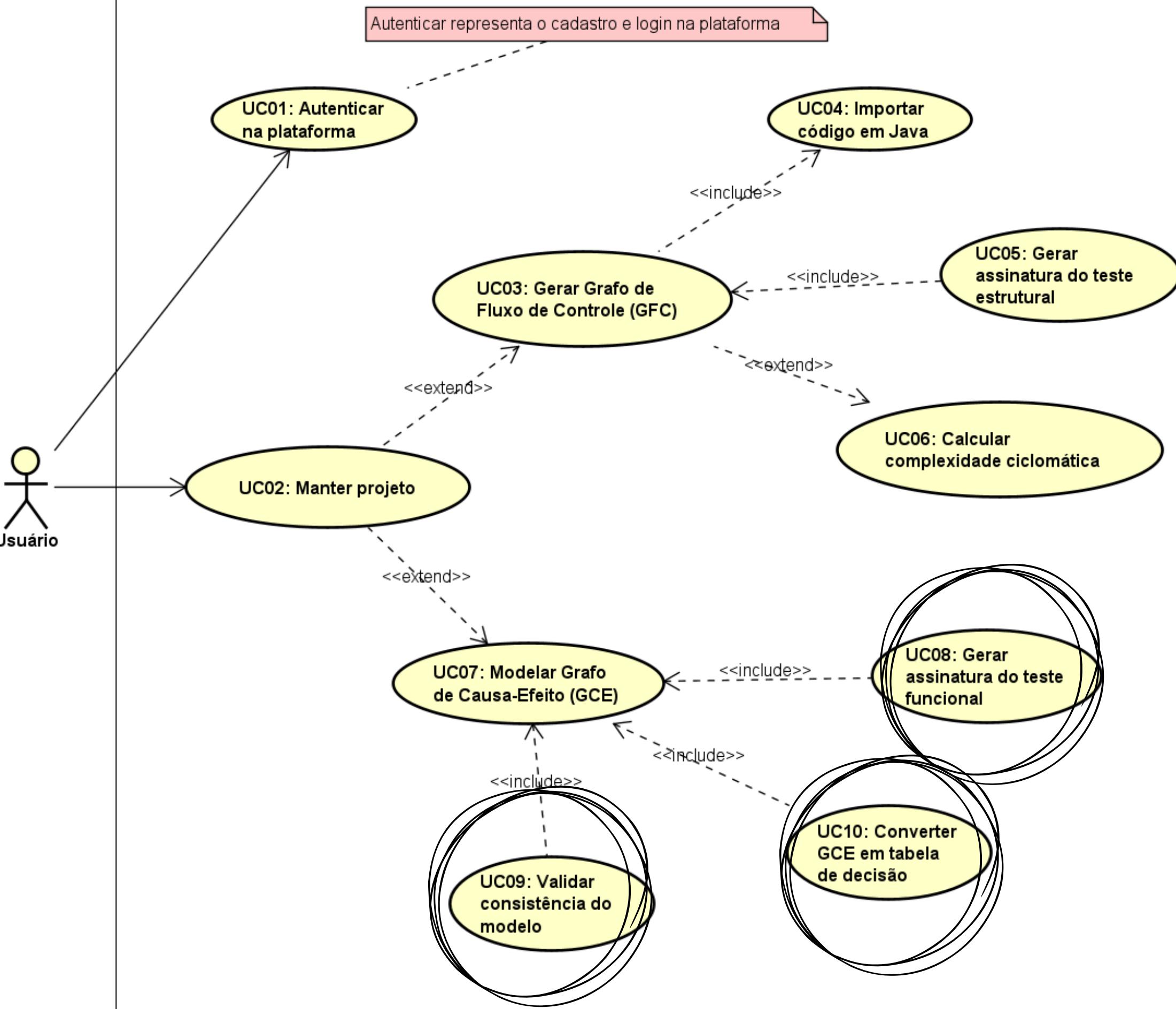
Plataforma de geração de grafos de teste



Modelagem do diagrama de caso de uso

- O sistema possui um único ator primário
 - Usuário
- Ao todo são 10 casos de uso

Plataforma de geração de grafos de teste



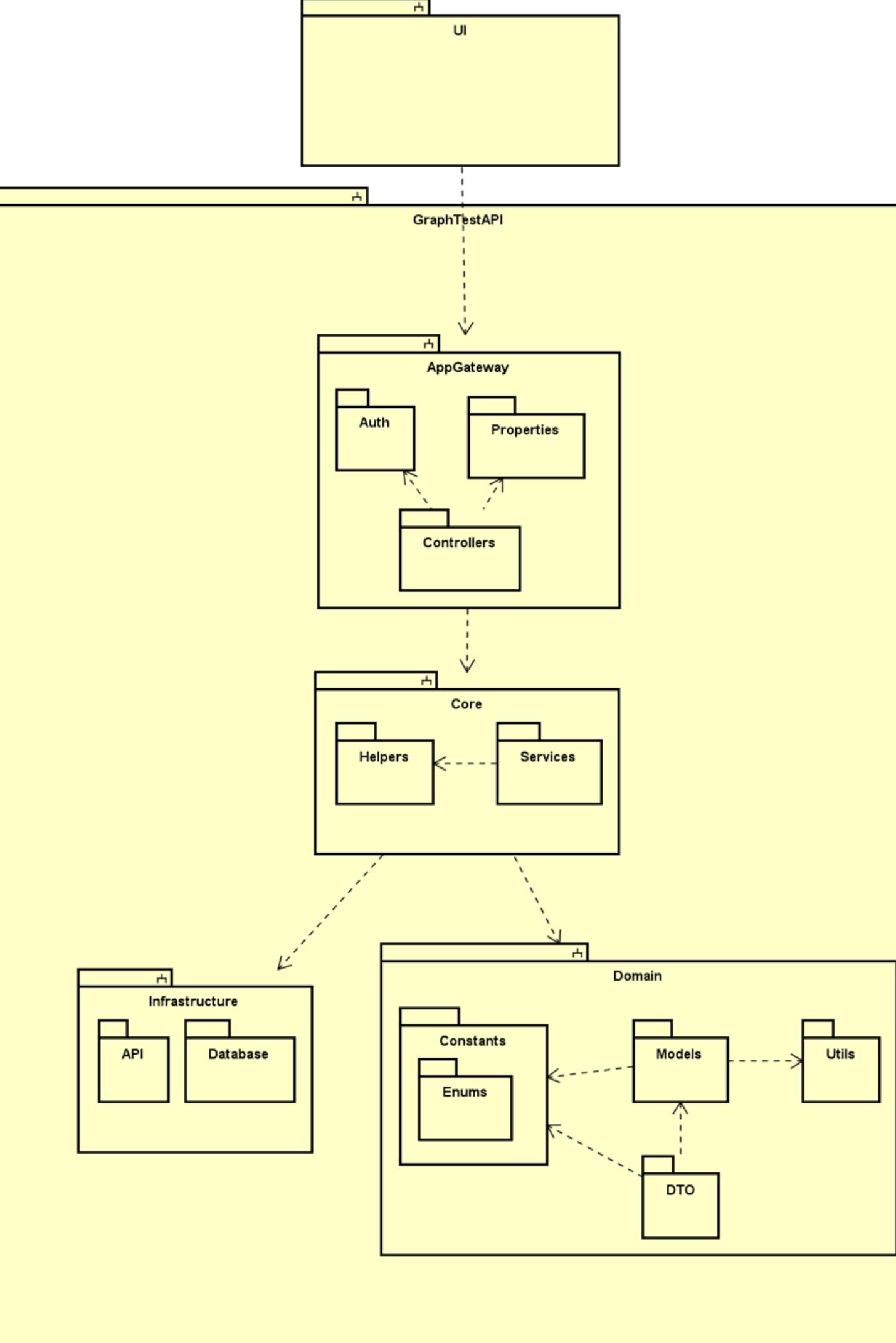
Modelagem do diagrama de caso de uso

- O sistema possui um único ator primário
 - Usuário
- Ao todo são 10 casos de uso

06

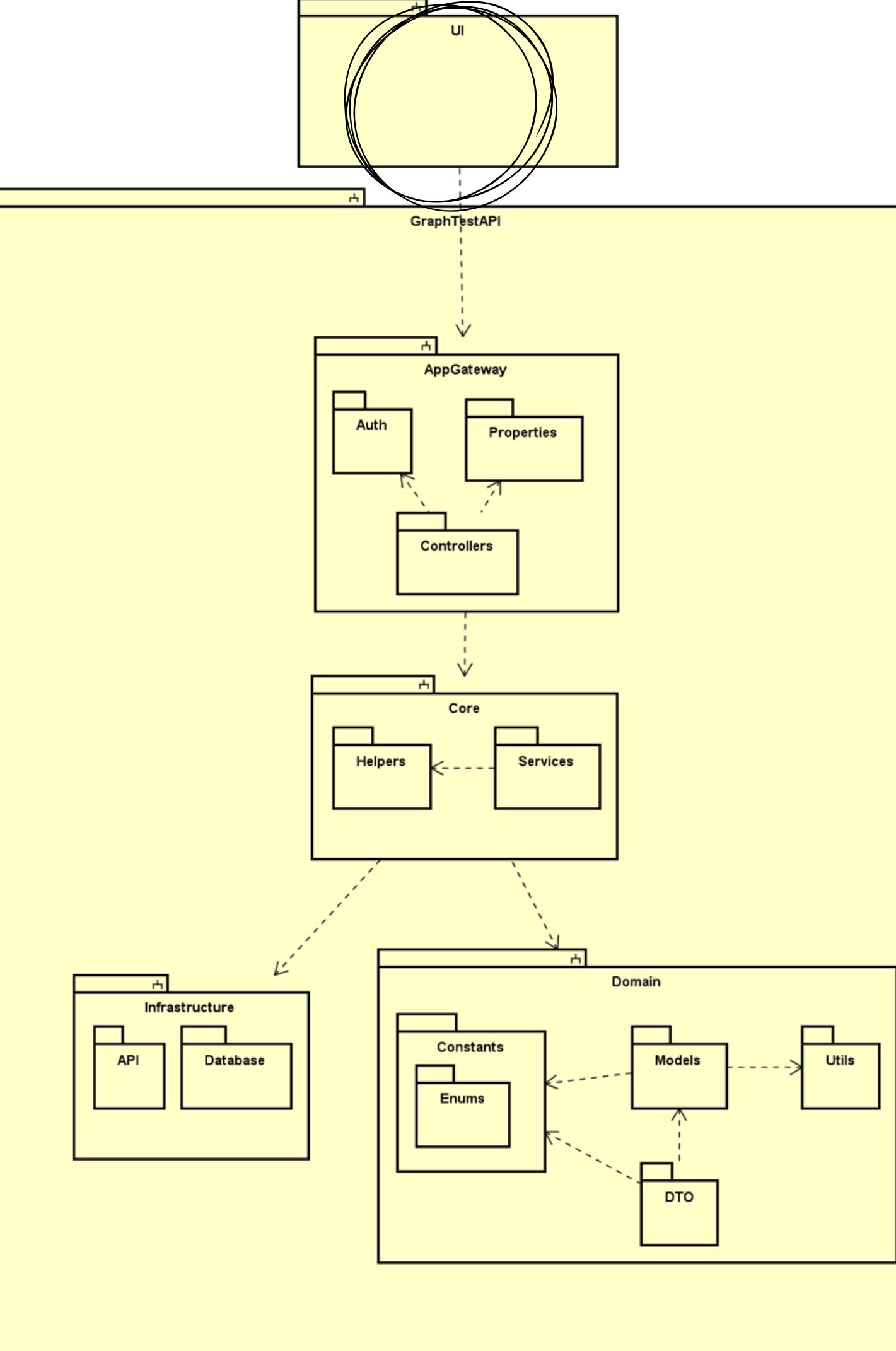
Modelagem da arquitetura de software

Visão geral do diagrama de pacotes



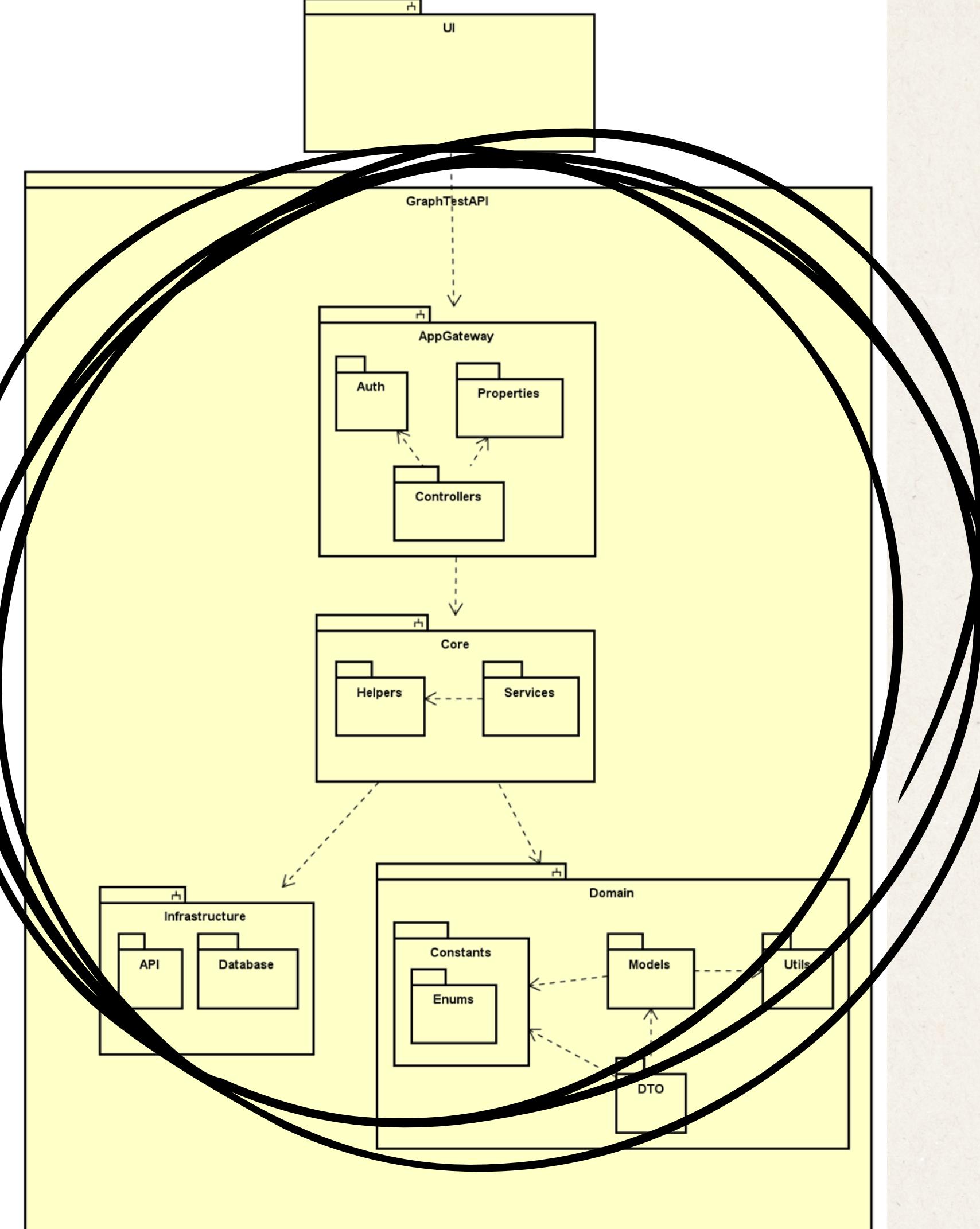
A arquitetura segue um modelo modular em camadas

Visão geral do diagrama de pacotes



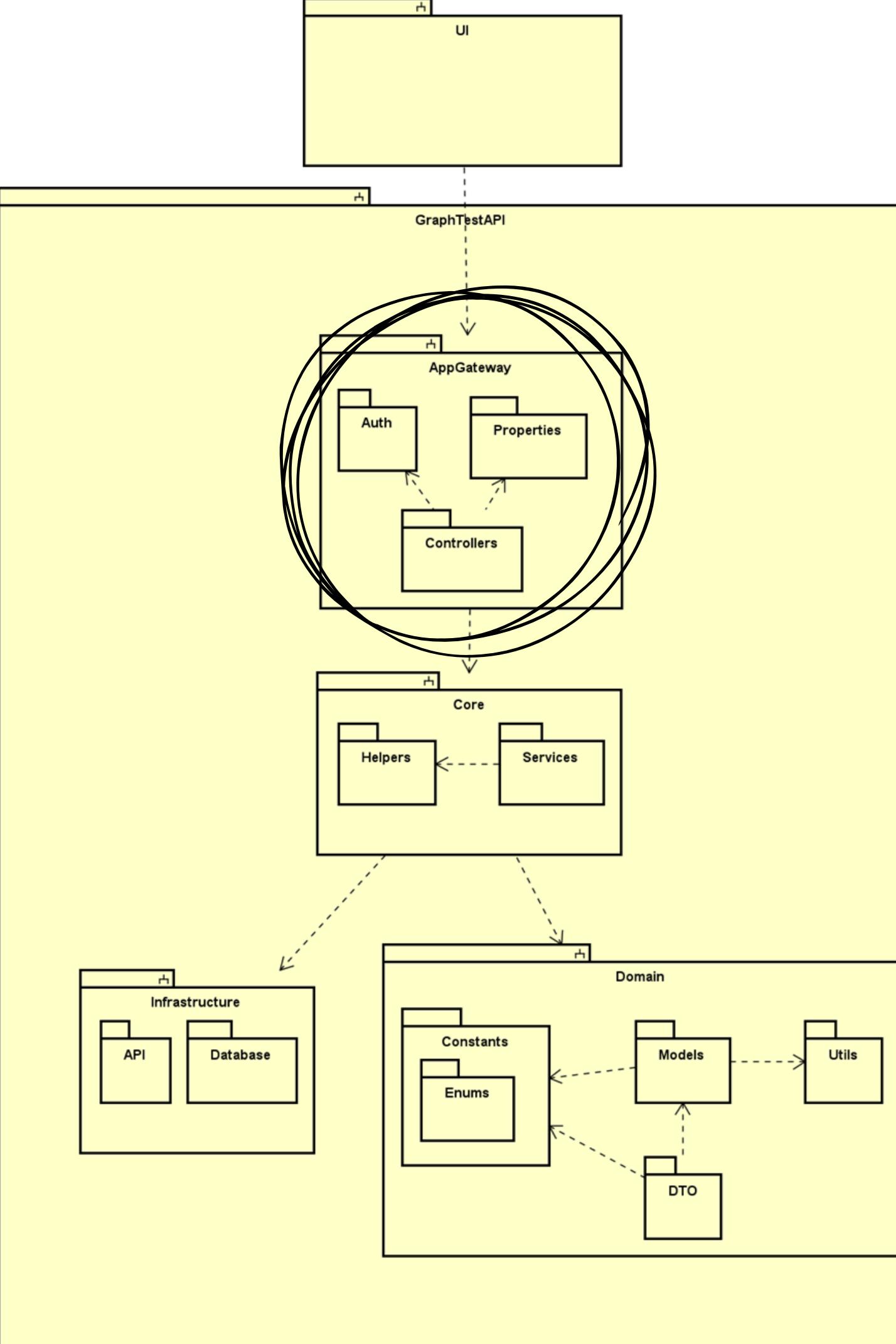
- **UI:** camada responsável pela interação direta com o usuário e envio das requisições ao sistema

Visão geral do diagrama de pacotes



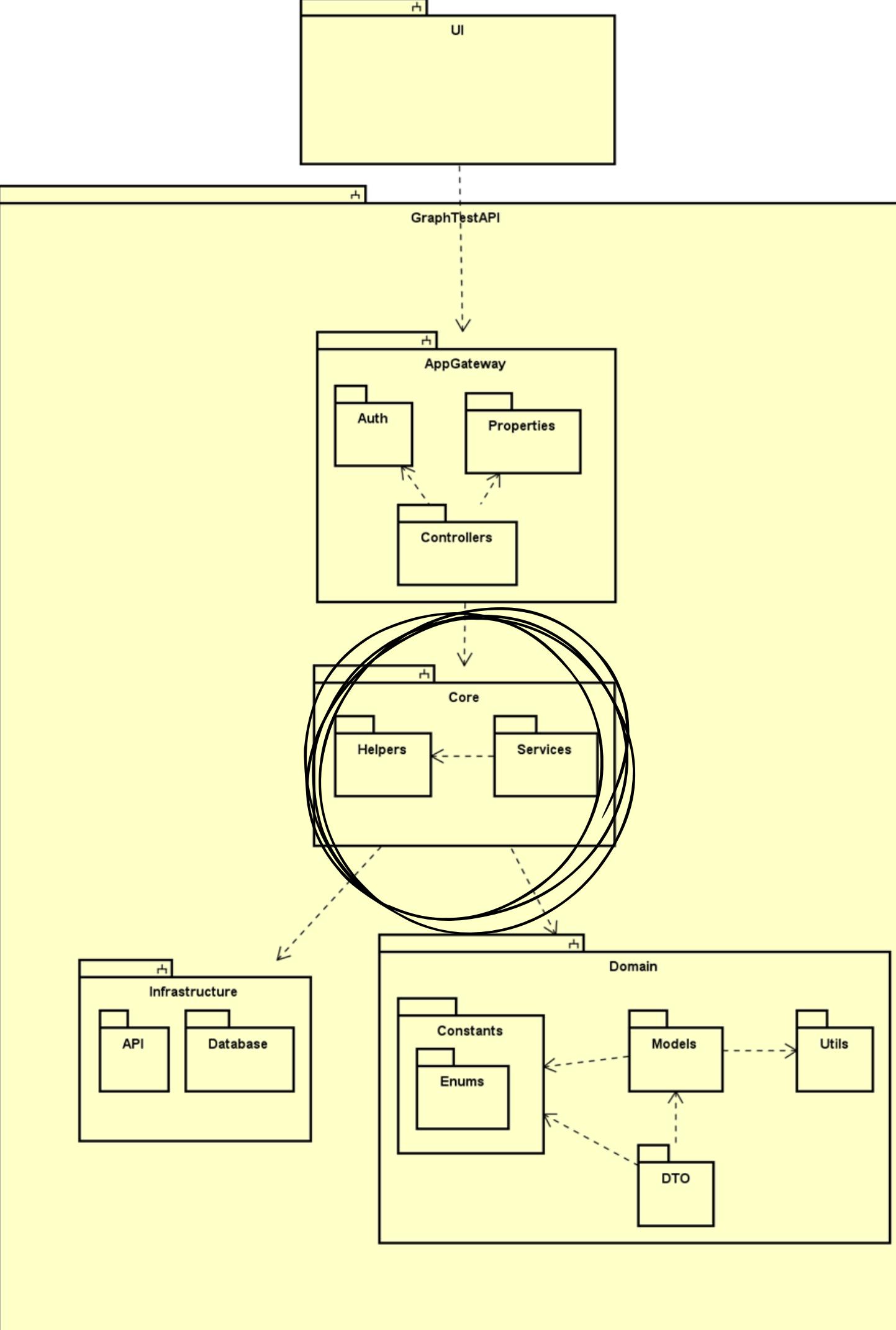
GraphTestAPI: a camada que expõe a aplicação para o exterior, funcionando como o ponto único de entrada das requisições provenientes da interface de usuário.

Visão geral do diagrama de pacotes



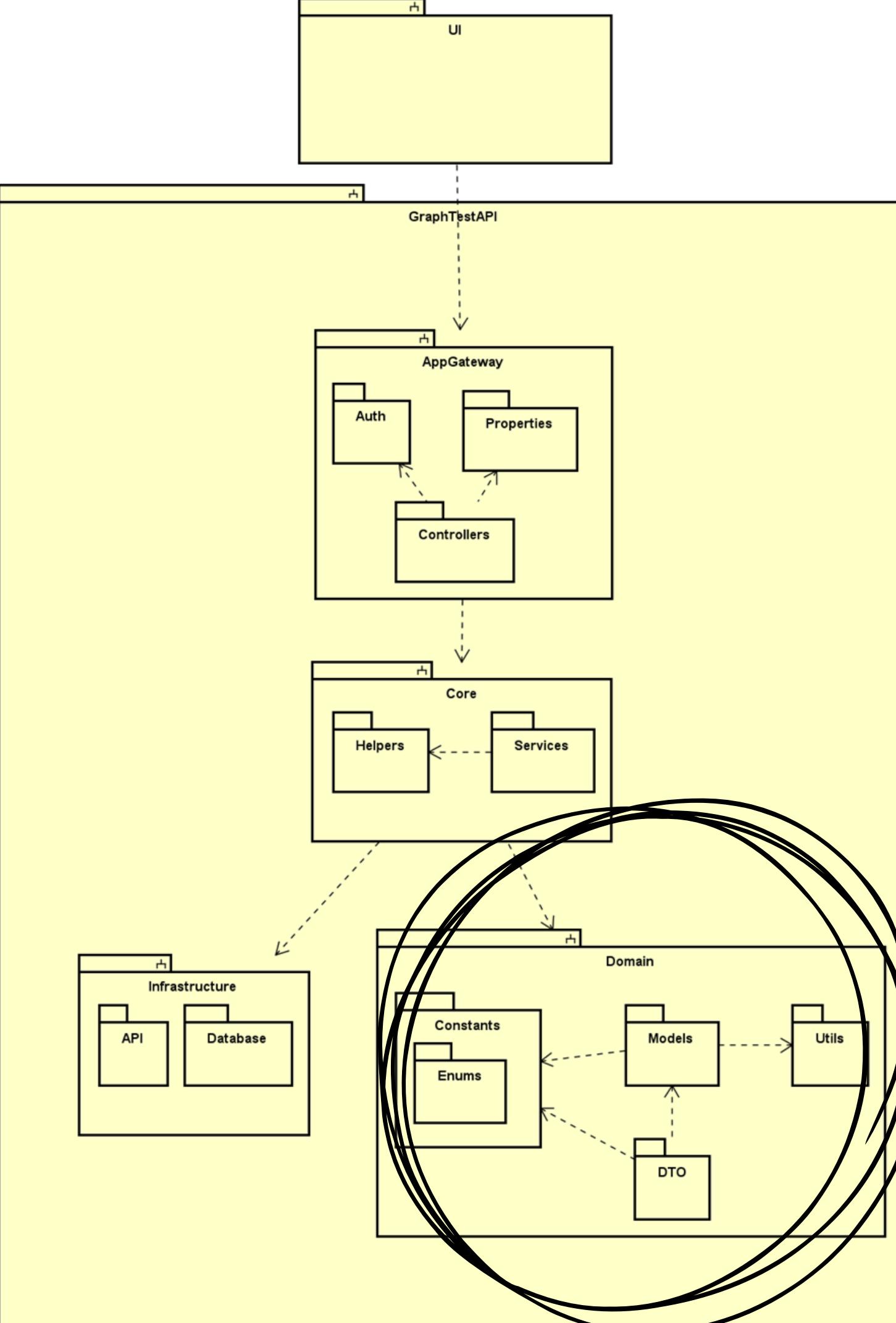
- **AppGateway**: camada que concentra os pontos de entrada, controla acesso e direciona requisições para o núcleo da aplicação

Visão geral do diagrama de pacotes



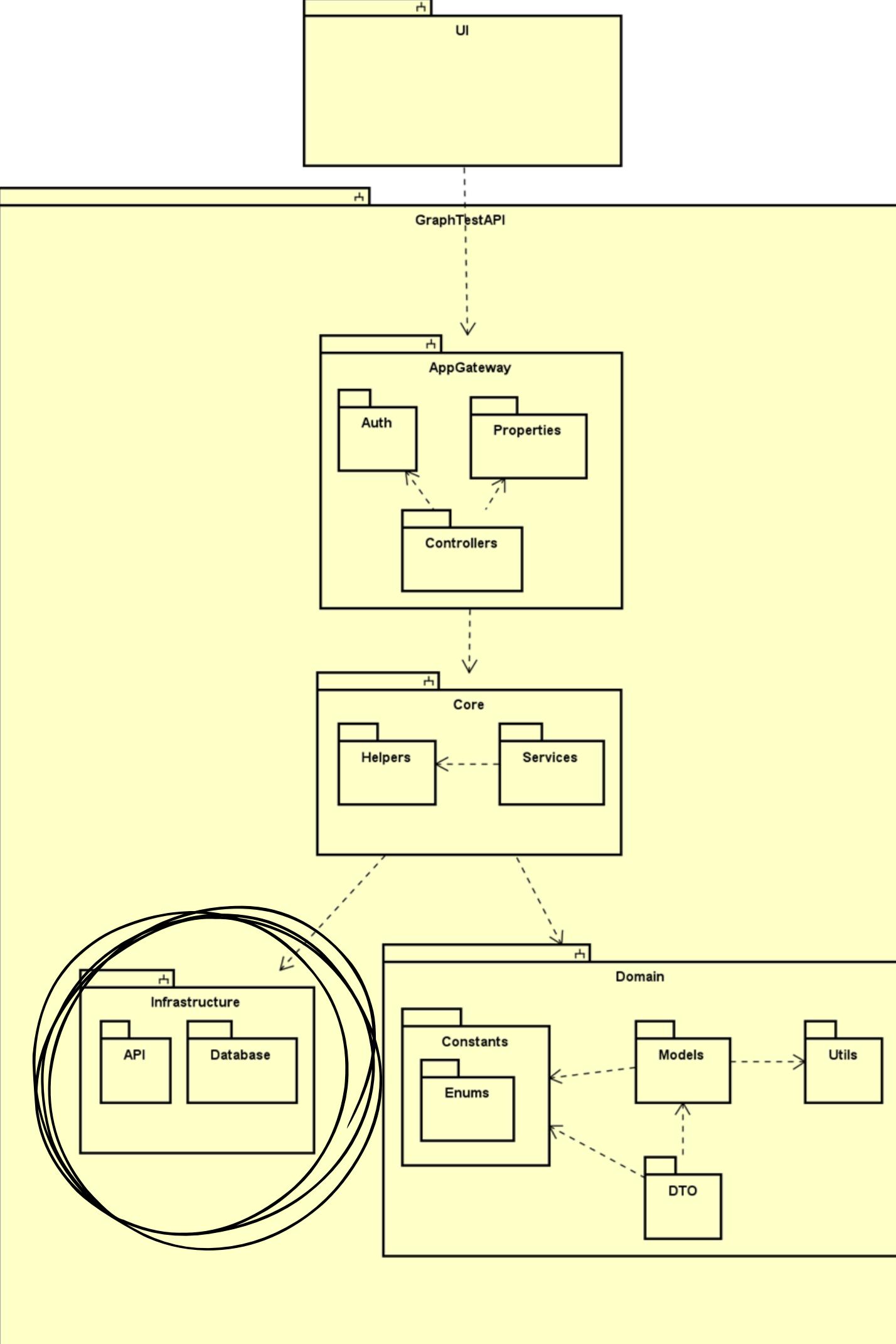
- **Core:** camada que implementa a lógica de negócio, executando geração de grafos, cálculos e operações centrais

Visão geral do diagrama de pacotes



- **Domain:** camada que define as entidades, regras e estruturas conceituais que representam o domínio do sistema

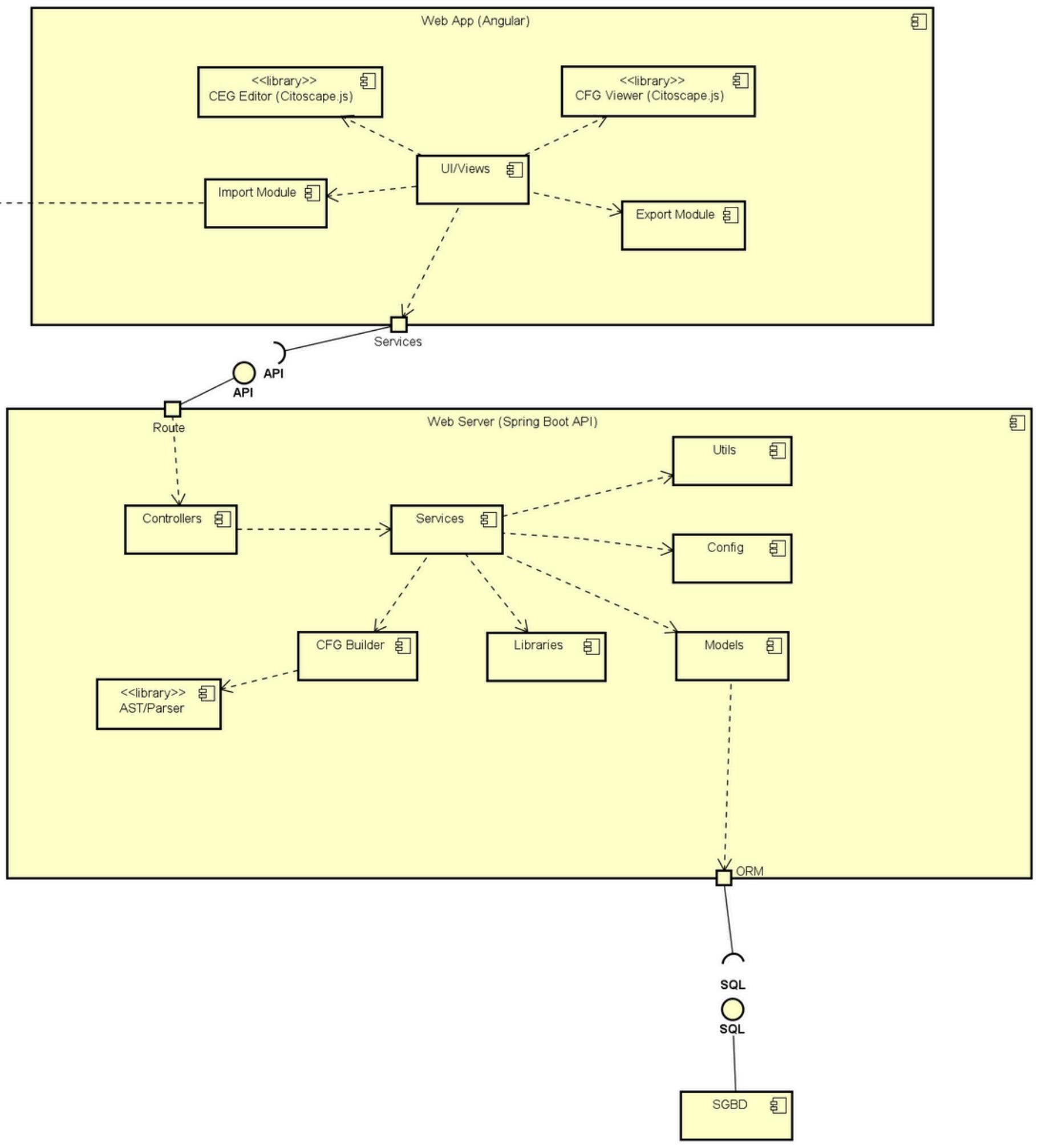
Visão geral do diagrama de pacotes



- **Infrastructure:** camada que fornece persistência e integração técnica, abstraindo detalhes de banco de dados e serviços externos.

07

Modelagem dos principais componentes

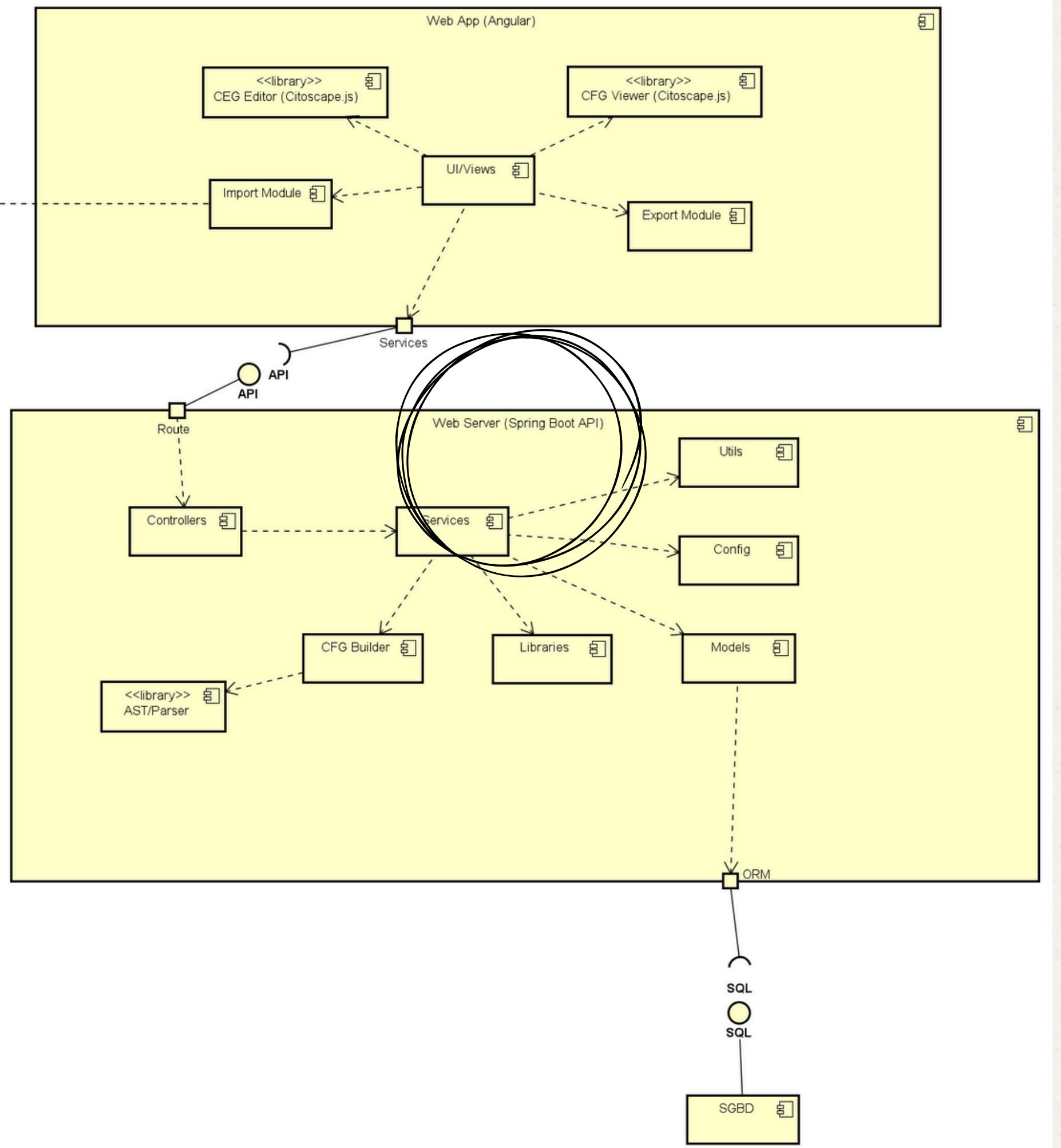


Visão geral do diagrama de componentes

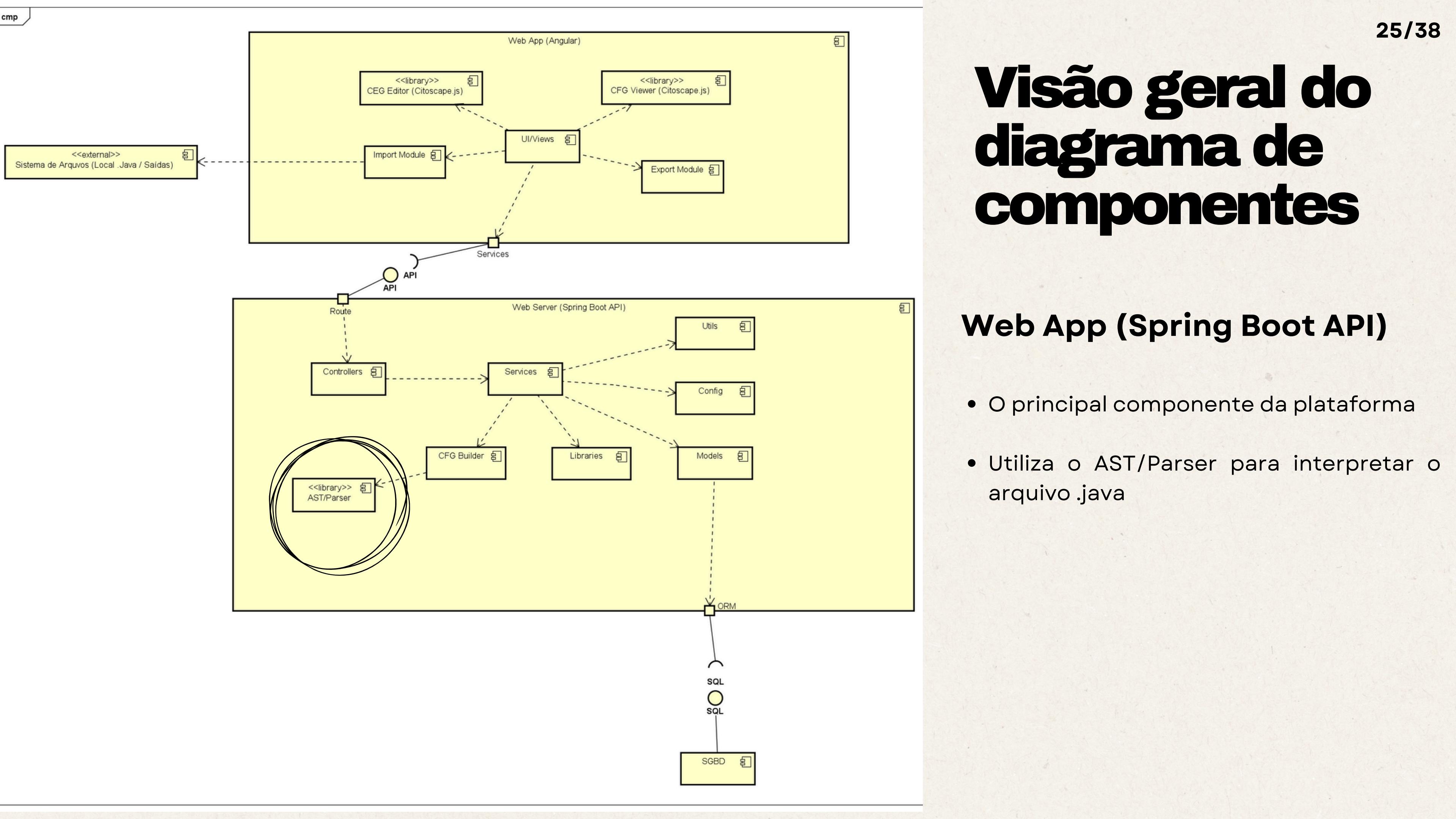
Visão geral do diagrama de componentes

Web App (Spring Boot API)

- O principal componente da plataforma



Visão geral do diagrama de componentes



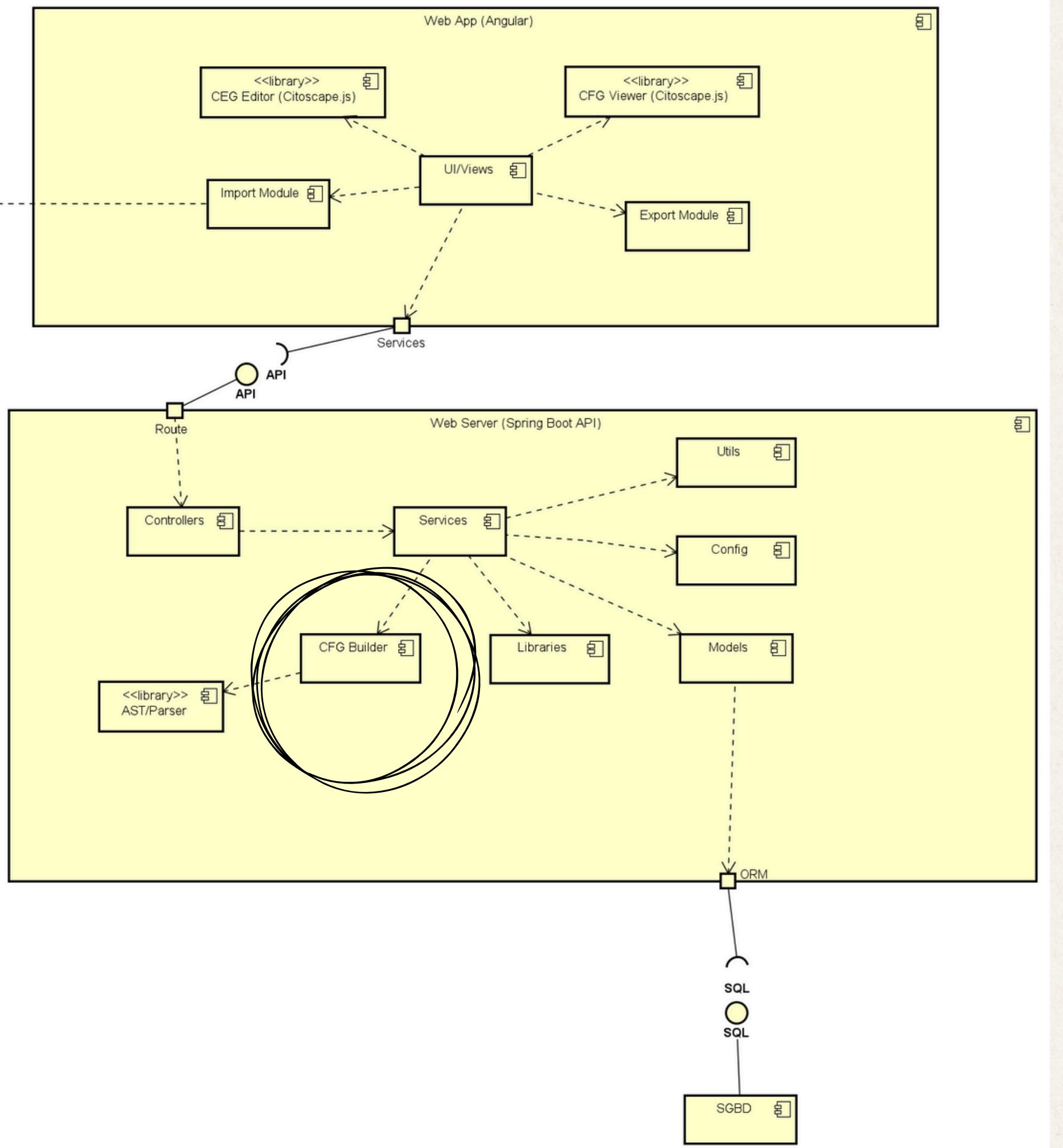
Web App (Spring Boot API)

- O principal componente da plataforma
- Utiliza o AST/Parser para interpretar o arquivo .java

Visão geral do diagrama de componentes

Web App (Spring Boot API)

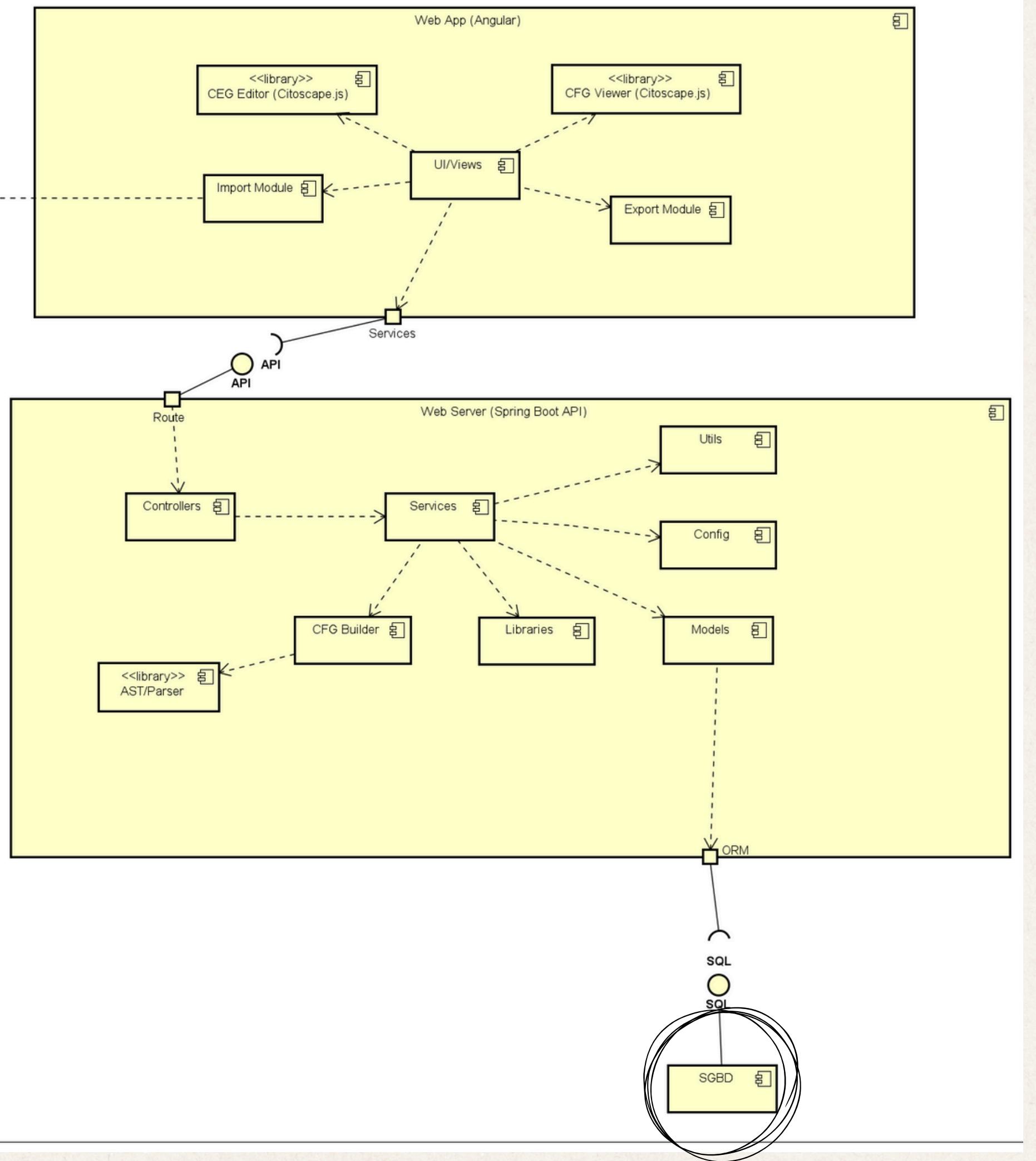
- O principal componente da plataforma
- Utiliza o AST/Parser para interpretar o arquivo .java
- Aciona o CFG Builder para construir o grafo



Visão geral do diagrama de componentes

Web App (Spring Boot API)

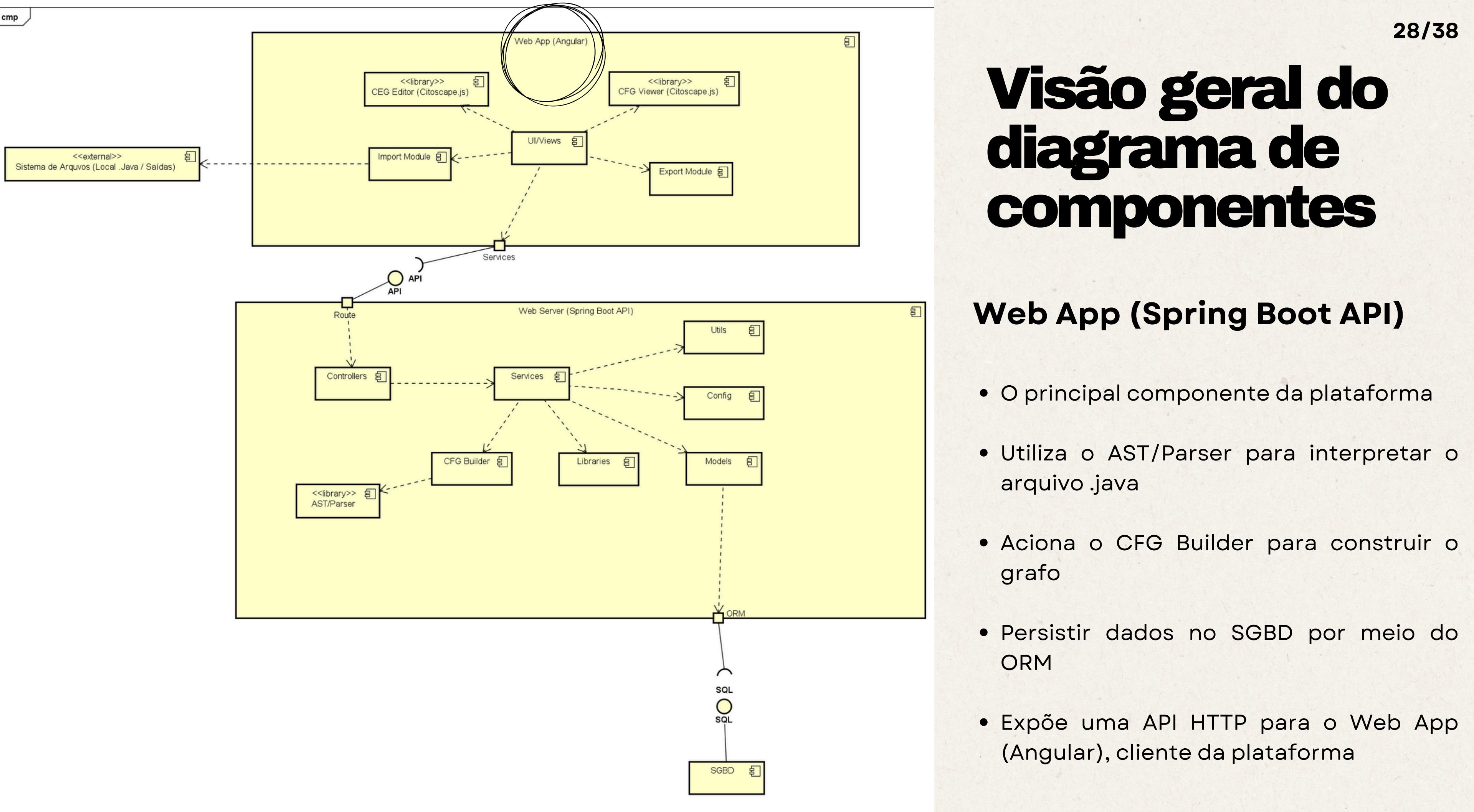
- O principal componente da plataforma
- Utiliza o AST/Parser para interpretar o arquivo .java
- Aciona o CFG Builder para construir o grafo
- Persistir dados no SGBD por meio do ORM.



Visão geral do diagrama de componentes

Web App (Spring Boot API)

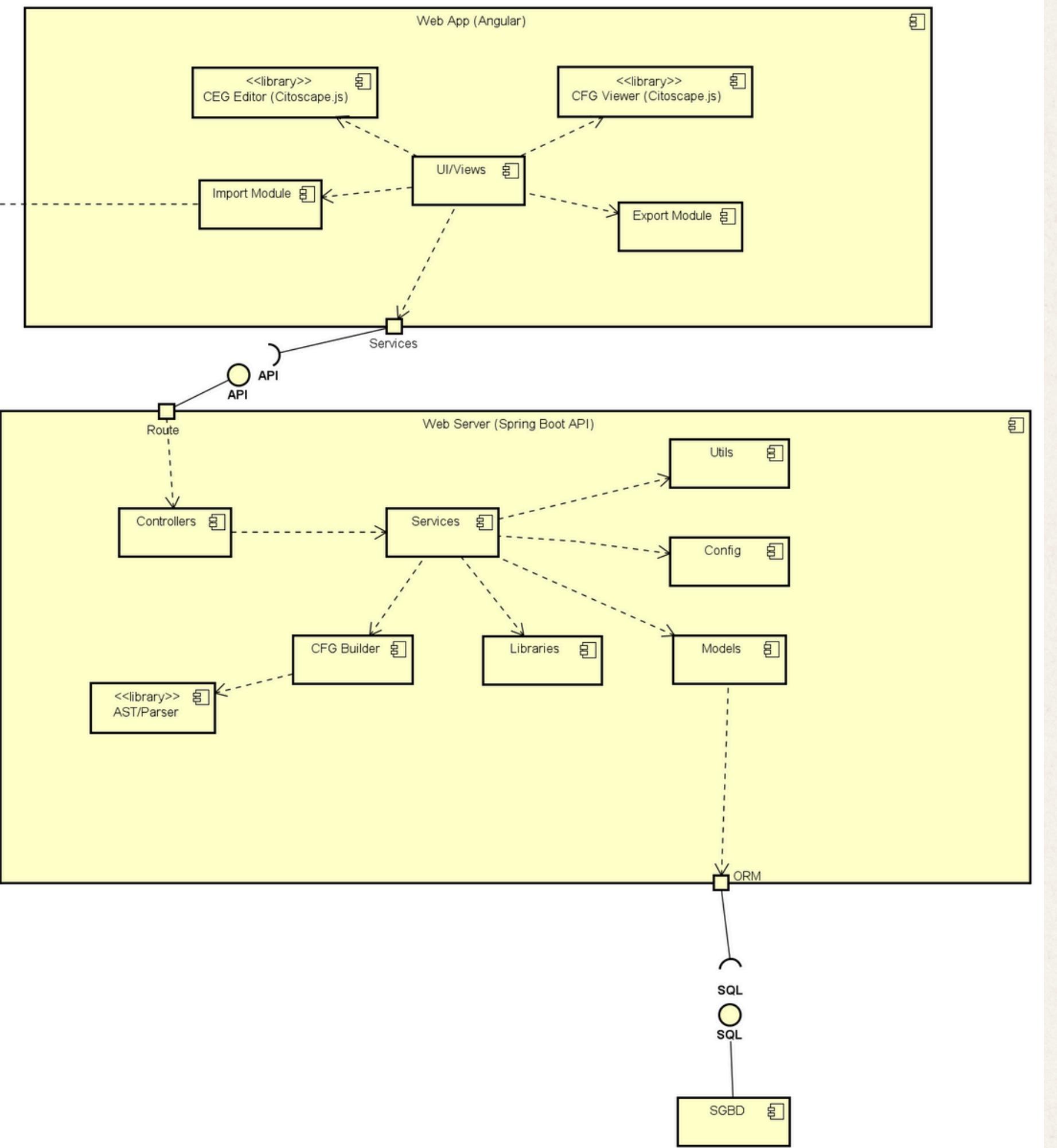
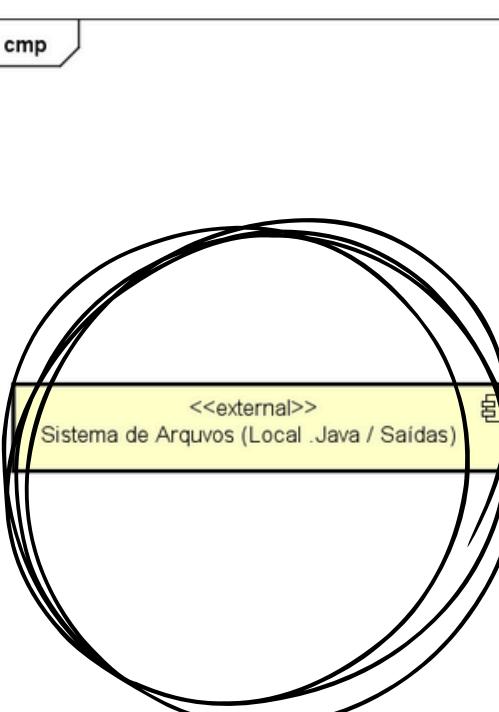
- O principal componente da plataforma
- Utiliza o AST/Parser para interpretar o arquivo .java
- Aciona o CFG Builder para construir o grafo
- Persistir dados no SGBD por meio do ORM
- Expõe uma API HTTP para o Web App (Angular), cliente da plataforma



Visão geral do diagrama de componentes

Sistema de Arquivos

- Ponto de entrada e do arquivo .java
- Permite a leitura de arquivos .java para análise e a gravação das imagens dos grafos importados.



08

Projeto de interfaces com o usuário

GraphTest Projeto GFC GCE

Análise Estrutural e Funcional de Código Java
Gere grafos, derive tabelas de decisão a partir de trechos de código.

+ Criar novo projeto Importar código

Seus Projetos

Buscar projeto... Filtrar por data

Sistema de Login Criado em 09/11/2025 8 artefatos Abrir

Calculadora Científica Criado em 07/11/2025 12 artefatos Abrir

Ordenação de Arrays Criado em 04/11/2025 5 artefatos Abrir

Busca em Grafos Criado em 31/10/2025 15 artefatos Abrir

Ações Rápidas

Gerar GFC Grafo de Fluxo de Controle a partir de código importado

Modelar GCE Grafo de Causa e Efeito para análise funcional

Gerar Tabela de Decisão Derive tabelas a partir de condições e ações

Últimos Artefatos Acessados

NOME	TIPO	ÚLTIMO ACESSO	AÇÃO
calcularMedia()	GFC	15/11/2025	Abrir
validarEntrada()	GFC	14/11/2025	Abrir
Sistema de Autenticação	GCE	13/11/2025	Abrir
Decisão de Aprovação	Tabela	12/11/2025	Abrir
processarDados()	GFC	11/11/2025	Abrir

© 2025 GraphTest. Todos os direitos reservados.
Versão 1.0.0

Documentação Suporte Repositório

Validação de Cadastro – Projeto

Sistema de validação de dados de entrada com geração automática de casos de teste

[Editar Projeto](#)[Excluir Projeto](#)[Visão Geral](#)[Artefatos de Código](#)[GFC – Fluxo de Controle](#)[GCE – Causa e Efeito](#)[Tabelas de Decisão](#)

Resumo do Projeto

[Artefatos](#)

12

[GFCs](#)

8

[GCEs](#)

5

[Tabelas de Decisão](#)

3

Progresso Geral

68%

Última atualização
2 horas atrásColaboradores
4 membros

Ações Rápidas

[Gerar GFC](#)

Criar Grafo de Fluxo de Controle a partir de código

[Importar Arquivo](#)

Adicionar arquivo Java ou código-fonte

[Modelar GCE](#)

Criar Grafo de Causa e Efeito manualmente

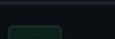
[Gerar Tabela](#)

Criar Tabela de Decisão a partir de grafo

Artefatos Recentes

[calcularMedia\(\)](#)Grafo de Fluxo
Calculadora.java

2 horas atrás

[Abrir](#)[Validação de Login](#)Grafo Causa-Efeito
Modelagem manual

5 horas atrás

[Abrir](#)[validarEntrada\(\)](#)Grafo de Fluxo
Validador.java

1 dia atrás

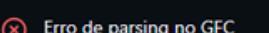
[Abrir](#)[Tabela_Cadastro_v1](#)Tabela de Decisão
Derivado de GCE

1 dia atrás

[Abrir](#)

Avisos e Validações

4 itens

[Erro de parsing no GFC](#)

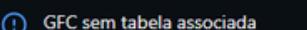
Método processarDados() contém estrutura não reconhecida na linha 47

Artefato: [Processador.java](#)[Inconsistência detectada no GCE](#)

Nó de efeito "E3" está desconectado do grafo principal

Artefato: [Validação de Login](#)[Tabela de decisão incompleta](#)

3 regras não possuem ações definidas

Artefato: [Tabela_Cadastro_v1](#)[GFC sem tabela associada](#)

O grafo validarCampos() ainda não gerou tabela de decisão

Artefato: [validarCampos\(\)](#)

GraphTest > ProjetoTeste > GFC > Calculadora > calcularMedia()

[Gerar novo GFC](#) [Voltar ao Projeto](#)

Métodos

Buscar método...

- calcularMedia()**
Calculadora • L15-28
- validarEntrada()**
Calculadora • L30-45
- processarDados()**
Calculadora • L47-68
- gerarRelatorio()**
Calculadora • L70-95
- salvarResultado()**
Calculadora • L97-112
- ordenarLista()**
Utils • L114-135
- buscarElemento()**
Utils • L137-155

Legenda

- █ Comando
- █ Decisão
- █ Laço
- █ Retorno

Estatísticas do Grafo

Total de nós	7
Total de arestas	8
Complexidade ciclomática	4

```

graph TD
    A[int x = 0] --> D{if (x > 0)}
    D -- true --> B[x++]
    D -- false --> C[while (i < 10)]
    B --> C
    C -- true --> E[sum += i]
    C -- false --> F{return sum}
    E --> I[i++]
    I --> E
    
```

Informações do Nó

█ Tipo
Laço

Linha
20

</> Código
while (i < 10)

█ Descrição
Laço: itera de 0 a 9

█ Cobertura
Coberto

</> Destacar no código

Cobertura de Comandos: 85.7% Cobertura de Desvios: 71.4%

Status: GFC gerado com sucesso

15/11/2025 14:32:18

GraphTest > ProjetoTeste > GCE > Editor de Grafo de Causa e Efeito

Gerar Tabela de Decisão **Validar GCE** **Salvar GCE**

Ferramentas
Selecionar Mover Excluir

Elementos
Causa Efeito

Operadores Lógicos
AND OR NOT

Arestas
Aresta Normal Aresta Negada

Legenda
Causa Efeito Operador

Total de nós: 4 Total de arestas: 3

O diagrama visualiza uma estrutura lógica AND. Dois nós verdes, rotulados "Usuário autenticado" e "Permissão de admin", estão conectados por uma seta que aponta para um operador laranja central rotulado "AND". Esta combinação resulta em um único nó azul final, rotulado "Acesso permitido".

Informações do Nô

- ID: 3
- Tipo: Operador
- Rótulo: AND
- Tipo de Operador: AND
- Posição: 300, 150

Controles laterais:
Zoom: 100% | Rotacionar: ↗ ↙ ↘ ↙ | Busca: ⌂ | Selecionar: ⌂

Log de Ação: • GCE salvo com sucesso 15/11/2025, 14:32:18

GraphTest > ProjetoTeste > GCE > Tabela de Decisão

Tabela de Decisão
GCE: Validação de Login

ESTRUTURA

Condições

Elemento	Regra 1	Regra 2	Regra 3	Regra 4	Regra 5	
C1 Usuário fornecido é válido	S N -	S N -	S N -	S N -	S N -	
C2 Senha fornecida é válida	S N -	S N -	S N -	S N -	S N -	
C3 Conta está ativa	S N -	S N -	S N -	S N -	S N -	
C4 Número de tentativas < 3	S N -	S N -	S N -	S N -	S N -	
+ Adicionar condição						

Ações

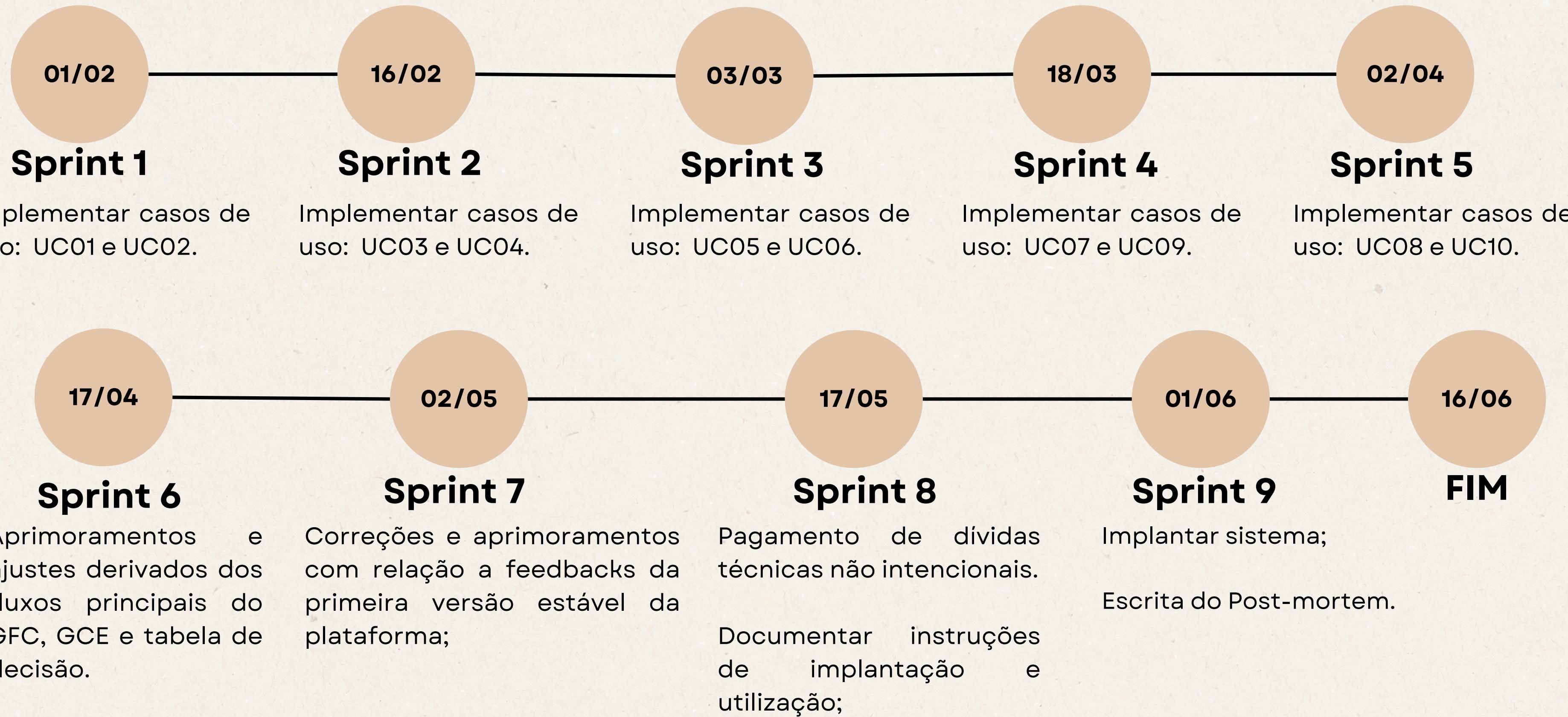
Ações						
A1 Conceder acesso (boolean)	A1 Conceder acesso	true	false	false	false	
A2 Exibir mensagem (string)	A2 Exibir mensagem	Login realizado	Usuário inválido	Senha incorreta	Conta bloqueada	Conta inativa
A3 Incrementar tentativas (boolean)	A3 Incrementar tentativas	false	false	true	false	false
A4 Bloquear conta (boolean)	A4 Bloquear conta	false	false	false	true	false
+ Adicionar ação						

Condições: 4 Ações: 4 Regras: 5 | Política: Primeira

Salvar **Reordenar Regras** **Validar Tabela**

Tabela válida

08 Cronograma 1/2026



Obrigado

ALUNOS:

LUCAS CABRAL SOARES
MARIA EDUARDA AMARAL MUNIZ

ORIENTADORES:

CLEITON SILVA TAVARES
DANILO DE QUADROS MAIA FILHO
LEONARDO VILELA CARDOSO
RAPHAEL RAMOS DIAS COSTA