**Teste Java Developer**

* **Criação backend-tgid**

Nessa aplicação está todo core do projeto, onde utilizo do Spring framework com os pacotes de:

**services**: Contém classes responsáveis por implementar a lógica de negócios do sistema.

**repositories**: Armazena interfaces que definem operações de acesso a dados para modelos específicos.

**models**: Inclui classes que representam as entidades do sistema.

**dto**: Contém classes de Transferência de Dados que são usadas para transferir dados entre as camadas do sistema.

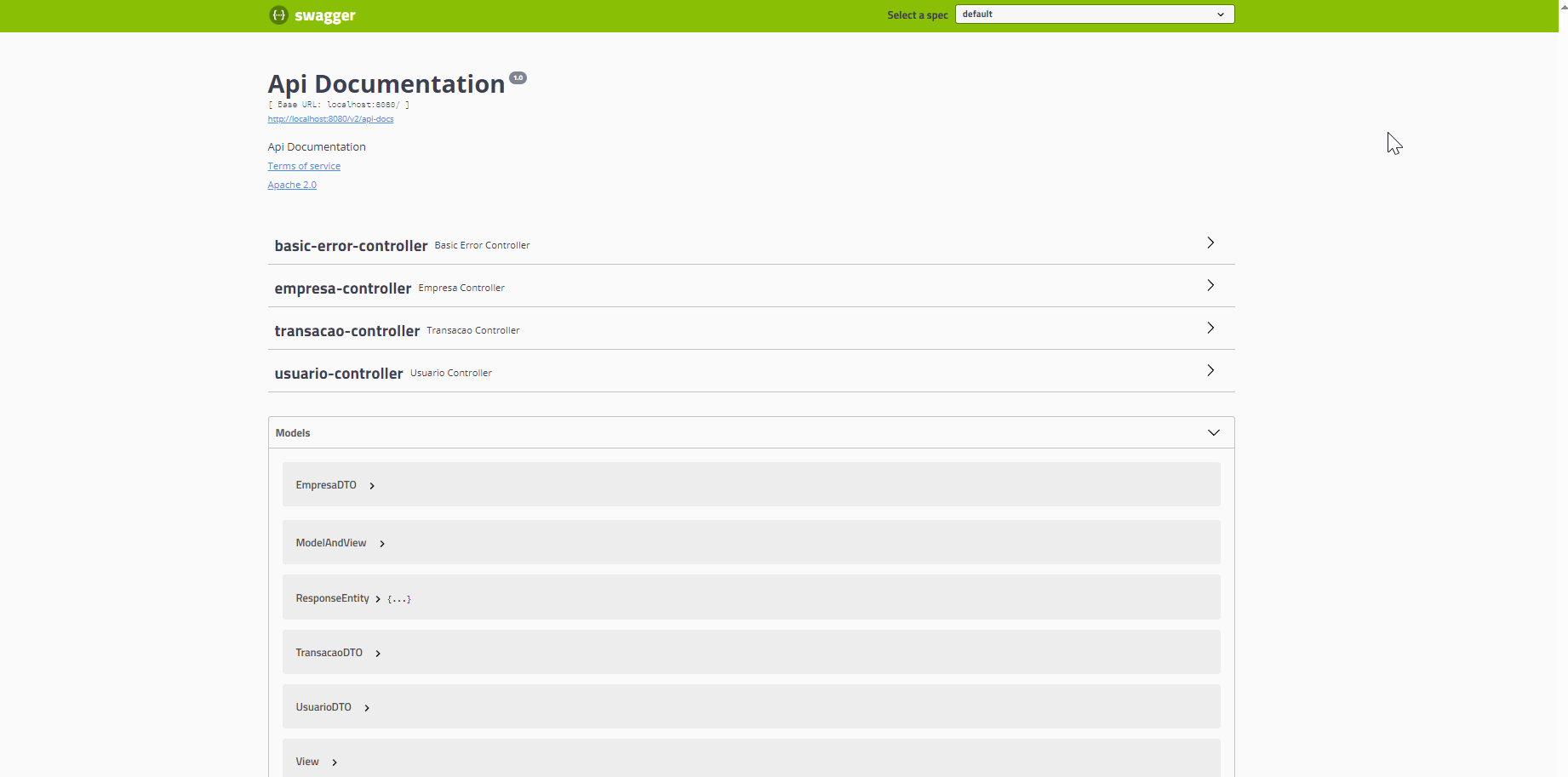
**configs**: Contém classes de configuração da aplicação.

**exceptions**

**producer**: Classe do UsuarioProducer que implementa o envio assíncrono de mensagens para filas de mensagens para chegar no microserviço de email.

**controllers:**

**Swagger implementado para documenetação da api:** http://localhost:8080/swagger-ui.html



* **Testes unitários feitos utilizando o JUnit e Mockito das UsuarioController, EmpresaController apenas devido ao tempo disponível.**

A screenshot of a computer program

Description automatically generated

* **Microserviço email**

Utilizei de um microserviço que já tinha implementado para estudo anteriormente com a utilização do RabbitMQ como um message broker para comunicação assíncrona entre os serviços. O EmailProducer responsável pelo envio do email utilizando seerviço SMTP do Gmail.

A screenshot of a computer

Description automatically generated