#### Yuri dos Reis de Oliveira

# **UFG Impacto**Documentação arquitetural

Trabalho apresentado no curso de graduação em Ciência da Computação da Universidade Federal de Goiás na disciplina Padrões de Arquitetura de Software ministrada pelo professor Jacson Rodrigues Barbosa.

#### Sumário

- 1. Visões documentadas
  - a. Visão Lógica
  - b. Visão de desenvolvimento
  - c. Visão de processo
  - d. Visão física
  - e. Visão de caso de uso
- 2. Padrão utilizado
- 3. Decisões arquiteturais
- 4. Ilustração arquitetural

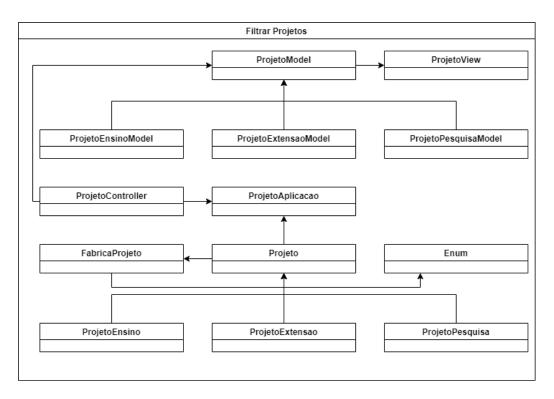
#### 1. Visões Documentadas

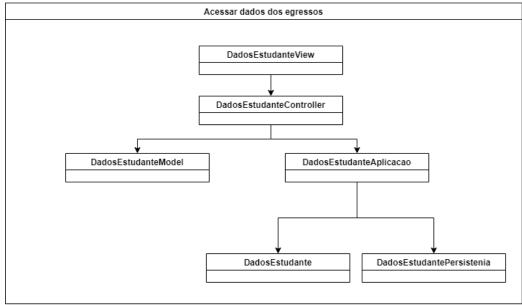
- a. Visão Lógica
  - i. Diagrama de classes

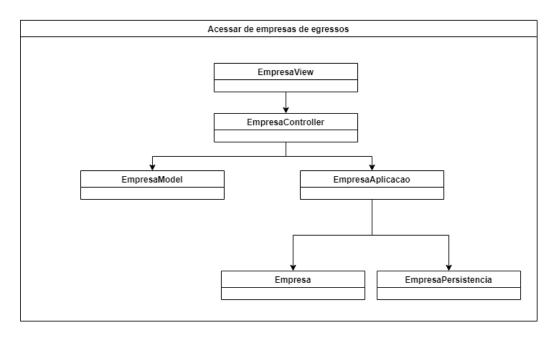
#### Model AbstractProjetoModel DadosEstudantes Empresa -id\_projeto -cidade -razao\_social -tipo\_projeto -ano\_desvinculo -capital\_social -titulo\_projeto -ano\_vinculo -total\_colaboradores -media\_relativa -id\_unidade\_projeto +getRazao\_social -sigla\_unidade\_projeto -categoria\_ingresso +setRazao\_social -nome\_unidade\_projeto -motivo\_desvinculo +getCapital\_social -coordenacao\_projeto +getCidade +setCapital\_social -resumo\_projeto +setCidade +getTotal\_colaboradores +getAno\_desvinculo +setTotal\_colaboradores +getId\_projeto +setId\_projeto +setAno\_desvinculo +getMedia\_relativa +getTipo\_projeto +setMedia\_relativa +setTipo\_projeto +getTitulo\_projeto +getCategoria\_ingresso +setTitulo\_projeto +setCategoria\_ingresso +getId\_unidade\_projeto +getMotivo\_desvinculo +setId\_unidade\_projeto +setMotivo\_desvinculo +getSigla\_unidade\_projeto +getAno\_vinculo +setSigla\_unidade\_projeto +setAno\_vinculo +getNome\_unidade\_projeto +setNome\_unidade\_projeto +getCoordenacao\_projeto +setCoordenacao\_projeto +getResumo\_projeto +setResumo\_projeto ProjetoEnsinoModel ProjetoExtensaoModel ProjeotPesquisaModel

#### b. Visão de desenvolvimento

i. Diagrama de pacotes

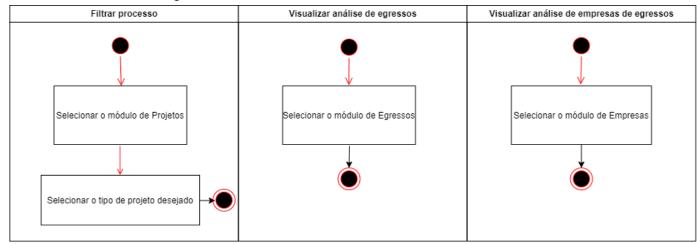






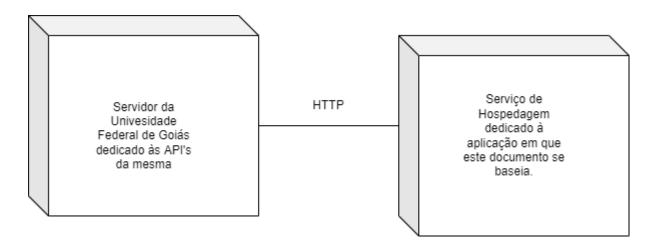
#### c. Visão de processo

i. Diagrama de atividades



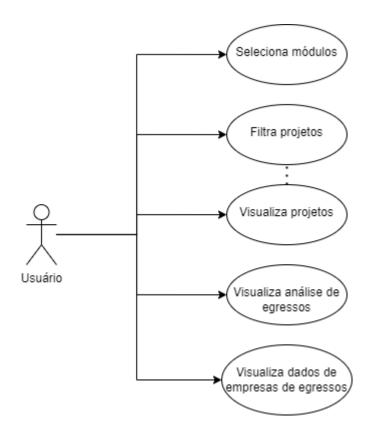
#### d. Visão física

i. Diagrama de implantação



#### e. Visão de caso de uso

i. Diagrama de caso de uso



### 2. Padrão Utilizado

Para este projeto, o padrão designado é composto por 2 estilos arquiteturais, são eles:

- MVC
- Camadas

Estes combinados promovem além de atributos de qualidade de usabilidade, utilizando do estilo arquitetural MVC e modificabilidade, utilizando do estilo Camadas, atributos de segurança, portabilidade, reúso e outros.

## 3. Decisões Arquiteturais

Decisão de Segurança	
ID do RAS	01
Tipo de Decisão arquitetural	Existencial.
Decisão	Unir o estilo em Camadas (sendo: Aplicação, Negócio, Persistência) junto ao MVC.

Justificativa	Assim conseguimos deixar que na Model tenhamos apenas os dados trazidos da view ou levados para mesma e que as regras de negócio cruciais fiquem nas camadas mais inferiores na pirâmide.
Forma de implementação	Utilizando da stack desejada e aplicando o conceito descrito na justificativa, basta que façamos este padrão de arquitetura ou semelhante.  O UFG_IMPACTO  Camadas  Aplicacao  Negocio  Persistencia  MVC  Controller  Model  View

Decisão de Modificabilidade	
ID do RAS	02
Tipo de Decisão arquitetural	Existencial.
Decisão	Design pattern Factory
Justificativa	Como temos 3 tipos de projetos, sendo de ensino, pesquisa e extensão, podemos utilizar de uma fábrica para criar tais projetos, considerando que todos herdem de projeto abstrato e que possuam ou possam vir a possuir atributos diferentes um dos outros, sendo assim vantajoso a utilização de classes diferentes e da fábrica para criá-las.
Forma de implementação	Uma forma simples se dá por criar uma classe com um método estático que recebe os atributos do objeto que será criado e também o tipo deste objeto e a partir de tal tipo se define qual classe será utilizada para se instanciar o objeto desejado.

Decisão de Disponibilidade	
ID do RAS	03
Tipo de Decisão arquitetural	Existencial.
Decisão	O sistema conta com um gerenciamento de token para acesso a API.
Justificativa	Como o token para acesso a API disponibilizada pela UFG possui limite de duração, é gerado então novos tokens automaticamente para tal acesso e assim mesmo atingindo o limite de tempo do token atual, o serviço a partir de outro token conseguirá realizar a requisição.
Forma de implementação	Implementar, além da própria requisição para API que trará os dados dos projetos, uma requisição que trará o acess_token e a partir do retorno dessa, realizar a requisição desejada inicialmente.

## 4. Ilustração Arquitetural

