Laboratrios de Informtica III

Grupo 43
62608 - Marco Sousa
93271 - Jos Malheiro
88000 - Gerson Junior

26 de Junho de 2021

Resumo

Este projeto permitiu desenvolver competncias de **Engenharia de Software**, nomeadamente *modularidade, encapsulamento, reutilizao e escalabilidade* de programas. Foi utilizada a arquitetura *Model - View - Controller* com estratgia *facade*, juntamente com a linguagem de programao *Java*.

Num contexto semelhante ao trabalho anteriormente elaborado, programou-se um **Sistema de Gesto de Reviews**, incluindo uma interao I/O dinmica, carregamento e gravao de informao, bem como uma gesto interna desta, otimizada atravs da comunicao efetiva elaborada entre as diferentes classes criadas.

1 Introduo

O presente relatrio foi redigido no mbito da unidade curricular (UC) Laboratrios de Informtica e remete-se elaborao de um projeto na liguagem de programao *Java* para um **Sistema de Gesto de Reviews**.

A construo do projeto teve como referncia a orientao dos docentes da UC e principal objetivo de desenvolver conceitos de modularidade, encapsulamento, construo de cdigo reutilizvel e otimizao atravs da escolha de estratgias para aumentar a rapidez do programa. Em adio, permitiu o aprofundamento sobre o desenvolvimento de programas na linguagem Java.

2 Estrutura do Projeto

Optou-se por um modelo que baseia-se nos princpios do MVC *(Model View Controller)*, mas em conjunto com uma estratgia *facade*, para diminuir as dependncias entre classes, encapsular e esconder aquelas que se encontram atrs, de modo a evoluirem de forma autnoma; tudo no sentido de uma melhor organizao e modularizao do cdigo.

Tomou-se, ainda, a liberdade de construir interfaces para basicamente todas as classes que constituem o projeto, de modo a aumentar a *abstrao* e tornar o cdigo mais flexvel; o programa fica menos voltil e frgil, sendo possvel variaes na sua implementao.

A arquitetura criada compreende, essencialmente, seis camadas de desenvolvimento:

- Domain
  - Estruturas de dados bsicas
  - Lgica inicial
- Services
  - Catlogos das classes base
  - Expande a lgica inicial para um maior grupo de objetos

  - Permitem estabelecer a relao entre as estruturas
  - Facilita a manipulao dos dados

- GestReviews
  - Classe agregadora
  - Utiliza a lgica do *Domain* extendida por *Services*
  - Calcula a informao a ser passada ao *Controller*

- Middleware
  - Excees criadas
  - Melhora a visualizao de possveis erros
  - Providncia um *debug* facilitado

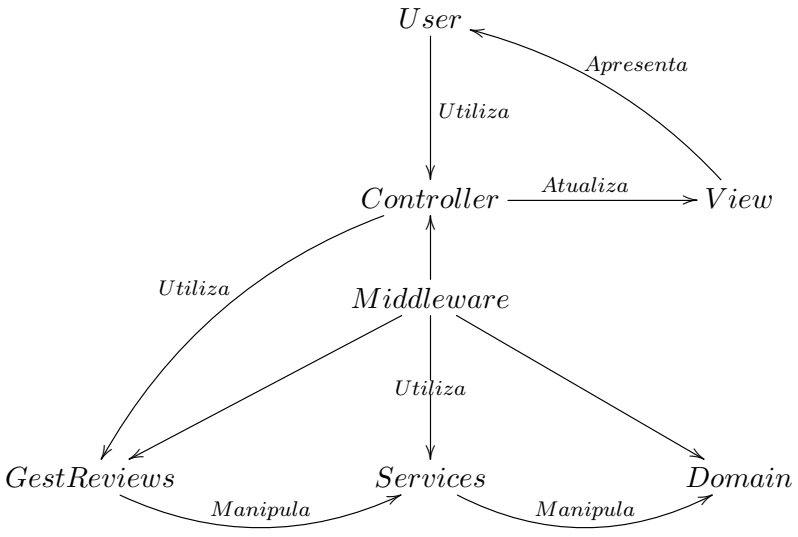
- Controller
  - Ponto de entrada do utilizador
  - Comunica com o *GestReviews* para gerar dados

- View
  - atualizado pelo *Controller*
  - Apresenta a informao ao utilizador
  - o ponto de sada da informao

A interao com o utilizador foi conseguida atravs do desenvolvimento de vrias *frames* que facilitam a visualizao do trabalho, a escolha das consultas iterativas e a insero dos valores no programa.

A execuo do programa tem quatro propsitos principais:

1. Ler ficheiros e carregar/popular a estrutura interna
2. Atribuir um resultado para cada consulta e visualiz-lo
3. Apresentar valores estatsticos dos ficheiros carregados
4. Exportar a informao para um ficheiro CSV



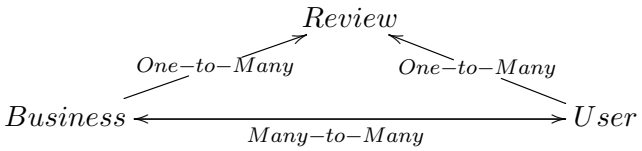
2.1 Estruturas de Dados

As estruturas de dados principais criadas para representar a informao necessria para a gesto de Reviews foram Review, Business e User.

No sentido de garantir o encapsulamento da informao, as classes criadas foram desenvolvidas de modo a que estrutura seja desconhecida por quem utiliza a API.

Assim, so disponibilizados mtodos que constroem, acedem e alteram a classe e as suas variveis de instncia (por exemplo, Construtores, *getters* e *setters*); implementado, em adio a estes mtodos, a interface *Comparable*

A relao estabelecida entre as estruturas principais pode ser descrito como:



Na package do *Domain* tambm podemos contar com classes que vo entrar no novo parmetro de estatsticas adicionado, como o caso das estruturas FileRead, auxiliar de FilesRead e Crono.

Adicionado a estes temos estruturas, como *Accumulator* e *KeySetValue*, criadas por utilidade, no mbito de melhorar a forma de guardar a informao nas outras classes.

Para todas as classes inferiores, para alm de serem criadas *interfaces* prprias, foi ainda implementada a interface *Serializable* para permitir a leitura e escrita em binrio. Devido hierarquia do projeto as classe superiores vo tambm herdar esta implementao.

2.1.1 Review

```
public class Review implements IReview, Serializable, Comparable<Review> {
        private String review_id;
        private String user_id;
        private String business_id;
        private Double stars;
        private Integer useful;
        private Integer funny;
        private Integer cool;
        private LocalDateTime date;
        private String text;
}
```

Armazena a informao de cada Review. Ver **??**.

Na sua interface **IReview** foram inseridos