



Universidade do Minho
Departamento de Informática

Sistema de Gestão de Recomendações
Laboratórios De Informática III
Grupo 54

25 de Junho, 2021



Mariana Rodrigues
(a93229)



Matilde Bravo
(a93246)



Pedro Alves
(a93272)

Conteúdo

1	Introdução	4
1.1	Descrição do Problema	4
1.2	Análise da Solução	4
2	Modelo	5
2.1	Interfaces	5
2.1.1	IUser, IBusiness e IReview	5
2.1.2	ICatalog	6
2.1.3	IStats	6
2.2	Queries	6
2.3	GestReviews	7
2.3.1	Catálogo de Users	7
2.3.2	Catálogo de Businesses	8
2.3.3	Catálogo de Reviews	8
2.3.4	Estatísticas	9
3	Apresentação	11
3.1	Interfaces	11
3.1.1	IView	11
3.1.2	IQueryView	11
3.1.3	IQueryViewFX	12
3.2	Implementação	12
4	Controlador	13
4.0.1	Queries	13
A	Estatísticas de Execução	15
A.1	Load de ficheiros	15
A.2	Queries	15
B	Diagrama de Classes	18
B.1	Modelo	19
B.1.1	Exceptions	19
B.1.2	Queries	20
B.1.3	Negócios, utilizadores e reviews	23
B.1.4	Catálogo	24
B.1.5	StarsTuple	25
B.1.6	Stats	26
B.1.7	GestReviews	27
B.1.8	Outros	27
B.2	Apresentação	28
B.2.1	View principal	28
B.2.2	Pop up de abrir ficheiro	29
B.2.3	Queries	29
B.3	Controlador	30

B.3.1	Queries	31
B.3.2	Callbacks	31
B.4	Utils	31
B.4.1	Crono	31
B.4.2	Tuples	32
B.5	Classe principal	33
C	Screenshots	34

Capítulo 1

Introdução

O presente relatório apresenta o projeto realizado no âmbito da Unidade Curricular de Laboratórios de Informática III, ao longo da segunda metade do segundo semestre, do segundo ano do Mestrado Integrado em Engenharia Informática da Universidade do Minho.

Este consiste na elaboração de um programa em Java capaz de ler ficheiros CSV contendo informação de *reviews* de negócios na plataforma Yelp, de armazenar esta informação em ficheiros de objetos, e de realizar *queries* nestes ficheiros.

1.1 Descrição do Problema

Numa fase inicial é necessário carregar 3 ficheiros CSV: *users.csv*, *businesses.csv* e *reviews.csv*, contendo, respetivamente, informação sobre utilizadores, negócios e *reviews*. Um dos principais objetivos deste trabalho foi conseguir armazenar esta informação de uma forma que fizesse sentido na linguagem Java, respeitando o modelo de programação orientada a objetos, para facilitar o mais que possível a sua apresentação e manipulação a pedido do utilizador.

1.2 Análise da Solução

O nosso grupo decidiu criar uma aplicação gráfica utilizando a *framework* **JavaFX**, organizando a aplicação segundo o modelo MVC, isolando os componentes o máximo possível. Graças à linguagem Java, conseguimos cumprir estes objetivos sem sacrificar uso de memória ou *performance* da aplicação.

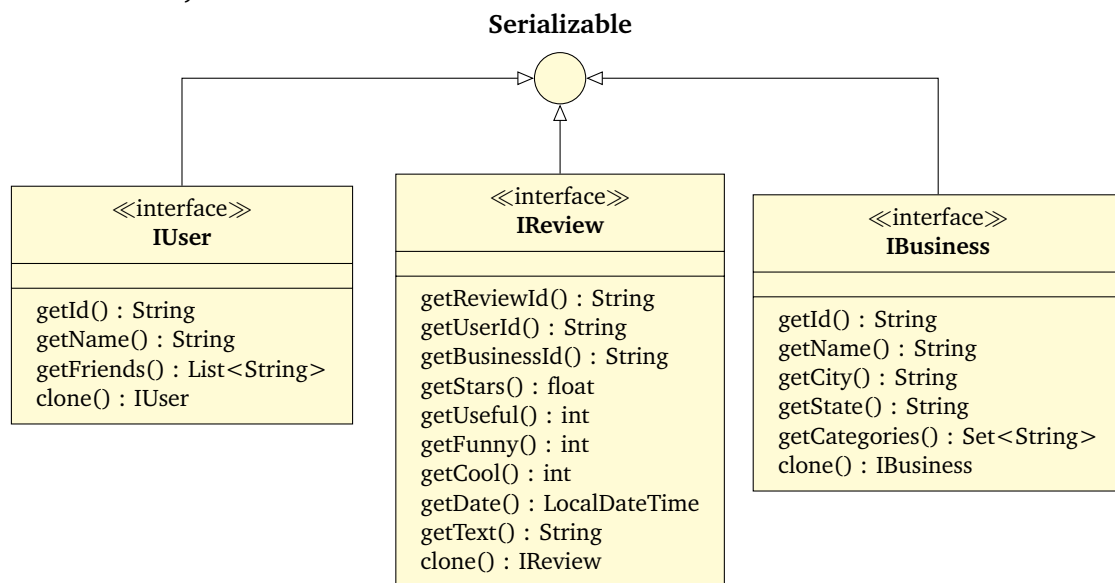
Capítulo 2

Modelo

A camada de módulos contém todas as classes e interfaces utilizadas para modelar os dados dentro da aplicação, os resultados das *queries*, e a lógica de leitura de ficheiros.

2.1 Interfaces

2.1.1 IUser, IBusiness e IReview



Estas três interfaces implementam o necessário para representar os três tipos básicos da aplicação. Qualquer código que lida com dados da aplicação utiliza estas interfaces, em vez de utilizar uma classe diretamente, permitindo a utilização de diferentes implementações. É de notar que todas estendem **Serializable**, necessário para as poder gravar num ficheiro de objetos.

2.1.2 ICatalog

Esta interface define uma estrutura de dados que pode ser utilizada para armazenar informação contida num ficheiro, criando assim um catálogo de um tipo. Particularmente importante é o método **callConstructor**, que chama o construtor do tipo que este catálogo contém com a linha fornecida.

2.1.3 IStats

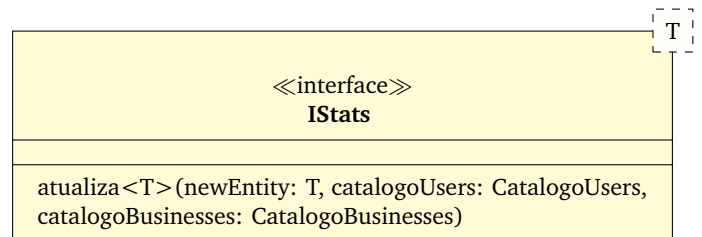
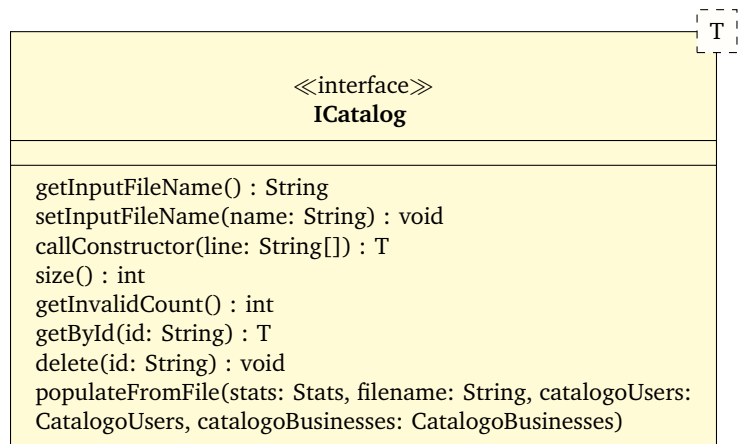
Esta interface fornece um método para atualizar as estatísticas da aplicação, e é utilizada ao preencher um catálogo a partir de um ficheiro.

2.2 Queries

Os resultados das *queries* implementam todos uma interface vazia, a **IQueryResults**, e especificam dentro de si as estruturas de dados necessárias. Como estas estruturas são todas *wrappers* simples à volta de um tipo, vamos apresentar só os tipos:

1. **Query1Results** - List<IBusiness>
2. **Query2Results** - MyPair<Integer, Integer>
3. **Query3Results** - List<MyTriple<Integer, Integer, Double>
4. **Query4Results** - List<MyTriple<Integer, Integer, Double>
5. **Query5Results** - List<MyPair<String, Integer>¹
6. **Query6Results** - Map<Integer, List<MyPair<IBusiness, Integer>>
7. **Query7Results** - List<MyTriple<String, String, Integer>
8. **Query8Results** - List<MyPair<String, Integer>
9. **Query9Results** - List<MyPair<String, Double>
10. **Query10Results** - Map<String, Map<String, Map<String, StarsTuple>>

¹Esta classe implementa alguns métodos, ver o diagrama de classes nos anexos para mais detalhes.

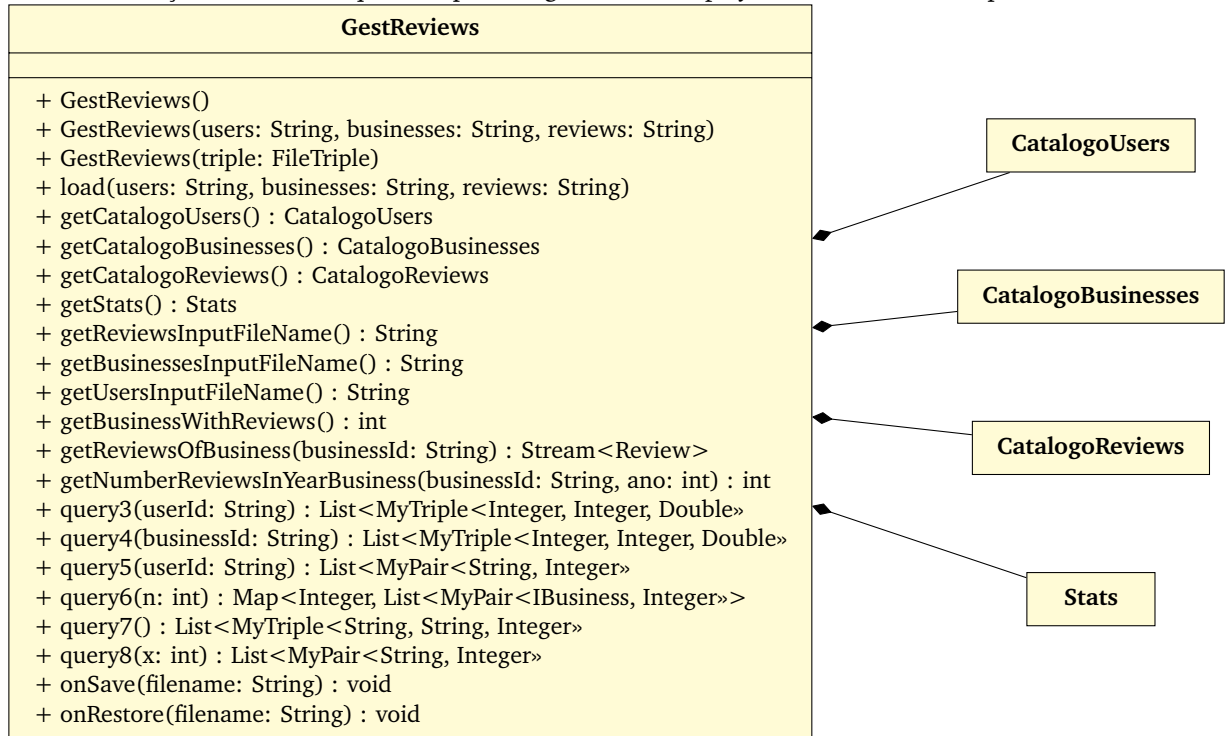


2.3 GestReviews

A leitura é feita na classe **GestReviews**, a principal classe do programa. Nesta classe, o método **load** cria três catálogos e utiliza os seus métodos de **populateFromFile** para os preencher.

Esta é também a classe de modelo principal da aplicação. É nesta classe que são implementadas as queries, e é esta que contém todos os dados necessários para a aplicação, através de outras quatro classes que iremos abordar.

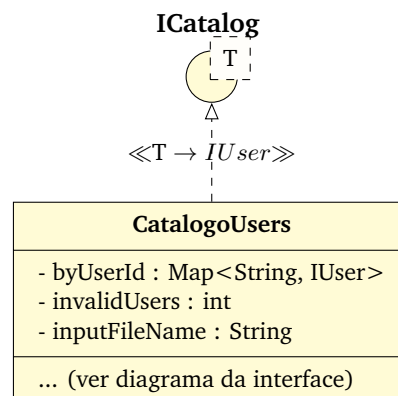
Neste caso em particular, os catálogos não são clonados antes de serem devolvidos - isto porque iria implicar a realocação de vários GB de memória sempre que se quiser aceder, só para evitar o risco de modificação. Decidimos que esta perda significativa de *performance* não valia a pena.



2.3.1 Catálogo de Users

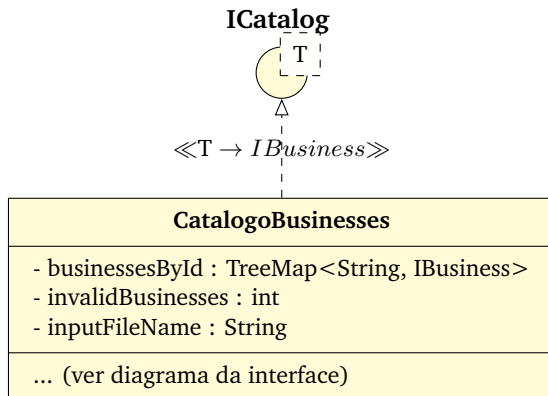
Cada um dos **user(s)** conterá um nome e um id.

Como se pode observar o **CatalogoUsers** conterá um *hash map* em que a chave corresponde ao id de cada utilizador, tendo em atenção que cada utilizador terá um único e exclusivo id. Conterá também um inteiro **invalidUsers**, que servirá para nos informar de quantos utilizadores inválidos foram obtidos ao ler um determinado ficheiro, cujo nome será guardado em **inputFileName**.



2.3.2 Catálogo de Businesses

Um **business**, à semelhança de um **user**, conterá também um identificador único, com isto o **catalogoBusiness** contém um hash map, onde a sua key será esse identificador e o value o respetivo **business**, conterá também o número de negócios inválidos lidos e o nome do respetivo ficheiro. Fora isso, face à necessidade de conseguirmos obter a lista ordenada por nome de negócios não avaliados, decidimos armazenar os negócios numa **TreeMap**. Numa fase inicial todos os negócios serão lá colocados, mas à medida que lemos o ficheiro das *reviews*, todos os *businesses* avaliados serão removidos dessa mesma árvore e colocados numa outra estrutura de dados que iremos mostrar posteriormente.

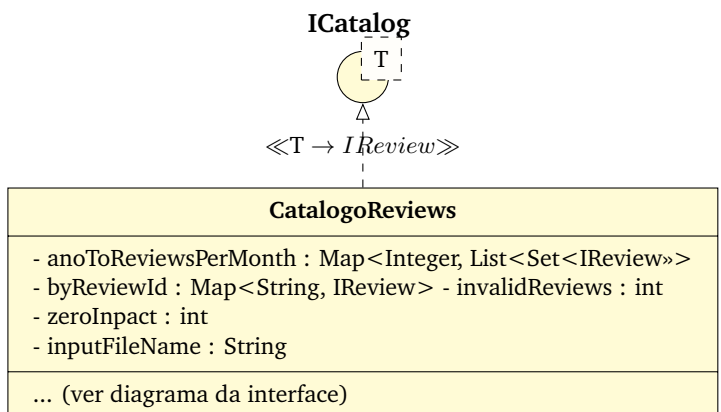


2.3.3 Catálogo de Reviews

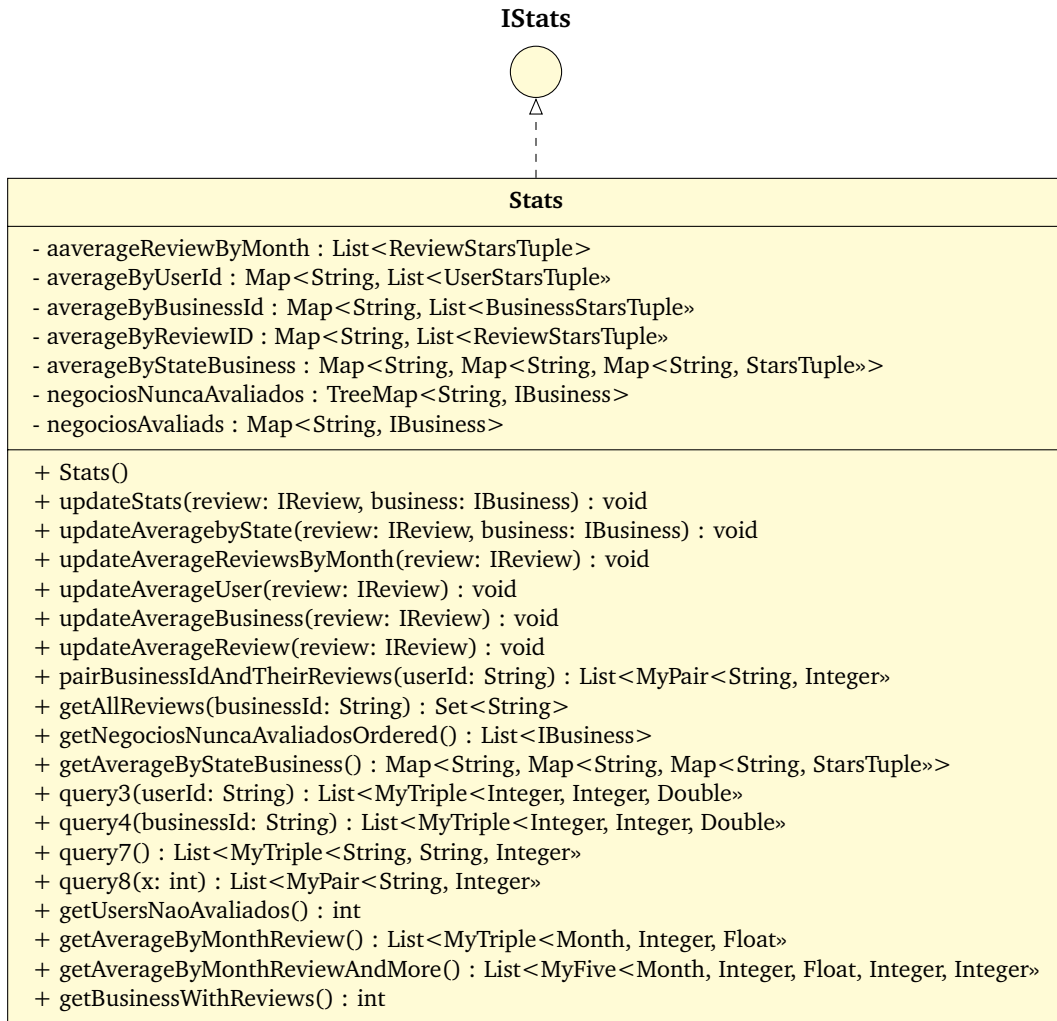
Cada **review** terá um *id* exclusivo e único, um **user_id**, um **business_id**, um número de **stars**, inteiros **useful**, **funny** e **cool** e uma **date** correspondente a quando foi realizada e um **text**.

A fim de armazenar todas as **reviews** válidas obtidas através da leitura, usamos a mesma estratégia usada tanto no **CatalogoUsers**, como no **CatalogoBusinesses**. Tendo com isto um **Hash Map** onde a key será o identificador único de cada **review** e o valor a armazenar a respetiva **review**, um inteiro que guardará o número de **review(s)** lidas e o nome do respetivo ficheiro. Fora isso teremos um contador, intitulado **zeroImpact**, que nos informará quantas **review(s)** têm zero impacto, isto é, uma **review** terá zero impacto se e só se o somatório dos campos cool, funny e useful der 0.

No campo **anoToReviewsPerMonth** temos um mapa entre um ano e uma lista com um *set* de *reviews* por cada mês do ano.



2.3.4 Estatísticas



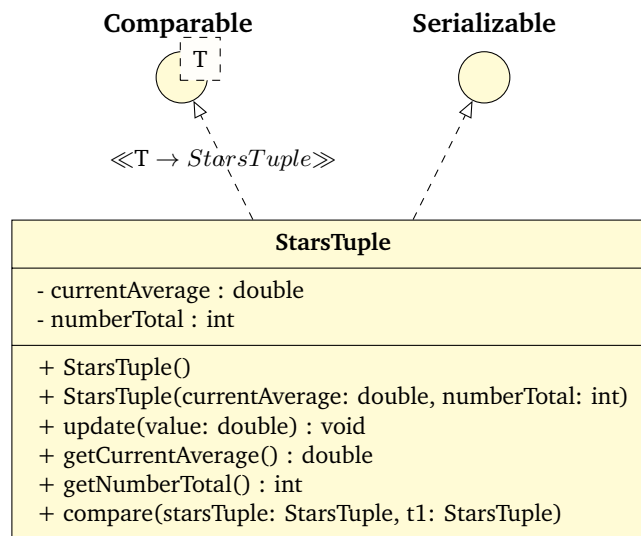
Esta classe encontra-se responsável pelo cálculo e armazenamento das estatísticas dos dados relativos aos três ficheiros lidos. Ao contrário das restantes, não se limita a armazenar os dados obtidos nas estruturas apropriadas, pois à medida que os vai lendo realiza cálculos relativos à média das estrelas. Poder-se-ia ter lido todas as *reviews* e, posteriormente, ter se feito esses mesmos cálculos, mas isso implicaria uma segunda travessia que é evitada mantendo-se um registo da média atual de estrelas com as *reviews* lidas até então, atualizando esse valor sempre que uma linha é lida. Com isto, conseguimos diminuir o tempo de execução sem custos desnecessários.

A fim de armazenar as médias das estrelas lidas em função dos seus identificadores, quer sejam *reviews*, negócios ou utilizadores, e de um determinado mês, decidimos que precisaríamos de 3 maps:

- **averageByUserId**: Onde a chave será o identificador do *user* e o seu valor será uma lista com objetos da classe **UserStarsTuple**
- **averageByBusinessId**: Onde a chave será o identificador do *business* e o seu valor será uma lista com objetos da classe **BusinessStarsTuple**
- **averageByReviewID**: Onde a chave será o identificador da *review* e o seu valor será uma lista com objetos da classe **ReviewStarsTuple**

Visto que existem 12 meses, estabelecemos que essa listas apenas terão 12 elementos, em que cada um dos índices corresponderia a um determinado mês, sendo janeiro o mês 0, fevereiro 1 e por aí adiante, com apenas um objeto da classe respetiva por mês.

Todas as classes encontradas nessas listas são sub-classes de **StarsTuple**.



É aqui que mantemos a nossa média lida até ao momento. Fora isso, cada uma dessas classes irá armazenar um **set de strings** relativos aos identificadores das reviews.

Um **ReviewStarsTuple** conterá mais dois **sets de strings**, um relativo aos identificadores dos negócios em questão e outro relativo aos utilizadores dessas mesmas *reviews* lidas.

Um **UserStarsTuple** conterá apenas mais um **set de strings** onde armazena todos os identificadores dos negócios relativos as *reviews* desse mesmo utilizador.

Já um **BusinessStarsTuple** conterá apenas mais um **set de strings** onde armazena todos os identificadores dos utilizadores relativos às *reviews* desse mesmo negócio.

Tendo a informação armazenada desta forma, conseguimos com grande eficiência obter pedidos relativos às *queries* 3, 4 e 5.

Face ao desafio proposto na *query* 10, decidiu-se que seria fundamental possuir uma estrutura de dados que fosse de estado para cidades, e de cidade para o seu respetivo **StarTuple**, intitulada **averageByStateBusiness**. Como podemos constatar, temos um Map, onde a sua chave será o nome de um dado estado e o seu valor correspondente será outro Map. Este terá como chave o nome de uma cidade e o seu valor um objeto da classe **StarsTuple**, onde será armazenada a média relativa a essa cidade desse estado correspondente.

Como outrora referido no catálogo de businesses, precisamos de armazenar quais negócios foram ou não analisados. Inicialmente, todos serão colocados num TreeMap, onde a chave corresponderá ao identificador de cada negócio e o seu valor será o negócio em questão. À medida que são lidas *reviews*, verificamos se o identificador do negócio contido nessa *review* encontra-se nessa mesma estrutura; se existir é removido e adiciona-se o mesmo a uma nova estrutura (**Map**), intitulada **negociosAvaliads**. Aqui, a estratégia é a mesma e, portanto, a chave será o identificador do negócio e o seu valor será o negócio em questão.

Constatámos que seria também fundamental armazenarmos a média das estrelas por meses. Face a essa necessidade, conteremos uma lista de **ReviewsStarsTuple**, intitulada **averageReviewByMonth**. Como outrora referido, essa mesma lista terá apenas 12 elementos em que cada um dos índices é atribuído um mês.

Com isto, é-nos possível ter acesso direto a uma grande parte da informação necessária para conseguirmos responder a todos os pedidos do utilizador.

Capítulo 3

Apresentação

A camada de apresentação é constituída pelas interfaces que permitem que o controlador a controle, bem como pelas suas implementações.

3.1 Interfaces

3.1.1 IView

A interface principal da camada de apresentação é a **IView**, definida no *package* **li3.grupo54.Views**. Esta interface possui métodos para apresentar a interface, pedir o nome dos ficheiros, apresentar erros, etc.

É de notar em especial a existência de métodos para definir *callbacks*. Estes *callbacks* são utilizados para facilitar o desenvolvimento de uma aplicação gráfica, pois são mais adequados ao seu modelo de desenvolvimento.

3.1.2 IQueryView

Esta é a interface utilizada para adicionar *queries* a uma classe que implemente **IView**. Uma *view* de uma *query* implementa esta interface e define os métodos **getName** e **getDescription** com o seu nome (por exemplo, "Query 1") e descrição, respetivamente.

O método **showResults** é utilizado para apresentar resultados depois de realizar a *query*. A classe implementadora deve fazer *cast* do objeto **IQueryResults** para o objeto concreto da *query* que lhe diz respeito.

«interface» IView
<pre>show() : void getFileTriple() : FileTriple showError(error: String, description: String) : void setOpenCallback(callback: CallbackFileTriple) : void addQuery(query: IQueryView, callback: ExecuteCallback) : void setOnSave(callback: SaveCallback) : void getSaveLocation() : File disableQueries(disable: boolean) : void setOnRestore(SaveCallback callback) : void</pre>

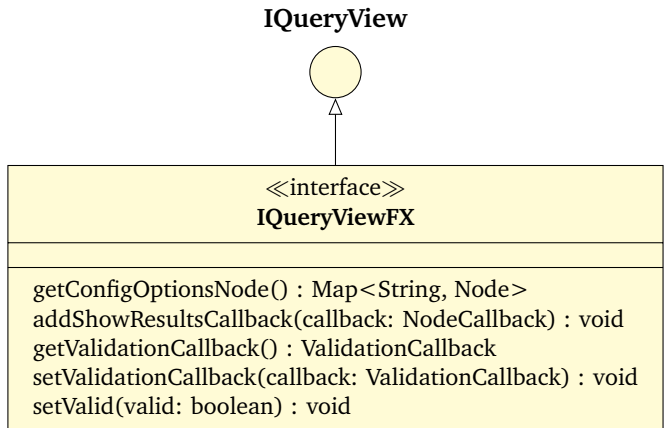
«interface» IQueryView
<pre>getName() : String getDescription() : String showResults(results : IQueryResults) : void</pre>

3.1.3 IQueryViewFX

Esta interface é uma extensão da **IQueryView** que implementa métodos específicos a uma implementação utilizando JavaFX. Se implementássemos uma interface em linha de comandos, haveria uma interface semelhante para esse tipo de interface.

O método mais importante aqui é o **getConfigOptionsNode**, pois devolve um conjunto de controlos de JavaFX que são utilizados para apresentar os parâmetros de uma *query*.

Existe também um conjunto de métodos para verificar se o *input* é válido, de modo a ativar o botão de executar apenas quando é possível.



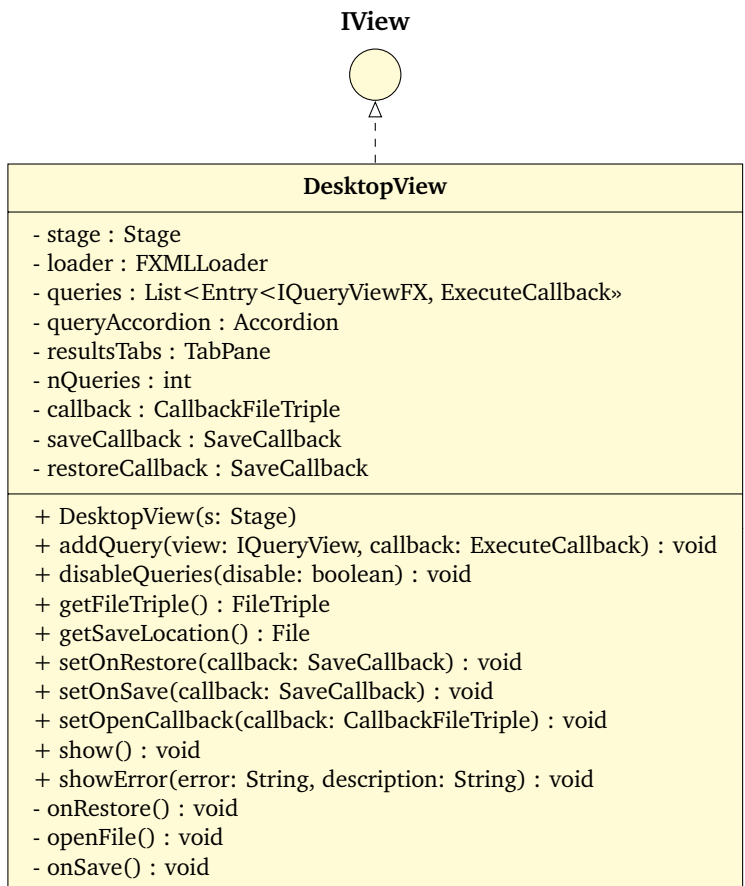
3.2 Implementação

Como foi referido na introdução, a nossa camada de apresentação foi implementada utilizando o **JavaFX**. Esta *framework* permite desenvolver aplicações gráficas de uma forma fácil, e fornece uma forma de definir as nossas interfaces declarativamente através do **FXML**, uma linguagem semelhante a HTML, opcionalmente gerada com a aplicação Scene Builder.

O FXML permite também que, através do decorador **@FXML**, indiquemos que uma variável ou método de instância de uma classe se refere a um elemento com o mesmo id definido no FXML.

A classe principal, que implementa **IView**, é a **DesktopView**, definida no *package* **li3.grupo54.Views**. Ao ser criada, esta classe utiliza o sistema de *resources* do Java para ir buscar o ficheiro **main_ui.fxml**. Quando **show** é chamado, ela itera pelas suas *queries* e adiciona-as ao **queryAccordion**, que foi definido pelo JavaFX quando carregámos o ficheiro.

Um fator importante a notar aqui é que estas *queries* têm o tipo **IQueryViewFX**, não **IQueryView**. Ao serem adicionadas, é verificado se as *IQueryViews* implementam esta interface e, caso não a implementem, são descartadas, pois não há nenhuma maneira de as apresentar.



Capítulo 4

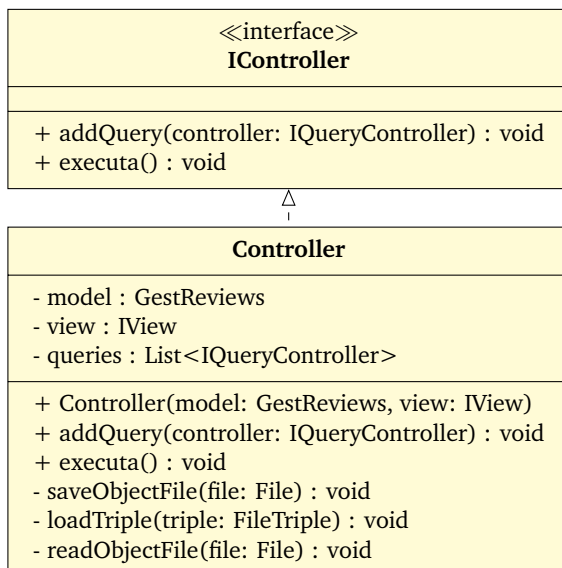
Controlador

A camada de controlador faz a união entre as camadas de modelo e apresentação. Por causa disto, o controlador acaba por ser maioritariamente composto por código de "cola", que não é muito complexo, mas que é fundamental para o funcionamento da aplicação.

Nós decidimos implementar o controlador através de um controlador principal, a classe **Controller**, que implementa uma interface **IController**, que define a API externa de um controlador.

Como o controlador implementa a lógica da aplicação, vemos que esta API é muito simples. Apenas necessita de uma forma de adicionar uma *query* (do qual iremos falar mais à frente) e de um método para executar a aplicação.

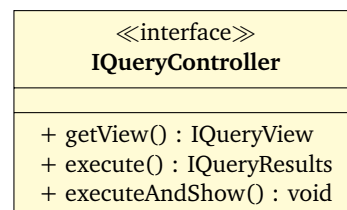
Podemos ver que o controlador implementa alguns métodos privados. Estes são utilizados como *callbacks* para ações da camada de apresentação - por exemplo, o método **loadTriple** é executado quando se carrega no botão para abrir um conjunto de ficheiros CSV.



4.0.1 Queries

As *queries* são definidas através da interface **IQueryController**. Como podemos ver pelo diagrama, esta é, também, uma interface bastante simples, servindo para fazer a conexão entre a *view* da *query* e o seu código, implementado no modelo.

Tomando como exemplo a *query* 1, vemos que é muito fácil definir um destes controladores:



```
public class Query1 implements IQueryController {
    private GestReviews model;
    private Query1View view;

    public Query1(GestReviews model, Query1View view) {
        this.model = model;
        this.view = view;
    }
}
```

```

@Override
public IQueryView getView() {
    return view;
}

@Override
public IQueryResults execute() {
    return new Query1Results(model.getStats().getNegociosNuncaAvaliadosOrdered());
}
}

```

Com estas classes conseguimos então criar uma API muito fácil de utilizar, tornando a inicialização da aplicação muito simples:

```

// Este método start é o equivalente ao método main de uma aplicação de JavaFX
public void start(Stage s) {
    DesktopView view = new DesktopView(s);
    GestReviews model = new GestReviews();
    IController controller = new Controller(model, view);

    controller.addQuery(new Query1(model, new Query1View()));
    // ... mais queries
    controller.addQuery(new Stats(model, new StatsView()));
    controller.executa();
}

```

Apêndice A

Estatísticas de Execução

A.1 Load de ficheiros

Ficheiros: Users (2.7 G), Businesses (17 M), Reviews (700 M)

Comparação de Tempos	Texec (s)	Memória (GB)
Load Friends	9.097	1.80
Load Business	0.902	0.03
Load Reviews	14.433	1.07
Total	24.432	2.90

A.2 Queries

	Argumentos recebidos: GestReviews e ...		Texec (s)	Número de dados obtidos	
Query 1	Apenas GestReviews		2.936E-6	132515	
			1.006E-6		
			1.207E-6		
Query 2	Ano 2020 Mês:		1	3.172E-6	Reviews: 7578 Business:6944
			5	5.6E-7	Reviews: 3375 Business: 3133
			9	3.16E-7	Reviews: 5194 Business: 4837
	Ano 2012 Mês:		1	8.72E-7	Reviews: 4415 Business: 3823
			6	4.67E-7	Reviews: 4524 Business: 4837
			11	7.03E-7	Reviews: 3986 Business: 3432
	Ano 2009 Mês:		3	6.05E-7	Reviews: 2357 Business: 1743
			7	4.92E-7	Reviews: 1747 Business: 1441
			12	6.95E-7	Reviews: 1544 Business: 1283
	Ano 1984	Mês:	1	0	Date not found

	Argumentos recebidos: GestReviews e ...		Texec (s)	Número de dados obtidos
Query 3	User id	"Invalido"	0	(Não existe)
		"1RCRKuHgP3FskGUVnmFdxg"	1.458E-6	4
		"J9m48CKaTlXvkqFMs7L6Kg"	4.47E-7	13
		"I4cfBoRWEhB90HBdjh8z5Q"	7.12E-7	59
Query 4	Bussiness id	"ak0TdVmGKo4pwqdJSTLwWw"	0	(Não existe)
		"6iYb2HFDywm3zjuRg0shjw"	2.775E-6	83
		"t35jsh9YnMttm69UCp7gw"	4.6E-6	7
		"Lye66-VKsO-npfyo3o7qeg"	3.6E-6	26
Query 5	User id	"Invalido"	0	(Não existe)
		"1RCRKuHgP3FskGUVnmFdxg"	6.57E-7	4
		"J9m48CKaTlXvkqFMs7L6Kg"	9.26E-7	11
		"I4cfBoRWEhB90HBdjh8z5Q"	9.09E-7	57
Query 6	N negócios	1	5.76E-7	18
		43	3.86E-7	742
		123	4.28E-7	2102
		450	4.97E-7	7661
		1000	5.06E-7	16526
		1500	5.35E-7	24237
		4000	5.627E-7	58409
Query 7	Apenas GestReviews		2.089E-6	900
			1.0396E-6	
			1.086E-6	
Query 8	N users	1	2.014E-6	1
		43	9.22E-7	43
		1000	1.254E-7	1000
		1500	8.41E-7	1500
		4000	8.85E-7	4000
		10000	1.021E-7	10000
		100000	9.1E-7	100000
		600000	8.02E-7	536836
Query 9	"6iYb2HFDywm3zjuRg0shjw Top:	5	7.07E-7	5
		23	8.06E-7	23
		50	3.74E-7	50
		75	3.52E-7	75
		100	2.3E-7	83
	"t35jsh9YnMttm69UCp7gw Top:	1	0.002383	1
		1	1.58E-7	1
		3	4.17E-7	3
		7	3.08E-7	7
		10	2.47E-7	10
	"Lye66-VKsO-npfyo3o7qeg Top:	10	0.001157	10
		5	6.18E-7	5
		10	3.76E-7	10
		15	5.4E-7	15
		30	7.33E-7	25
Query 10	Apenas GestReviews		3.134E-6	27952
			1.309E-6	
			5.94E-6	

Todos os resultados aqui apresentados foram obtidos através de um portátil com as seguintes especificações:

OS	Arch Linux x86_64
Host	20TACTO1WW ThinkPad E14 Gen 2
CPU	11th Gen Intel i7-1165G7 (8) @ 4.700GHz
Memory	SoDIMM de 16 GB DDR4 3.200 MHz
First Solid State Drive	512GB SSD NVMe

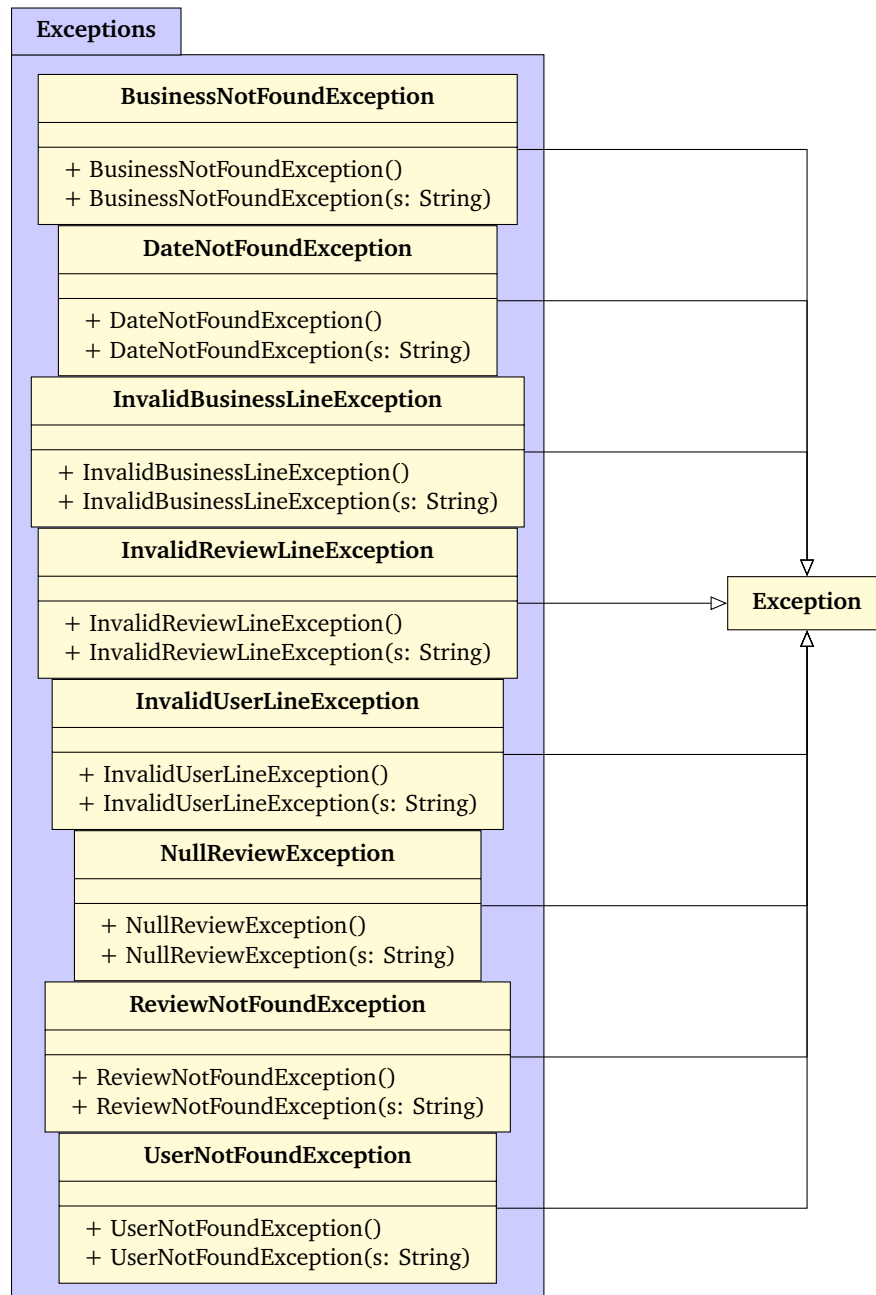
Apêndice B

Diagrama de Classes

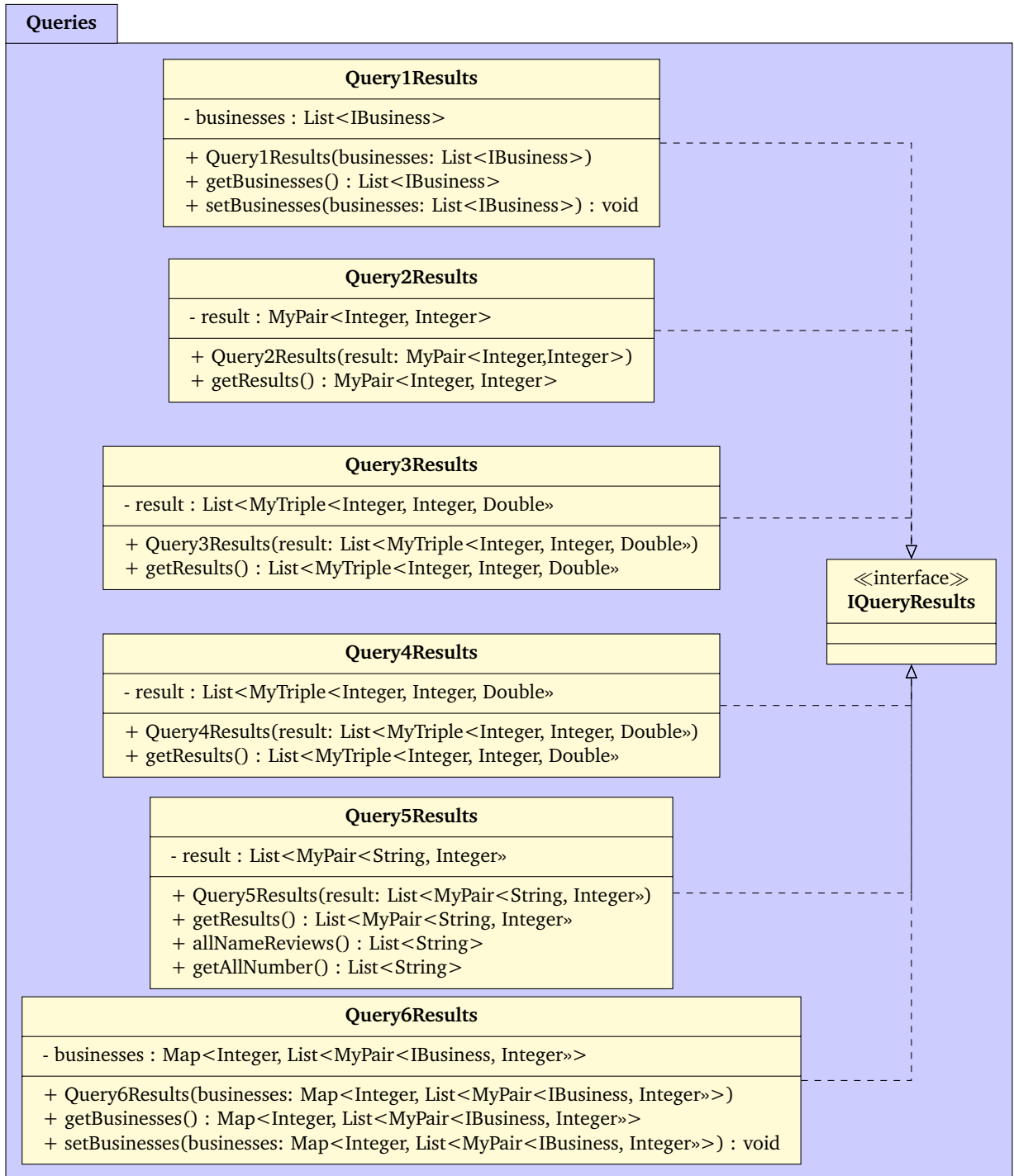
De modo a caber nas páginas do formulário, o diagrama será apresentado por secções relacionadas.

B.1 Modelo

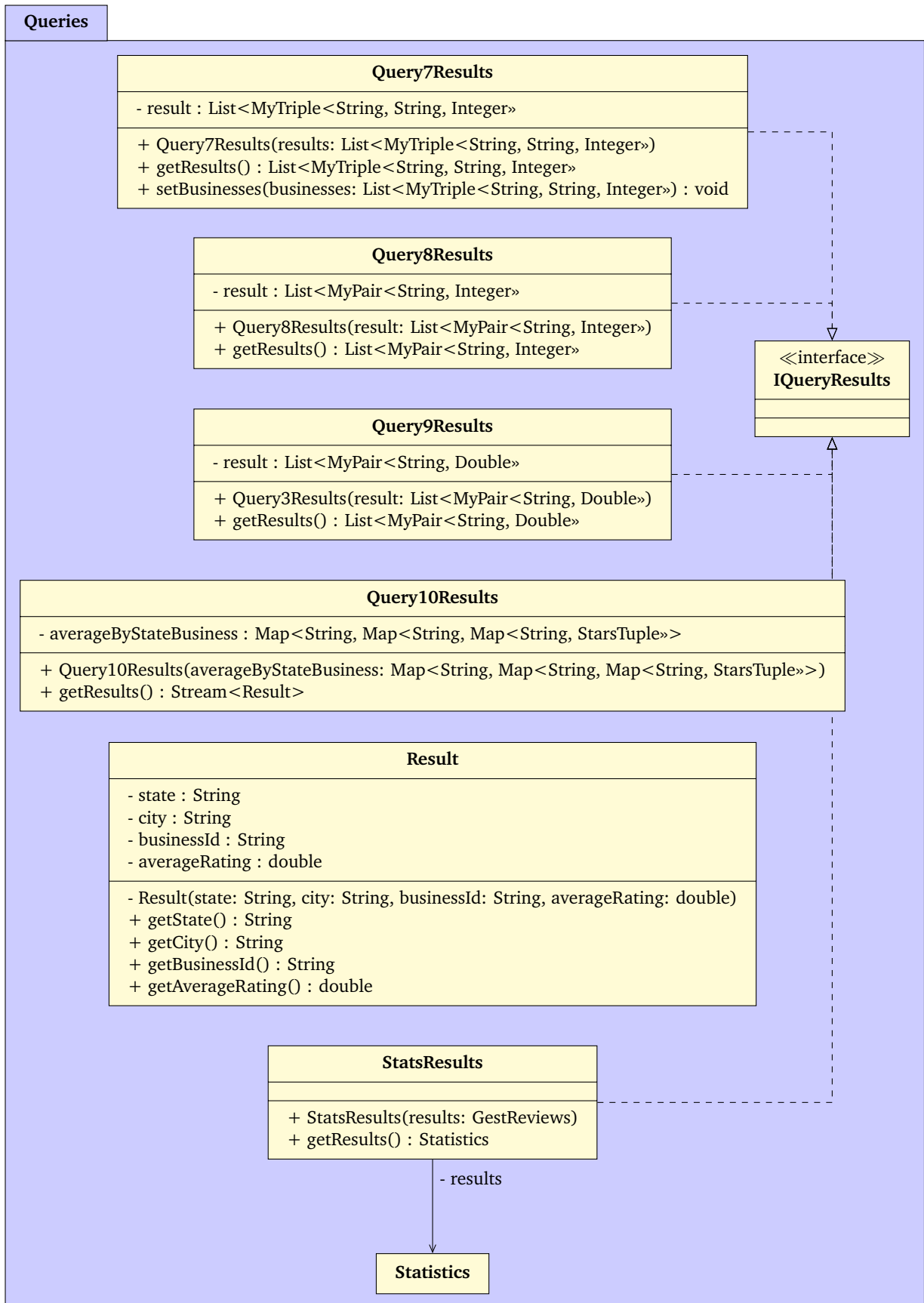
B.1.1 Exceptions



B.1.2 Queries

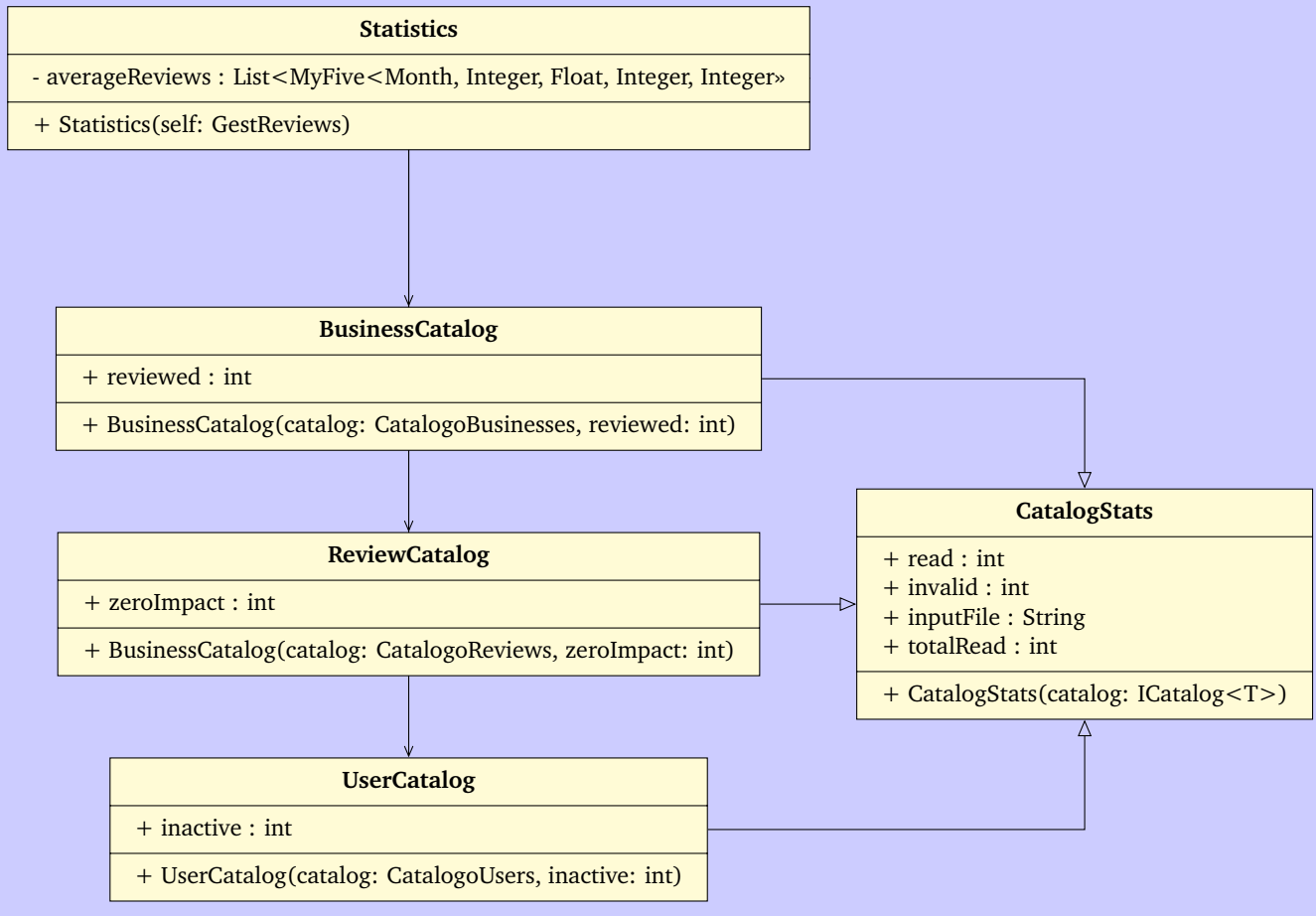


(Continua na próxima página)



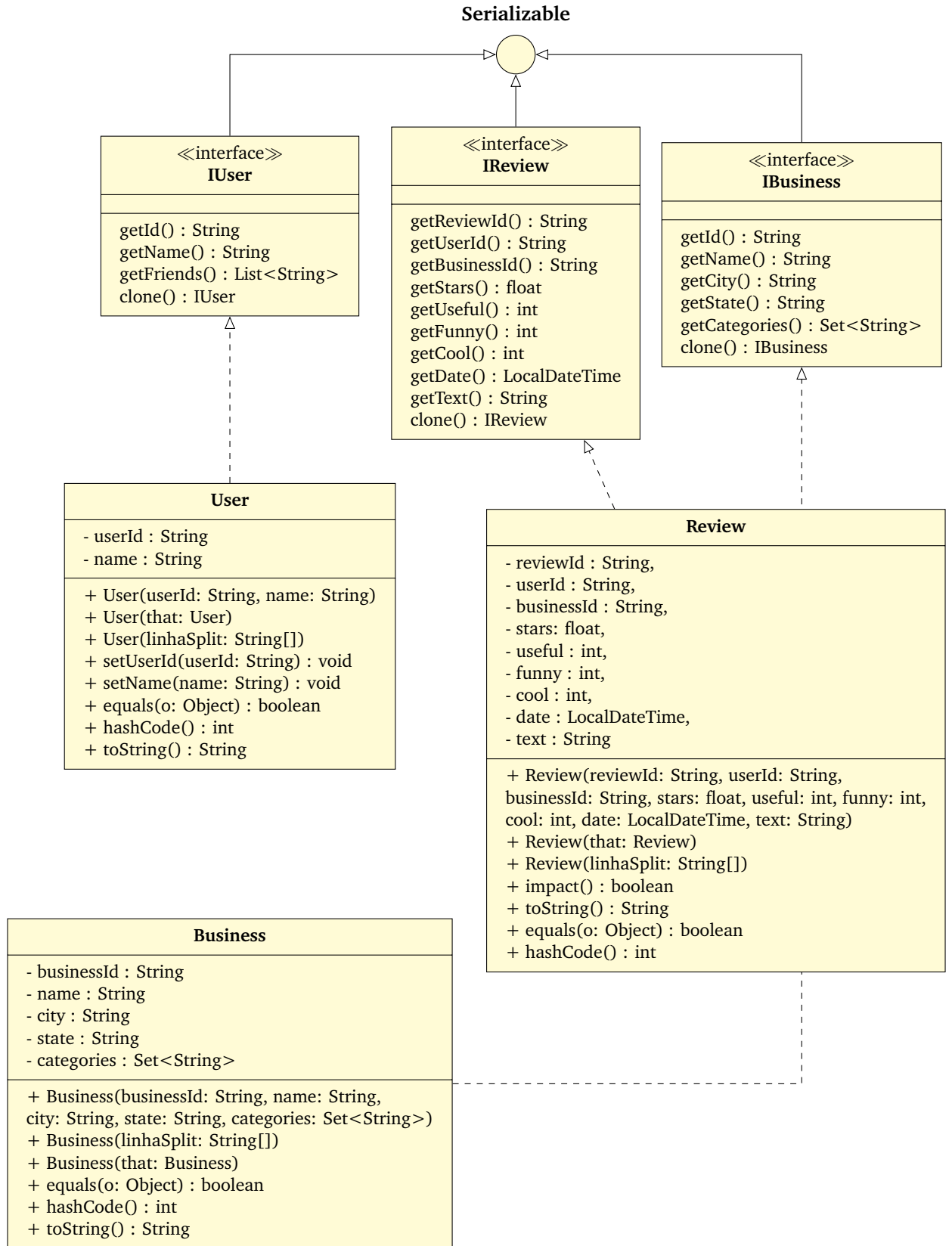
(Continua na próxima página)

Queries

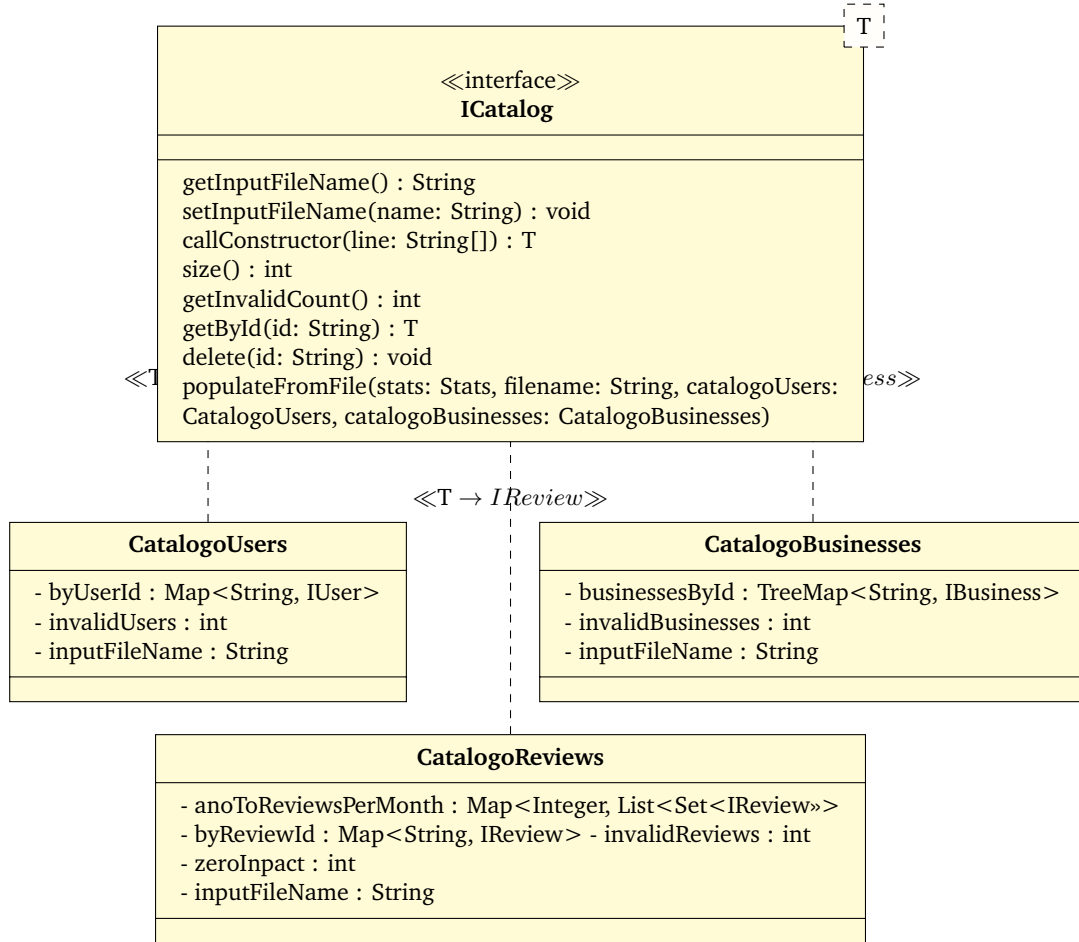


Nota: Os campos são públicos pois são final, e portanto impossíveis de modificar.

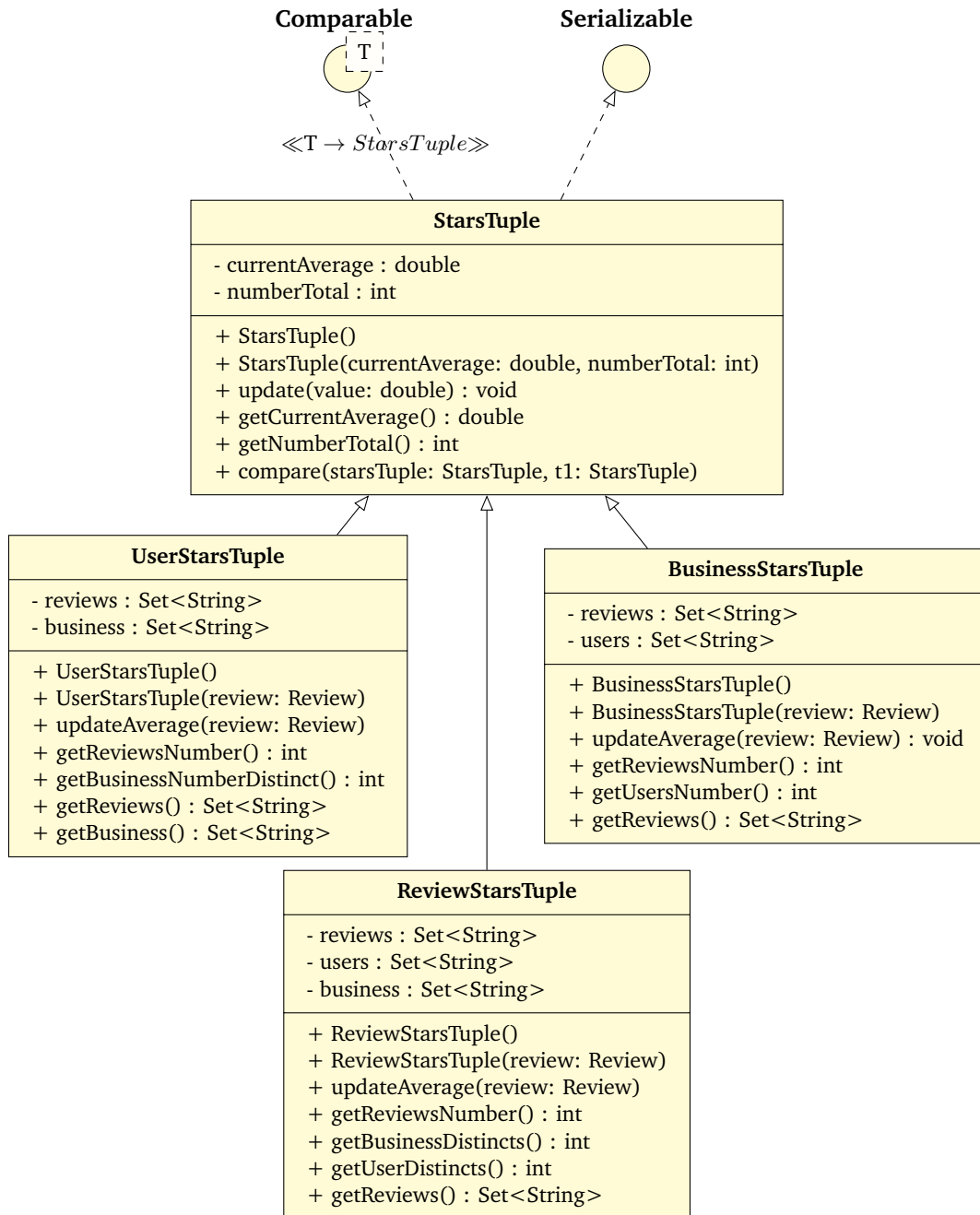
B.1.3 Negócios, utilizadores e reviews



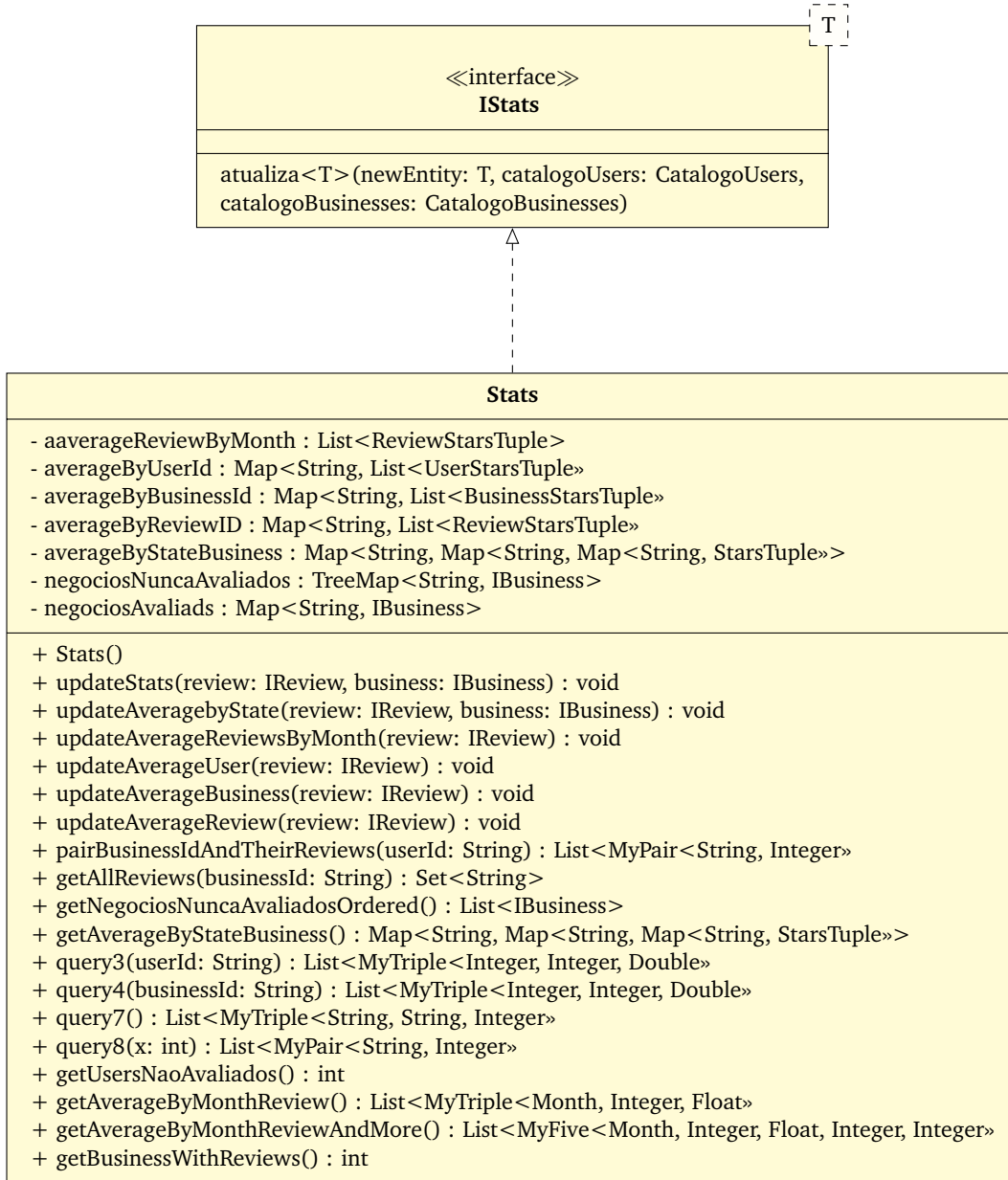
B.1.4 Catálogo



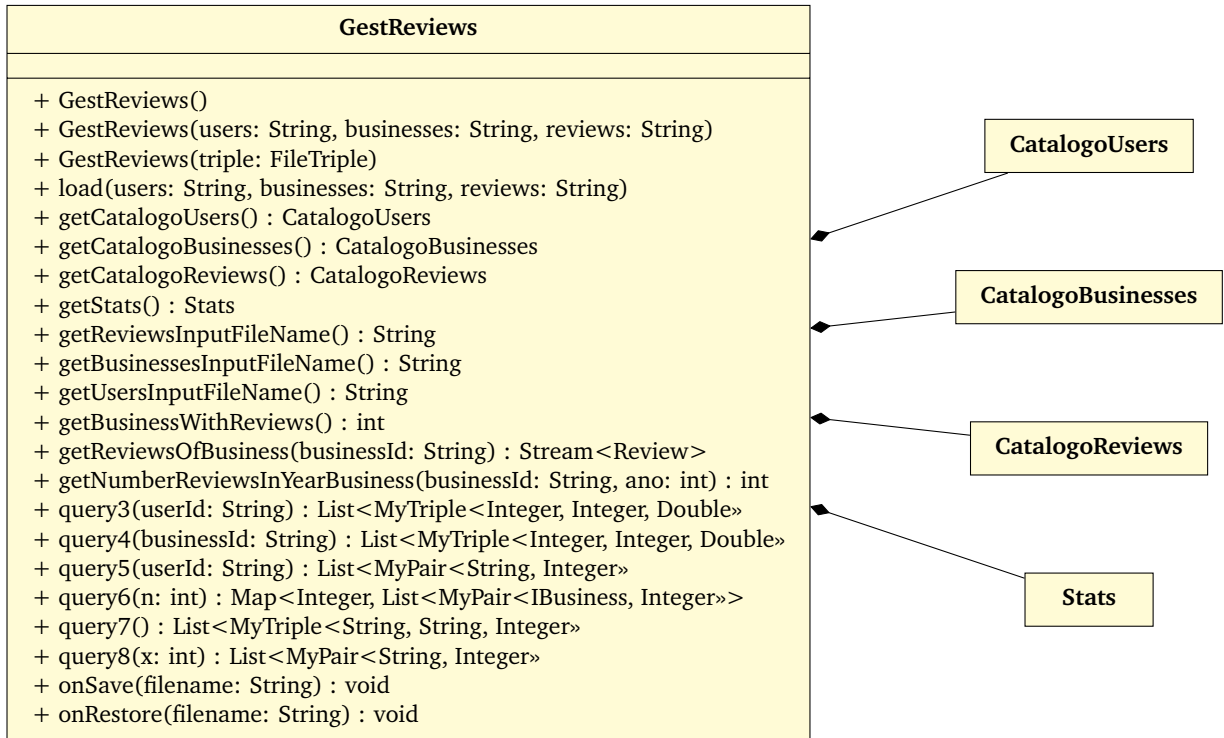
B.1.5 StarsTuple



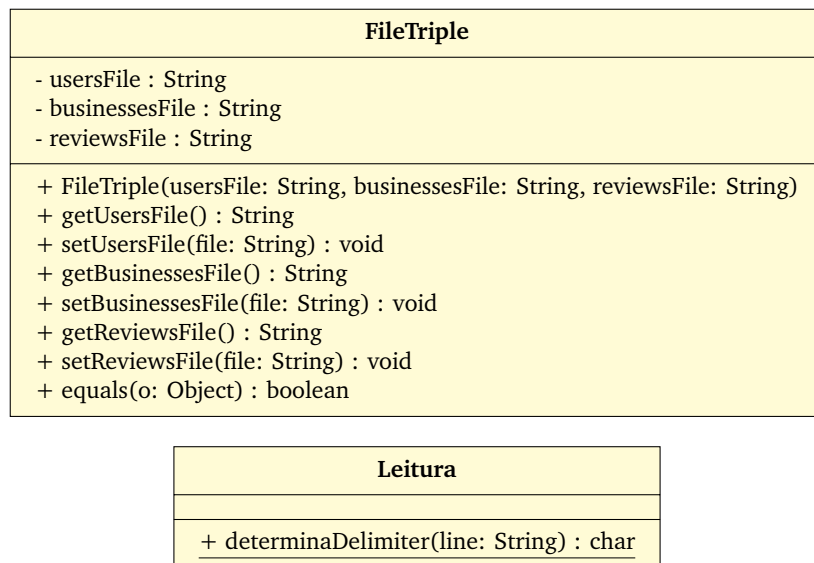
B.1.6 Stats



B.1.7 GestReviews

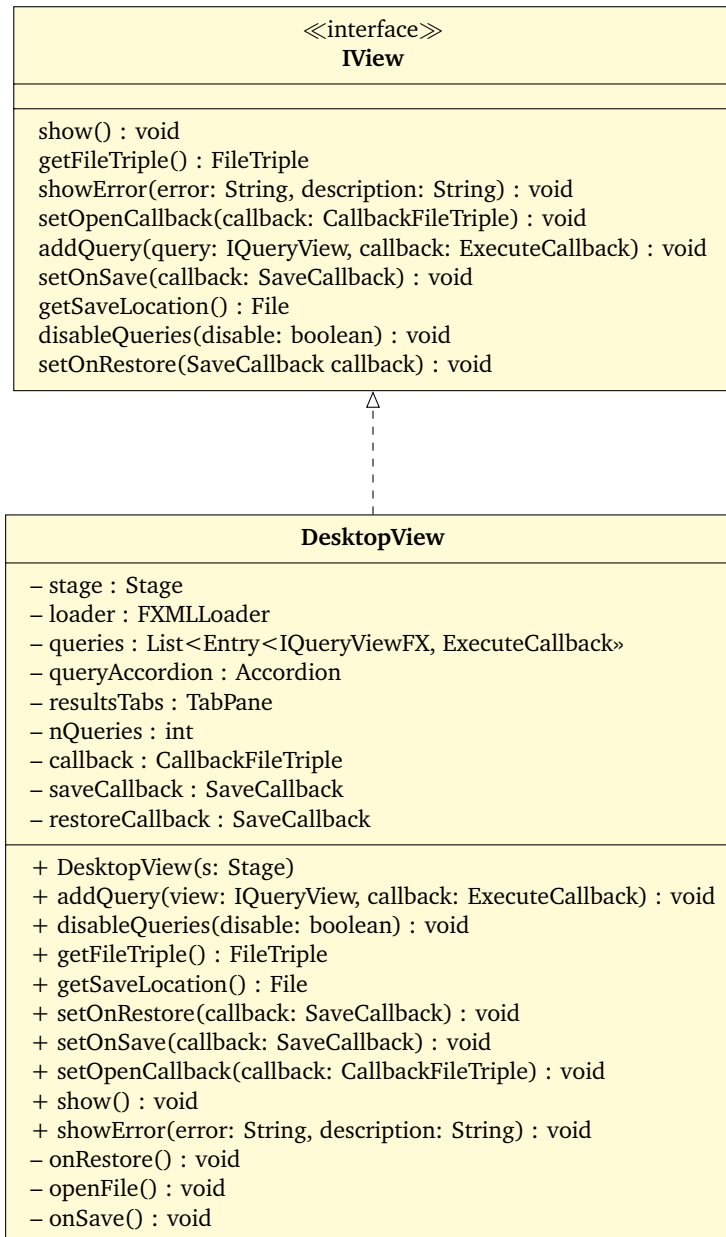


B.1.8 Outros

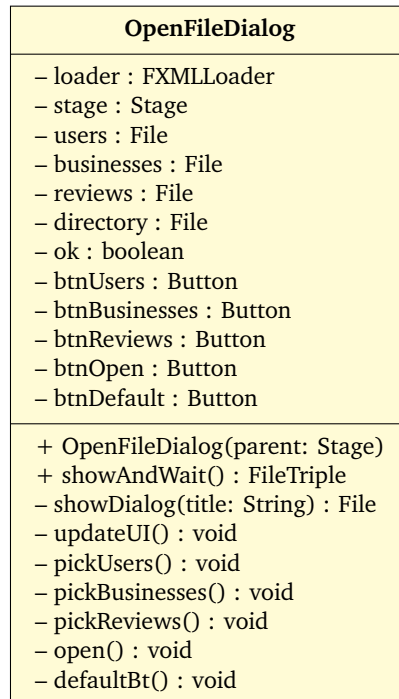


B.2 Apresentação

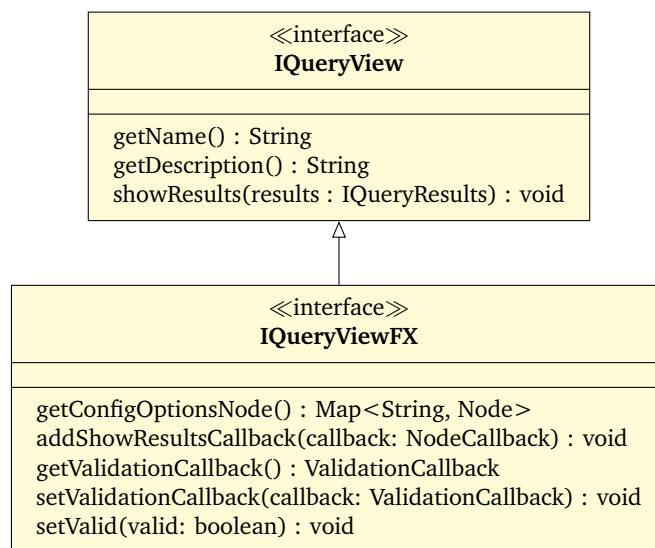
B.2.1 View principal



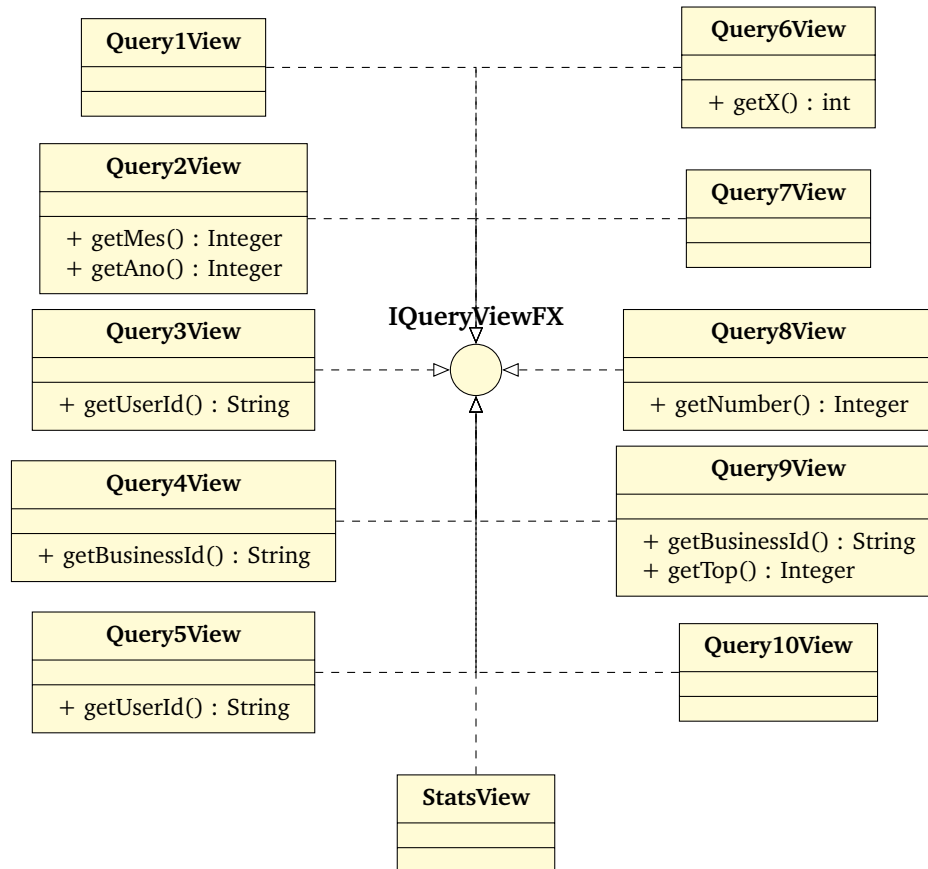
B.2.2 Pop up de abrir ficheiro



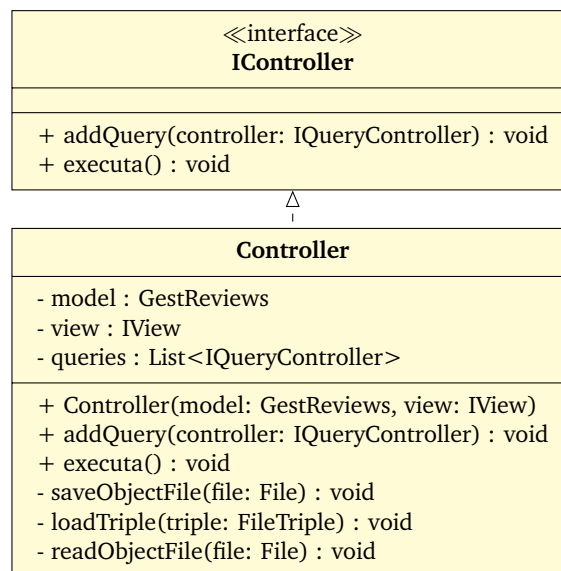
B.2.3 Queries



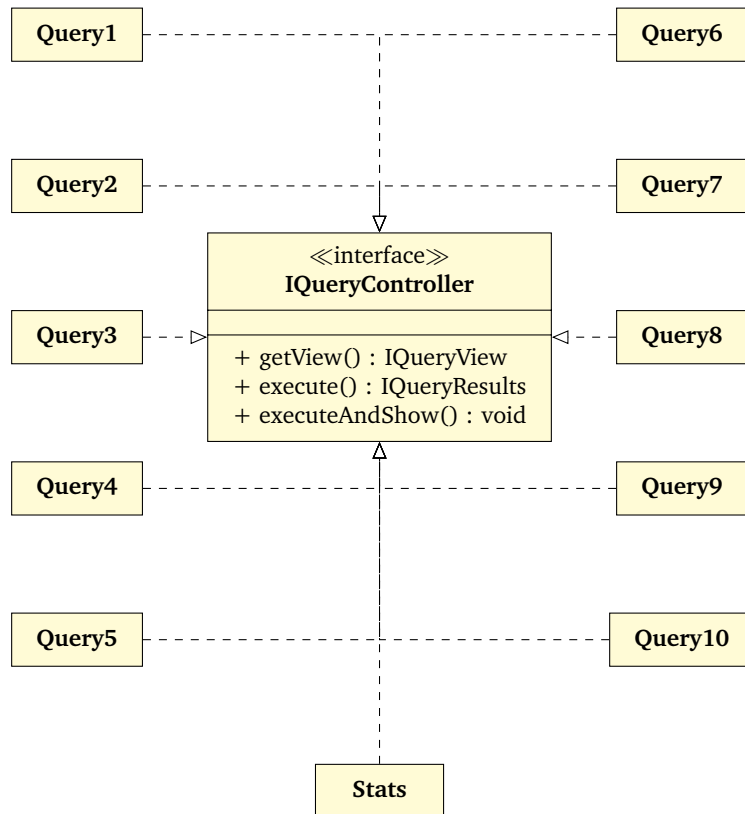
Para simplificar os diagramas, iremos apenas apresentar os getters/setters de campos específicos às views, caso aplicável.



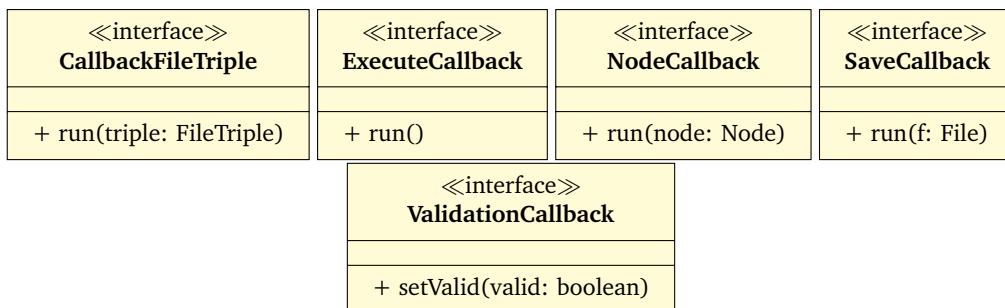
B.3 Controlador



B.3.1 Queries

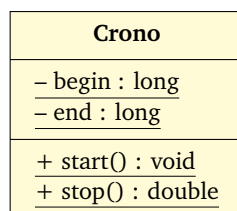


B.3.2 Callbacks

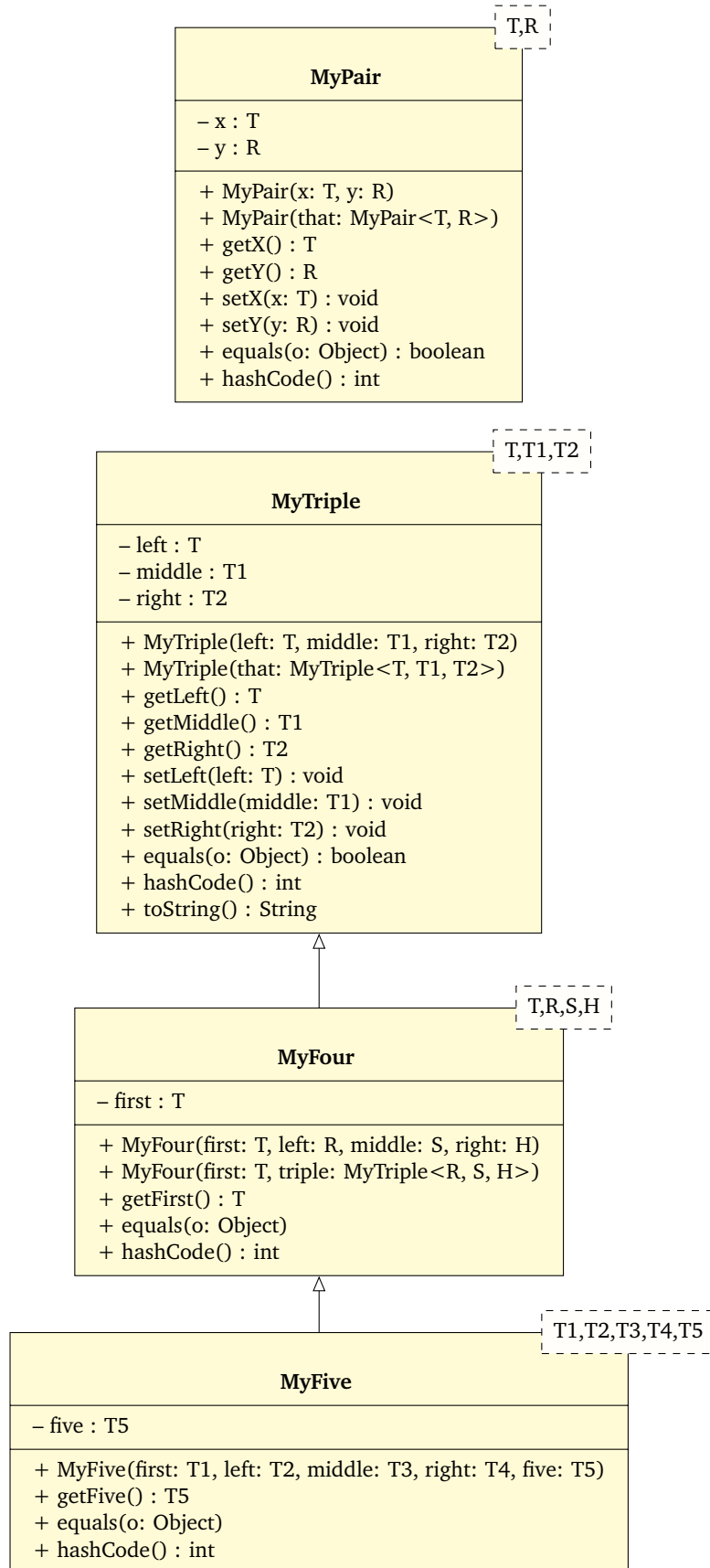


B.4 Utils

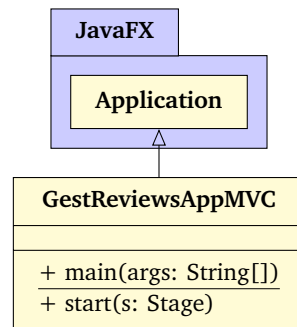
B.4.1 Crono



B.4.2 Tuples



B.5 Classe principal



Apêndice C

Screenshots

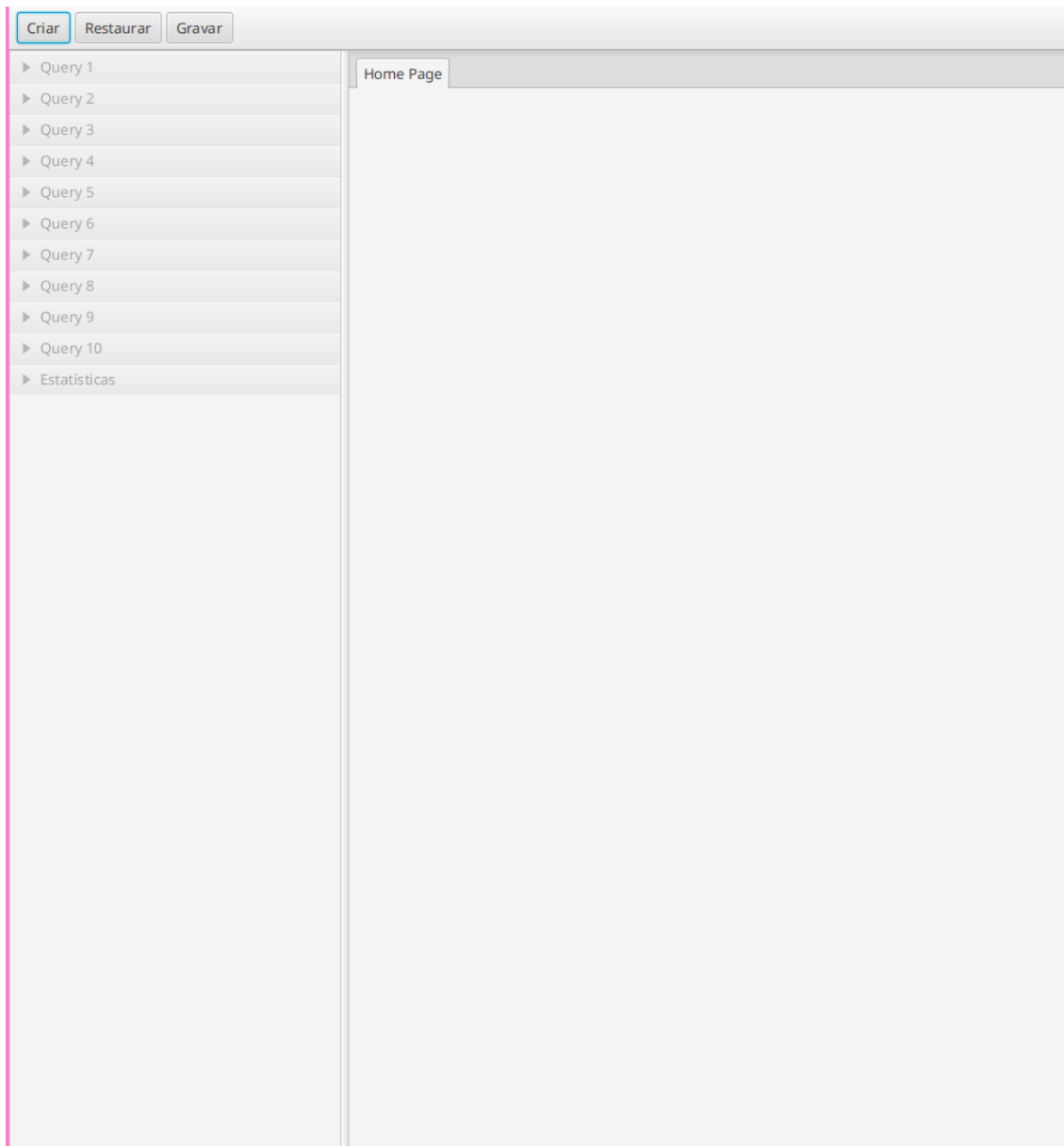


Figura C.1: Estado inicial da aplicação, antes de abrir algo

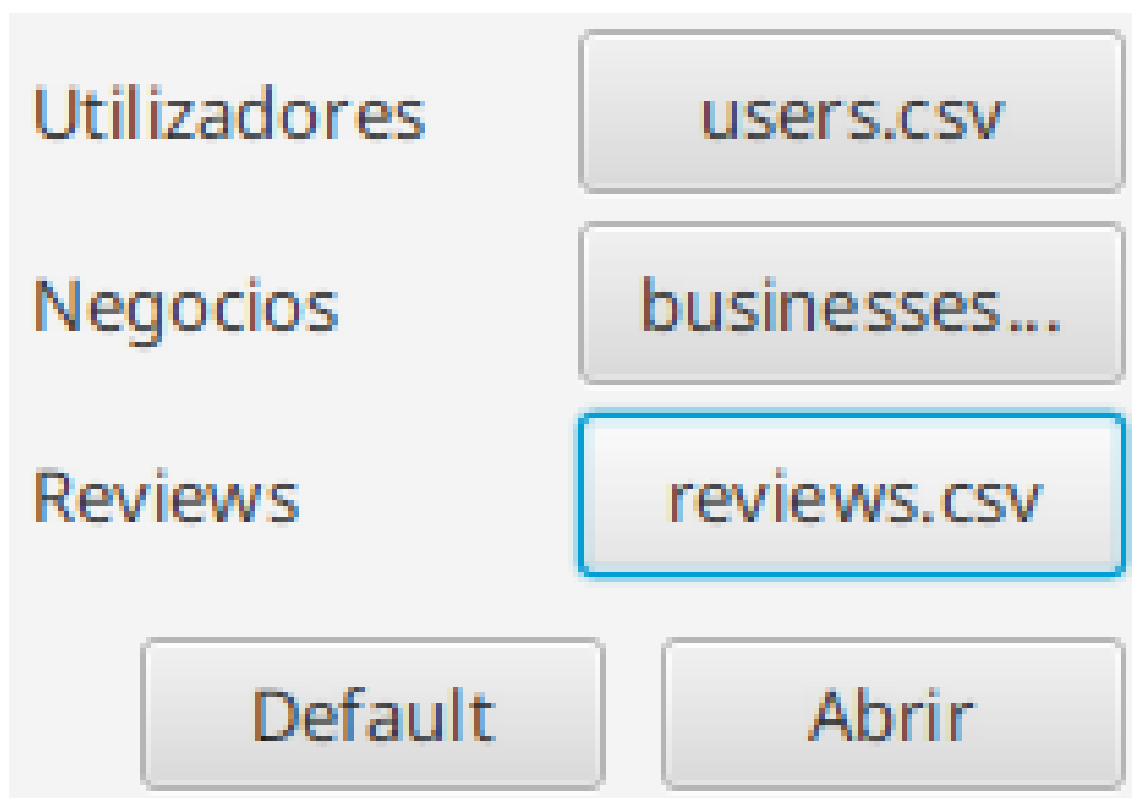


Figura C.2: Menu para abrir ficheiros CSV

<div>Criar Restaurar Gravar</div>																																																																																																																																																			
<div> <div> <div>▶ Query 1</div> <div>▶ Query 2</div> <div>▶ Query 3</div> <div>▶ Query 4</div> <div>▶ Query 5</div> <div>▶ Query 6</div> <div>▼ Query 7</div> <div> <div>Descrição</div> <div>Determinar, para cada cidade, a lista dos três mais famosos negócios em termos de número de reviews</div> <div>Executar</div> </div> </div> <div> <div>▶ Query 8</div> <div>▶ Query 9</div> <div>▶ Query 10</div> <div>▶ Estatísticas</div> </div> </div>																																																																																																																																																			
<div> <div>Home Page Resultado (3) × Resultado (4) ×</div> <div>Query Time: 1.816E-6</div> <table> <tr> <th>City</th><th>Business Id</th><th>Business Number total</th><th></th></tr> <tr><td>SURREY</td><td>aM7kKueLeGOHIM3LA05Btw</td><td>14</td><td></td></tr> <tr><td>West Vancouver</td><td>ECjtMCqbg7d0x1VSkfs80Q</td><td>76</td><td></td></tr> <tr><td>West Vancouver</td><td>S6d5KgGDyOlvk_6p03OkBw</td><td>63</td><td></td></tr> <tr><td>West Vancouver</td><td>ZqKb4X3BeqFTUIAgVH2PQ</td><td>60</td><td></td></tr> <tr><td>Vancouver</td><td>VPqWLP9kMiZEbctCebIZUA</td><td>2226</td><td></td></tr> <tr><td>Vancouver</td><td>KXCXaF5qimmtKKqnPc_LQA</td><td>830</td><td></td></tr> <tr><td>Vancouver</td><td>JgSGp5MHbGecAXs_o1rE_g</td><td>747</td><td></td></tr> <tr><td>West Point Grey</td><td>hEB1SiQSPsLE1cU0iWzZTQ</td><td>1</td><td></td></tr> <tr><td>Port Coquitlam</td><td>Yns2eROqk-f4BUL8YiuBAA</td><td>104</td><td></td></tr> <tr><td>Port Coquitlam</td><td>EGsftsHMMWmKF3mc2UQoQug</td><td>70</td><td></td></tr> <tr><td>Port Coquitlam</td><td>UmDELUApCI3O1kdJLfa_yw</td><td>58</td><td></td></tr> <tr><td>BURNABY</td><td>SwQEuhFLCx4fmft13Hyw</td><td>8</td><td></td></tr> <tr><td>Port Moody</td><td>Y8NNXzzvs4Ymu_NO0etyFw</td><td>158</td><td></td></tr> <tr><td>Port Moody</td><td>Safdgq_SqLzC0yTjWE_IpA</td><td>50</td><td></td></tr> <tr><td>Port Moody</td><td>8WQwPdNS7Gnc5bLBcXEq8A</td><td>23</td><td></td></tr> <tr><td>Richmond</td><td>93dx1pRsq51nFRm9bvJvTA</td><td>85</td><td></td></tr> <tr><td>burnaby</td><td>7x0EbQo_gHE3D0ibUnM_GA</td><td>6</td><td></td></tr> <tr><td>Coquitlam</td><td>3j_E0XfGLFAHKnQqIO9rXA</td><td>141</td><td></td></tr> <tr><td>Coquitlam</td><td>wTVzfHLOYUyd6wY9LJWC2w</td><td>99</td><td></td></tr> <tr><td>Coquitlam</td><td>xLpF-epmxiju8UYeNuyIog</td><td>70</td><td></td></tr> <tr><td>Greater Vancouver A</td><td>TaY5C0YOR7zV0wWQYAVd9w</td><td>11</td><td></td></tr> <tr><td>North Vancouver</td><td>bvNPKcLDnbhrOo7h92z2g</td><td>130</td><td></td></tr> <tr><td>North Vancouver</td><td>2GynEdoTt2S8uem1GyQF5w</td><td>90</td><td></td></tr> <tr><td>North Vancouver</td><td>tdw3YB_dIFPIA42PGMfwSQ</td><td>76</td><td></td></tr> <tr><td>Vancouver BC</td><td>FmybVSs7pB8zYO3W93hRcA</td><td>3</td><td></td></tr> <tr><td>Burnaby</td><td>Z1eRppVpwQXgzAxAc1unTA</td><td>194</td><td></td></tr> <tr><td>Burnaby</td><td>m1RAtwhe2MJU_rVidrbRXA</td><td>158</td><td></td></tr> <tr><td>Burnaby</td><td>RPnSUIZMR55T8KsSI1dwMA</td><td>155</td><td></td></tr> <tr><td>Delta</td><td>_Ief3C6t-930SBuwWuDAQ</td><td>526</td><td></td></tr> <tr><td>Delta</td><td>rDLPdfI6TOsQITf6AQaCTA</td><td>50</td><td></td></tr> <tr><td>Delta</td><td>mPjlsIrVPe_xrof3WU7HCw</td><td>39</td><td></td></tr> <tr><td>N Vancouver</td><td>_OfAq8W9cIDTta_YesbcbQ</td><td>6</td><td></td></tr> <tr><td>New Westminster</td><td>Zya97y9YEQwu3pPLDylw3g</td><td>152</td><td></td></tr> <tr><td>New Westminster</td><td>stLI_6xiiWpS75gkTosMow</td><td>104</td><td></td></tr> <tr><td>New Westminster</td><td>uuuKsf4hY2D6Ym8e7l8eFg</td><td>55</td><td></td></tr> </table> </div>				City	Business Id	Business Number total		SURREY	aM7kKueLeGOHIM3LA05Btw	14		West Vancouver	ECjtMCqbg7d0x1VSkfs80Q	76		West Vancouver	S6d5KgGDyOlvk_6p03OkBw	63		West Vancouver	ZqKb4X3BeqFTUIAgVH2PQ	60		Vancouver	VPqWLP9kMiZEbctCebIZUA	2226		Vancouver	KXCXaF5qimmtKKqnPc_LQA	830		Vancouver	JgSGp5MHbGecAXs_o1rE_g	747		West Point Grey	hEB1SiQSPsLE1cU0iWzZTQ	1		Port Coquitlam	Yns2eROqk-f4BUL8YiuBAA	104		Port Coquitlam	EGsftsHMMWmKF3mc2UQoQug	70		Port Coquitlam	UmDELUApCI3O1kdJLfa_yw	58		BURNABY	SwQEuhFLCx4fmft13Hyw	8		Port Moody	Y8NNXzzvs4Ymu_NO0etyFw	158		Port Moody	Safdgq_SqLzC0yTjWE_IpA	50		Port Moody	8WQwPdNS7Gnc5bLBcXEq8A	23		Richmond	93dx1pRsq51nFRm9bvJvTA	85		burnaby	7x0EbQo_gHE3D0ibUnM_GA	6		Coquitlam	3j_E0XfGLFAHKnQqIO9rXA	141		Coquitlam	wTVzfHLOYUyd6wY9LJWC2w	99		Coquitlam	xLpF-epmxiju8UYeNuyIog	70		Greater Vancouver A	TaY5C0YOR7zV0wWQYAVd9w	11		North Vancouver	bvNPKcLDnbhrOo7h92z2g	130		North Vancouver	2GynEdoTt2S8uem1GyQF5w	90		North Vancouver	tdw3YB_dIFPIA42PGMfwSQ	76		Vancouver BC	FmybVSs7pB8zYO3W93hRcA	3		Burnaby	Z1eRppVpwQXgzAxAc1unTA	194		Burnaby	m1RAtwhe2MJU_rVidrbRXA	158		Burnaby	RPnSUIZMR55T8KsSI1dwMA	155		Delta	_Ief3C6t-930SBuwWuDAQ	526		Delta	rDLPdfI6TOsQITf6AQaCTA	50		Delta	mPjlsIrVPe_xrof3WU7HCw	39		N Vancouver	_OfAq8W9cIDTta_YesbcbQ	6		New Westminster	Zya97y9YEQwu3pPLDylw3g	152		New Westminster	stLI_6xiiWpS75gkTosMow	104		New Westminster	uuuKsf4hY2D6Ym8e7l8eFg	55	
City	Business Id	Business Number total																																																																																																																																																	
SURREY	aM7kKueLeGOHIM3LA05Btw	14																																																																																																																																																	
West Vancouver	ECjtMCqbg7d0x1VSkfs80Q	76																																																																																																																																																	
West Vancouver	S6d5KgGDyOlvk_6p03OkBw	63																																																																																																																																																	
West Vancouver	ZqKb4X3BeqFTUIAgVH2PQ	60																																																																																																																																																	
Vancouver	VPqWLP9kMiZEbctCebIZUA	2226																																																																																																																																																	
Vancouver	KXCXaF5qimmtKKqnPc_LQA	830																																																																																																																																																	
Vancouver	JgSGp5MHbGecAXs_o1rE_g	747																																																																																																																																																	
West Point Grey	hEB1SiQSPsLE1cU0iWzZTQ	1																																																																																																																																																	
Port Coquitlam	Yns2eROqk-f4BUL8YiuBAA	104																																																																																																																																																	
Port Coquitlam	EGsftsHMMWmKF3mc2UQoQug	70																																																																																																																																																	
Port Coquitlam	UmDELUApCI3O1kdJLfa_yw	58																																																																																																																																																	
BURNABY	SwQEuhFLCx4fmft13Hyw	8																																																																																																																																																	
Port Moody	Y8NNXzzvs4Ymu_NO0etyFw	158																																																																																																																																																	
Port Moody	Safdgq_SqLzC0yTjWE_IpA	50																																																																																																																																																	
Port Moody	8WQwPdNS7Gnc5bLBcXEq8A	23																																																																																																																																																	
Richmond	93dx1pRsq51nFRm9bvJvTA	85																																																																																																																																																	
burnaby	7x0EbQo_gHE3D0ibUnM_GA	6																																																																																																																																																	
Coquitlam	3j_E0XfGLFAHKnQqIO9rXA	141																																																																																																																																																	
Coquitlam	wTVzfHLOYUyd6wY9LJWC2w	99																																																																																																																																																	
Coquitlam	xLpF-epmxiju8UYeNuyIog	70																																																																																																																																																	
Greater Vancouver A	TaY5C0YOR7zV0wWQYAVd9w	11																																																																																																																																																	
North Vancouver	bvNPKcLDnbhrOo7h92z2g	130																																																																																																																																																	
North Vancouver	2GynEdoTt2S8uem1GyQF5w	90																																																																																																																																																	
North Vancouver	tdw3YB_dIFPIA42PGMfwSQ	76																																																																																																																																																	
Vancouver BC	FmybVSs7pB8zYO3W93hRcA	3																																																																																																																																																	
Burnaby	Z1eRppVpwQXgzAxAc1unTA	194																																																																																																																																																	
Burnaby	m1RAtwhe2MJU_rVidrbRXA	158																																																																																																																																																	
Burnaby	RPnSUIZMR55T8KsSI1dwMA	155																																																																																																																																																	
Delta	_Ief3C6t-930SBuwWuDAQ	526																																																																																																																																																	
Delta	rDLPdfI6TOsQITf6AQaCTA	50																																																																																																																																																	
Delta	mPjlsIrVPe_xrof3WU7HCw	39																																																																																																																																																	
N Vancouver	_OfAq8W9cIDTta_YesbcbQ	6																																																																																																																																																	
New Westminster	Zya97y9YEQwu3pPLDylw3g	152																																																																																																																																																	
New Westminster	stLI_6xiiWpS75gkTosMow	104																																																																																																																																																	
New Westminster	uuuKsf4hY2D6Ym8e7l8eFg	55																																																																																																																																																	

Figura C.3: Resultado de uma query