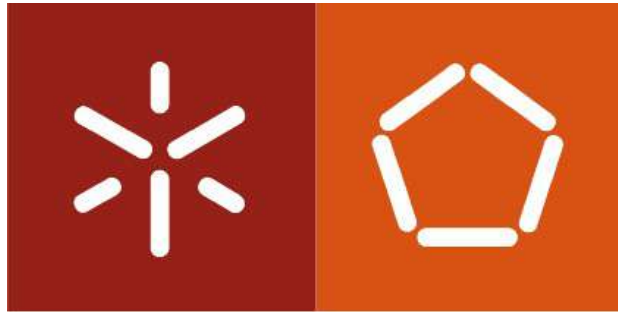


MESTRADO INTEGRADO EM ENGENHARIA
INFORMÁTICA

SISTEMAS BASEADOS EM SIMILARIDADE
GRUPO 3



Universidade do Minho

Relatório Trabalho Prático 3
Sistema de Recomendação EatNow



Bruno Nascimento, a67647
João Palmeira, a73864
Rafael Silva, a74264

5 de Janeiro de 2020

Conteúdo

1	Introdução	3
2	Descrição Dataset	4
3	Abordagens Utilizadas	5
3.1	Top-N	5
3.2	Filtragem colaborativa	5
3.3	Filtragem por Conteúdo	5
4	Descrição dos Workflows	6
4.1	Leitura de Dados	7
4.2	Manipulação de Dados	7
4.2.1	Fase inicial Manipulação	8
4.2.2	Fase Continuação 1 Manipulação	9
4.3	Método PCA	10
4.4	Extra Manipulação	11
4.5	<i>Elbow Method</i>	12
4.5.1	MSE	14
4.5.2	MSE Fórmula	14
4.5.3	<i>Elbow</i>	14
4.6	<i>Cluster</i> por Cidade	15
5	Base de Dados Desenvolvida	17
6	Sistema Recomendação Desenvolvido	21
6.1	<i>Home Page</i>	21
6.2	<i>Login/Registar</i>	24
6.3	<i>Explorer</i>	25
6.4	Restaurante Individual	26
6.5	<i>Review</i>	28
6.6	<i>News</i>	29
7	Conclusão	30

Lista de Figuras

1	Estrutura do <i>Workflow</i>	6
2	SubFlow Correspondente á Base de Dados	6
3	SubFlow Correspondente á Base de Dados	7
4	Nodo CSV Reader.	7
5	Metanodo Manipulação de dados.	7
6	Manipulação de dados.	8
7	Manipulação de dados Fase Inicial	8
8	Manipulação de dados Continuação 1	9
9	Manipulação de dados Continuação 2	9
10	Metanodo método PCA	10
11	Método PCA	10
12	Gráfico método PCA	11
13	Metanodo Extra Manipulação	11
14	Extra Manipulação	11
15	Metanodo <i>Elbow Method</i>	12
16	<i>Elbow Method</i>	12
17	Parte interessante do Método do Cotovelo	12
18	Representação Gráfica Elbow Method	13
19	Representação Tabular Elbow Method	13
20	Metanodo MSE	14
21	MSE	14
22	Metanodo <i>Elbow</i>	15
23	<i>Nodos pertencentes ao metanodo Elbow</i>	15
24	Metanodo <i>Cluster</i> por Cidade	15
25	<i>Cluster</i> por Cidade	16
26	Criar BD de estilos de cozinha	17
27	<i>Workflow</i> para diferentes tipos de filtragem	17
28	Leitura das coleções	18
29	Metanodo Manipulação	18
30	Metanodo Extra Manipulação	19
31	Metanodo <i>Cluster por User</i>	19
32	Metanodo Filtragem Colaborativa	19
33	Metanodo Filtragem por Conteúdo	20
34	Metanodo Filtragem por Estilo de Cozinha	20
35	Metanodo Escrita Mongo	20
36	<i>Home Page</i>	21
37	<i>Home Page</i> com filtragem	22
38	Top sugestões total	22
39	Top sugestões mais visualizados	23
40	Informação da quantidade de restaurantes	23
41	Top sugestões pós registo	24
42	Top sugestões baseadas em <i>reviews</i>	24
43	Registo na Plataforma	25
44	Login na Plataforma	25
45	<i>Explorer</i>	26
46	Lista de restaurantes	26
47	Restaurante individual	27
48	Descrição do restaurante	27
49	<i>Reviews</i> do restaurante	28
50	<i>Review</i>	28
51	Campos da <i>review</i>	29
52	<i>News</i>	29

1 Introdução

Este trabalho prático tem como objetivo fundamental a elaboração de um sistema de recomendação. Um sistema de recomendação combina vários paradigmas de *Machine Learning* para criar uma ferramenta que irá prever a preferência de um utilizador num determinado contexto, num determinado tema, propondo-lhe recomendações sobre o mesmo.

Decidiu-se elaborar um sistema de recomendação no âmbito da recomendação de restaurantes, visto que foi um dos temas abordados no primeiro trabalho prático e o que nos despertou maior interesse.

Neste relatório está descrita toda a informação relativa ao *dataset* escolhido, todas as abordagens utilizadas na criação do sistema, e, posteriormente, todo o desenvolvimento dos *workflows* realizados no KNIME para o tratamento de dados. Por último, está descrito todo o desenvolvimento do sistema de recomendação que decidimos denominar de ***EatNow*** bem como todas as diferentes funcionalidades disponibilizadas nesse mesmo sistema.

2 Descrição Dataset

O *dataset* escolhido foi o *TripAdvisor Restaurants Info for 31 Euro-Cities*, que contém dados de restaurantes de diferentes estilos de cozinha em 31 cidades diferentes de toda a Europa. Este é composto por mais de 125000 restaurantes diferentes e contém os seguintes atributos:

- ***Name***: nome do restaurante;
- ***City***: localização da cidade do restaurante;
- ***Cuisine Style***: estilos de cozinha do restaurante;
- ***Ranking***: classificação do restaurante face ao número total de restaurantes da cidade;
- ***Rating***: taxa do restaurante em uma escala de 1 a 5;
- ***Price Range***: faixa de preço do restaurante dividido em 3 categorias;
- ***Number of Reviews***: número de avaliações que os clientes deixaram no restaurante;
- ***Reviews***: *review* deixada pelo cliente no restaurante que contém o texto e a data da *review*;
- ***URL_TA***: parte do URL da página de restaurante detalhada que vem depois de "www.tripadvisor.com";
- ***ID_TA***: a identificação do restaurante na base de dados.

Como se pode verificar, este conjunto de dados não dispõe de utilizadores (clientes), algo que o grupo optou por introduzir numa fase mais avançada do trabalho, de modo a que fosse possível abranger mais paradigmas de *Machine Learning* e utilizar as suas técnicas.

3 Abordagens Utilizadas

Para a realização deste trabalho foram utilizadas diferentes abordagens no sistema de recomendação.

Numa fase inicial utilizou-se abordagens à base de Top-N, uma vez que os o *dataset* escolhido não possui dados de utilizadores e apenas possui dados dos restaurantes. Numa fase mais avançada, decidiu-se introduzir utilizadores permitindo fazer uso de abordagens como a filtragem colaborativa e filtragem baseada em conteúdo.

3.1 Top-N

A partir do *ranking* de restaurantes por parte de outros utilizadores. Este método retorna os Top-N melhores restaurantes, sendo N um número arbitrário no geral ou consoante requisitos previamente definidos.

Por exemplo, um exemplo desta recomendação será o sistema recomendar os 10 melhores restaurantes independentemente da cidade onde se encontra e os tipo de comida oferecido.

Caso sejam definidos critérios de escolha perante o restaurante, este método retornará na mesma os N melhores restaurantes mas consoante as condições previamente estabelecidas, casos como cidades diferentes poderá já não ser aceitável.

3.2 Filtragem colaborativa

Para implementar filtragem colaborativa foram implementados *clusters* por utilizadores do Sistema de Recomendação.

Os *clusters* foram definidos pela associação dos restaurantes e os clientes que os visitaram e os respectivos estilos de comida.

Sempre com base na similaridade entre utilizadores, as recomendações são assim feitas, é preciso ter atenção neste ponto que a similaridade tem de ser elevada com o intuito de fornecer recomendações credíveis e nada de desprezável

3.3 Filtragem por Conteúdo

Esta forma de recomendação tem por base o histórico do utilizador, que ao longo de entradas no sistema cria um associação de gostos do utilizador com novas sugestões do mesmo estilo, A forma como este processo é executado é através de regras de associação, que fornece a consequência para um dado utilizador.

Ao fornecer novos restaurantes nunca antes visitados pelo o utilizador o Sistema de Recomendação está a sugerir novos produtos de acordo com as preferências do utilizador.

De forma a possuir os estilos de cozinha e cidades separados por *clusters* permite uma melhor e maior eficiência e credibilidade, consoante o *ranking* associado a cada local.

4 Descrição dos Workflows

Após a descrição do *dataset*, para proceder à implementação de um Sistema de Recomendação (S.R), é necessário tratar e processar os dados envolvidos na implementação.

Para tal, foi usada a ferramenta KNIME, além de ser requisito na parte de tratamento de dados é também a ferramenta mais familiarizada pelos os elementos do grupo para essa função.

Além do tratamento de dados, também usamos o KNIME para ligação ao Sistema de recomendação, portanto o workflow está composto por uma parte principal (Tratamento de Dados) e partes extras.

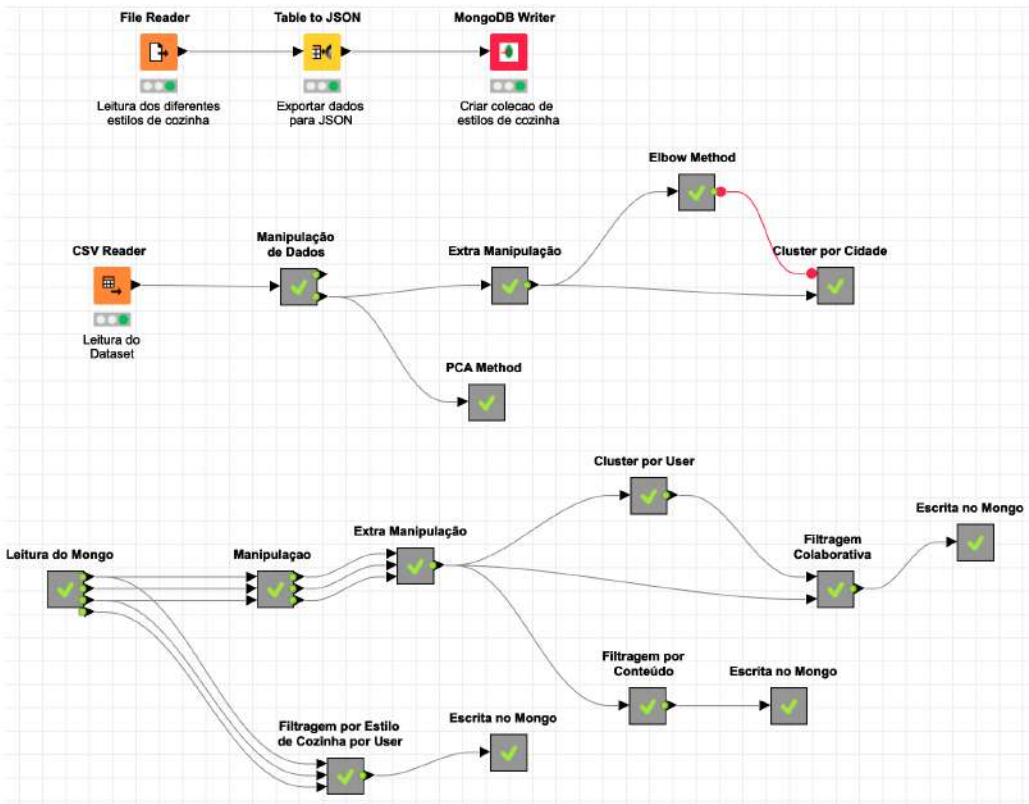


Figura 1: Estrutura do *Workflow*

Quando aberto o KNIME, a figura acima representada é o Workflow completo, no entanto as funcionalidades dos *subworkflows* são diferentes.

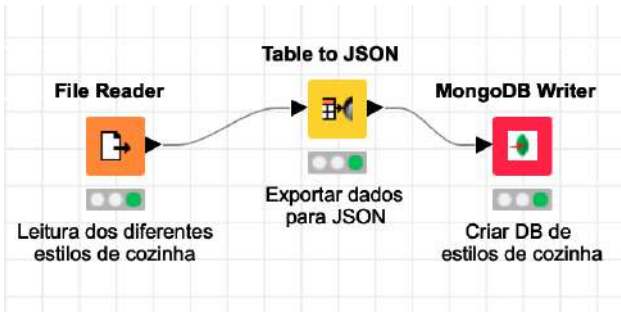


Figura 2: SubFlow Correspondente á Base de Dados

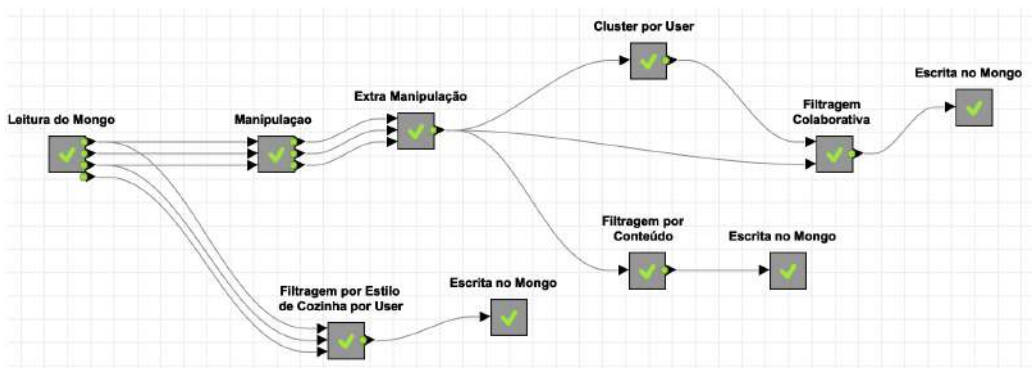


Figura 3: SubFlow Correspondente á Base de Dados

Estes *SubWorkflows* representados acima serão descritos no capítulo seguinte **Base de Dados Desenvolvida**, como neste capítulo apenas nos interessa o tratamento de dados, vamos falar do *SubWorkflow* correspondente ao tratamento de dados do dataset inicial.

4.1 Leitura de Dados

Como visto anteriormente, em trabalhos práticos passados e fichas realizadas nas aulas, procedemos á leitura do *dataset* já mencionado em **Descrição Dataset**. Com o nodo *CSV Reader* importamos o ficheiro para o KNIME para proceder ao tratamento de dados.



Figura 4: Nodo CSV Reader.

4.2 Manipulação de Dados

Após a importação e leitura do *dataset* em formato CSV, começa agora todo o processo de manipular os dados contidos nele para melhor eficiência ao reduzir redundância e dados desprezáveis para o sistema de recomendação a ser elaborado.

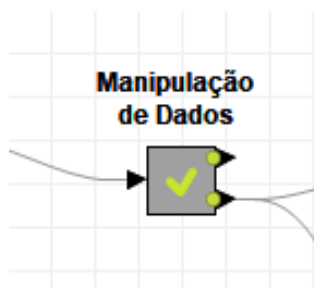


Figura 5: Metanodo Manipulação de dados.

Uma vez que foi recomendado o uso de metanodo para melhorar interpretação e organização do *workflow* final, certas fases, como *Manipulação de Dados* possuem bastantes nodos intervenientes no tratamento de dados e ao apresentar simplesmente o metanodo, cabe ao visitante/interessado, abrir o metanodo para observar em que consiste esse passo.

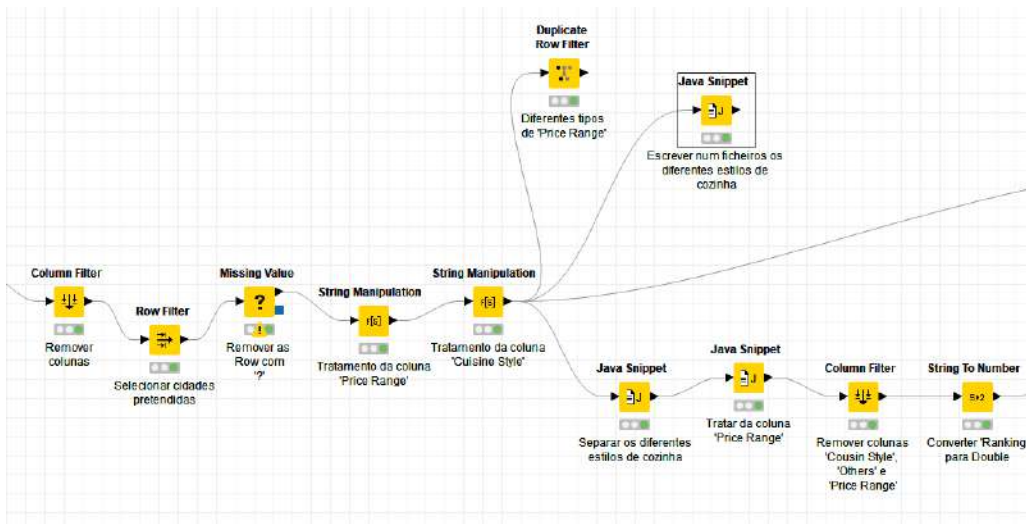


Figura 6: Manipulação de dados.

Para uma melhor explicação e compreensão dos nodos aqui implementados, vamos explicar esta fase em 3 partes.

4.2.1 Fase inicial Manipulação

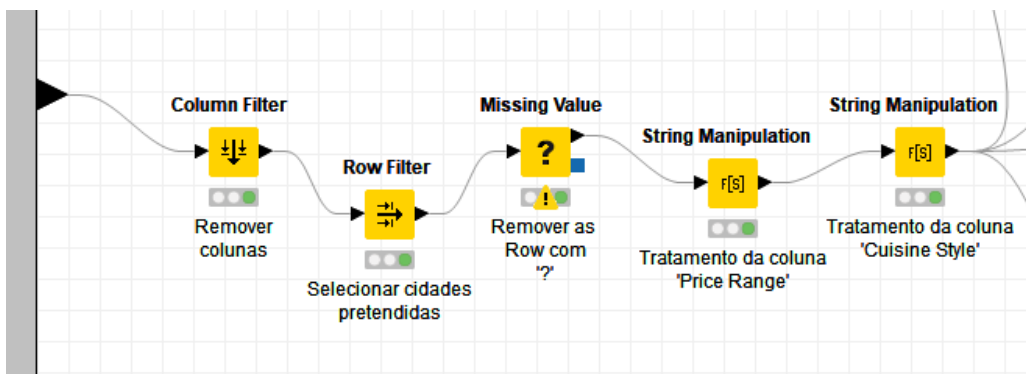


Figura 7: Manipulação de dados Fase Inicial

- **Column Filter:** Neste nodo, retiramos as colunas que não contêm informação importante, por exemplo, colunas como `_TA`, que indica o *url* do restaurante no *website Trip Advisor*, de nada nos interessa no nosso projeto e, como este exemplo, existem outras colunas que possuíam informação redundante ou desprezável;
- **Row Filter:** A decisão para este passo, provém depois do conselho da equipa docente, como *dataset* inicial é bastante longo, devido ao conteúdo abrangente de cidades nele incluído, decidimos restringir a oferta, para isso reduziu-se o tamanho de 31 cidades para apenas 4 cidades : Porto, Lisboa, Madrid e Barcelona;
- **Missing Value:** Várias linhas do *dataset* contêm o símbolo `?`, e como não é possível contabilizar os dados que apenas possuem `?`, foi decidido retirar todas as linhas onde exista ocorrência, quer numa célula ou em todas as células;
- **String Manipulation, Price Range:** Na coluna *Price Range*, é procedida a eliminação de valores numéricos, por exemplo, até este ponto os possíveis de valores na coluna *Price Range* eram os seguintes, \$, \$\$-\$\$\$\$, \$\$\$\$\$, 2.5, 3.5 4.0, ao efectuar esta manipulação, o resultado que se obteve apenas regista estes valores possíveis \$, \$\$-\$\$\$\$, \$\$\$\$\$;
- **String Manipulation, Cuisine Style:** Nesta coluna, procedeu-se ao tratamento das componentes de separação de estilos de cozinha, caracteres tais como `'[,]`, foram

eliminados, ficando agora apenas o carácter | a delimitar os vários estilos de cozinha numa célula;

4.2.2 Fase Continuação 1 Manipulação

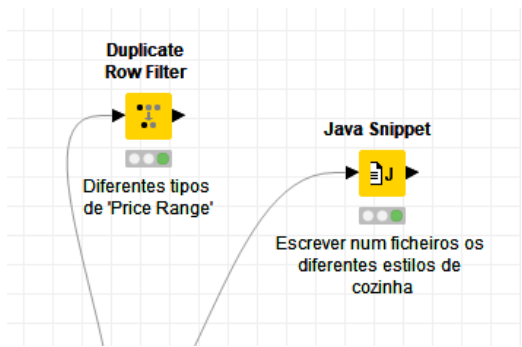


Figura 8: Manipulação de dados Continuação 1

- **Duplicate Row Filter:** Esta funcionalidade serve apenas para demonstrar as ocorrências na coluna Price Range, com um exemplo de cada caso possível, \$, \$\$-\$\$\$,\$\$\$\$;
- **Java Snippet:** Uma vez implementada a separação de estilos de cozinha através do caractere |, para proceder a uma melhor interpretação para o desenvolvimento do sistema de Recomendação, aplicou-se este nodo para separar todos os estilos de cozinha e guardá-los num ficheiro, com uma só ocorrência de cada;

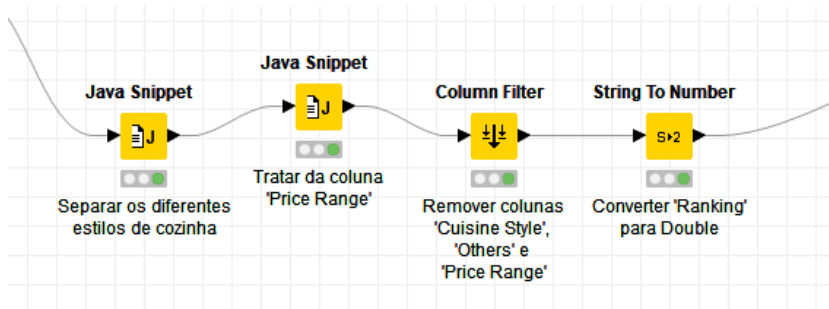


Figura 9: Manipulação de dados Continuação 2

- **Java Snippet:** Esta manipulação é complexa, porque a cada estilo de cozinha descoberto anteriormente será adicionada uma coluna ao *dataset* com esse nome, e consoante o restaurante possua ou não esse estilo uma *flag* 0 ou 1. A desvantagem deste nodo é a expansão significativa da tabela, mas ao mesmo tempo torna-se mais fácil tratar consoante um estilo de cozinha quais os restaurantes que estão incorporados neles;
- **Java Snippet:** De certo modo como no nodo anterior, uma coluna é desdobrada em várias, no entanto em menor quantidade que anteriormente. neste passo procede-se a adição de novas colunas referentes ao *Price Range* de cada restaurante, foram criadas as colunas: *Cheap*, *Medium*, *Expensive* e *Very Expensive*, mais uma vez com *flags* associadas ao respectivos restaurantes;
- **Column Filter:** Após a adição das colunas anteriores, neste nodo elimina-se as colunas que foram analisadas e posteriormente separadas *Cuisine Style*, *Price Range*;
- **String To Number:** Na coluna *ranking* alterou-se o tipo da coluna *Ranking* do tipo *String* para *Double*;

4.3 Método PCA

O PCA é caracterizado por identificar as dimensões ao longo das quais os dados se encontram mais dispersos. Desta forma, conseguimos identificar as dimensões que melhor diferenciam o conjunto de dados em análise, ou seja, os seus componentes principais.

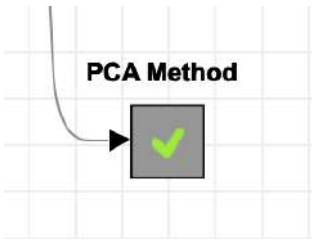


Figura 10: Metanodo método PCA

Assim sendo definiu-se o metanodo *PCA Method* onde se introduziu um PCA onde incluímos todas as estílo de cozinha e reduzimos a duas dimensões. Posteriormente adicionamos um *Scatter Plot* de modo a poder visualizar num gráfico a disposição dos dados de maneira a retirar conclusões relativamente à quantidade de *clusters* que poderíamos formar.

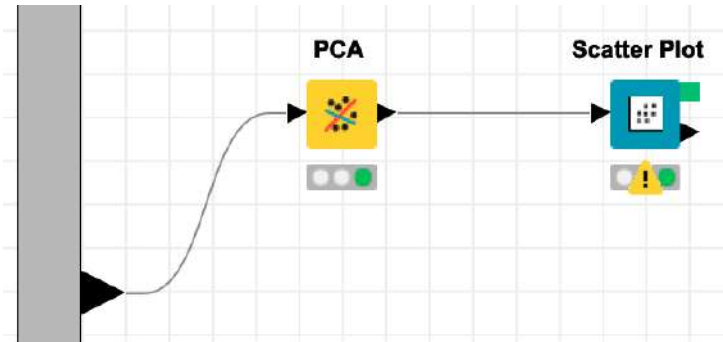


Figura 11: Método PCA

Como se pode visualizar pelo gráfico elaborado, é possível definir 4 conjuntos de dados distintos e, a partir desta análise, decidiu-se definir 4 *clusters* onde cada um irá dizer respeito a cada cidade.

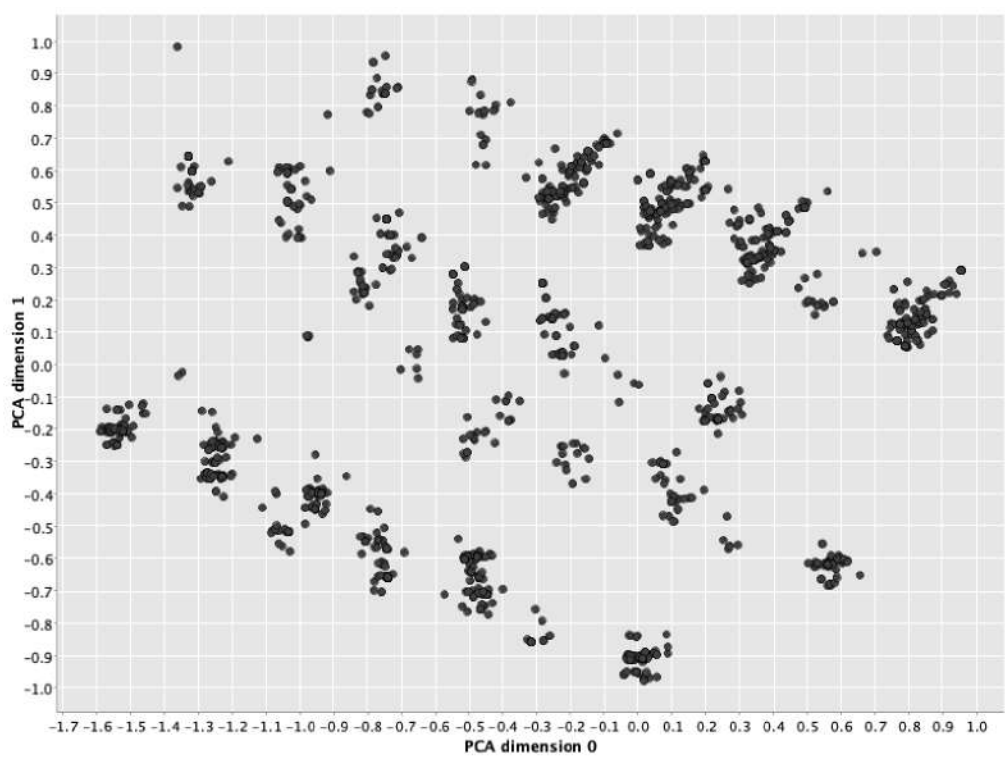


Figura 12: Gráfico método PCA

4.4 Extra Manipulação

Voltando à manipulação, a criação deste metanodo diz respeito ao tratamento do *Rating*.

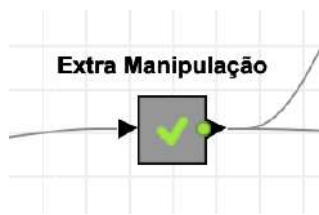


Figura 13: Metanodo Extra Manipulação

Optou-se por realizar um *GroupBy* de forma a selecionar a variável *Rating* para que fosse possível criar a média desta variável para cada restaurante. De seguida, aplicou-se um *Normalizer* para normalizar estes novos valores obtidos (a normalização foi realizar utilizando valores mínimos e máximos de 0 e 1, respetivamente).

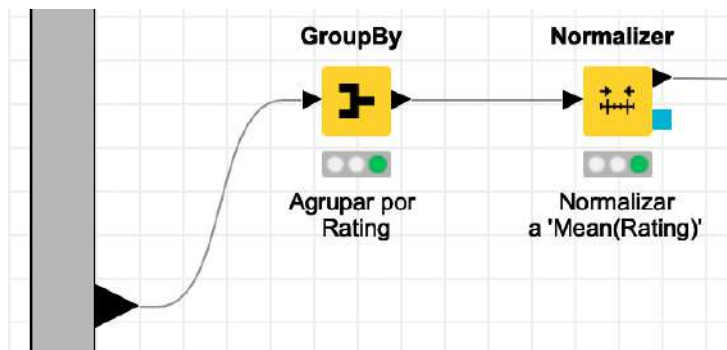


Figura 14: Extra Manipulação

4.5 Elbow Method

Após a visualização do PCA em que se pode ter ideia do número de *clusters*, no *Método do Cotovelo*, podemos calcular esse valor de forma iterativa e matemática através de vários nodos.

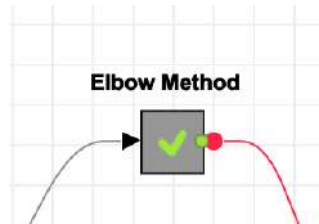


Figura 15: Metanodo *Elbow Method*

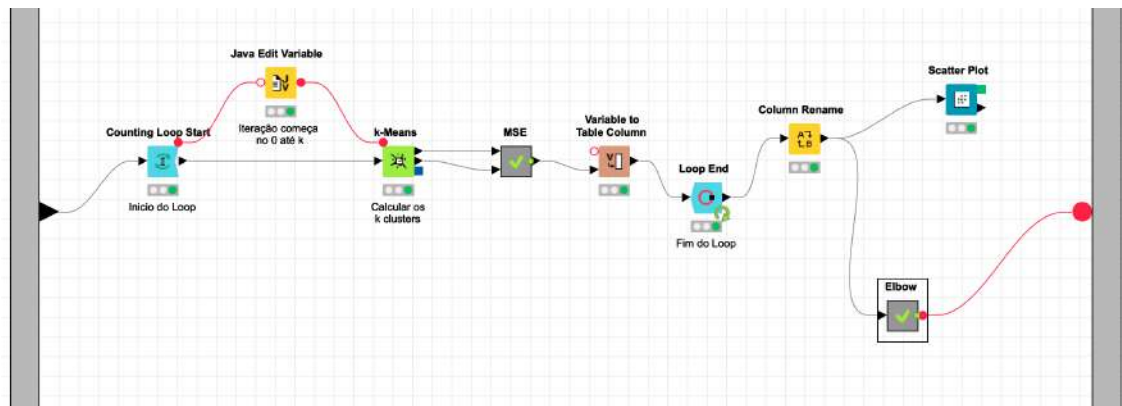


Figura 16: *Elbow Method*

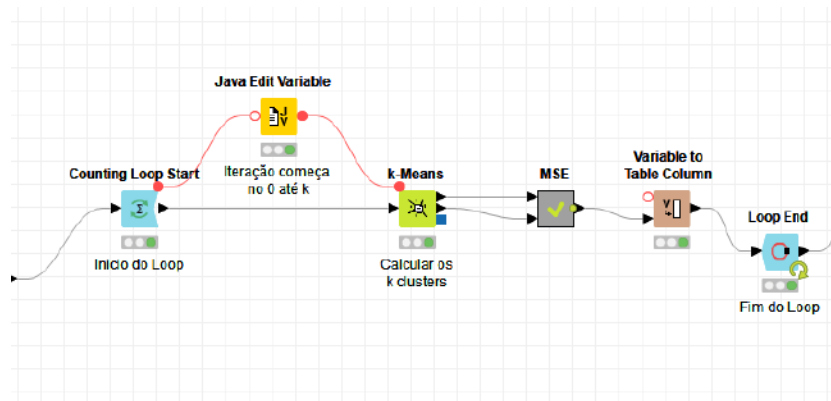


Figura 17: Parte interessante do Método do Cotovelo

- **Counting Loop Start:** Variável Current Iteration é disponibilizada por este nodo;
- **Java Edit Variable:** Pela variável disponibilizada anteriormente, é definida o número de iterações com a variável a começar com o valor 0;
- **K-Means:** Algoritmo que retorna número de clusters associados ao dataset fornecido até este ponto;

- **Variable to Table Column:** Inclusão da lista que contém todas variáveis que são convertidas em novas colunas na tabela final, excluindo variáveis que não são usadas;
- **Loop End:** Colecciona e combina todos os modelos previstos durante total de iterações numa tabela com respectivas colunas;
- **Column Rename:**Coluna *Mean (Error Sum)*, passa a ter o nome *MSE*;
- **Scatter Plot:** Representação Gráfica dos pontos, com a variável K quantificada no *x axis* e a variável y quantificada em *MSE*;

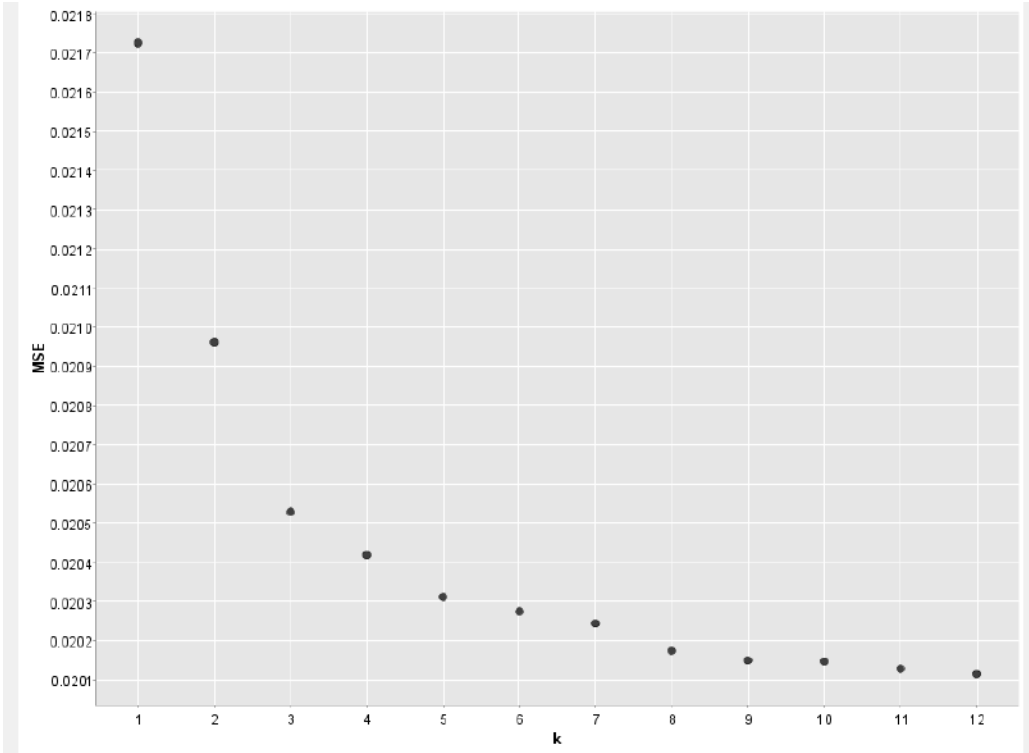


Figura 18: Representação Gráfica Elbow Method

Row ID	MSE	k	Iteration	Selecte...
Row0#0	0.022	1	0	false
Row0#1	0.021	2	1	false
Row0#2	0.021	3	2	false
Row0#3	0.02	4	3	false
Row0#4	0.02	5	4	false
Row0#5	0.02	6	5	false
Row0#6	0.02	7	6	false
Row0#7	0.02	8	7	false
Row0#8	0.02	9	8	false
Row0#9	0.02	10	9	false
Row0#10	0.02	11	10	false
Row0#11	0.02	12	11	false

Figura 19: Representação Tabular Elbow Method

4.5.1 MSE

Neste processo é realizado o cálculo do erro quantificado a K, que consiste na média de diferenças entre o valor previsto e o actual observado, ao quadrado, sem ter em consideração a direcção.

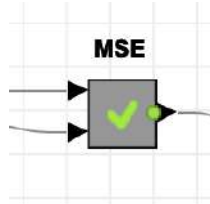


Figura 20: Metanodo MSE

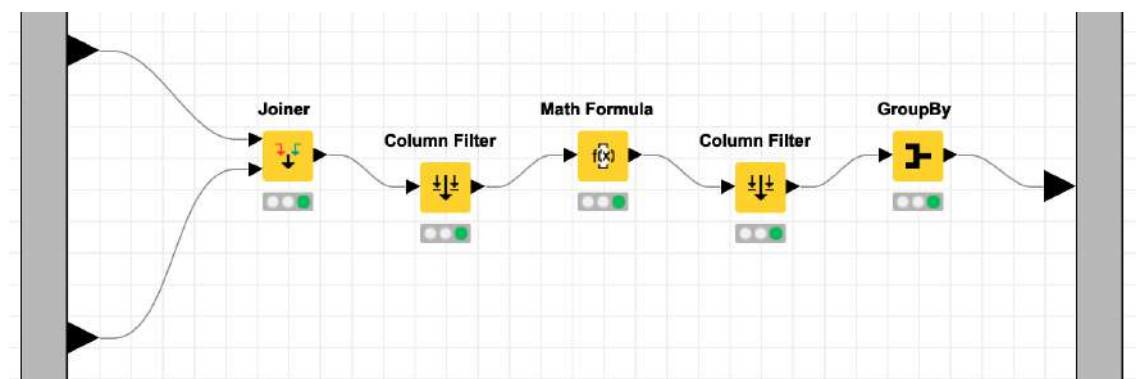


Figura 21: MSE

- **Joiner**: Recebe os outputs do nodo K-Means, nomeadamente *Labeled Input* e *Clusters*, de forma a ter ambas informações num só elemento com associação do centróide e elemento;
- **Column Filter**: As colunas *PCA dimension 0*, *PCA dimension 1*, *class* e *Cluster* são retiradas;
- **Math-Formula**: Neste nodo é efetuado o cálculo do erro ao quadrado;
- **Column Filter**: Selecionar as colunas que contem o erro por entrada;
- **GroupBy**: Resultado MSE;

4.5.2 MSE Fórmula

O cálculo de MSE, é com base na seguinte fórmula $\frac{1}{n} \sum_{t=1}^n (e_t - \hat{e}_t)^2$

- **n**: Número total de observações, neste caso estilos de cozinha;
- e_t : Valor observado;
- \hat{e}_t : Valor previsto;

4.5.3 Elbow

Após a detecção do numero de *clusters*, e das componentes que pertencem a cada um, p.e, centróides e pontos, neste metanodo é calculado a distância de cada ponto ao centróide do cluster correspondente.

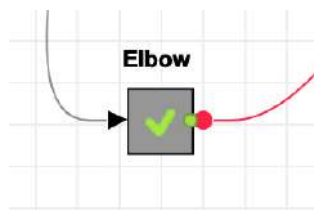


Figura 22: Metanodo *Elbow*

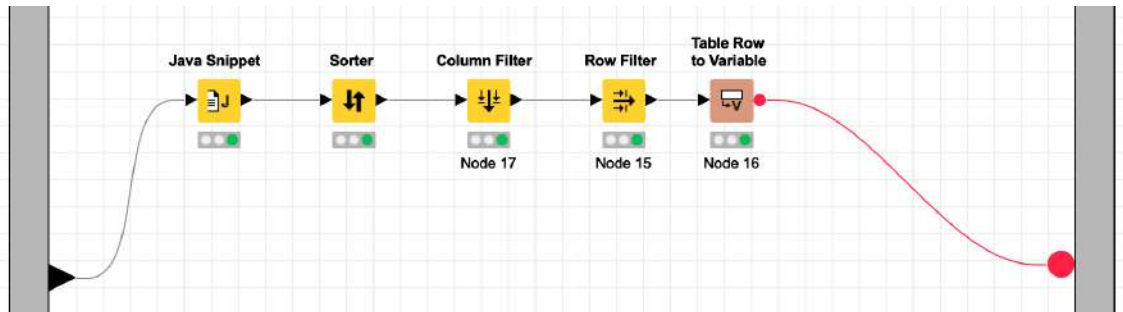


Figura 23: Nós pertencentes ao metanodo *Elbow*

- **Java Snippet:** Neste nodo é calculada a diferença entre os valores de MSE, entre os *clusters*, e guardado numa coluna nova com o nome *MseDiff*;
- **Sorter:** Ordenar de forma descendente com os valores de *MseDiff*, de forma a organizar os *cluster* com maior diferença entre eles ;
- **Column Filter:** Eliminação da coluna *Iteration* para menos informação redundante;
- **Row Filter:** Selecionar a linha que possui o valor *MseDiff* mais elevado e eliminar todas as restantes;
- **Table Row To Variable:** Este nodo usa apenas a primeira linha da tabela para definir uma nova variável *flow* e respectivos valores;

4.6 Cluster por Cidade

Como explicado no início do relatório, o *dataset* debruça-se em restaurantes e respetivos *rankings* em 31 cidades europeias que foram reduzidas para apenas 4 cidades europeias.

Como tal foi decidido separar os restaurantes por cidade ao inserir na Base de Dados.



Figura 24: Metanodo *Cluster por Cidade*

- **K-means:** Algoritmo que retorna número de *clusters* associados ao input, como neste caso possuímos 4 cidades e pretendemos criar *clusters* associados a cidades o resultado deste nodo foi de 4 *clusters*;
- **Normalizer:** A todas as linhas o valor médio de *rating* foi normalizado;

- **Column Filter:** A coluna *Cluster* foi eliminada;
- **Java Snippet:** De modo a introduzir os restaurantes na *Base de Dados* de acordo com o estilo de cozinha, por cada linha é registado num *array* os estilos para serem associados ao restaurante posteriormente na BD;
- **Java Snippet:** Do mesmo modo como o *Java Snippet* para estilos de cozinha, para inserir o intervalo de preços na *Base de Dados* por cada linha é registado num *array* os preços para serem associados ao restaurante posteriormente na BD;
- **Column Filter:** Remoção das colunas desnecessárias, nomeadamente as colunas estilos de cozinha ;
- **Column Rename:** Coluna Mean(Rating), passa a ter o nome Rating ;
- **Duplicate Row Filter:** Esta funcionalidade serve apenas para demonstrar as ocorrências na coluna *Name* e *City*, com um exemplo de cada caso possível;
- **Table to JSON:** Transformação da tabela que contem os dados para formato JSON ;
- **MongoDB Writer:** Cria a colecção de Restaurantes em MongoDB

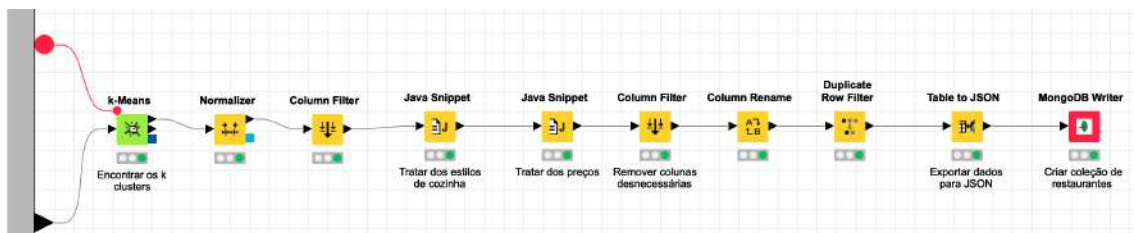


Figura 25: *Cluster* por Cidade

5 Base de Dados Desenvolvida

Neste capítulo, iremos abordar a introdução das coleções na base de dados desenvolvida no *MongoDB*.

Como é possível verificar pelo que foi mencionado no tópico anterior, é criada uma coleção de restaurantes por cidade que é introduzido na base de dados denominada *Eat-Now* através de um *MongoDB Writer*.

Através de um ficheiro com todos os estilos de cozinha, gerado pelo Tratamento de Dados realizado, é criada uma tabela em JSON através da leitura desse ficheiro. Após a criação dessa tabela, é introduzido essa tabela (coleção) na base de dados, mais uma vez, utilizando um *MongoDB Writer*.

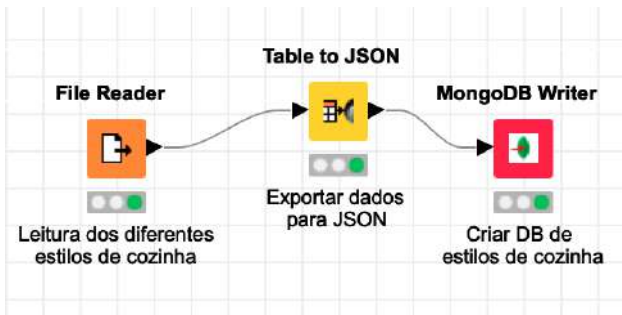


Figura 26: Criar BD de estilos de cozinha

Criadas as coleções dos estilos de cozinha e dos restaurantes por cidade, foi necessário introduzir os utilizadores na base de dados bem como as suas *reviews*, relembrando que estas coleções foram criadas fora do KNIME através de *jQuery*'s.

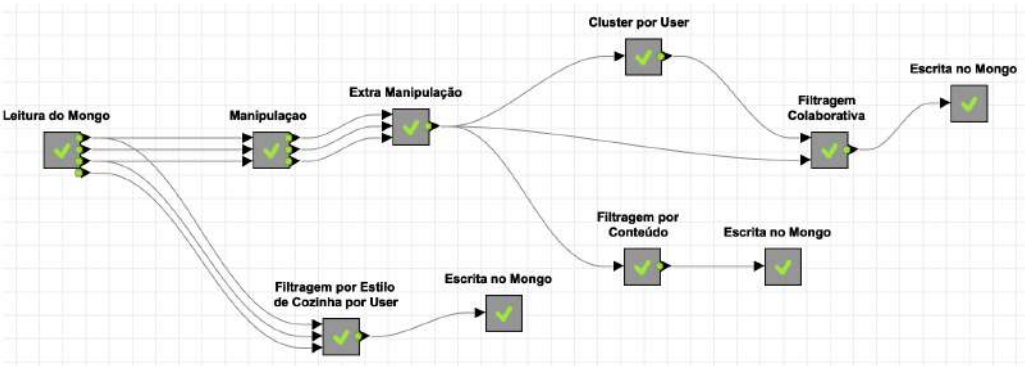


Figura 27: *Workflow* para diferentes tipos de filtragem

Inicialmente foi necessário ler da base de dados todas as coleções, de maneira a que fosse possível importá-las para tabelas de JSON.

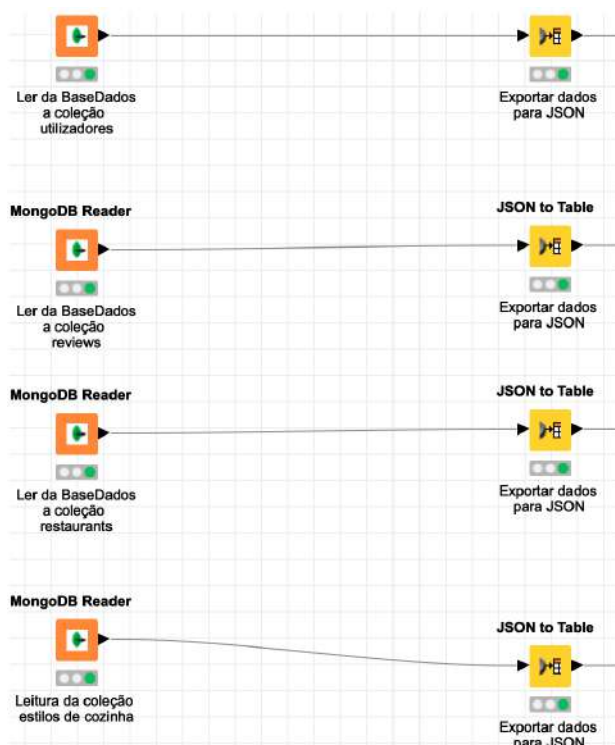


Figura 28: Leitura das coleções

Efetuada a importação dos dados, tanto para a Filtragem Colaborativa como para a Filtragem por Conteúdo irá ser necessário realizar uma manipulação dos dados. Como tal, iniciamos pelo metanodo Manipulação que recebe 3 coleções: Estilos de cozinha, utilizadores e *reviews*. De um modo geral, é realizada uma seleção da informação mais importante de cada coleção para realizar os dois tipos de filtragem como se comprova pela utilização de nodos como *Column Filter*, *Column Rename*, e ainda o *Java Snippet* para o tratamento de algumas colunas específicas.

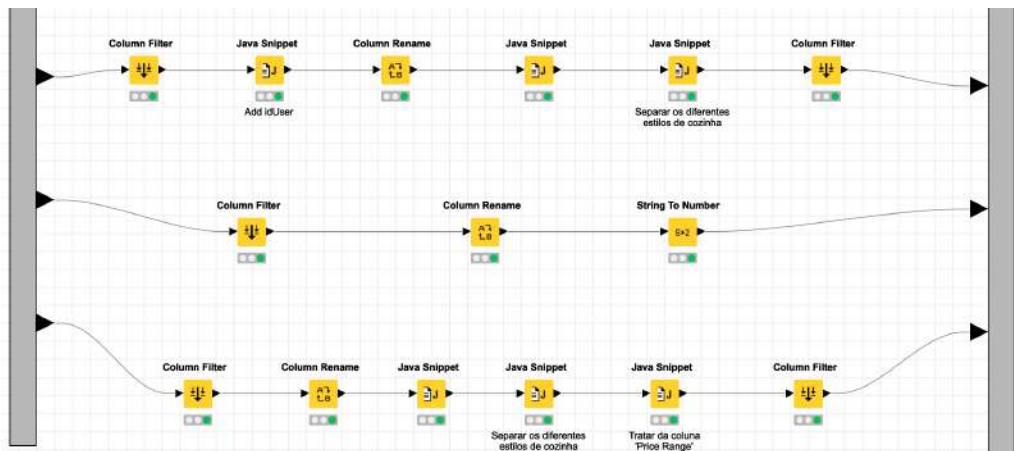


Figura 29: Metanodo Manipulação

Após a manipulação inicial, são realizadas duas associações importantes no metanodo Extra Manipulação. Numa primeira fase, associamos as coleções dos utilizadores e das *reviews* para que cada utilizador tenha associado a si todas as suas *reviews*. Numa segunda fase, é associado as coleções anteriores e os restaurantes que estão associados aos utilizadores e as *reviews* dadas.

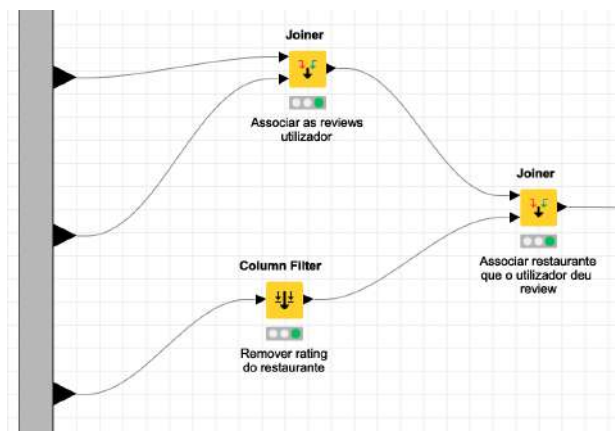


Figura 30: Metanodo Extra Manipulação

A partir daqui, vai ser possível realizar as duas filtrações mencionadas anteriormente. Começando pela **Filtragem Colaborativa**, o metanodo Filtragem Colaborativa vai receber 2 tipos de informação, um que vem diretamente do metanodo Extra Manipulação e outra que vem do metanodo *Cluster por User*. Este último é utilizado para recolher informação sobre a quantidade de *clusters* a utilizar para os *users* e, à semelhança do que foi descrito no capítulo anterior relativo ao *Cluster por Cidade*, este metanodo contém o *Elbow Method* e o MSE para o cálculo do erro, auxiliando o cálculo do número de *clusters*.

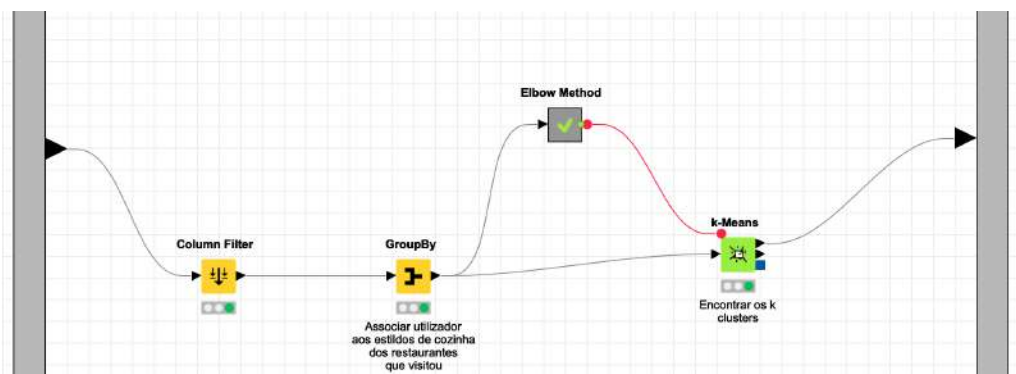


Figura 31: Metanodo *Cluster por User*

Já no metanodo Filtragem Colaborativa é realizada uma procura por semelhança onde são associados utilizadores com gostos semelhantes de modo a que seja possível recolher informação relativa aos restaurantes que os utilizadores gostam para recomendar a um específico utilizador.

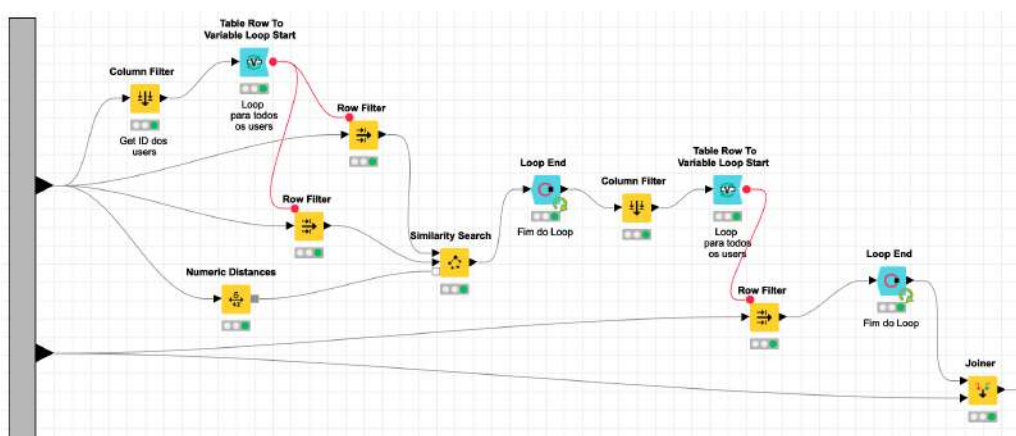


Figura 32: Metanodo Filtragem Colaborativa

Considerando agora a **Filtragem por Conteúdo**, os dados utilizados no metanodo Filtragem por Conteúdo são os dados provenientes do metanodo Extra Manipulação. A partir destes dados realiza-se um *loop* de modo a percorrer todos os utilizadores e a verificar se faz *match* com os restaurantes obtidos pelo *Association Rule Learner*, mais concretamente é verificada a quantidade de *reviews* e a associação é feita consoante a quantidade de *reviews* e gostos do utilizador em questão. Em último, faz-se uma junção dos dados iniciais aos novos dados gerados.

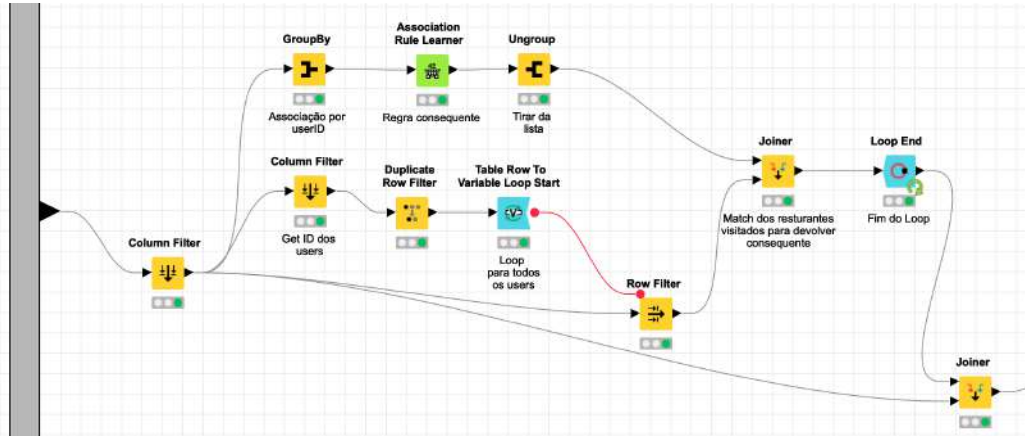


Figura 33: Metanodo Filtragem por Conteúdo

Relativamente à **Filtragem por Estilos de Cozinha**, no metanodo correspondente optou-se por estabelecer associações entre os restaurantes e os estilos de cozinha, entre os utilizadores e os estilos de cozinha e, por fim, entre as duas associações. Posteriormente é realizado um *loop* para obter restaurantes associados aos utilizadores e elaborar uma lista (*GroupBy*) para cada utilizador com esses mesmos restaurantes. Estas novos dados criados são introduzidos num novo *loop* onde, através de um *Association Rule Learner*, se descubra os restaurantes que fazem *match* com os estilo de cozinha dos restaurantes associados aos utilizadores. Posto isto, realiza-se um *Joiner* de todos os dados.

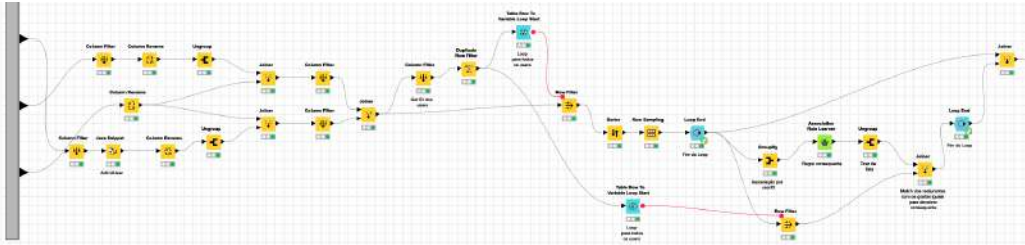


Figura 34: Metanodo Filtragem por Estilo de Cozinha

Finalmente, após a realização de cada tipo de filtragem existe um metanodo, Escrita Mongo, que escreve para a base de dados todos os novos dados gerados e recolhidos.

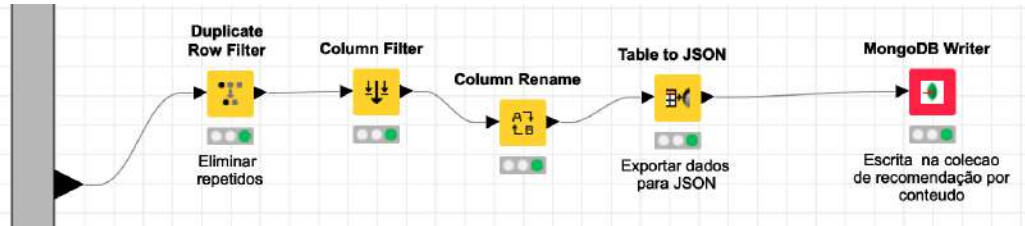


Figura 35: Metanodo Escrita Mongo

6 Sistema Recomendação Desenvolvido

Neste capítulo, apresentamos a interface do Sistema de Recomendação desenvolvido, o ***EatNow***. Esta plataforma é composta por diversas páginas onde serão apresentadas várias sugestões em forma de *TOP-N* tanto da plataforma como de acordo com as preferências dos clientes. Ainda irá ser possível visualizar informações de cada restaurante bem como fazer *reviews* entre outras funcionalidades.

6.1 Home Page

Esta página do sistema de recomendação é composta por várias ferramentas. Inicialmente é possível ao cliente elaborar uma pesquisa filtrada através do preço, localização e estilos de cozinha.

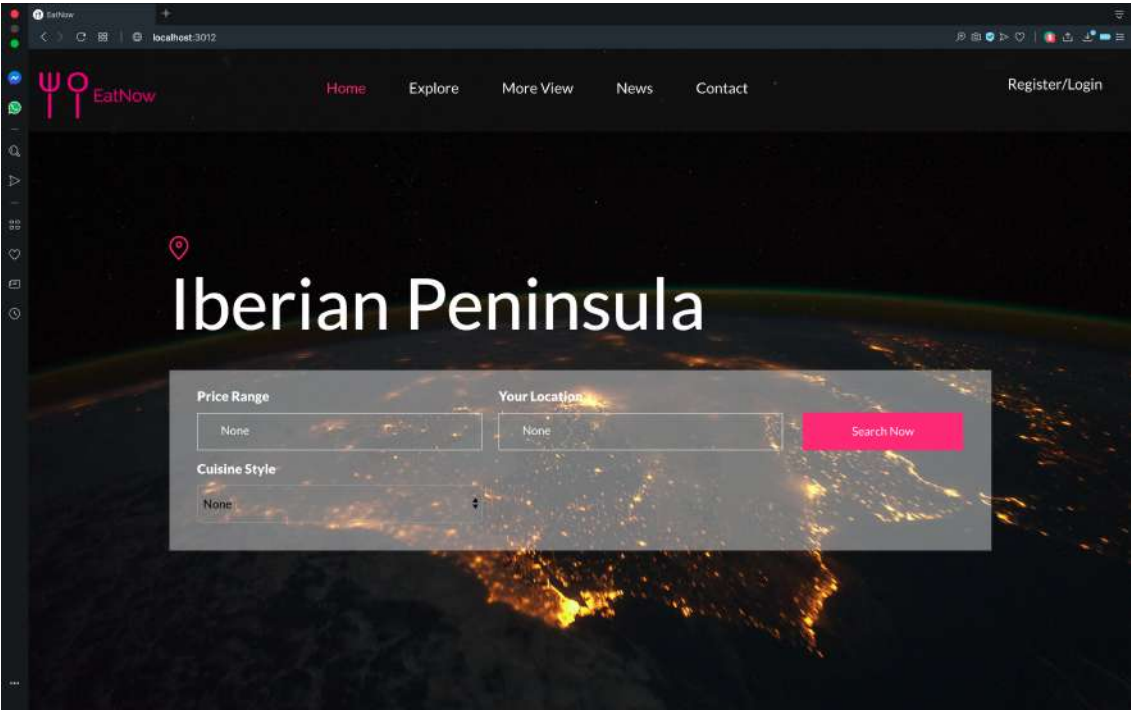


Figura 36: Home Page

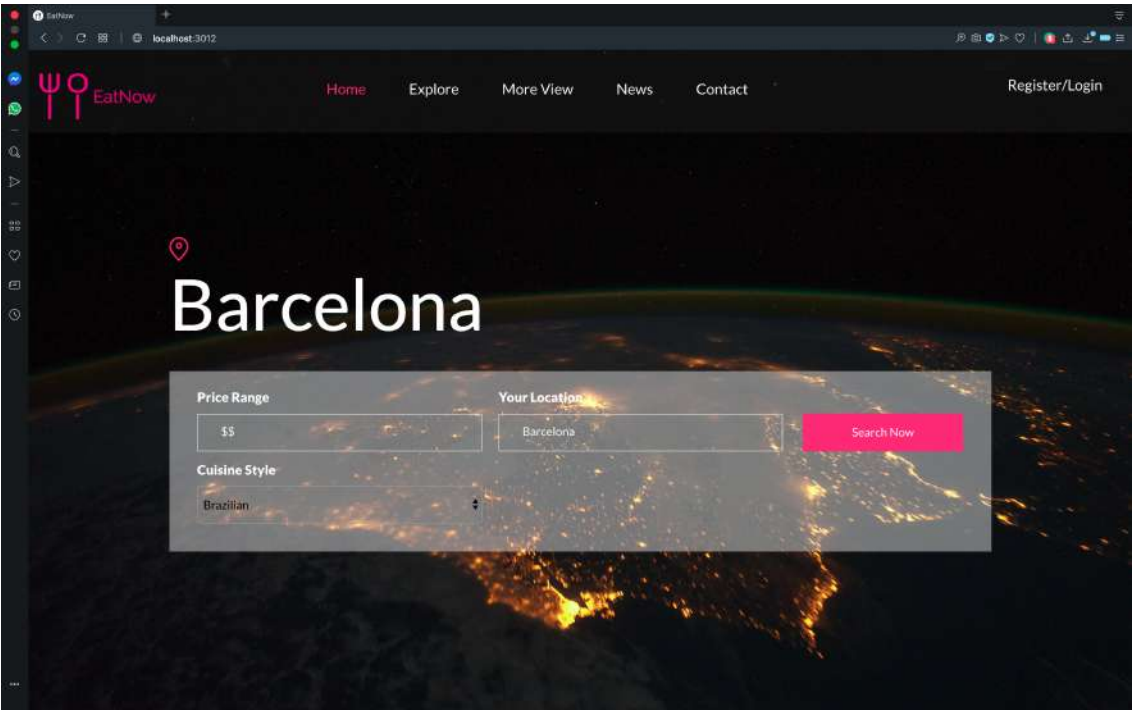


Figura 37: Home Page com filtragem

De seguida é apresentado o Top 10 dos melhores restaurantes da Península Ibérica em termos de *Raking*.

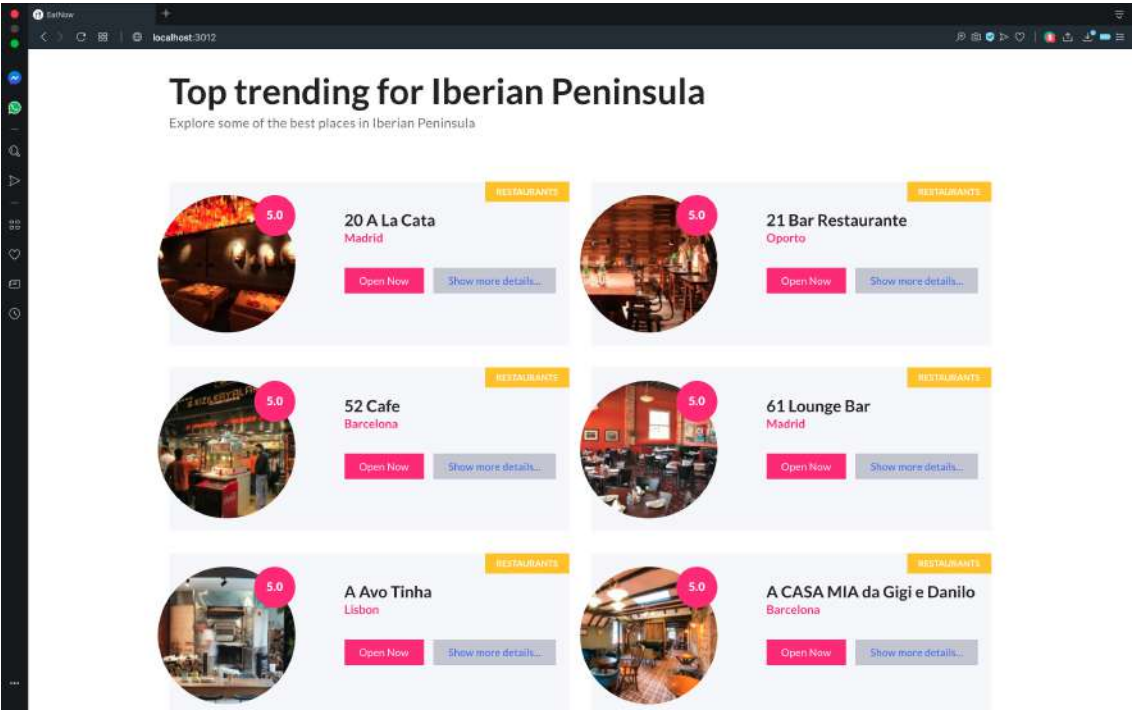


Figura 38: Top sugestões total

Para além da lista anterior, ainda é possível visualizar o Top 10 dos restaurantes mais visualizados na plataforma.

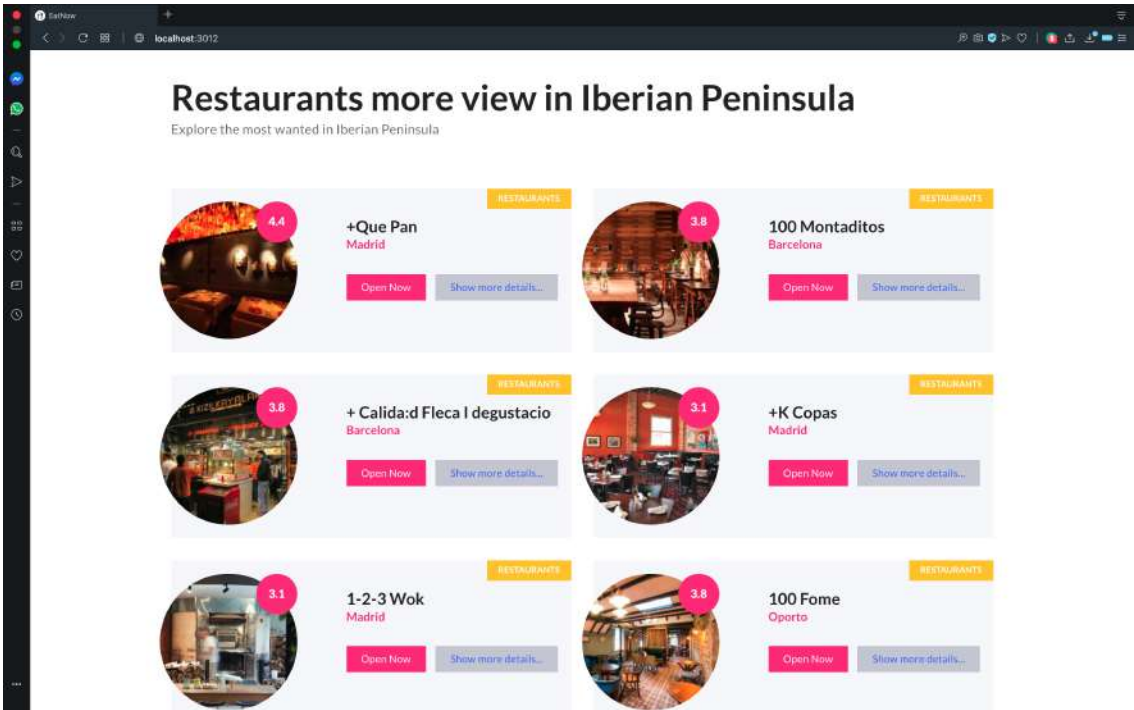


Figura 39: Top sugestões mais visualizados

Também será apresentado ao cliente informação relativa ao número total de restaurantes na Península Ibérica ou, caso tenha efetuado *login*, será apresentado o número total de restaurantes na sua localidade.

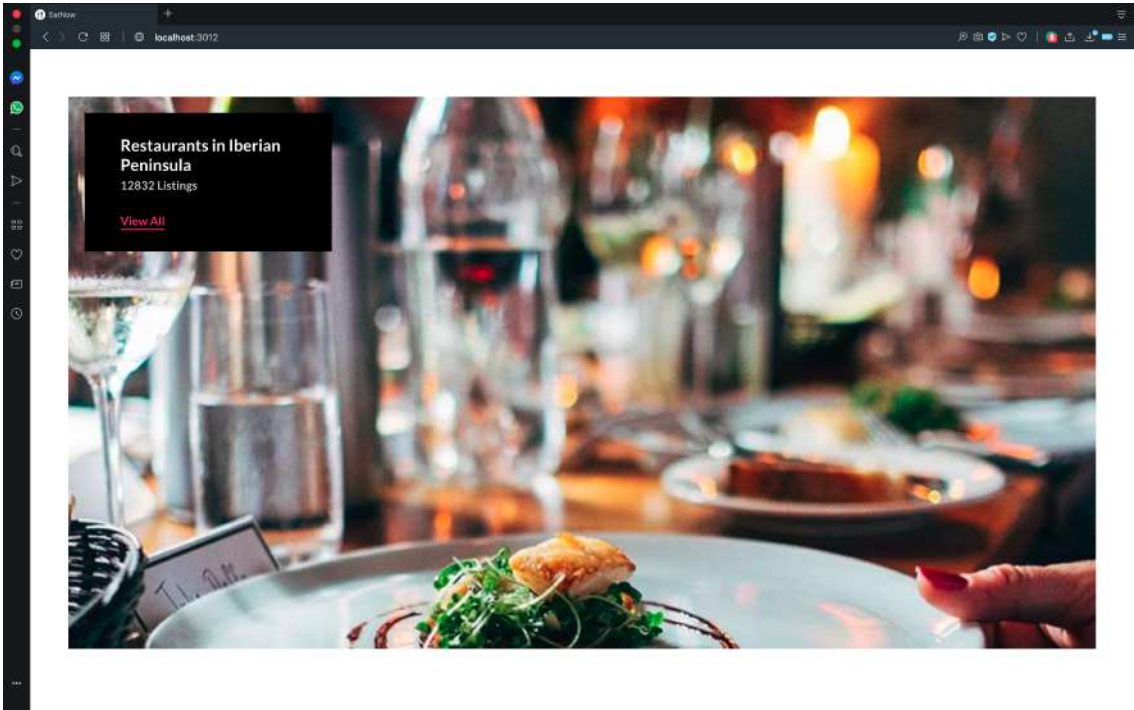


Figura 40: Informação da quantidade de restaurantes

Caso o utilizador já tenha efetuado o *login*, será apresentado, para além do top anterior, também o Top 10 dos melhores restaurantes segundo os seus gostos em termos de estilo de cozinha.

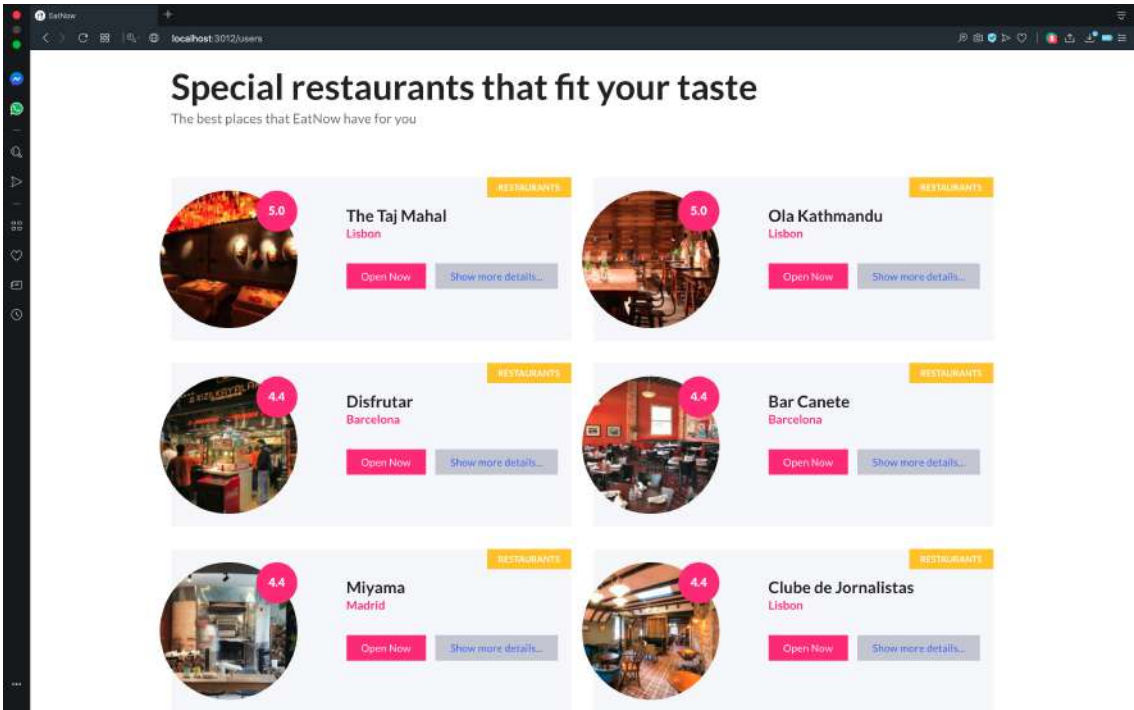


Figura 41: Top sugestões pós registo

Por último, são ainda apresentados vários restaurantes baseados nas *reviews* de outros utilizadores consoante os gostos do utilizador em questão.

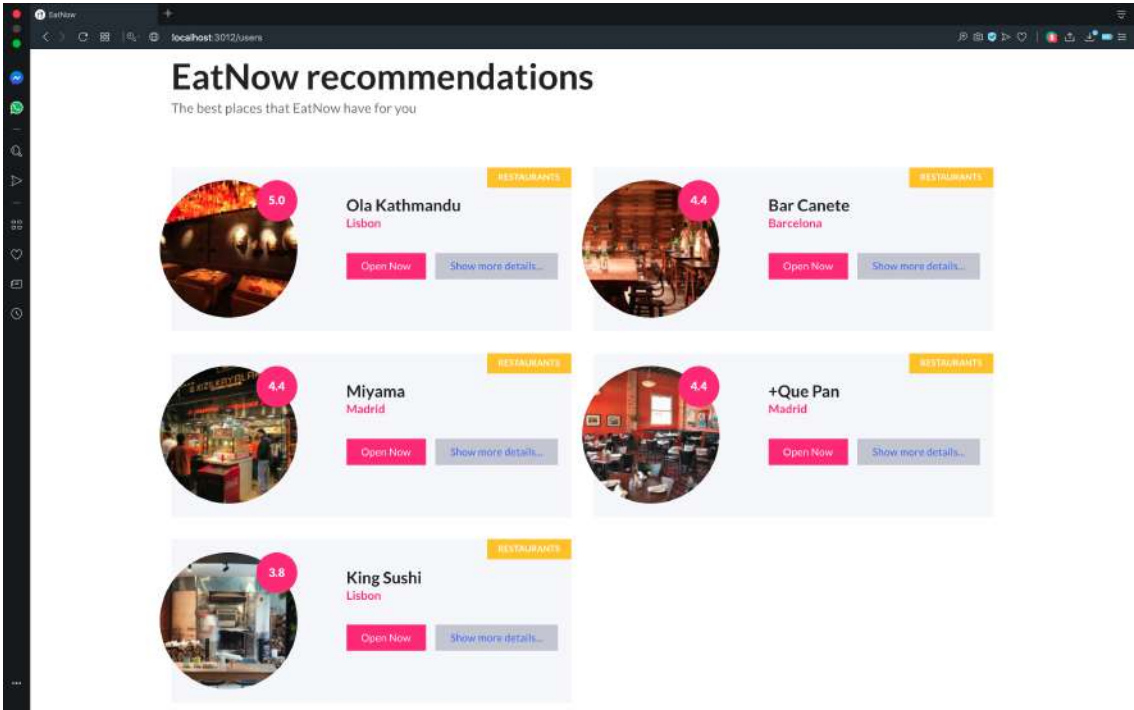


Figura 42: Top sugestões baseadas em *reviews*

6.2 Login/Registar

Quanto ao registo e ao *login*, o cliente precisa de indicar o nome, o *email*, a cidade, a *password* e os estilos de cozinha que mais gosta.

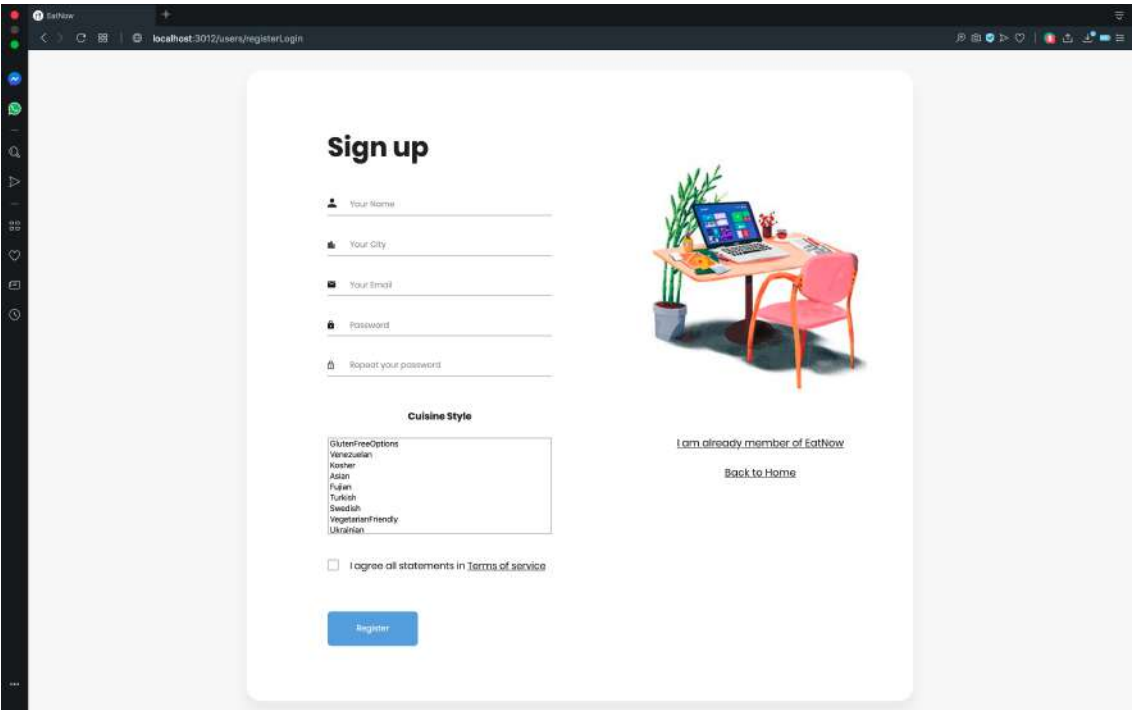


Figura 43: Registo na Plataforma

Já no *login*, basta ao cliente colocar o seu *email* e a sua *password* para se autenticar na plataforma.

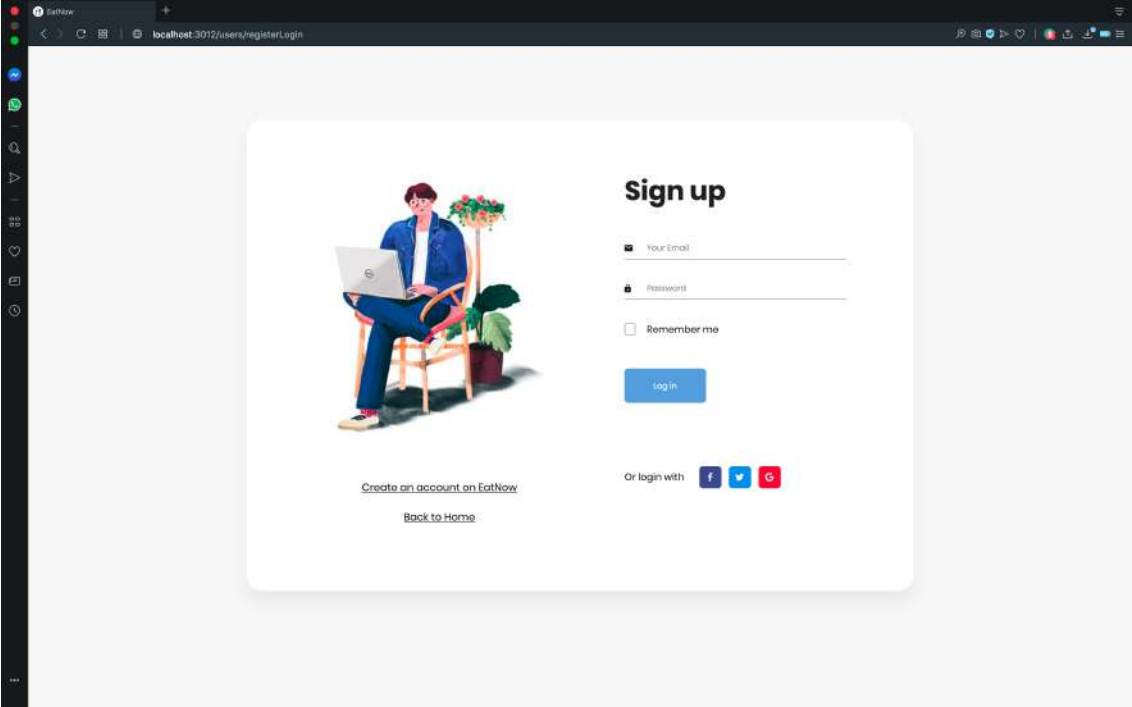


Figura 44: Login na Plataforma

6.3 Explorer

Na página *Explorer* é apresentado ao cliente um mapa da área abrangente do sistema de recomendação, seguido de uma possível filtragem através do estilo de cozinha e localização.

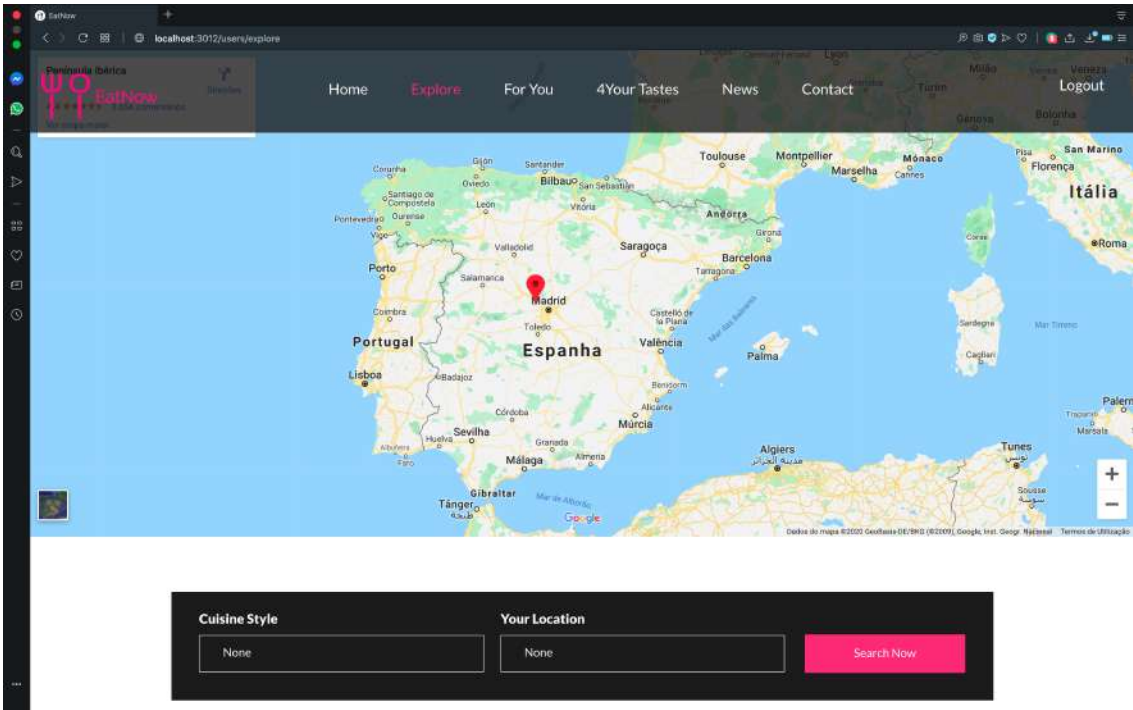


Figura 45: *Explorer*

Na sequência da página oferece-se uma filtragem mais detalhada, onde o cliente pode também selecionar a faixa de preço, os *Ratings* desejados e a distância da sua localização. Após feita a filtragem, pode selecionar um restaurante da lista sugerida.

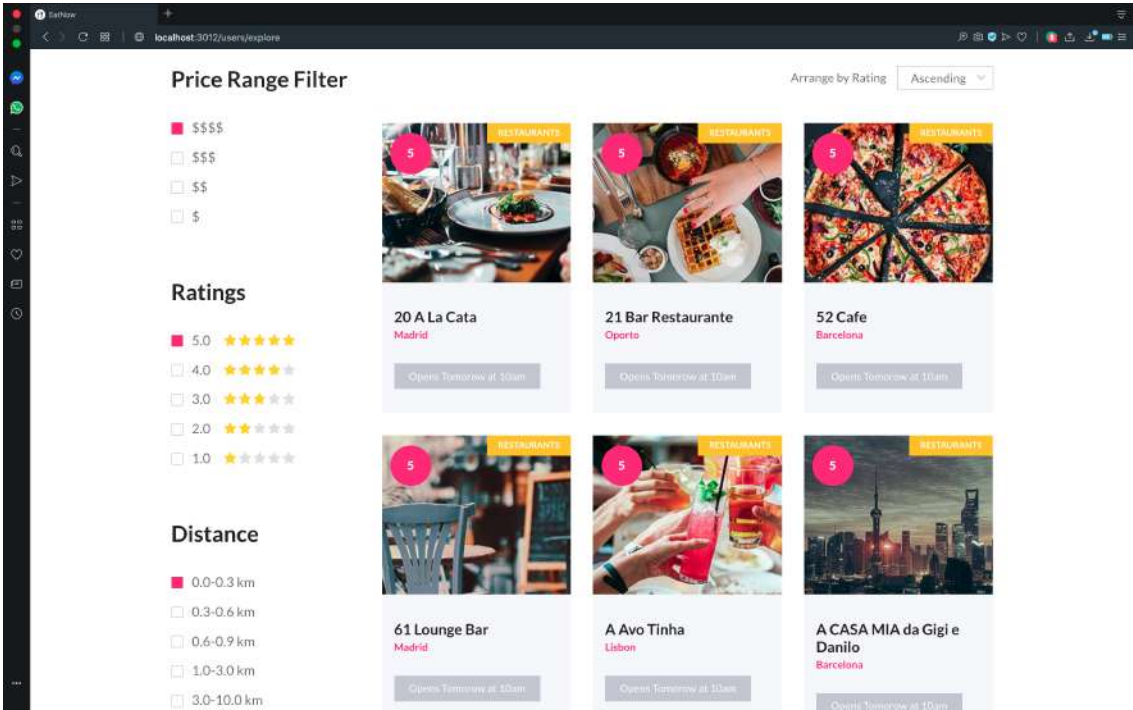


Figura 46: Lista de restaurantes

6.4 Restaurante Individual

Após o cliente selecionar um restaurante, é-lhe apresenta a página individual do restaurante com diversas informações sobre o mesmo. Tendo opção de fazer uma *review* do mesmo.

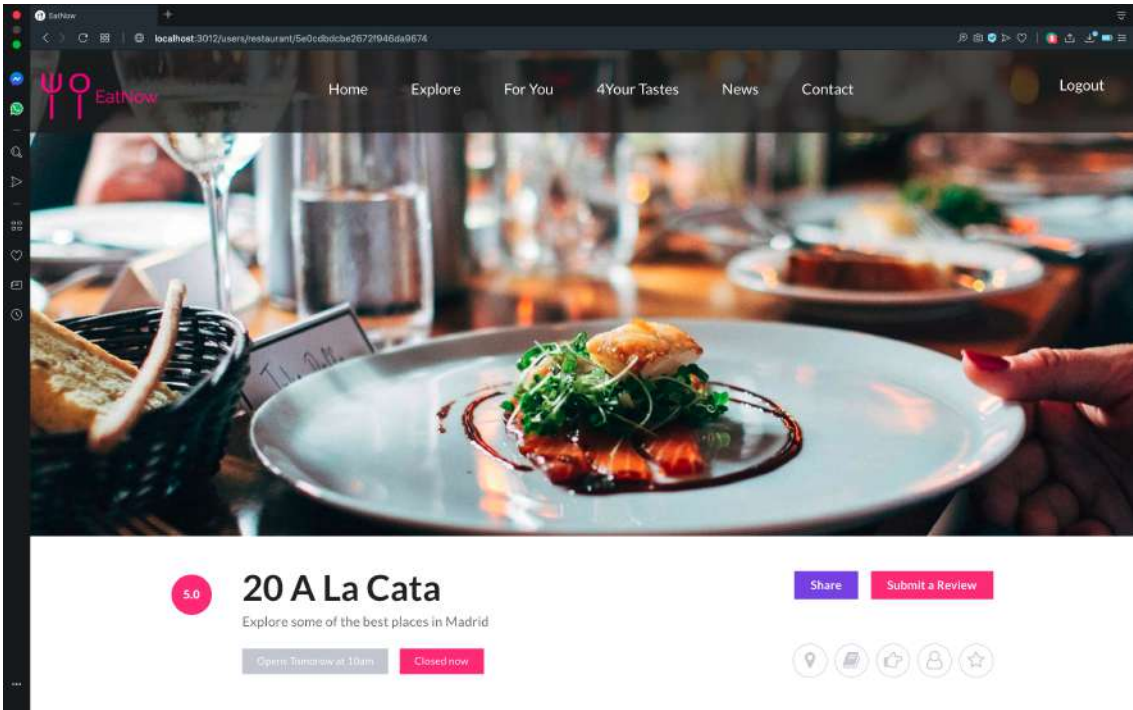


Figura 47: Restaurante individual

É apresentado uma breve descrição do restaurante, bem como um mapa da sua localização e contactos.

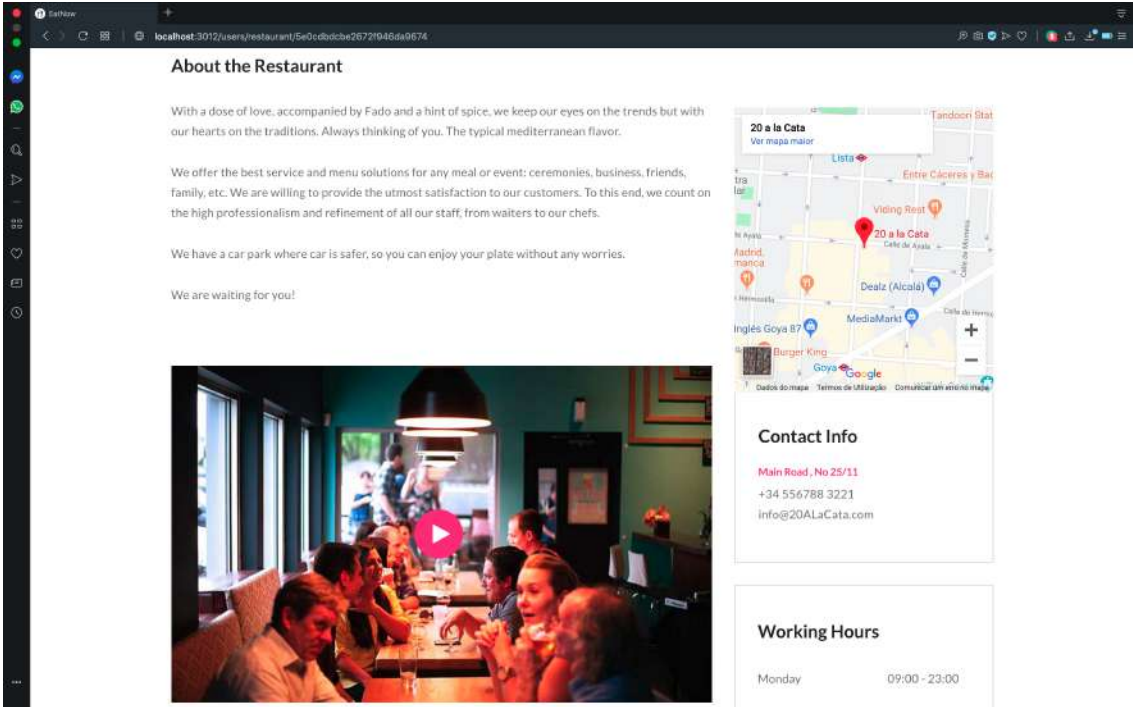


Figura 48: Descrição do restaurante

Existe ainda informação relativa ao horario, estilos de cozinha e faixa de preço. Por último, são colocadas as *reviews* que o restaurante teve.

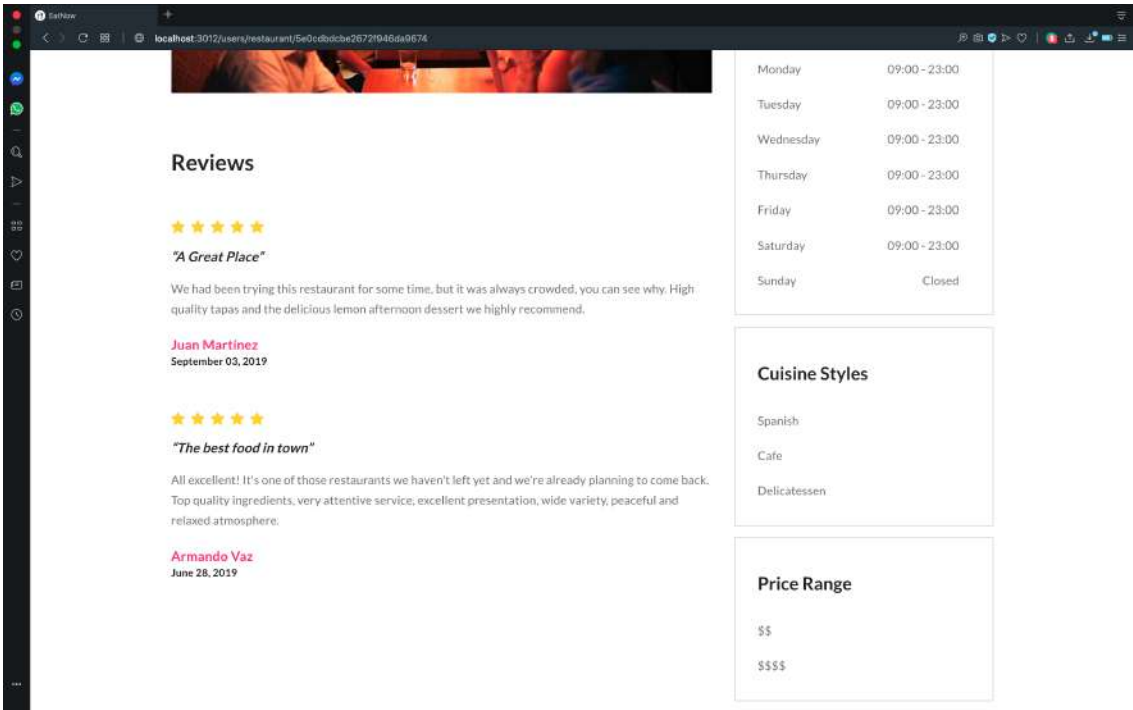


Figura 49: *Reviews* do restaurante

6.5 Review

Quando é selecionada a opção de realizar uma *review* a um restaurante, surge uma página com um mapa do local do restaurante, seguido dos campos necessários a preencher para efetuar a *review*.

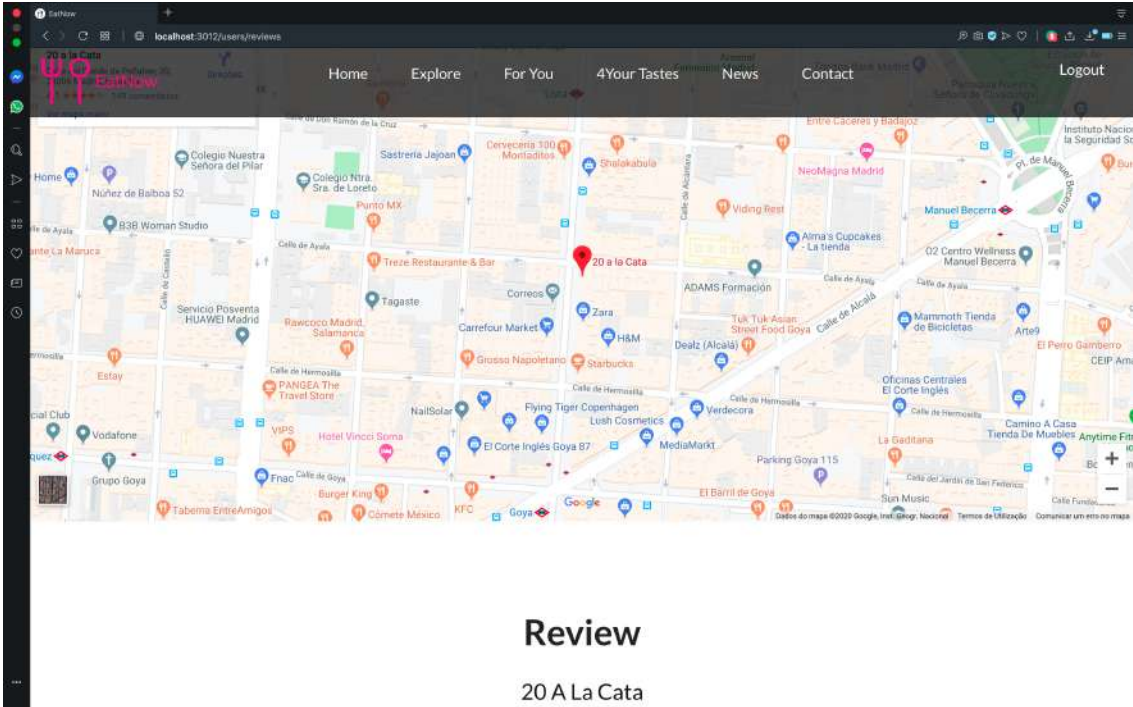


Figura 50: *Review*

The screenshot shows a web browser window with the address bar displaying 'localhost:3012/users/reviews'. The main heading is 'Review' followed by '20 A La Cata'. Below this, there are four input fields: a rating field with the placeholder 'Rating, Ex: 0 to 5', a date field with 'Date of Reviews, Ex: 24 December 2019', a 'Short Comment' field, and a larger 'Full review' field. At the bottom of the form is a pink 'Submit Review' button.

Figura 51: Campos da *review*

6.6 News

Finalmente, a página *News* é o local onde são apresentadas algumas notícias gastronômicas de relevância para o cliente.

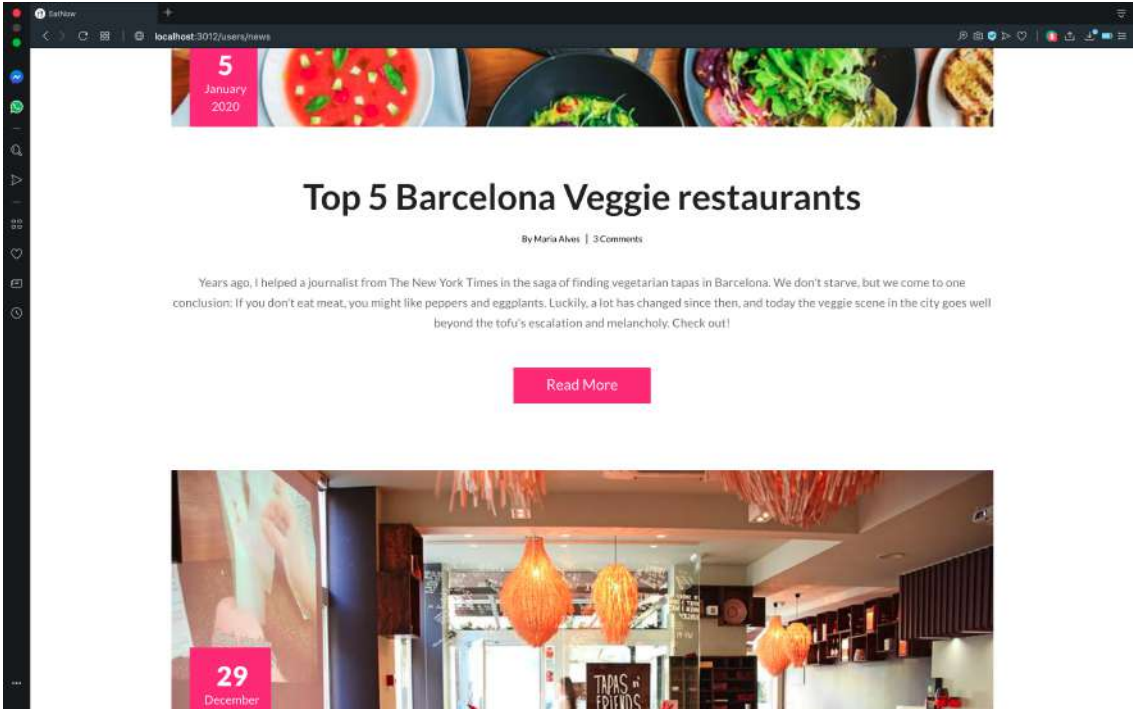


Figura 52: *News*

7 Conclusão

O sistema de recomendação desenvolvido neste trabalho prático apresenta várias estratégias de diversos paradigmas de *Machine Learning*. Como foi mencionado ao longo deste relatório, foram utilizados vários tipos de filtragem, tais como todas as estratégias Top-N utilizadas, as Filtragens Colaborativa e por Conteúdo, permitindo classificar o nosso Sistema, o **EatNow**, como um **Sistema de Recomendação Híbrido**.

Podemos realçar que, neste projeto, o grupo tentou resolver vários problemas como, por exemplo, o *cold start* através de filtragem baseada nos gostos do utilizador, uma vez que qualquer cliente ao registar-se na plataforma terá de indicar os estilos de cozinha que mais se identifica permitindo fazer uma filtragem inicial para esse cliente dos top-n restaurantes conforme os seus gostos.

Uma outra conclusão que não podemos deixar de destacar, é o facto de o sistema poder realizar sempre recomendações independentemente do utilizador estar ou não autenticado, tal como foi explicitado anteriormente.

Por fim, é de enfatizar que, de um modo geral, todo o projeto desenvolvido correspondeu a expectativas e que o grupo soube sempre superar as dificuldades que foi encontrando com o decorrer do mesmo.