AASMA Relatório Final - Fridge&Pantry

André Patrício
87631 - MEIC-T
Instituto Superior Técnico
Torres Vedras - Lisboa
andremppatricio@gmail.com

Diogo Viegas
87649 - MEIC-T
Instituto Superior Técnico
Cacém - Lisboa
diogo.viegas@tecnico.ulisboa.pt

Francisco Barata
87656 - MEIC-T
Instituto Superior Técnico
Cacém - Lisboa
franciscofariabarata@tecnico.ulisboa.pt

1. DEFINIÇÃO DO PROBLEMA E SOLUÇÃO IMPLEMENTADA

Como mencionado na nossa proposta decidimos abordar uma temática relevante na atualidade visto que vivemos uma fase nunca antes experienciada. A pandemia de COVID-19 que nos assola alterou o nosso dia-a-dia expondo alguns problemas que estavam já inerentes na nossa sociedade. Um desses problemas é comprovado por estudos que mostram que cerca de 30% da comida produzida em todo o Mundo acaba inevitavelmente desperdiçada. No outro lado da moeda, algo nunca antes visto nestas proporções, a ONU, em 2019, comunica que a fome atinge um total de 820 milhões de pessoas (https://nacoesunidas.org/fome-aumenta-no-mundo-e-atinge-820-milhoes-de-pessoas-diz-relatorio-da-onu/). É portanto imperial que certos hábitos de desperdício sejam evitados e substituídos com hábitos de partilha e cooperação entre as comunidades. Para além desta problemática, esta pandemia revelou ainda a necessidade de recomendação de refeições equilibradas com o que temos no nosso frigorífico/despensa, tentando minimizar idas ao supermercado. Estas sugestões também têm uma maior utilidade nos tempos que correm devido ao facto de muitos trabalhadores se encontrarem em regime de Teletrabalho ou de Lay-off, ficando nas suas casas, fazendo com que muitos estabelecimentos de restauração deixassem de servir a refeição do almoço aos seus clientes diários, os trabalhadores. Deixando de se deslocar para o local de trabalho muitas pessoas passaram a fazer as suas refeições em casa, algo que alguns não estavam acostumados a fazer. Isto levou a mudanças repentinas na quantidade de alimentos que é necessário ter em casa e o número de vezes que é feita a deslocação até um supermercado, pensando na sua minimização para reduzir possibilidades de contágio.

A tentativa de solução para os problemas anteriormente referidos motivou-nos para a nossa proposta de projeto para a unidade curricular de Agentes Autónomos e Sistemas Multi-Agentes. A nossa proposta consiste num sistema que recomenda refeições equilibradas para uma ou mais pessoas presentes numa casa, tendo em conta alguns dados biométricos dos utilizadores e dados dos alimentos (quantidade, data de validade, etc.). Aliada a essa recomendação pretendemos incentivar a partilha de alimentos entre pessoas da mesma comunidade, reduzindo o desperdício alimentar.

Desenvolvemos então um sistema multi-agente composto por vários agentes *Fridge&Pantry* onde os utilizadores podem armazenar os seus alimentos. Cada agente pode, através de conhecimento pré-adquirido como: dados biométricos do utilizador (peso, altura, idade e género), preferências alimentares e prazo de validade dos alimentos, recomendar refeições. Uma *feature* que também decidimos implementar foi a lista de compras, que vai sendo formada em paralelo com o pedido em sugestões de receitas. Esta lista de compras tem como base a ideia que certos grupos de utilizadores têm gostos mais específicos, sendo assim, o *Fridge&Pantry* aprende que ingredientes são da preferência dos seus utilizadores e adiciona-os à lista de compras quando eles já não existem no agente ou se encontram em pouca quantidade.

2. AMBIENTE DE SIMULAÇÃO E COMPORTAMENTO DO AGENTE

Ao longo do projeto o ambiente que idealizamos para a nossa simulação foi um condomínio. Nesse condomínio as casas estavam equipadas com um *Fridge&Pantry*.

Cada um destes *Fridge&Pantry's* encontra-se conectado à Unidade Central (*CentralUnit*). Esta unidade tem um papel fundamental na vertente multi-agente do nosso projeto que é a partilha de alimentos. Pensando no exemplo em que possuo quase todos os ingredientes necessários para realizar uma refeição saudável e do meu agrado, mas falta-me um ingrediente, o *Fridge&Pantry* emite um pedido para a Unidade Central para saber se alguém do condomínio tem esse ingrediente em quantidade suficiente para ser realizada a refeição. Olhando para o agente implementado existe uma componente fulcral de cooperação entre todos os agentes. Cada um tem o objetivo de alimentar da melhor forma possível todos os residentes do condomínio. Assim sendo, após a decisão ser tomada pela Unidade Central o *Fridge&Pantry* dador realiza a transferência.

Na proposta de tema para o projeto, entregue anteriormente, já tinham sido referidas as propriedades do nosso agente, sendo as seguintes:

- Adaptatividade - o agente ao longo do tempo molda-se aos gostos dos utilizadores ao receber no final da refeição o *feedback* e associar esse valor aos ingredientes utilizados na refeição
- Racionalidade - faz uma escolha racional na sugestão de receitas avaliando as datas de validade dos ingredientes armazenados, o consumo médio calórico por refeição e o gosto pessoal dos utilizadores
- Curiosidade - nas sugestões que apresenta, o nosso agente sugere uma receita que ainda não tenha sido preparada com o objetivo de obter *feedback*

- Reatividade - o agente reage a diversos *inputs* do utilizador
- Proatividade - ao longo do tempo é formada a lista de compras que sugere que sejam adquiridos alimentos do agrado dos utilizadores e também em certa quantidade para um X número de refeições
- Sociabilidade - como explicado anteriormente os nossos agentes colaboram em todos os momentos uns com os outros partilhando o mesmo objetivo

O nosso ambiente de simulação apresenta as seguintes características:

- Acessível - em qualquer momento é possível aceder aos conteúdos de todos os *Fridge&Pantries* através da Unidade Central
- Não Determinístico - no momento de pedir uma sugestão pode ocorrer que tanto o agente como os agentes mais próximos não tenham ingredientes para fazer a sugestão de uma receita. Existe portanto um não determinismo.
- Estático - enquanto o agente delibera, com o objetivo de apresentar as sugestões, o seu conteúdo não pode ser alterado
- Discreto - o conjunto de receitas é finito, havendo um limite de receitas que o agente pode sugerir
- Não-Episódico - ações escolhidas têm impacto em decisões futuras

Outro aspecto em que também nos debruçamos foi como seria a interação dos utilizadores com o agente. Pensamos então que cada *Fridge&Pantry* teria um ecrã tátil onde pudessem ser selecionados diferentes opções (pedir sugestões, mostrar lista de compras e na escolha da receita a realizar) e teria também um reconhecimento de voz para outro tipo de instruções (na adição de novos alimentos e para dar feedback no final da receita).

Desenvolvemos então vários ficheiros de texto para simular conjuntos de comandos e serem executados por diferentes agentes numa determinada ordem, com o objetivo de observarmos o comportamento dos agentes e as mudanças nas nossas métricas de avaliação. Na Figura 1 é apresentado um exemplo de ficheiro de *input*, neste caso para testar a realização de uma refeição sem pedir ingredientes a outros.

```
1  add FridgePantry 1
2  add User Carla F 65 70 160 [] [] []
3  add User Maria F 60 60 165 [] [] []
4  add Ingredient 1 Abobora 500 01/01/2021
5  add Ingredient 1 Brocolos 600 01/01/2021
6  add Ingredient 1 Grao 400 01/01/2021
7  add Ingredient 1 Batata-Doce 500 01/01/2021
8  add Ingredient 1 Cebola 100 01/01/2021
9  add Ingredient 1 Alho 10 01/01/2021
10 get Suggestion 1
11 choose Suggestion 1 1
12 end
13
14 testa a remocao de alimentos de uma receita que um
15 frigorifico consegue fazer com ingredientes exclusivamente seus
```

Figura 1 - exemplo de teste

Olhando para a Figura 1, observamos parte dos comandos que criamos:

- *add FridgePantry 1* - adicionada à Unidade Central um *Fridge&Pantry*
- *add User* - adiciona ao *Fridge&Pantry* um utilizador com gênero, idade, peso, altura, alergias, gostos e desgostos, pela respectiva ordem
- *add Ingredient 1* - adiciona ao *Fridge&Pantry 1* o ingrediente, a sua quantidade e data de validade
- *get Suggestion 1* - pede as sugestões de receitas para o *Fridge&Pantry 1*
- *choose Suggestion 1 1* - é feita a seleção 1 na lista de sugestões do *Fridge&Pantry 1*

Temos também outros comandos que não são mostrados no exemplo e que desempenham funções chave:

- *give Feedback 1* - no final da refeição do *Fridge&Pantry 1* é dado um feedback entre 0 e 5 que é atribuído à receita
- *show ShoppingList 1* - Apresenta no ecrã a lista de compras do *Fridge&Pantry 1*

3. PRINCÍPIOS ARQUITETURAIS E DETALHES DE IMPLEMENTAÇÃO

Para esta nossa proposta usámos dois *datasets* distintos criados por nós. O primeiro *dataset* consiste em receitas retiradas manualmente de https://www.pingodoce.pt/receitas/. Cada receita contém o seu nome, a sua lista de ingredientes, a lista das quantidades de cada ingrediente, o seu teor calórico, o número de pessoas e a descrição da preparação da mesma. Foi ainda necessário assegurar que ingredientes iguais em receitas diferentes seguem a mesma ordem de quantidades, ou seja, se na receita X for usada 1 cebola e noutra receita Y usar 100g de cebola, normalizamos de forma a que 1 cebola correspondesse a 100g. O segundo *dataset* foi criado para haver um mapeamento entre os ingredientes e as suas regras de partilha. Este mapeamento consiste numa lista de ingredientes (todos os ingredientes presentes nas receitas) e num valor 0 ou 1 correspondente a cada alimento. Caso o valor seja 0 uma pessoa pode pedir qualquer quantidade desse alimento aos outros utilizadores, não precisando de ter nenhuma quantidade a priori. Caso seja 1 passa a ser preciso ter alguma quantidade desse alimento para poder pedir aos outros utilizadores (e.g. proteínas).

Como descrito anteriormente, a nossa proposta consiste em 3 funcionalidades principais, das quais: sugestão de receita, adição automática de ingredientes à *Shopping List* e partilha de ingredientes entre FPs (*Fridge&Pantries*) da mesma rede.

3.1 Sugerir uma receita

Esta funcionalidade permite ao utilizador de um FP pedir uma sugestão de receitas a qualquer momento, sendo retornado uma lista com um número que variável de sugestões, entre 0 e 5, consoante a possibilidade de fazer refeições com os ingredientes introduzidos no sistema. A ordenação das receitas foi determinada com base num algoritmo que será explicado mais à frente em pormenor. De realçar ainda que cada FP tem a sua própria data. Data essa que é incrementada a cada 2 receitas pedidas (de forma

a simular almoço e jantar). Esta data será útil para nos podermos guiar em relação ao prazo de validade de cada alimento.

No que toca ao algoritmo de geração de sugestões, o FP filtra primeiramente todas as receitas possíveis. Neste passo são retiradas todas as receitas que não são possíveis de se fazer com os ingredientes disponíveis (quer do próprio FP que pediu a sugestão, quer dos restantes FPs da rede) assim como aquelas que possuem ingredientes cujos utilizadores são alérgicos. Pode ainda haver o caso em que o utilizador pretende obter apenas sugestões a partir dos ingredientes que ele próprio possui (i.e se não quiser receber ingredientes de outros FPs por qualquer razão, ainda que não vá ao encontro da nossa ideia original). Nesse caso, uma *flag* é ativada, evitando que receitas que necessitem de ingredientes não possuídos pelo utilizador sejam filtradas. Assumimos ainda que todos os utilizadores possuem os ingredientes de ordem qb.

Após esta primeira filtragem, as receitas restantes vão ser categorizadas de acordo com 3 categorias fundamentais na nossa óptica: prazo de validade dos ingredientes das receitas (de modo a dar melhor classificação às receitas cujos ingredientes estão mais próximos do término da data de validade), teor calórico da receita, ou seja, através de um processo que será explicado num ponto mais à frente, conseguimos prever qual o teor calórico ideal que determinada receita devia ter, sendo então possível categorizar as receitas de acordo com o desvio face a esse valor ideal e, finalmente, o gosto pessoal de cada utilizador.

3.1.1 Prazo de Validade

Para cada ingrediente presente na receita, é feita a diferença entre o seu prazo de validade e a data do frigorífico (dia em que será realizada a refeição). Soma-se todos os valores para cada ingrediente e normaliza-se com o número de ingredientes na receita. Uma receita é tanto melhor classificada quanto menor for este valor, ou seja, as primeiras opções usam ingredientes com data de validade próxima de acabar

3.1.2 Teor Calórico

Para esta ordenação assumimos que cada utilizador de um FP tem uma quantidade calórica média que deve ingerir por refeição. Para obter esse valor, utilizamos os dados biométricos de cada utilizador recorrendo à seguinte fórmula da taxa de metabolismo basal presente em https://www.vidaativa.pt/necessidades-energeticas-diarias/.
Assumindo ainda que cada utilizador tem um fator de atividade baixo, conseguimos obter a necessidade energética diária, ou seja, a quantidade calórica que determinada pessoa deve ingerir por dia de modo a seguir um estilo de vida saudável. Tendo este valor conseguimos então prever qual a quantidade calórica que cada pessoa deve ingerir por refeição.

De forma a obter uma classificação para cada refeição, assumimos que o utilizador vai consumir a refeição e adicionamos o valor calórico da mesma a uma *Moving Average* que calcula a média calórica das refeições consumidas nos últimos 7 dias. Quanto menor a diferença entre o valor estimado pela *Moving Average* e o valor calculado a partir da fórmula das necessidades calóricas, melhor será classificada a receita. Ficando assim nas primeiras posições refeições que caso fossem consumidas aproximavam os utilizadores da média calórica que eles deviam consumir por refeição.

3.1.3 Gosto Pessoal

Relativamente à classificação de receitas por gosto pessoal de cada utilizador, usámos aprendizagem para determinar quais as receitas que serão mais prováveis de obter bom feedback por parte do utilizador. Para tal, atribuímos pesos aos ingredientes que serão atualizados conforme o feedback dado pelo utilizador a partir da fórmula:

*newIngredientWeight = oldIngredientWeight * (1 - weightUpdate) + user Feedback*weightUpdate*

Sendo que se trata de uma técnica de *Reinforcement Learning* é necessário um *learning rate* que neste caso é o *weightUpdate*.

A partir do momento que o peso de cada ingrediente é atualizado, podemos então obter qual a possibilidade do utilizador gostar de determinada receita. Para tal, basta somar o valor dos pesos de cada ingrediente e normalizar com o número de ingredientes de cada receita. Quanto maior for este valor, melhor classificada é determinada receita.

Finalmente, tendo as três listas de receitas ordenadas conforme descrito anteriormente, podemos então calcular uma medida de agregação geral que tenha em conta estes 3 valores. Para isso, vamos dar pesos a cada medida. Inicialmente os pesos vão ser uniformes, mas conforme os resultados obtidos podem variar, por exemplo caso um *Fridge&Pantry* esteja a mandar uma grande quantidade de comida fora o peso das receitas vindas da lista por prazo de validade pode aumentar.

De modo a assegurar que os agentes têm curiosidade nas suas ações, decidimos assumir que a 5ª sugestão é uma receita que o utilizador nunca consumiu, de entre as possíveis.

3.2 Adicionar à *Shopping List*

O agente tem ainda a funcionalidade de adicionar ingredientes apreciados pelo utilizador e que se encontramos em quantidade reduzida ou já não existem no FP à sua lista de compras. Caso o ingrediente tenha um peso superior a um certo valor, numa escala de 0 a 5, e a quantidade do ingrediente guardada não for necessária para a realização de 2 refeições é adicionado à *Shopping List.* Para tal, calculamos a média da quantidade de cada ingrediente em cada umas das receitas e utilizamos esse valor na fórmula descrita previamente.

3.3 Partilha de ingredientes

Esta funcionalidade foi implementada com o objetivo de combater o desperdício alimentar e de minimizar o número de deslocações às compras.

Uma vez que todos os agentes partilham o mesmo objetivo, sempre que é feito um pedido à Unidade Central de um alimento numa certa quantidade, esta vai encontrar qual o FP que possui o ingrediente a ser pedido e cuja data de validade está mais próxima do término. Este FP “dador” irá dar a quantidade pedida sem esperar algo em troca, numa relação de simbiose entre agentes.

4. ANÁLISE EXPERIMENTAL

4.1 Definição do Ambiente

Para a análise experimental do nosso sistema de agentes foram definidos 2 ambientes. Estes ambientes consistem de um grupo de *Fridge&Pantries* com utilizadores associados. Os utilizadores têm associados a si os dados biométricos já mencionados e 3 conjuntos de ingredientes correspondentes às suas alergias, gostos e desgostos, sendo que estes 2 últimos grupos são desconhecidos pelos FPs. De realçar que foram tomados os cuidados necessários para que um número muito diferente de gostos/desgostos não levasse a um bias positivo/negativo no feedback geral do utilizador. Visto que as alergias apenas levam à diminuição do número de receitas possíveis, não foram tidas em conta em nenhum dos ambientes.

De forma a melhor testar o desempenho do nosso sistema, atribuímos capacidade de decisão aos nossos utilizadores "mock". Estes têm a capacidade de **escolher uma das sugestões** sugeridas pelo FP, de **dar feedback** acerca da receita escolhida e de **reabastecer o seu FP** baseando-se na lista de compras gerada. Os comportamentos referidos estão implementados da seguinte forma:

- **Escolha de sugestões -** tendo em conta que os utilizadores na sua generalidade têm tendência a escolher os primeiros elementos de qualquer lista que lhes seja apresentada, os nossos "mocks" vão escolher 1 das 5 sugestões do FP seguindo a distribuição [0.3, 0.3, 0.2, 0.1, 0.1].
- **Feedback à receita -** numa receita que não possua nenhum ingrediente que o utilizador goste ou desgoste, então o feedback dado será um valor aleatório entre 2 e 3. Este intervalo é movido consoante o número de ingredientes que gosta/desgosta na receita, sendo que cada um move o intervalo por +1/-1. Por exemplo no caso de uma receita com 2 ingredientes "bons" e 1 "mau", o feedback será um valor entre 3 e 4.
- **Reabastecer FP -** o utilizador vai automaticamente, de X em X refeições reabastecer o seu FP com os ingredientes presentes na lista de compra. O valor X foi definido por nós e é constante.

4.1.1 Ambiente 1 - Análise de Grupo

Neste ambiente inicial, possuímos 4 FPs, cada um com 1 utilizador associado (o agente está preparado para ter em conta gostos de grupos de utilizadores, usamos apenas 1 para simplicidade). Estes utilizadores possuem diferentes dados biométricos e "personalidades" definidas pelos seus gostos alimentares. São os seguintes:

FP 1:

Carla, Feminino, 35 anos, 70kg, 1.60 metros. Gosta de carne de aves e peixe. Desgosta de carnes vermelhas e enchidos.

FP 2:

Nelson, Masculino, 25 anos, 75kg, 1.70 metros. Possui gostos inversos aos da Carla.

FP 3:

Carminho, Feminino, 50 anos, 50kg, 1.50 metros. Gosta de arroz, carne, e algumas carne processadas. Desgosta de massas e peixe.

FP 4:

Marcelo, Masculino, 65 anos, 70kg, 1.70 metros. Tem gostos vegetarianos, gosta de vegetais e leguminosas. Desgosta de carnes e peixe.

Os alimentos iniciais em cada FP são compostos por uma base de alimentos igual para todos (acompanhamentos, proteínas e legumes comuns, entre outros), e um conjunto de ingredientes mais "especializados" (condimentos e alimentos menos comuns) que diferem um FP de outros FP.

Este ambiente conta com 50 iterações, realizadas igualmente por todos os FPs, constituídas por: pedido sugestões, escolha uma delas e feedback adequado.

4.1.2 Ambiente 2 - Análise Individual

Este ambiente, é constituído por apenas um FP, cujo utilizador é idêntico ao utilizador Carla do Ambiente 1. Os alimentos iniciais são também idênticos aos do FP1 do Ambiente 1.

Este ambiente visa comparar as diferenças de 2 FPs idênticos, com e sem cooperação com outros FPs.

O ambiente conta com 50 iterações.

4.2 Definição de Métricas

De forma a avaliarmos o desempenho de cada um dos nossos agentes, definimos 3 métricas principais:

- ***Mean Feedback* -** corresponde à média de feedbacks dados pelo utilizador ao longo de todas as receitas executadas.
- ***Trash Value*** - corresponde ao valor em kg de comida que foi desperdiçada por ter passado de validade. No caso de ingredientes que são guardados como unitários (ex: Ovo), foi lhes atribuído um valor de peso estimado. Esta métrica não é adequada para uma previsão de valores de desperdício, visto que os nossos FPs são inicializados com ingredientes e quantidades *default* que normalmente poderiam não ser comprados pelos utilizadores por não gostarem deles (o que inflaciona os valores desta métrica). É no entanto possível, fazer uma análise comparativa através da subida/descida deste valor.
- ***Calorie Intake Deviation* -** corresponde à diferença absoluta entre o valor calórico recomendado (calculado através dos dados biométricos) e o valor calórico médio consumido pelo(s) utilizador(es). O valor calórico médio é calculado através de uma média pesada, de forma a dar mais peso a consumos recentes.

É também analisado, no caso de um grupo de FPs, o **número de trocas de ingredientes** entre os vários membros desse grupo.

4.3 Resultados Experimentais

De forma a reduzir o efeito dos fatores aleatórios da nossa implementação, todos os valores apresentados são correspondentes à média dos valores adquiridos ao longo de 20 *runs* de cada ambiente.

4.3.1 Ambiente 1

A primeira *run*, tida como *default*, foi efectuada com os pesos previamente mencionados com os valores: wed=⅓, wci=⅓, wst=⅓ (wed:*Expiration date*, wci: *Calorie Intake*, wst:*Self-Taste*). O update dos pesos dos ingredientes é feito com um *learning rate* de 0.7. Um ingrediente é inicializado com peso 2.5 e é adicionado à lista de compras se possuir um peso superior ou igual a 2.6.

Obtemos as seguintes tabelas (com valores arredondados):

	Feedback	Trash Value	Deviation CI
F1	3	31.9	35
F2	2.94	20	70
F3	2.67	25	44
F4	3.88	35	48
Mean	3.19	28	47

Tabela 1 - valores finais das métricas da primeira *run*

Através da análise desta Tabela 1 podemos concluir que:

- Com os parâmetros atuais, os nossos agentes são efectivamente capazes de obter uma satisfação superior à que seria obtida com sugestões aleatórias (média de *feedback* **3.19**).
- Parte desta eficácia provém também da funcionalidade *Shopping List*, que apenas sugere a compra de ingredientes com um weight "positivo" associado.
- Conseguimos também bons resultados no que toca ao *calorie intake* adequado, com um desvio médio de apenas **47**.
- Não tiramos conclusões acerca do *Trash Value* pelas razões mencionadas na secção 4.2.

É ainda passível de se mencionar que ambos os valores mais reduzidos/elevados dos FPs 2 e 4 podem ser explicados pelo facto de a nossa base de dados de receitas ser maioritariamente de receitas vegetarianas ou com carnes brancas/peixe. Isto enquadra-se com os gostos do utilizador do FP 4 mas vai contra os gostos do utilizador do FP 2.

F1<>F2	F1<>F3	F1<>F4	F2<>F3	F2<>F4	F3<>F4
24	30	19	22	15	26

Tabela 2 - trocas entre Fps

Total F1	Total F2	Total F3	Total F4
73	61	78	60

Tabela 3 - número total de trocas

A Tabela 2 acima representam o número de trocas entre FPs, com F1<>F2 a representar o número de trocas entre o FP 1 e o FP2, e Total F1 a representar o número total de trocas que o FP 1 realizou, representado na Tabela 3.

Não fazemos diferenciação entre dar/receber ingredientes pois estamos num ambiente cooperativo, onde não nos interessa analisar as diferenças entre cedências e pedidos.

Ao analisar as tabelas anteriores, é de realçar os seguintes pontos:

- O maior número de trocas ocorre entre o FP 1 e o FP 3, isto devido ao facto de a Carminho (FP3) ter um gosto por carne e desgostar de peixe, e a Carla (FP1) ter um gosto por aves e peixe e desgostar carnes vermelhas. Isto leva a mais trocas entre eles por terem tendência a pedir ingredientes que o outro tende a não usar.
- O raciocínio acima descrito pode também ser aplicado aos utilizador Carla (FP1) e Nelson (FP2), no entanto, o valor de F1<>F2 é de 24 contra 30 de F1<>F3. Isto é provavelmente explicado devido ao facto de tanto a Carla como o Nelson apenas terem nas suas preferências ingredientes que não podem ser pedidos na totalidade, o que leva a um menor número de pedidos total. Este não é o caso da Carminho (FP3), que tem nas suas preferências ingredientes como Arroz, que podem ser pedido na totalidade.
- O utilizador Marcelo (FP4) possui o menor número de pedidos totais devido ao facto de ter preferências vegetarianas, que não é o caso de nenhum dos outros utilizadores, ou seja, estes outros utilizadores acabam por não comprar ingredientes que o Marcelo goste, e quando o Marcelo os requisita, ninguém os possui.

A segunda *run,* vai ser feita com todos os parâmetros iguais excepto os pesos, que passam a ser: wed=1/100, wci=1/100, wst=98/100. Obtemos então a Tabela 4 (valores arredondados):

	Feedback	Trash Value	Deviation CI
F1	3.5	36	71
F2	3.07	25	120
F3	3.52	25	158
F4	4.28	38	96
Mean	3.59	32	111

Tabela 4 - valores finais das métricas segunda *run*

Através da análise da Tabela 4 é possível concluir que:

- Ao alterar os pesos no sentido de dar mais valor às receitas que o utilizador irá gostar conseguimos um aumento da média de feedback de 0.4. Este valor é ainda maior quando comparado com *runs* de wst=1/100 e 99/100.
- A diminuição do peso atribuído a um *calorie intake* adequado leva também a que a métrica associada tenha um aumento de 57% (o que representa um resultado mais negativo).
- A mesma observação aplica-se à métrica associada ao desperdício alimentar, com um desperdício médio de mais 4kg. Supomos que a mudança dos pesos tenha um efeito reduzido nesta métrica pois a grande parte da redução de desperdício passa pelo facto dos pedidos feitos a vizinhos, serem sempre feitos ao vizinho com data de validade mais próxima de terminar, como mencionado na secção 3.

4.3.2 Ambiente 2

A primeira *run* foi executada com os valores *default* como na primeira *run* da secção *4.3.2 Ambiente 2*, nomeadamente: wed=⅓, wci=⅓, wst=⅓. O update dos pesos dos ingredientes é feito com um *learning rate* de 0.7. Um ingrediente é inicializado com peso 2.5 e é adicionado à lista de compras se possuir um peso superior ou igual a 2.6.

Obtemos a Tabela 5:

	FeedBack	Trash Value	Deviation CI
FP 1 Ambiente1	3	31.9	35
FP 1 Ambiente 2	3.44	43	34

Tabela 5 - valores finais das métricas de primeira *run*

Ao comparar os valores da Tabela 5, concluimos que:

- Num ambiente isolado (Ambiente 2), um FP com as mesmas condições vai gerar um Feedback Médio superior (aumento de 0.44). Isto deve-se provavelmente ao facto que certos ingredientes serem mais "concorridos", isto é, gostados por vários Utilizadores, e num ambiente isolado não existe ninguém que os peça, logo podemos usá-los mais vezes.
- O *Trash Value* aumenta substancialmente (11.1kg), tal deve-se à não implementação da nossa funcionalidade de pedidos de ingredientes o que leva a que ingredientes perto da sua data de validade sejam enviados a outros utilizadores que os queiram.
- Como seria de esperar, a métrica correspondente a um *Calorie Intake* adequado não foi afetada pela mudança de ambiente, pois em ambos os casos existem várias receitas que permitam manter esta métrica com valores positivos.

Vale também a pena analisar os pesos que são calculados para cada ingrediente. No final desta *run*, temos um total de 76 ingredientes que tem um peso atribuído. Por questões de simplicidade, vamos apenas ter em conta os ingredientes cujos pesos sejam >= 3 ou <=2.

O utilizador em questão "Carla", tem definido como preferencias o seguinte:

- ingredientes que gosta - Bacalhau, Maruca, Camarão, Frango, Bife de Frango, Lula, Atum, Pato e Pescada.
- ingredientes que não gosta - Costeletas de Porco, Fiambre, Carne de Vaca Picada, Presunto, Salsicha Fresca, Costeleta de Porco, Chouriço e Carne de Porco.

Ao aplicarmos a restrição acima referida, ficamos com os pesos dos seguintes ingredientes: ['Atum', 'Alho', 'Batata', 'Natas', 'Cebola', 'Bacalhau', 'Arroz', 'Tomate', 'Bife de Frango', 'Cogumelos', 'Grão', 'Pescada', 'Folha de Louro', 'Carne de Porco', 'Ovo', 'Batata-palha', 'Brócolos', 'Pimento', 'Leite', 'Manteiga', 'Limão', 'Massa de Pimentão', 'Ameijoa', 'Azeitona', 'Azeite', 'Salsa', 'Coentros', 'Vinho Branco', 'Noz-Moscada', 'Vinagre', 'Tomilho', 'Molho Inglês', 'Alecrim']

Os pesos correspondentes são os mostrados no Gráfico 1, com o valor do eixo do X a identificar o ingrediente pelo seu index na lista anterior. A linha a tracejado representa o valor inicial de cada weight, 2.5.

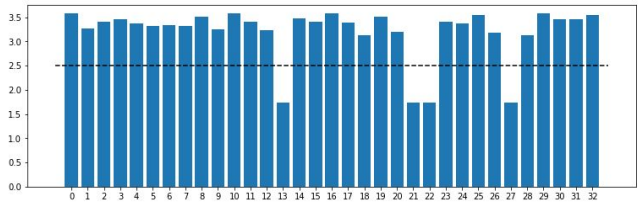

Gráfico 1 - pesos dos ingredientes para de um agente

Ao comparar os valores dos pesos dos ingredientes apresentados com as preferências do utilizador, reparamos que efectivamente existe uma ligação entre eles. Ingredientes que o utilizador Carla gosta, como por exemplo, Atum, Bacalhau e Bife de Frango, apresentam respectivamente valores de 3.578, 3.324 e 3.505. Simultaneamente, ingredientes que o utilizador não gosta, como Carne de Porco, têm valores baixos, neste caso, 1.73.

Os outros ingredientes que o utilizador não gosta não estão representados neste grupo pois encontram-se no intervalo 2 < valor < 3 que foi retirado antes da criação do gráfico. Como são ingredientes não desejados, da primeira vez que são consumidos os seus pesos diminuem, o que leva a que não sejam escolhidos mais vezes e por isso o seu peso estagna num valor inferior a 2.5. Esta é também a razão para haver mais ingredientes com pesos "positivos" (superiores a 2.5), pois estes são os ingredientes que mais updates recebem por serem escolhidos mais vezes.

Aplicando um intervalo mais restrito e mostrando apenas os ingredientes com valores >=3.5 ou <=2, obtemos o Gráfico 2:

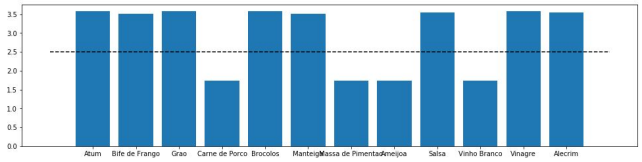

Gráfico 2 - pesos dos ingredientes no intervalo y >= 3.5 ou y <=2

De notar mais uma vez a presença de ingredientes que o utilizador gosta, como Atum e Bife de Frango. A presença de ingredientes

que são "neutros" para o utilizador deve-se à presença destes ingredientes em receitas onde aparecem também ingredientes "bons". O Vinagre por exemplo, das 6 receitas em que está presente, 1 possui também Frango, 3 possuem Atum, 1 possui Pato, e apenas 1 possui um ingrediente "mau", Costeletas de Porco (as receitas são, respectivamente: Arroz de Cabidela, Salada de Feijão-Frade com Atum, Salada Morna de Grão com Atum, Beringelas com Atum, Pato Recheado e Costeletas de Porco com Lentilhas). O nosso agente aprendeu, como desejado, a associar certos ingredientes a outros ingredientes "bons", o que nos permite fazer uma previsão de receitas que o utilizador irá gostar.

Este mesmo raciocínio é usado para explicar o porquê de ingredientes como Vinho Branco terem valores tão "negativos".

Ao executarmos uma *run* de 40 iterações com os mesmos parâmetros obtemos o Gráfico 3, onde é apresentada a evolução do peso associado ao ingrediente Atum (eixo de X representa dias e não refeições):

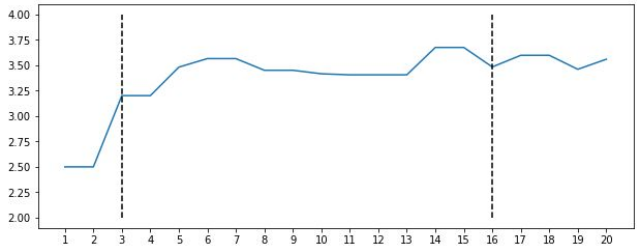

Gráfico 3 - variação do peso do Atum

Como já seria de esperar, o ingrediente, sendo preferido pelo utilizador em questão, vai tender para um valor superior ao longo do tempo. Ao analisar com mais detalhe certos pontos de mudança, observamos que as 2 variações realçadas pelas linhas verticais correspondem respectivamente a:

1. O utilizador escolhe a sugestão "Salada Morna de Grão com Atum" e dá um feedback de 3.5, o que aumenta o peso do ingrediente para 3.2 = 2.5 * 0.3 + 3.5 * 0.7.
2. O utilizador escolhe a sugestão "Salada de Feijão-Frade com Atum", e como Atum é o único ingrediente de que gosta na receita, dá um feedback de 3.4. O peso era nesta altura de 3.564 e vai descer para 3.449 = 3.564*0.3 + 3.4*0.7.

5. CONCLUSÃO

Podemos com toda a certeza dizer que este projeto foi, para todos nós, um dos mais satisfatórios que já fizemos ao longo do nosso processo académico no IST. Para além de podermos pôr em prática ensinamentos concretos na área de Agentes Autónomos exemplificando com propriedades reais, tivemos ainda a hipótese de desenvolver uma ideia por nós concebida sobre um tema que nos diz muito. Ideia essa que é ainda mais relevante no presente e que nos permite, de certa forma, alterar a vida das pessoas de uma maneira positiva.

No que à implementação diz respeito, podemos afirmar que ficamos bastante satisfeitos com o resultado final. Ao longo do tempo de desenvolvimento foram sempre surgindo novas ideias e detalhes de como encaixariam as diversas funcionalidades do agente no nosso dia-a-dia. Concretamente, em relação aos pesos de cada métrica na sugestão de uma receita, gostaríamos de ter dotado o nosso agente com a capacidade de aprender quais os pesos que oferecessem um melhor equilíbrio ou, quem sabe, satisfizesse os desejos do utilizador (quer seja evitar o desperdício, uma dieta calórica mais rígida ou simplesmente comer aquilo que mais gosta). Para além disso, também ponderamos desenvolver uma interface gráfica, ainda que rudimentar, de modo a simular a interação com o nosso *Fridge&Pantry* de uma maneira realista. No entanto, e talvez por querermos aperfeiçoar o nosso agente ao ponto de satisfazer todas as nossas ideias iniciais, o objetivo de desenvolver uma interface gráfica foi deixado para segundo plano e substituído pela criação de comandos como forma de dar *input*.

Numa análise mais pessoal, e face a uma nova realidade mais remota, este projeto trouxe-nos algumas dificuldades no que à comunicação intragrupo diz respeito, no entanto, e graças a uma enorme boa vontade por parte de todos os membros do grupo, todos os problemas foram ultrapassados. Esta experiência deixou-nos mais próximos enquanto amigos e mais fortes enquanto profissionais.