



**FEUP** FACULDADE DE ENGENHARIA  
UNIVERSIDADE DO PORTO

3ºano – MIEIC – 20 de Maio 2018

# Otimização na Organização de um Jantar Relatório

Inteligência Artificial  
Grupo B2\_3

Bruno Alexandre Oliveira Dias, up201504859@fe.up.pt  
Fernando André Bezerra Moura Fernandes, up201505821@fe.up.pt  
Maria Eduarda Santos Cunha, up201506524@fe.up.pt



# Índice

<b>1. OBJETIVO .....</b>	<b>3</b>
<b>2. ESPECIFICAÇÃO .....</b>	<b>3</b>
2.1. ABORDAGEM .....	3
2.1.1. REPRESENTAÇÃO DO PROBLEMA .....	3
2.1.2. ALGORITMO GENÉTICO .....	6
2.1.3. SIMULATED ANNEALING.....	7
2.2. INPUT DO PROGRAMA.....	8
<b>3. DESENVOLVIMENTO.....</b>	<b>9</b>
3.1. FERRAMENTAS/APIs UTILIZADAS.....	9
3.2. ESTRUTURA DA APLICAÇÃO .....	9
<b>4. EXPERIÊNCIAS .....</b>	<b>10</b>
<b>5. CONCLUSÕES .....</b>	<b>12</b>
<b>6. MELHORAMENTOS.....</b>	<b>13</b>
<b>7. RECURSOS.....</b>	<b>14</b>
<b>8. APÊNDICE.....</b>	<b>15</b>
8.1. MANUAL DO UTILIZADOR .....	15
8.2. FICHEIROS DE INPUT .....	15
8.2.1. EXPERIÊNCIA 1 .....	15
8.2.2. EXPERIÊNCIA 2 .....	16
8.2.3. EXPERIÊNCIA 3 .....	16
8.2.4. EXPERIÊNCIA 4 .....	17
8.2.5. EXPERIÊNCIA 5 .....	17
8.2.6. EXPERIÊNCIA 6 .....	18



# 1. Objetivo

Este trabalho tem como objetivo resolver o problema de otimização da distribuição de pessoas por mesas num jantar solidário, onde várias pessoas se registaram. Idealmente, os grupos de pessoas que se inscreveram juntas não deverão ser separados e é importante a afinidade entre pessoas de grupos diferentes quando juntas na mesma mesa. Essa afinidade é determinada segundo interesses ou características em comum: idade, trabalho, hobby, interesse e família.

Pretende-se estabelecer a respetiva distribuição das pessoas pelos lugares.

## 2. Especificação

O problema em questão inclui-se nos problemas de otimização dado que se pretende maximizar a afinidade existente entre pessoas no conjunto de todas as mesas.

### 2.1. Abordagem

#### 2.1.1. Representação do problema

De forma a representar o problema proposto, procedemos à criação de 3 classes *Person*, *Table* e *Dinner*, representativas de cada indivíduo, mesa e cenário de jantar possível respetivamente.

Atribuímos às classes os seguintes atributos:

- *Person*: *id*, *name*, *age*, *group* (id do grupo de pessoas com que se inscreveu), *hobie*, *family*, *job* e *interest*;
- *Table*: *id*, *size* e *people* (lista de pessoas a sentar nessa mesa);



- *Dinner: id, tables, fitness, affinity, probabilityMin, probabilityMax e probability.*

Uma vez que a disposição das pessoas pelas mesas tem em conta a afinidade entre elas, procedemos à criação da função *getAfinity(self, person)* na classe *Person* responsável por devolver essa mesma afinidade. Calcula-a fazendo várias comparações entre 2 pessoas e incrementado o atributo *afinity* segundo o seguinte sistema de prioridades definido por nós:

Se as 2 pessoas a ser comparadas...

- Estiverem no mesmo grupo, incrementa 1 unidade;
- Tiverem o mesmo interesse, incrementa 0.5 unidades;
- Tiverem a mesma família, incrementa 0.5 unidades;
- Tiverem a mesma idade, incrementa 0.5 unidades;
- Tiverem o mesmo emprego, incrementa 0.5 unidades;
- Tiverem o mesmo hobby, incrementa 0.5 unidades;

Isto, pois o mais importante é manter um grupo de pessoas inscritas juntas, já que partiu da sua iniciativa própria juntarem-se e, conseqüentemente, deverão querer manter-se na mesma mesa, em detrimento da companhia de pessoas com quem possam ter outras relações de proximidade. As restantes características têm todas a mesma importância, dado que todas constituem um elemento comum e passível de ser usado como tema de conversa.

Em *Person.py*:

```
def getAfinity(self, person):  
    afinity = 0  
    if(self.id != -1 and person.id != -1):  
        if (person.group == self.group):  
            afinity += 1.0  
        if (person.interest == self.interest):  
            afinity += 0.5  
        if (person.family == self.family):
```



```
        affinity += 0.5
    if (person.age == self.age):
        affinity += 0.5
    if (person.job == self.job):
        affinity += 0.5
    if (person.hobie == self.hobie):
        affinity += 0.5
    return affinity
```

Como a afinidade total do jantar corresponde à soma da afinidade em cada mesa, *getAfinity(self)* na classe *Table* trata de retornar a afinidade total de uma mesa ao somar as afinidades entre todas as pessoas sentadas na mesma e *calcAfinity(self)* na classe *Dinner* soma as afinidades das mesas todas, gerando assim a afinidade total do jantar.

Em *Table.py*:

```
def getAfinity(self):
    affinity = 0
    if (self.size > len(self.people)):
        for x in range(0, len(self.people)):
            if (x < len(self.people) - 1):
                affinity += self.people[x].getAfinity(self.people[x + 1])
    else:
        for x in range(0, self.size):
            if (x < self.size - 1):
                affinity += self.people[x].getAfinity(self.people[x + 1])
    return affinity
```

Em *Dinner.py*:

```
def calcAfinity(self):
    self.affinity = 0
    for x in range(0, len(self.tables)):
        self.affinity += self.tables[x].getAfinity()
```

Ainda, é tratado que não se podem sentar mais pessoas numa mesa do que o número de lugares existentes nela, representado por *size*.



Por fim, tentamos procurar a solução ótima através dos 2 algoritmos Algoritmo Genético e Simulated Annealing, sendo que em ambos a população será constituída por diferentes versões de um jantar, ou seja, diferentes combinações de pessoas nas mesas.

### 2.1.2. Algoritmo Genético

Os Algoritmos Genéticos têm por base os processos genéticos de organismos biológicos. Para um problema existe uma população cujos indivíduos representam as possíveis soluções e a cada indivíduo/solução é atribuída uma pontuação que diz respeito ao quão ideal essa solução é. As soluções com maior pontuação são selecionadas para sofrer o cruzamento, gerando novas soluções e as melhores dessa nova população passam pela mutação. Idealmente, deverá convergir numa solução ideal.

O **cruzamento** entre dois “jantares” é feito trocando entre dois jantares uma pessoa pela mesma no outro jantar, ficando essa pessoa numa posição diferente e, consequentemente, gerando uma nova afinidade.

Em Dinner.py:

```
def mate(self, other):
    firstTableChoiceIndex = choice(range(0,len(self.tables)))
    secondTableChoiceIndex = choice(range(0,len(other.tables)))
    firstPersonChoiceIndex =
choice(range(0,len(self.tables[firstTableChoiceIndex].people)))
    secondPersonChoiceIndex =
choice(range(0,len(other.tables[secondTableChoiceIndex].people)))
    ownSubs = self.tables[firstTableChoiceIndex].people[firstPersonChoiceIndex]
    otherSubs =
other.tables[secondTableChoiceIndex].people[secondPersonChoiceIndex]
    self.subs(firstTableChoiceIndex,firstPersonChoiceIndex,otherSubs)
    other.subs(secondTableChoiceIndex,secondPersonChoiceIndex,ownSubs)
```

Em genetic.py:

```
def mate(selectedPopulation):
    cruzProb = 0.5
    l = len(selectedPopulation)
    selectedForMating = list()
    for x in range(0, l):
```



```
r = random()
if (r < cruzProb):
    selectedForMating.append(selectedPopulation[x])
l2 = len(selectedForMating)
for y in range(0, l2,2):
    if(y == l2-1):
        selectedPopulation[y].mate(selectedPopulation[y])
    else:
        selectedPopulation[y].mate(selectedPopulation[y + 1])
```

A **mutação** consiste em trocar duas pessoas de sítio dentro de cada versão do jantar.

Em `Dinner.py`:

```
def mutate(self):
    firstTableChoiceIndex = choice(range(0, len(self.tables)))
    secondTableChoiceIndex = choice(range(0, len(self.tables)))
    firstPersonChoiceIndex = choice(range(0,
len(self.tables[firstTableChoiceIndex].people)))
    secondPersonChoiceIndex = choice(range(0,
len(self.tables[secondTableChoiceIndex].people)))
    ownSubs = self.tables[firstTableChoiceIndex].people[firstPersonChoiceIndex]
    self.subs(secondTableChoiceIndex,secondPersonChoiceIndex, ownSubs)
```

Em `genetic.py`:

```
def mutate(population):
    probMut = 0.2
    for x in range(0, len(population)):
        r = random()
        if(r < probMut):
            population[x].mutate()
```

O processo descrito é repetido ao longo de uma série de iterações, cujo critério de paragem é quando a diferença entre a afinidade máxima anterior e a atual for mínima ou nula.

### 2.1.3. Simulated Annealing

O algoritmo Simulated Annealing ou Arrefecimento Simulado é uma técnica de procura aleatória baseada na forma como um metal arrefece e congela numa estrutura cristalina com energia mínima e direciona-se a problemas não lineares.



Começa-se por estabelecer uma temperatura inicial alta o suficiente para permitir uma mudança para outro estado da vizinhança e uma temperatura final de 0 como critério de paragem. A cada iteração, decrementa-se a temperatura.

```
Em simulated_annealing.py:  
def updateTemperature(temperature):  
    t = temperature - 1  
    return t
```

A partir de cada solução chega-se a uma solução diferente e se esta for melhor, continua, se não, calcula-se a probabilidade de ser usada para a próxima iteração.

```
Em simulated_annealing.py:  
def calcProbability(old_afinity, new_afinity, temperature):  
    diff = new_afinity - old_afinity  
    return exp(-(abs(diff)/temperature)))
```

De seguida, é gerado um número aleatório e se for superior a essa probabilidade, escolhe-se essa solução. Caso contrário, é descartada. A qualidade de uma solução é determinada através da sua energia (semelhante à pontuação no caso do Algoritmo Genético).

```
Em simulated_annealing.py:  
def measureProbability(r):  
    ra = random()  
    return ra > r
```

## 2.2. Input do Programa

A receção da informação necessária ao nosso programa quanto às pessoas inscritas, grupos e respetivas características é feita através de um ficheiro intitulado *input*.

As primeiras duas linhas do ficheiro contêm inteiros que dizem respeito ao número de mesas e ao número de lugares de cada mesa, respetivamente.





As restantes linhas possuem cada uma a informação pessoal de cada pessoa inscrita, organizada da seguinte forma <Nome> : <idade> - <grupo> - <hobby> - <família> - <trabalho> - <interesse>.

## 3. Desenvolvimento

### 3.1. Ferramentas/APIs Utilizadas

Desenvolvemos o projeto no Sistema Operativo Ubuntu 18.04, no IDE PyCharm e a linguagem de programação foi Python.

### 3.2. Estrutura da Aplicação

Módulos:

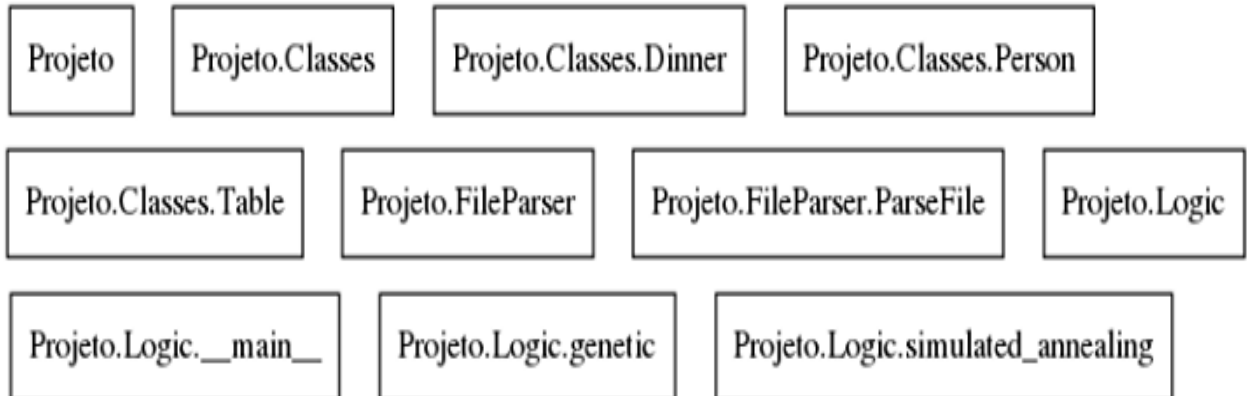


Figure 1 Módulos



### Diagrama de Classes:

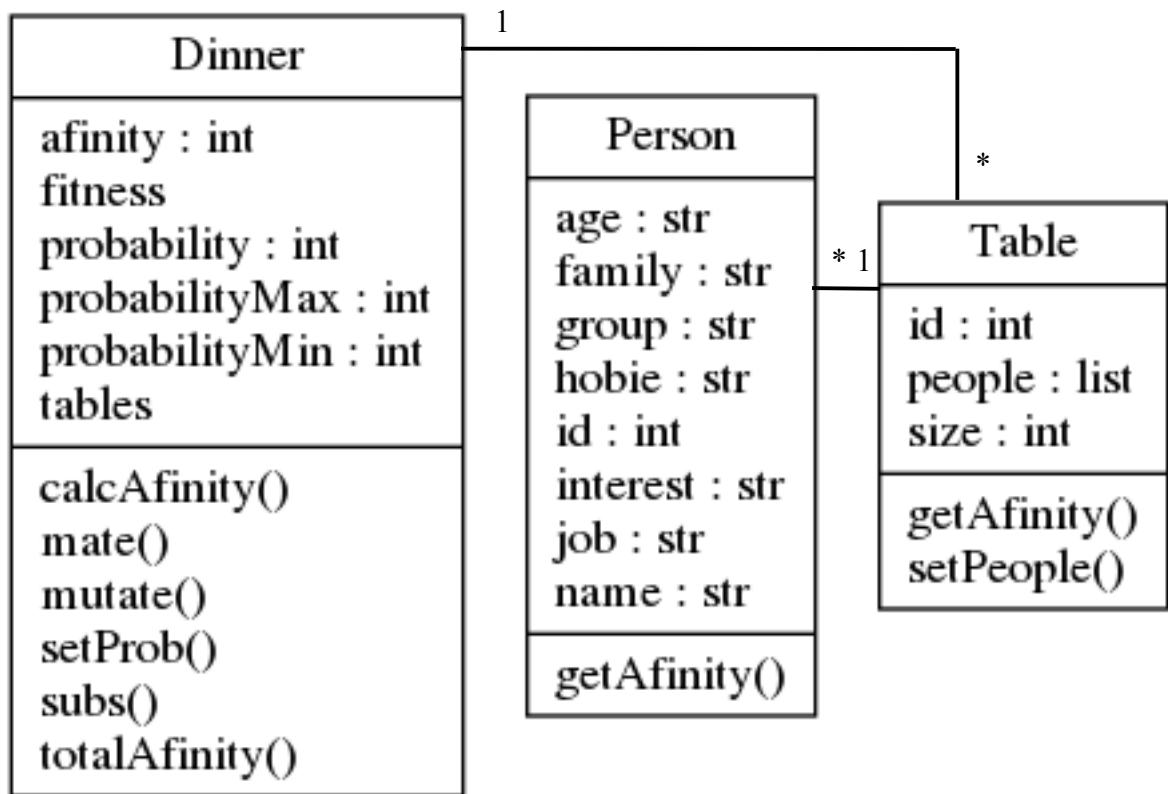


Figure 2 Diagrama de Classes

## 4. Experiências

No quadro seguinte, estão apresentados os diferentes valores de afinidade total do jantar obtidos para cada uma de 6 experiências repetidas 5 vezes segundo os algoritmos Algoritmo Genético e Simulated Annealing.

Algoritmo Genético					
Experiência 1	12 pessoas inscritas em 6 grupos, 3 mesas de 5 lugares cada				
Afinidade Total:	5	3.5	4.5	9	6



Experiência 2	15 pessoas inscritas em 6 grupos, 3 mesas de 5 lugares cada				
Afinidade Total:	9	8	8.5	5.5	9
Experiência 3	12 pessoas inscritas em 5 grupos, 3 mesas de 5 lugares cada				
Afinidade Total:	5.5	7	5	8	6.5
Experiência 4	15 pessoas inscritas em 5 grupos, 3 mesas de 5 lugares cada				
Afinidade Total:	10.5	5.5	11.5	5	7.5
Experiência 5	45 pessoas inscritas em 9 grupos, 9 mesas de 5 lugares cada				
Afinidade Total:	30	28.5	31	28	33.5
Experiência 6	45 pessoas inscritas em 6 grupos, 9 mesas de 5 lugares cada				
Afinidade Total:	26.5	29.5	28.5	28	33.5

Simulated Annealing					
Experiência 1	12 pessoas inscritas em 6 grupos, 3 mesas de 5 lugares cada				
Afinidade Total:	8.5	9.5	10.5	11	9.5
Experiência 2	15 pessoas inscritas em 6 grupos, 3 mesas de 5 lugares cada				
Afinidade Total:	15.5	14	17	13.5	18
Experiência 3	12 pessoas inscritas em 5 grupos, 3 mesas de 5 lugares cada				
Afinidade Total:	11	9.5	11.5	9.5	9
Experiência 4	15 pessoas inscritas em 5 grupos, 3 mesas de 5 lugares cada				
Afinidade Total:	14	13	14.5	15.5	13.5
Experiência 5	45 pessoas inscritas em 9 grupos, 9 mesas de 5 lugares cada				
Afinidade Total:	36,5	38,5	40,5	43	42,5
Experiência 6	45 pessoas inscritas em 6 grupos, 9 mesas de 5 lugares cada				



Afinidade Total:	36	41	35.5	37	41
------------------	----	----	------	----	----

Figure 3 Tabela de Valores da Afinidade Total p/ 6 Experiências

As experiências 1 e 2 mantêm exatamente as mesmas condições à exceção do número de pessoas inscritas, de forma a mostrar a relação da afinidade conforme o número de pessoas.

As experiências 1 e 3, 2 e 4 ou 5 e 6 mantêm exatamente as mesmas condições à exceção do número de grupos em que as pessoas se inscreveram, de forma a mostrar a relação da afinidade conforme o número de grupos.

## 5. Conclusões

Através das experiências 1 e 2, em que todas as condições são mantidas exatamente à exceção do número de pessoas, conseguimos observar que conforme o número de pessoas inscritas aumenta, também a afinidade total do jantar aumenta.

Através dos pares de experiências 1 e 3, 2 e 4 ou 5 e 6, em que todas as condições são mantidas exatamente à exceção do número de grupos, procurávamos identificar o efeito do número de grupos em que as pessoas se inscreveram na solução final. No entanto, os resultados são bastante dispersos, sendo que em algumas situações o aumento do número de grupos não afeta a solução, noutros aumenta a afinidade e noutros até a diminui. Isto verificou-se ao longo de mais testes dos que os apenas representados para efeito deste relatório.

Ainda, concluímos que através do algoritmo Simulated Annealing conseguimos chegar a soluções muito mais otimizadas do que com o Algoritmo Genético, sendo que a diferença das médias da mesma experiência é sempre bastante elevada entre ambos.



## 6. Melhoramentos

Futuramente, a nível da interação com o utilizador poderíamos desenvolver uma interface para que se tornasse mais intuitivo e orgânico para ele correr o programa, mostrar a disposição do jantar final em formato de diagrama e permitir correr ambos os algoritmos devolvendo algumas estatísticas de comparação entre as soluções obtidas e tempo demorado.

Quanto a funcionalidades, seria interessante também receber um ficheiro com mesas de diferentes tamanhos, em vez daquilo que fazemos atualmente, que é assumir que todas têm o mesmo número de lugares.

Em termos de algoritmos, ainda que estejamos contentes por termos tido a oportunidade de explorar o Algoritmo Genético e o Simulated Annealing, pensamos que seria também uma mais valia implementar alguns algoritmos adicionais que não tivessem sido tão abordados nas aulas para podermos expandir o nosso conhecimento.



## 7. Recursos

Biblioteca DEAP, *Distributed Evolutionary Algorithms in Python*,  
<https://github.com/DEAP/deap>

<http://mat.uab.cat/~alseda/MasterOpt/Beasley93GA1.pdf>

<http://www.aiinfinance.com/saweb.pdf>

### Contributo dos elementos do grupo para o projeto final:

Bruno Dias	33,(3)%
Fernando Fernandes	33,(3)%
Maria Eduarda Cunha	33,(3)%



## 8. Apêndice

### 8.1. Manual do Utilizador

Dentro do diretório do projeto, correr no terminal:

```
python -m Logic
```

Ao que é devolvido na consola um menu que permite a escolha entre os algoritmos Algoritmo Genético (1) e Simulated Annealing (2). Escrever 1 ou 2 conforme algoritmo pretendido.

```
1. Genetic Algorithm
2. Simulated Annealing
Option: █
```

Figure 4 Menu Apresentado na Consola

São impressas na consola as composições das várias mesas e a afinidade total do jantar, por exemplo:

```
TableID: 1 Size: 5
PersonID: 1 name: Andre age: 13 group: 1 hobbie: hobie1 family: family1 job: job1 interest: interest2
PersonID: 9 name: Ines2 age: 14 group: 2 hobbie: hobie3 family: family1 job: job4 interest: interest4
PersonID: 2 name: Bruno age: 14 group: 2 hobbie: hobie2 family: family2 job: job3 interest: interest3
PersonID: 8 name: Pedro2 age: 15 group: 4 hobbie: hobie1 family: family4 job: job4 interest: interest1
PersonID: 5 name: Maria age: 16 group: 3 hobbie: hobie4 family: family1 job: job1 interest: interest4
```

Figure 5 Exemplo da Composição da Mesa 1 na Experiência 1, com Algoritmo Genético

```
Afinity: 5.0
```

Figure 6 Exemplo de Afinidade na Experiência 1, com Algoritmo Genético

### 8.2. Ficheiros de Input

#### 8.2.1. Experiência 1

```
3
5
Andre : 13 - 1 - hobie1 - family1 - job1 - interest2
Bruno : 14 - 2 - hobie2 - family2 - job3 - interest3
Pedro : 15 - 1 - hobie3 - family3 - job2 - interest1
Ines : 14 - 5 - hobie1 - family4 - job2 - interest2
```



Maria : 16 - 3 - hobie4 - family1 - job1 - interest4  
 Andre2 : 13 - 1 - hobie3 - family2 - job2 - interest3  
 Bruno2 : 14 - 1 - hobie2 - family3 - job2 - interest1  
 Pedro2 : 15 - 4 - hobie1 - family4 - job4 - interest1  
 Ines2 : 14 - 2 - hobie3 - family1 - job4 - interest4  
 Maria2 : 16 - 3 - hobie4 - family2 - job3 - interest3  
 Ines3 : 14 - 6 - hobie8 - family1 - job4 - interest4  
 Maria3 : 16 - 6 - hobie7 - family2 - job3 - interest3

### 8.2.2. Experiência 2

3  
 5  
 Andre : 13 - 1 - hobie1 - family1 - job1 - interest2  
 Bruno : 14 - 2 - hobie2 - family2 - job3 - interest3  
 Pedro : 15 - 1 - hobie3 - family3 - job2 - interest1  
 Ines : 14 - 5 - hobie1 - family4 - job2 - interest2  
 Maria : 16 - 3 - hobie4 - family1 - job1 - interest4  
 Andre2 : 13 - 1 - hobie3 - family2 - job2 - interest3  
 Bruno2 : 14 - 1 - hobie2 - family3 - job2 - interest1  
 Pedro2 : 15 - 4 - hobie1 - family4 - job4 - interest1  
 Ines2 : 14 - 2 - hobie3 - family1 - job4 - interest4  
 Maria2 : 16 - 3 - hobie4 - family2 - job3 - interest3  
 Ines3 : 14 - 6 - hobie8 - family1 - job4 - interest4  
 Maria3 : 16 - 6 - hobie7 - family2 - job3 - interest3  
 MariaEduarda : 16 - 6 - hobie7 - family2 - job3 - interest3  
 Goncalo : 16 - 6 - hobie7 - family2 - job3 - interest3  
 Joao : 16 - 6 - hobie7 - family2 - job3 - interest3

### 8.2.3. Experiência 3

3  
 5  
 Andre : 13 - 1 - hobie1 - family1 - job1 - interest2  
 Bruno : 14 - 2 - hobie2 - family2 - job3 - interest3  
 Pedro : 15 - 1 - hobie3 - family3 - job2 - interest1  
 Ines : 14 - 5 - hobie1 - family4 - job2 - interest2  
 Maria : 16 - 3 - hobie4 - family1 - job1 - interest4  
 Andre2 : 13 - 1 - hobie3 - family2 - job2 - interest3  
 Bruno2 : 14 - 1 - hobie2 - family3 - job2 - interest1  
 Pedro2 : 15 - 4 - hobie1 - family4 - job4 - interest1  
 Ines2 : 14 - 2 - hobie3 - family1 - job4 - interest4  
 Maria2 : 16 - 3 - hobie4 - family2 - job3 - interest3  
 Ines3 : 14 - 1 - hobie8 - family1 - job4 - interest4  
 Maria3 : 16 - 5 - hobie7 - family2 - job3 - interest3





#### 8.2.4. Experiência 4

3

5

Andre : 13 - 1 - hobie1 - family1 - job1 - interest2

Bruno : 14 - 2 - hobie2 - family2 - job3 - interest3

Pedro : 15 - 1 - hobie3 - family3 - job2 - interest1

Ines : 14 - 5 - hobie1 - family4 - job2 - interest2

Maria : 16 - 3 - hobie4 - family1 - job1 - interest4

Andre2 : 13 - 1 - hobie3 - family2 - job2 - interest3

Bruno2 : 14 - 1 - hobie2 - family3 - job2 - interest1

Pedro2 : 15 - 4 - hobie1 - family4 - job4 - interest1

Ines2 : 14 - 2 - hobie3 - family1 - job4 - interest4

Maria2 : 16 - 3 - hobie4 - family2 - job3 - interest3

Ines3 : 14 - 1 - hobie8 - family1 - job4 - interest4

Maria3 : 16 - 5 - hobie7 - family2 - job3 - interest3

MariaEduarda : 16 - 5 - hobie7 - family2 - job3 - interest3

Joao : 16 - 5 - hobie7 - family2 - job3 - interest3

Goncalo : 16 - 5 - hobie7 - family2 - job3 - interest3

#### 8.2.5. Experiência 5

9

5

Andre : 13 - 1 - hobie1 - family1 - job1 - interest2

Bruno : 14 - 2 - hobie2 - family2 - job3 - interest3

Pedro : 15 - 1 - hobie3 - family3 - job2 - interest1

Ines : 14 - 5 - hobie1 - family4 - job2 - interest2

Maria : 16 - 3 - hobie4 - family1 - job1 - interest4

Andre2 : 13 - 1 - hobie3 - family2 - job2 - interest3

Bruno2 : 14 - 1 - hobie2 - family3 - job2 - interest1

Pedro2 : 15 - 4 - hobie1 - family4 - job4 - interest1

Ines2 : 14 - 2 - hobie3 - family1 - job4 - interest4

Maria2 : 16 - 3 - hobie4 - family2 - job3 - interest3

Ines3 : 14 - 1 - hobie8 - family1 - job4 - interest4

Maria3 : 16 - 5 - hobie7 - family2 - job3 - interest3

MariaEduarda : 16 - 5 - hobie7 - family2 - job3 - interest3

Joao : 16 - 5 - hobie7 - family2 - job3 - interest3

Goncalo : 16 - 5 - hobie7 - family2 - job3 - interest3

Filipe : 13 - 1 - hobie1 - family1 - job1 - interest2

Bruna : 14 - 2 - hobie2 - family2 - job3 - interest3

Paula : 15 - 1 - hobie3 - family3 - job2 - interest1

Isaura : 14 - 5 - hobie1 - family4 - job2 - interest2

Mario : 16 - 3 - hobie4 - family1 - job1 - interest4

Andreia : 13 - 1 - hobie3 - family2 - job2 - interest3



Bernardo : 14 - 1 - hobie2 - family3 - job2 - interest1  
 Patricia : 15 - 4 - hobie1 - family4 - job4 - interest1  
 Ilia : 14 - 2 - hobie3 - family1 - job4 - interest4  
 Joaquina : 16 - 3 - hobie4 - family2 - job3 - interest3  
 Duarte : 14 - 1 - hobie8 - family1 - job4 - interest4  
 Monica : 16 - 5 - hobie7 - family2 - job3 - interest3  
 Eduarda : 16 - 5 - hobie7 - family2 - job3 - interest3  
 Fernando : 16 - 9 - hobie7 - family2 - job3 - interest3  
 Jose : 16 - 5 - hobie7 - family2 - job3 - interest3  
 AndreMarques : 13 - 1 - hobie1 - family1 - job1 - interest2  
 BrunoDias : 14 - 2 - hobie2 - family2 - job3 - interest3  
 PedroCunha : 15 - 9 - hobie3 - family3 - job2 - interest1  
 InesMelao : 14 - 5 - hobie1 - family4 - job2 - interest2  
 MariaMelo : 16 - 3 - hobie4 - family1 - job1 - interest4  
 Diana : 13 - 1 - hobie3 - family2 - job2 - interest3  
 Debora : 14 - 1 - hobie2 - family3 - job2 - interest1  
 Miranda : 15 - 7 - hobie1 - family4 - job4 - interest1  
 Anita : 14 - 2 - hobie3 - family1 - job4 - interest4  
 Beatriz : 16 - 3 - hobie4 - family2 - job3 - interest3  
 Armandina : 14 - 1 - hobie8 - family1 - job4 - interest4  
 Diogo : 16 - 6 - hobie7 - family2 - job3 - interest3  
 Miguel : 16 - 8 - hobie7 - family2 - job3 - interest3  
 Claudio : 16 - 6 - hobie7 - family2 - job3 - interest3  
 Domingos : 16 - 7 - hobie7 - family2 - job3 - interest3

### 8.2.6. Experiência 6

9  
 5  
 Andre : 13 - 1 - hobie1 - family1 - job1 - interest2  
 Bruno : 14 - 2 - hobie2 - family2 - job3 - interest3  
 Pedro : 15 - 1 - hobie3 - family3 - job2 - interest1  
 Ines : 14 - 5 - hobie1 - family4 - job2 - interest2  
 Maria : 16 - 3 - hobie4 - family1 - job1 - interest4  
 Andre2 : 13 - 1 - hobie3 - family2 - job2 - interest3  
 Bruno2 : 14 - 1 - hobie2 - family3 - job2 - interest1  
 Pedro2 : 15 - 4 - hobie1 - family4 - job4 - interest1  
 Ines2 : 14 - 2 - hobie3 - family1 - job4 - interest4  
 Maria2 : 16 - 3 - hobie4 - family2 - job3 - interest3  
 Ines3 : 14 - 1 - hobie8 - family1 - job4 - interest4  
 Maria3 : 16 - 5 - hobie7 - family2 - job3 - interest3  
 MariaEduarda : 16 - 5 - hobie7 - family2 - job3 - interest3  
 Joao : 16 - 5 - hobie7 - family2 - job3 - interest3  
 Goncalo : 16 - 5 - hobie7 - family2 - job3 - interest3



Filipe : 13 - 1 - hobie1 - family1 - job1 - interest2  
Bruna : 14 - 2 - hobie2 - family2 - job3 - interest3  
Paula : 15 - 1 - hobie3 - family3 - job2 - interest1  
Isaura : 14 - 5 - hobie1 - family4 - job2 - interest2  
Mario : 16 - 3 - hobie4 - family1 - job1 - interest4  
Andreia : 13 - 1 - hobie3 - family2 - job2 - interest3  
Bernardo : 14 - 1 - hobie2 - family3 - job2 - interest1  
Patricia : 15 - 4 - hobie1 - family4 - job4 - interest1  
Ilia : 14 - 2 - hobie3 - family1 - job4 - interest4  
Joaninha : 16 - 3 - hobie4 - family2 - job3 - interest3  
Duarte : 14 - 1 - hobie8 - family1 - job4 - interest4  
Monica : 16 - 5 - hobie7 - family2 - job3 - interest3  
Eduarda : 16 - 5 - hobie7 - family2 - job3 - interest3  
Fernando : 16 - 5 - hobie7 - family2 - job3 - interest3  
Jose : 16 - 5 - hobie7 - family2 - job3 - interest3  
AndreMarques : 13 - 1 - hobie1 - family1 - job1 - interest2  
BrunoDias : 14 - 2 - hobie2 - family2 - job3 - interest3  
PedroCunha : 15 - 1 - hobie3 - family3 - job2 - interest1  
InesMelao : 14 - 5 - hobie1 - family4 - job2 - interest2  
MariaMelo : 16 - 3 - hobie4 - family1 - job1 - interest4  
Diana : 13 - 1 - hobie3 - family2 - job2 - interest3  
Debora : 14 - 1 - hobie2 - family3 - job2 - interest1  
Miranda : 15 - 4 - hobie1 - family4 - job4 - interest1  
Anita : 14 - 2 - hobie3 - family1 - job4 - interest4  
Beatriz : 16 - 3 - hobie4 - family2 - job3 - interest3  
Armandina : 14 - 1 - hobie8 - family1 - job4 - interest4  
Diogo : 16 - 6 - hobie7 - family2 - job3 - interest3  
Miguel : 16 - 6 - hobie7 - family2 - job3 - interest3  
Claudio : 16 - 6 - hobie7 - family2 - job3 - interest3  
Domingos : 16 - 6 - hobie7 - family2 - job3 - interest3