

Relatório do Trabalho Prático de AED

Grupo: João Victor Vasconcellos de Santana, Lucas Silva Tanure e Yan Silva Braga

Professora: Ana Paula

1- Desenvolvimento

Início:

```
Run | Debug
public static void main(String[] args) throws IOException {
    //dá início ao processo de vestibular
    Vestibular UniversidadeStark = new Vestibular();
    System.out.println(x:"Vestibular iniciado");
    System.out.println(x:"Iniciando leitura dos arquivos...");
    UniversidadeStark.lerArquivo();
    System.out.println(x:"Leitura dos arquivos concluída.");
    //após a leitura, os dados da entrada serão salvos nas devidas estruturas ou classe
```

O código inicia através do main, que está na classe “Teste”. Lá, um objeto do tipo Vestibular chamado UniversidadeStark é instanciado, este que será a principal ferramenta do programa.

Ao na linha “UniversidadeStark.lerArquivo();”, o método para ler o arquivo de entrada é acionado.

```
public void lerArquivo() {  
    try {  
        // Leitura do arquivo  
        File entrada = new File("pathname:entrada.txt");  
        Scanner scannerEntrada = new Scanner(entrada);  
  
        // Criação de um vetor "valores" que armazena a quantidade de cursos e de  
        // candidatos  
        String[] valores = scannerEntrada.nextLine().split(" ");  
  
        // qtdCursos vai armazenar a quantidade de cursos  
        this.qtdCursos = Integer.parseInt(valores[0]);  
        System.out.println("Quantidade de cursos: " + qtdCursos);  
  
        // qtdCandidatos armazena o número de candidatos  
        this.qtdCandidatos = Integer.parseInt(valores[1]);  
        System.out.println("Quantidade de candidatos: " + qtdCandidatos);  
  
        // Com a quantidade de cursos analisada e salva, agora devemos analisar o nome  
        // dos cursos, qual o código deles e quantas vagas estão disponíveis  
    }  
}
```

Após a leitura da primeira linha, a quantidade de cursos e a quantidade de candidatos são armazenadas nas variáveis respectivas. Como ambas são atributos da classe, utilizando o “this.” o método da leitura do arquivo faz uma função similar ao método construtor.

```

// itera sobre os cursos
for (int i = 0; i < qtdCursos; i++) {

    // valores recebe o conteúdo da próxima linha do arquivo
    valores = scannerEntrada.nextLine().split(regex:"");

    // armazena o código do curso na variável (na posição 0)
    int codigo = Integer.parseInt(valores[0]);
    System.out.println("Codigo: " + codigo);

    // o nome do curso (na posição 1)
    String nome = valores[1];
    System.out.println("Curso: " + nome);

    // e a quantidade de vagas para esse curso (na posição 2)
    int vagas = Integer.parseInt(valores[2]);
    System.out.println("Vagas: " + vagas);

    // após serem salvas, as informações sobre codigo, nome e vagas serão salvas
    // em um objeto tipo Curso
    Curso curso = new Curso(codigo, nome, vagas);

    // esse objeto é adicionado a lista de cursos (classe ListaCursos)
    cursos.inserirFim(curso);

    System.out.println(x:"-----");
}

// Depois do for, todos os cursos foram salvos na lista de cursos.
System.out.println("Cursos disponíveis: " + cursos.mostrar());

```

Nessa parte o for itera sobre todos os cursos, salvando as informações (como Código do curso, nome e vagas) em variáveis de mesmo nome. Por fim, um objeto tipo Curso é instanciado e enviado a uma lista de cursos, pelo método da classe "ListaCursos".

```

for (int i = 0; i < qtdCandidatos; i++) {

    // valores recebe a linha do arquivo contando um candidato
    valores = scannerEntrada.nextLine().split(regex:"");

    // armazena o nome do candidato (na posição 0)
    String nome = valores[0];
    System.out.println("Nome do aluno: " + nome);

    // As notas desse candidato (nas posições 1, 2 e 3)
    double notaRed = Double.parseDouble(valores[1]);
    double notaMat = Double.parseDouble(valores[2]);
    double notaLing = Double.parseDouble(valores[3]);
    System.out.println("Nota em Redação: " + notaRed);
    System.out.println("Nota em Matemática: " + notaMat);
    System.out.println("Nota em Linguagens: " + notaLing);

    // os cursos selecionados pelo aluno (nas posições 4 e 5)
    int codCursoOp1 = Integer.parseInt(valores[4]);
    int codCursoOp2 = Integer.parseInt(valores[5]);
    System.out.println("Curso 1 selecionado: " + codCursoOp1);
    System.out.println("Curso 2 selecionado: " + codCursoOp2);

    // Um objeto do tipo candidato é criado
    // As informações de nome, notas e o códigos são salvos
    Candidato candidato = new Candidato(nome, notaRed, notaMat, notaLing, codCursoOp1, codCursoOp2);
    // Candidato adicionado ao vetor de candidatos
    this.candidatos[i] = candidato;
}

```

Ao fim do for, outro com o mesmo propósito se inicia, mas dessa vez são dados dos candidatos a serem salvos. Um objeto tipo Candidato é instanciado, levando todas as características do candidato atual. Depois disso, esse objeto é armazenado em um vetor de tipo Candidato.

```

//hora de ordenar os candidatos, de acordo com a média.

//aciona o método de ordenação, que coloca as maiores médias no fim do vetor candidatos
System.out.println(x:"Iniciando ordenação dos candidatos...");
UniversidadeStark.ordenar();
System.out.println(x:"Candidatos ordenados.");

```

De volta ao main, inicia a ordenação dos candidatos, através do objeto Vestibular "UniversidadeStark", invocando o método "ordenar()";

```

public static void ordenar() {
    // Após ser acionado pelo main, os candidatos (salvos no vetor) e quantidade de
    // candidatos são objetificados
    Teste ordenar = new Teste(candidatos, qtdCandidatos);
    // método da classe teste.
    ordenar.ordenarCandidatos();
    // candidatos ordenados em ordem crescente de média

    // Como os candidatos estão agora ordenados, analisar o critério de desempate
    // que é a nota de redação, para que isso também fique ordenado.

    // itera sobre o vetor candidatos, começando da maior média (fim do vetor)
    for (int i = qtdCandidatos - 1; i != 0; i--) {
        // se a média do candidato for igual ao do que está atrás dele na fila
        if (candidatos[i].getMedia() == candidatos[i - 1].getMedia())
            // e a nota de Redação (do anterior) for maior
            && candidatos[i].getNotaRed() < candidatos[i - 1].getNotaRed()) {
                // são trocados de posição
                Candidato temp = candidatos[i - 1];
                candidatos[i - 1] = candidatos[i];
                candidatos[i] = temp;
                temp = null;
            }
    }
    // vetor candidatos ordenado em ordem crescente, tratando médias iguais com a
    // maior nota de Red.
}

```

Nesse método, será instanciado um objeto do tipo Teste, chamado ordenar, que contém o vetor de candidatos e a quantidade de candidatos (definidos através do arquivo). A chamada "ordenar.ordenarCandidatos();" faz com que o método de ordenação QuickSort seja ativado.

Também nele será feito o tratamento para o caso de alunos com a mesma média, utilizando como critério de desempate a nota de redação diferente (a maior nota ficará mais ao fim do vetor).

```

private void quicksortNota(int esq, int dir) {
    int i = esq, j = dir;
    double pivo = alunos[(dir + esq) / 2].getMedia();
    Candidato temp;
    while (i <= j) {
        while (alunos[i].getMedia() < pivo)
            i++;
        while (alunos[j].getMedia() > pivo)
            j--;
        if (i <= j) {
            temp = alunos[i];
            alunos[i] = alunos[j];
            alunos[j] = temp;
            i++;
            j--;
        }
    }
    if (esq < j)
        quicksortNota(esq, j);
    if (i < dir)
        quicksortNota(i, dir);
}

```

```

public void ordenarCandidatos() {
    //acionado em Vestibular (1.26)
    //usa o quickSort para ordenar em ordem crescente de média.
    /*As médias são calculadas na própria classe do candidato */

    quicksortNota(esq:0, qtdAlunos - 1);
}

```

Nosso grupo escolheu o QuickSort por se tratar de um método de eficiência média, com complexidade média $O(n \log n)$, ou seja, o tempo de execução do algoritmo aumenta de forma razoável quando o número de dados aumenta. Já que o máximo de candidatos no processo é 1000 (um número consideravelmente baixo) faz com que o QuickSort atenda muito bem no processo de ordenação, além de ser um algoritmo simples de ser implementado.

```

System.out.println(x: "Candidatos Ordenados. ");

//com candidatos ordenados, hora de classifica-los nos cursos desejados
System.out.println(x: "Iniciando cálculo da Classificação...");
UniversidadeStark.calcularClassificacao();
System.out.println(x: "Classificação concluída.");

```

Agora que os candidatos estão organizados em ordem crescente de média, podemos classificá-los.

```

public void calcularClassificacao()
{
    //Com os candidatos ordenados em média crescente, começamos pela maior média
    for(int i=qtdCandidatos-1;i>=0;i--)
    {
        System.out.println(x:"<<<<<CANDIDATO>>>>>>>");
        Candidato candidatoAtual=candidatos[i];
        System.out.println(candidatoAtual.getNome());

        int op1=candidatoAtual.getCodCursoOp1();
        int op2=candidatoAtual.getCodCursoOp2();
        Curso curso1=cursos.pesquisar(op1);
        System.out.println(curso1.getNome());
        Curso curso2=cursos.pesquisar(op2);
        System.out.println(curso2.getNome());

        //1º Op tem vaga

```

Neste método, vamos iterar sobre todos os candidatos, que foram organizados pelo método anterior. Começamos pelo candidato com maior média, para que o algoritmo funcione apropriadamente (maior média significa preferência na escolha do curso 1). Nas variáveis, analisamos quais as opções 1 e 2, e quais os cursos referentes a elas, instanciados e achados na lista de cursos pelo método “.pesquisar”.

```

//1º Op tem vaga
if(curso1.getSelecionados().size()<curso1.getVagas())
{
    curso1.inserirSelecionados(candidatoAtual);
    System.out.println("Candidato "+candidatoAtual.getNome()+" inserido em "+curso1.getNome());
}
//1º Op NÃO tem vaga
else if(curso1.getSelecionados().size()>=curso1.getVagas())
{
    //2º Op tem vaga
    if(curso2.getSelecionados().size()<curso2.getVagas())
    {
        curso2.inserirSelecionados(candidatoAtual);
        System.out.println("Candidato "+candidatoAtual.getNome()+" inserido em "+curso2.getNome());

        System.out.println(x:"na segunda opção, então...");
        curso1.inserirFilaEspera(candidatoAtual);
        System.out.println("Candidato "+candidatoAtual.getNome()+" inserido na fila de espera de "
        +curso1.getNome());
    }

    //2º Op NÃO tem vaga
    else if(curso2.getSelecionados().size()>=curso2.getVagas())
    {
        curso1.inserirFilaEspera(candidatoAtual);
        curso2.inserirFilaEspera(candidatoAtual);
        System.out.println("Candidato "+candidatoAtual.getNome()+" inserido na fila de espera de "
        +curso1.getNome()+" e de "+curso2.getNome()+" por não ter vagas");
    }
}

```

Agora, as condicionais apontarão:

- Se a opção 1 tem vagas disponíveis, se tiver, ele é inserido na lista de selecionados do curso
 - Se a opção 1 não tiver vagas, entra em outras condicionais:
- Se a opção 2 tem vagas, se tiver, é selecionado para o curso, ficando na fila de espera da primeira opção.
- Se a opção 2 não tem vagas, será inserido na fila de espera de ambas as opções.

```
//Com os candidatos em seus devidos cursos, calculamos a nota de corte
for(int i=1;i<=qtdCursos;i++)
{
    Curso cursoAtual=cursos.pesquisar(i);
    ArrayList<Candidato> selecionados=cursoAtual.getSelecionados();
    double menorMedia=9999999999;

    //iteramos a fila de selecionados do curso
    for(Candidato candidato : selecionados)
    {
        double mediaCandidato=candidato.getMedia();
        //se a média do candidato for a menor entre as médias, salva na variável
        if(mediaCandidato<menorMedia)
        {
            if(mediaCandidato<menorMedia)
            {
                menorMedia=mediaCandidato;
            }
        }
    }

    //a menor média salva é inserida na nota de corte do curso
    cursoAtual.setNotaCorte(menorMedia);
}
```

Feita a seleção, vamos definir a nota de corte dos cursos, iterando sobre todos, começando com o de menor código.

Um objeto Curso é instanciado, definido pelo método de pesquisa na lista de cursos (utilizando o código atual do for). Uma ArrayList do tipo Candidato é instanciada, definida pela lista de selecionados do curso atual. Uma variável de nome menorMedia é declarada, a ela atribuído um valor altíssimo, para que sempre ocorra de a média do primeiro candidato analisado ser menor.

Com isso, iteramos a lista de selecionados do curso atual, no for each, instanciando um objeto Candidato que recebe as informações de cada candidato presente na fila (a cada iteração). Salvamos a média desse candidato numa variável para comparar se essa é menor que a então menor média, se sim, a menor média passa a ser igual

a média do candidato em questão. Ao fim do for, utilizamos o método `setNotaCorte`, enviando a menor média encontrada.

```
System.out.println(x:"Classificação concluída.");

System.out.println(x:"Iniciando Escrita no arquivo de saída...");
UniversidadeStark.escreverSaida();
System.out.println(x:"Escrita finalizada. Cheque o arquivo de saída.");
System.out.println(x:"Vestibular finalizado!");
```

Com os candidatos devidamente classificados, voltamos ao main para acionar o método que escreve no arquivo de saída.

```
public void escreverSaida() throws IOException {

    FileWriter escreverSaida = new FileWriter(fileName:"saida.txt");

    int i = 1;

    while (cursos.pesquisar(i) != null) {
        Curso curso = cursos.pesquisar(i);
        // Escreve o nome do curso da iteração e a nota de corte dele
        escreverSaida.write(curso.getNome() + " " + String.format(format:"%.2f", curso.getNotaCorte()) + "\n");
        escreverSaida.write(str:"Selecionados\n");
        // escreve o nome dos selecionados daquele curso
        for (Candidato candidato : curso.getSelecionados()) {
            escreverSaida.write(candidato.getNome() + " " + String.format(format:"%.2f", candidato.getMedia()) + "\n");
        }
        // escreve o nome dos que ficaram na fila de espera
        escreverSaida.write(str:"Fila de Espera\n");
        for (Candidato candidato : curso.getFilaEspera()) {
            escreverSaida.write(candidato.getNome() + " " + String.format(format:"%.2f", candidato.getMedia()) + "\n");
        }

        escreverSaida.write(str:"\n");
        i++;
    }

    escreverSaida.close();
}
```

Usando a lógica similar de iteração de cursos, começamos escrevendo o curso de menor código, até o maior, na ordem:

- Nome do Curso, Nota de Corte

- Seleccionados (for each itera a lista de seleccionados, imprimindo o nome do candidato e sua média, que estão em ordem decrescente)
- Fila de Espera (for each itera a fila, imprimindo o nome do candidato e sua média, que estão em ordem decrescente)

Ao fim desse método, o arquivo de saída terá todas as informações relevantes sobre os cursos, os seleccionados, os na fila de espera, as médias e notas de corte.

2- Testes

Entrada Padrão:

```
1 4;8
2 1;Matemática;2
3 2;Física;2
4 3;Química;2
5 4;Estatística;5
6 Bob Esponja;600;700;800;1;2
7 Pato Donald;700;700;800;1;2
8 Mickey Mouse;800;700;800;1;2
9 Peppa Pig;700;700;500;3;2
10 Super Mario Bros;600;600;500;4;3
11 Peter Parker;700;850;800;3;4
12 Deadpool;900;700;800;3;4
13 Tio Patinhas;700;700;600;4;2
```

Terminal Padrão:

```
Vestibular iniciado
Iniciando leitura dos arquivos...
Quantidade de cursos: 4
Quantidade de candidatos: 8
Codigo: 1
Curso: Matemática
Vagas: 2
-----
Codigo: 2
Curso: Física
Vagas: 2
-----
Codigo: 3
Curso: Química
Vagas: 2
-----
Codigo: 4
Curso: Estatística
Vagas: 5
-----
Cursos disponíveis: Matemática Física Química Estatística
-----
Nome do aluno: Bob Esponja
Nota em Redação: 600.0
Nota em Matemática: 700.0
Nota em Linguagens: 800.0
Curso 1 selecionado: 1
Curso 2 selecionado: 2
-----
Nome do aluno: Pato Donald
Nota em Redação: 700.0
Nota em Matemática: 700.0
Nota em Linguagens: 800.0
Curso 1 selecionado: 1
Curso 2 selecionado: 2
-----
```

Nome do aluno: Mickey Mouse
Nota em Redação: 800.0
Nota em Matemática: 700.0
Nota em Linguagens: 800.0
Curso 1 selecionado: 1
Curso 2 selecionado: 2

Nome do aluno: Peppa Pig
Nota em Redação: 700.0
Nota em Matemática: 700.0
Nota em Linguagens: 500.0
Curso 1 selecionado: 3
Curso 2 selecionado: 2

Nome do aluno: Super Mario Bros
Nota em Redação: 600.0
Nota em Matemática: 600.0
Nota em Linguagens: 500.0
Curso 1 selecionado: 4
Curso 2 selecionado: 3

Nome do aluno: Peter Parker
Nota em Redação: 700.0
Nota em Matemática: 850.0
Nota em Linguagens: 800.0
Curso 1 selecionado: 3
Curso 2 selecionado: 4

Nome do aluno: Deadpool
Nota em Redação: 900.0
Nota em Matemática: 700.0
Nota em Linguagens: 800.0
Curso 1 selecionado: 3
Curso 2 selecionado: 4

Nome do aluno: Tio Patinhas
Nota em Redação: 700.0
Nota em Matemática: 700.0
Nota em Linguagens: 600.0
Curso 1 selecionado: 4
Curso 2 selecionado: 2

Leitura dos arquivos concluída.
Iniciando ordenação dos candidatos...
Deadpool - Peter Parker - Mickey Mouse - Pato Donald - Bob Esponja - Tio Patinhas - Peppa Pig - Candidatos ordenados.
Iniciando cálculo da Classificação...

<<<<<<CANDIDATO>>>>>>

Deadpool

Química

Estatística

Candidato Deadpool inserido em Química

<<<<<<CANDIDATO>>>>>>

Peter Parker

Química

Estatística

Candidato Peter Parker inserido em Química

<<<<<<CANDIDATO>>>>>>

Mickey Mouse

Matemática

Física

Candidato Mickey Mouse inserido em Matemática

<<<<<<CANDIDATO>>>>>>

Pato Donald

Matemática

Física

Candidato Pato Donald inserido em Matemática

```

<<<<<<CANDIDATO>>>>>>>
Bob Esponja
Matemática
Física
Candidato Bob Esponja inserido em Física
na segunda opção, então...
Candidato Bob Esponja inserido na fila de espera de Matemática
<<<<<<CANDIDATO>>>>>>>
Tio Patinhas
Estatística
Física
Candidato Tio Patinhas inserido em Estatística
<<<<<<CANDIDATO>>>>>>>
Peppa Pig
Química
Física
Candidato Peppa Pig inserido em Física
na segunda opção, então...
Candidato Peppa Pig inserido na fila de espera de Química
<<<<<<CANDIDATO>>>>>>>
Super Mario Bros
Estatística
Química
Candidato Super Mario Bros inserido em Estatística
Classificação concluída.
Iniciando Escrita no arquivo de saída...
Escrita finalizada. Cheque o arquivo de saída.
Vestibular finalizado!

```

Saída Padrão:

```

1  Matemática 733,33
2  Selecionados
3  Mickey Mouse 766,67
4  Pato Donald 733,33
5  Fila de Espera
6  Bob Esponja 700,00
7
8  Física 633,33
9  Selecionados
10 Bob Esponja 700,00
11 Peppa Pig 633,33
12 Fila de Espera
13
14 Química 783,33
15 Selecionados
16 Deadpool 800,00
17 Peter Parker 783,33
18 Fila de Espera
19 Peppa Pig 633,33
20
21 Estatística 566,67
22 Selecionados
23 Tio Patinhas 666,67
24 Super Mario Bros 566,67
25 Fila de Espera
26

```

Entrada alternativa:

```
1 4;10
2 1;Matemática;2
3 2;Física;2
4 3;Química;2
5 4;Estatística;5
6 Hulk;850;750;900;2;3
7 Mulher Maravilha;900;800;850;1;2
8 Homem de Ferro;950;900;950;4;1
9 Batman;800;850;800;3;2
10 Elsa;750;800;700;2;4
11 Superman;1000;900;950;1;3
12 Thor;900;800;900;3;2
13 Capitã Marvel;950;850;900;4;1
14 Pikachu;700;650;600;2;3
15 Goku;1000;1000;1000;1;4
```

Saída alternativa:

1	Matemática	950,00
2	Selecionados	
3	Goku	1000,00
4	Superman	950,00
5	Fila de Espera	
6	Mulher Maravilha	850,00
7		
8	Física	833,33
9	Selecionados	
10	Mulher Maravilha	850,00
11	Hulk	833,33
12	Fila de Espera	
13	Elsa	750,00
14	Pikachu	650,00
15		
16	Química	816,67
17	Selecionados	
18	Thor	866,67
19	Batman	816,67
20	Fila de Espera	
21	Pikachu	650,00
22		
23	Estatística	750,00
24	Selecionados	
25	Homem de Ferro	933,33
26	Capitã Marvel	900,00
27	Elsa	750,00
28	Fila de Espera	

Entrada Padrão com caso de média igual e nota de Redação diferente:

```

1  4;9 +1
2  1;Matemática;2
3  2;Física;2
4  3;Química;2
5  4;Estatística;5
6  Bob Esponja;600;700;800;1;2
7  Mickey Mouse;800;700;800;1;2
8  Peppa Pig;700;700;500;3;2
9  Super Mario Bros;600;600;500;4;3
10 Peter Parker;700;850;800;3;4
11 Deadpool;900;700;800;3;4
12 Tio Patinhas;700;700;600;4;2
13 Pato Donald;800;700;700;1;2
14 Ganso Donald Universo Alternativo;801;700;699;1;2
15                                     -1

```

O candidato Pato Donald possui a mesma média que o candidato Ganso Donald, porém Ganso Donald tem uma nota de Redação superior.

Terminal com médias iguais e nota de Redação diferente:

```

-----
Nome do aluno: Pato Donald
Nota em Redação: 800.0
Nota em Matemática: 700.0
Nota em Linguagens: 700.0
Curso 1 selecionado: 1
Curso 2 selecionado: 2
-----
Nome do aluno: Ganso Donald Universo Alternativo
Nota em Redação: 801.0
Nota em Matemática: 700.0
Nota em Linguagens: 699.0
Curso 1 selecionado: 1
Curso 2 selecionado: 2
-----
Leitura dos arquivos concluída.
Iniciando ordenação dos candidatos...
Deadpool - Peter Parker - Mickey Mouse - Ganso Donald Universo Alternativo - Pato Donald - Bob Esponja - Tio Patinhas - Peppa Pig - Candidatos ordenados.

```

Com a aplicação do critério de desempate de médias iguais, **Ganso Donald** ficou à frente de **Pato Donald** na ordem de preferência para entrar na Opção 1, logo...


```

<<<<<<CANDIDATO>>>>>>>>
Mickey Mouse
Matemática → 2 Vagas
Física
Candidato Mickey Mouse inserido em Matemática -1
<<<<<<CANDIDATO>>>>>>>>
Ganso Donald Universo Alternativo
Matemática → 1 Vaga
Física
Candidato Ganso Donald Universo Alternativo inserido em Matemática -1
<<<<<<CANDIDATO>>>>>>>>
Pato Donald
Matemática → 0 vagas
Física
Candidato Pato Donald inserido em Física
na segunda opção, então...
Candidato Pato Donald inserido na fila de espera de Matemática

```

Pato Donald foi inserido em sua segunda opção, já que a primeira não tinha vagas.

```

<<<<<<CANDIDATO>>>>>>>>
Peppa Pig
Química
Física
Candidato Peppa Pig inserido na fila de espera de Química e de Física por não ter vagas

```

Com isso, o candidato **Peppa Pig** (que antes foi selecionado em Física, e agora teve sua vaga pega pelo Pato Donald) ficou na fila de espera de ambas as escolhas, devido à falta de vagas nas duas opções.

Saída com médias iguais e nota de Redação diferente:

1 Matemática 733,33
2 Seleccionados
3 Mickey Mouse 766,67
4 Ganso Donald Universo Alternativo 733,33

5 Fila de Espera
6 Pato Donald 733,33
7 Bob Esponja 700,00

8
9 Física 700,00

10 Seleccionados
11 Pato Donald 733,33
12 Bob Esponja 700,00

13 Fila de Espera
14 Peppa Pig 633,33

15
16 Química 783,33
17 Seleccionados
18 Deadpool 800,00
19 Peter Parker 783,33

20 Fila de Espera
21 Peppa Pig 633,33

22
23 Estadística 566,67
24 Seleccionados
25 Tio Patinhas 666,67
26 Super Mario Bros 566,67
27 Fila de Espera
28