



SÃO
PAULO
TECH
SCHOOL

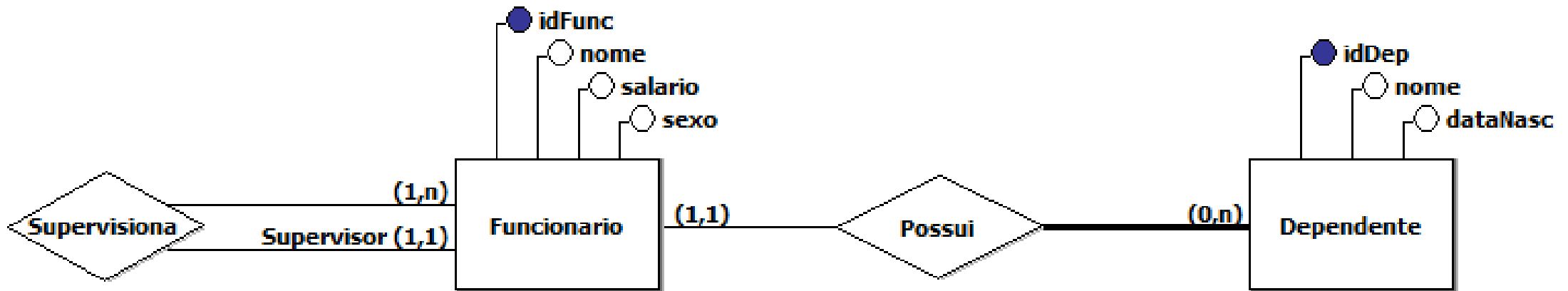


BD – Banco de Dados

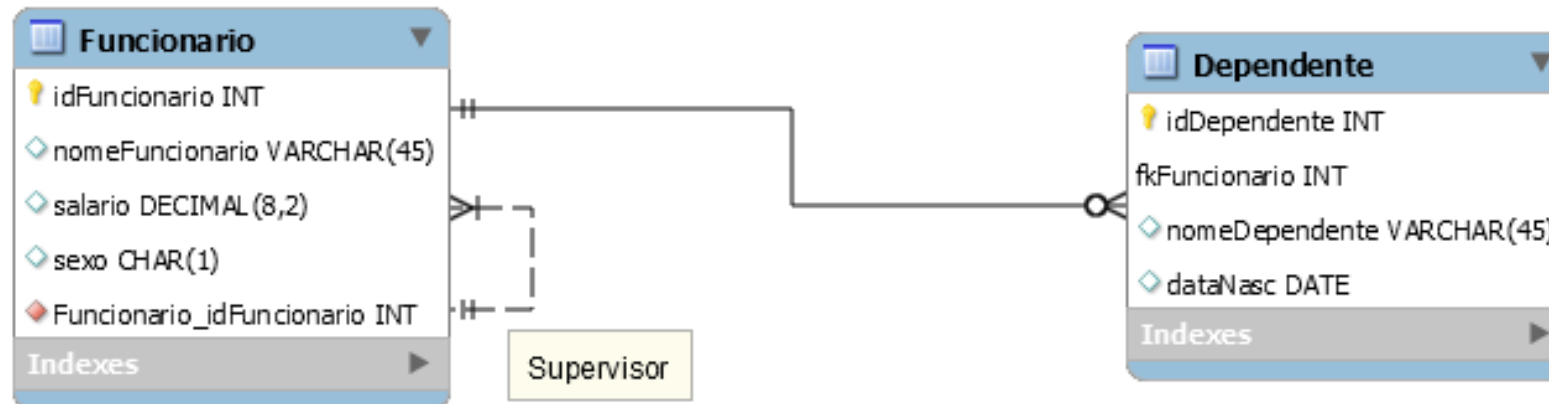
Modelagem Lógica – MySQL
Workbench

Modelagem Lógica – MySQL Workbench

- Podemos elaborar o modelo lógico correspondente a este DER:

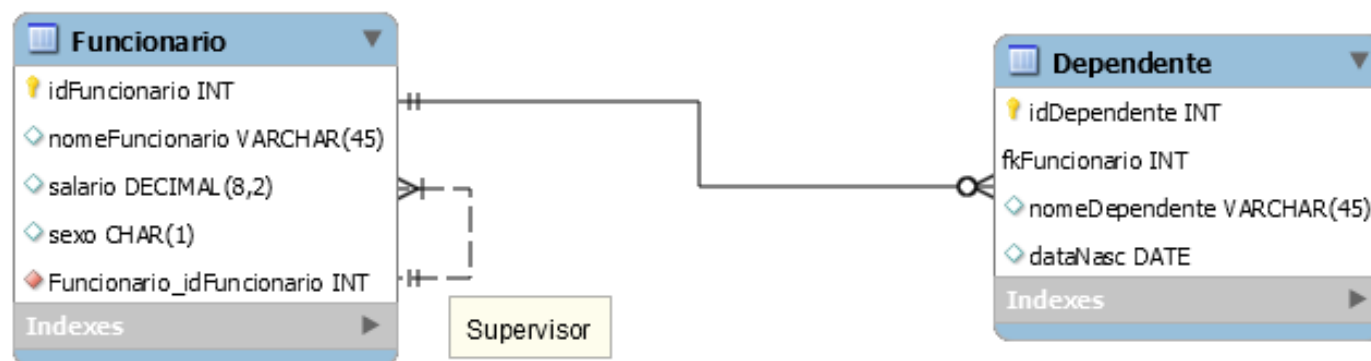


- Ficaria assim:



Modelagem Lógica – MySQL Workbench





- Repare que agora o relacionamento entre Funcionario e Dependente é um relacionamento entre uma entidade forte e uma fraca
- Por isso, a linha é cheia



- A tabela Dependente tem uma chave primária composta pelo idDependente e pelo fkFuncionario (informação que vem da tabela Funcionario)

Dependente - Table

Table Name: Dependente

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI
 idDependente	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 fkFuncionario	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 nomeDependente	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 dataNasc	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Implementação da Entidade Fraca no SQL

- Na Aula08, do Moodle, temos o Script-Funcionario, que implementa o modelo do Funcionario e do Dependente
- Create table da tabela Funcionario (valor de idFunc começará de 1000):

```
> create table Funcionario (  
    idFunc int primary key auto_increment,  
    nome varchar(40),  
    sexo char(1),  
    check (sexo = 'm' or sexo = 'f' or sexo = 'n'),  
    -- no MySQL, versões anteriores a 8.0.16, devem usar enum no lugar do check:  
    -- sexo enum('m','f','n'),  
    salario decimal(7,2), -- 7 significa que o número tem 7 dígitos, considerando os dígitos antes da  
                        -- vírgula e os depois  
                        -- 2 significa que tem 2 dígitos após a vírgula  
                        -- decimal(7,2) significa um número do tipo 99999,99  
    check (salario > 0), -- isso não funciona em versões anteriores a 8.0.16 do MySQL  
                        -- se fosse nota: check (nota >= 0 and nota <= 10)  
    fkSupervisor int, -- implementação de um relacionamento recursivo um para muitos  
    foreign key(fkSupervisor) references Funcionario(idFunc)  
~ ) auto_increment = 1000;
```

Implementação da Entidade Fraca no SQL

- Inserção de registros na tabela Funcionario

```
-- inserção de um funcionário, que não tem supervisor
```

```
insert into Funcionario values
```

```
(null, 'Maria Aparecida','f',15000,null);
```

```
-- inserção de mais um funcionário (supervisor deste funcionário será a Maria Aparecida - id 1000)
```

```
insert into Funcionario values
```

```
(null, 'José da Silva','m',5000,1000);
```

```
-- inserção de mais um funcionário (supervisor deste funcionário será o José da Silva - id 1001)
```

```
insert into Funcionario values
```

```
(null, 'João Oliveira','m',2000,1001);
```

- Exibição da tabela Funcionario

19 • select * from Funcionario;

Result Grid | Filter Rows: | Edit: |

	idFunc	nome	sexo	salario	fkSupervisor
▶	1000	Maria Aparecida	f	15000.00	NULL
	1001	José da Silva	m	5000.00	1000
	1002	João Oliveira	m	2000.00	1001
*	NULL	NULL	NULL	NULL	NULL

Implementação da Entidade Fraca no SQL

- Criação da tabela Dependente (entidade fraca)

```
1  -- Criação da tabela Dependente, que é uma entidade fraca em relação a Funcionario
2  • ○ create table Dependente (
3      fkFunc int,
4      foreign key (fkFunc) references Funcionario(idFunc),
5      idDep int,
6      primary key (fkFunc, idDep), -- chave primária composta
7      nome varchar(40),
8      dataNasc date
9  );
```

- Inserção de registros na tabela Dependente:

```
    -- Inserção de dependentes
1  insert into Dependente values
2      (1000, 1, 'Ana', '2012-10-01'),
3      (1000, 2, 'Paulo', '2014-03-05'),
4      (1001, 1, 'Antônio', '2013-09-06'),
5      (1002, 1, 'Clara', '2014-06-01');
```

- Observe que o idDep começa com o valor 1 para cada fkFunc diferente

Função SUM – Retorna a soma

- Quando uma coluna tem valores numéricos, podemos obter a soma dos valores da coluna
- Ex: podemos obter a soma da coluna salario da tabela Funcionario

```
72 • select sum(salario) from Funcionario;
```

```
73
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
sum(salario)				
22000.00				

- Costumamos usar o **as** para nomear a coluna resultante do comando:

```
73 • select sum(salario) as Total_Salario from Funcionario;
```

```
74
```


Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Total_Salario				
22000.00				

Função AVG – Retorna a média (average)

- Da mesma forma, é possível obter a média dos valores da coluna

```
75 • select avg(salario) from Funcionario;
```

```
76
```




The screenshot shows a SQL query result in a web interface. The query is `select avg(salario) from Funcionario;`. The result is displayed in a table with one column labeled `avg(salario)` and one row with the value `7333.333333`. The interface includes a 'Result Grid' tab, a 'Filter Rows' input field, and buttons for 'Export' and 'Wrap Cell Content'.

avg(salario)
7333.333333

- Repetindo o comando, mas nomeando a coluna resultante com o **as**:

```
76 • select avg(salario) as 'Média dos Salários' from Funcionario;
```

```
77
```



The screenshot shows a SQL query result in a web interface. The query is `select avg(salario) as 'Média dos Salários' from Funcionario;`. The result is displayed in a table with one column labeled `Média dos Salários` and one row with the value `7333.333333`. The interface includes a 'Result Grid' tab, a 'Filter Rows' input field, and buttons for 'Export' and 'Wrap Cell Content'.

Média dos Salários
7333.333333

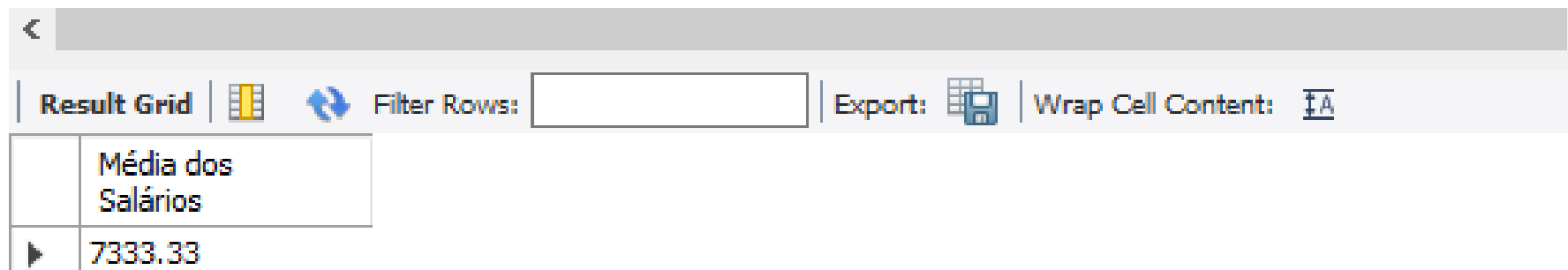
- Quando o nome após o **as** contém espaços, use aspas simples, como **'Média dos Salários'**

Função ROUND – Arredonda um valor

- Quando o valor retornado tem várias casas decimais, como a média do slide anterior (7333.333333), podemos usar a função **round** para arredondar o valor para a quantidade de casas decimais desejadas:
- Para arredondar para 2 casas decimais:

```
77 • select round(avg(salario),2) as 'Média dos Salários' from Funcionario;
```

```
78
```



The screenshot shows a database query result interface. At the top, there is a navigation bar with a back arrow. Below it, a toolbar contains icons for 'Result Grid', 'Filter Rows' (with a dropdown menu), 'Export' (with a download icon), and 'Wrap Cell Content' (with a text icon). The main area displays a table with one row. The first column is empty, and the second column contains the text 'Média dos Salários'. Below this, a second row shows a right-pointing arrow in the first column and the value '7333.33' in the second column.

	Média dos Salários
▶	7333.33

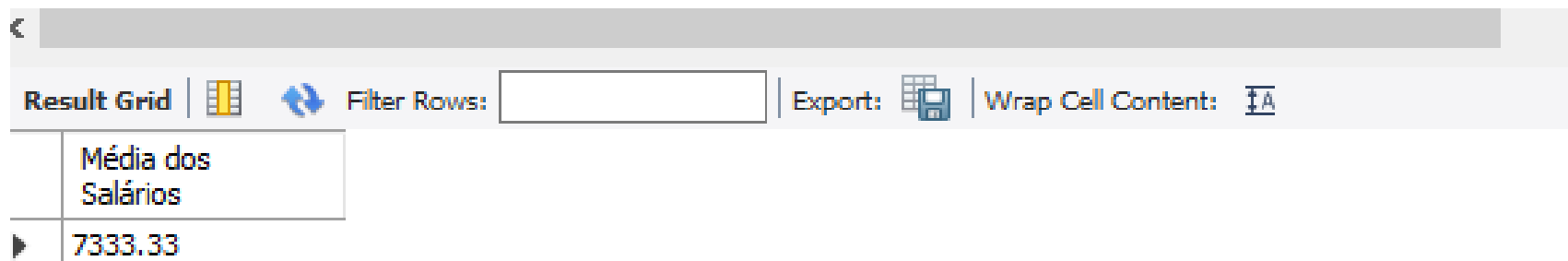
Função TRUNCATE – Trunca um valor

- Quando o valor retornado tem várias casas decimais, como a média 7333.333333, podemos usar também a função **truncate** para truncar o valor para a quantidade de casas decimais desejadas:

- Para truncar para 2 casas decimais:

```
78 • select truncate(avg(salario),2) as 'Média dos Salários' from Funcionario;
```

```
79
```



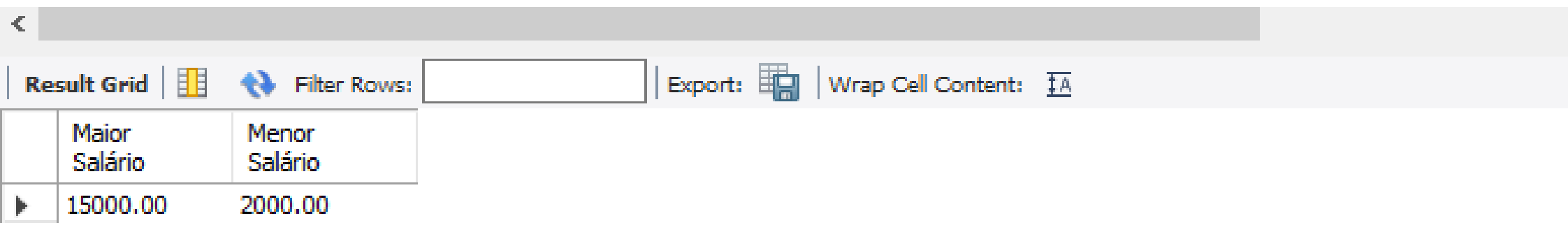
Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Média dos Salários			
	7333.33			

- Nesse caso, o round e o truncate produziram o mesmo resultado
- Se o número fosse 7333.338888:
 - Round para 2 casas decimais produziria 7333.34
 - Truncate para 2 casas decimais produziria 7333.33

Função MAX e MIN – Retornam maior e menor valor

- Da mesma forma, é possível obter o maior e o menor valor de uma coluna:

```
79 • select max(salario) as 'Maior Salário', min(salario) as 'Menor Salário' from Funcionario;  
80
```



	Maior Salário	Menor Salário
▶	15000.00	2000.00

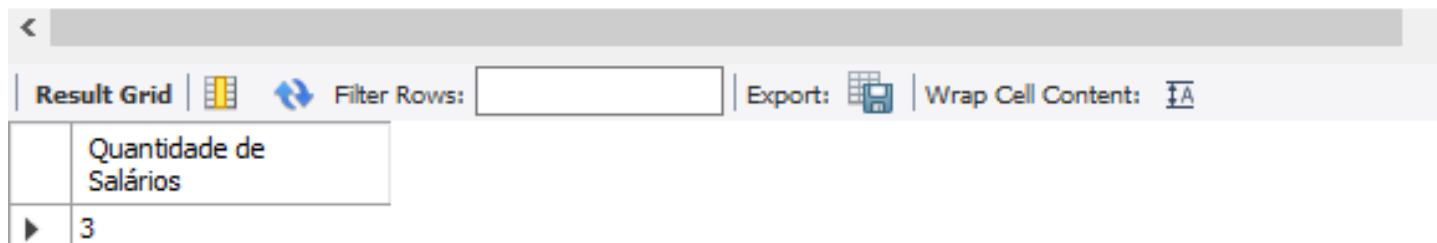
- Obs.: No exemplo acima, solicitei os 2 valores num único comando, mas é possível também fazer separadamente.

Função COUNT – Retorna a quantidade

- A função **count** é utilizada para saber a quantidade de valores:

```
80 • select count(salario) as 'Quantidade de Salários' from Funcionario;
```

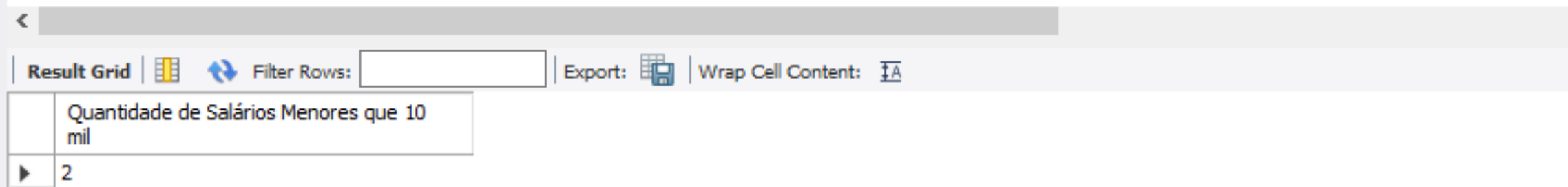
```
81
```



Quantidade de Salários
3

- No exemplo a seguir, usamos **count** para saber quantos salários são menores do que 10 mil:

```
81 • select count(salario) as 'Quantidade de Salários Menores que 10 mil' from Funcionario where salario < 10000;
```



Quantidade de Salários Menores que 10 mil
2

Soma e média somente de alguns salários

- Também é possível somar e obter a média de apenas alguns salários da coluna, como por exemplo, os salários menores que 10000.

```
82 • select sum(salario) as 'Total Salários Abaixo 10k',  
83      avg(salario) as 'Média Salários Abaixo 10k' from Funcionario where salario < 10000;
```

<

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	Total Salários Abaixo 10k	Média Salários Abaixo 10k
▶	7000.00	3500.000000



Agradeço a sua atenção!

Vivian Silva

vivian.silva@sptech.school

SÃO
PAULO
TECH
SCHOOL