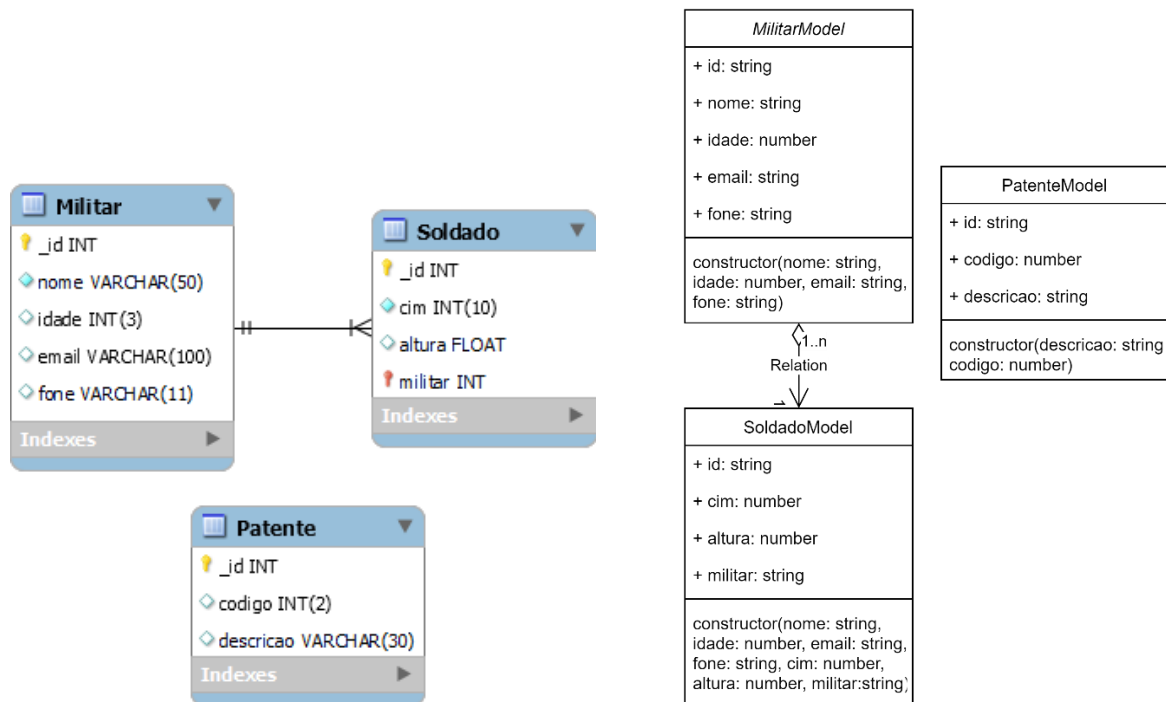


Prova III – Técnicas de Programação I – DSM - Prof. Henrique Louro - 28/11/2024

1) Dado o MER e o diagrama UML de Classes a seguir, implemente-os em TypeScript com o Mongoose:



2) Validações (validator) dos campos dos documentos **Militar**:

a. email:

i. validado pelo *validator* com requisitos informados abaixo:

- O e-mail precisa ter pelo menos um "@" e um ponto. Utilize a expressão regular `/^[^@s@]+@[^@s@]+\.[^@s@]+$/` para fazer tal validação;
- Só poderão ser aceitos e-mails de militares das forças armadas brasileira e devem conter `@eb` ou `@marinha` ou `@fab`, bem como o sufixo `".mil.br"`;

b. fone:

i. validado pelo *validator* com requisitos informados abaixo:

- O número de telefone deverá ter de 10 a 11 dígitos entre 0 e 9. Utilize a expressão regular `/^[0-9]{10,11}$/` para fazer tal validação;
- Os dois primeiros dígitos do número de telefone devem indicar um DDD válido. Segue array numérico, com os números de DDDs válidos no Brasil:

```
const ddds = [11,12,13,14,15,16,17,18,19,21,22,24,27,28,31,32,33,34,35,37,38,41,42,
43,44,45,46,47,48,49,51,53,54,55,61,62,63,64,65,66,67,68,69,71,73,74,75,77,79,81,82,
83,84,85,86,87,88,89,91,92,93,94,95,96,97,98,99];
```

3) Validações (validator) dos campos dos documentos **Soldado**:

a. altura: valores reais $\geq 1,62$ m;

b. militar:

i. validado pelo *validator* que deverá verificar se o ID pertence a um documento Militar cadastrado no banco;

4) Persista os seguintes dados nas coleções utilizando a API Fetch e as rotas criadas:

Prova III – Técnicas de Programação I – DSM - Prof. Henrique Louro - 28/11/2024

a. Militar:

```
var militares: Array<MilitarModel> = [];
var militar = new MilitarModel("Marcos da Silva", 21, "marcos.silva@fab.mil.br", "12912343567");
militares.push(militar);
militar = new MilitarModel("Ana Maria Brega", 25, "ana.brega@fab.mil.br", "12999979999");
militares.push(militar);
militar = new MilitarModel("Paulo França", 18, "paulo.fraca@fab.mil.br", "12999967999");
militares.push(militar);
militar = new MilitarModel("Edson Arantes", 30, "edson.arantes@gmail.sp.gov.br", "12999957999");
militares.push(militar);
```

b. Soldado:

```
var soldados: Array<SoldadoModel> = [];
var soldado = new SoldadoModel(1234567891, 1.73, "Marcos da Silva");
soldados.push(soldado);
soldado = new SoldadoModel(1212121212, 1.69, "Ana Maria Brega");
soldados.push(soldado);
soldado = new SoldadoModel(2121212121, 1.8, "Paulo França");
soldados.push(soldado);
```

c. Patente:

```
var patentes: Array<PatenteModel> = [];
var patente = new PatenteModel("Marechal do Ar", 1);
patentes.push(patente);
patente = new PatenteModel("Brigadeiro", 2);
patentes.push(patente);
patente = new PatenteModel("Coronel", 3);
patentes.push(patente);
var patente = new PatenteModel("Major", 4);
patentes.push(patente);
var patente = new PatenteModel("Capitão", 5);
patentes.push(patente);
```

5) Implemente no index.ts da pasta /src a seguinte função de máscara para números de telefone:

```
function phoneMask(v: string | undefined) {
    if (v == undefined) {
        return
    }
    let r = v.replace(/\D/g, "");
    r = r.replace(/^0/, "");
    if (r.length >= 11) {
        r = r.replace(/^(\d\d)(\d{5})(\d{4}).*/ , "($1) $2-$3");
    } else if (r.length > 7) {
        r = r.replace(/^(\d\d)(\d{5})(\d{0,5}).*/ , "($1) $2-$3");
    } else if (r.length > 2) {
        r = r.replace(/^(\d\d)(\d{0,5})/ , "($1) $2");
    } else if (v.trim() !== "") {
        r = r.replace(/^(\d*)/ , "($1)");
    }
    return r;
}
```

Prova III – Técnicas de Programação I – DSM - Prof. Henrique Louro - 28/11/2024

- 6) No mesmo index implemente uma rotina que liste no console todos os soldados cadastrados e seus respectivos relacionamentos com a coleção Militar. A saída esperada é essa:

```
<< Soldado >>  
CIM: 1234567891  
Nome: Marcos da Silva  
Idade: 21  
Altura: 1,73 m  
e-Mail: marcos.silva@fab.mil.br  
Telefone: (12) 91234-3567  
<< Soldado >>  
CIM: 1212121212  
Nome: Ana Maria Brega  
Idade: 25  
Altura: 1,69 m  
e-Mail: ana.brega@fab.mil.br  
Telefone: (12) 99997-9999  
<< Soldado >>  
CIM: 2121212121  
Nome: Paulo França  
Idade: 18  
Altura: 1,8 m  
e-Mail: paulo.fraca@fab.mil.br  
Telefone: (12) 99996-7999
```

- 7) Pesquise o método **toLocaleString** para formatar a saída da altura do soldado, conforme mostrado anteriormente.



Boa prova!