

Projeto BD - Parte 1

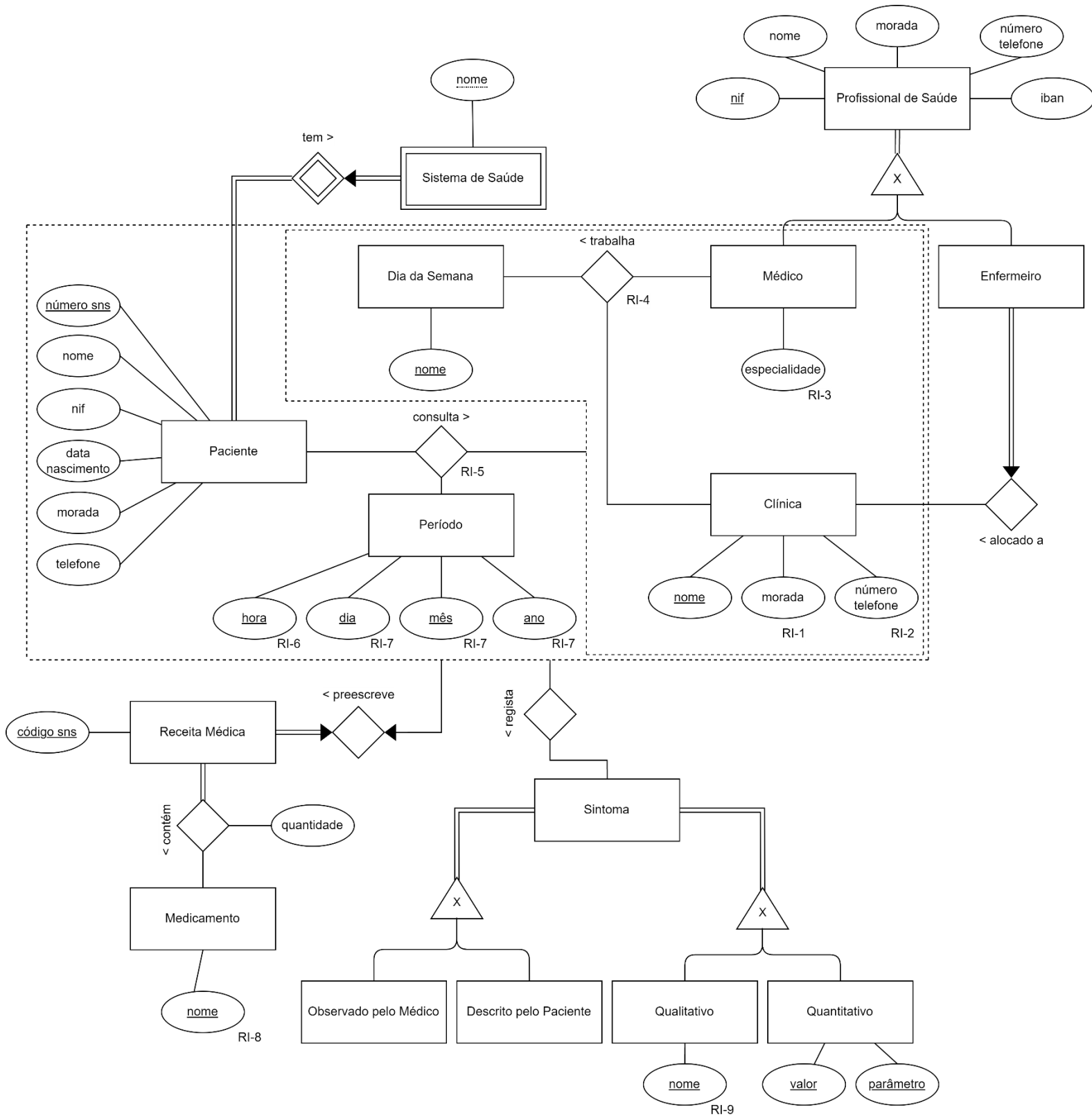
Professor Daniel Faria

GRUPO 39 - TURNO BD25L05

Aluno	Esforço (horas)
Vasco Conceição (106481)	17 horas (33.3%)
Henrique Luz (99417)	17 horas (33.3%)
Nuno Martins (107273)	17 horas (33.3%)

1. Modelação Entidade-Associação

Modelo Entidade-Associação:



Restrições de integridade:

RI-1: A morada é única em Clínica.

RI-2: O número telefone é único em Clínica.

RI-3: A **especialidade** em **Médico** tem de pertencer a uma lista de especialidades reconhecidas pela ordem dos médicos.

RI-4: Cada tuplo **nif, dia, mês, ano** em **trabalha** é único.

RI-5: Os pares **período, médico** e **período, paciente** em **consulta** são únicos.

O **Período** em **consulta** tem de ocorrer associado ao **nome** do **Dia da Semana** correspondente.

RI-6: A **hora** em **Período** tem de ser um período de 30 minutos entre as 8:00 e as 20:00.

RI-7: O **dia, mês, ano** em **Período** tem de corresponder a um dia útil.

RI-8: O **nome** em **Medicamento** tem de pertencer à lista oficial da Infarmed.

RI-9: O **nome** em **Qualitativo** tem de pertencer a uma lista proveniente do vocabulário controlado SNOMED CT.

2. Conversão E-A-Relacional

Modelo Relacional:

A(a1, a2, a3)
B(a1, b1)
 a1: FK(A)
C(a1)
 a1: FK(A)
E(e1, e2)
rCE(a1, e1, e2, rce1)
 a1: FK(C)
 e1, e2: FK(E) NOT NULL
F(f1, f2, f3)
G(g1)
H(h1, h2)
rAF(a1, f1, f2, g1, h1)
 a1: FK(A)
 f1, f2: FK(F)
 g1: FK(G) NOT NULL
 h1: FK(H)
D(a1, f1, f2, d1)
 a1, f1, f2: FK(rAF)

Restrições de integridade:

RI-1: cada a1 em A tem de ocorrer ou em B ou em C mas não em ambos.
RI-2: qualquer f1 em F tem de ocorrer em rAF.
RI-3: qualquer h1 em H tem de ocorrer em rAF.

3. Álgebra Relacional & SQL

1. $\Pi_{SSN, especialidade}(consulta \bowtie_{consulta.NIF=medico.NIF} medico) \div \Pi_{especialidade}(medico)$

2. Qual a especialidade com mais consultas?

3. Quais os pacientes que num dado dia têm mais que uma consulta?

4. A expressão não está semanticamente correta face à interrogação pretendida.

▪ No código:

```
WHERE c.periodo IN (  
  SELECT MIN(periodo)  
  FROM consulta  
  WHERE NIF = m.NIF  
  GROUP BY SSN  
)
```

seleciona-se apenas as primeiras consultas entre um dado médico e um dado paciente.

▪ Depois, em:

```
COUNT(c.periodo) AS consultas_medico  
COUNT(DISTINCT p.NIF) AS total_pacientes  
COUNT(c.periodo) / COUNT(DISTINCT p.NIF) AS proporcao_fidelidade  
(...)  
GROUP BY m.nome
```

contam-se as consultas de cada médico, os pacientes diferentes de cada médico, e, por fim, dividem-se os resultados obtidos anteriormente, respetivamente.

No entanto, como já se retirou qualquer eventual repetição de uma consulta entre médico e paciente, todas as consultas são com pacientes diferentes e, portanto, o número de consultas de cada médico é igual ao número de pacientes diferentes de cada médico.

Naturalmente, a coluna `proporcao_fidelidade` vai ser totalmente preenchida de 1's, por estar a dividir números iguais.

▪ A expressão apresentada pelo chatGPT também não engloba a especialidade, que foi mencionada na interrogação.

▪ Para uma semântica correta, propomos a seguinte expressão:

```
SELECT m.nome AS nome_medico,
```

```

COUNT(c.periodo) AS consultas_especialidade,
e.num AS total_pacientes,
CAST(e.num - COUNT(c.periodo) AS DECIMAL) / COUNT(periodo) AS proporcao_fidelidade
FROM medico m
INNER JOIN consulta c ON m.NIF = c.NIF
INNER JOIN paciente p ON c.SSN = p.SSN
INNER JOIN (
    SELECT especialidade,
    COUNT(periodo) AS num
    FROM consulta c
    INNER JOIN medico m ON m.NIF = c.NIF
    GROUP BY especialidade
) AS e ON e.especialidade = m.especialidade
GROUP BY m.nome, total_pacientes
ORDER BY proporcao_fidelidade;

```

Desta forma, a coluna `proporcao_fidelidade` apresenta o pedido: a quantidade de vezes que os pacientes consultam outro médico da mesma especialidade em proporção ao número de vezes que consultam o médico. Assim, não é necessário de apresentar os resultados por ordem decrescente, tal como sugeriu o chatGPT, porque uma menor `proporcao_fidelidade` significa que o médico tem pacientes mais fiéis.

Um problema que identificámos com a solução do chatGPT foi também a divisão inteira em `proporcao_fidelidade`. Para resultados mais verdadeiros, precisamos de usar a divisão normal (fazendo cast para decimal de um dos participantes da divisão) por se conseguir ordenar fidelidades diferentes, embora muito próximas, ordem essa que se poderia perder se optarmos pela divisão inteira, que poderia dar resultados iguais.