

Dupla JL
Etapa 4

Nome: João Paulo Soubihe
Nome: Leandro Ferlin Viana

RA: 151106
RA: 148729

Resumo:

Nosso trabalho consiste em auxiliar o médico na hora de diagnosticar seus pacientes.

Através de um banco de dados, que será alimentado, nosso sistema irá indicar probabilidades de diagnósticos, ou seja, qual doença é mais possível que o paciente tenha contraído.

Após a análise o médico poderá avaliar, ou pedir um exame, para confirmar o diagnóstico. Caso essa avaliação não confirme a doença com mais probabilidade o médico poderá alimentar o sistema com essa informação e permitir com que as probabilidades sejam recalculadas a partir daquela informação.

Requisitos:

- a) a aplicação deve manter um banco de dados com os sintomas apresentados pelos pacientes e o diagnóstico confirmado após o tratamento.
- b) o sistema deve armazenar dados pessoais do paciente para identificá-lo. Porém, é de extrema importância assegurar que esses dados serão usados apenas para manter um registro de cada paciente¹.
- c) identificar o usuário, mantendo sua privacidade de dados garantida: o médico busca o cadastro do paciente na aplicação e, caso não exista, insere os dados do paciente.
- d) com o cadastro do paciente realizado, o médico insere os sintomas apresentados por ele.
- e) a aplicação então realiza análise estatística no banco de dados a partir do histórico de diagnósticos coletados até o presente momento, e informa quais são as doenças mais prováveis e suas respectivas probabilidades.
- f) o médico deve utilizar a previsão de diagnóstico como um ponto de partida para o tratamento, realizando, caso julgue necessário, outros exames laboratoriais
- g) ao final do tratamento, o médico deve voltar à aplicação e finalizar tratamento, informando se a previsão do sistema estava correta e, caso contrário, reportar o diagnóstico correto.
- h) a aplicação deve alimentar o banco de dados com os dados coletados (sintomas apresentados e doença detectada) e utilizar esses dados para aperfeiçoar futuras previsões de .

¹ Apenas os dados referentes aos conjunto de sintomas e diagnóstico final serão usados para análise estatísticas e futuras previsões, dados pessoais serão mantidos sob sigilo.

Banco de dados:

a) Disease-Symptom Knowledge Database

<http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html>

Trabalharemos com o dados coletados de pacientes do Hospital Presbiteriano de Nova York, no ano de 2004.

O Disease-Symptom Knowledge Database reúne as 150 doenças mais comuns durante esse ano, o número de casos positivos e os sintomas apresentados pelos pacientes em ordem decrescente segundo a força da relação sintoma - doença.

b) Diseases Database

<http://www.diseasesdatabase.com/content.asp>

O Diseases Database é índice de referência cruzada de doenças humanas, medicamentos, sintomas, sinais e resultados de investigações anormais. Esse site fornece um índice semelhante a um livro de texto médico e um mecanismo de pesquisa

c) National Health Interview Survey

https://ftp.cdc.gov/pub/Health_Statistics/NCHS/NHIS/SHS/2016_SHS_Table_A-4.pdf

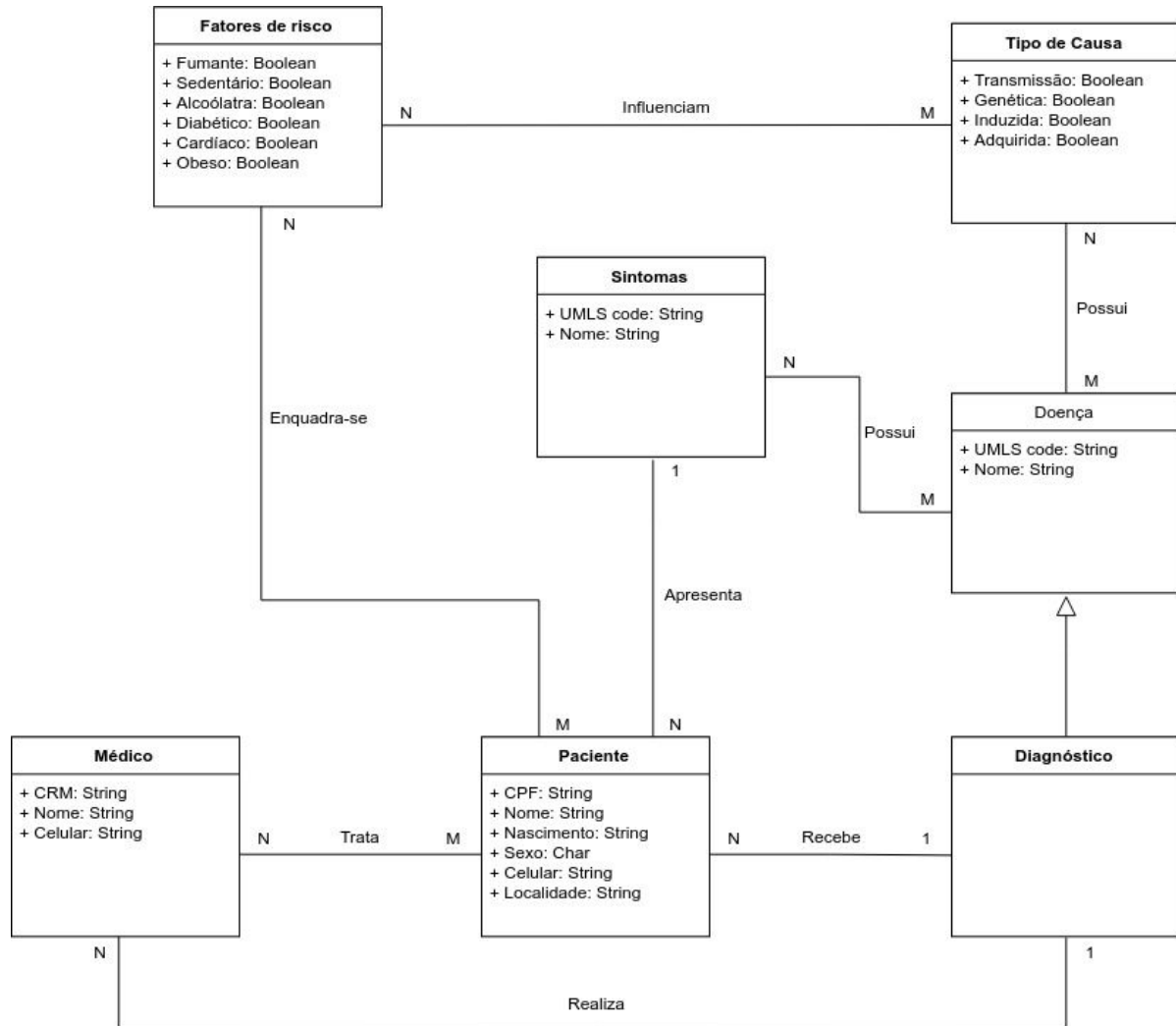
https://ftp.cdc.gov/pub/Health_Statistics/NCHS/NHIS/SHS/2016_SHS_Table_A-2.pdf

https://ftp.cdc.gov/pub/Health_Statistics/NCHS/NHIS/SHS/2016_SHS_Table_A-1.pdf

O CDC (Centers of Disease Control and Prevention) realiza todo ano uma pesquisa relacionando ocorrências de doenças a hábitos e características das pessoas. Num estudo bem amplo estudam a relação de Sexo, Idade, Raça, Escolaridade, Renda familiar, Região em que mora, Estado Civil, entre outros fatores.

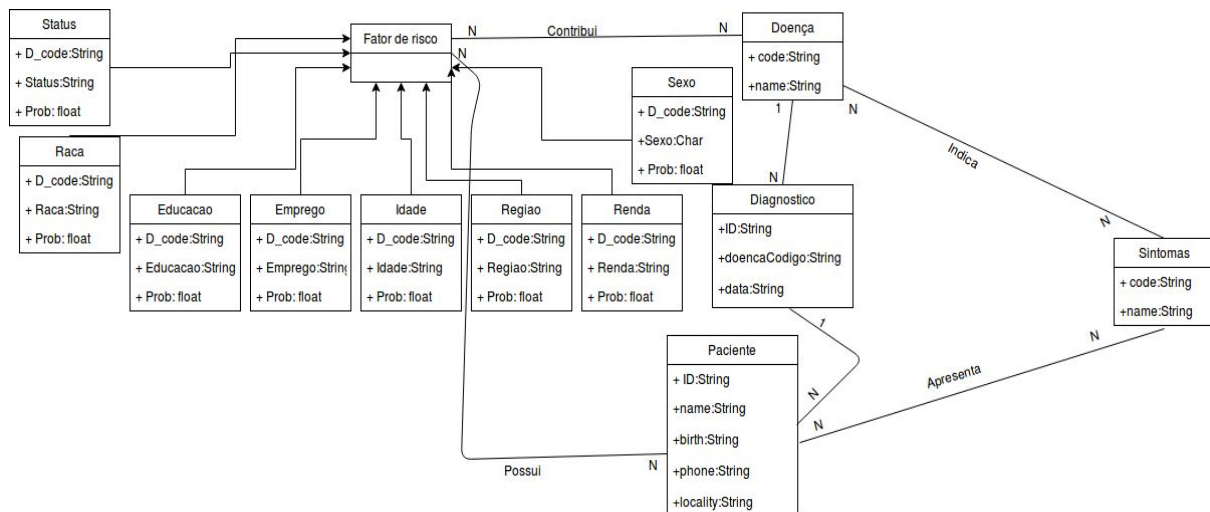
Modelagem conceitual:

- Modelagem inicial



Modelo Etapa 1

- Após algumas avaliações e pesquisa de databases na internet chegamos a um novo modelo:

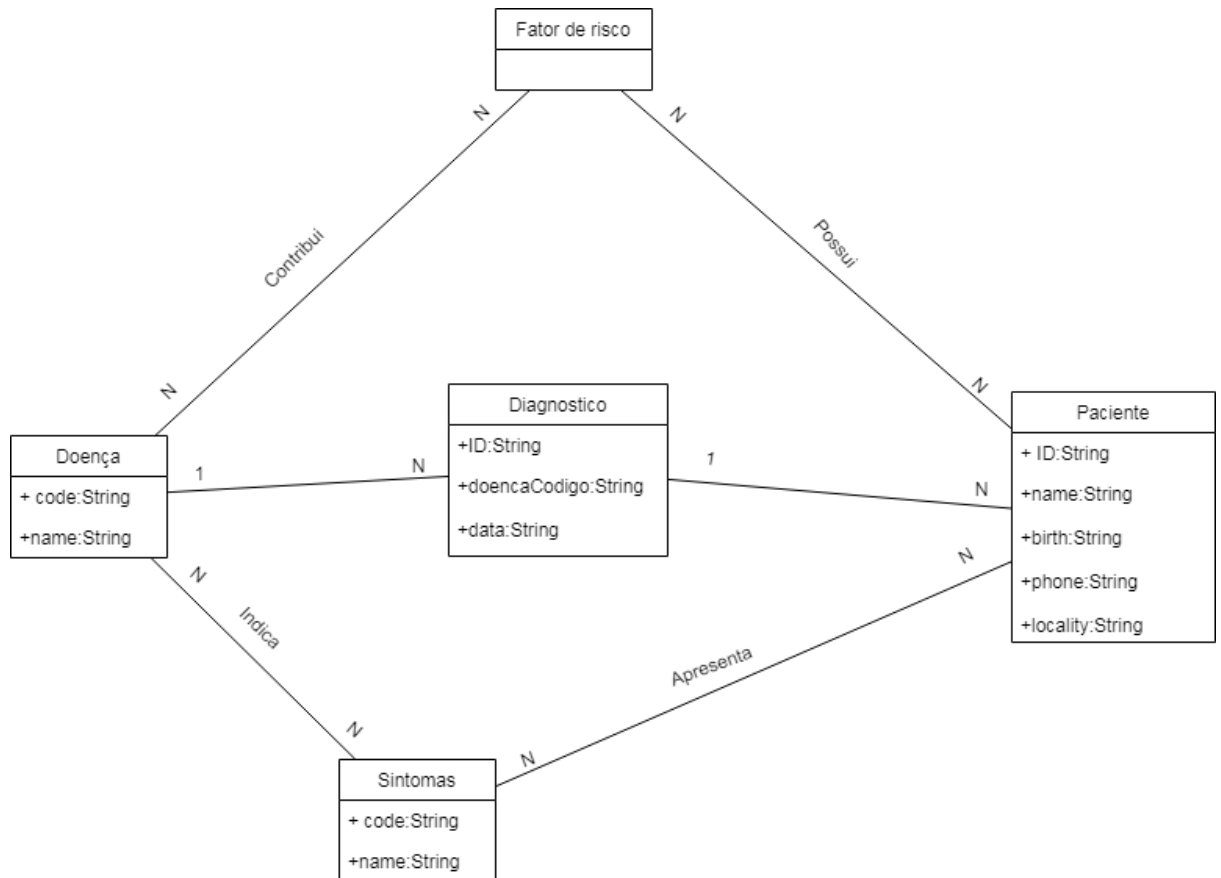


Modelo Etapa 2

Onde as classes *Médico* e *Tipos de Causa* foram removidas do nosso domínio, pois pensamos que elas seriam irrelevantes para a proposta do nosso projeto. A classe *Fator de Risco* foi especializada em cada tipo de fator avaliado pelo nosso sistema.

- Após a exposição do modelo ao professor, surgiu a ideia de mantermos uma única tabela para todos os fatores de risco. Essa alteração visa simplificar nosso modelo, reduzindo o número de classes e permitindo maior flexibilização do quesito Fator de Risco.

Com essas alterações obtemos o diagrama de classes abaixo:



Modelo Etapa 3

Modelagem lógica:

A partir da 3ª versão do diagrama podemos traduzir uma modelagem lógica como a descrita nessa seção.

Doenca (codigo,nome)

<u>codigo</u>	nome
---------------	------

Sintomas (codigo,nome)

<u>codigo</u>	nome
---------------	------

Paciente(ID, nome,nascimento,celular,localidade)

<u>ID</u>	nome	nascimento	celular	localidade
-----------	------	------------	---------	------------

Fator(Codigo,Fator,Tipo)

<u>Codigo</u>	<u>Fator</u>	Tipo
---------------	--------------	------

Diagnostico (ID, doencaCodigo, data)

CHE: doencaCodigo -> Doenca (codigo)

<u>ID</u>	<u>doencaCodigo</u>	data
-----------	---------------------	------

PacienteSintoma(pacienteID, sintomaCodigo)

CHE: IDpaciente -> Paciente (ID)

CHE: CodSintoma -> Sintomas (codigo)

<u>Idpaciente</u>	<u>CodSintoma</u>
-------------------	-------------------

DoencaSintomas (doencaCodigo, sintomaCodigo)

CHE: doencaCodigo -> Doenca (codigo)

CHE: sintomaCodigo -> Sintomas (codigo),

<u>doencaCodigo</u>	<u>sintomaCodigo</u>	<u>Ocorrencias</u>
---------------------	----------------------	--------------------

PacienteDiagnostico (pacienteID, diagnosticoID)

CHE: pacienteID -> Paciente (ID)

CHE: diagnosticoID -> Diagnostico(ID)

pacienteID	diagnosticoID
------------	---------------

FatorDoenca(CodFator,Dcode,Prob)

CHE: CodFator -> Fator(Codigo)

CHE: Dcode -> Doenca(codigo)

CodFator	Dcode	Prob
----------	-------	------

FatorPaciente(CodFator,IDpaciente)

CHE: CodFator -> Fator(Codigo)

CHE: IDpaciente -> Paciente(ID)

<u>CodFator</u>	Idpaciente
-----------------	------------

Nessa fase, trabalhamos a partir de grafos, com o software disponibilizado pelo Neo4j (<https://neo4j.com/sandbox-v2/?ref=hcard>). O trabalho feito pode ser encontrado em nossa página do GitHub, especificada no fim do documento.

Ao fim dessa etapa de desenvolvimento, observamos que o modelo baseado em grafos nos auxilia melhor para o fim de trabalho a partir de nosso banco de dados.

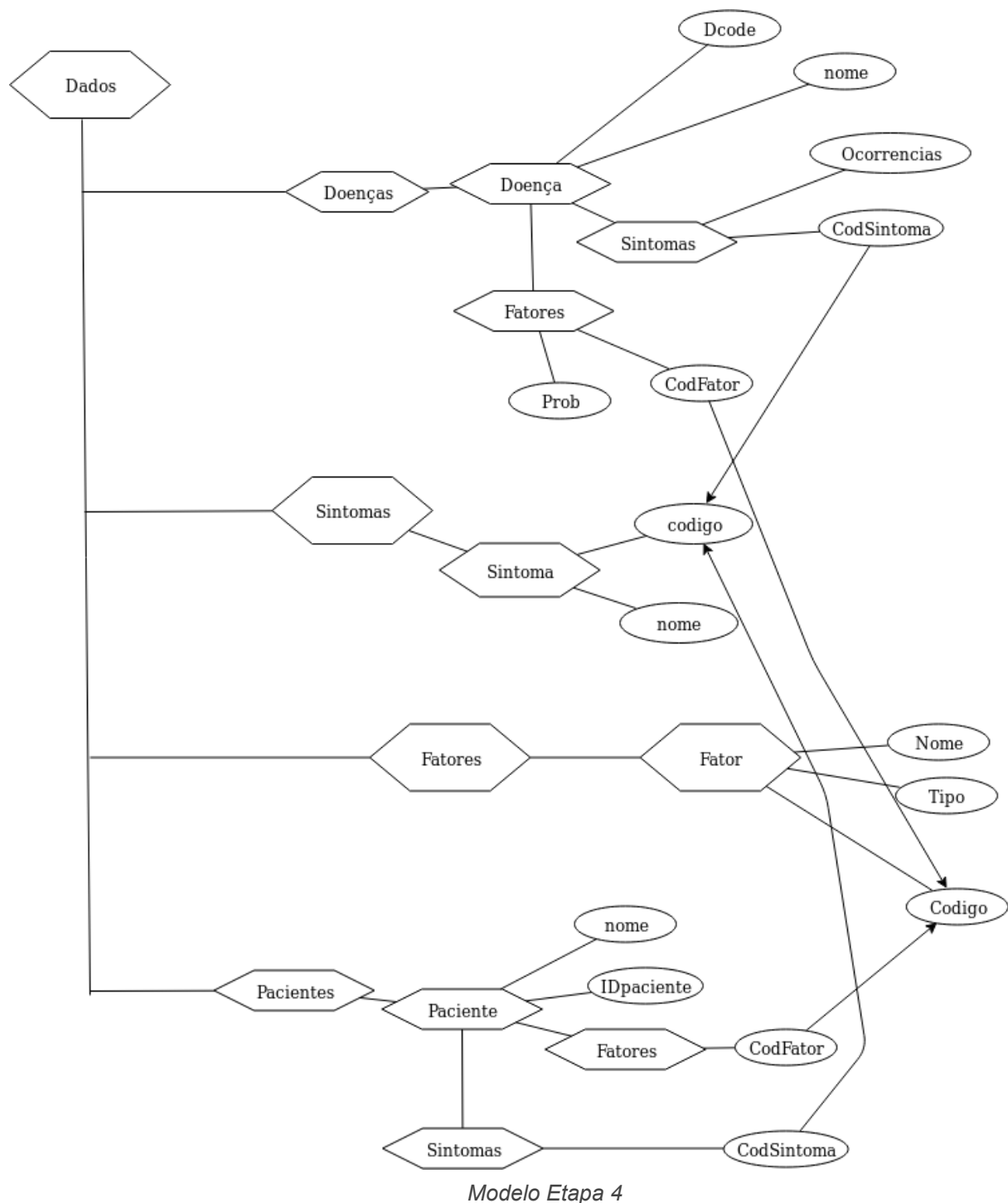
Notamos essa diferença pela maior facilidade em lidar com relações e “posses” de cada entidade (no caso de fator de risco com paciente, por exemplo).

Também permite uma implementação mais robusta, com funções mais complexas, além de que ele se baseia em caminhos, facilitando a visualização das relações entre as entidades.

Com a execução da etapa 4, na qual adaptamos nosso modelo para um modelo hierárquico, procuramos manter nossos dados já coletados e manipulá-los, de forma a se organizarem conforme foi requisitado.

Surge então um novo modelo lógico, pela não utilização de diagnósticos até o momento optamos por omiti-lo do modelo nessa fase. Posteriormente a função de avaliação segundo diagnósticos passados pode ser implementada sem complicações.

Agora há uma classe Dados, que nada mais é que um repositório que guarda todos os dados coletados, a classe superior em nossa hierarquia, que une as demais classes Doenças, Sintomas, Fatores e Pacientes. Em algumas dessas classes existe um relacionamento cruzado, onde um atributo serve como referência a outro atributo de outra classe, como é o caso de CodFator referenciando Codigo, da categoria Fatores, mesmo sendo um atributo que responde hierarquicamente a Doenças (ou Pacientes).



Conclusão:

Essa distribuição hierárquica dos dados permitiu com que organizássemos os dados de maneira mais clara e mais propícia a nossa manipulação e entendimento. Relacionar dados de hierarquias diferentes é um processo muito menos custoso que juntar tabelas e ir filtrando dados, etc. A redundância existente, no nosso caso, acabou não afetando a melhor execução do programa.

Um leigo muito possivelmente preferiria usar um modelo relacional pela sua fácil manipulação e aproximação com nosso mundo e em muitos casos seria realmente a melhor opção.

O modelo hierárquico é menos flexível que o modelo relacional, do ponto de vista que alterações no banco podem alterar cadeias inteiras no armazenamento dos dados, o modelo relacional consegue criar uma independência entre as tabelas, fazendo com que adições e alterações num banco relacional bem estruturado sejam menos “tumuladas”.