Identificação de Doenças Mentais

MC536 - UNICAMP - 2s 2018

Gabriel Alves Tabchoury - RA 171828
Paulo Afonso Martins Januário - RA 185441

Parte 1

1. Descrição

Banco de dados que reúne informações sobre as doenças mentais conhecidas, quais seus sintomas e dados sobre incidências conhecidas. Serão mapeadas as regiões e probabilidades de a doença ocorrer no local, relacionando com faixa etária e gênero. Ao relacionar com os sintomas, esses dados podem ajudar a prever probabilidades de certo paciente possuir a doença, bem como gerar dados de estatística.

2. Requisitos

- **Sintoma**
  - Nome
  - Descrição

- **Doença**
  - Nome
  - Descrição
  - Sintomas

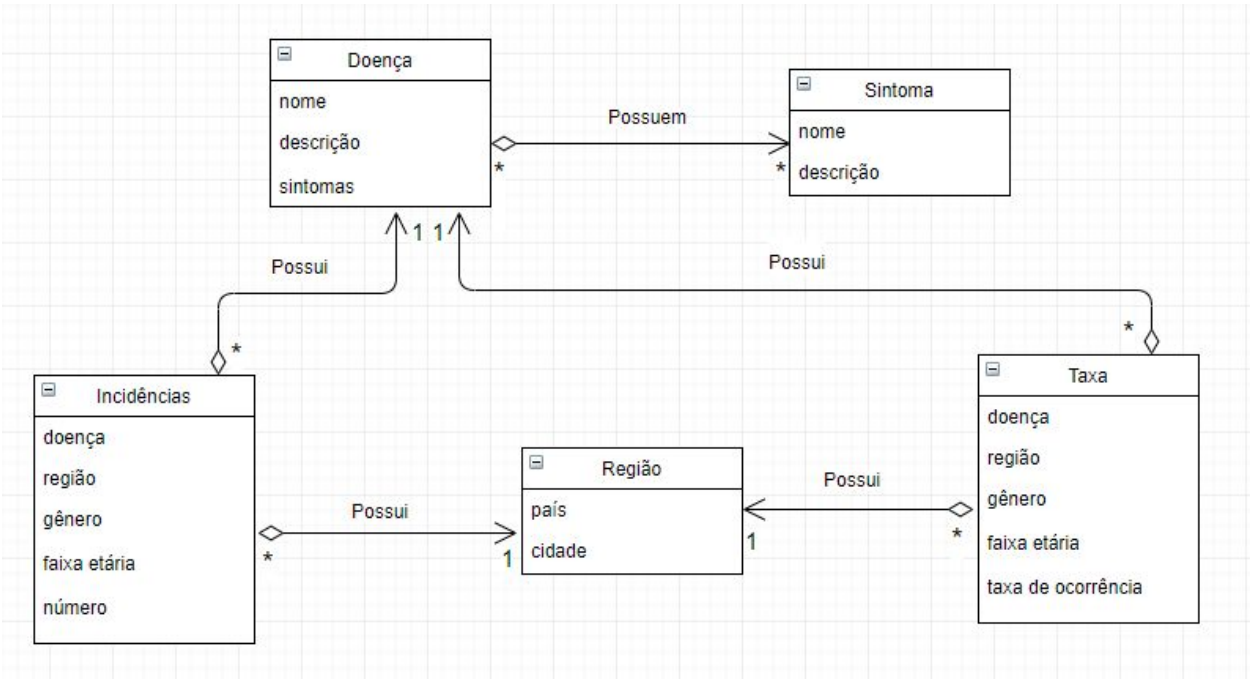
- **Região**
  - País
  - Cidade

- **Incidência**
  - Doença
  - Região
  - Gênero
  - Faixa etária

    - Número de ocorrências

- **Taxa**
    - Doença
    - Região
    - Gênero
    - Faixa etária
    - Taxa de ocorrências

3. Diagrama UML



4. Fontes

- Para alimentar o banco de dados serão utilizados os seguintes sites:
    - malacards.org - Human Disease Database - esse site possui dados sobre doenças conhecidas.
    - ghdx.healthdata.org - Global Health Data Exchange - esse site possui dados de incidências de doenças ao redor do mundo.

- Para entender mais sobre o tema e coletar estatísticas, os seguintes sites serão consultados:
    - ourworldindata.org - Our World In Data

- nimh.nih.gov - National Institute of Mental Health

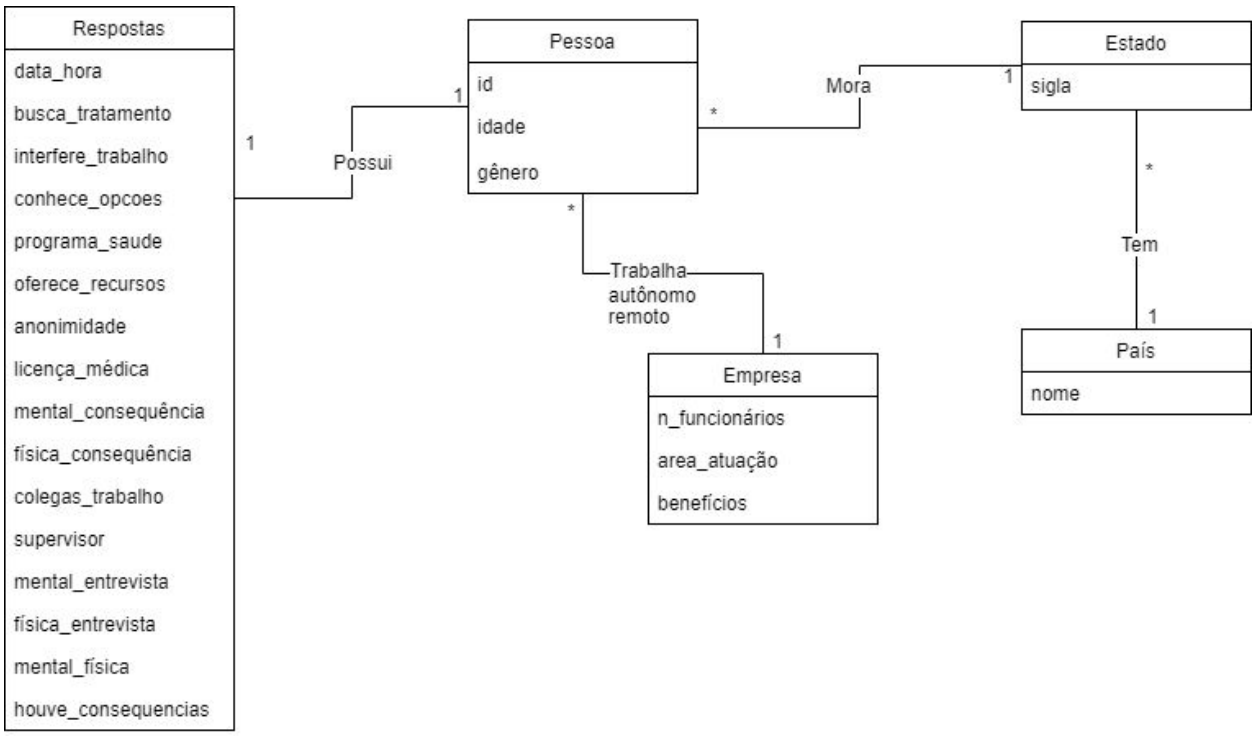
Parte 2

Na segunda etapa, a qual tivemos que procurar os dados para alimentar nosso banco, tivemos que alterar o nosso modelo conceitual pois não conseguimos encontrar dados compatíveis com o modelo que havíamos planejado.

A base do nosso modelo consiste em uma pesquisa de 2014 que mede atitudes em relação à saúde mental e frequência de transtornos mentais no ambiente de trabalho técnico, ou seja, em empresas de tecnologia.

Sendo assim, os nossos modelos conceitual e lógico ficaram assim:

1. Modelo Conceitual



2. Modelo Lógico

- **PESSOA** (id, idade, genero, estado, idEmpresa)
    - idEmpresa-> Chave Estrangeira: Empresa(id)
    - estado> Chave Estrangeira: Estado(sigla)

- **PESSOA_RESPOSTA** (id, idPessoa, data_hora, busca_tratamento, interfere_trabalho, conhece_opcoes, programa_saude, oferece_recursos, anonimidade, licenca_medica, mental_consequencia, fisica_consequencia, colegas_trabalho, supervisor, mental_entrevista, fisica_entrevista, mental_fisica, houve_consequencias)
    - idPessoa -> Chave Estrangeira: Pessoa(id)

- **EMPRESA**(id, n_funcionarios, area_atuacao, beneficios)

- **ESTADO**(sigla, pais)
    - pais> Chave Estrangeira: Pais(nome)

- **PAIS**(nome)

3. Fontes

- Dados da pesquisa : https://www.kaggle.com/osmi/mental-health-in-tech-survey

Parte 3

Na terceira etapa do trabalho tivemos algumas alterações no modelo lógico. Essas alterações foram feitas pelos seguintes motivos:

- Melhorar o nome dos campos para facilitar o entendimento;
- Analisando melhor, vimos que não temos dados suficientes para uma boa análise por Empresa, Estado e País uma vez que temos apenas 1256 entrevistas registradas. Portanto, decidimos unificar essas tabelas nas tabelas de Pessoa e Pessoa_Resposta. Para se ter uma ideia, mais da metade das respostas que temos está com o estado vazio além de que grande parte dos dados são apenas dos Estados Unidos e Inglaterra.

Sendo assim, o nosso modelo lógico ficou assim:

- **PESSOA** (id, idade, genero, estado,pais)

- **PESSOA_RESPOSTA** (idPessoa, data_hora, trabalho_proprio, istorico_familiar, tratamento, interferencia_trabalho, trabalho_remoto, empresa_ti, conhece_opcoes, programa_saude, busca_tratamento, anonimidade, licenca_medica, mental_consequencia, fisica_consequencia, mental_entrevista, fisica_entrevista, mental_fisica, houve_consequencias)
    - idPessoa -> Chave Estrangeira: Pessoa(id)

Modelo de Grafos

O modelo de banco de dados de grafos, possui relacionamentos mais naturais. As entidade (conhecidas como vértices) as quais são relacionadas por arestas podem guardar dados entre os relacionamentos além de que cada relacionamento pode ter uma direção. Um exemplo interessante é o Facebook no qual temos as entidades pessoas (vértices) que se conectam umas as outras e cada relacionamento existem diversas informações como por exemplo a data em que ele ocorreu, quem solicitou a amizade, quanto tempo demorou para a outra pessoa aprovar, etc.

Falando sobre a teoria de grafos, esse tipo de modelo está contido nas bases noSQL que são potencialmente mais rápidos e fáceis de escalar. Essa rapidez vem na hora de se comparar uma busca repleta de joins em um banco de dados relacional com a simplicidade nos relacionamentos em grafos. Além da velocidade, outra grande vantagem do modelo de grafos, a qual será fundamental nesse trabalho, é a visualização das informações. Como queremos fazer análise desses dados e como eles se relacionam, com o modelo de grafos fica bem mais fácil visualizar essas relações pela quantidade de ligações entre os vértices do grafo.

Parte 4

Essa última parte do trabalho envolve duas extensões das etapas anteriores com introdução na proposta das análises que envolvam inferências usando o RDF e SPARQL e percorrendo fontes de dados baseadas em XML usando o XQuery.

1. SPARQL

Nesta primeira parte da etapa 4, como fonte de dados, utilizamos o Medical Subject Heading RDF (https://id.nlm.nih.gov/mesh/) para realizarmos as consultas SPARQL.

O SPARQL é uma linguagem de query para realizar consultas em dados estruturados usando o padrão RDF. O termo SPARQL é um acrônimo recursivo que significa SPARQL Protocol and RDF Query Language.

Uma query SPARQL consiste numa estrutura simples de duas cláusulas: SELECT e WHERE. O SELECT identifica as variáveis que aparecerão nos resultados da query e o WHERE mostra o padrão básico do grafo que bate com os dados.

2. XQuery

Para as consultas XQuery, utilizamos como fonte a mesma pesquisa da Parte 2 do trabalho e portanto os modelos lógico e conceitual também são os mesmos. Desse modo, executamos uma query para fazer todos os JOIN's gerando um único CSV o qual utilizamos o site "http://convertcsv.com" para converter para XML.

A estrutura XML utilizada, está representada abaixo:

```
<PESQUISAS>
  <PESQUISA>
        <IDPESSOA></IDPESSOA>
        <DATA_HORA></DATA_HORA>
        <TRABALHO_PROPRIO></TRABALHO_PROPRIO>
        <HISTORICO_FAMILIAR></HISTORICO_FAMILIAR>
        <TRATAMENTO></TRATAMENTO>
        <INTERFERENCIA_TRABALHO></INTERFERENCIA_TRABALHO>
        <TRABALHO_REMOTO></TRABALHO_REMOTO>
        <EMPRESA_TI></EMPRESA_TI>
        <CONHECE_OPCOES></CONHECE_OPCOES>
        <PROGRAMA_SAUDE>o</PROGRAMA_SAUDE>
        <BUSCA_TRATAMENTO></BUSCA_TRATAMENTO>
        <ANONIMIDADE></ANONIMIDADE>
        <LICENCA_MEDICA></LICENCA_MEDICA>
        <MENTAL_CONSEQUENCIA></MENTAL_CONSEQUENCIA>
        <FISICA_CONSEQUENCIA></FISICA_CONSEQUENCIA>
        <MENTAL_ENTREVISTA></MENTAL_ENTREVISTA>
```

```
        <FISICA_ENTREVISTA></FISICA_ENTREVISTA>
        <MENTAL_FISICA></MENTAL_FISICA>
        <HOUVE_CONSEQUENCIAS></HOUVE_CONSEQUENCIAS>
        <ID></ID>
        <IDADE></IDADE>
        <GENERO></GENERO>
        <PAIS></PAIS>
        <ESTADO></ESTADO>
  </PESQUISA>
</PESQUISAS>
```

Vantagens:

O modelo XML tem como uma das principais vantagens a não necessidade de JOIN's para realizar consultas, ou seja, todos os dados estão concentrados. Como ele é apenas um arquivo contendo todos os dados, a busca é mais simples e direta e portanto, dependendo do volume de dados, os resultados podem ser mais rápidos. Além disso, um documento XML é de fácil visualização e entendimento uma vez que os próprios navegadores da internet fornecem suporte para leitura desses arquivos. Por fim, o XML também oferece vantagens quando a questão de conversão para outras extensões uma vez que o arquivo .xml pode ser facilmente convertido para diversos outros tipos de arquivos, como por exemplo CSV.