Fase 2 de Laboratório de Banco de Dados

MAC 0439

Professora: Kelly Rosa

Integrantes do grupo:

André Meneghelli
Fernanda de Camargo Magano
Florence Alyssa Sakuma Shibata
Yoshio Mori

Arrumando a fase 1

Requisitos funcionais:

==========
Versão 3 (modificada dia 26/10) - Atualização e modificações na estrutura dos requisitos
==========

- Esse sistema tem como objetivo viabilizar um prontuário digital contendo informações dos pacientes por meio de uma consulta.

- Nesse sistema serão inclusas doenças relacionadas aos pacientes e os médicos que estão acompanhando o progresso do tratamento, a dosagem dos medicamentos, local e procedimentos adotados para o tratamento, análise dos exames e também a manutenção do histórico referente à evolução do estado de saúde do enfermo.

- Uma doença está estruturada em hierarquia, podendo ser uma subcategoria de outra, isto é, pode haver uma doença mais genérica que engloba outras, de acordo com a Classificação Internacional de Doenças (CID). Se a doença for crônica, haverá uma indicação.

- Os medicamentos tem como objetivo tratar uma ou mais doenças e possuem princípios ativos, os quais se dividem em específico (como é o caso de analgésicos, antiinflamatórios) ou inespecífico (adstringentes, hidratantes, entre outros). A concentração dos princípios ativos presentes nos medicamentos serão armazenados.

- Definimos como procedimento médico todo exame, intervenção cirúrgica, intervenção clínica (administração de medicamentos), terapias (fisioterapia, acompanhamento psicológico, etc). Todo procedimento é realizado em um estabelecimento médico e pode utilizar um plano de saúde. Caso não possua plano, foi realizado pelo SUS. Os procedimentos podem ser realizados em diversos locais e podem ser analisados em um outro atendimento. A avaliação dada pelo médico com relação à qualidade do resultado do procedimento se divide em: ‘aproveitado’, ‘falso positivo’ ou ‘refazer’.

- Serão arquivados todos os procedimentos, prescrições e os resultados dos exames em forma de texto para cada paciente.

- Matriz e filiais podem ser identificadas pelo cnpj. Por isso foi utilizado o cnpj para a identificação do

estabelecimento médico.

- O usuário será restrito a utilizar no máximo um plano por vez.

- Cada atendimento pode ser dos seguintes tipos: particular, plano de saúde ou SUS. Em um atendimento uma doença pode ser diagnosticada e a cada doença diagnosticada o paciente pode checar se a doença é crônica ou não.

- Nos atendimentos serão levantados sintomas do paciente, podendo requisitar novos exames ou não. Independente da decisão tomada pelo médico a respeito da requisição de exames, tudo será registrado no histórico.

- O sistema terá suporte para armazenar informações de medicamentos tais como nome, princípio ativo e a tarja. Os medicamentos podem ter uma tarja indicando se é genérico (tarja amarela), se precisam de receita (tarja vermelha) ou se apresentam risco para o paciente (tarja preta).

- Serão oferecidas informações dos convênios e de seus planos, se fazem a cobertura de alguns exames. Os planos podem ser divididos por faixa etária e tipo (enfermaria_perfil, enfermaria_flex, apartamento_flex, apartamento_tpe, entre outros). Será imposta uma restrição de cobertura de uso (se está ou não na carência).

- Para acessar o sistema será necessário que o indivíduo possua um cadastro, fornecendo seu CPF, RG, contato (telefone de emergência, outros telefones, e-mail), endereços e o tipo sanguíneo.

- Haverá distinção entre os tipos de usuários**: pacientes e médicos.**

- No caso de médicos serão inclusas informações adicionais como nome, endereço e telefone do local em que trabalha (hospital, laboratório, etc).

- Os médicos também podem ser pacientes e podem ter especialidade ou não, como o caso do clínico geral por formação. Os médicos poderão alterar os prontuários no intuito de atualizar a evolução da enfermidade e o estado de saúde do paciente.

- Quando ocorre uma consulta, o médico pode realizar diagnóstico, fazer prescrições e/ou analisar os exames. Com base nisso, o médico receita medicamentos numa determinada dose.

- Os pacientes que possuírem acesso ao sistema poderão recuperar informações a respeito dos tratamentos

realizados para suas doenças, os diagnósticos e prescrições realizadas pelo médico que acompanhou o tratamento, além da dosagem dos medicamentos.

- Com relação às notações, usamos o BrModelo para o modelo conceitual. Os agregados foram representados de duas maneiras diferentes: usando entidades fracas ou por meio daquelas relações que estão dentro de um retângulo (losango dentro de retângulo).

==========

Descrição dos Use Cases

1) Caso de Uso: Validação

Atores: paciente, médico
Objetivo no contexto: Validar o acesso ao sistema.
Pré-condições: O ator deve possuir um cadastro no sistema.
Trigger: O ator decide consultar o sistema

Cenário:
1. O ator entra na página linux.ime.usp.br/ imecare
2. O sistema apresenta um formulário para que o ator entre com sua identificação e senha.
3. O ator digita seus dados.
4. O sistema verifica se os dados são consistentes com o que há no banco de dados.
5. Caso os dados sejam consistentes, a interação entre o usuário e o sistema é liberada.
6. Caso contrário, o sistema devolve erro e volta para o segundo item.

Prioridade: Alta prioridade
Frequência de uso: Alta frequência
Canal para o ator: Web site

Questões abertas:
1. O que acontece quando o ator tenta logar no sistema por várias vezes sem sucesso?

2. Haverá algum sistema de recuperação de identificação/senha?

2) Caso de Uso: Leitura de prontuário

Ator primário: médico, paciente
Objetivo no contexto: Ler as informações contidas no prontuário de
um paciente.

Pré-condições: Estar logado no sistema e o paciente possuir um prontuário no sistema.
Trigger: Necessidade de recuperar informações sobre determinado paciente

Cenário:
1. Ver caso de validação
2. No site, terá um campo de busca por um paciente onde o ator deverá
inserir o identificador do paciente.
3. O sistema mostra o resultado na tela.

Exceções: 1. Prontuário inexistente.
Prioridade: Alta
Frequência de uso: Alta
Canal para o ator: Website

3) Caso de Uso: Escrita de prontuário

Ator primário: médico

Objetivo no contexto: Modificar as informações contidas no prontuário de
um paciente.

Pré-condições: Estar logado no sistema e o paciente possuir um prontuário no sistema.
Trigger: Necessidade de recuperar informações sobre determinado paciente

Cenário:
1. Ver caso de validação
2. No site, terá um campo de busca por um paciente onde o ator deverá

inserir o identificador do paciente.

3. O sistema mostra o resultado na tela e os campos de edição.

Exceções: 1. O médico tenta modificar um prontuário ao qual não tem acesso.

Prioridade: Alta

Frequência de uso: Alta

Canal para o ator: Website

==========

Descrição dos requisitos

Gerenciamento dos pacientes

O sistema precisa prover uma forma dos pacientes se cadastrarem no sistema e acessarem seus dados quando fizerem login.
Os pacientes têm cpf, rg, contato (seja telefone/ endereço), tipo sanguíneo.

Um paciente pode ter uma ou mais doenças associadas a ele. Cada doença vai gerar um determinado grupo de exames pedidos. Um mesmo exame pode ser pedido para problemas de saúde diferentes e podem ser feitos em laboratórios diferentes.

Uma maneira encontrada de armazenar os dados do paciente de forma a conter seu histórico de consultas, exames e informações em geral, foi criar 6um prontuário que conterá um conjunto de registros sobre os mais diversos conteúdos relacionados ao paciente.

Assim, tudo isso poderá ser visto através do prontuário do paciente que está consultando o sistema.

Além disso, o sistema deve possibilitar consultar quais serão as datas/horários das próximas consultas e, além disso, permitir alteração das datas (caso o paciente deseje e os médicos tenham aqueles dias disponíveis para atendimento).

Gerenciamento dos funcionários
Como funcionários o sistema terá: médicos, enfermeiros e secretários. Eles têm cpf, rg, contato (seja telefone/ endereço), unidade(s) em que trabalham, já que o sistema integra mais de um hospital.

No caso do médico, outro atributo é a especialidade em que trabalha. Os médicos devem poder acessar o sistema, consultar e modificar dados dos pacientes.

Quando ocorre uma consulta, o médico pode realizar diagnóstico, fazer prescrições e/o analisar os exames. Com base nisso, o médico receita medicamentos numa determinada dose.

Os enfermeiros podem apenas consultar informações dos pacientes em que atende.

Os secretários são responsáveis pela parte de agendamento das consultas.

Descrição dos Use Cases

1)
Caso de Uso: Validação

Atores: paciente, médico, secretário, enfermeiro

Objetivo no contexto: Validar o acesso ao sistema.

Pré-condições: O ator deve possuir um cadastro no sistema.

Trigger: O ator decide consultar o sistema

Cenário:
1. O ator entra na página linux.ime.usp.br/ imecare
2. O sistema apresenta um formulário para que o ator entre com sua identificação e senha.
3. O ator digita seus dados.
4. O sistema verifica se os dados são consistentes com o que há no banco de dados.
5. Caso os dados sejam consistentes, a interação entre o usuário e o sis-

tema é liberada.
6. Caso contrário, o sistema devolve erro e volta para o segundo item.
Prioridade: Alta prioridade
Quando estará disponível: Primeiro incremento
Frequência de uso: Alta frequência
Canal para o ator: Web site
Questões abertas:
1. O que acontece quando o ator tenta logar no sistema por várias vezes sem sucesso?
2. Haverá algum sistema de recuperação de identificação/senha?

2)
Caso de Uso: Leitura de prontuário

Ator primário: médico, enfermeiro e secretário

Objetivo no contexto: Ler as informações contidas no prontuário de um paciente.

Pré-condições: Estar logado no sistema e o paciente possuir um prontuário no sistema.

Trigger: Necessidade de recuperar informações sobre determinado paciente

Cenário:
1. Ver caso de validação
2. No site, terá um campo de busca por um paciente onde o ator deverá inserir o identificador do paciente.
3. O sistema mostra o resultado na tela.
Exceções: 1. Prontuário inexistente.
Prioridade: Alta
Quando estará disponível: Segunda incrementação
Frequência de uso: Alta
Canal para o ator: Website

3)
Caso de Uso: Escrita de prontuário

Ator primário: médico, enfermeiro e secretário

Objetivo no contexto: Ler as informações contidas no prontuário de um paciente.

Pré-condições: Estar logado no sistema e o paciente possuir um prontuário no sistema.

Trigger: Necessidade de recuperar informações sobre determinado paciente
Cenário:
1. Ver caso de validação
112. No site, terá um campo de busca por um paciente onde o ator deverá inserir o identificador do paciente.
3. O sistema mostra o resultado na tela e os campos de edição.
Exceções: 1. O médico tenta modificar um prontuário ao qual não tem acesso.
Prioridade: Alta

Quando estará disponível: Segunda incrementação

Frequência de uso: Alta
Canal para o ator: Website

NoSQL – DESCRIÇÂO GERAL

Devido à complexidade deste projeto é possível pensar em inúmeros contextos onde armazenar as informações usando diferentes modelos NoSQL pode ser vantajoso.
Citaremos abaixo algumas dessas abordagens, sendo que a implementação de todas é incabível no tempo de curso.

Há algumas informações que precisam de bastante espaço e, para permitir tal escalabilidade, é necessário usar o bd NoSQL.

1. Orientado a documentos:

Por exemplo, como um mesmo paciente pode ter vários exames associados a eles, a quantidade total de exames é bem grande se pensarmos na totalidade de pessoas atendidas por todos os hospitais. Para manter tantos registros é necessário usar bd NoSQL. As prescrições também podem ser armazenadas.
Pretendemos usar JSON para fazer banco de dados orientado a documentos .

2. Orientado a colunas:

Outra abordagem para o uso de NoSQL seria BigTable para otimizar a velocidade de consultas ao acessar informações de pacientes e respectivas doenças de acordo com uma necessidade específica. O modelo orientado a colunas seria usado para evitar JOINs.
Exemplos:

- Mapeamento de regiões para identificar doenças endêmicas, utilizando informações como o endereço do paciente, data de descoberta da doença no paciente;
- Disponibilizar dados que permitam analisar estatisticamente eventos e relações entre doenças, diagnósticos, exames, medicamentos, entre outros. Para isso, poderia usar o NoSQL aliado com os dados armazenados no SQL. Um exemplo de evento seria: qual é a chance de que pessoas que conviveram com outras que tiveram catapora e não pegaram a doença, tenham ficado imunes a ela;
- Auxiliar o gerenciamento de recursos hospitalares, identificando como distribuir equipamentos de acordo com a necessidade de uso no local.

---

4 - NoSQL

Como mencionado na primeira etapa do projeto, pretendemos utilizar o banco de dados NoSQL MongoDB para poder tratar de maneira eficiente a falta de padrão dos documentos gerados nos procedimentos médicos, principalmente em exames.

Tendo em vista que a motivação do sistema é gerar um histórico de dados por paciente, a priori não é necessária a

Por que usar o MongoDB para busca/armazenamento de procedimentos:

- Definimos o procedimento como qualquer intervenção em apenas um paciente, seja ela laboratorial, cirúrgica, terapêutica e etc.
- O resultado do procedimento é composto de todos os dados quantitativos obtidos através deste procedimento. Podendo até mesmo ser um conjunto vazio, como no caso de uma cirurgia. Ex:
    - Cirurgia de redução do estômago: Neste caso o procedimento não gera resultados textuais, apenas um laudo.
    - Exame de glicemia de jejum: Este exame gera diferentes tipos de dados, como a taxa glicêmica, nível de hemoglobina glicosilada e até mesmo contabilização de glóbulos brancos, estes resultados variam bastante entre os laboratórios. É comum que estes procedimentos não possuam laudo.
    - Cirurgia de biópsia: Embora seja uma círurgia, a biópsia gera resultados e um laudo.
- O laudo de um procedimento é o parecer técnico textual dado por um responsável pela realização deste procedimento, por exemplo uma ata de cirurgia ou então uma descrição de fratura no exame de raio X.
- Tanto os resultados quanto o laudo de um procedimento não possuem um padrão, podendo variar de acordo com a semântica ou até mesmo com a apresentação dos resultados adotado pelo.

Considerações gerais:

Instalação:

sudo apt-get install mongodb

Criar novo banco de dados:

use novo

obs: só é possível enxergar o novo BD depois de inserir um dado, uma vez selecionado o banco de dados, é possível se referir diretamente a ele usando apenas db.

Apagar um banco de dados:

use novo

db.dropDatabase()

Inserir um dado:

db.novo.insert(`{"nome":"João"}`***)***

A inserção será usada para registrar os procedimentos no MongoDB. Mais pra frente mostraremos exemplos mais adequados para a nossa aplicação.

Apagando dados:

db.novo.remove(`{"nome_paciente":"exame"}`***)***

Devido à característica de armazenamento do sistema, tanto a remoção quanto a alteração de dados não faz sentido a priori, no entanto pode ser últil para corrigir eventuais dados defeituosos, ocasionados por erros de digitação, por exemplo.

Verificando os bancos existentes

show dbs

Verificando o banco atual

db;

Para criar uma coleçao:

db.createCollection("paciente");
db.createCollection("médico");
db.createCollection("exame");

Para mostrar as coleções existentes:

show collections;

Para excluir alguma coleção:

db.nome_da_colecao_drop();
db.exame.drop();

Para encontrar informações dentro de uma coleção:

db.nome_da_colecao.find({“chave”: “valor” });
db.nome_da_colecao.findOne();

Inserções relevantes:

A comunicação entre o MongoDB e o postgresql se dará com a redundância da chave primária da entidade Procedimento Realizado, desta forma o armazenamento de informações dos procedimentos terá o seguinte protótipo:

```
db.novo.insert(
{
        "data":"28/10/2015",
        "horário":"09:42:51",
        "CPF":"999.999.999/99"
        "nome_paciente":"João Eduardo Ferreira",
        "CRM":"87273",
        "cod_procedimento":"42",
        "cnpj":"999.999.999/99",
        "data_cadastro":"29/10/2015",
        "horário_cadastro":"10:42:51"
})
```

É esperado que procedimentos de mesmo código possuam informações similares, mas não idênticas, já que isso depende também do modo como esta informação é gerada no estabelecimento médico. Seguem outros exemplos completos de armazenamento:

```
db.novo.insert(
{
        "data":"28/10/2015",
        "horário":"09:43:51",
        "CPF":"999.999.888/99"
        "nome_paciente":"André Meneghelli Vale",
        "CRM":"87273",
        "cod_procedimento":"65",
        "cnpj":"999.999.999/99",
        "data_cadastro":"29/10/2015",
        "horário_cadastro":"10:42:51",
        "glicemia":40,
        "hemoglobina glicosilada":8
})
```

```
db.novo.insert(
{
        "data":"28/10/2015",
        "horário":"09:44:51",
        "CPF":"999.999.777/99"
        "nome_paciente":"Fernanda Camargo",
        "CRM":"87273",
        "cod_procedimento":"65",
        "cnpj":"999.999.999/99",
        "data_cadastro":"29/10/2015",
        "horário_cadastro":"10:42:51”,
        "glicemia":87,
        "hemoglobina glicosilada":4
})
```

```
db.novo.insert(
{
        "data":"28/10/2015",
        "horário":"09:44:51",
        "CPF":"999.999.666/99"
        "nome_paciente":"Goku Kakaroto",
        "CRM":"87273",
        "cod_procedimento":"65",
        "cnpj":"999.999.999/99",
        "data_cadastro":"29/10/2015",
        "horário_cadastro":"10:42:51”,
        "glicemia":230,
        "hemoglobina glicosilada":11
})
```

Consultas relevantes:

Uma vez que os dados tenham sido armazenados, é possível fazer consultas que seriam bastante demoradas no postgresql, por exemplo:

Buscando procedimentos constando glicemia normal:

```
db.novo.find({"glicemia":{$gte:70, $lte:110}})
```

Buscando todos os procedimentos dos dependentes do CPF 123.456.789/99:

```
db.novo.find({"CPF":”123.456.789/99”})
```

Buscando todos os procedimentos realizados no estabelecimento médico de CNPJ 9876:

```
db.novo.find({"cnpj":”9876”})
```

Fontes utilizadas:

Site para Json:

http://jsonlint.com/

Tutorial:

http://www.tutorialspoint.com/mongodb/mongodb_data_modeling.htm