Universidade de São Paulo
ICMC – Instituto de Ciências Matemáticas e de Computação

SSC0240 – Bases de Dados

Prof. Dra. Elaine Parros M. de Sousa
PAE: André Moreira Souza

Projeto – Inclusão Digital

PLATAFORMA DE ENSINO DE FUNDAMENTOS DE COMPUTAÇÃO PARA INCLUSÃO DIGITAL

Bruno Berndt Lima *12542550*
Daniel Henrique Lelis de Almeida *12543822*
Thiago Shimada *12691032*
Vinicius Kazuo Fujikawa Noguti *11803121*

São Carlos
16 de Abril de 2023

Sumário

1 Introdução

Tendo em vista o desafio de prover Inclusão Digital, principalmente para populações que não cresceram tendo acesso aos meios digitais, seja por falta de recursos e oportunidade, seja por questões temporais, propomos o desenvolvimento de uma plataforma educacional voltada para este público. A ideia é prover cursos e guias de simples acesso para guiar o uso dos meios digitais desde os primeiros fundamentos, promovendo instrução de uso, conscientização e uma futura independência digital.

Sendo assim, tomamos como inspiração uma das maiores plataformas educacionais de livre acesso hoje disponíveis, a *Khan Academy*, porém mudando seu propósito para além do mundo acadêmico e tentando enfatizar a acessibilidade àqueles que não estão habituados aos meios digitais.

2 Modelo Entidade-Relacionamento

2.1 Levantamento de Requisitos

Um usuário do sistema poderá utilizar a plataforma de forma anônima (não autenticada) ou de maneira identificável. Para que isso seja possível, um usuário pode se cadastrar de maneira simples ***(usuário autenticado)***, inserindo apenas um **nome de usuário** (3 a 16 caracteres), uma **senha** (que será armazenada como uma hash, estimando 144 caracteres necessários para sua persistência) e, opcionalmente, um **e-mail** (ocupando no máximo 254 caracteres). Além disso, para moderar a plataforma e atualizá-la com mais conteúdos, teremos também ***usuários administrativos***, esses são semelhante aos usuários convencionais, sendo diferenciados por um booleano **admin**, além disso todo administrador deve obrigatoriamente conter um e-mail cadastrado que será utilizado para autenticação em dois fatores.

Um *administrador* pode banir um usuário da plataforma, quando isso ocorre temos a entidade de ***banimento***, nela são indicados: o **responsável** (referência ao administrador) responsável pelo banimento, o **usuário banido** (referência ao usuário banido), o **horário** (data e hora de criação), a **validade** (data e hora em que o banimento expira) e, opcionalmente, a **causa** (texto) do banimento. Administradores não podem ser banidos.

A ideia é que esse seja um projeto dirigido e voltado para a comunidade, por conta disso é bastante importante que usuários consigam prover ***feedbacks***, sugerindo melhorias, apontando problemas encontrados e solicitando novos tópicos. Esses feedbacks, então, terão uma referência de qual é o **autor** (referência ao usuário que criou o feedback), o **horário** (data e hora de criação do feedback) a **mensagem** (texto de tamanho dinâmico, limitado arbitrariamente à 4096 caracteres) e, opcionalmente, a **data de fechamento** (data e hora quando o feedback foi fechado). Além disso, a fim de garantir a visualização dos feedbacks, temos uma entidade de ***visualização de feedback***, ela associa um **leitor** (referência ao usuário administrador) e um **feedback** (referência ao feedback) junto à informação da **data de visualização** (data e hora quando a visualização foi feita).

A fim de organizar o conteúdo, seguiremos uma divisão hierárquica bastante comum, seguindo em: categorias, tópicos, unidades e conteúdos, do maior para o menor, respectivamente. Primeiramente, teremos a ***categoria***, sendo o nível mais alto da hierarquia, ela é bastante simples, tendo apenas um **nome** (texto), uma **descrição** (texto) e um **ícone** (url). Em sequência, temos os ***cursos***, que contém um **nome** (texto), uma **descrição** (texto) e a **categoria** (referência à categoria). Seguindo adiante, teremos as ***unidades***, nosso próximo nível na hierarquia. As unidades seguem uma estrutura similar, contendo um **nome** (texto), uma **descrição** (texto), sua **ordem** (inteiro), que é usada para controlar a ordem em que as unidades são apresentadas, e o **curso** (referência à curso) a qual pertence. Por último temos os ***tópicos***, eles são o nível mais baixo na hierarquia organizacional, abrigando os conteúdos em si. Os tópicos seguem praticamente a mesma estrutura das unidades, ou seja, possuem um **nome** (texto), uma **descrição** (texto), uma **ordem** (inteiro) e, por fim, a **unidade** (referência à unidade) a qual pertencem.

Um dos principais elementos da plataforma, que seria a linha final desta hierarquia, são os

conteúdos, todos os conteúdos terão alguns campos em comum: um **título** (texto), um **subtítulo** (texto), uma **duração estimada** (inteiro indicando o número de minutos) e o **tópico** (referência ao tópico) a qual pertence. Além disso, inicialmente, teremos três tipos de conteúdo: ***artigos***, que seriam conteúdos primariamente textuais, contendo, além dos comuns entre os conteúdos, o **corpo** do artigo (que será um texto formatado em MD). Teremos também, os ***vídeos***, que contém a **url** do vídeo (texto) e uma **descrição** (texto) do vídeo. Por fim, teremos os ***exercícios***, eles são compostos de um **enunciado (corpo)** (texto formato em MD) e o **limite de seleções de alternativas** (inteiro indicando quantas alternativas um usuário pode selecionar). Além disso, os *exercícios* possuem um número variável de alternativas associadas, cada ***alternativa*** consiste de um **corpo** (texto formatado em MD), opcionalmente, uma **explicação** (texto formatado em MD), se é uma alternativa **correta** (booleano) e, por fim, o **exercício** (referência ao exercício) que faz parte.

Para *usuários autenticados*, será mantido um histórico do progresso dele na plataforma, isto é: para cada *conteúdo*, será registrado um ***evento de progresso***, indicando o **horário** (data e hora) onde foi feito o progresso, qual o **conteúdo** (referência ao conteúdo) que foi visitado e, caso o conteúdo seja um *exercício*, as **alternativas selecionadas** (referências às alternativas). Importante notar que as alternativas selecionadas precisam fazer parte do conteúdo do tipo exercício referente ao evento. Os eventos de progresso podem ser visualizados pelo usuário e podem existir mais de um evento para o mesmo conteúdo. O usuário pode deletar eventos antigos a fim de reiniciar o seu progresso de acordo com a granularidade desejada.

Além disso, *usuários autenticados*, podem deixar comentários em conteúdos, cada ***comentário*** é composto por um **corpo** (texto em MD), o **autor** (referência ao usuário que o criou), o **conteúdo** (referência ao conteúdo) em que o comentário foi feito, o **horário** (data e hora de criação) em que o horário foi feito, se ele é **visível** (booleano indicando se ele é exibido) para casos de deleção e, por fim, um comentário pode ser uma resposta, nesses casos temos o campo de **comentário pai** (referência opcional a um comentário) que indica qual comentário aquele responde. A presença desse mecanismo de respostas cria uma restrição onde precisamos garantir que, caso o comentário seja uma resposta, o conteúdo associado ao comentário pai seja o mesmo que o conteúdo referenciado presente em si. Além disso, a fim de evitar ciclos de relacionamento, precisamos garantir que, ao ir seguindo a cadeia de comentários pai, não haja a existência de um ciclo no momento da criação.

A fim de garantir a moderação da plataforma, *usuários autenticados* podem reportar comentários, para isso temos a entidade de ***report***. Cada report consiste de um **autor** (referência ao usuário que fez o report), o **comentário reportado** (referência ao comentário sendo reportado), o **horário** (data e hora de criação), o **motivo** (texto) e, por fim, o indicador de **verificado** (booleano), que indica que aquele report já foi avaliado por um administrador.

2.2 Funcionalidades

Analisando por cima dos atores do sistema, conseguimos sintetizar as funcionalidades da plataforma da seguinte forma:

- **Usuário Não Cadastrado**
  - Cadastrar-se
  - Navegar pelas categorias, cursos, tópicos, unidades e conteúdos
  - Consumir os conteúdos
  - Realizar exercícios
  - Visualizar comentários de um conteúdo
- **Usuário Cadastrado**
  - *(Todas de um Usuário Não Cadastrado)*
  - Autenticar-se
  - Alterar senha
  - Atualizar e-mail
  - Registrar (automaticamente) o progresso na plataforma
  - Visualizar o progresso atual
  - Reiniciar o progresso (de maneira granular)
  - Criar comentários em um conteúdo (caso não esteja banido)
  - Responder comentários existentes (caso não esteja banido)
  - Reportar um comentário (caso não esteja banido)
  - Deletar um próprio comentário
  - Enviar feedbacks (caso não esteja banido)
- **Usuário Administrador**
  - *(Todas de um Usuário Cadastrado)*
  - Inserir, remover e atualizar: categorias, cursos, tópicos, unidades, artigos, aulas, exercícios e alternativas
  - Deletar e atualizar comentários (de outros usuários)
  - Banir e perdoar usuários cadastrados (que não sejam administradores)
  - Visualizar os reports e comentários associados
  - Marcar reports de um comentário como resolvido (todos até o momento atual)
  - Visualizar feedbacks (histórico de visualização persistido)
  - Marcar (ou desmarcar) feedback como resolvido (mantendo a data de resolução)

2.3 Diagrama do Modelo Entidade-Relacionamento

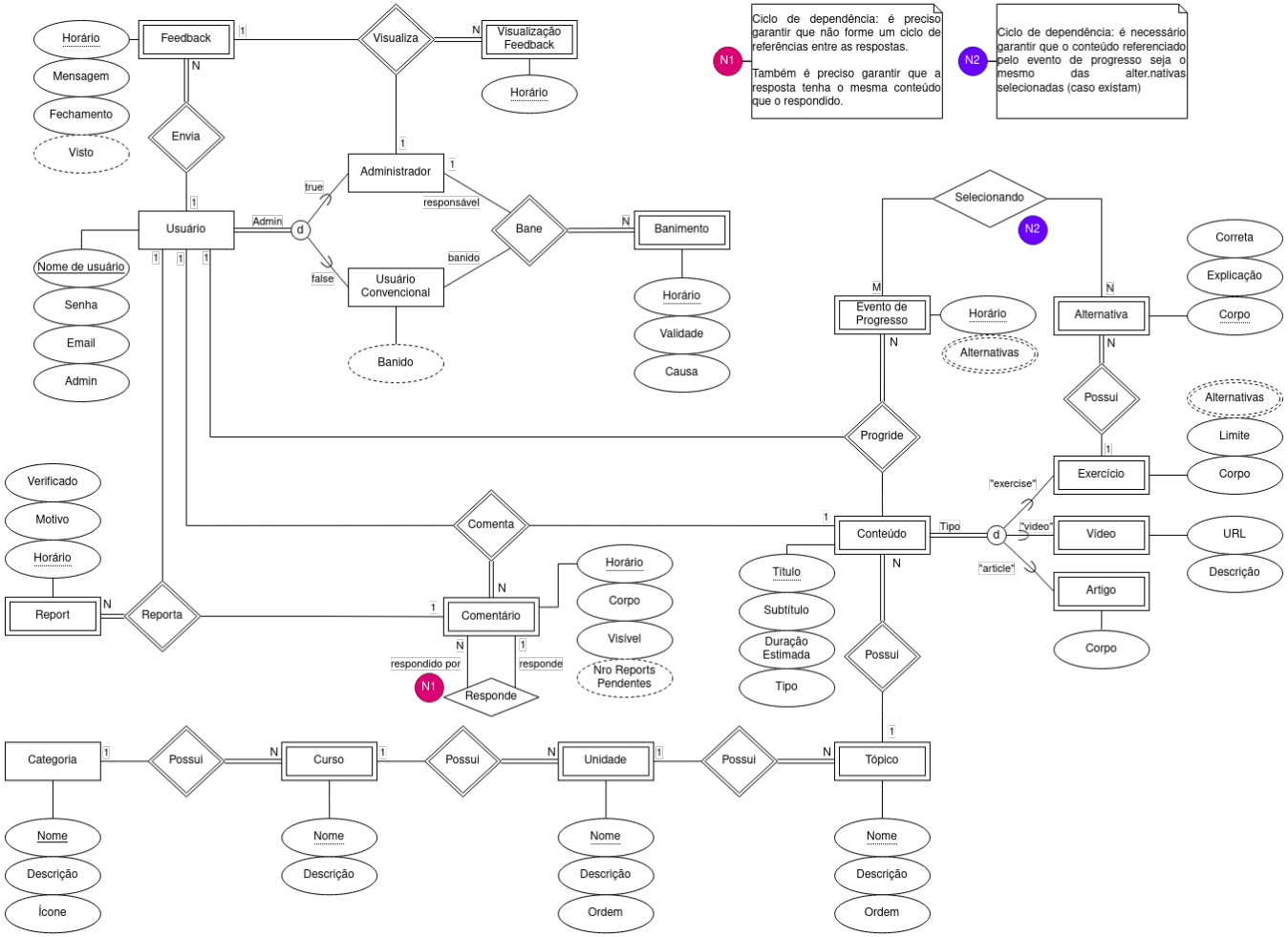

Figura 1 – Diagrama do Modelo Entidade-Relacionamento

2.4 Ciclos e Restrições de Integridade

A modelo proposto apresenta alguns ciclos naturais da solução e restrições de integridade necessárias, trataremos-as individualmente a seguir:

2.4.1 Ciclo: Comentário (Resposta) $\rightarrow$ Comentário (Respondido)

Este é um dos ciclos mais complicados do diagrama, ele se trata de um ciclo de dependência natural da modelagem, para solucioná-lo precisaremos garantir que duas restrições sejam satisfeitas:

1. Ao criar uma resposta, precisamos garantir que a referência de conteúdo do comentário respondido seja a mesma da resposta sendo criada.
2. Ao criar uma resposta, precisamos garantir que, ao seguir a cadeia de comentários pais, não haja repetição de comentários (evitar ciclos, ex: $C_1 \rightarrow\rightarrow C_2 \rightarrow C_3 \rightarrow C1$). É possível garantir isso ao impedir que o campo de comentário respondido seja editável.

2.4.2 Ciclo: Usuário $\rightarrow$ Comentário $\rightarrow$ Report

Este é um ciclo natural da modelagem, não é necessário nenhum tratamento especial, já que apenas indica a possibilidade (irônica) de um usuário reportar o próprio comentário.

2.4.3 Ciclo: Administrador $\rightarrow$ Feedback $\rightarrow$ Visualização Feedback

Mais um ciclo natural do diagrama, no sistema optamos por não restringir o administrador de fazer a maior parte das interações que um usuário convencional faria, sendo assim, caso um administrador crie um feedback, existe a possibilidade de que ele visualize o próprio feedback. Não é necessário nenhum tratamento adicional neste caso.

2.4.4 Ciclo: Conteúdo $\rightarrow$ Evento de Progresso $\rightarrow$ Alternativa

Este é um ciclo natural do diagrama, ele até poderia ser evitado por meio de uma especialização do Evento de Progresso, mas acreditamos que isso criaria uma complexidade desnecessária na modelagem sem benefícios práticos. Consistindo de um ciclo de dependência onde um Evento de Progresso, pode, opcionalmente, ter referências a alternativas selecionadas. Quando isso acontece, é necessário garantir que o conteúdo que a alternativa referencia (do tipo Exercício) seja o mesmo que é referenciado pelo Evento de Progresso, originando, então, uma restrição de integridade para que no momento de cadastro das alternativas associadas ao evento de progresso, seja verificada essa igualdade dos conteúdos referenciados.

2.4.5 Restrição adicional: Administradores precisam de um E-mail cadastrado

Para que um usuário se torne um administrador (campo admin igual a true), é necessário que seu campo de email não seja nulo, criando assim a necessidade de uma restrição de integridade adicional.

2.4.6 Restrição adicional: Administradores não podem ser banidos

Usuários administradores não podem ser banidos, por conta disso é necessário uma restrição de integridade que garanta que usuários banidos não se tornem administradores e que banimentos não sejam criados para administradores. É possível evitar isso também ao impedir que usuários existentes não se tornem administradores e nem administradores tornem-se usuários convencionais, mas não pretendemos seguir por este caminho.

Referências

ELMASRI, R.; NAVATHE, S. B. *Fundamentals of Database Systems*. 7. ed. [S.l.]: Pearson, 2021.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Database Systems Concepts*. 7. ed. [S.l.]: McGraw-Hill Education, 2020.