

Universidade Federal de Alagoas
Curso de Ciência da Computação

Projeto de Engenharia de Software 1 (1ª Etapa)
Classmates Forum

Francisco Vieira, Luís Eduardo, Roque Barbosa

Índice

1. Introdução.....	3
2. Requisitos.....	3
2.1. Elicitação dos requisitos.....	3
2.2. Requisitos funcionais.....	5
2.3. Requisitos não funcionais.....	7
3. Projeto.....	8
3.1. Projeto orientado a objetos.....	8
3.2. Projeto de banco de dados.....	10
4. Planejamento de entregas	11

1. Introdução

Classmates Forum é um web site para que colegas de sala (turma) possam ter discussões sobre os conteúdos que estão aprendendo dentro de sua instituição ou sobre temas além do que se vê na instituição. Ou seja, tem como objetivo integrar mais os estudantes a participar de uma comunidade de conhecimento em comum e para partilhar conhecimentos e experiências. É um sistema para estudantes de uma turma e também administrado por estudantes dessa mesma turma. Nesse sistema haverão tópicos, criados por administradores, que representam um tema em comum a ser discutido, por exemplo: Java, Matemática Discreta, Engenharia de Software 1 etc. E dentro desses tópicos os usuários poderão criar postagens, que representam dúvidas ou algum conhecimento que deseja compartilhar, e dentro dessas postagens os usuários podem criar discussões através de comentários e dar feedbacks sobre os conteúdos das postagens.

2. Requisitos

2.1. Elicitação dos requisitos

Entrevista

- (1) O que você acha sobre os alunos gerenciarem os suas próprias postagens a respeito de um tópico?

Acredito ser uma excelente abordagem, pois estende a discussão sobre temas interessantes e edificantes, do mundo físico para o mundo virtual, permitindo assim um público maior, trazendo experiências e diversos pontos de vista.

- (2) Você acha necessário ter um controle de conteúdo com relação as postagens feitas?

Acho necessário para que exista uma conduta a se seguir, mantendo a ordem na plataforma e a boa relação entre os alunos.

- (3) Com relação a forma de como os tópicos são dispostos, você tem alguma preferência por ordenação, ou isso não influencia na usabilidade?

Acredito que a implementação dessa função pode ajudar o usuário a encontrar conteúdos mais relevantes de acordo com seus filtros, acho válido a ideia.

- (4) Você optaria por uma função de busca?

Sim, pois permitiria ao usuário um maior desempenho no uso da plataforma.

- (5) Nos comentários de cada postagem, você acredita que seria interessante responder a uma pergunta marcando o usuário que perguntou?

Uma ótima ideia.

- (6) O que você acha sobre ser notificado quando algum tópico for fechado?

Acho uma ideia válida, quando um determinado tópico não está sendo muito usado é bom notificar as pessoas que estão/estavam usando este tópico em suas postagens para serem avisadas

- (7) Sobre avaliação de um comentário em uma postagem, você acha interessante ter um contador de votos para classificar o comentário como “bom” ou “ruim”?

É uma boa funcionalidade, pois pode ajudar a encontrar os comentários que foram mais informativos ou que ajudou em um determinado problema.

- (8) Você acha interessante que outros alunos consigam ter acesso ao seu perfil e consigam visualizar suas informações acadêmicas?

Acredito ser uma boa ideia, pois pode ter acesso as informações sobre as atividades atuadas pelos outros alunos e verificar a quais tópicos e postagens estão relacionados.

2.2. Requisitos funcionais

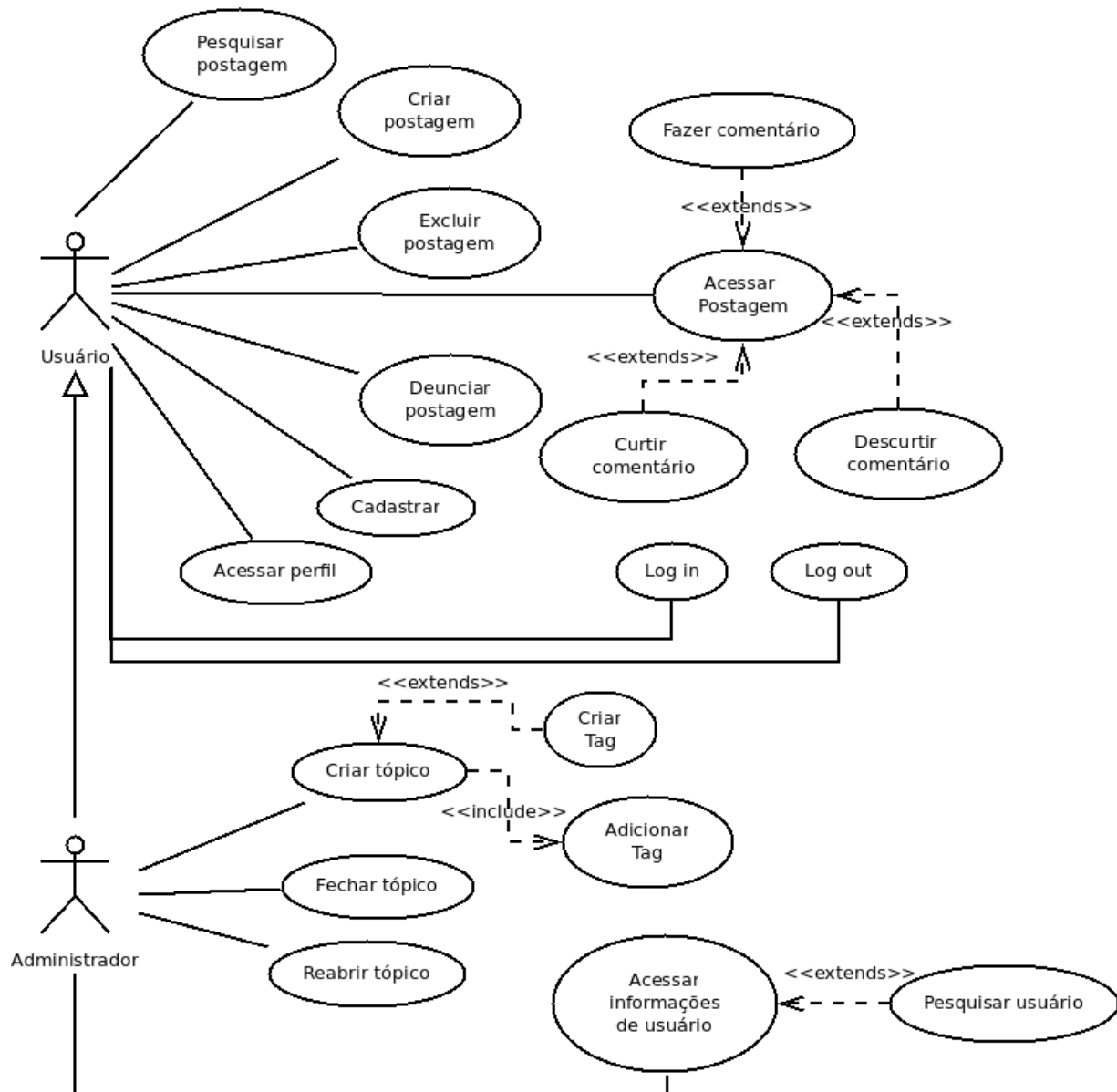


Fig. 1 – Diagrama de casos de uso – Funcionalidades do sistema através de dois atores: usuário e administrador

Caso de uso:	Criar tópico
Ator(es):	Administrador
Pré-condições:	É necessário acessar o formulário de criação de tópico através de um botão que irá estar posicionado na página inicial do site, em que somente os usuários que tiverem o privilégio de administrador podem visualizar.
Fluxo padrão:	No formulário de criação do tópico irá aparecer um campo para inserir o nome do tópico, abaixo haverá uma opção para criar tags, <<extends>> Criar tag, e do lado direito opções de seleção de tags já criadas ou que existem por padrão, <<include>> Selecionar tag, mais abaixo irá ter outro campo que será destinado a descrição do que o tópico se trata. Por fim, irá

	haver um botão que confirma a criação do tópico e irá redirecionar o administrador para a página inicial.
Fluxo de exceção 1:	Criar tópico com o mesmo nome ou com um nome vazio
Fluxo de exceção 2:	Tentativa de criar tópico sem tags
Pós-condições:	Depois de redirecionado para a página inicial, o tópico criado estará listado na página, com o nome do tópico, suas tags e o nome do administrador que o criou. E este tópico criado se transforma em um link para a página deste tópico.
Prioridade:	Desejável – Necessária – Imprescindível

Caso de uso:	Criar tópico
Fluxo de exceção 1:	Criar tópico com o mesmo nome ou com um nome vazio
Descrição:	Na tentativa de criar um tópico com o mesmo nome que outro tópico existente será considerado redundância e irá se lançado uma mensagem de erro. Nomes de tópicos podem ter caracteres e números, porém não pode estar vazio, e será enviado uma mensagem de erro caso aconteça.
Fluxo de exceção 2:	Tentativa de criar tópico sem tags
Descrição:	Tags são atributos que descrevem os tipos de postagens contidas dentro de um tópico, ex: Programação, Java, Estrutura de Dados, etc, que podem ser criadas, <<extends>> Criar tag, pelos administradores ou selecionar as que já existem por padrão ou previamente criadas, <<include>> Selecionar tag. Um tópico para ser criado necessita de pelo menos uma tag associada, se na tentativa de criar um tópico sem tag irá ser lançado uma mensagem de erro.

Caso de uso:	Criar postagem
Ator(es):	Usuário, Administrador
Pré-condições:	Deve existir um tópico na qual a postagem será criada
Fluxo padrão:	Postagens serão criadas dentro de tópicos. Então, dentro do tópico selecionado terá a opção (botão) de criação de postagem em que irá redirecionar para o formulário de criação de postagem, neste formulário terá os campos de: nome da postagem e descrição, e terá um botão para confirmação da criação da postagem.
Pós-condições:	Após a postagem ser criada, a página será redirecionada para a página do tópico a qual a postagem foi criada, e a postagem irá aparecer listada como um link com o nome da postagem e o nome do usuário que publicou, que ao ser acessado (esse link), será redirecionado para a página desta postagem.
Prioridade:	Desejável – Necessária – Imprescindível

Caso de uso:	Acessar informações de usuário
---------------------	--------------------------------

Ator(es):	Administrador
Pré-condições:	Para acessar o campo de pesquisa necessita estar na página inicial
Fluxo padrão:	O administrador pode acessar as informações de outro usuário através de dois meios: inserindo o nome do usuário na rota do site e sendo redirecionado para a página de usuário ou através do campo de busca, <<extends>> Pesquisar usuário, que dará acesso a um link de redirecionamento para a página do usuário.
Fluxo de exceção 1:	Usuário não existe
Pós-condições:	Após ser redirecionado para a página de usuário, irá ter acesso a todos os dados públicos daquele usuário.
Prioridade:	Desejável – Necessária – Imprescindível

Caso de uso:	Acessar informações de usuário
Fluxo de exceção 1:	Usuário não existe
Descrição:	Ao colocar um nome de um usuário que não existe no campo de pesquisa, irá retornar uma mensagem dizendo que “usuário não foi encontrado”, caso a busca seja na inserção do nome do usuário na rota do site, a página será redirecionada para uma “page not found”.

2.3. Requisitos não funcionais

- **Usabilidade:**

- **RFN-1** - Ordenar postagens por hiperparâmetros

Meta: Disponibilizar as postagens ordenada conforme a disponibilidade dos hiperparâmetros

Verificação: Se a quantidade de comentários em uma postagem for maior do que em outras e se nessa postagem apresenta um comentário verificado (de melhor comentário) são propriedades que agregam valor a essa postagem para a ordenação.

- **RFN-2** - Mostrar os resultados semelhantes em caso de busca mal sucedida

Meta: Retornar resultados semelhantes quando inserido uma busca que não existe

Verificação: Caso a busca não corresponda a algo exato no banco, é retornado o que possua uma semelhança de acordo com o que foi requisitado. Esse requisito é verificado se as informações retornadas apresentarem semelhança com o que foi pesquisado.

- **RFN-3** - Adicionar rótulo de melhor comentário

Meta: Marcar o comentário como a melhor daquela postagem.

Verificação: A verificação do comentário será feita conforme a quantidade de curtidas, se a quantidade de curtidas for superior a 10 e for maior que as dos outros comentários.

3. Projeto

3.1. Projeto orientado a objetos

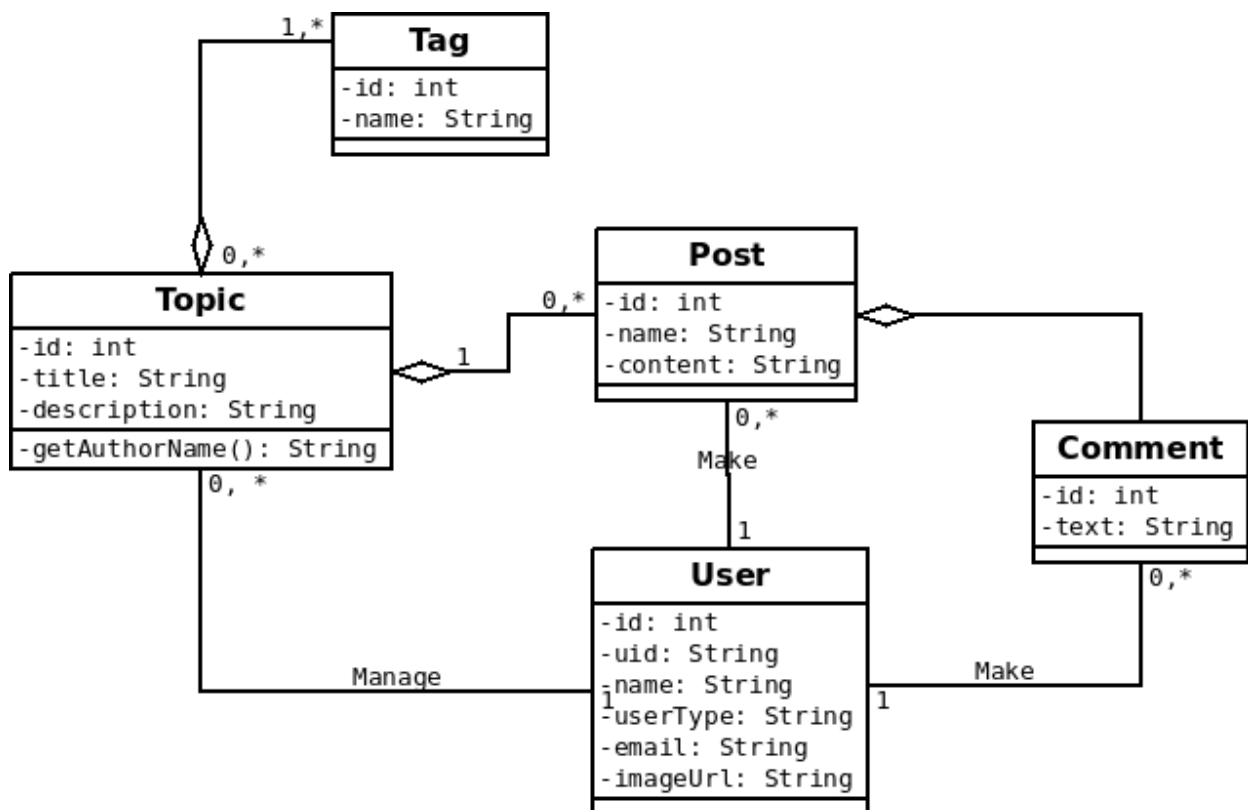


Fig. 2 – Diagrama de Classe

No diagrama de classe acima pode-se ver quais são as classes envolvidas no sistema e os relacionamentos entre elas: User é quem manipula os tópicos (Topic), e nesses tópicos existem postagens (Post), e essas postagens são manipuladas pelos usuários (User), que por sua vez podem criar postagens e comentá-las (Comment).

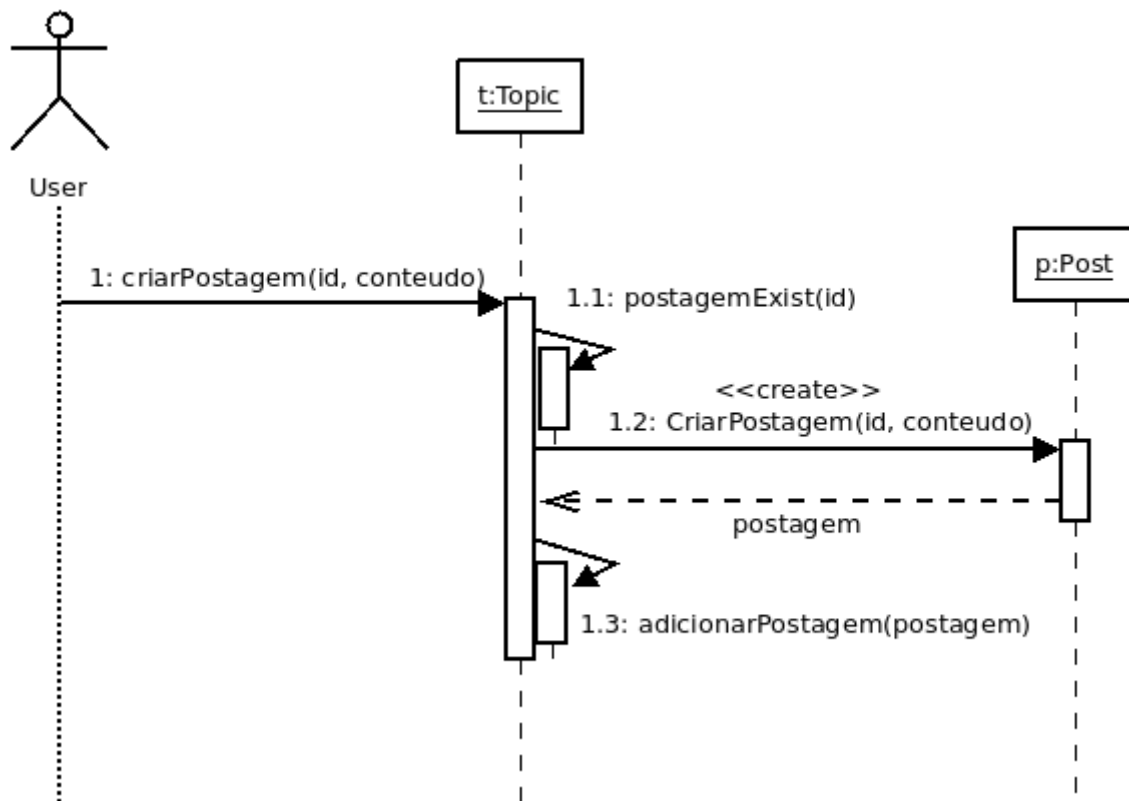


Fig. 3 – Diagrama de Sequência

O User pode criar uma postagem dentro de um tópico chamando o método `criarPostagem()` em um objeto de Topic, que irá verificar se já existe uma postagem com o id que foi passado, `postagemExist()`, caso não exista, um objeto do tipo Post irá ser criado, e posteriormente adicionado à lista de postagens do Topic através do método `adicionarPostagem()`.

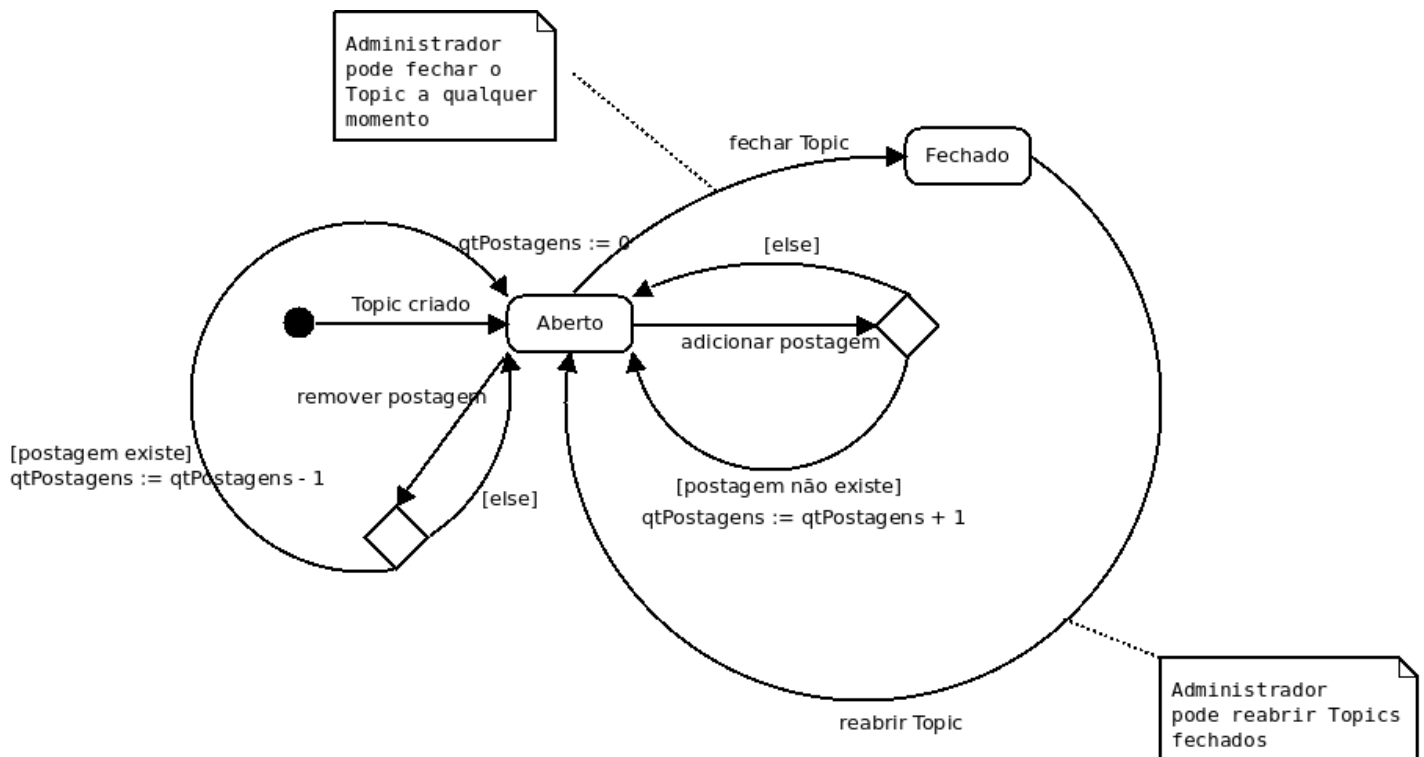


Fig. 4 – Diagrama de Estado

Em um Topic existem 2 estados, aberto e fechado, assim que o objeto é criado ele se encontra aberto, e todo usuário que tiver privilégio de administrador poderá mudar o estado do objeto para fechado, ou vice-versa. Nas transições de adicionar e remover postagens sempre parte do estado de aberto e retorna para o estado de aberto.

3.3 Projeto de banco de dados

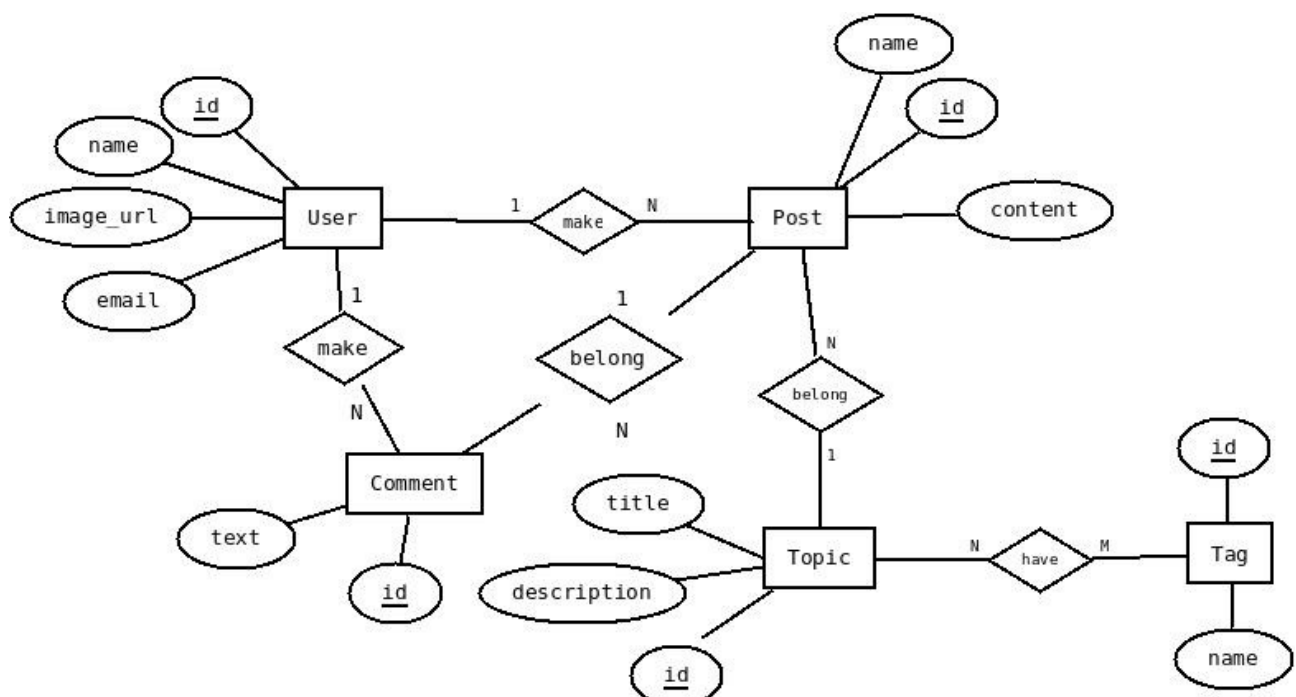


Fig. 5 – Diagrama de Entidade-Relacionamento

No modelo conceitual (ER) mostrado na imagem acima, mostra quais as entidades, atributos e relacionamentos envolvidos no banco de dados do sistema, que posteriormente irão virar tabelas, dentre elas vão existir: User, Post, Topic, Tag e Comment com seus respectivos relacionamentos, junto com seus atributos que compõem o esquema da tabela.

O Post vai estar ligado ao User que o criou, como também o Topic que ele está inserido. O Comment irá guardar o User que comentou e o respectivo Post. O relacionamento entre Tag e Topic irá virar uma tabela que vai referenciar ambos.

4. Planejamento de entregas

Iteração 1 (Versão base) – Mapeamento do modelo conceitual (ER) para o banco de dados; implementação das entidades (classes), seus atributos, métodos e relacionamentos; página inicial; formulário de registro dos usuários; login do usuário; formulário de criação de tópico; formulário de criação de postagem dentro de um tópico; página dos tópicos; página das postagens.

Iteração 2 (incremento 1) – Comentários nas postagens; curtir e descurtir comentários; página de usuário; campo de busca; corrigir erros e implementar mudanças.