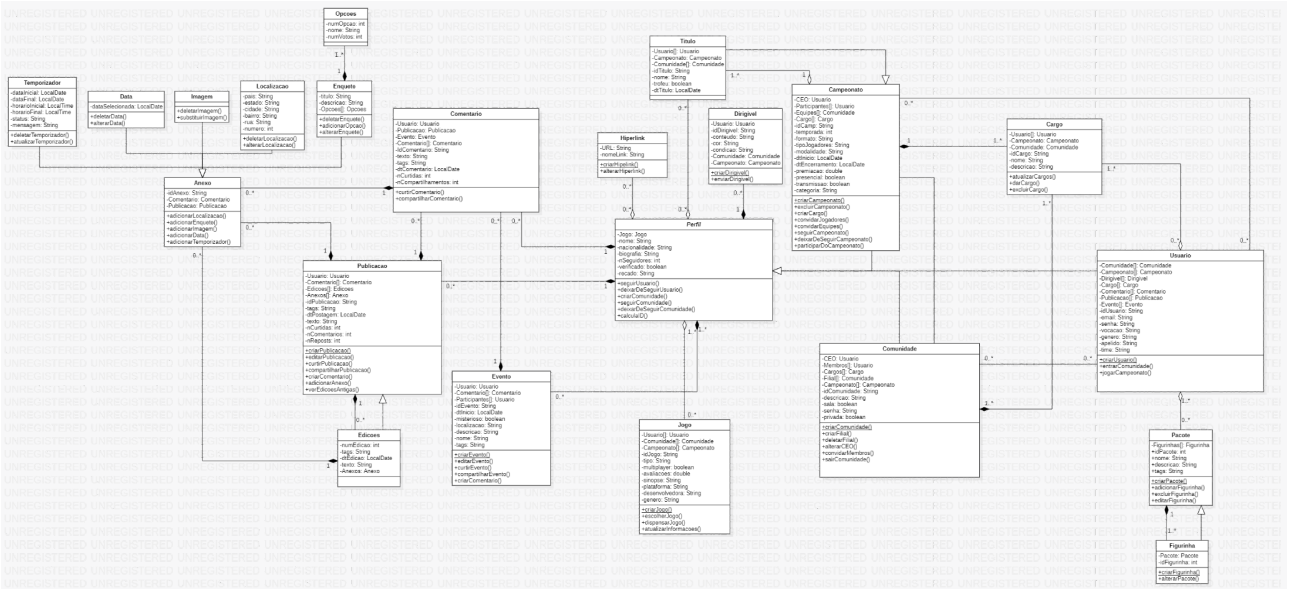
TRABALHO INTERDISCIPLINAR

Banco de Dados, Programação Orientada a Objetos e Desenvolvimentos de Aplicações Web

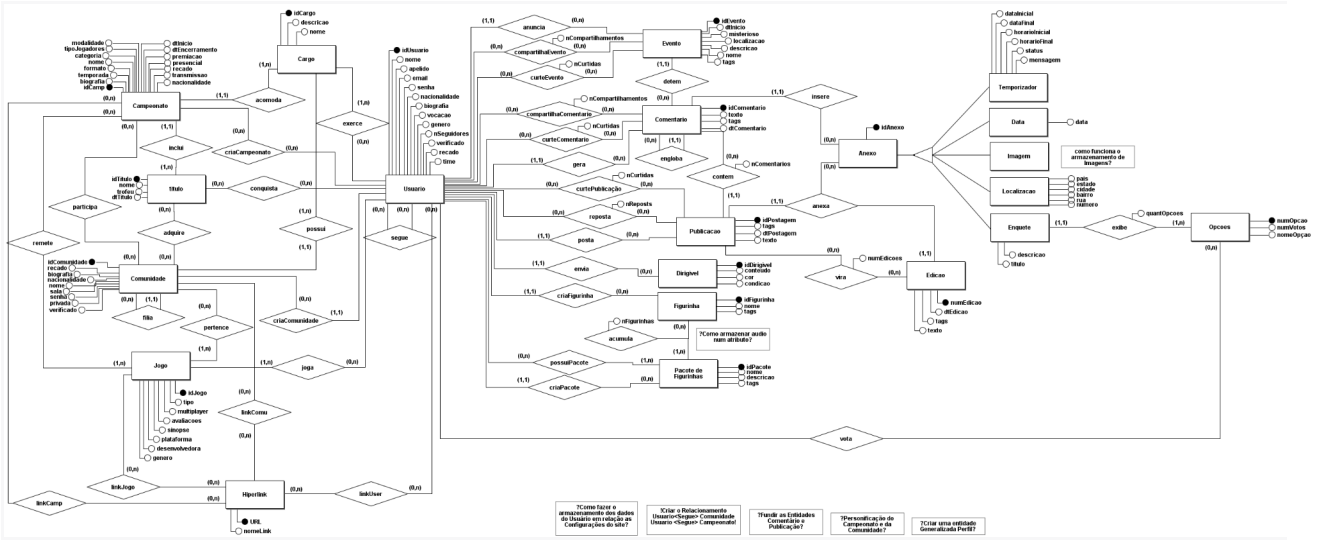
Alunos: Nathália Campos Lessa e Renan Felliphe de Moura Diogo Silva - 2° Informática.

IMAGEM DO DIAGRAMA(UML):



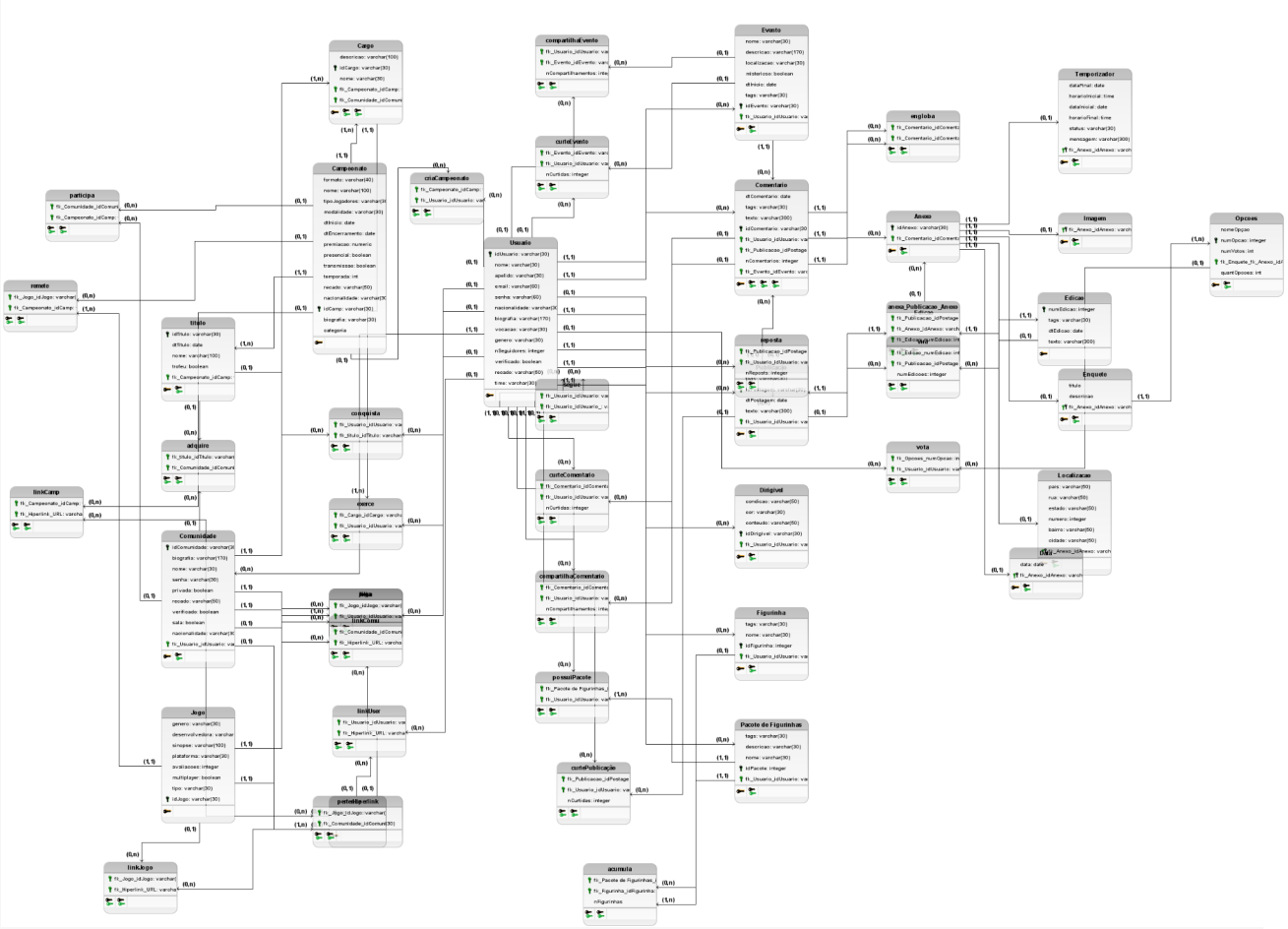
(Imagem meramente representativa. Imagem original disponibilizada na pasta do projeto)

IMAGEM DO DIAGRAMA(DER):



(Imagem meramente representativa. Imagem original disponibilizada na pasta do projeto)

IMAGEM DO DIAGRAMA(DTR):



(Imagem meramente representativa. Imagem original disponibilizada na pasta do projeto)

DESCRIÇÃO GERAL DO DIAGRAMAS:

OBS: *Os três diagramas, UML (Unified Modeling Language), DER (Diagrama de Entidade-Relacionamento) e DTR (Diagrama de Tabelas de Relacionamento), foram criados pelos alunos Renan Felliphe e Nathália Lessa de maneira espelhada para oferecer uma compreensão completa e geral do sistema. Assim, o resumo em seguida pode ser usado para representar todos os três modelos feitos para o projeto Bonfire.*

O Bonfire é uma plataforma de mídia social focada principalmente na interação entre os perfis de seus usuários. No Bonfire, existem três tipos diferentes de perfis: Usuário, Campeonato e Comunidade. Ao criar um perfil, é necessário especificar o tipo, portanto não é possível criar um perfil genérico, apenas suas variantes.

Para validar um perfil no Bonfire, um cliente, seja ele um espectador, jogador ou outro, deve obrigatoriamente se relacionar com um ou vários jogos previamente registrados no site. No entanto, um jogo não precisa necessariamente ter um perfil relacionado a ele. Essa interação entre perfil e jogo é herdada por todas as subclasses(Especializações).

Um perfil, manipulado por um cliente, pode criar quantas publicações e/ou eventos quiser. Tanto as publicações quanto os eventos podem incluir nenhum ou vários comentários. Além disso, ambos podem incorporar anexos como localização, data, imagem, temporizador e até uma enquete com ilimitadas opções para votação.

Quando alguns atributos específicos de uma publicação são atualizados por um cliente, os dados antigos são armazenados como edições da mesma publicação. Assim, desde que a publicação exista para ser editada, ela poderá ter um número ilimitado de edições.

Um cliente também pode possuir vários títulos e/ou dirigíveis. No entanto, há uma diferença na relação entre eles. Se um cliente deixar de existir, os títulos continuarão a existir e poderão continuar a ser exibidos em outros perfis, enquanto os dirigíveis serão deletados. Além disso, a relação entre título e campeonato é de pertencimento(1, 1), pois um título pertence a apenas um campeonato, enquanto a relação de título com comunidade e usuário é de vitória ou conquista(0…*). Um título herdará alguns atributos do campeonato ao qual pertence.

Tanto em um perfil do tipo comunidade quanto em um perfil do tipo campeonato, podem existir diversos cargos que poderão ser atribuídos aos participantes. Ambas as instâncias podem armazenar vários cargos com funções diferentes, mas um cargo só pode existir se a comunidade ou o campeonato ao qual pertence também existir. Ao criar um campeonato ou uma comunidade também são criados 2 cargos padrões iniciais, o de CEO, criador do perfil, e o de participantes, pessoas que serão convidadas ao perfil.

Um cliente pode apresentar em seu perfil inúmeros links (hiperlinks) para diferentes endereços, e da mesma forma, um link também pode estar agregado em vários perfis diferentes.

Por fim, exclusivamente, um usuário pode criar pacotes de figurinhas ou figurinhas individuais diretamente. No entanto, uma figurinha não pode existir sem um pacote, e um

pacote pode conter várias figurinhas ao mesmo tempo. Deste modo, sempre que uma figurinha for criada, ela deve ser armazenada em seu pacote de origem. Uma figurinha herdará alguns atributos do pacote ao qual pertence.

OBSERVAÇÕES SOBRE O DIAGRAMA DE CLASSES:

1. Os métodos:

+criarPublicacao()	+criarJogo()	+criarFigurinha()
+criarEvento()	+criarComunidade()	+criarHiperlink()
+criarDirigivel()	+criarUsuario()	
+criarCampeonato()	+criarPacote()	

Estão representados como métodos estáticos(static), pois os objetos em específico, serão criados dentro do métodos. Ex de implementação:

```
public class Publicacao {
    public static void criarPublicacao() {
        Publicacao p = new Publicacao();
        }
    }
```

2. A classe “Perfil” foi definida como abstrata(abstract) pois a criação de uma instância direta dessa classe genérica não deve ser permitida, apenas a criação de uma comunidade, um campeonato ou um usuário.

3. Todas as relações de associação, agregação e composição para com a classe “Perfil” devem ser herdadas para as suas subclasses.

4. As classes “Edição” e ”Figurinha”, respectivamente, possuem dupla relação com suas classes mãe, “Publicação” e ”Pacote”, pois elas devem herdar todos seus atributos e métodos e também não podem existir sem a suas classes mãe.

5. Foram opcionalmente ocultados os métodos especiais getters e setters, além dos métodos construtores pela grande quantidade de atributos de cada classe. Caso contrário, apenas serviria para aumentar o tamanho do diagrama.

OBSERVAÇÕES SOBRE O DIAGRAMA DE ENTIDADE-RELACIONAMENTO:

1. A entidade “Imagem” se encontra sem nenhum atributo representado pois o seu atributo identificador será emprestado da classe generalizada “Anexo”, enquanto que não foi encontrado um método de representar o armazenamento de imagens em atributos para esse caso além de um tipo de dados chamado “Blob” em SQL.

2. No diagrama foram deixadas pelos alunos, algumas legendas para representar algumas dúvidas que tivemos e alguns elementos que poderão vir a ser acrescentados ou retirados nos próximos bimestres.

3. No diagrama apresentado, ainda existem alguns erros de normalização que ainda não foram corrigidos, como o atributo "Temporada" na entidade "Campeonato". Esse atributo representa as edições que um campeonato possui. No entanto, se um campeonato tiver mais de uma edição, o atributo "Temporada" se tornará multivalorado, o que viola as regras da primeira forma normal: “Nenhum atributo composto ou multivalorado”.

*/*As imagens apresentadas no documento também estão disponíveis para a visualização como arquivo na pasta.*/*