

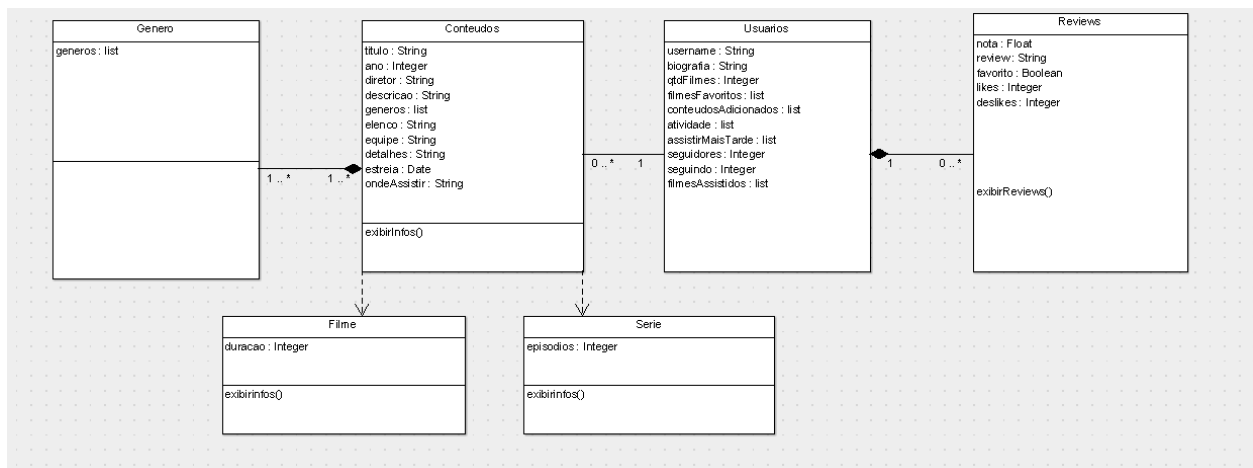
AP2 Programação Orientada a Objetos por Julia Siodaro e Jeronimo Chiclana

Escolhemos como nosso projeto um site crowdsourcing em que os usuários avaliam e publicam filmes, com o objetivo de incentivar os usuários a assistirem filmes e series novas e dividir seus gostos e opiniões com outros usuários.

1. Começo: Diagrama de Classes.

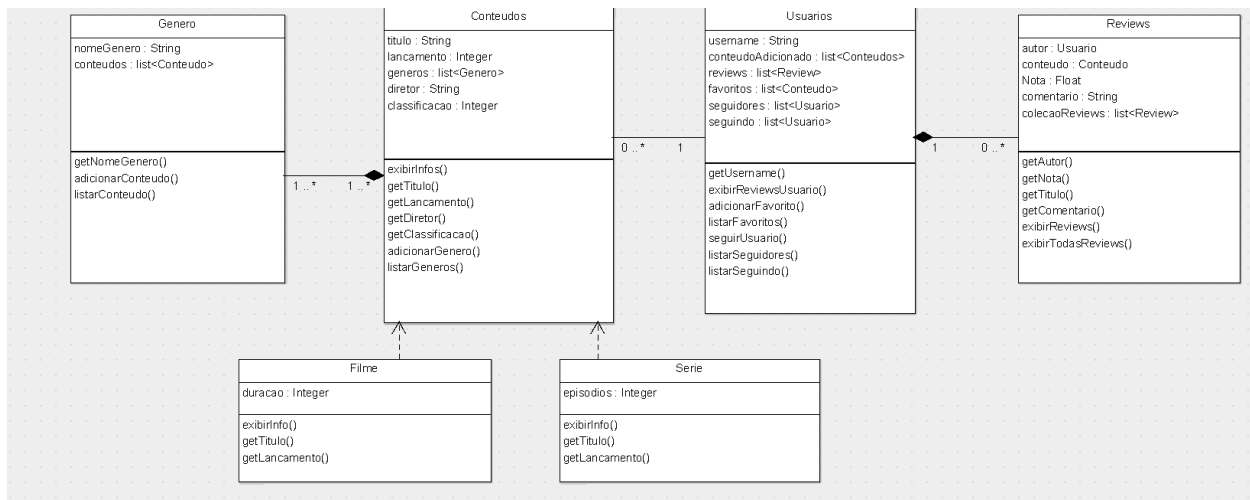
O começo do nosso trabalho se deu na criação de um diagrama de classes para que assim pudéssemos discutir juntos nossas ideias do que desejamos para o projeto final.

- A primeira versão do diagrama de classes:



AP2 Programação Orientada a Objetos por Julia Siodaro e Jeronimo Chiclana

- A versão final do diagrama de classes:



Decidimos tirar atributos que achamos desnecessários para a finalização do nosso projeto e adicionamos mais classes e métodos que se tornaram essenciais para a programação do banco de dados e do nosso código Java.

2. DESENVOLVIMENTO

Depois de revisar o Diagrama de classes nos dividimos nosso desenvolvimento em 3 etapas:

AP2 Programação Orientada a Objetos por Julia Siodaro e Jeronimo Chiclana

1. Fazer o banco de dados e código Java base/ inicial;
2. Revisão e aprimoramento de ambos os códigos;
3. Finalizar com a aplicação do DAO e JDBC;

A revisão e correção do código Java foi documentada no repositório:

<https://github.com/JeChicla/AP2-POO>

Nestas correções, consideramos novos métodos, removemos outros que consideramos desinteressantes para versão final e otimizamos a escrita e a funcionalidade do código.

3. Considerações finais:

O DAO foi o nosso maior desafio por conta da implementação levou em conta relacionamentos, importância de identificadores e a teoria de encapsulamento de métodos.

Concluimos com esse projeto que a melhor forma de aplicação seria primeiro o código em Java, banco de dados MYSQL e por fim o DAO, pois o nível de complexidade aumentou a cada passo que tomamos, se tornou também, mais difícil de fazer alterações em partes “antigas” do código, pois sua mudança levaria a edição de todas as partes subsequentes.

O Repositório oficial: https://github.com/JeChicla/AP2_POO_FINAL

AP2 Programação Orientada a Objetos por Julia Siodaro e Jeronimo Chiclana

```
public Filme buscarPorId(int id) {
    String sql = "SELECT c.id, c.titulo, c.lancamento, c.diretor, c.classificacao
        "FROM conteudo c JOIN filme f ON c.id = f.conteudo_id WHERE c.id = ?";

    try (PreparedStatement pstmt = connection.prepareStatement(sql)) {
        pstmt.setInt(1, id);

        try (ResultSet rs = pstmt.executeQuery()) {
            if (rs.next()) {
                Filme filme = new Filme(
                    rs.getInt(columnLabel: "id"),
                    rs.getString(columnLabel: "titulo"),
                    rs.getInt(columnLabel: "lancamento"),
                    generos: null, // generos - pode carregar depois
                    rs.getString(columnLabel: "diretor"),
                    rs.getInt(columnLabel: "classificacao"),
                    rs.getInt(columnLabel: "duracao")
                );
                return filme;
            }
        }
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
    return null;
}
```

Salvando os objetos no banco usando Dao.

AP2 Programação Orientada a Objetos por Julia Siodaro e Jeronimo Chiclana

```
public abstract class Conteudo { 36 usages 3 inheritors 1 JeChicla +1 *
    protected int id;
    protected String titulo; 7 usages
    protected int lancamento; 6 usages
    protected List<Genero> generos; 7 usages
    protected String diretor; 6 usages
    protected int classificacao; 6 usages

    public Conteudo(String titulo, int lancamento, String diretor, int classificacao) { 2 usages 1 Julia +1
        this.titulo = titulo;
        this.lancamento = lancamento;
        this.generos = new ArrayList<>();
        this.diretor = diretor;
        if (classificacao == 0 || classificacao == 10 || classificacao == 12 || classificacao == 14 || classificacao == 16 ||
            this.classificacao = classificacao;}
    }

    public Conteudo(int id, String titulo, int lancamento, List<Genero> generos, String diretor, int classificacao) { 2
        this.id = id;
        this.titulo = titulo;
        this.lancamento = lancamento;
        this.generos = generos;
        this.diretor = diretor;
        this.classificacao = classificacao;
    }
}
```

Atributos e métodos da classe abstrata conteúdo, que foram herdados pela classe Filme e Serie.

AP2 Programação Orientada a Objetos por Julia Siodaro e Jeronimo Chiclana

```
Main.java x ConnectionFactory.java Conteudo.java ForumFilmes.iml BaseDAO.java FilmeDAO.java ConteudoReposito
9 public class Main { JeChicla +1
10 public static void main(String[] args) { JeChicla +1
235 Serie theCrown = new Serie( titulo: "The Crown", lancamento: 2016, diretor: "Peter Morgan", classificacao: 14, episodios: 40);
236 sdao.salvar(gameOfThrones);
237 System.out.println("ID gerado: " + gameOfThrones.getId());
238 sdao.salvar(theCrown);
239 System.out.println("ID gerado: " + theCrown.getId());
240
241 ConteudoRepository cdao = new ConteudoRepository(connection);
242
243 cdao.adicionarGeneroAoConteudo(gameOfThrones.getId(), fantasia.getId());
244 cdao.adicionarGeneroAoConteudo(gameOfThrones.getId(), guerra.getId());
245
246
247
248
249 Review r20 = new Review(pedro, gameOfThrones, nota: 5f, titulo: "Muito ruim", comentario: "Gostei nada");
250 Review r21 = new Review(pedro, theGodfather, nota: 8.5f, titulo: "Muito bom", comentario: "Gostei bastante do filme.");
251
252 udao.seguirUsuario(pedro.getId(), camila.getId());
253 udao.seguirUsuario(camila.getId(), pedro.getId());
254 udao.adicionarConteudoAoUsuario(pedro.getId(), gameOfThrones.getId());
255 udao.adicionarFavoritoAoUsuario(pedro.getId(), gameOfThrones.getId());
256 rdao.buscarPorId(3);
257 udao.listarUsuarios2seguidores();
258 udao.listarUsuariosCom1Conteudo();
259 rdao.salvar(r20);
260 rdao.salvar(r21);
```

Print da Main com métodos da classe filmeDAO.