|  |  |
| --- | --- |
| **Nome** | **RA** |
| Leandro Cezar Pavin | 1680481512 |
| Pedro Luiz Weselowski | 1680481512025 |
| Rodrigo Garbe Cavreti | 1680481512 |
| Roger Paulino Donegati | 1680481512030 |



**Laboratório de Engenharia de Software**

**Aplicativo de Entretenimento**

**Relatório de andamento do projeto**

**Aluno:** nome do aluno 1 **RA:** RA do aluno 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Data inicial** | **Data final** | **Horas** | **Descrição sucinta das atividades** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Relatório de andamento do projeto**

**Aluno:** Pedro Luiz Weselowski da Cunha **RA:** 1680481512025

|  |  |  |  |
| --- | --- | --- | --- |
| **Data inicial** | **Data final** | **Horas** | **Descrição sucinta das atividades** |
| 19/04/17 | 19/04/17 | 3 horas | Debate Geral sobre o Sistema. |
| 19/04/17 | 19/04/17 | 5 horas | Fazer toda a parte escrita do projeto e auxiliar os outros integrantes em suas tarefas. |
| 19/04/17 | 20/04/17 | 5 horas | Principal desenvolvedor da parte escrita do projeto e dos diagramas de sequência. |
| 19/04/17 | 20/04/17 | 3 horas | Desenvolvedor dos diagramas de classe. |
| 20/04/17 | 20/04/17 | 2 horas | Revisão geral na documentação completa do projeto. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Relatório de andamento do projeto**

**Aluno:** Rodrigo Garbe Cavreti **RA:** 1680481512029

|  |  |  |  |
| --- | --- | --- | --- |
| **Data inicial** | **Data final** | **Horas** | **Descrição sucinta das atividades** |
| 17/04/17 | 18/04/17 | 4 horas | Levantamento dos requisitos funcionais. |
| 18/04/17 | 19/04/17 | 5 horas | Desenvolvimento dos casos de uso do sistema. |
| 19/04/17 | 20/04/17 | 3 horas | Auxilio na criação das classes e de seus respectivos diagramas. |
| 20/04/17 | 20/04/17 | 1 hora | Revisão da documentação finalizada. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Relatório de andamento do projeto**

**Aluno:** Roger Paulino Donegati **RA:** 1680481512030

|  |  |  |  |
| --- | --- | --- | --- |
| **Data inicial** | **Data final** | **Horas** | **Descrição sucinta das atividades** |
| 19/04/17 | 19/04/17 | 3 horas | Dirigente do Debate Geral sobre o Sistema. |
| 19/04/17 | 19/04/17 | 2 horas | Abstrair todas as classes do sistema com seus atributos e métodos. |
| 19/04/17 | 20/04/17 | 5 horas | Principal desenvolvedor dos Diagramas de classe. |
| 19/04/17 | 20/04/17 | 2 horas | Desenvolvedor do diagrama de Pacotes. |
| 20/04/17 | 20/04/17 | 4 horas | Unificação e Revisão geral na documentação completa do projeto. |
| 16/05/17 | 17/05/17 | 5 horas | Atualização e Detalhamento do Diagrama de Classes. |
| 17/05/17 | 17/05/17 | 3 horas | Unificação e revisão da documentação. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Sumário**

[1.1. Descrição do projeto 9](#_Toc482917742)

[1.2. Cronograma 9](#_Toc482917743)

[1.3. Recursos 9](#_Toc482917744)

[1.3.1. Recursos humanos 9](#_Toc482917745)

[1.3.2. Recursos físicos 9](#_Toc482917746)

[2. Especificação dos Requisitos do Sistema 10](#_Toc482917747)

[2.1. O produto 10](#_Toc482917748)

[2.1.1. Propósito 10](#_Toc482917749)

[2.1.2. Clientes 10](#_Toc482917750)

[2.1.3. Usuários 10](#_Toc482917751)

[2.1.4. Convenções de nomes e definições 10](#_Toc482917752)

[2.2. Restrições do produto 11](#_Toc482917753)

[2.3. Escopo do projeto 11](#_Toc482917759)

[2.4. Requisitos funcionais 12](#_Toc482917760)

[2.5. Requisitos não funcionais 13](#_Toc482917761)

[2.6. Protótipos do produto 14](#_Toc482917762)

[2.6.1. Modelo de navegação 14](#_Toc482917763)

[2.6.2. Protótipos de tela 15](#_Toc482917764)

[2.6.2.1. Tela 1 15](#_Toc482917765)

[2.6.2.2. Tela 2 16](#_Toc482917766)

[2.6.2.3. Tela 3 17](#_Toc482917767)

[2.6.2.4. Tela 4 18](#_Toc482917768)

[2.6.2.5. Tela 5 19](#_Toc482917769)

[2.6.2.6. Tela 6 20](#_Toc482917770)

[2.6.2.7. Tela 7 21](#_Toc482917771)

[2.6.2.8. Tela 8 22](#_Toc482917772)

[2.6.2.9. Tela 9 23](#_Toc482917773)

[3. Modelo de Casos de Uso do Sistema 24](#_Toc482917774)

[3.1. Diagrama de Casos de Uso do sistema 24](#_Toc482917775)

[3.2. Documentação dos Atores 25](#_Toc482917776)

[3.3. Documentação dos Casos de Uso 26](#_Toc482917777)

[3.3.1. Caso de uso Realizar Cadastro 26](#_Toc482917778)

[3.3.2. Caso de uso Realizar Login 27](#_Toc482917779)

[3.3.3. Caso de uso Editar Perfil 27](#_Toc482917780)

[3.3.4. Caso de uso Criticar Conteúdo 28](#_Toc482917781)

[3.3.5. Caso de uso Inserir Conteúdo em Lista 29](#_Toc482917782)

[3.3.6. Caso de uso Inserir Exibição de Conteúdos Favoritos 29](#_Toc482917783)

[4. Modelos de Análise do Sistema 30](#_Toc482917784)

[4.1. Especificação das classes de análise do sistema 30](#_Toc482917785)

[4.1.1. Visão geral dos pacotes de análise 30](#_Toc482917786)

[4.1.1.1. Pacote Usuário 30](#_Toc482917787)

[4.1.1.2. Pacote Relatorio 30](#_Toc482917788)

[4.1.1.3. Pacote Conteudo 30](#_Toc482917789)

[4.1.1.4. Pacote ComunicacaoBD 31](#_Toc482917790)

[4.1.1.5. Pacote InterfaceGrafica 31](#_Toc482917791)

[4.1.2. Diagramas e especificação das classes 31](#_Toc482917792)

[4.1.2.1. Classe Usuario 31](#_Toc482917793)

[4.1.2.2. Classe ControleUsuario 33](#_Toc482917794)

[4.1.2.3. Classe ControlePerfil 34](#_Toc482917795)

[4.1.2.4. Classe ControleRelatorio 36](#_Toc482917796)

[4.1.2.5. Classe ControleHome 37](#_Toc482917797)

[4.1.2.6. Classe Conteudo 38](#_Toc482917798)

[4.1.2.7. Classe Temporadas 40](#_Toc482917799)

[4.1.2.8. Classe Filme 41](#_Toc482917800)

[4.1.2.9. Classe EpsodioSerie 42](#_Toc482917801)

[4.1.2.10. Classe EpsodioAnime 42](#_Toc482917802)

[4.1.2.11. Classe ControleConteudo 43](#_Toc482917803)

[4.1.2.12. Classe FormPerfil 45](#_Toc482917804)

[4.1.2.13. Classe FormHome 45](#_Toc482917805)

[4.1.2.14. Classes ControleTemporada, ControleFilme, ControleSerie e ControleAnime 46](#_Toc482917806)

[4.1.2.15. Classes FormUsuario, FormRelatorios, FormConteudo e, FormTemporada 46](#_Toc482917807)

[4.2. Realizações de Casos de Uso 47](#_Toc482917808)

[4.2.1. Cenário: Transferir Valor 47](#_Toc482917809)

[4.2.2. Cenário: Outra realização … 48](#_Toc482917810)

[5. Modelos de Projeto do Sistema 48](#_Toc482917811)

[5.1. Classes de projeto 48](#_Toc482917812)

[5.2. Refinamento dos relacionamentos 50](#_Toc482917813)

[5.3. Realizações de casos de uso – Projeto 51](#_Toc482917814)

7/17

## 1.1. Descrição do projeto

Esse projeto tem como objetivo auxiliar consumidores de entretenimento online (filmes, séries, desenhos, novelas). Será desenvolvido um aplicativo cuja função é prover informações gerais e específicas sobre conteúdos que o usuário decidir, e gerenciar a agenda que o mesmo deve seguir para conseguir assistir seus filmes, séries, animes e novelas de maneira prazerosa.

## 1.2. CronogramaC:\Users\Duperron Tecnologia\Downloads\O plano\O plano\Fotos\Cronograma.jpeg

(Imagem em anexo para melhor visualização)

## 1.3. Recursos

### 1.3.1. Recursos humanos

Leandro Cezar Pavin ficará encarregado em produzir toda a interface do aplicativo. Também auxiliará em outras atividades gerais. (30 horas por semana)

Pedro Luiz Weselowski da Cunha ficará encarregado de desenvolver telas do aplicativo, auxiliará também na documentação do projeto. (30 horas por semana)

Rodrigo Garbe Cavreti ficará encarregado em gerenciar a equipe e auxiliar na programação. (25 horas por semana)

Roger Donegati será o programador principal do projeto. (20 horas por semana)

### 1.3.2. Recursos físicos

Para o desenvolvimento do projeto serão necessários computadores ou notebooks capazes de executar as ferramentas que serão utilizadas e smarphones para realização de testes. Serão utilizados os seguintes recursos para desenvolvimento do aplicativo: android studio, mysql, json, restful para execução do projeto.

# Especificação dos Requisitos do Sistema

## O produto

### Propósito

Atualmente existem milhares de pessoas que consomem entretenimento online, seja ele qual for. Por haver essa grande demanda milhares de filmes, séries, desenhos, novelas dentre outros, são produzidas por inúmeras empresas diferentes. O consumidor muitas vezes fica perdido sem saber quando novas temporadas e episódios de séries serão postados, novos filmes, avisos de atrasos, data e horário de transmissões e etc. O aplicativo irá unir todas as informações de diversos tipos diferentes de entretenimento online em um único lugar, facilitando a vida do usuário poupando o mesmo de passar trabalho procurando informações em lugares diferentes.

### Clientes

Empresas que tenham interesse em colocar o seu produto (filme, série e etc.) com algum tipo de destaque no aplicativo.

### Usuários

Os usuários serão pessoas interessadas em entretenimento online elas terão total acesso as ferramentas do aplicativo e poderão utiliza-las como acharem melhor. Haverá também uma tela específica para feedback do usuário, onde ele poderá nos dar dicas, sugestões e reclamações sobre o aplicativo para que possamos melhorar o app e deixar o usuário mais confortável.

### Convenções de nomes e definições

|  |  |  |
| --- | --- | --- |
| **Termo** | **Sigla** | **Definição** |
| Push Up | Nenhuma | São notificações que o usuário pode receber do aplicativo mesmo não estando com ele aberto. |
| Aplicativo | App | É um software para smatphones. |
| Trending | Nenhuma | É um termo em inglês mas comumente usado em outros idiomas também, serve para definir qualquer tipo de informação que tem sido muito acessada nos últimos dias ou horas, seu equivalente em português seria “estar em alta”. No caso deste software irá se referir majoritariamente aos conteúdos disponíveis |

## Restrições do produto

# O sistema deverá ser um produto de entretenimento voltado a área cinematográfica e televisiva.

# O sistema deverá permitir interação online entre usuários.

# O sistema deverá ter suporte apenas a plataforma mobile sendo categorizado como App.

# O sistema deverá divulgar informações relacionadas aos conteúdos de entretenimento.

# O sistema deverá ter suporte para com sistemas externos.

# 

## Escopo do projeto

FALTANDO

Explicar o que o produto a ser desenvolvido contemplará, de uma forma resumida, de modo que se tenha uma ideia clara de suas capacidades.

Sugere-se a utilização de um diagrama de contexto – ferramenta que explica as entradas e saídas dos sistemas e seus interagentes externos. Depois, em texto, explicar seus componentes. Exemplo:

Estoque Químico

Comprador

Químico

Sistema de

Rastreio

Químico

atualização de

estoques

container

químico

relatório de

estoques

container

químico

requisição

de produto

informação

do catálogo

consulta ao

catálogo

requisição de

compra

status do

pedido

Alternativamente pode ser utilizado um diagrama de casos de uso, resumindo cada caso (sem entrar em detalhes).

## Requisitos funcionais

|  |  |  |
| --- | --- | --- |
| **ID do Requisito** | **Descrição** | **Caso de Uso Relacionado** |
| R001 | O sistema deverá ser capaz de armazenar o cadastro dos usuários. | CSU01 |
| R002 | O sistema deverá disponibilizar login no mesmo via os dados necessários. | CSU02 |
| R003 | O sistema deverá sera capaz de realizar login via sistema externo(Facebook). | CSU02 |
| R004 | Sistema deverá possuir sessão de perfil do usuário, dispondo de opções para edição do mesmo. | CSU03 |
| R005 | Sistema deverá disponibilizar opções de críticas para conteúdos conteúdo. | CSU04 |
| R006 | Sistema deverá disponibilizar lista de conteúdos para usuários, sendo possível adicionar, remover e editar a lista conforme o desejado. | CSU05 |
| R007 | Sistema irá filtrar os conteúdos melhor avaliados e irá mostrar na tela determinada. | CSU06 |

## Requisitos não funcionais

|  |  |  |
| --- | --- | --- |
| **ID do Requisito** | **Descrição** | **Caso de Uso Relacionado** |
| RN001 | O sistema deverá ter uma interface amigável e iterativa (baseada em IHC) de forma que dispense muitas explicações. | CSU01, CSU02, CSU03, CSU04, CSU05 e CSU06 |
| RN002 | O sistema deve garantir a segurança das senhas e os dados pessoais dos usuários por meio de alguma criptografia. | CSU01, CSU02 e CSU03 |
| RN003 | O sistema deve ser funcionar em todos dispositivos android acima da versão 4.0 | CSU01, CSU02, CSU03, CSU04, CSU05 e CSU06 |
| RN004 | O sistema deve ter delay de no máximo 5s em suas ações com feedback ao usuário caso a ação esteja demorando mais do que isso por motivos de conexão de banda larga por exemplo. | CSU04, CSU05 e CSU06 |
| RN005 | O sistema deve ser feito permitindo uma fácil portabilidade para com os dados, caso seja necessário a troca de servidor e até mesmo de linguagem para com o banco de dados. | CSU01, CSU02, CSU03, CSU04, CSU05 e CSU06 |
| RN006 | O sistema deve possibilitar a implantação futura de mais API’s externas para absorver dados necessários. | CSU04, CSU05 e CSU06 |
| RN007 | O sistema deve deve bloquear conteúdos adultos para menores de 18 anos. | CSU04, CSU05 e CSU06 |
| RN08 | O sistema deve realizar conexões seguras com sistemas externos sem expor dados importantes do usuário. | CSU04, CSU05 e CSU06 |

## Protótipos do produto

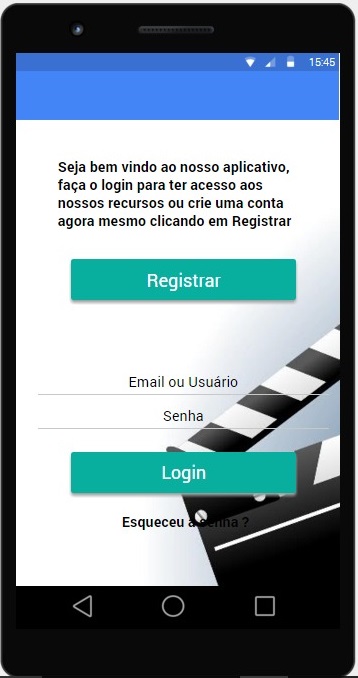
### C:\Users\Duperron Tecnologia\Desktop\Foto Modelo Navegação.pngModelo de navegação

(Foto disponível em anexo para melhor visualização)

### Protótipos de tela

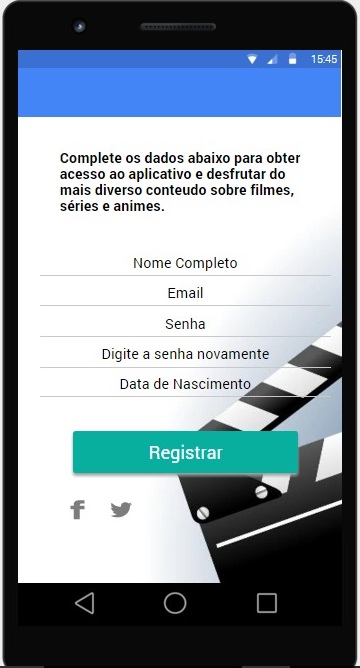
(Todas as fotos estão disponíveis em anexo para melhor visualização caso necessário)

#### Tela 1



Explicação: Esta é a tela de login, onde o usuário, se estiver cadastro, deve efetuar o login para ter acesso ao conteúdo. Caso contrário, deverá efetuar o cadastro de um novo perfil para conseguir acessar as informações.

#### Tela 2



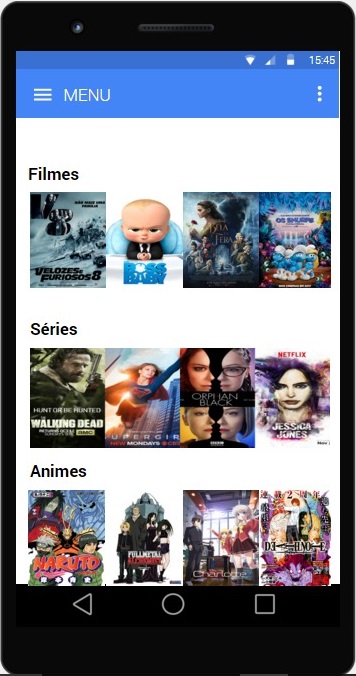
Explicação: Esta tela é onde é efetuado o registro do usuário, onde o usuário colocará informações básicas sobre ele e criará uma conta para prosseguir no aplicativo.

#### Tela 3



Explicação: Esta tela é usada caso o usuário em questão perdeu ou esqueceu a senha, então é usado um método de recuperação de senha via e-mail.

#### Tela 4



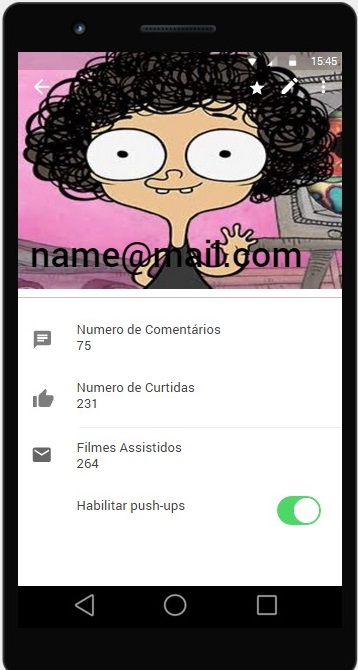
Explicação: Esta tela consiste na parte principal do aplicativo, onde o usuário ao efetuar o login terá acesso ao mais diversos conteúdo de filmes, series e anime. Podendo também, visualizar as listas salvas pelo mesmo.

#### Tela 5



Explicação: Esta tela é onde é apresentado o perfil de cada usuário onde mostrará conteúdos em alta e terá acesso as configurações do aplicativo, podendo efetuar mudanças a seu gosto.

#### Tela 6



Explicação: Esta tela mostrará as estatísticas de cada usuário, o tanto de comentários, curtidas, filmes assistidos e terá a opção de notificação.

#### Tela 7



Explicação: Esta tela mostra o conteúdo de cada filme em particular, ou seja, as informações básicas de cada filme, como ano de lançamento, elenco, curiosidades gerais, etc.

#### Tela 8



Explicação: Esta tela é um exemplo de como seria as informações de curiosidade de um filme, apenas para o usuário saber um pouco mais sobre os filmes que gosta.

#### Tela 9



Explicação: Esta tela tem por objetivo avaliar o filme e de certa forma fazer uma crítica se gostou ou não do conteúdo.

# Modelo de Casos de Uso do Sistema

## C:\Users\Duperron Tecnologia\Downloads\O plano\O plano\Fotos\Diagrama de Caso de Uso.pngDiagrama de Casos de Uso do sistema

(Foto disponível em anexo para melhor visualização)

## Documentação dos Atores

|  |
| --- |
| **Ator**: Usuario |
| ID: A01 |
| **Resumo:** Usuário ira controlar e utilizar as funcionalidades do sistema, sendo assim uma peça extremamente importante para o mesmo. |
| Responsabilidade:   1. Controlar o sistema, realizando cadastro de seus dados, logando ou realizando tarefas disponíveis no mesmo. |
| **Ambiente Físico**  Irrelevante, sistema estará disponível em dispositivos moveis. |
| **Número e Tipo**  Sendo o ator essencial do sistema, a quantidade do mesmo representa o sucesso do sistema. |
| **Frequência com que usa o sistema**  Enquanto o sistema estiver sendo utilizado este ator estará interagindo por ser o piloto do mesmo. |

|  |
| --- |
| **Ator**: Banco De Dados |
| ID: A02 |
| **Resumo:** Irá armazenar as informações necessárias para continuidade do sistema. |
| Responsabilidade:   1. Armazenar informações de Usuários e Conteúdos. 2. Disponibilizar informações armazenadas para o andamento do sistema. |
| **Ambiente Físico**  Armazenamento em nuvem. |
| **Número e Tipo**  Banco deverá possuir quantidade suficiente de armazenamento para suprir necessidades do sistema. |
| **Frequência com que usa o sistema**  Sempre que o sistema requerer informações este ator será requisitado, sendo assim, seu uso será extremamente constante. |

|  |
| --- |
| **Ator**: Sistema Externo FaceBook |
| ID: A03 |
| **Resumo:** Irá disponibilizar informações necessárias para login via dados do FaceBook |
| Responsabilidade:   1. Auxiliar no login disponibilizando informações necessárias para o mesmo. |
| **Ambiente Físico**  Armazenamento em nuvem. |
| **Número e Tipo**  Sistema externo. |
| **Frequência com que usa o sistema**  Sempre que opção de login via faceBook ser solicitada, ator entrara em ação. |

## Documentação dos Casos de Uso

### Caso de uso Realizar Cadastro

|  |
| --- |
| **CASO DE USO**: REALIZAR CADASTRO |
| **ID:** CSU01 |
| **RESUMO:** O usuario poderá fazer cadastro por meio da função nativa do aplicativo ou sincronizar com o Facebook. |
| **Atores primários:** Usuário, Banco de Dados. |
| **Atores secundários:** Sistema Externo Facebook. |
| **Pré-Condições:** 1 – O usuário deverá estar na tela inicial de login e selecionar criar cadastro. |
| **Fluxo de Evento Principal:**   1. O usuário seleciona o método de cadastro.    1. Se método de sincronização do Facebook for selecionado então abrira uma tela do sistema externo para sincronização do cadastro.    2. Se método de cadastro nativo for selecionado, então abrira a tela para inserir os dados. 2. O usuário irá inserir os dados referentes ao cadastro. 3. Sistema verifica validade de todos os dados. 4. Usuário é cadastrado no banco. |
| **Pós-Condições:**   1. Usuário esta cadastrado no sistema. 2. Sistema está na tela de login aguardando ações. |
| **Fluxo de eventos alternativos:** |

### Caso de uso Realizar Login

|  |
| --- |
| **CASO DE USO**: REALIZAR LOGIN |
| **ID:** CSU01 |
| **RESUMO:** Usuário ira realizar o login utilizando seus dados cadastrados ou via Facebook. |
| **Atores primários:** Usuário, Banco de Dados. |
| **Atores secundários:** Sistema externo Facebook |
| **Pré-Condições:**   1. Usuário deverá ter conta cadastrada ou conta externa do Facebook. 2. Sistema precisa estar na tela inicial de login. |
| **Fluxo de Evento Principal:**   1. Usuário seleciona método de login.    1. Se método de sincronização for selecionado então abrira uma tela para sincronização com o Facebook.    2. Se método de login padrão for selecionado então usuário devera que inserir os dados para login. 2. Usuário insere dados de login.   **{Verifica Login}**   1. Com os dados verificados o usuário ira logar no sistema. 2. Sistema ficara na tela inicial aguardando ações. |
| **Pós-Condições:**   1. Usuário estará na tela inicial do sistema podendo utilizar as demais funções de usuário logado. |
| **Fluxo de eventos alternativos:**   1. Em **{Verifica Login}** caso os dados de login estejam errados, o sistema ira limpar os campos e enviara uma mensagem pop-up para o usuário. 2. Sistema limpa os campos de login. 3. Sistema envia mensagem de pop-up relatando os dados errados. |

### Caso de uso Editar Perfil

|  |
| --- |
| **CASO DE USO**: EDITAR PERFIL |
| **ID:** CSU01 |
| **RESUMO:** Usuário poderá editar seus dados pessoais caso necessário. |
| **Atores primários:** Usuário. |
| **Atores secundários:** Banco de Dados. |
| **Pré-Condições:**   1. Usuário deverá estar logado no sistema. 2. Usuário precisa selecionar opção de edição de perfil. |
| **Fluxo de Evento Principal:**   1. Usuário seleciona opção de edição de perfil. 2. Usuário modifica os dados desejados e seleciona a opção de salvar.   **{Verifica Dados}**   1. Sistema salva os novos dados editados do usuário. |
| **Pós-Condições:**   1. Usuário estará na tela inicial do sistema podendo utilizar as demais funções de usuário logado. |
| **Fluxo de eventos alternativos:** |

### Caso de uso Criticar Conteúdo

|  |
| --- |
| **CASO DE USO**: Criticar Conteúdo |
| **ID:** CSU04 |
| **RESUMO:** Usuário será capaz de comentar ou disponibilizar curtidas no conteúdo desejado. |
| **Atores primários:** Usuário. |
| **Atores secundários:** Banco de Dados. |
| **Pré-Condições:**   1. Usuário deverá estar logado no sistema. 2. Usuário precisa selecionar um conteúdo (filme, serie ou anime). 3. Usuário precisa selecionar um título para critica. |
| **Fluxo de Evento Principal:**   1. Usuário seleciona a seção de conteúdo desejada. 2. Usuário seleciona o titulo desejado. 3. Usuário critica o filme utilizando as opções disponíveis.    1. Usuário escolhe a opção de comentário e insere o mesmo.    2. Usuário clica no botão referente a curtir |
| **Pós-Condições:**   1. O conteúdo estará com a crítica do usuário |
| **Fluxo de eventos alternativos:** |

### Caso de uso Inserir Conteúdo em Lista

|  |
| --- |
| **CASO DE USO**: INSERIR CONTEUDO EM LISTA. |
| **ID:** CSU05 |
| **RESUMO:** Usuário será capaz de criar uma lista de conteúdos que deseja assistir ou que já assistiu e manipula-la. |
| **Atores primários:** Usuário. |
| **Atores secundários:** Banco de Dados. |
| **Pré-Condições:**   1. Usuário deverá estar logado no sistema. 2. Usuário precisa selecionar um conteúdo (filme, serie ou anime). 3. Usuário precisa selecionar um título para adicionar em sua lista. |
| **Fluxo de Evento Principal:**   1. Usuário pesquisa o conteúdo desejado. 2. Usuário seleciona a opção de adicionar o conteúdo em sua lista |
| **Pós-Condições:**   1. O conteúdo estará adicionado na lista do usuário. |
| **Fluxo de eventos alternativos:** |

### Caso de uso Inserir Exibição de Conteúdos Favoritos

|  |
| --- |
| **CASO DE USO**: Exibição conteúdos favoritos. |
| **ID:** CSU06 |
| **RESUMO:** O sistema ira filtrar os conteúdos mais bem avaliados e irá mostrar na tela inicial. |
| **Atores primários:** Banco de Dados. |
| **Atores secundários:** |
| **Pré-Condições:**   1. Sistema deverá estar na tela inicial com algum usuário logado. |
| **Fluxo de Evento Principal:**   1. Sistema calcula os conteúdos com melhores avaliações.    1. Caso não haja informações suficientes será escolhido aleatoriamente. 2. Sistema exibe os conteúdos encontrados em sua respectiva área de favoritos |
| **Pós-Condições:**   1. Conteúdos serão mostrados na tela inicial. |
| **Fluxo de eventos alternativos:** |

# Modelos de Análise do Sistema

## Especificação das classes de análise do sistema

### C:\Users\Duperron Tecnologia\Desktop\foto pacote.pngVisão geral dos pacotes de análise

(Foto disponível anexo para melhor visualização)

**O diagrama de pacotes acima exibe as seguintes classes do sistema**: Usuario, ControleUsuario, ControlePerfil, ControleRelatorio, ControleHome, Conteudo, Temporadas, Filme, EpisodioSerie, EpisodioAnime, ControleConteudo, FormPerfil, FormHome, ControleTemporada, ControleFilme, ControleSerie, ControleAnime, FormUsuario, FormRelatorios, FormConteudo e FormTemporada.

#### Pacote Usuário

Agrupa as classes responsáveis por gerenciar tudo que é relativo ao usuário, como suas configurações, seus dados, seu perfil traçado, etc... Esse Pacote Interage com todos os outros pacotes do sistema uma vez que o usuário é um objeto fundamental nesse sistema.

#### Pacote Relatorio

É o pacote responsável por analisar o perfil do usuário e mostrar as informações estatísticas disponíveis sobre ele.

#### Pacote Conteudo

Este pacote tem como objetivo unificar as principais classes relacionadas aos conteúdos do sistema, contando com sua classe gerencial que controla todos eles e as classes que geram conteúdos como, filmes, series e anime.

#### Pacote ComunicacaoBD

São todas as classes que serão acessadas por grande parte do sistema para realizar ações no banco de dados, agrupadas em um único pacote facilita a visualização, entendimento do projeto e também é um meio facilitador para o desenvolvimento.

#### Pacote InterfaceGrafica

Neste pacote estão reunidas as classes que fazem a comunicação entre o sistema e o usuário final, seu principal objetivo é receber informações de outras classes e mostra-las da melhor forma possível. Além de gerar boas interações com o usuário para manter o sistema rodando dinamicamente.

### C:\Users\Duperron Tecnologia\Desktop\foto classe.pngDiagramas e especificação das classes

(Foto disponível anexo para melhor visualização)

#### Classe Usuario

**Descrição**

A classe Usuario representa um usuário do sistema.

**Superclasse(s)**

Nenhuma

**Subclasse(s)**

Nenhuma

**Atributos**

|  |  |
| --- | --- |
| **Nome:** | **nome** |
| **Descrição:** | Representa o nome do usuário para acessar a sua conta na aplicação. |
| **Tipo:** | String |
| **Domínio:** | Qualquer string limitado a 50 caracteres. |
| **Nome:** | **senha** |
| **Descrição:** | Representa a senha do usuário para acessar a sua conta na aplicação. |
| **Tipo:** | String |
| **Domínio:** | Qualquer string limitado a 50 caracteres. |
| **Nome:** | **sincronizado** |
| **Descrição:** | Informa se a conta esta sincronizada com um sistema externo como Facebook. |
| **Tipo:** | int, limitado a 1 digito |
| **Domínio:** | De 0 a 9 |
| **Nome:** | **nascimento** |
| **Descrição:** | Representa a data de nascimento do usuário |
| **Tipo:** | Date |
| **Domínio:** | Qualquer Date acima de “01/01/1900” e menor que a data atual. |
| **Nome:** | **email** |
| **Descrição:** | Representa o email do usuário na aplicação. |
| **Tipo:** | String |
| **Domínio:** | Qualquer string limitado a 150 caracteres. |
| **Nome:** | **curtidas** |
| **Descrição:** | Representa o número de conteúdos curtidos pelo usuário. |
| **Tipo:** | int |
| **Domínio:** | Qualquer valor no intervalo numérico de int |
| **Nome:** | **comentarios** |
| **Descrição:** | Representa o número de comentários em filmes feito pelo usuário. |
| **Tipo:** | int |
| **Domínio:** | Qualquer valor no intervalo numérico de int |
| **Nome:** | **lista** |
| **Descrição:** | Representa os filmes que o usuário deseja salvar em uma lista para consultas pessoais. |
| **Tipo:** | Conteudo |
| **Domínio:** | Qualquer Conteudo cadastrado no sistema. |

**Operações**

|  |  |
| --- | --- |
| **Nome:** | Getters and Setters |
| **Descrição:** | Obter dados do Usuario e Inseri-los. |
| **Pré-condições:** | |
| 1. O usuário deve estar instanciado no sistema. | |

#### Classe ControleUsuario

**Descrição**

A classe ControleUsuario é responsável por cadastrar e validar o usuário nos sistema, além de validar campos como o nascimento.

**Superclasse(s)**

Nenhuma

**Subclasse(s)**

Nenhuma

**Atributos**

|  |
| --- |
| Nenhum |

**Operações**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome:** | validaNascimento | | |
| **Descrição:** | Recebe um data de nascimento e verifica se ela é valida | | |
| **Parâmetros** | | | |
| **Nome** | | **Tipo** | **Propósito** |
| data | | Date | Representa a data ser validada. |
| **Pré-condições:** | | | |
| 1. O usuário deve tentar realizar o seu cadastro no sistema | | | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | | | |
| 1. O sistema recebe a data e realiza cálculos aritméticos para valida-la | | | |
| **Pós-condições:** | | | |
| 1. Informar que a data é válida. | | | |
| **Nome:** | CRUD | | |
| **Descrição:** | Representa o ato de cadastrar, recupera, atualizar e deletar os dados de um usuário no banco de dados. | | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | | | |
| 1. O sistema realiza alguma ação referente a classe Usuario no banco de dados retorna o resultado desta. | | | |

#### Classe ControlePerfil

**Descrição**

A classe ControlePerfil é responsável por gerenciar as ações do usuário no sistema para ter um maior controle sobre as preferências de cada usuário individualmente.

**Superclasse(s)**

Nenhuma

**Subclasse(s)**

Nenhuma

**Atributos**

|  |
| --- |
| Nenhum |

**Operações**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome:** | controlarCurtidas | | |
| **Descrição:** | Busca as curtidas recentes do usuário as agrupando e cadastrando no banco de dados, ajudando a gerar listas de sugestões de conteúdos ao usuário. | | |
| **Parâmetros** | | | |
| **Nome** | | **Tipo** | **Propósito** |
| usuario | | Usuario | Representa todas as informações do Usuário a ser gerenciado. |
| **Pré-condições:** | | | |
| 1. O usuário deve estar instanciado no sistema e ter curtido pelo menos um conteúdo. | | | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | | | |
| 1. Buscar informações sobre os conteúdos curtidos pelo usuário como duração, gênero, lançamento, etc... para atualizar as opções de sugestões ao usuário. | | | |
| **Pós-condições:** | | | |
| 1. Após levantar conteúdos inseri-los em listas de sugestões aos usuários. | | | |
| **Nome:** | controlarComentarios | | |
| **Descrição:** | Busca as comentarios recentes do usuário as agrupando e cadastrando no banco de dados, ajudando a gerar listas de sugestões de conteúdos ao usuário. | | |
| **Parâmetros** | | | |
| **Nome** | | **Tipo** | **Propósito** |
| usuario | | Usuario | Representa todas as informações do Usuário a ser gerenciado. |
| **Pré-condições:** | | | |
| 1. O usuário deve estar instanciado no sistema e ter comentado pelo menos um conteúdo. | | | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | | | |
| 1. Buscar informações sobre os conteúdos comentados pelo usuário como duração, gênero, lançamento, etc... para atualizar as opções de sugestões ao usuário. | | | |
| **Pós-condições:** | | | |
| 1. Após levantar conteúdos inseri-los em listas de sugestões aos usuários. | | | |
| **Nome:** | controlarAssistidos | | |
| **Descrição:** | Busca os conteúdos assistidos recentemente pelo usuário as agrupando e cadastrando no banco de dados, ajudando a gerar listas de sugestões de conteúdos ao usuário. | | |
| **Parâmetros** | | | |
| **Nome** | | **Tipo** | **Propósito** |
| usuario | | Usuario | Representa todas as informações do Usuário a ser gerenciado. |
| **Pré-condições:** | | | |
| 1. O usuário deve estar instanciado no sistema e ter marcado pelo menos um conteúdo como assistido. | | | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | | | |
| 1. Buscar informações sobre os conteúdos assistidos pelo usuário como duração, gênero, lançamento, etc... para atualizar as opções de sugestões ao usuário. | | | |
| **Pós-condições:** | | | |
| 1. Após levantar conteúdos inseri-los em listas de sugestões aos usuários. | | | |
| **Nome:** | controlarLista | | |
| **Descrição:** | Analisa os conteúdos presentes nas listas salvas pelos usuário e busca conteúdos similares para sugestão. | | |
| **Parâmetros** | | | |
| **Nome** | | **Tipo** | **Propósito** |
| usuario | | Usuario | Representa todas as informações do Usuário a ser gerenciado. |
| **Pré-condições:** | | | |
| 1. O usuário deve estar instanciado no sistema e ter pelo menos 10 filmes em uma de suas listas. | | | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | | | |
| 1. Buscar informações sobre os conteúdos em listas do usuário para atualizar as opções de sugestões. | | | |
| **Pós-condições:** | | | |
| 1. Após levantar conteúdos inseri-los em listas de sugestões aos usuários. | | | |

#### Classe ControleRelatorio

**Descrição**

A classe ControleRelatorio é a classe responsável por gerar gráficos para o usuário.

**Superclasse(s)**

Nenhuma

**Subclasse(s)**

Nenhuma

**Atributos**

|  |
| --- |
| Nenhum |

**Operações**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome:** | gerCurtidas | | |
| **Descrição:** | Recebe o número de curtidas do usuário e gera relatórios e gráficos de acordo com as datas das curtidas. | | |
| **Parâmetros** | | | |
| **Nome** | | **Tipo** | **Propósito** |
| curtidas | | int | Representa o número de curtidas do usuário |
| **Pré-condições:** | | | |
| 1. O usuário deve estar instanciado no sistema e ter curtido pelo menos um conteúdo. | | | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | | | |
| 1. Gerar gráficos para o usuário a partir do parâmetro recebido e comunicação com o banco de dados. | | | |
| **Pós-condições:** | | | |
| 1. Exibir os gráficos gerados | | | |
| **Nome:** | gerComentarios | | |
| **Descrição:** | Recebe o número de comentários do usuário e gera relatórios e gráficos de acordo com as datas dos comentários. | | |
| **Parâmetros** | | | |
| **Nome** | | **Tipo** | **Propósito** |
| comentarios | | int | Representa o número de comentários do usuário |
| **Pré-condições:** | | | |
| 1. O usuário deve estar instanciado no sistema e ter comentado pelo menos um conteúdo. | | | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | | | |
| 1. Gerar gráficos para o usuário a partir do parâmetro recebido e comunicação com o banco de dados. | | | |
| **Pós-condições:** | | | |
| 1. Exibir os gráficos gerados | | | |
| **Nome:** | gerAssistidos | | |
| **Descrição:** | Recebe o número de conteúdos assistido pelo usuário e gera relatórios e gráficos de acordo com a duração dos conteúdo e datas em que foram assistidos. | | |
| **Parâmetros** | | | |
| **Nome** | | **Tipo** | **Propósito** |
| views | | int | Representa o número de filmes assistidos pelo usuário. |
| **Pré-condições:** | | | |
| 1. O usuário deve estar instanciado no sistema e ter curtido pelo menos um conteúdo. | | | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | | | |
| 1. Gerar gráficos para o usuário a partir do parâmetro recebido e comunicação com o banco de dados. | | | |
| **Pós-condições:** | | | |
| 1. Exibir os gráficos gerados | | | |

#### Classe ControleHome

**Descrição**

A classe ControleHome é responsável por analisar o perfil do usuário, organizar e mostrar os conteúdos na homepage do app de melhor forma possível para cada usuario.

**Superclasse(s)**

Nenhuma

**Subclasse(s)**

Nenhuma

**Atributos**

|  |
| --- |
| Nenhum |

**Operações**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome:** | analisaPerfil | | |
| **Descrição:** | Busca as sugestões já cadastradas para o usuário e as combina com os conteúdos de maior trending relacionados ao usuario. | | |
| **Parâmetros** | | | |
| **Nome** | | **Tipo** | **Propósito** |
| usuario | | Usuario | Representa todas as informações do Usuário a ser gerenciado. |
| **Pré-condições:** | | | |
| 1. O usuário deve estar instanciado no sistema. | | | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | | | |
| 1. Buscar informações sobre o usuário já cadastradas no banco. 2. Busca informações sobre os conteúdos mais acessados no sistema. 3. Salva as informações levantadas apenas em tempo de execução no sistema. | | | |
| **Pós-condições:** | | | |
| 1. Guardas as informações levantadas para serem utilizadas em seguida pela própria classe. | | | |
| **Nome:** | AgrupaConteudos | | |
| **Descrição:** | Recebe uma lista de conteúdos e as agrupa da melhor forma possível na homepage para o usuario. | | |
| **Parâmetros** | | | |
| **Nome** | | **Tipo** | **Propósito** |
| lista | | List<Conteudo> | Representa uma lista de Conteúdo a ser organizada. |
| **Pré-condições:** | | | |
| 1. O usuário deve estar instanciado no sistema e ter comentado pelo menos um conteúdo. 2. O perfil do usuário deve ser analisado anteriormente. | | | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | | | |
| 1. Analisa os conteúdos recebido comparando notas, duração, gênero, etc...E os agrupando da melhor forma possível na homepage | | | |
| **Pós-condições:** | | | |
| 1. Mostrar ao usuário os conteúdos agrupados no FormHome. | | | |

#### Classe Conteudo

**Descrição**

A classe Conteudo é uma classe abstrata que representa de forma genérica qualquer tipo de contudo que o usuário gostaria de acessar.

**Superclasse(s)**

Nenhuma

**Subclasse(s)**

Filmes, Temporada.

**Atributos**

|  |  |
| --- | --- |
| **Nome:** | **titulo** |
| **Descrição:** | Representa o título do conteúdo seja ele um filme, serie ou anime. |
| **Tipo:** | String |
| **Domínio:** | Qualquer string limitado a 90 caracteres. |
| **Nome:** | **lancamento** |
| **Descrição:** | Representa a data de lançamento do conteúdo. |
| **Tipo:** | Date |
| **Domínio:** | Qualquer valor tipo Date. |
| **Nome:** | **classificacao** |
| **Descrição:** | Representa a faixa etária para a qual o conteúdo é indicado |
| **Tipo:** | Int, limitado a 2 digitos |
| **Domínio:** | Qualquer valor no intervalo numérico de int |
| **Nome:** | **duracao** |
| **Descrição:** | Representa a duração de um conteúdo seja ele: filme, episódio de serie ou anime. |
| **Tipo:** | int |
| **Domínio:** | Qualquer valor no intervalo numérico de int |
| **Nome:** | **produtora** |
| **Descrição:** | Representa a produtora responsável pelo conteúdo. |
| **Tipo:** | String |
| **Domínio:** | Qualquer string limitado a 90 caracteres. |
| **Nome:** | **curtidas** |
| **Descrição:** | Informa o número de curtidas fornecidas pelos usuários ao Conteudo. |
| **Tipo:** | int, limitado a 9 digitos |
| **Domínio:** | Qualquer valor no intervalo numérico de int. |
| **Nome:** | **nota** |
| **Descrição:** | Representa a nota média do Conteudo fornecida pelos usuários. |
| **Tipo:** | Int, limitado a 1 digito |
| **Domínio:** | Qualquer valor no intervalo numérico de int. |
| **Nome:** | **genero** |
| **Descrição:** | Representa o/os gêneros do conteudo |
| **Tipo:** | List<String> |
| **Domínio:** | Qualquer string limitado a 40 caracteres. |
| **Nome:** | **roteristas** |
| **Descrição:** | Representa todos os roteiristas responsáveis pelos conteúdo. |
| **Tipo:** | List<String> |
| **Domínio:** | Qualquer string limitado a 60 caracteres. |
| **Nome:** | **comentarios** |
| **Descrição:** | Representa todos os comentários dos usuários para com o Conteudo. |
| **Tipo:** | List<String> |
| **Domínio:** | Qualquer string limitado a 350 caracteres. |
| **Nome:** | **comentarios** |
| **Descrição:** | Representa todos os diretores responsáveis pelos conteúdo. |
| **Tipo:** | List<String> |
| **Domínio:** | Qualquer string limitado a 60 caracteres. |

**Operações**

|  |  |
| --- | --- |
| **Nome:** | Getters and Setters |
| **Descrição:** | Obter dados do Conteudo e Inseri-los. |
| **Pré-condições:** | |
| 1. O Conteudo deve estar instanciado no sistema. | |

#### Classe Temporadas

**Descrição**

A classe Temporadas “Todo” que agrupa uma lista de episódios de series ou animes a serem mostrados no FormConteudo ou Homepage.

**Superclasse(s)**

Conteudo

**Subclasse(s)**

EpsodioSerie, EpsodioAnime

**Atributos**

|  |  |
| --- | --- |
| **Nome:** | **episodios** |
| **Descrição:** | Informa o número de episódios presentes na temporada. |
| **Tipo:** | int, limitado a 4 digitos |
| **Domínio:** | Qualquer valor no intervalo numérico de int. |
| **Nome:** | **seasons** |
| **Descrição:** | Representa o numero de temporadas presentes na serie ou anime. |
| **Tipo:** | Int, limitado a 2 digitos. |
| **Domínio:** | Qualquer valor no intervalo numérico de int. |
| **Nome:** | **status** |
| **Descrição:** | Representa o estado da série ou anime, se continua lançando novos episódios ou se já foi encerrado. |
| **Tipo:** | boolean |
| **Domínio:** | True ou false. |

**Operações**

|  |  |
| --- | --- |
| **Nome:** | Getters and Setters |
| **Descrição:** | Obter dados das Temporadas e Inseri-los. |
| **Pré-condições:** | |
| 1. As Temporadas devem estar instanciado no sistema. | |

#### Classe Filme

**Descrição**

A classe Filme representa um filme a ser visualizado pelo usuário.

**Superclasse(s)**

Conteudo

**Subclasse(s)**

Nenhuma

**Atributos**

|  |  |
| --- | --- |
| **Nome:** | **atores** |
| **Descrição:** | Informa os atores que trabalharam no filme. |
| **Tipo:** | List<string> |
| **Domínio:** | Qualquer caractere possível em uma string. |

**Operações**

|  |  |
| --- | --- |
| **Nome:** | Getters and Setters |
| **Descrição:** | Obter dados do Filme e Inseri-los. |
| **Pré-condições:** | |
| 1. O Filme deve estar instanciado no sistema. | |

#### Classe EpsodioSerie

**Descrição**

A classe EpsodioSerie representa um episódio de uma serie a ser visualizado pelo usuário.

**Superclasse(s)**

Temporadas

**Subclasse(s)**

Nenhuma

**Atributos**

|  |  |
| --- | --- |
| **Nome:** | **atores** |
| **Descrição:** | Informa os atores que trabalharam no filme. |
| **Tipo:** | List<string> |
| **Domínio:** | Qualquer caractere possível em uma string. |
| **Nome:** | **season** |
| **Descrição:** | Informa a qual temporada pertence esse episódio. |
| **Tipo:** | string |
| **Domínio:** | Qualquer caractere possível em uma string. |

**Operações**

|  |  |
| --- | --- |
| **Nome:** | Getters and Setters |
| **Descrição:** | Obter dados do episódio e Inseri-los. |
| **Pré-condições:** | |
| 1. O episódio deve estar instanciado no sistema. | |

#### Classe EpsodioAnime

**Descrição**

A classe EpsodioAnime representa um episódio de um anime a ser visualizado pelo usuário.

**Superclasse(s)**

Temporadas

**Subclasse(s)**

Nenhuma

**Atributos**

|  |  |
| --- | --- |
| **Nome:** | **dubladores** |
| **Descrição:** | Informa os dubladores dos personagens presentes no anime. |
| **Tipo:** | List<string> |
| **Domínio:** | Qualquer caractere possível em uma string. |
| **Nome:** | **season** |
| **Descrição:** | Informa a qual temporada pertence esse episódio. |
| **Tipo:** | string |
| **Domínio:** | Qualquer caractere possível em uma string. |

**Operações**

|  |  |
| --- | --- |
| **Nome:** | Getters and Setters |
| **Descrição:** | Obter dados do episódio e Inseri-los. |
| **Pré-condições:** | |
| 1. O episódio deve estar instanciado no sistema. | |

#### Classe ControleConteudo

**Descrição**

A classe ControleConteudo é responsável por gerenciar e adicionar informações fornecidas pelos usuários aos Conteudos.

**Superclasse(s)**

Nenhuma

**Subclasse(s)**

Nenhuma

**Atributos**

|  |
| --- |
| Nenhum |

**Operações**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome:** | curtir | | |
| **Descrição:** | Incrementa o número de curtidas ao Conteudo. | | |
| **Parâmetros** | | | |
| **Nome** | | **Tipo** | **Propósito** |
| usuario | | Usuario | Representa todas as informações do Usuário a ser gerenciado. |
| **Pré-condições:** | | | |
| 1. O usuário deve estar instanciado no sistema. 2. O conteúdo deve estar instanciado no sistema. | | | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | | | |
| 1. Verificar se o usuário já Curtiu esse Conteudo e caso não tenho curtido incrementa o número de curtidas do mesmo. | | | |
| **Pós-condições:** | | | |
| 1. Atualiza a informação do Conteudo no banco de Dados. | | | |
| **Nome:** | comentar | | |
| **Descrição:** | Adiciona o comentário do usuário ao Conteudo. | | |
| **Parâmetros** | | | |
| **Nome** | | **Tipo** | **Propósito** |
| usuario | | Usuario | Representa todas as informações do Usuário a ser gerenciado. |
| c | | String | Representa o comentário a ser cadastrado. |
| **Pré-condições:** | | | |
| 1. O usuário deve fornecer um comentário dentro das políticas do aplicativo e não nulo. 2. O usuário deve estar instanciado no sistema. 3. O conteúdo deve estar instanciado no sistema. | | | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | | | |
| 1. Validar o comentário e cadastra-lo junto ao Conteudo. | | | |
| **Pós-condições:** | | | |
| 1. Salvar o comentário e o Usuario que realizou o comentário para exibir junto ao Conteudo. | | | |
| **Nome:** | adicionarLista | | |
| **Descrição:** | Possibilita ao usuário que cadastre algum Conteudo que ele deseja e agrupe por “Listas” como: conteúdos que ele deseja assistir, conteúdos favoritos, conteúdos que ele deseja receber push ups. | | |
| **Parâmetros** | | | |
| **Nome** | | **Tipo** | **Propósito** |
| usuario | | Usuario | Representa todas as informações do Usuário a ser gerenciado. |
| **Pré-condições:** | | | |
| 1. O usuário deve estar instanciado no sistema. 2. O conteúdo deve estar instanciado no sistema. | | | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | | | |
| 1. Salvar o Conteudo na Lista selecionada pelo usuário. | | | |
| **Pós-condições:** | | | |
| 1. Após inserir o Conteudo na lista do usuário, atualizar a lista para o mesmo. | | | |
| **Nome:** | atualizarNota | | |
| **Descrição:** | Somar uma nota fornecida pelo usuário ao Conteudo e atualizar a sua média. | | |
| **Parâmetros** | | | |
| **Nome** | | **Tipo** | **Propósito** |
| usuario | | Usuario | Representa todas as informações do Usuário a ser gerenciado. |
| n | | int | Representa a nota escolhida pelo usuário para o Conteudo. |
| **Pré-condições:** | | | |
| 1. O usuário deve estar instanciado no sistema.  2. O conteúdo deve estar instanciado no sistema. | | | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | | | |
| 1. Recalcular a nota do Conteudo com a nota fornecida pelo usuário. 2. Salvar a nota dada pelo usuário para o Conteudo. | | | |
| **Pós-condições:** | | | |
| 1. Após levantar recalcular a nota, inseri-la no banco de dados. 2. Inserir também a nota fornecida pelo usuario. | | | |

#### Classe FormPerfil

**Descrição**

A classe FormPerfil é uma classe de fronteira com o usuário que grava em seu aparelho as configurações selecionada por ele.

**Superclasse(s)**

Nenhuma

**Subclasse(s)**

Nenhuma

**Atributos**

|  |
| --- |
| **Nenhum** |

**Operações**

|  |  |
| --- | --- |
| **Nome:** | atualizaConfiguracao |
| **Descrição:** | Salvar no aparelho as configurações desejadas pelo usuário |

#### Classe FormHome

**Descrição**

A classe FormHome é uma classe de fronteira que seria o Menu Principal da aplicação, também é responsável por gravar um log das ações do usuário para traçarmos o seu perfil.

**Superclasse(s)**

Nenhuma

**Subclasse(s)**

Nenhuma

**Atributos**

|  |
| --- |
| **Nenhum** |

**Operações**

|  |  |
| --- | --- |
| **Nome:** | salvaAcoes |
| **Descrição:** | Cadastrar todas ações relevantes que o usuario faz, como filmes mais acessados e paginas mais acessadas. |
| **Pré-condições:** | |
| 1. O usuário deve estar instanciado no sistema. | |
| **Especificação (diagrama de atividade ou pseudocódigo)** | |
| 1. Gravar as ações no banco de dados em uma tabela de log. | |

#### Classes ControleTemporada, ControleFilme, ControleSerie e ControleAnime

**Descrição**

Essas quatro classes são responsáveis apenas por realizar a inserção, obtenção, atualização e deletar dados de suas respectivas entidades.

**Superclasse(s)**

Nenhuma

**Subclasse(s)**

Nenhuma

**Atributos**

|  |
| --- |
| **Nenhum** |

**Operações**

|  |  |
| --- | --- |
| **Nome:** | Inserir, Obter, Atualizar, Excluir |
| **Descrição:** | Tem como função realizar ações no banco de dados. |
| **Pré-condições:** | |
| 1. Seus objetos devem estar instanciados | |

#### Classes FormUsuario, FormRelatorios, FormConteudo e, FormTemporada

**Descrição**

Essas quatro classes são classes de fronteiras sem métodos e atributos, seus objetivos são apresentar de forma gráfica, agradável e sintetizada as informações processadas nas classes de controle.

**Superclasse(s)**

Nenhuma

**Subclasse(s)**

Nenhuma

**Atributos**

|  |
| --- |
| **Nenhum** |

**Operações**

|  |
| --- |
| **Nenhuma** |

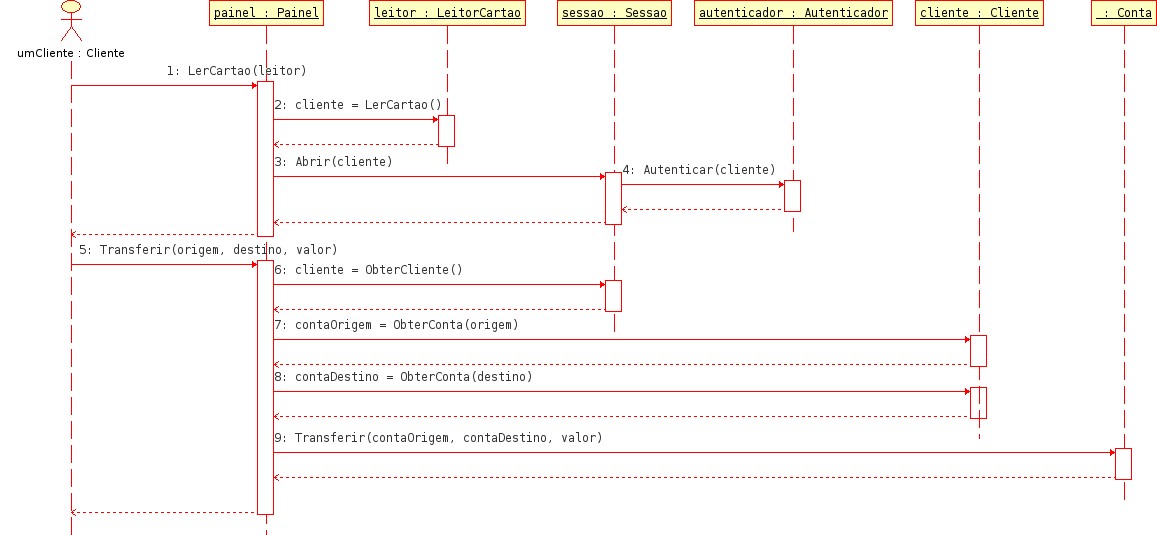
FALTANDO FULL DAQUI PRA BAIXO MAS ACHO QUE O PEDRO TEM UM POUCO FEITO JA

## Realizações de Casos de Uso

O objetivo aqui é demonstrar com mais detalhes como os casos de uso podem ser simulados pelas classes descobertas na Análise. Aqui, cada caso de uso é realizado com objetos das classes de análise determinadas anteriormente. Utilizar diagramas de sequência para representar os cenários de utilização mais importantes do sistema.

### Cenário: Transferir Valor

Por exemplo, imagine que no banco Transferir Valor seja um dos casos de uso. Pode-se comprovar as classes de análise desenvolvidas até então com o seguinte diagrama de sequência:



Pode-se anotar textos explicativos logo após o diagrama, se necessário.

### Cenário: Outra realização …

COLOCAR APENAS CASO EXISTA

# Modelos de Projeto do Sistema

## C:\Users\Duperron Tecnologia\Downloads\O plano\O plano\Fotos\Diagrama de classes atualizado.pngClasses de projeto

São elaboradas a partir das classes de Análise – **domínio do problema** – e de classes existentes em um **domínio de soluçã**o – classes, bibliotecas, componentes reutilizáveis que já estão prontas e que resolvem algum problema específico (por exemplo, classes Date, Time, persistência de objetos com “frameworks” tais como Hibernate, classes de interface homem-máquina como Swing e assim por diante). Utilizar as observações a seguir como um “checklist”:

(Imagem também em anexo para melhor visualização)

Respostas:

Foram importadas as classes Date e sua filha DateTime para servirem como meio de ligação entre a linguagem java e a linguagem SQL do banco de dados, possibilitando a manipulação de horários e datas de forma que sejam usadas por ambas as linguagens.

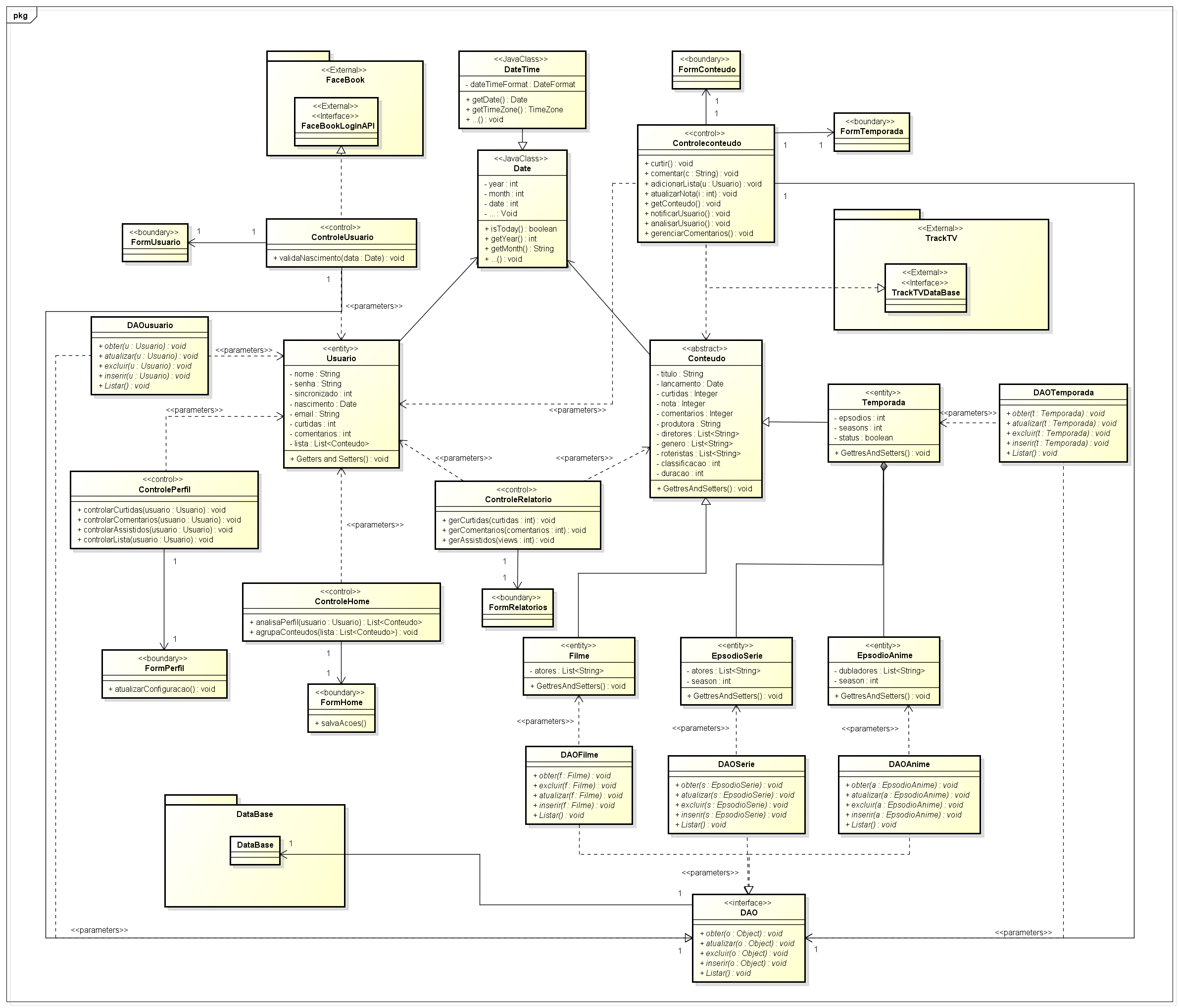
Foi também adicionado a ligação com uma API externa do facebook que permitira ao usuário sincronizar sua conta em nosso sistema com sua conta do facebook para compartilhamento de informações e também como meio de divulgação do aplicativo.

Adicionado outro sistema externo a ser utilizado, a API do site TrackTV, o qual fornecerá os dados necessários dos conteúdos a serem utilizados pelo sistema. Conteúdos como filmes, séries , animes e diversas informações relacionadas aos mesmos.

Implementado também as classes DAO (Data Access Object) responsáveis por realizar funções no banco de dados de acordo com as necessidades do sistema.

E por fim há agora também a representação do banco de dados.

## Refinamento dos relacionamentos

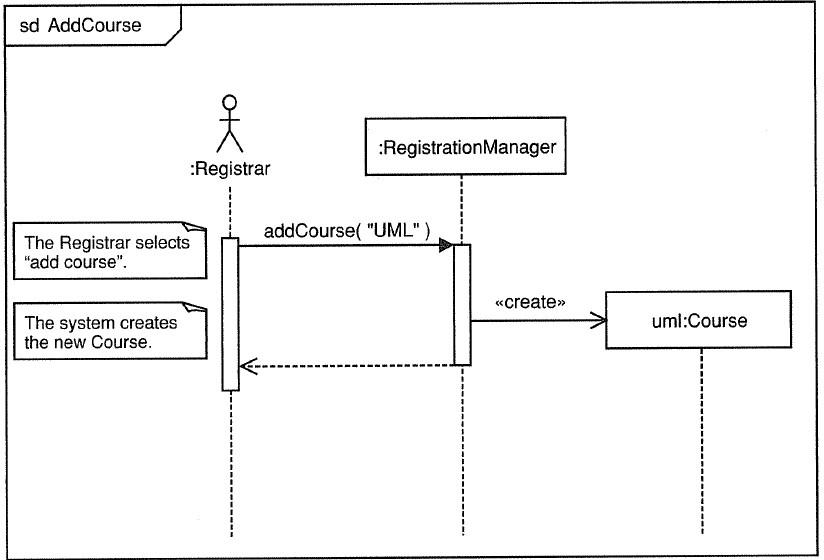


(Imagem também em anexo para melhor visualização)

## Realizações de casos de uso – Projeto

Aqui refinam-se os diagramas de interação da análise ou ainda acrescentam-se outros, para suportar os novos elementos adicionados em projeto, tais como persistência de objetos.

Por exemplo, um cenário de Análise como o apresentado a seguir:



Poderia ser assim detalhado em Projeto, adicionado detalhes de como um objeto será salvo em um banco de dados:

