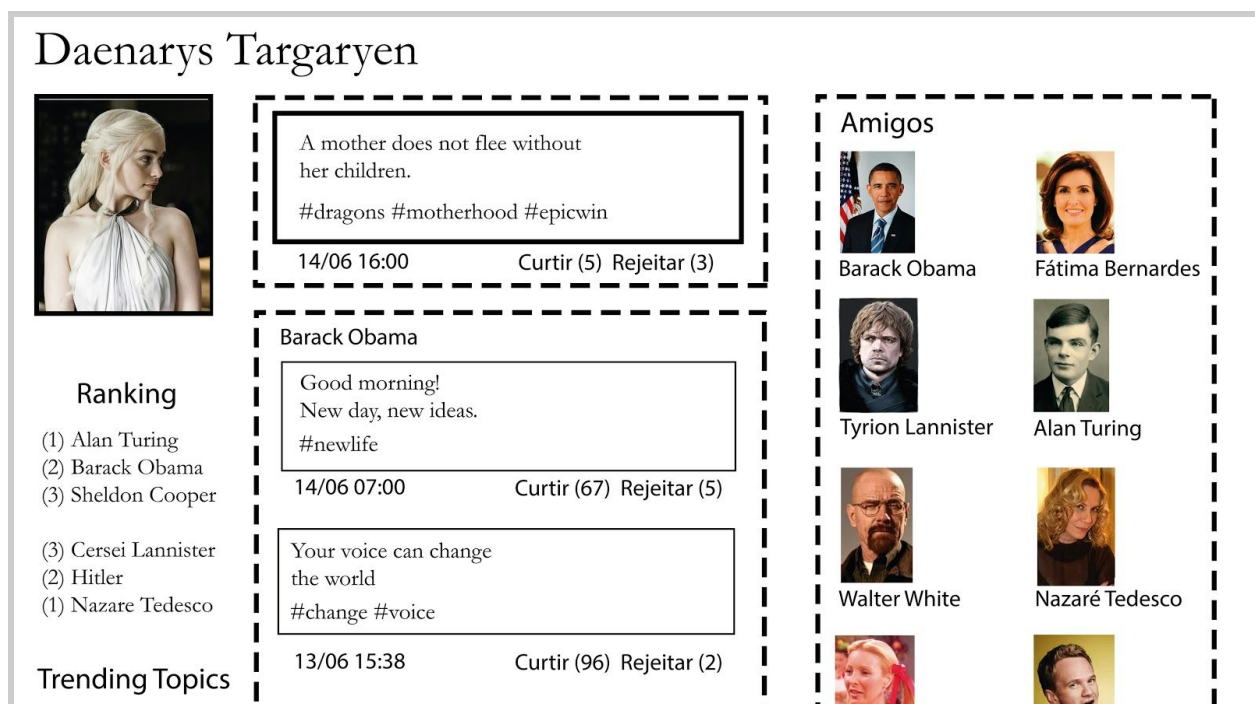


## Projeto da Disciplina - +Pop

### Descrição:

O +Pop (ou Mais Popularidade) é uma **rede social** voltada para a interação entre usuários por meio de diferentes **níveis de popularidade**. O objetivo é fornecer, de forma direta, uma das características principais de qualquer rede social atual: o desejo pela popularidade. Assim como em redes sociais comuns, cada usuário possui uma conta que deve ser, inicialmente, cadastrada usando um endereço e-mail e uma senha. Ao acessar a rede, o usuário pode adicionar outros usuários como amigos, postar mensagens em seu mural, usar hashtags para marcar seus posts, dentre outras possibilidades.



**Figura 1 - Projeto conceitual do +Pop**

Cada usuário terá um nível de popularidade que o tornará mais popular dentre os demais usuários da rede. Use suas habilidades sociais para ficar amigo dos usuários mais influentes e com isso, se tornar uma das pessoas mais populares do +Pop. Os usuários populares são aqueles que chamam mais atenção, pois seus posts ganham mais pontos e determinam os usuários mais populares da

rede. Consequentemente, os demais usuários farão de tudo para ganhar sua amizade e suas curtidas.

### **Implementação:**

Será usada uma **arquitetura** considerando uma Façade e um Controller. As demais entidades do sistema devem ser definidas de acordo com o seu design. Lembre de respeitar o GRASP e de utilizar o máximo de conceitos explorados na disciplina. Serão fornecidos scripts de teste de aceitação que serão usados como um dos critérios de avaliação. Utiliza os scripts fornecidos pelos monitores para criar seus próprios testes de aceitação.

### **Caso de Uso 1: Cadastrar Usuários, Login e Logout**

Como usuário desejo me cadastrar no sistema e realizar login para obter acesso ao +pop. O cadastro de usuários é feito na tela inicial do programa ao fornecer os seus respectivos dados, como: e-mail, senha para acesso, nome, a data de nascimento e uma Imagem correspondente a sua foto (caso o usuário não especifique uma foto, ele(a) terá uma imagem de um avatar default).

Após o cadastro, o usuário cadastrado pode usar seu e-mail e senha para acessar o sistema, assim como, ele ou ela pode realizar logout voltando para a tela inicial do sistema.

### **Caso de Uso 2: Pesquisar e atualizar informações de Usuários**

Como usuário desejo atualizar minhas informações pessoais como nome, e-mail, senha, data de nascimento e foto para disponibilizar minhas informações para meus amigos. Para alterar a senha, é necessário informar a senha atual e a nova senha, de forma que a senha só será alterada se o usuário fornecer realmente a senha atual. Note que essas informações atualizadas serão utilizadas para realizar o login novamente do sistema.

### **Caso de Uso 3: Postar mensagens no Mural**

Como usuário desejo postar mensagens no meu mural para divulgar minhas opiniões e atividades. Um post deve possuir uma popularidade (que começa com zero), o momento em que ele foi criado (i.e. informação de data e hora de criação do post), e um conteúdo composto por vários itens. Esses itens podem ser: texto (máximo de 200 caracteres), áudios, imagens e hashtags. Note que o conteúdo pode ter combinações desses itens: Por exemplo: <mensagem> <imagem>, <imagem> e <audio>. Após o conteúdo (texto, imagens e audios) é que estarão as hashtags, que podem ser várias e que serão, eventualmente, usadas para pesquisa de posts e trending topics.

Os usuários podem curtir (like) ou rejeitar (dislike) um post. Ao curtir, o post ganha pontos positivos de popularidade, e ao ser rejeitado, o post perde pontos de popularidade. Note que isso só poderá ser testado após estabelecer uma amizade (ver Caso de Uso 4).

**OBS:** A Imagem e o áudio serão tratados apenas como Strings que correspondem ao caminho de um arquivo de áudio ou imagem que será carregada pela Interface Gráfica (GUI).

#### Caso de Uso 4: Adicionar e Remover amigos

Como usuário desejo adicionar outros usuários como amigos. O usuário pode enviar solicitações de amizade a outros usuários que podem ser aceitas ou rejeitadas. Ao aceitar o pedido uma amizade é criada entre os dois usuários, e os dois usuários podem curtir/rejeitar os posts um do outro. Ao rejeitar o pedido, a amizade não é criada e o Usuário que enviou a solicitação é notificado. Dica: para isso crie uma entidade Notificações que funciona como uma lista de Strings.

Como usuário desejo remover outros usuários da minha lista de amigos. Note que, nesses casos, a amizade deve ser removida dos dois Usuários.

#### Caso de uso 5: Popularidade

Como usuário desejo acumular popularidade no +pop, para que eu possa ter mais visibilidade na rede. Cada usuário terá um valor de popularidade na rede medida através de pontos denominados “**pops**” (*points of popularity*). A popularidade é obtida por meio das curtidas que ele ou ela receber nos posts que criar. Considere também que essa popularidade **pode ser reduzida** por meio das rejeições dos posts. Porém a curtaida/rejeição não é contada de uma só forma. Os pontos obtidos ao receber uma curtaida dependem do(s) outro(s) usuário(s) que curtiu(ram) seu post. Todo usuário tem um **tipo de popularidade** que pode ser: Normal, CelebridadePOP ou ÍconePOP. A **atribuição de pontos** ocorre da seguinte forma:

- Um usuário **normal** que curtir um post irá adicionar 10 pops no post. A rejeição removerá 10 pops do post.
- Um usuário **CelebridadePOP** que curtir um post irá fornecer 25 pops para o post e 10 pops adicionais se o post for recente, ou seja, se o post foi postado no mesmo dia em que o sistema está sendo executado. A mesma estratégia é usada para reduzir pops se o post for rejeitado.
- Um usuário **ÍconePOP** que curtir um post irá fornecer 50 pops para o post e marcar o post com a hashtag #epicwin para que todo mundo da rede possa ver os postst apoiados pelos ícones. De forma semelhante, o post rejeitado por um ÍconePOP irá perder 50 pops e terá a hashtag #epicfail.

Em resumo, **os usuários mais populares irão fornecer mais pontos que os usuários menos populares**, afinal, visibilidade importa bastante. A popularidade de um usuário é a soma das

popularidades dos posts que ela(e) possuir, que poderá ser acrescido ou diminuída dependendo das curtidas e rejeições dos outros usuários.

A popularidade é utilizada para determinar as pessoas mais influentes da rede. A rede possui um ranking dos 3 usuários mais populares, e os 3 usuários menos populares. Cabe a cada usuário usar suas conexões para subir nesse ranking, assim como, diminuir o ranking dos seus rivais.

**OBS:** Podemos ter futuras modificações nesse caso de uso. Faça um design modularizado, versátil e dinâmico para lidar com a popularidade.

### Caso de Uso 6: Mudança nos Tipos de Usuário

Como usuário desejo mudar meu tipo para poder me tornar um usuário mais popular na rede. O usuário pode mudar seu tipo de popularidade à medida que ganhar ou perder popularidade na rede. **Essa mudança deve ser automática** ao modificar a quantidade de pops do usuário. A quantidade de pops é a seguinte:

- O tipo Normal possui menos que 500 pops.
- O tipo CelebridadePOP possui entre 500 e 1000 pops.
- O tipo ÍconesPOP possui acima de 1000 pops.

### Caso de Uso 7: Ranking de Usuários e Trending Topics

Como usuário desejo ver qual o ranking de usuários e informações mais/menos populares que estão da rede. O sistema deve fornecer algumas informações da rede que devem estar disponíveis para os usuários. Essas informações são: o *ranking* dos 3 usuários mais/menos populares, e o ranking das 3 **hashtags** mais presentes nos posts da rede (*trending topics*). Essas informações são atualizadas a partir da chamada do método 'atualizaRankings' na Façade do projeto. Portanto, não precisa se preocupar em atualizar esse Ranking de forma dinâmica, e sim, apenas na chamada pela Façade.

**OBS:** Em casos de **empate** no ranking de Usuários ou no Trending Topics, o critério para desempate é **a ordem alfabética (decrescente)** do e-mail do usuário e da hashtag. Portanto se:

- Italo (**italo@email.com**) e Manoel (**abc\_manuel@email.com**) possuem 900 pops cada, o usuário com maior ranking será aquele cujo e-mail será maior considerando a ordem alfabética. No caso, **Italo > Manoel**.
- Se **#casa = 3; #familia = 3**; então, **#familia > #casa**.

**Fiquem atentos aos casos com caracteres maiúscula e minúscula.** No caso:

- Se **#soulinda = 3; #souQuemSou = 3**; então **#souQuemSou > #soulinda**. Note que o compareTo default de String resulta um resultado contrário devido à codificação da String.

### Caso de Uso 8: Feed de notícias

Como usuário desejo ter um feed de notícias para observar o que meus amigos populares estão compartilhando. Os usuários possuem um feed de notícias que mostram os posts dos seus amigos na rede, porém a quantidade de posts obtidas dos amigos depende da popularidade de seu amigo. Os usuários do tipo Normal fornecem os 2 posts mais recentes, enquanto que as CelebritiesPOP fornecem 4 posts e os IconsPOP fornecem 6 posts. O feed de notícias é ordenado de acordo com a data de postagem de cada post (do mais recente, para o mais antigo). Forneça também uma outra forma de ordenação do feed de notícias utilizando a popularidade dos posts do feed.

**OBS:** A atualização desses feeds não é automática, é necessário solicitar para o próprio Usuário a atualização de seu próprio feed de notícias.

### Caso de Uso 9: Exportar e Importar Posts por arquivos

Como usuário desejo salvar as informações do meu post para proteger minha contribuição para a rede. Hoje em dia com acusações e riscos das informações divulgadas na Internet, o +Pop permite que seus usuários salvem o seu histórico de Posts em arquivos de texto. Para isso, cada post é armazenado seguindo o seguinte template:

```
Post #1 <Dia e Hora>
Conteúdo:
<Texto>
<nome do arquivo da imagem>
<nome do arquivo de áudio>
<conjunto de hashtags>
+Pop: <pontos de popularidade adquirida>
```

**OBS:**

- Todos os posts são salvos em um mesmo arquivo.
- Cada post é enumerado de acordo com a ordem em que foi criado, usando a notação #1, #2, ... , #n.

### Caso de Uso 10: Persistência em arquivos

Como usuário desejo que as informações que alterei no meu sistema persistam ao realizar logout e sejam carregadas ao realizar login. As informações da rede devem persistir ao fechar o sistema. Para isso use arquivos, **não serão aceitos projetos com Banco de Dados** (não é escopo da disciplina). O uso de **arquivos** deve ser feito com Streaming de Objetos.

### Considerações Importantes:

- **Realize o tratamento adequado de Exceptions**, considerando uma hierarquia de exceções voltada para o tratamento de Exceptions checked e, se for o caso, Exceptions unchecked.
- Os testes fornecidos pelos professores e monitores serão testes de aceitação. Aumente a qualidade de seu código **criando testes de unidade pelo JUnit**, e também, seus próprios scripts de teste de aceitação. Apesar de ser um critério **extra** para a avaliação, os testes de unidade (e adicionais de aceitação) terão um efeito importante para determinar se seu código está bem feito, e se atinge as funcionalidades de todos os casos de uso.
- **O projeto foi criado para explorar todos os conteúdos vistos na disciplina.** Portanto, fique atento a cada trecho dele. Uma dica é enviar com o projeto um arquivo texto (README) detalhando, em seu design, onde está cada assunto visto na disciplina como por exemplo: Herança, as distribuições de responsabilidades (Expert, Creator, Alta coesão, etc.), Composição, Coleções (quais e por quê), dentre outros.