



Proposta Comercial





1. Descrição do problema

Uma das mídias de entretenimento que cresce desde o último quartel é o jogo de interpretação de papéis (RPG). Uma das modalidades de jogo utiliza a exploração de uma malha social para o desenrolar da história, apresentando aos jogadores encontros com personagens não jogadores (NPCs) que se conectam entre si dentro dessa malha social.

No entanto, é uma dificuldade notável a construção dessa malha social manualmente, principalmente mantendo condições coesas. Idealmente essa malha social deve conter comunidades, i.e. grupos de NPCs em que a chance de se conhecerem é maior do que a chance de conhecerem alguém de fora da comunidade). Essas comunidades devem apresentar algum nível de coesão interna, com indivíduos que apresentam características semelhantes ou compatíveis. No entanto, deve haver um mínimo de diversidade dentro das comunidades, bem como relacionamento inter comunitários, para evitar pontos no jogo em que a continuação da história é dificultada.

Atualmente esse processo é feito de forma manual, descrevendo textualmente a malha social. Isso é feito descrevendo cada um dos personagens e as ligações que eles tem entre si. Esse processo é realizado muitas vezes de forma mental, sem a produção de um artefato para consultas posteriores. Isso cria uma limitação no tamanho das malhas sociais que podem ser usadas.

2. Soluções do mercado para o problema

Não foram localizadas ferramentas disponíveis para a execução automatizada desse processo.

3. Projetos acadêmicos correlatos

- <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0122777>
 - Largeron, C., Mougél, P. N., Rabbany, R., & Zaiane, O. R. (2015). Generating attributed networks with communities. *PloS one*, 10(4), e0122777.
- https://mpacer.org/s/causal_bayesian_networkx.pdf
 - Pacer, M. D. (2015). Causal-Bayesian-NetworkX.

4. Descrição da solução concebida

A proposta de solução é um sistema para a geração de malhas sociais. Composto de uma interface de configuração, uma interface de geração, e uma interface de consulta. Essas interfaces interagirão



com a implementação do algoritmo proposto por Largeron (2015), adaptado para considerar atributos configuráveis (descrito em mais detalhes mais a baixo).

A interface de configuração disponibilizada para o administrador possibilita o cadastro dos parâmetros para o algoritmo, bem como o cadastro de quais serão os atributos considerados e quais são os valores possíveis para esses atributos.

A interface de geração disponibiliza formulários dinâmicos para o preenchimento de uma combinatória entre as opções dentro do atributo cadastrado pelo administrador. O cadastro dessa análise combinatória define qual é o fator de distanciamento entre dois personagens que possuam esse atributo preenchido com esses valores. Depois do preenchimento, é disponibilizada a geração da malha social, nesse processo, os atributos são preenchidos pelo algoritmo, e com base nesses o algoritmo determina quais são as relações entre os personagens.

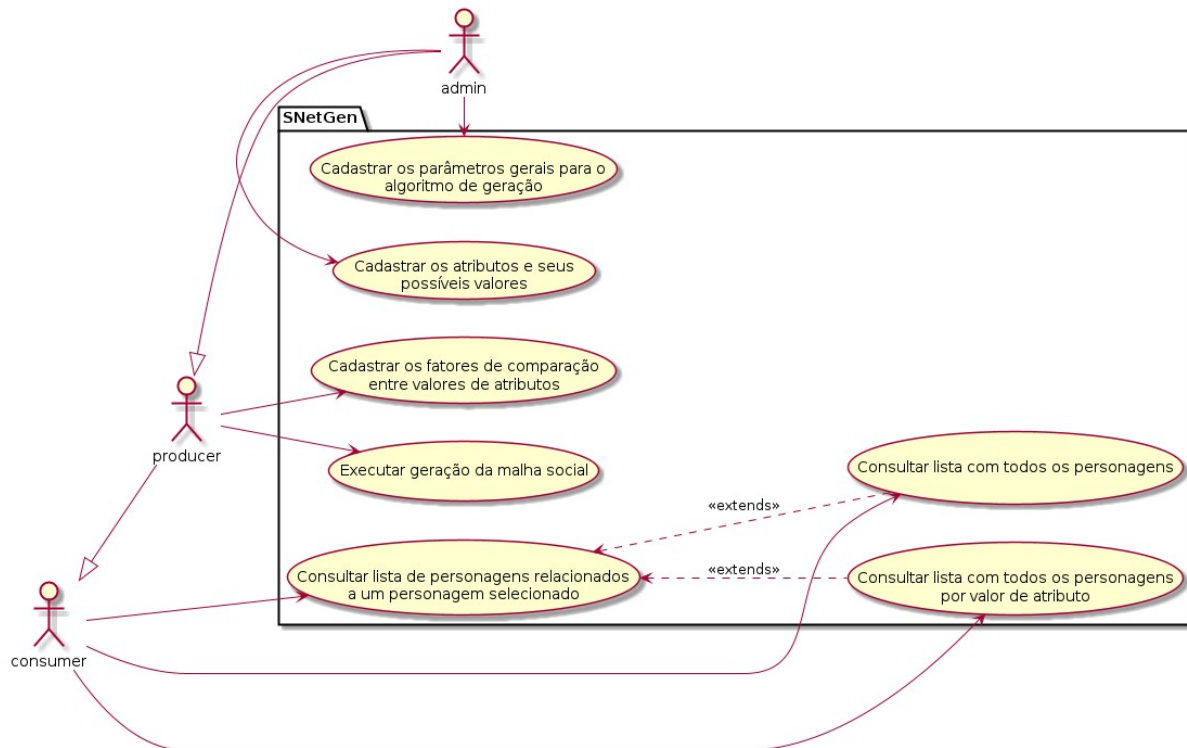
A interface de consulta disponibiliza uma navegabilidade pelo grafo, possibilitando listar os personagens gerados, listar por atributo, e uma vez selecionado um personagem, listar os personagens que possuem ligação com ele.

A respeito da adaptação do algoritmo proposto por Largeron (2015): o processo proposto originalmente utiliza os atributos para calcular as distâncias entre diferentes nodos dentro do grafo e usa essa distância para determinar a compatibilidade de dois nodos, isso é, a chance de esses dois nodos possuírem uma ligação. Isso apresenta alguns problemas que necessitam de adaptação, como esses atributos são uma distribuição normal dentro de um espaço cartesiano, é sabido que a distância entre três nodos A, B e C, é sempre tal que a distância mais longa não supera a soma das duas anteriores. Isso precisa ser tratado para que seja viável a construção de um grafo onde um nodo, personagem, tenha uma distância entre dois outros que seja menor que a distância entre esses dois outros em si. É necessária essa adaptação para que seja possível que um personagem se ligue a dois que não compartilhariam características nenhuma.

A implementação para a solução envolve a customização de quais são os atributos utilizados (e seus valores) pelo administrador do sistema. E a customização da distância entre cada um dos valores para cada atributo, pelo usuário que gerará a malha social. Para os processos no algoritmo original que utilizariam o valor da distância entre dois vertices será considerada a soma dos valores dos pares de cada atributo.



5. Diagrama de casos de uso



6. Indicativo da tecnologia

Para o desenvolvimento do sistema, será utilizado a linguagem de programação python, utilizando as bibliotecas disponíveis para trabalhar com grafos, bem como Django e outras bibliotecas para o desenvolvimento da interface de usuário.

Para a persistência dos dados, será utilizado diretamente o sistema de arquivos acessível ao servidor, gerando arquivos JSON para a descrição dos dados.

7. Justificativa para escolha da tecnologia

A linguagem, bem como algumas das bibliotecas que foram selecionadas, são ferramentas já conhecidas pelos membros da equipe e eficientes no tratamento de grafos. A utilização da plataforma Django para o desenvolvimento facilita a criação de uma interface.

O uso dos arquivos JSON para o armazenamento dos dados facilita o mapeamento entre os objetos e a persistência nos casos de cadastros dinâmicos.



8. Requisitos funcionais

- RF01 – O sistema deverá possibilitar o cadastro dos parâmetros para o algoritmo para o administrador do sistema (CRUD).
- RF02 – O sistema deverá possibilitar o cadastro dos atributos e dos valores possíveis para cada atributo para o administrador do sistema (CRUD).
- RF03 – O sistema deverá possibilitar o cadastro dos fatores de distanciamento entre os pares de valores de cada atributo para o usuário responsável pela geração do grafo (CRUD).
- RF04 – O sistema deverá possibilitar a execução do algoritmo de geração da malha social para o usuário responsável por essa geração (processo/back end).
- RF05 – O sistema deverá possibilitar a navegação de um grafo gerado para os usuário que receberem acesso a esse grafo disponibilizando a lista de todos os personagens bem como os personagens agrupados por valor de atributo (web/report).
- RF06 – O sistema deverá possibilitar a navegação de um grafo gerado para os usuário que receberem acesso a esse grafo disponibilizando a lista dos personagens conhecidos por um determinado personagem (web/report).

9. Regras de negócio

- RN01 – No acesso à página de configuração é requisitado o login de administrador (administrador).
- RN02 – Acessando a página de configuração dos atributos é necessário preencher pelo menos dois valores possíveis para cada atributo cadastrado antes de concluir o cadastro (administrador).
- RN03 – Acessando a página de configuração dos parâmetros do algoritmo é necessário preencher todos os valores no formulário antes de concluir o cadastro (administrador).
- RN04 – Acessando a página de cadastro dos fatores de distanciamento é necessário preencher o fator para cada par de valores de cada atributo para acessar a página de geração da malha social (usuário produtor).
- RN05 – Acessando a página de geração da malha, e executando a geração, é disponibilizado um link para acessar a navegação da malha gerada para compartilhamento (usuário produtor).



- RN06 - Acessando a página de navegação de uma malha social gerada devem ser apresentadas as listas com todos os personagens e a lista com todos os personagens agrupados por valor de atributo (usuário navegador).
- RN07 - Acessando a página de navegação e selecionando um personagem, deve ser apresentado o conjunto de atributos dele e a lista de todos os personagens que possuem ligação com o personagem selecionado (usuário navegador).

10. Projeto das telas/interfaces do Sistema

a. Entrada dos parâmetros do algoritmo:

Dados Gerais	
n	<input type="text" value="150"/>
max_wth	<input type="text" value="20"/>
max_btw	<input type="text" value="10"/>
mte	<input type="text" value="500"/>
k	<input type="text" value="5"/>
teta	<input type="text" value="0.01"/>
nbRep	<input type="text" value="5"/>
<input type="button" value="OK"/>	

b. Cadastro dos atributos:

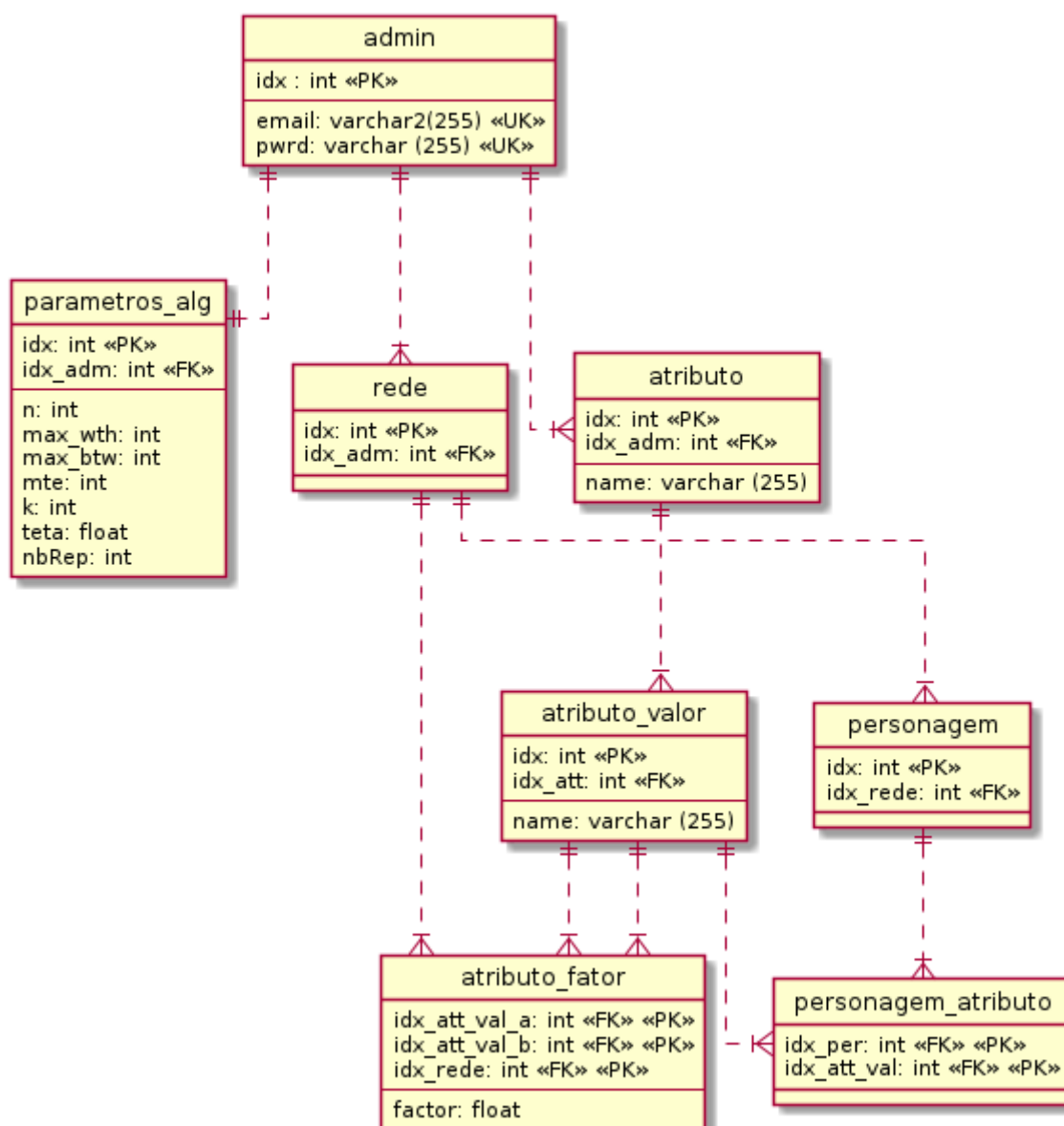
cadastro de atributos	
<input type="text" value="Classe social"/>	<input type="text" value="Rico"/>
<input type="text"/>	<input type="text" value="Classe média"/>
<input type="button" value="Add valor"/>	
<input type="text" value="Etnia"/>	<input type="text" value="Calcasiano"/>
<input type="text"/>	<input type="text" value="Pardo"/>
<input type="text"/>	<input type="text" value="Negro"/>
<input type="button" value="Add valor"/>	
<input type="button" value="Add attribute"/>	
<input type="button" value="OK"/>	



c. Cadastro de fatores de distanciamento:

Fatores do atributo 'Característica psicológica'				
-	Controlador	Submisso	Afável	Paciente
Controlador	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>
Submisso	-	<u>0</u>	<u>0</u>	<u>0</u>
Afável	-	-	<u>0</u>	<u>0</u>
Paciente	-	-	-	<u>0</u>
<input type="button" value="OK"/>				

d. Modelo de dados:





Nota: esse modelo de dados será representado em uma estrutura de arquivos.

e. Tela de consulta da malha social:

consulta de dos personagens

☒ todos
 ☐ por atributo

att_1

▼

val

▼

☐ por seleção

idx

▼

0 - att_1:val; att_2:val; att_3:val

Selecionar

1 - att_1:val_alternativo; att_2:val; att_3:val

Selecionar

2 - att_1:val; att_2:val_alternativo; att_3:val

Selecionar

3 - att_1:val; att_2:val; att_3:val_alternativo

Selecionar

11. Cronograma para execução do projeto (2020-2)

2020-09-25 - aula06 desenvolvimento do projeto	
Atividades previstas	Atividades realizadas
Implementação do algorimo conforme proposto no artigo correlato (parte 1 de 2)	
2020-10-02 - aula07 desenvolvimento do projeto	
Atividades previstas	Atividades realizadas
Implementação do algorimo conforme proposto no artigo correlato (parte 2 de 2)	
2020-10-09 - aula08 desenvolvimento do projeto	
Atividades previstas	Atividades realizadas
Desenvolvimento da interface de entrada de dados e persistência (parte 1 de 2)	
2020-10-16 - aula09 Avaliacao1	
Atividades previstas	Atividades realizadas
Desenvolvimento da interface de entrada de dados e persistência (parte 2 de 2)	



2020-10-23 - aula10 desenvolvimento do projeto	
Atividades previstas	Atividades realizadas
Adaptação do algoritmo para as regras de negócio (clusterização).	
2020-10-30 - aula11 desenvolvimento do projeto	
Atividades previstas	Atividades realizadas
Adaptação do algoritmo para as regras de negócio (distacia).	
2020-11-06 - aula12 desenvolvimento do projeto	
Atividades previstas	Atividades realizadas
Revisão e melhoria do código	
2020-11-13 - aula13 Avaliacao2	
Atividades previstas	Atividades realizadas
Interface de consulta (parte 1 de 2)	
2020-11-20 - aula14 desenvolvimento do projeto	
Atividades previstas	Atividades realizadas
Interface de consulta (parte 2 de 2)	
2020-11-27 - aula15 Apresentação ao prof e produção vídeo	
Atividades previstas	Atividades realizadas
Testes e revalidações	
2020-12-04 - aula16 Apresentação Final à turma	
Atividades previstas	Atividades realizadas
Testes e revalidações	

12. Perfil técnico da equipe construtora

Exemplo: A equipe será formada por 1 programador, conforme detalhes abaixo:

Programador 1

Formação: Bacharelado em Ciências da Computação;

Experiência: Trabalha na Philips Clinical Informatics

Conhecimentos: Java, Delphi, Python, Javascript, bash script, sql, pl/sql;

Gustavo Henrique Spiess



Equipe: 9

13. Link para o repositório do projeto no GIT

- https://github.com/gustavospieess/bcc_2020_2_prjsft2/
- `git@github.com:gustavospieess/bcc_2020_2_prjsft2.git`