

Usuário	
<ul style="list-style-type: none">• nome;• email;• id_usuario;• //a classe irá obter funções que garantam a validação dos dados;	<ul style="list-style-type: none">• Gerenciar Usuários;

Aluno		Usuario
<ul style="list-style-type: none">• interesses• materias	<ul style="list-style-type: none">• Gerenciar Post's; //para colocar dúvidas;	

Monitor		Usuario
<ul style="list-style-type: none">• id_publico_monitor;• area_de_atuacao;• avaliacao;	<ul style="list-style-type: none">• Gerenciar Respostas;	

Professor_adm		Usuario
<ul style="list-style-type: none">• validar_monitor(id_publico_monitor)	<ul style="list-style-type: none">• Repositório Usuários;	

Post	
<ul style="list-style-type: none"> • titulo; • materia; • descricao; • nome_usuario; • followed_by_many(nome_usuario); //importante para deixar em destaque os post com mais follows; • id_post; • id_usuario; 	

Resposta		Post
<ul style="list-style-type: none"> • avaliacao; • id_post; //identidade do post respondido; • id_resposta; //semelhante ao "id_post", ainda irá funcionar com as funções que gerenciam os posts, porém será possível identificar a mudança no id o que implica em um tratamento especial a classe; 		

Gerenciar Usuários	
<ul style="list-style-type: none"> • encontrar_usuario(email, senha); //retorna um objeto do tipo usuário desejado, utiliza das funções da classe "Repositório Usuário" para isso; • encontrar_usuarios(filtros); //solicita uma busca de usuários com determinada característica ao Repositório Usuários, retorna uma lista de objetos do tipo Usuário; • editar_informacoes(id_usuario, novas_info); //altera as informações relacionadas a determinado usuário; • avaliar_monitor(id_publico_monitor, avaliacao); //altera o campo avaliação do monitor identificado pelo "id_publico_monitor"; • obter_posts(id_usuario); //retorna uma lista com todos os posts associado a um usuário; 	<ul style="list-style-type: none"> • Repositório Usuários

Gerenciar Post's

- `denunciar_postagem(id_post, denuncia);`
- `new_post(id_usuario, nome_usuario, descricao, materia, titulo);`
- `edit_post(id_post, id_usuario, novas_info);`
- `delete_post(id_post, id_usuario);`
- `follow_post(id_post, id_usuario);`
- `unfollow_post(id_post, id_usuario);`

- Gerenciador Usuários;
- Repositório Post's;

Gerenciar Respostas

- `nova_reposta(id_usuario, id_post, nome_monitor, descricao, materia, titulo);`
- `edit_post(id_resposta, id_usuario, novas_info);`
- `delete_resposta(id_resposta, id_usuario);`

- Gerenciador Usuários;
- Repositório Post's;

search_feature

- `topico`
- `palavra_chave`
- `materia`
- `buscar();` //retorna uma lista de objetos do tipo "Post" aos moldes escolhidos;

- Repositório Post's;

Notificação

- `Usuario;`
- `mensagem_notificacao;`
- `notificar();`

Interface Repositório Usuários	
<ul style="list-style-type: none">• <code>criar_usuario(email, senha, outras_info);</code> //cria uma nova repartição no arquivo que gere os usuários com uma identidade única; (retorno personalizado a cada tipo de usuário);• <code>remover_usuario(id_usuario);</code> //apaga a repartição identificada pelo "id_usuario" do arquivo que gere os usuários;• <code>identidade_usuario(email, senha);</code> //retorna o "id_usuario" atrelado ao usuário com tal login;• <code>autenticar_login(email, senha)</code> //retorna o "true" caso o usuário exista;• <code>dados_usuario(id_usuario);</code> //retorna uma string formatada com os dados essenciais para construir a instancia do usuário desejado;• <code>buscar_usuarios(filtros);</code> //retorna uma string formatada com os dados dos usuários encontrados;• <code>editar_usuario(id_usuario, novas_info);</code> //altera as informações de determinado usuário;	

Interface Repositório Post's	
<ul style="list-style-type: none">• <code>criar_post(Post);</code> //recebe um objeto do tipo "Post" e cria uma partição para ele no arquivo responsável por gerir os post's, além disso adiciona um identificador exclusivo a esse post e o adiciona as informações do usuário que o criou em sua repartição no arquivo do repositório usuários;• <code>remover_post(id_post);</code> //remove um post do arquivo que gere os mesmos;• <code>editar_post(id_post, nova_info);</code> //altera as informações de determinado post;• <code>buscar_posts(filtros);</code> //retorna uma string formatada com os dados dos post's encontrados;	<ul style="list-style-type: none">• Repositório Usuários;

Repositório Respostas	
<ul style="list-style-type: none">• criar_resposta(Resposta);• remover_resposta(id_resposta);• editar_resposta(id_resposta, novo_texto);• buscar_resposta(filtros);	<ul style="list-style-type: none">• Gerenciar Respostas