

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

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

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

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

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

Resposta		Post
<ul style="list-style-type: none"><li>• avaliacao;</li><li>• id_post; //identidade do post respondido;</li><li>• 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;</li></ul>		

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

### 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_resposta(id_usuario, id_post, nome_monitor, descricao, materia, titulo);`
- `editar_resposta(id_resposta, id_usuario, novas_info);`
- `delete_resposta(id_resposta, id_usuario);`

- Gerenciador Usuários;
- Repositório Post's;

### Ferramenta de Busca

- `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"><li>• <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);</li><li>• <code>remover_usuario(id_usuario);</code> //apaga a repartição identificada pelo "id_usuario" do arquivo que gere os usuários;</li><li>• <code>identidade_usuario(email, senha);</code> //retorna o "id_usuario" atrelado ao usuário com tal login;</li><li>• <code>autenticar_login(email, senha)</code> //retorna o "true" caso o usuário exista;</li><li>• <code>dados_usuario(id_usuario);</code> //retorna uma string formatada com os dados essenciais para construir a instancia do usuário desejado;</li><li>• <code>buscar_usuarios(filtros);</code> //retorna uma string formatada com os dados dos usuários encontrados;</li><li>• <code>editar_usuario(id_usuario, novas_info);</code> //altera as informações de determinado usuário;</li></ul>	

Interface	
Repositório Post's	
<ul style="list-style-type: none"><li>• <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;</li><li>• <code>remover_post(id_post);</code> //remove um post do arquivo;</li><li>• <code>editar_post(id_post, nova_info);</code> //altera as informações de determinado post;</li><li>• <code>buscar_posts(filtros);</code> //retorna uma string formatada com os dados dos post's encontrados;</li></ul>	<ul style="list-style-type: none"><li>• Repositório Usuários;</li></ul>