

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">• (por enquanto apenas um usuario comum)	<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;	<ul style="list-style-type: none">• Gerenciar Respostas;	

Post	
<ul style="list-style-type: none">• titulo;• materia;• descricao;• nome_usuario;• id_post;• id_usuario;	

Gerenciar Usuários

- encontrar_usuario(email, senha); //retorna um objeto do tipo usuário desejado, utiliza das funções da classe "Repositório Usuário" para isso;
- editar_informacoes(id_usuario, novas_info); //altera as informações relacionadas a determinado usuário;
- obter_posts(id_usuario); //retorna uma lista com todos os posts associado a um usuário;

- Repositório Usuários

Gerenciar Post's

- new_post(id_usuario, nome_usuario, descricao, materia, titulo);
- edit_post(id_post, id_usuario, novas_info);
- delete_post(id_post, id_usuario);

- Gerenciador Usuários;
- Repositório Post's;

Interface

Repositório Usuários

- criar_usuario(email, senha, outras_info); //cria uma nova repartição no arquivo que gere os usuários com uma identidade única; (retorno personalizado a cada tipo de usuário);
- remover_usuario(id_usuario); //apaga a repartição identificada pelo "id_usuario" do arquivo que gere os usuários;
- identidade_usuario(email, senha); //retorna o "id_usuario" atrelado ao usuário com tal login;
- autenticar_login(email, senha) //retorna o "true" caso o usuário exista;
- dados_usuario(id_usuario); //retorna uma string formatada com os dados essenciais para construir a instancia do usuário desejado;
- editar_usuario(id_usuario, novas_info); //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;