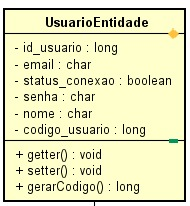
**DESCRIÇÃO DAS FUNCIONALIDADES**

**Classe Usuário Entidade**



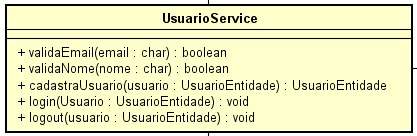
**Campos**:

1. **Id\_Usuario**: Primary Key
2. **email:** email do usuário
3. **status\_conexao:** informa se o usuário está conectado ou desconectado
4. **senha:** chave de acesso do usuário
5. **nome:** nickname
6. **codigo\_usuario:** é código de identificação do usuário para toda a plataforma, ele será usado pois o id não será visível para os usuários, apenas para fins de aplicação e controle

**Métodos**

1. **getters e setters**, são os acessos da classe usuário
2. **gerarCodigo:** método que gera o código do usuário

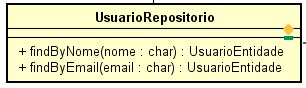
**Classe Usuário Service**



**Campos**:

1. **validaEmail:** verifica se o email é bem informado - possui um domínio
2. **validaNome:** ver se dentro da base de dados o nome já existe
3. **cadastraUsuario:** salva um usuário na base de dados
4. **login:** coloca a variável status conexão como verdadeiro
5. **logout:** coloca a variável status conexão como falso

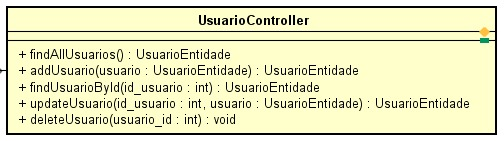
**Classe Usuário Repositório**



**Campos**:

1. **findByNome:** procura o usuário no banco de dados pelo nome
2. **findByEmail:** procura o usuário no banco de dados pelo email

**Classe Controller**



**Campos**:

1. **findAllUsuarios:** lista todos os usuário
2. **addUsuario:** criar um usuário
3. **findUsuario:** lista usuário específico
4. **updateUsuario:** faz uma atualização cadastral
5. **deleteUsuario:** delata as informações do usuário