**Aluna: Maria Lorena Muralha Lima**

**Matéria: DOO – Avaliação 2**

***EXTRA – TRABALHO DE DOO 2***

**Fora o modelo MVC implementado no trabalho, outro extra que foi adicionado foi a forma de validação de email utilizando padrões reais de email.**

**Basicamente, existem padrões de caracteres que fazer um email ser válido ou não. Esses padrões são, por exemplo, a presença do “@” no conjunto de caracteres. Ou a presença de, após o @, uma sequência de caracteres seguida de um “.” (exemplo:** [**aluno@email.com**](mailto:aluno@email.com)**). Com isso em mente, foi utilizado a classe *Parttern* do java, que tem a função de verificar se uma sequência de caracteres segue o padrão definido nela.**

**Como no código abaixo:**

**public static boolean patternMatches(String emailAddress, String regexPattern) {**

**return Pattern.compile(regexPattern).matcher(emailAddress).matches();**

**}**

**Caso o padrão “*regexPattern”* seja respeitado no parâmetro *emailAddress,* é retornado true.**

**No caso do código enviado, foi utilizado vários *regexPattern* (padrões de email), que fazem com que o usuário DEVA inserir um endereço de email válido dentro desses padrões.**

**Exemplos de emails válidos:**

* [**professor@gmail.com**](mailto:professor@gmail.com)
* [**Fulano23@email.com**](mailto:Fulano23@email.com)

**Um exemplo de email inválido**

* professor@.com
* .user.name@domain.com

**Todos os padrões utilizados no trabalho estão descritos aqui:**

**https://www.baeldung.com/java-email-validation-regex**