|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FGA 0238 *-* Testes de Software – Turma:** | | T01 | **Semestre:** | 2023.2 |
| **Nome:** | Guilherme Soares Rocha | | **Matrícula:** | 211039789 |
| **Equipe:** | Testeiros de Jeová | |  |  |

Atividade 4 – TDD

# **Funcionalidade**

* 1. **Identificação da Issue**

Issue #95.

Link: <https://gitlab.com/lappis-unb/projects/mec-energia/mec-energia-api/-/issues/95>

* 1. **Especificação**

Atualmente o sistema tem o processo de envio de email para processos vinculados a senha do usuário:

* Criação de conta nova, cria a senha;
* Alteração de senha.

Sendo assim, o sistema hoje só consegue verificar se o email foi enviado. Mas não consegue garantir se o email do usuário existe de fato ou se o email chegou com sucesso.

* 1. **Descrição da Funcionalidade**

Será implementado uma funcionalidade para verificar se os formatos de emails enviados são válidos, de acordo com regras comuns para emails, como ter um “@” por exemplo.

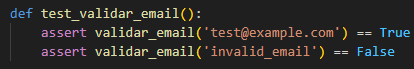
* 1. **Ciclos**

Foi feito um ciclo para atender às exigências, de acordo com o que era necessário. Neste ciclo foi implementado o teste e a funcionalidade que contém uma expressão regular (regex) para validar emails.

# **Execução**

**2.1 Primeiro Ciclo**

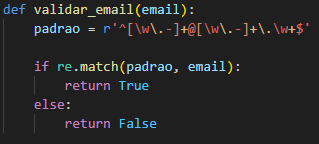
Criação do teste para verificar se a validação de email está sendo feita corretamente.



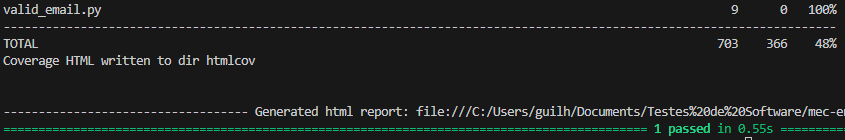
Teste falhando:



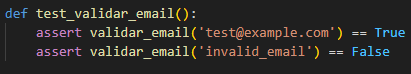
Com a criação do seguinte código:



O teste foi aprovado e passou, como mostra a imagem:

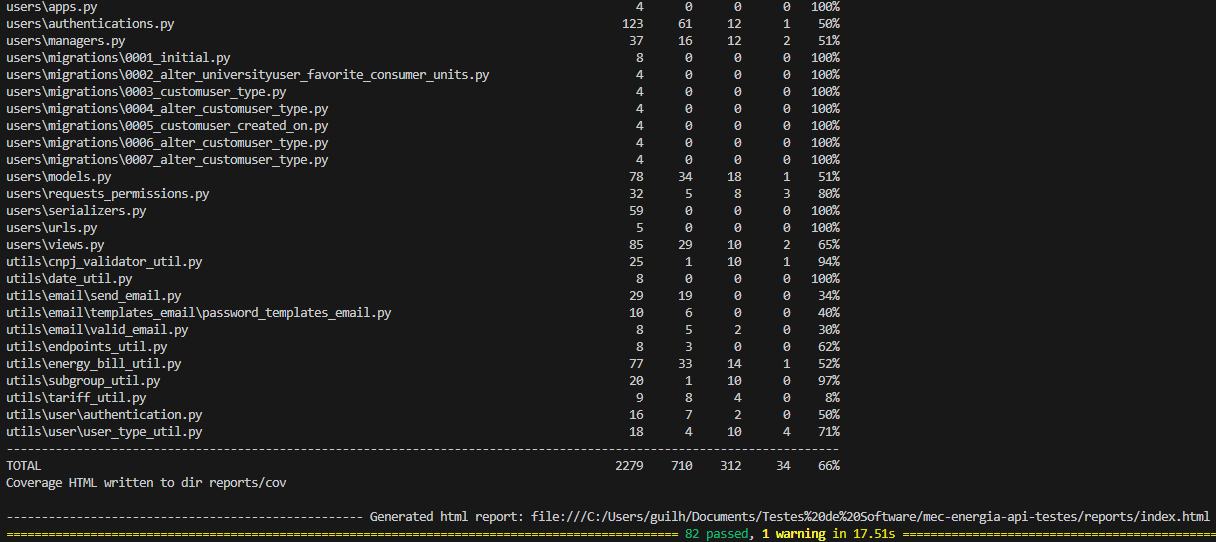


# Código Fonte Testes

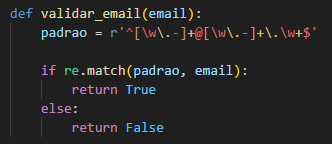


<https://gitlab.com/GuilhermeSoaress/mec-energia-api-testes/-/blob/develop/utils/email/valid_email.py?ref_type=heads>

# Resultado Final Execução Testes



# Código Fonte da Funcionalidade Implementada



<https://gitlab.com/GuilhermeSoaress/mec-energia-api-testes/-/blob/develop/utils/email/valid_email.py?ref_type=heads>

# Pull Request

Não foi feito pull request.

# Conclusão

Desenvolver com TDD foi uma maneira nova de programar. Inicialmente desafiador, o ciclo de escrever testes antes do código proporcionou uma compreensão mais profunda dos requisitos. A abordagem resultou em código confiável, fácil de manter e uma perspectiva aprimorada sobre qualidade de software.