**Aluna: Ana Caroline da Rocha Braz**

**Matricula: 212008482**

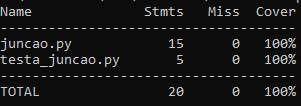
**Informações úteis:**

* **Foi utilizado o pytest para realizar os testes, o Coverage.py para a verificar a cobertura e o Sphinx para a documentação dos códigos realizados em pyhton.**
* **Para rodar o programa basta deixar os arquivos juncao.py, testa\_juncao.py, os arquivos testes descrito na função que deseja testar no mesmo diretório e no terminal utilizar “coverage run -m pytest testa\_juncao.py”, assim irá realizar os testes. E para visualizar a cobertura “coverage report”.**
* **Importante lembrar que para usar uma função teste deve-se comentar todas as outras pois a função principal cria apenas 1 arquivo final. Caso duas funções testes estejam sendo testadas ao mesmo tempo, apenas a última terá o resultado no arquivo 3.**
* **Foi utilizado os slides de teste como base para realizar os testes, sendo assim as strings também foram utilizadas de acordo com os slides.**
* **A documentação se entra na pasta html -> Documentação.html**
* **Foram feitos testes gerais, sendo assim passando nesses testes quaisquer tipo de arquivos ordenados serão intercalados e ordenados no arquivo final.**

**Testes realizados:**

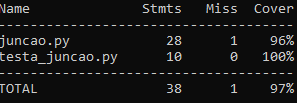
**Teste 1 – Verifica o caminho (ABECD): Abrir, leitura, fechar**

Para o primeiro teste foi feito apenas a abertura dos arquivos, a leitura e o fechamento dos arquivos vazios. Ao passar pelo Coverage.py foi obtido a seguinte cobertura:



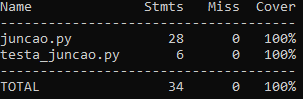
**Teste 2 – Verifica o caminho (ABECFID): Arquivo 1 com 1 elemento e arquivo 2 com 0 elementos**

Ao passar pelo Coverage.py foi obtido a seguinte cobertura:



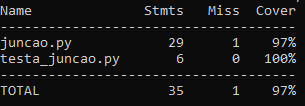
**Teste 3 – Verifica o caminho (ABECFHD): Arquivo 1 com 1 elemento e arquivo 2 com 1 elemento, ambos elementos dos arquivos iguais**

Ao passar pelo Coverage.py foi obtido a seguinte cobertura:



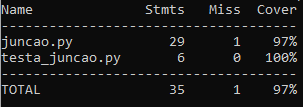
**Teste 4 – Verifica o caminho (ABECFGD): Arquivo 1 vazio e arquivo 2 com 1 elemento**

Ao passar pelo Coverage.py foi obtido a seguinte cobertura:



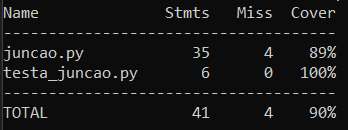
**Teste 5 – Verifica o caminho (ABECFIFID): Arquivo 1 com 2 elemento e arquivo 2 vazio**

Ao passar pelo Coverage.py foi obtido a seguinte cobertura:



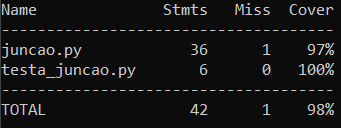
**Teste 6 – Verifica o caminho (ABECFIFID): Arquivo 1 com 2 elemento e arquivo 2 com 1 elemento igual do arquivo 1**

Ao passar pelo Coverage.py foi obtido a seguinte cobertura:

****

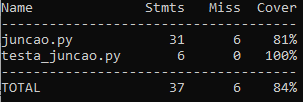
**Teste 7 – Verifica o caminho (ABECFGFID): Arquivo 1 com 2 elemento e arquivo 2 com 1 elemento maior do que o elemento correspondente da linha do arquivo 1**

Ao passar pelo Coverage.py foi obtido a seguinte cobertura:



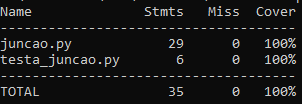
**Teste 8 – Verifica o caminho (ABECFIFID): Arquivo 1 com 2 elemento e arquivo 2 com 1 elemento menor do que o elemento corresponde da linha do arquivo 1**

Ao passar pelo Coverage.py foi obtido a seguinte cobertura:

****

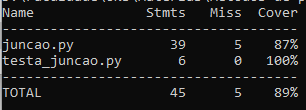
**Teste 9 – Verifica o caminho (ABECFHFHD) – Arquivo 1 com 2 elemento e arquivo 2 com 2 elementos iguais ao do arquivo 1**

Ao passar pelo Coverage.py foi obtido a seguinte cobertura:



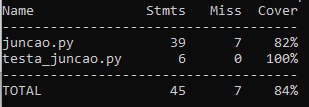
**Teste 10 – Verifica o caminho (ABECFHFGD) – Arquivo 1 com 1 elementos, igual do arquivo 2, e arquivo 2 com 2 elementos**

Foi realizado o teste utilizando o arquivo 1 com 2 elementos e o arquivo 2 com 2 elementos iguais ao do arquivo 1. Ao passar pelo Coverage.py foi obtido a seguinte cobertura:

****

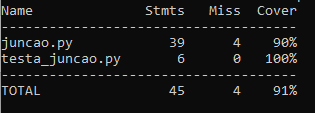
**Teste 11 – Verifica o caminho (ABECFGFGD) – Arquivo 1 com 1 elemento, menor do que elemento correspondente a linha do arquivo 2, e arquivo 2 com 2 elementos**

Ao passar pelo Coverage.py foi obtido a seguinte cobertura:

****

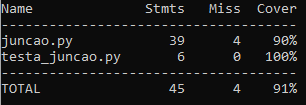
**Teste 12 – Verifica o quinto caminho (ABECFIFGD) – Arquivo 1 com 1 elemento, maior do que o elemento da linha correspondente do arquivo 2, e arquivo 2 com 2 elementos**

Ao passar pelo Coverage.py foi obtido a seguinte cobertura:

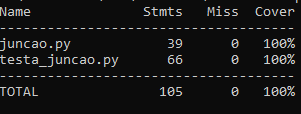
****

**Teste 13 – Verifica o exemplo dado na descrição no trabalho**

Nesse teste foi realizado o caminho ABECFGFGFID. Ao passar pelo Coverage.py foi obtido a seguinte cobertura:

****

**Também foi realizado um teste com todos os outros testes sem a forma comentada e foi possível obter pelo Coverage.py a seguinte cobertura:**

****