

Grupo 3

Nome	Matrícula	Contribuição
Djorkaeff Alexandre Vilela Pereira	16/0026822	100%
Gabriela Medeiros da Silva	16/0121817	100%
Ignacio Sanabria Alonso de Caso	19/0055511	100%
Joao Lucas Sousa Reis	16/0009758	100%
Larissa Siqueira Sales	16/0130883	100%
Lucas Gomes Silva	16/0133505	100%
Luciano dos Santos Silva	16/0013321	100%

Código analisado: <https://github.com/VictorDeon/SpaceX-API>

1. Método Walkthrough

Walkthrough é um método em equipe onde os integrantes se juntam numa “reunião de mentes” com a intenção de encontrar soluções para os problemas, testando-os, mas não corrigindo-os.

Num grupo de até 4 pessoas, com um dos participantes sendo o autor do código, é feita uma análise do programa levando em conta as observações de quem não é o autor, por ser mais efetivo.

Durante o Walkthrough, o criador do código, ou o líder da equipe repassa ou faz uma checagem desse código, através de uma simulação manual do sistema usando dados de testes. Os dados de testes devem ser simples, considerando as restrições da simulação humana. A equipe deve fazer perguntas e levantar questões sobre o produto que pode identificar defeitos.

Esta forma de teste permite encontrar de 30% a 70% dos erros presentes, com um porém de não ser fácil detectar erros de design, como processos de análise e requisição. Isso significa que o Walkthrough não substitui testes efetuados por computadores, mas serve como complemento a eles.

2. Pseudocódigo

2.1 SpaceX-API/spaceX.py

```
program SpaceX
begin
    PROXIMO_LANCAMENTO, ULTIMO_LANCAMENTO, PROXIMOS_LANCAMENTOS,
    LANCAMENTOS_PASSADOS: inteiro

    PROXIMO_LANCAMENTO <- 1
    ULTIMO_LANCAMENTO <- 2
    PROXIMOS_LANCAMENTOS <- 3
    LANCAMENTOS_PASSADOS <- 4

    rodar(cls)
    while True do
        write("0 que você deseja visualizar? \n")
        write("1) Próximo Lançamento")
        write("2) Último Lançamento")
        write("3) Próximos Lançamentos")
        write("4) Lançamentos Passados")
        write("5) Sair\n")

        try
            opcao = inteiro(read("Insira uma opção: "))
        exception
            write("Você deve inserir somente números inteiros
            de preferencia de 1 a 5")
            opcao = 0

        if opcao < 1 ou opcao > 5
            then
                write("Essa opção não existe, por favor insira
                uma opção válida.\n")
                cls.limpar(3)

                if opcao==5
                then
```

```

        cls.fechar()
        break
    else
        cls.mostra_resultado(opcao)

    resposta = read("Deseja sair da aplicação? (S/N)")

    if resposta.minusculo.comecacom("s")
    then
        cls.fechar()
        break

    cls.limpar(1)

mostra_resultado(cls, opcao){
    write()

    if option == SpaceX.PROXIMO_LANCAMENTO
    then
        cls.next_launch()
    if option == SpaceX.ULTIMO_LANCAMENTO
    then
        cls.latest_launch()
    if option == SpaceX.PROXIMOS_LANCAMENTOS
    then
        cls.upcoming_launches()
    if option == SpaceX.LANCAMENTOS_PASSADOS
    then
        cls.past_launches()
    else
        print("Opção invalida")

    mostra_resultado(cls, opcao)
end

```

2.2 SpaceX-API/models/launch.py

```
constructor_launch(flight_number, mission_name, launch_date,
launch_year, launch_date)
    flight_number <- flight_number
    mission_name <- mission_name
    launch_date <- launch_date
    launch_year <- launch_year
    rocket <- rocket
    rocket_type <- rocket_type
    launch_success <- launch_success

str()
    return
"flight_number\nmission_name\nrocket\nlaunch_year\nlaunch_date\n"

flight_number()
    return "Número do voo:flight_number"

mission_name()
    return "Missão:mission_name"

launch_date()
    date <-datetime
    return "Data de Lançamento (UTC):launch_name"

launch_year()
    return "Ano de Lançamento:launch_year"

rocket()
    return "Foguete:rocket"

launch_sucess()
    if launch_sucess
        then
            return "Lançamento realizado com sucesso!"
    end
    return "Lançamento falhou!"
```

2.3 SpaceX-API/connections/api_connection.py

```
constructor_connect(url, headers=None, params=None)
    if headers
        then
            headers = headers
        else
            headers = {'Accept': 'application/json'}
    try
        if params
            then
                response = request.get(url, headers,
params=params)
            else
                response = request.get(url, headers)

        result = response.json()
    exception
        write("Ocorreu um erro na comunicacao com a API SpaceX")

result(self)
    if type(result) == dict
        then
            Launch(flight_number = result.get('flight_number'),
mission_name = result.get('mission_name'), rocket =
result.get('rocket_name'), rocket_type =
result.get('rocket_type'), launch_success =
result.get('launch_success'), launch_date =
result.get('launch_date'), launch_year =
result.get('launch_year'))

    launches = []
    for result to result
        launches = Launch(flight_number =
result.get('flight_number'), mission_name =
result.get('mission_name'), rocket =
result.get('rocket_name'), rocket_type =
result.get('rocket_type'), launch_success =
result.get('launch_success'), launch_date =
```

```
        result.get('launch_date'), launch_year =  
        result.get('launch_year'))  
    return launches
```

```
response(self)  
    return  
    response
```

3. Inserindo dados ao código

Primeira Entrada de Teste

Variáveis / funções	Valores / argumentos
option	1
connection	Retorno de Connect("https://api.spacexdata.com/v3/launches/next") -> json
print()	connect.result

Connect("https://api.spacexdata.com/v3/launches/next")

Variáveis / funções	Valores / argumentos
self	Instância de Connect
url	"https://api.spacexdata.com/v3/launches/next"
headers	none
params	none
__headers	{'Accept': 'application/json'}
__response	Retorno em json

Connect.result

return	New Launcher(__result)
--------	------------------------

Saída
Número do Voo: 78 Missão: CRS: 17 Foguete: Falcon 9 (FT) Ano de Lançamento: 2019 Data de Lançamento (UTC): 25/04/2019 às 10:15
Sair da aplicação

3.1 SpaceX-API/models/launch.py

```
constructor_launch(flight_number, mission_name, launch_date,
launch_year, launch_date)
    flight_number <- flight_number -> "1934"
    mission_name <- mission_name -> "Apolo 23"
    launch_date <- launch_date -> "17/04/2019"
    launch_year <- launch_year -> "2019"
    rocket <- rocket -> "MEMES"
    rocket_type <- rocket_type -> "Liquid fuel"
    launch_success <- launch_success -> "1934"

str()
    return
"flight_number\nmission_name\nrocket\nlaunch_year\nlaunch_date\n"

flight_number()
    return "Número do voo:flight_number"

mission_name()
    return "Missão:mission_name"

launch_date()
    date <-datetime
    return "Data de Lançamento (UTC):launch_name"

launch_year()
    return "Ano de Lançamento:launch_year"

rocket()
    return "Foguete:rocket"

launch_sucess()
    if launch_sucess
    then
        return "Lançamento realizado com sucesso!"
end
```

```
return "Lançamento falhou!"
```

3.2 SpaceX-API/connections/api_connection.py

```
url = "https://api.spacexdata.com/v3"
headers = none
params = none

constructor_connect(url, headers=none, params=none)
    if headers --> headers = none
        then
            headers = headers
        else
            headers = {'Accept': 'application/json'} -> headers
= {'Accept': 'application/json'}
    try
        if params -> params = none
            then
                response = request.get(url, headers,
params=params)
            else
                response = request.get(url, headers) ->
response = 200
                result = response.json() -> result = {'
                    project_name': 'SpaceX-API', 'version': '3.1.0',
                    'project_link': 'https://github.com/r-spacex/SpaceX-
API',
                    'docs': 'https://documenter.getpostman.com/view/20253
50/RWaEzAiG',
                    'organization': 'r/SpaceX', 'organization_link':
'https://github.com/r-spacex',
                    'description': 'Open Source REST API for rocket,
core, capsule, pad, and launch data, created and
maintained by the developers of the r/SpaceX
organization'}
    exception
        write("Ocorreu um erro na comunicacao com a API SpaceX")

result(self)
    if type(result) == dict
        then
```

```

Launch(flight_number = result.get('flight_number'),
mission_name = result.get('mission_name'), rocket =
result.get('rocket_name'), rocket_type =
result.get('rocket_type'), launch_success =
result.get('launch_success'), launch_date =
result.get('launch_date'), launch_year =
result.get('launch_year')) -> Launch(flight_number
= "1934", mission_name = "Apolo 23", rocket =
"MEMES", rocket_type = "Liquid fuel",launch_success
= , launch_date = "17/04/2019", launch_year =
"2019")

```

```

launchs = []
for result to result
    launchs = Launch(flight_number =
    result.get('flight_number'), mission_name =
    result.get('mission_name'), rocket =
    result.get('rocket_name'), rocket_type =
    result.get('rocket_type'), launch_success =
    result.get('launch_success'), launch_date =
    result.get('launch_date'), launch_year =
    result.get('launch_year')) -> launchs =
Launch(flight_number = "1934", mission_name =
"Apolo 23", rocket = "MEMES", rocket_type = "Liquid
fuel",launch_success = , launch_date = "17/04/2019",
launch_year = "2019")

```

```

    return launchs -> return flight_number, mission_name,
launch_date, launch_year, rocket, launch_sucecss

```

```

response(self)
return
response

```

4. Sobre o código:

4.1. O código tem problemas?

Foram encontrados alguns defeitos no programa analisado e foram sugeridas algumas correções para elas:

- Não trata erros nas requisições, quebra o código se não houver conexão à internet.
- A entrada de opção tem uma requisição assíncrona e não há bloqueio de entrada enquanto a requisição não responde.
- Não valida a entrada do usuário após a requisição (deseja sair do programa?)

4.2. O que é qualidade de software?

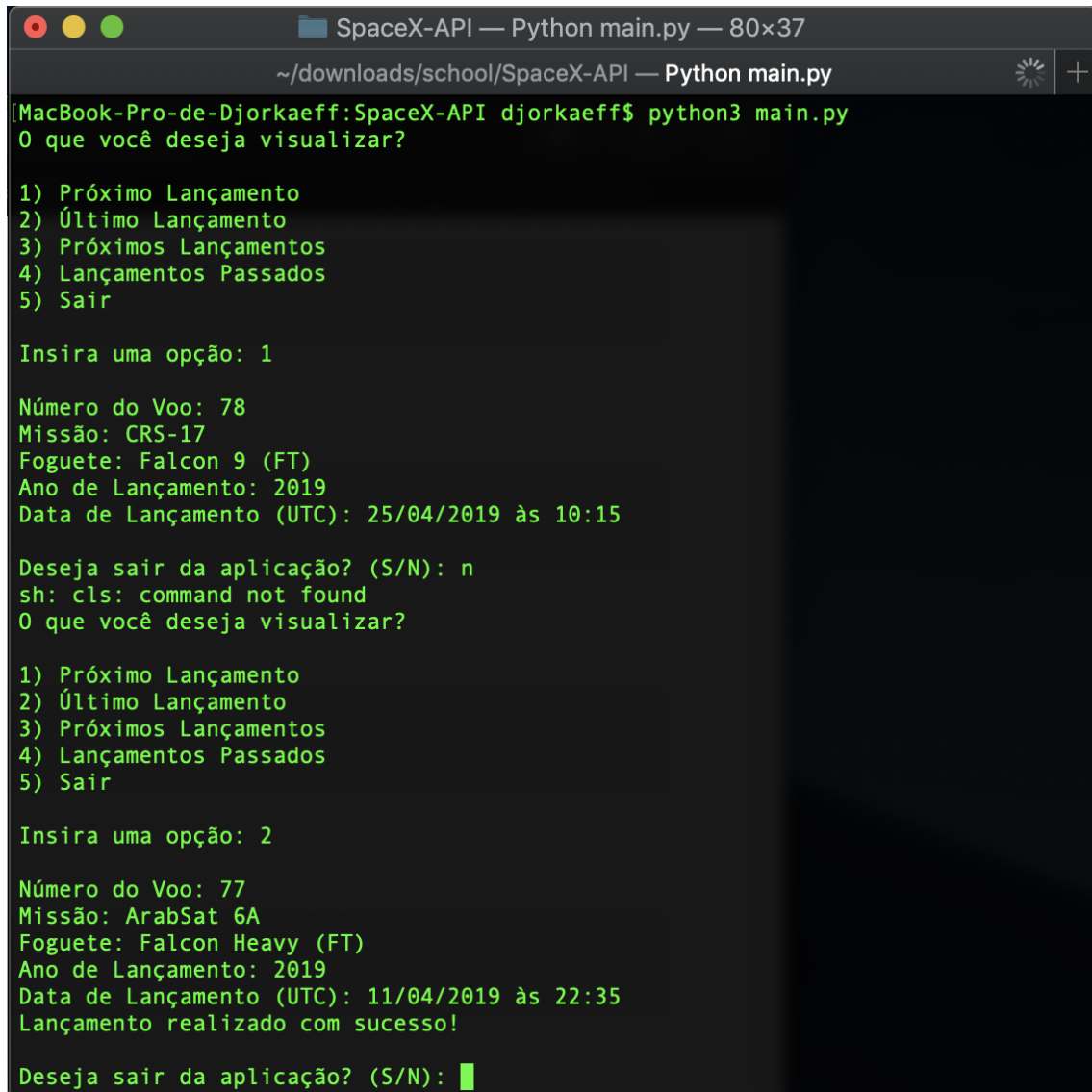
Para um produto ser considerado de qualidade, alguns atributos devem ser levados em consideração:

- Funcionalidade: deve fornecer funções que correspondam às necessidades;
- Usabilidade: deve ser usado com eficácia, eficiência e satisfação em um contexto de uso determinado;
- Confiabilidade: executa funções específicas sob condições determinadas em um dado período de tempo, de forma confiável, apresentando situações de erro controladas e reversíveis;
- Eficiência: relaciona-se ao desempenho em relação à quantidade dos recursos utilizados.
- Manutenibilidade: um sistema deve possuir a capacidade de ser modificado com determinado grau de eficácia e eficiência por um conjunto de mantenedores.
- Portabilidade: um sistema, produto ou componente pode ser transferido a partir de um hardware, software ou outro ambiente operacional com determinado grau de eficácia e eficiência

Além disso, o cliente deve sempre estar em primeiro lugar, ou seja, qualquer decisão feita deve ser pautada em fatos e dados, mas aprovadas pelo cliente.

5. Evidências de execução do código

O código foi executado numa máquina MacOS.



```
SpaceX-API — Python main.py — 80x37
~/downloads/school/SpaceX-API — Python main.py
[MacBook-Pro-de-Djorkaeff:SpaceX-API djorkaeff$ python3 main.py
0 que você deseja visualizar?

1) Próximo Lançamento
2) Último Lançamento
3) Próximos Lançamentos
4) Lançamentos Passados
5) Sair

Insira uma opção: 1

Número do Voo: 78
Missão: CRS-17
Foguete: Falcon 9 (FT)
Ano de Lançamento: 2019
Data de Lançamento (UTC): 25/04/2019 às 10:15

Deseja sair da aplicação? (S/N): n
sh: cls: command not found
0 que você deseja visualizar?

1) Próximo Lançamento
2) Último Lançamento
3) Próximos Lançamentos
4) Lançamentos Passados
5) Sair

Insira uma opção: 2

Número do Voo: 77
Missão: ArabSat 6A
Foguete: Falcon Heavy (FT)
Ano de Lançamento: 2019
Data de Lançamento (UTC): 11/04/2019 às 22:35
Lançamento realizado com sucesso!

Deseja sair da aplicação? (S/N): 
```

Evidência 1: A aplicação se comporta normalmente quando o acesso a internet é garantido, há problemas de compatibilidade no comando de limpar a tela para o bash MacOS.

```
SpaceX-API — Python main.py — 80x37
~/downloads/school/SpaceX-API — Python main.py +

Número do Voo: 78
Missão: CRS-17
Foguete: Falcon 9 (FT)
Ano de Lançamento: 2019
Data de Lançamento (UTC): 25/04/2019 às 10:15

Deseja sair da aplicação? (S/N): n
sh: cls: command not found
0 que você deseja visualizar?

1) Próximo Lançamento
2) Último Lançamento
3) Próximos Lançamentos
4) Lançamentos Passados
5) Sair

Insira uma opção: 2

Número do Voo: 77
Missão: ArabSat 6A
Foguete: Falcon Heavy (FT)
Ano de Lançamento: 2019
Data de Lançamento (UTC): 11/04/2019 às 22:35
Lançamento realizado com sucesso!

Deseja sair da aplicação? (S/N): y
sh: cls: command not found
0 que você deseja visualizar?

1) Próximo Lançamento
2) Último Lançamento
3) Próximos Lançamentos
4) Lançamentos Passados
5) Sair

Insira uma opção: █
```

Evidência 2: A entrada para a pergunta "Deseja sair da aplicação? (S/N)" não é validada, sendo considerada qualquer resposta diferente de "S" ou "s" como "não" e continua o programa.


```
SpaceX-API — -bash — 80x37
~/downloads/school/SpaceX-API — -bash
Missão: ArabSat 6A
Foguete: Falcon Heavy (FT)
Ano de Lançamento: 2019
Data de Lançamento (UTC): 11/04/2019 às 22:35
Lançamento realizado com sucesso!

Deseja sair da aplicação? (S/N): y
sh: cls: command not found
0 que você deseja visualizar?

1) Próximo Lançamento
2) Último Lançamento
3) Próximos Lançamentos
4) Lançamentos Passados
5) Sair

Insira uma opção: 1

Ocorreu um erro na comunicação com a API SpaceX
Traceback (most recent call last):
  File "main.py", line 13, in <module>
    main()
  File "main.py", line 9, in main
    SpaceX.run()
  File "/Users/djorkaeff/Downloads/school/SpaceX-API/spaceX.py", line 47, in run
    cls.__show_result(option)
  File "/Users/djorkaeff/Downloads/school/SpaceX-API/spaceX.py", line 67, in __show_result
    cls.__next_launch()
  File "/Users/djorkaeff/Downloads/school/SpaceX-API/spaceX.py", line 107, in __next_launch
    print(connection.result)
  File "/Users/djorkaeff/Downloads/school/SpaceX-API/connections/api_connection.py", line 40, in result
    if type(self.__result) == dict:
AttributeError: 'Connect' object has no attribute '_Connect__result'
MacBook-Pro-de-Djorkaeff:SpaceX-API djorkaeff$
```

Evidência 3: O sistema não tem tratamento das requisições, se for usado sem internet simplesmente quebra o código no módulo de Connections.