

# TBL1 - Fase 3

---

## Integrantes

Nome	Matrícula	Github	Contribuição
Ícaro Oliveira	15/0129807	@icarooliv	100%
Letícia Karla	15/0135939	@leticiaarj	100%
Matheus Blanco	16/0138400	@MatheusBlanco	100%
Max Henrique	16/0047013	@Maxlobo	100%
Cauê Mateus	14/0056068	@caue96	100%
Saleh Nazih Abdel Kader	16/0144485	@devsalula	100%
Pedro Henrique Andrade Féo	17/002046	@phe0	100%
Guilherme Marques Rosa	160007739	@guilhesme23	100%

## Por que utilizar *walkthrough*?

Como uma técnica de revisão walkthrough é um esforço conjunto com a finalidade de melhorar a qualidade do produto em trabalhos de desenvolvimento de softwares. A ideia consiste que vários desenvolvedores analisem o código de forma visual a fim de encontrar erros. Também é eficiente para a detecção de múltiplos erros para que todos possam ser corrigidos em uma única vez. Utilizamos a técnica revisando em conjunto os arquivos do código, procurando erros na aplicação, isso auxiliou para o entendimento conjunto mais amplo do seu funcionamento, dessa forma fazendo os integrantes a chegarem em algumas conclusões quanto aos problemas encontrados e a algumas sugestões de melhoria.

## Pseudocódigo

- ./connection/api\_connection.py

```
IMPORT requests

CLASS Connect(String: url, Header: header, Dict: params)

    self <- Connect

    FUNCTION init(Connect: self, String: url, Header: header, Dict: params)
        BEGIN
            IF header THEN
                self.headers <- headers
            ELSE IF
                self.headers <- {'Accept': 'application/json'}
            ENDIF
            TRY:
                IF params THEN
```

```

        self.response <- requests.get(url, self__headers,
params=params)
    ELSE
        self.response <- requests.get(url, self__headers)
    ENDIF
    self.result <- requests.get(url, self.headers)
EXCEPT (requests.exceptions.RequestException):
    WRITE "Ocorreu um erro na comunicação com a API SpaceX"
END

FUNCTION result(Connect: self):
    BEGIN
        IF TYPE OF self.result IS EQUAL TO dict
            Launch <- (
                Launch.flight_number <-
self.result.get('flight_number')
                Launch.mission_name <-
self.result.get('mission_name')
                Launch.rocket <-
self.result.get('rocket').get('rocket_name'),
                Launch.rocket_type <-
self.result.get('rocket').get('rocket_type')
                Launch.launch_success <-
self.result.get('launch_success')
                Launch.launch_date_utc <-
self.result.get('launch_date_utc')
                Launch.launch_year <-
self.result.get('launch_year')
            )
            RETURN Launch
        ENDIF
        launches <- Array[]
        FOR result IN self.result:
            Launch <- (
                Launch.flight_number <-
self.result.get('flight_number')
                Launch.mission_name <-
self.result.get('mission_name')
                Launch.rocket <-
self.result.get('rocket').get('rocket_name'),
                Launch.rocket_type <-
self.result.get('rocket').get('rocket_type')
                Launch.launch_success <-
self.result.get('launch_success')
                Launch.launch_date_utc <-
self.result.get('launch_date_utc')
                Launch.launch_year <-
self.result.get('launch_year')
            )
            launches.append <- Launch
        RETURN launches
    END

FUNCTION response(Connect: self)

```

```
BEGIN
    RETURN self.response
END
```

- ./models/launch.py

CLASS LAUNCH

```
    DECLARE (flight_number, mission_name, rocket, rocket_type,
launch_success, launch_year, launch_date):
```

```
    CONSTRUCT:
```

```
    self.flight_number <- flight_number
    self.mission_name <- mission_name
    self.launch_date <- launch_date
    self.launch_year <- launch_year
    self.rocket <- rocket
    self.rocket_type <- rocket_type
    self.launch_sucess <- launch_sucess
```

```
    DECLARE object as string:
```

```
        IF launch_sucess IS NOT NULL THEN
```

```
            return in format "{0}\n{1}\n{2}\n{3}\n{4}\n{5}"(
                flight_number,
                mission_name,
                rocket, rocket_type,
                launch_year,
                launch_date,
                launch_success
            )
```

```
        return in format "{0}\n{1}\n{2}\n{3}\n{4}\n{5}"(
            flight_number,
            mission_name,
            rocket,
            rocket_type,
            launch_year,
            launch_date,
            launch_success
        )
```

```
    DECLARE flight_number instance:
```

```
        get flight number of launch: "Numero do voo:{0}".format of
flight_number instance
```

```
    DECLARE mission_name instance:
```

```
        get mission name of launch: "Missão:{0}".format of mission_name
instance
```

```

DECLARE launch_date instance:

    date <- datetime - __launch_date instance in "%Y-%m-%dT%H:%M:%S.%fZ" format

    return date of launch: "Data de Lançamento (UTC):{0}".format of
__launch_date instance in "%Y-%m-%dT%H:%M:%S.%fZ"

DECLARE launch_year instance:

    return year of launch: "Ano de Lançamento (UTC):{0}".format of
launch_year instance

DECLARE rockett instance:

    return rocket and rocket_type: "Foguete:{0}".format of rocket and
rocket_type instance

DECLARE launch_sucess instance:

    IF launch_sucess instanciaes THEN
        return "Lançamento realizado com sucesso"

    ELSE return "Lançamento falhou!"

```

- ./spaceX.py

```

CLASS SpaceX
    NEXT_LAUNCH <- 1
    LATEST_LAUNCH <- 2
    UPCOMING_LAUNCHES <- 3
    PAST_LAUNCHES <- 4

    FUNCTION _run:
        BEGIN
            WHILE True THEN:
                WRITE "1) Próximo Lançamento"
                WRITE "2) Último Lançamento"
                WRITE "3) Próximos Lançamentos"
                WRITE "4) Lançamentos Passados"
                WRITE "5) Sair"
                TRY:
                    option <- int(input("Insira uma opção: "))
                EXCEPT ValueError:
                    WRITE "Você deve inserir somente números inteiros de
preferencia de 1 a 5"
                    option <- 0
                IF option IS SMALLER THAN 1 or option IS BIGGER THAN 5 THEN
                    WRITE "Essa opção não existe, por favor insira uma
opção válida.\n"

```

```

        cls.__clean(3)
    ELSE IF option IS EQUAL TO 5 THEN
        cls.__close()
        break
    ELSE THEN
        cls.__show_result(option)
        answer <- input("Deseja sair da aplicação? (S/N): ")
        IF answer.lower().startswith("s") THEN
            cls.__close()
            break
        cls.__clean(1)
    ENDIF
END WHILE
END

FUNCTION _show_result:
BEGIN
    WRITE
    IF option IS EQUAL TO SpaceX.NEXT_LAUNCH THEN
        cls.__next_launch()
    ELSE IF option IS EQUAL TO SpaceX.LATEST_LAUNCH THEN
        cls.__latest_launch()
    ELSE IF option IS EQUAL TO SpaceX.UPCOMING_LAUNCHES THEN
        cls.__upcoming_launches()
    ELSE IF option IS EQUAL TO SpaceX.PAST_LAUNCHES THEN
        cls.__past_launches()
    ELSE THEN
        WRITE "Opção invalida"
    ENDIF
END

FUNCTION _clean(int: seconds):
BEGIN
    time.sleep(seconds)
    IF 'win' IN sys.platform THEN
        os.system("cls")
    ELSE THEN
        os.system("clear")
    ENDIF
END

FUNCTION _close:
BEGIN
    WRITE "Finalizando o programa..."
    time.sleep(1)
END

FUNCTION _next_launch:
BEGIN
    connection <-
Connect("https://api.spacexdata.com/v3/launches/next")
    WRITE connection.result
END

```

```

    FUNCTION _upcoming_launches:
    BEGIN
        connection <-
Connect("https://api.spacexdata.com/v3/launches/upcoming")
        FOR result IN connection.result:
            WRITE result
            WRITE "-----\n"
        END FOR
    END

    FUNCTION _latest_launch:
    BEGIN
        connection <-
Connect("https://api.spacexdata.com/v3/launches/latest")
        WRITE connection.result
    END

    FUNCTION _past_launches:
    BEGIN
        connection <-
Connect("https://api.spacexdata.com/v3/launches/past")
        FOR result IN connection.result:
            WRITE result
            WRITE "-----\n"
        END FOR
    END

```

## Instâncias do pseudocódigo

- ./spaceX.py

```

CLASS SpaceX
    NEXT_LAUNCH <- 1
    LATEST_LAUNCH <- 2
    UPCOMING_LAUNCHES <- 3
    PAST_LAUNCHES <- 4

    FUNCTION _run:
    BEGIN
        WHILE True THEN:
            WRITE "1) Próximo Lançamento"
            WRITE "2) Último Lançamento"
            WRITE "3) Próximos Lançamentos"
            WRITE "4) Lançamentos Passados"
            WRITE "5) Sair"
            TRY:
                option <- 1
            EXCEPT ValueError:
                WRITE "Você deve inserir somente números inteiros de

```

```

preferencia de 1 a 5"
        option <- 0
        IF 1 IS SMALLER THAN 1 or option IS BIGGER THAN 5 THEN
            WRITE "Essa opção não existe, por favor insira uma
opção válida.\n"
            cls.__clean(3)
        ELSE IF 1 IS EQUAL TO 5 THEN
            cls.__close()
            break
        ELSE THEN
            cls.__show_result(1)
            answer <- "S"
            IF answer.lower().startswith("s") THEN
                cls.__close()
                break
            cls.__clean(1)
        ENDIF
    END WHILE
END

FUNCTION _show_result:
    BEGIN
        WRITE
        IF 1 IS EQUAL TO SpaceX.NEXT_LAUNCH THEN
            cls.__next_launch()
        ELSE IF 1 IS EQUAL TO SpaceX.LATEST_LAUNCH THEN
            cls.__latest_launch()
        ELSE IF 1 IS EQUAL TO SpaceX.UPCOMING_LAUNCHES THEN
            cls.__upcoming_launches()
        ELSE IF 1 IS EQUAL TO SpaceX.PAST_LAUNCHES THEN
            cls.__past_launches()
        ELSE THEN
            WRITE "Opção invalida"
        ENDIF
    END

FUNCTION _clean(3):
    BEGIN
        time.sleep(3)
        IF 'win' IN sys.platform THEN
            os.system("cls")
        ELSE THEN
            os.system("clear")
        ENDIF
    END

FUNCTION _close:
    BEGIN
        WRITE "Finalizando o programa..."
        time.sleep(1)
    END

FUNCTION _next_launch:
    BEGIN

```

```

        connection <-
Connect("https://api.spacexdata.com/v3/launches/next")
        WRITE connection.result
    END

    FUNCTION _upcoming_launches:
    BEGIN
        connection <-
Connect("https://api.spacexdata.com/v3/launches/upcoming")
        FOR result IN connection.result:
            WRITE result
            WRITE "-----\n"
        END FOR
    END

    FUNCTION _latest_launch:
    BEGIN
        connection <-
Connect("https://api.spacexdata.com/v3/launches/latest")
        WRITE connection.result
    END

    FUNCTION _past_launches:
    BEGIN
        connection <-
Connect("https://api.spacexdata.com/v3/launches/past")
        FOR result IN connection.result:
            WRITE result
            WRITE "-----\n"
        END FOR
    END

```

## Problemas encontrados

- É necessário ter conexão com a internet para o Software realizar a requisição na API Space X. Todavia, não há nenhuma documentação que deixe isso explícito.
- A entrada, "Deseja sair da aplicação? (S/N)", aceita entradas diferentes de S e N. Sendo que aquelas descoincidentes o software entende como N.

## Sugestões de Correção

- Criar documentação para o site deixando claro para o usuário quais são os requisitos necessários para que o software funcione.
- Adicionar uma restrição para a entrada, "Deseja sair da aplicação? (S/N)", onde se o usuário inserir algo distinto de S e N, o software retorna uma mensagem de erro.

## Qualidade do Software



A definição de qualidade de software leva em consideração alguns fatores que o produto deve atingir, em diferentes níveis, variando de acordo com o contexto.

- **Funcionalidade:** o software deve fornecer funções que correspondam às necessidades;
  - No caso do código analisado, o escopo do projeto é bem definido e é deixado claro na sua documentação (README.md)
- **Usabilidade:** deve ser usado com eficácia, eficiência e satisfação em um contexto de uso determinado;
  - No código analisado, há instruções de como o software deve ser utilizado na documentação (README.md) mas não especifica em quais condições o seu uso pode ser afetado (sem conexão com a internet, por exemplo.)
- **Confiabilidade:** executa funções específicas sob condições determinadas em um dado período de tempo, de forma confiável, apresentando situações de erro controladas e reversíveis;
  - No software analisado, nem todos os erros [falham graciosamente](#), apresentando situações de erro onde é preciso sair do programa de forma bruta.
- **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.
  - No software analisado, [houve a preocupação](#) de fazer portabilidade para outros SO como Windows

## Evidências da execução do código

Testador(a)	Sistema Operacional	Data da execução
Letícia Karla	Ubuntu 18.04	06/10/2019

### Funcionalidade: Visualizar Último Lançamento.

- Entrada: 2.
- Saída esperada: Número do Voo, missão, foguete, ano de lançamento e data de lançamento do último lançamento.
- Resultados: Saída sem presença de irregularidades.

```
leticia@leticia-Aspire-5742 ~/Downloads/SpaceX-API-master (master) $ 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: 2

Número do Voo: 83
Missão: Amos-17
Foguete: Falcon 9 (FT)
Ano de Lançamento: 2019
Data de Lançamento (UTC): 06/08/2019 às 22:52
Lançamento realizado com sucesso!

Deseja sair da aplicação? (S/N): ☐
```

- Observações: N/A.
- Erros encontrados: N/A.

#### Funcionalidade: Visualizar Próximo Lançamento.

- Entrada: 1.
- Saída esperada: Número do Voo, missão, foguete, ano de lançamento e data de lançamento do próximo lançamento.
- Resultados: Saída sem presença de irregularidades.

```
leticia@leticia-Aspire-5742 ~/Downloads/SpaceX-API-master (master) $ 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: 84
Missão: Starlink 2
Foguete: Falcon 9 (FT)
Ano de Lançamento: 2019
Data de Lançamento (UTC): 17/10/2019 às 00:00

Deseja sair da aplicação? (S/N): ☒
```

- Observações: N/A.
- Erros encontrados: N/A.

#### Funcionalidade: Visualizar Próximos Lançamentos.

- Entrada: 3.
- Saída esperada: Número do Voo, missão, foguete, ano de lançamento e data de lançamento dos próximos lançamentos.
- Resultados: Saída com erro.

```

leticia@leticia-Aspire-5742 ~/Downloads/SpaceX-API-master (master) $ 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: 3

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 "/home/leticia/Downloads/SpaceX-API-master/spaceX.py", line 47, in run
    cls.__show_result(option)
  File "/home/leticia/Downloads/SpaceX-API-master/spaceX.py", line 71, in __show_result
    cls.__upcoming_launches()
  File "/home/leticia/Downloads/SpaceX-API-master/spaceX.py", line 117, in __upcoming_launches
    for result in connection.result:
  File "/home/leticia/Downloads/SpaceX-API-master/connections/api_connection.py", line 40, in result
    if type(self.__result) == dict:
AttributeError: 'Connect' object has no attribute '_Connect_result'
leticia@leticia-Aspire-5742 ~/Downloads/SpaceX-API-master (master) $ █

```

- Observações: O computador em que o teste estava sendo realizado, estava sem conexão com a internet.
- Erros encontrados: O software não funciona sem conexão com a internet.

### Funcionalidade: Visualizar Lançamentos Passados.

- Entrada: 4.
- Saída esperada: Número do Voo, missão, foguete, ano de lançamento e data dos lançamentos passados.
- Resultados: Saída sem presença de irregularidades.

```

0 que você deseja visualizar?

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

Insira uma opção: 4

Número do Voo: 1
Missão: FalconSat
Foguete: Falcon 1 (Merlin A)
Ano de Lançamento: 2006
Data de Lançamento (UTC): 24/03/2006 às 22:30
Lançamento falhou!

-----

Número do Voo: 2
Missão: DemoSat
Foguete: Falcon 1 (Merlin A)
Ano de Lançamento: 2007
Data de Lançamento (UTC): 21/03/2007 às 01:10
Lançamento falhou!

-----

Número do Voo: 3
Missão: Trailblazer

```

- Observações: N/A.
- Erros encontrados: N/A.

**Funcionalidade: Sair.**

- Entrada: "Sim".
- Saída Esperada: Finalização do Software.
- Resultados: Software continuou mostrando o Menu.

```
leticia@leticia-Aspire-5742 ~/Downloads/SpaceX-API-master (master) $ 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: 84
Missão: Starlink 2
Foguete: Falcon 9 (FT)
Ano de Lançamento: 2019
Data de Lançamento (UTC): 17/10/2019 às 00:00

Deseja sair da aplicação? (S/N): nao
```

```
0 que você deseja visualizar?

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

Insira uma opção: 
```

- Observações: Entrada diferente de "S" e "N".
- Erros encontrados: Quando há inserção de entrada diferente de "S" ou "N". Ele identifica como "N".