**TBL 1 - Fase 3 - Grupo 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Estudante** | **Matrícula** | | **Git** | **E-mail** |
| Guilherme de Lyra P. | 150128231 | | guilhermedelyra | guilyra12@gmail.com |
| Heron Rodrigues Sousa | 160124484 | | heronsousa | [heronrs1@gmail.com](http://|heronrs1@gmail.com/) |
| Lucas Ganda Carvalho | 170039668 | | [lucasgandac](https://web.telegram.org/#/im?p=%40lucasgandac) | lucasgandac@hotmail.com |
| Victor Rodrigues Silva | 160019516 | | VictorRodriguesS0 | victorrsilva17@gmail.com |
| Marcos Vinícius R. da Conceição | 17/0150747 | | marcos-mv | vinicius\_kof36@hotmail.com |
| Henrique Martins de Messias | 17/0050394 | | Henrike100 | henrikem100@gmail.com |

Repositório GIT da equipe : [**https://github.com/fga-verival/2019-2Grupo4**](https://github.com/fga-verival/2019-2Grupo4)

**1. Walkthrough**

É uma técnica aplicada em grupo, onde o autor leva a equipe através de uma execução simulada ou manual do produto utilizando critérios definidos, é interessante de ser aplicada, pois, além do autor do programa, diversas pessoas estão envolvidas no processo, oferecendo diferentes pontos de vista sobre o artefato. Esse procedimento frequentemente expõe uma grande quantidade de erros e suas localizações precisas no código, permitindo que sejam corrigidos de uma só vez.

O Walkthrough pode ser feito de maneira automática, estática ou funcional, geralmente o custo do uso desta técnica pode variar de acordo com a quantidade de pessoas envolvidas, tamanho e complexidade do projeto e artefatos envolvidos, geralmente classificada com custo médio.

A técnica é interessante, pois fornece uma base para questionar a lógica do programador e ajudar a identificar problemas no sistema.

**2. Pseudocódigo**

**spaceX.py**

|  |
| --- |
| IMPORT Connect DE connections   NEXT\_LAUNCH = 1  LATEST\_LAUNCH = 2  UPCOMING\_LAUNCHES = 3  PAST\_LAUNCHES = 4   WHILE 1 DO  PRINT ("O que você deseja visualizar?")  PRINT ("1) Próximo Lançamento")   PRINT ("2) Último Lançamento")   PRINT ("3) Próximos Lançamentos")   PRINT ("4) Lançamentos Passados")  PRINT ("5) Sair")   PRINT ("Insira uma opção: ")  SCAN(option)     IF option != INT THEN  PRINT ("Você deve inserir somente números inteiros de preferência de 1 a 5")  option = 0  END IF   IF option < 1 OR option > 5 THEN  PRINT ("Essa opção não existe, por favor insira uma opção válida.")  clean(3)  END IF  ELSE  IF option == 5 THEN  END\_LOOP  END IF  ELSE  show\_result(option)  PRINT ("Deseja sair da aplicação? (S/N): ")  SCAN (answer)  IF answer[0] == "s" THEN  END\_LOOP  END IF  clean(1)  END ELSE  END ELSE  END WHILE   FUNCTION show\_result(option)  BEGIN  IF option == NEXT\_LAUNCH THEN  next\_launch()  END IF  ELSE IF option == LATEST\_LAUNCH THEN  latest\_launch()  END ELSE IF  ELSE IF option == UPCOMING\_LAUNCHES THEN  upcoming\_launches()  END ELSE IF  ELSE IF option == PAST\_LAUNCHES THEN  past\_launches()  END ELSE IF  ELSE  PRINT ("Opção invalida")  END ELSE  END  FUNCTION clean(seconds)  BEGIN  SLEEP(seconds)    IF PLATAFORMA == WINDOWNS THEN  LIMPAR\_TELA\_WINDOWNS()  END IF  ELSE  LIMPAR\_TELA\_OUTRA\_PLATAFORMA()  END ELSE  END   FUNCTION close()  BEGIN  PRINT("Finalizando o programa...")  SLEEP(1)  END   FUNCTION next\_launch()  BEGIN  connection <- Connect("https://api.spacexdata.com/v3/launches/next")  PRINT(connection.result)  END   FUNCTION upcoming\_launches()  BEGIN  connection <- Connect("https://api.spacexdata.com/v3/launches/upcoming")  FOR EACH result IN connection.result  PRINT(result)  PRINT("----------------------------------------------------------\n")  END FOR  END   FUNCTION latest\_launch()  BEGIN  connection <- Connect("https://api.spacexdata.com/v3/launches/latest")  PRINT(connection.result)  END   FUNÇÃO past\_launches():  BEGIN  connection <- Connect("https://api.spacexdata.com/v3/launches/past")  FOR EACH result IN connection.result  PRINT(result)  PRINT("----------------------------------------------------------\n")  END |

**api\_connection.py**

|  |
| --- |
| IMPORT launch FROM models   CLASS Connect BEGIN  constructor(url, header<-None, params<-none)  IF header != NIL then  self.\_\_headers <- headers  END IF  ELSE then  header <- {'Accept': 'application/json'}  END ELSE   try  BEGIN   IF params != NIL then  self.\_\_response <- request.get(url, self.\_\_headers, params=params)  END IF  ELSE then  self.\_\_response <- requests.get(url, self.\_\_headers)  END ELSE   self.\_\_result <- self.\_\_response.json()   END   except  BEGIN  print("Ocorreu um erro na comunicação com a API SpaceX")  END    FUNCTION result():  BEGIN  IF result type == dict type then  return Launch(  flight\_number<-self.\_\_result.get('flight\_number'),  mission\_name<-self.\_\_result.get('mission\_name'),  rocket<-self.\_\_result.get('rocket').get('rocket\_name'),  rocket\_type<-self.\_\_result.get('rocket').get('rocket\_type'),  launch\_success<-self.\_\_result.get('launch\_success'),  launch\_date<-self.\_\_result.get('launch\_date\_utc'),  launch\_year<-self.\_\_result.get('launch\_year')  )  END IF   launchs <- []   FOR each result in self.\_\_result  launchs.append(  Launch(  flight\_number<-result.get('flight\_number'),  mission\_name<-result.get('mission\_name'),  rocket<-result.get('rocket').get('rocket\_name'),  rocket\_type<-result.get('rocket').get('rocket\_type'),  launch\_success<-result.get('launch\_success'),  launch\_date<-result.get('launch\_date\_utc'),  launch\_year<-result.get('launch\_year')  )  )  END FOR   return launchs  END   FUNCTION response  BEGIN  return self.\_\_response  END  END |

**launch.py**

|  |
| --- |
| FROM datetime IMPORT datetime  CLASS launch BEGIN   FUNCTION init(flight\_number, mission\_name, rocket,  rocket\_type, launch\_success,  launch\_year, launch\_date)  BEGIN   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    END   FUNCTION str  BEGIN   IF launch\_success != NIL then  print( flight\_number, mission\_name, rocket,  launch\_year, launch\_date, launch\_success  END IF    print( flight\_number, mission\_name, rocket,  launch\_year, launch\_date, launch\_success   END   FUNCTION flight\_number  BEGIN   print ("Número do Voo", flight\_number)   END   FUNCTION mission\_name  BEGIN   print( "Missão", mission\_name)   END    FUNCTION launch\_date  BEGIN   date <- datetime.(launch\_date)   print( "Data de Lançamento (UTC):",date)   END    FUNCTION launch\_year  BEGIN   print( "Ano de Lançamento",launch\_year)   END     FUNCTION rocket  BEGIN   print( "Foguete",rocket,rocket\_type)   END     FUNCTION launch\_success  BEGIN   IF launch\_success == TRUE then  print( "Lançamento realizado com sucesso!")  END IF   ELSE then  print( "Lançamento falhou!")  END ELSE    END  END |

**3. Simulação**

|  |
| --- |
| -- Inicia programa "spaceX.py" --  :output do programa "O que você deseja visualizar? " "1) Próximo Lançamento "2) Último Lançamento "3) Próximos Lançamentos "4) Lançamentos Passados "5) Sair " "Insira uma opção: :fim output  >input do usuário = 2  $ atribui à variavel "option" o valor 2  $ passa pelo primeiro if, cujo qual têm o papel de garantir que a entrada é válida (um número inteiro)  $ passa pelo segundo if, cujo qual têm o papel de garantir que o número escolhido é válido (entre 1 e 5)  $ entra no else  $ passa pelo if, que têm o papel de finalizar o programa caso o input do usuário tenha sido 5  $ entra no else  $ chama a função "show\_result()" com a opção selecionada pelo usuário (2)  $ passa pelo primeiro if, que tem o objetivo de mostrar o próximo lançamento  $ entra no segundo if, que corresponde à entrada do usuário: mostrar o último lançamento  $ chama a função "latest\_launch()"  $ consome a api da spaceX  $ retorna o resultado   :output do programa " "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! " :fim output   $ fim da função "latest\_launch()"  $ fim da função "show\_result()"  $ solicita se o usuário quer continuar a interagir com o programa ou sair :output do programa "Deseja sair da aplicação? (S/N): :fim output  >input do usuário = "S"   $ entra no primeiro if, que termina a execução do programa  :output do programa "Finalizando o programa... :fim output  -- Fim do programa "spaceX.py" -- |

**4. Funcionamento do programa**

O programa não apresentou erros de execução durante sua execução quando conectado a internet.

\* Durante a execução do programa, ao selecionarmos alguma das opções 3 ou 4, o programa dá a sensação ao usuário de que está travado, mas na verdade está buscando as informações requisitadas na API, então uma simples mensagem de "carregando" ou "buscando" ajudaria muito na percepção do usuário sobre o programa.

Durante a execução do programa sem conexão com internet

\* Durante a execução do programa, ao selecionarmos quaisquer opções exceto a de finalizar o programa, temos um feedback de falha de comunicação com a api, mas ao mesmo tempo temos um erro de atributo da classe Connect, o que para um possível usuário deste software sem conhecimentos em programação ou até mesmo um programador que não possui acesso ao código, ou não conhece o código, não conseguiria resolver este problema.

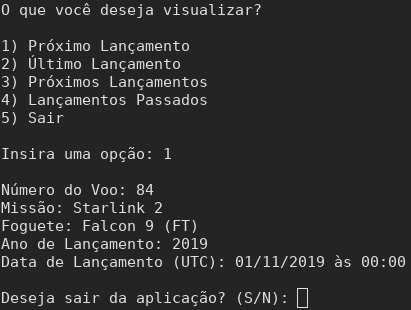
Execução com entradas incorretas

\* Durante a execução do programa, ao colocar entradas não esperadas pelo programa, ele consegue fazer o tratamento das entradas e avisa ao usuário que a entrada está incorreta, mas ao dar esta mensagem, o programa leva alguns segundo para retornar o usuário à tela principal, o que pode gerar desconforto, até mesmo por que o programa retorna sozinho, e por um descuido do usuário, ele pode desviar o olhar e acabar não vendo a mensagem de erro.

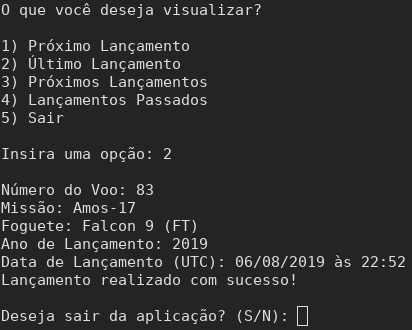
**5. Execução do programa**

**Execução com internet**

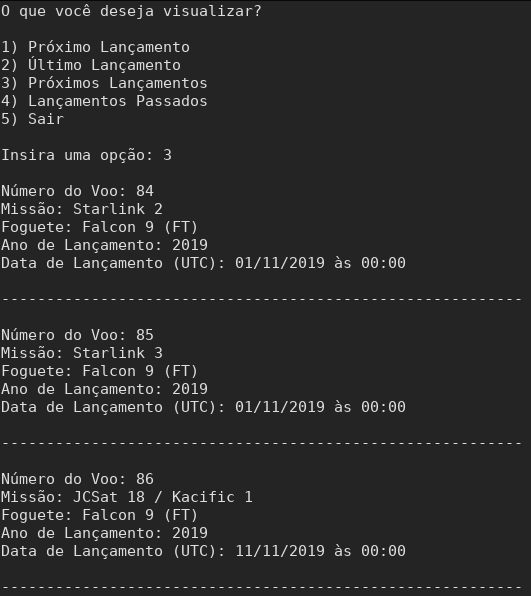
**- Opção 1:**



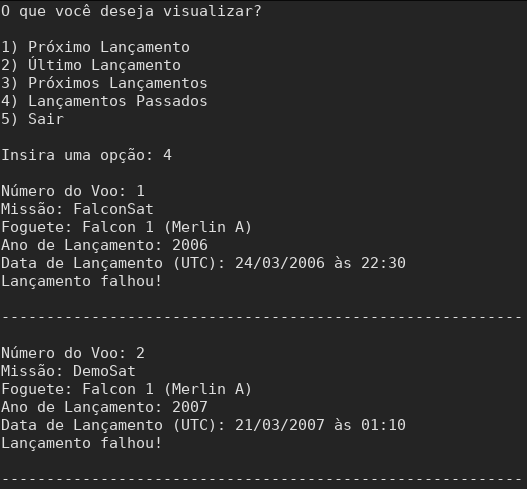
**- Opção 2:**



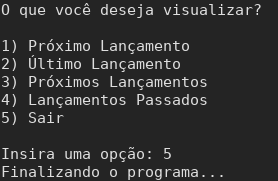
**- Opção 3:**



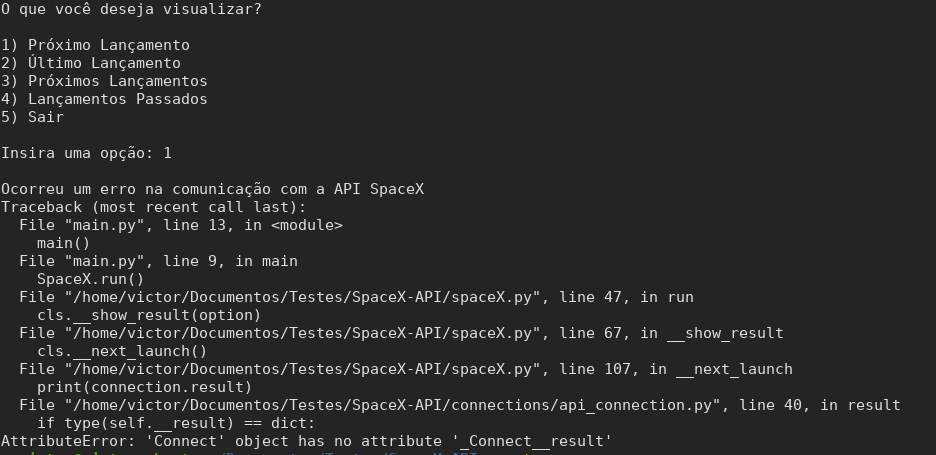
**- Opção 4:**



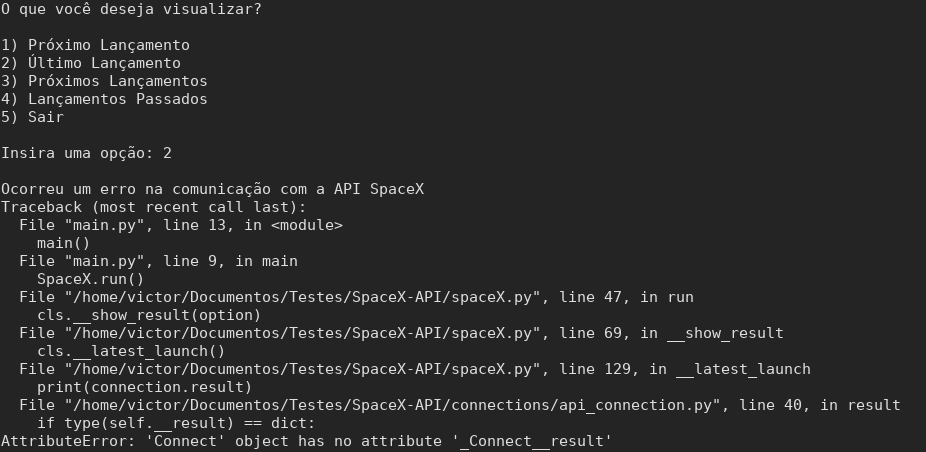
**- Opção 5:**



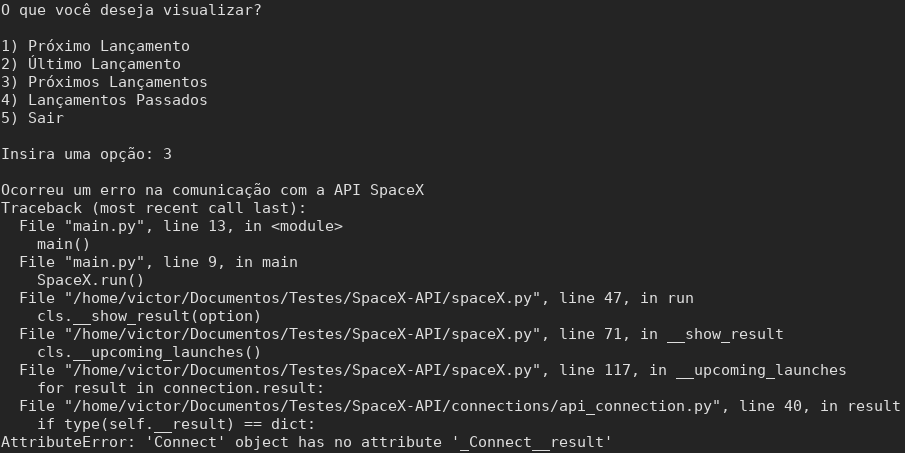
**Execução sem internet**



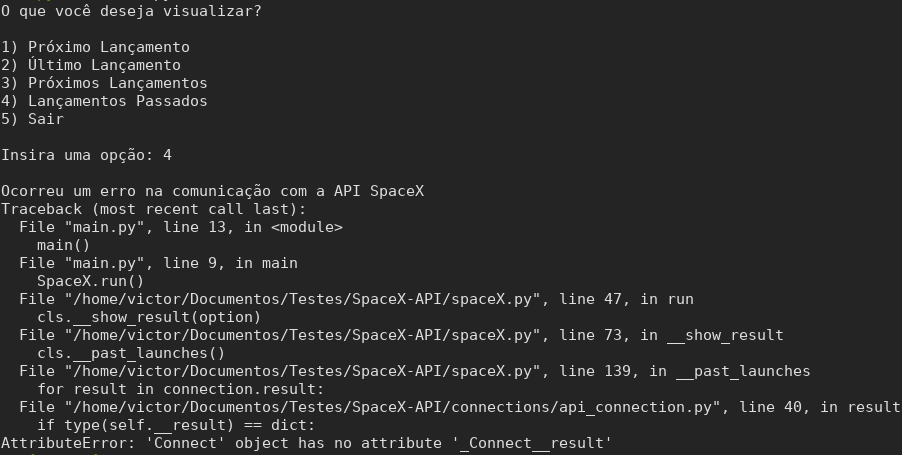
**- Opção 2:**



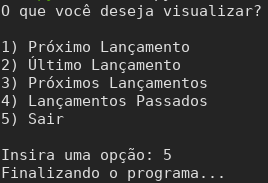
**- Opção 3:**



**- Opção 4:**



**- Opção 5:**



**Execução com entradas incorretas**

