

TBL 1 - FASE 3

Grupo 02

Alunos: Lucas Midlhey - 100%
Geovana Ramos - 100%
Geovanne Santos - 100%
Caio César Beleza - 100%
Érico Maximiano Bandeira - 100%
Igor Aragão - 100%
Guilherme Guy - 100%

Data: 17/04/2019

Com base no código obtido <https://github.com/VictorDeon/SpaceX-API>

1. Explique, por que a técnica de walkthrough é interessante para verificar se o código funcionará corretamente (para isso use a definição desta técnica. O Livro do Myers tem de forma bem detalhada).

A técnica walkthrough é interessante devido a maneira que ela é conduzida, onde a equipe não corrige o erro, mas percorre todo código como um computador realizando caso de teste. A técnica é consistida geralmente em um time de 4 pessoas onde apenas uma delas foi desenvolvedor do código e as outras mapeiam falhas durante sua execução. Entre suas vantagens está o baixo custo de debugging devido ao levantamento primeiramente dos erros e a precisão de localização de erros no código. Um outro diferencial é que depois de encontrado vários erros em pontos específicos, torna mais fácil corrigi-los em massa. Além disso, o processo humano tende a ser mais efetivo que os processos computacionais para achar certos tipos de erros.

2. Percorra o código procurando entendê-lo e avaliar se ele está correto. Como evidência dessa atividade construa um pseudocódigo usando os padrões publicados no Moodle (fonte: Cormen et al, 2002, pp. 14,15) e tem na biblioteca da FGA, mas o que vcs precisam sobre padrões de pseudocódigo que adotaremos na disciplina está no Moodle. Para exemplos de como ele usa o pseudocódigo, vejam o livro.

```
NEXT_LAUNCH[SpaceX] <- 1  
LATEST_LAUNCH[SpaceX] <- 2
```

```
UPCOMING_LAUNCHES[SpaceX] <- 3
PAST_LAUNCHES[SpaceX] <- 4
```

```
MAIN()
  chamar função RUN
```

```
RUN[SpaceX]()
  while true do
    PRINT("O que você deseja visualizar?\n")
    PRINT("1) Próximo Lançamento")
    PRINT("2) Último Lançamento")
    PRINT("3) Próximos Lançamentos")
    PRINT("4) Lançamentos Passados")
    PRINT("5) Sair\n")

    try
      option <- inteiro digitado pelo usuário
    catch não é inteiro
      PRINT("Você deve inserir somente números inteiros de
preferencia de 1 a 5")
      option <- 0

    if option < 1 ou option > 5 then
      PRINT("Essa opção não existe, por favor insira uma opção
válida.\n")
      __CLEAN[SpaceX](3)
    else if option = 5 then
      __CLOSE[SpaceX]()
    else
      __SHOW_RESULT[SpaceX](option)
      PRINT("Deseja sair da aplicação? (S/N): ")
      answer <- resposta do usuário

      if answer convertido para minúsculo começa com 's' then
        __CLOSE[SpaceX]()
        break

      __CLEAN[SpaceX]()

  __SHOW_RESULTS[SpaceX](option)
  PRINT( )
  if option igual à SpaceX.NEXT_LAUNCH then
```

```

        NEXT_LAUNCH[SpaceX]()
    else if option igual à SpaceX.LATEST_LAUNCH then
        LATEST_LAUNCH[SpaceX]()
    else if option igual à SpaceX.UPCOMING_LAUNCHES then
        UPCOMING_LAUNCHES[SpaceX]()
    else if option igual à SpaceX.PAST_LAUNCHES then
        PAST_LAUNCHES[SpaceX]()
    else
        PRINT("Opção Inválida")

```

```

__CLEAN[SpaceX](seconds)

```

```

    SLEEP(seconds)

```

```

    if sistema é windows then
        chamada_de_sistema("cls")
    else
        chamada_de_sistema("clear")

```

```

__CLOSE[SpaceX]()

```

```

    PRINT("Finalizando o programa...")

```

```

    SLEEP(1)

```

```

__NEXT_LAUNCH[SpaceX]()
    connection <-
    __init__[Connect]("https://api.spacexdata.com/v3/launches/next")
    PRINT(results[connection])

```

```

__LATEST_LAUNCH[SpaceX]()
    connection <-
    __init__[Connect]("https://api.spacexdata.com/v3/launches/latest")
    PRINT(results[connection])

```

```

__PAST_LAUNCHES[SpaceX]()
    connection <-
    __init__[Connect]("https://api.spacexdata.com/v3/launches/latest")

```

```

    foreach result in results[connection]
        PRINT(results)

```

```
PRINT("-----")
```

```
__UPCOMING_LAUNCHES[SpaceX]()
```

```
__init__[Connect]("https://api.spacexdata.com/v3/launches/upcoming")
```

```
foreach result in results[connection]
```

```
    PRINT(result)
```

```
    PRINT("-----\n")
```

```
__LATEST_LAUNCHES()
```

```
__init__[Connect]("https://api.spacexdata.com/v3/launches/latest")
```

```
PRINT(results[connection])
```

```
__PAST_LAUNCHES()
```

```
_init__[Connect]("https://api.spacexdata.com/v3/launches/past")
```

```
foreach result in results[connection]
```

```
    PRINT(result)
```

```
    PRINT("-----\n")
```

```
__init__[Connect](url, headers, params)
```

```
self <- Connect
```

```
if headers não é null then
```

```
    __headers[self] <- headers
```

```
else
```

```
    __headers[self] <- {'Accept': 'application/json'}
```

```
try
```

```
    if params não é null then
```

```
        __response[self] <- GET_URL(url, __headers[self], params)
```

```
    else
```

```
        __response[self] <- GET_URL(url, __headers[self])
```

```
catch Exceção de request
```

```
    PRINT("Ocorreu um erro na comunicação com a API SpaceX")
```

```
return self
```

```

result[Connect](self)
  if __result[self] é um dicionário then
    return __init__[Launch](
      flight_number[__result[self]],
      mission_name[__result[self]],
      rocket_name[rocket[__result[self]]],
      rocket_type[rocket[__result[self]]],
      launch_success[__result[self]],
      launch_date_utc[__result[self]],
      launch_year[__result[self]]
    )

```

```

launchs <- []
i <- 1

```

```

foreach result in __result[self]:
  launchs[i] <- __init__[Launch](
    flight_number[__result[self]],
    mission_name[__result[self]],
    rocket_name[rocket[__result[self]]],
    rocket_type[rocket[__result[self]]],
    launch_success[__result[self]],
    launch_date_utc[__result[self]],
    launch_year[__result[self]]
  )
  i <- i + 1

```

```

return launchs

```

```

response[Connect](self)
  return __response[self]

```

```

__init__[Launch](flight_number, mission_name, rocket, rocket_type,
launch_success, launch_year, launch_date)
  self <- Launch

```

```

  __flight_number[self] = flight_number
  __mission_name[self] = mission_name
  __launch_date[self] = launch_date
  __rocket[self] = rocket
  __rocket_type[self] = rocket_type
  __launch_success[self] = launch_success

```

```

    return self

__str__[Launch](self)
    if __launch_success[self] não é null then
        return flight_number[self]() + "\n" + mission_name[self]() + "\n" +
        rocket[self]() + "\n" + launch_year[self]() + "\n" + launch_date[self]() + "\n" +
        launch_success[self]()
    else
        return flight_number[self]() + "\n" + mission_name[self]() + "\n" +
        rocket[self]() + "\n" + launch_year[self]() + "\n" + launch_date[self]()

flight_number[Launch](self)
    return "Número do Voo: " + __flight_number[self]

mission_name[Launch](self)
    return "Missão: " + __mission_name[self]

launch_date[Launch](self)

    return "Data de Lançamento (UTC): " + __launch_date[self] formatado como
    dd/mm/yyyy às hh:mm

launch_year[Launch](self)

    return "Ano de Lançamento: " + __launch_year[self]

rocket[Launch](self)

    return "Foguete: " + __rocket[self] + " " + __rocket_type[self]

launch_success[Launch](self)

    if __launch_success[self] não é null then
        return "Lançamento realizado com sucesso!"

    return "Lançamento falhou!"

```

3. Instancie com alguns dados válidos o pseudocódigo demonstrando seu funcionamento passo a passo (variáveis, funcionalidades, etc). Isso evidencia que vcs entenderem o funcionamento do código.

Passo a passo da execução do código para mostrar o próximo lançamento

OPTION <- 5

```
__SHOW_RESULTS[SpaceX](OPTION)->  
  {if option == SpaceX.NEXT_LAUNCH é executado}->
```

```
__NEXT_LAUNCH[SpaceX]() ->  
  __init__[Connect]->  
  result[Connect] ->
```

```
result[Connect](self)  
  { if __result[self] é um dicionário é executado } ->  
  __init__[Launch] -> Fim
```

Irá retornar:

Número do Voo: X

Missão: XXXXX

Foguete: XXXXXX

Ano de Lançamento: XXXX

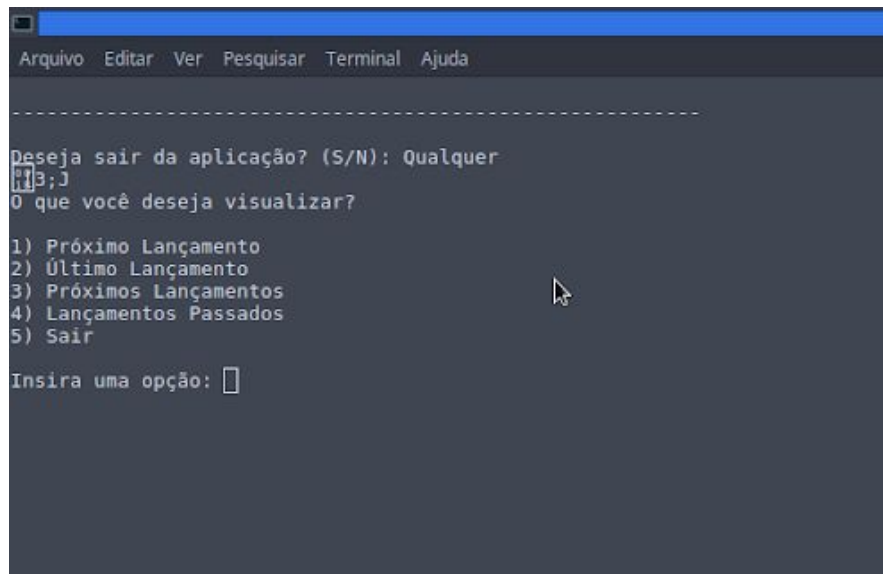
Data de Lançamento (UTC): XX/XX/XXXX às XX:XX

4. Pergunta-se: O código tem problemas? Defeitos? Se sim, quais e quais as suas sugestões de correção? Se não, o que vc achou da qualidade do código. Para isso defina o que para vc é qualidade de código.

Sim. Ao receber a resposta do usuário, o código não valida corretamente, ou seja, o usuário digita qualquer coisa e o programa interpreta como um não. Apenas a resposta “Sim” está bem definida. Para correção do código seria interessante adicionar uma condição “if else” para as respostas “Nao” e a partir daí um loop de correção para qualquer outro tipo de resposta que não deveriam ser aceitas. Qualidade de código pode ser definida através de boas práticas e as seguintes métricas:

- Volume, refere-se ao tamanho do código, mas também a produtividade por ponto de função.
- Redundância, análise de duplicidade de código.
- Complexidade, analisa a solução mais eficiente do problema, ou seja, o código que soluciona mas tem menor quantidade de linhas de execução.
- Acoplamento, analisa a dependência entre módulos.

5. Execute o código e copie as evidências de execução no relatório.



```
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda

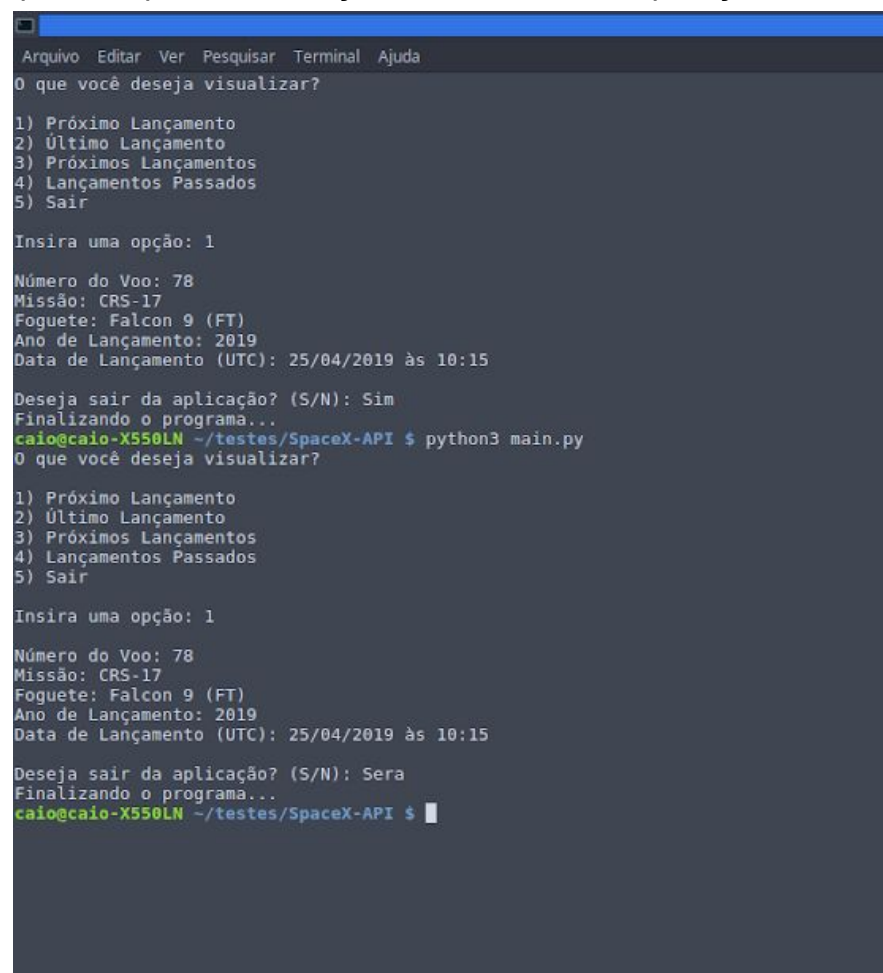
-----

Deseja sair da aplicação? (S/N): Qualquer
3;3
0 que você deseja visualizar?

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

Insira uma opção: 
```

Qualquer palavra que não começa com a letra “S”, a aplicação trata como “Não”.



```
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda

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): Sim
Finalizando o programa...
caio@caio-X550LN ~/testes/SpaceX-API $ 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): Sera
Finalizando o programa...
caio@caio-X550LN ~/testes/SpaceX-API $ 
```


Qualquer palavra que comece com a letra “S”, a aplicação trata como “SIM”.