

Modeling Language for Interaction as Conversation

MoLIC

Matias Giuliano Gutierrez Benitez
Vinicius Takeo Friedrich Kuwaki

Universidade do Estado de Santa Catarina

Seções

Introdução

Elementos e seus usos

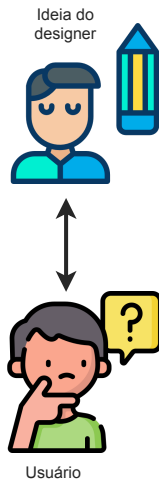
Construção de um exemplo

Conclusão

Atividade

Introdução

- *Modeling Language for Interaction as Conversation* (MoLIC) é uma **linguagem de modelagem** que permite modelar uma interface;
- Proposto por (PAULA, 2003) em sua Dissertação de Mestrado, apresenta um modelo baseado na Engenharia Semiótica;
- Busca se desapegar da ideia de linguagens de modelagem com ênfase no usuário e propõe uma linguagem com **foco na ideia do projetista** para com o usuário;
- Diferente de outras representações e modelos, a MoLIC **ênfatisa o papel do designer como interlocutor** durante a interação, através de seu preposto (a interface de usuário) (BARBOSA; SILVA, 2014).



- MoLIC é uma **linguagem de modelagem** que como o próprio nome já diz, é utilizada para modelar uma **interação**, tratando-a como uma **conversa**;
- Um exemplo de conversa pode ser visto ao lado;
 - U: representa uma fala do usuário;
 - D: representa uma fala do preposto do designer;

U: Bom dia.

D: Qual conta deseja manipular?

U: Conta 9876-5 da agência 345-6, senha *****.

D: O que deseja fazer?

U: Quero efetuar uma transferência.

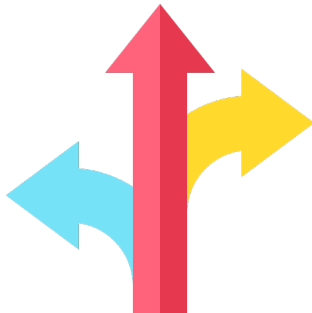
D: Para qual conta?

U: Para a conta pré-cadastrada de Ana Barros.

D: Qual é o valor?

U: R\$ 2.500,00.

- MoLIC é usado para **representar** todas as possíveis **interações** que os **usuários** podem ter com o **preposto do designer**, i.e, todos os caminhos possíveis de interação, incluindo **caminhos alternativos** para atingir o mesmo objetivo e **caminhos para a recuperação de falhas** (SILVA; BARBOSA, 2007);



Seções

Introdução

Elementos e seus usos

Construção de um exemplo

Conclusão

Atividade

Elementos e seus usos

- Veremos agora alguns elementos que podem ser utilizados na para modelar interações com o MoLIC;
 - Ponto de abertura;
 - Ponto de fechamento;
 - Cena, tópicos de cena e diálogo;
 - Falas;
 - Processamento do sistema;
 - Acesso ubíquo;
 - Comunicação síncrona;
- A ideia é compor esses elementos para modelar uma interação;

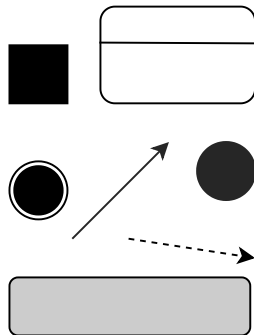


Figura 1: Alguns componentes da MoLIC.

Elementos e seus usos

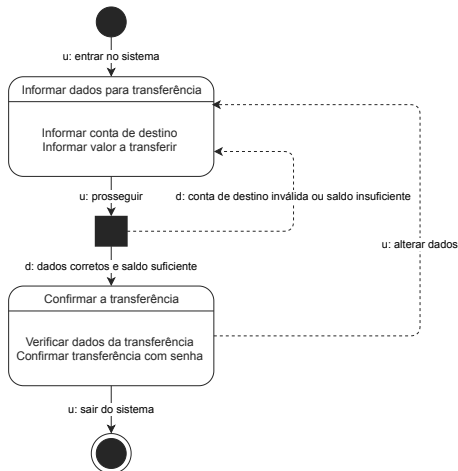


Figura 2: Exemplo de uso da MoLIC para modelar uma transferência bancária.

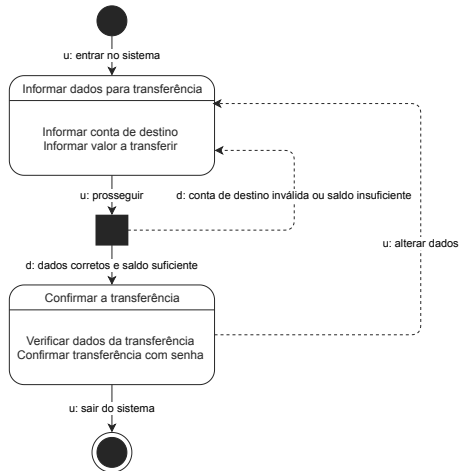
Elementos e seus usos

Ponto de abertura

- Onde a conversa se inicia;
- Podem existir mais de um;



Figura 3: Representação visual do ponto de abertura.



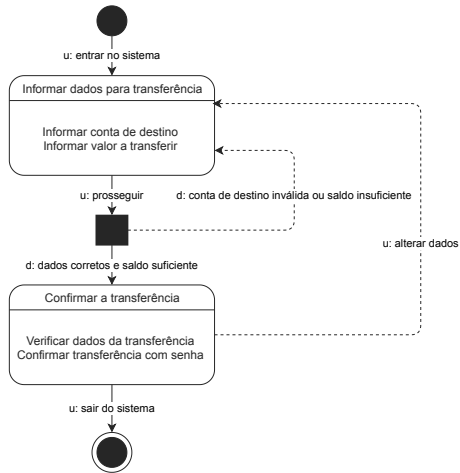
Elementos e seus usos

Ponto de fechamento

- Onde a conversa se encerra;



Figura 4: Representação visual do ponto de fechamento.



Elementos e seus usos

Cena

- Palco da conversa entre o usuário e o preposto do designer sobre um assunto específico;
- Representada por um retângulo de bordas arredondadas;
- Dividida em duas partes:
 - Tópico da cena;
 - Diálogo da cena;

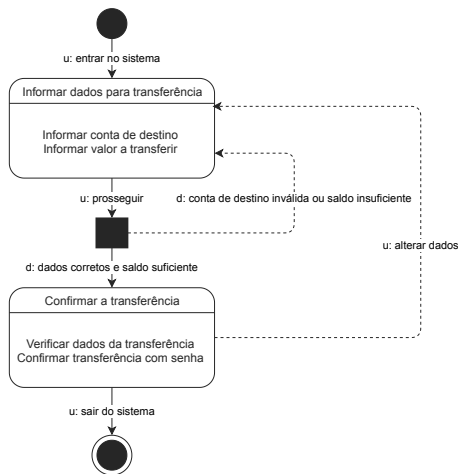


Figura 5: Representação visual de uma cena.

Elementos e seus usos

Tópico da cena

- Assunto da conversa travada na cena;
- Representado por uma frase no infinitivo (executar, calcular, efetuar etc.);
- Pode ser lida como:
 - Preposto do designer → usuário;
 - “Você agora deve **tópico**”.

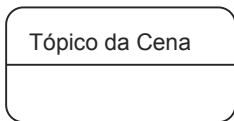
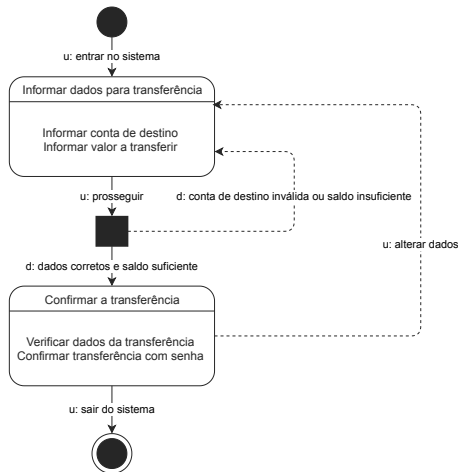


Figura 6: Representação visual do tópico da cena.



Elementos e seus usos

Diálogo da cena

- Unidades da conversa que focam diferentes aspectos do tópico da cena;

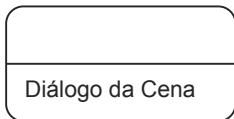
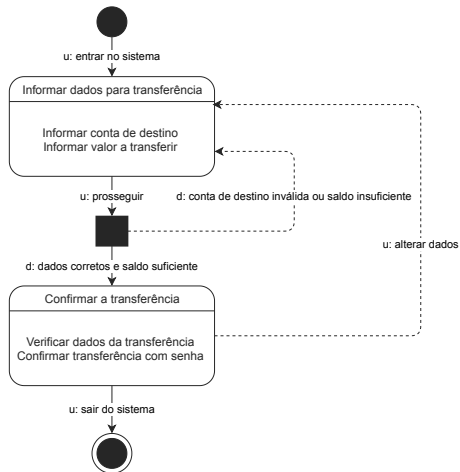


Figura 7: Representação visual do diálogo da cena.



Elementos e seus usos

Falas para troca de turno

- Representam trocas de turnos;
- Quando o controle é passado para outro:
 - Fala do usuário para troca de turno;
 - Fala do preposto do designer para troca de turno;

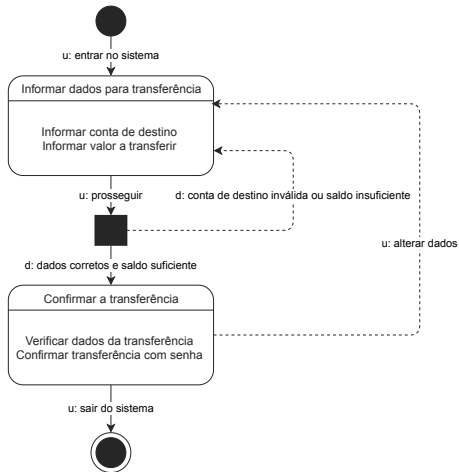


Figura 8: Representação visual de uma fala.

Elementos e seus usos

Fala do usuário para troca de turno

- Troca de turno em que o usuário passa o controle da conversa para o preposto do designer;

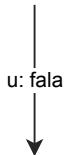
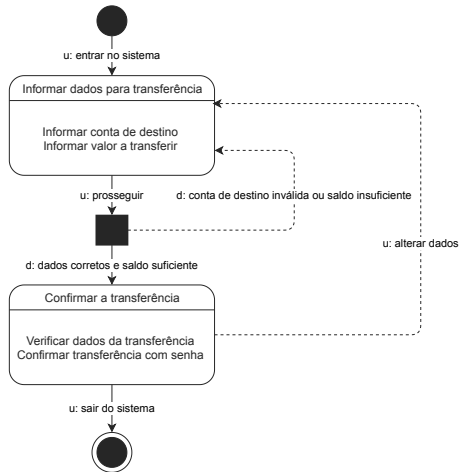


Figura 9: Representação visual.



Elementos e seus usos

Fala do preposto do designer para troca de turno

- Troca de turno em que o preposto do designer informa ao usuário sobre o resultado de um processamento;
- Pode devolver o controle da conversa para o usuário;

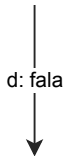
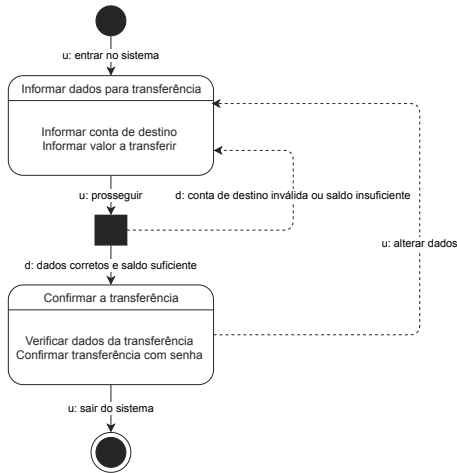


Figura 10: Representação visual.



Elementos e seus usos

Falas para recuperação de ruptura

- Representam rupturas no sistema ou na decisão do usuário;
 - Falas do preposto do usuário para recuperação de ruptura;
 - Falas do usuário para recuperação de ruptura;

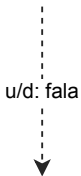
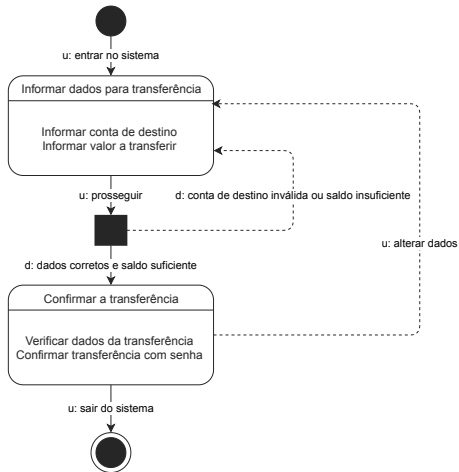


Figura 11: Representação visual.



Elementos e seus usos

Falas do preposto do usuário para recuperação de ruptura

- Troca de turno;
- Preposto do designer informa o usuário sobre um resultado inesperado de um processamento do sistema;

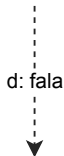
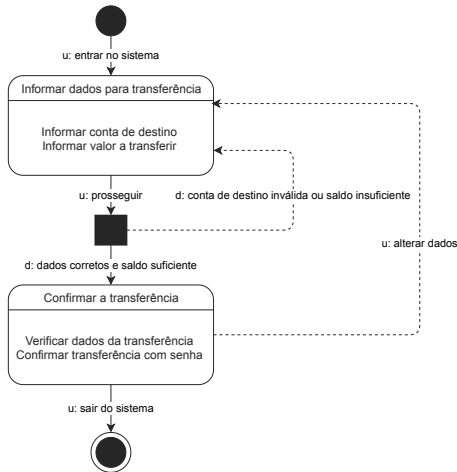


Figura 12: Representação visual.



Elementos e seus usos

Falas do usuário para recuperação de ruptura

- Troca de turno;
- Usuário percebe que se enganou e muda de ideia;
- Retornando para uma cena em que possa retificar parte da conversa travada previamente;

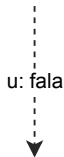
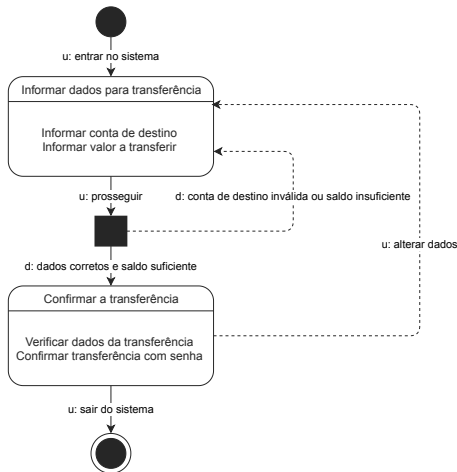


Figura 13: Representação visual.



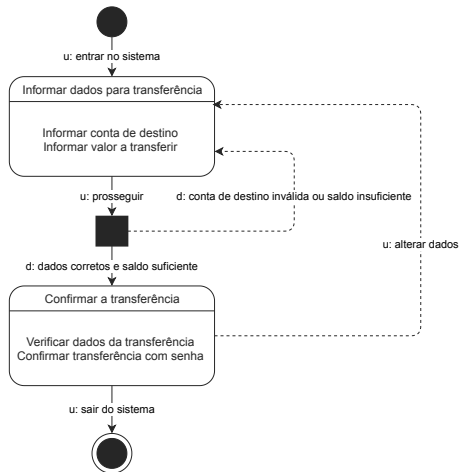
Elementos e seus usos

Processamento do sistema

- Momento oculto para o usuário;
- Aguarda uma operação ser realizada dentro do sistema;



Figura 14: Representação visual.



Elementos e seus usos

Acesso ubíquo

- Ponto de onde o usuário pode sair e voltar de qualquer outro lugar e em qualquer momento da interação;



Figura 15: Representação visual.

Elementos e seus usos

Pré-condições

- Indica, como o próprio nome já diz, uma pré-condição;
- Utilizado para a mudança de turno;
- Pode ser iniciada tanto pelo usuário, quanto pelo preposto do designer;

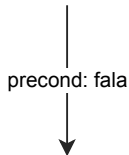


Figura 16: Representação visual.

Seções

Introdução

Elementos e seus usos

Construção de um exemplo

Conclusão

Atividade

Construção de um exemplo

- Não há uma ordem estrita para a construção de diagramas MoLIC;
- Vamos apresentar um possível caminho a seguir na hora de começar essa construção;

Construção de um exemplo

- Para exemplificar vamos construir um diagrama para um sistema bancário;
- O sistema possui três objetivos:
 - Consultar saldo;
 - Consultar extrato;
 - Efetuar transferência;

Construção de um exemplo

- O design da interação com MoLIC começa com a definição dos objetivos dos usuários que o sistema deve contemplar;
- Posicionando os objetivos no diagrama representando-os em forma de cenas;

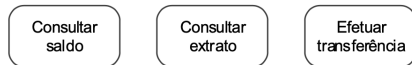


Figura 17: Início de construção de um diagrama MoLIC, representando apenas os objetivos em forma de cenas.

Construção de um exemplo

- Em seguida, é necessário identificar relações entre os objetivos do usuário;
- Deve-se fazer a pergunta: a conclusão de um objetivo motiva ao usuário a formular outro?
- Se sim coloca-se uma fala entre os objetivos;
- Deve-se pensar também de quem a fala está partindo (D ou U);

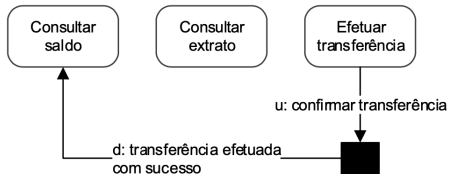


Figura 18: Inclusão de falas entre as cenas dos objetivos.

Construção de um exemplo

- Após, decide-se para quais cenas haverá acessos ubíquos;
- Normalmente os acessos ubíquos são projetados para os objetivos principais.

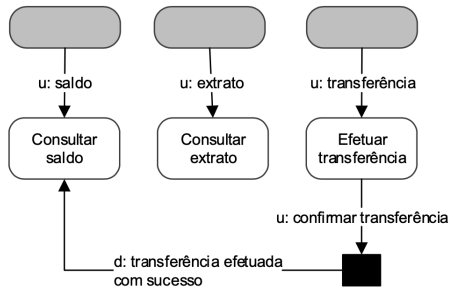


Figura 19: Adição dos acessos ubíquos para os objetivos.

Construção de um exemplo

- Depois, é definido o início da conversa com o sistema, ele é projetado através de pontos de abertura;
- Os pontos de abertura indicam se o usuário ao entrar no sistema possui um caminho pre-definido para algum objetivo ou se ele é levado a uma cena de início.

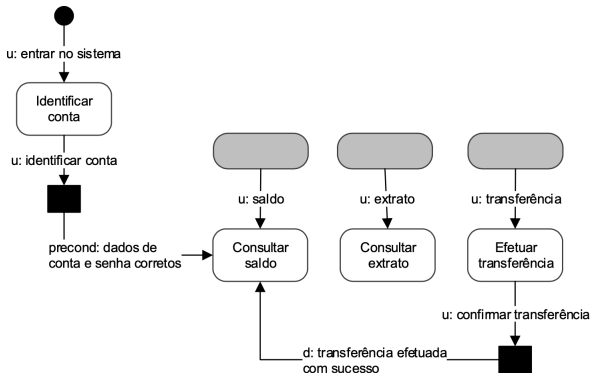


Figura 20: Ponto de abertura do sistema.

Construção de um exemplo

- A partir deste momento adicionar mais elementos ao diagrama depende do *designer*
- Ele pode tomar caminhos diferentes dependendo de qual seja o objetivo apontado.

Seções

Introdução

Elementos e seus usos





Construção de um exemplo

Conclusão

Atividade

Conclusão

- Apresentamos a MoLIC, alguns de seus elementos e exemplos de aplicações;
- Não foram abordados todos os seus componentes, visto que a ideia era apresentar uma prévia da linguagem;
- A MoLIC foi proposta inicialmente para representar a interação de um único usuário com o sistema.
- Foram propostas novas versões que atualizam certos aspectos da linguagem e até mesmo inclusões de novos recursos;
- (BARBOSA; SILVA, 2006) redefiniu e adicionou novos elementos que ajudam na interação em sistemas colaborativos ao MoLIC originalmente proposto por (PAULA, 2003).

-  BARBOSA, S. D. J.; SILVA, B. S. da. Molic segunda edição: Revisão de uma linguagem para modelagem da interação humano-computador. p. 153, 2006.
-  BARBOSA, S. D. J.; SILVA, B. S. da. Design da interação humano-computador com molic. **Livro dos Tutoriais do XIII Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais**, p. 109–138, 2014.
-  PAULA, M. Projeto da interação humano-computador baseado em modelos fundamentados na engenharia semiótica: construção de um modelo de interação. **Masters Dissertation. Informatics Department, PUC-Rio, Brasil**, 2003.
-  SILVA, B. S. D.; BARBOSA, S. D. J. Designing human-computer interaction with molic diagrams—a practical guide. **Monografias em Ciência da Computação**, Citeseer, v. 12, p. 12, 2007.

Seções

Introdução

Elementos e seus usos

Construção de um exemplo

Conclusão

Atividade

Exemplo

- Observe o trecho de código a seguir em Python:

Exemplo

```
import sys

email = ""

while True:
    print('Informe seu email: (Digite -1 para ENCERRAR)')
    email = input()
    if email == '-1':
        print('Encerrando...')
        sys.exit(1)
    if '@' in email:
        break
    print('Opcao invalida!')

print('Bem vindo {}'.format(email))
print('Encerrando...')
```

- O código requisita ao usuário que ele digite um endereço de e-mail;
- Até que o usuário digite 0;
- Caso o usuário insira um e-mail válido (cadeia de string contendo @) o usuário é recepcionado e o programa se encerra.
- Caso o usuário digite 0 o programa se encerra;

Exemplo

- Qual das figuras apresenta a representação MoLIC do código anterior?

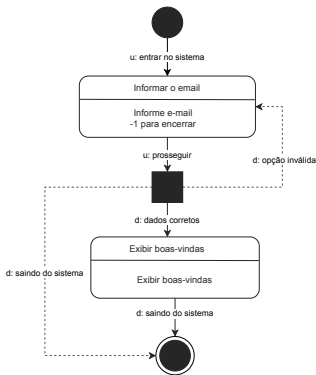


Figura 21: Alternativa A

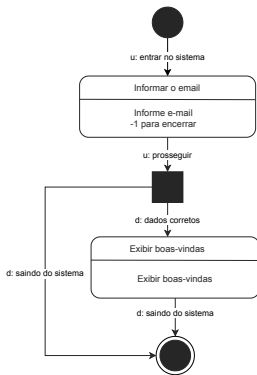


Figura 22: Alternativa B

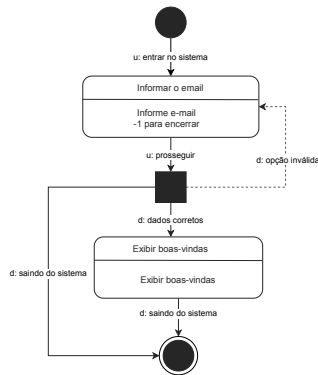


Figura 23: Alternativa C

Exemplo - Resposta

- A resposta é a **alternativa C**;
- É necessário que haja uma **recuperação de ruptura** após o processamento;
- Bem como uma **fala do preposto do usuário** após o processamento;

```
import sys
```

```
email = ''
```

```
while True:
```

```
    print('Informe seu email: (Digite -1 para ENCERRAR)')
```

```
    email = input()
```

```
    if email == '-1':
```

```
        print('Encerrando...')
```

```
        sys.exit(1)
```

```
    if '@' in email:
```

```
        break
```

```
    print('Opcao invalida!')
```

```
print('Bem vindo {}'.format(email))
```

```
print('Encerrando...')
```

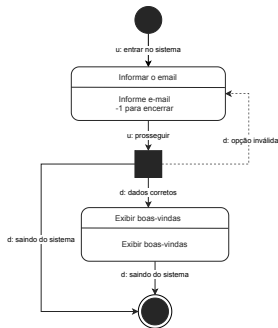


Figura 24: Alternativa C

- A partir do exemplo, use o MoLIC para representar o código-fonte na linguagem Python a ser descrito no slide seguinte.
- Três opções serão apresentadas para a discussão.
- Slides e o código-fonte podem ser encontrados nesse [link](#).

Atividade

```
def leValor() -> int:
    try:
        x = int(input())
        return x
    except Exception:
        return 0

soma = 0
while True:
    print('Digite 0 para SAIR')
    print('Digite 1 para SOMAR')

    lido = input()

    if lido == '0':
        break

    elif lido == '1':
        print('Informe um inteiro:')
        soma = soma + leValor()
        print('Valor da soma atual = {}'.format(soma))

print('Encerrando...')
```

- A função **leValor()** lê do usuário um número inteiro. Caso o usuário digite algo que não seja um valor inteiro, a função retorna 0.
- O programa requisita ao usuário que ele escolha uma opção:
 - 0 - O programa encerra;
 - 1 - O programa lê um valor, soma aos valores anteriores e exibe ao usuário o resultado.
 - Caso o usuário digite qualquer outro valor, o programa repete a pergunta.

Atividade - Opções

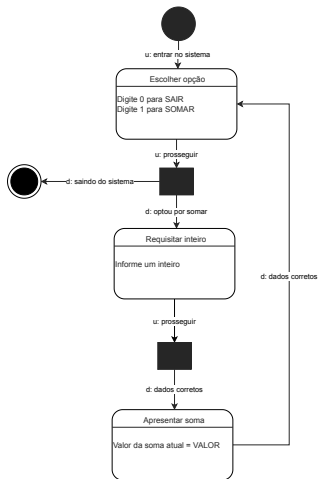


Figura 25: Alternativa A.

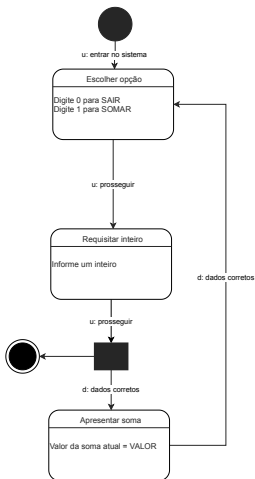


Figura 26: Alternativa B.

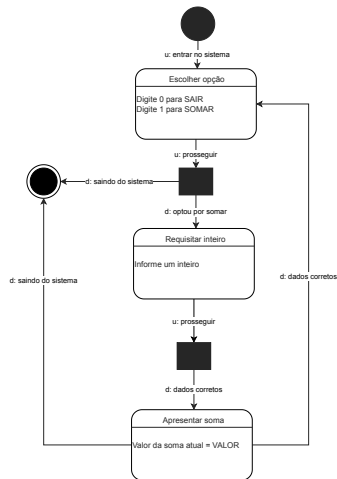


Figura 27: Alternativa C.

Atividade - Alternativa A

```
def leValor() -> int:
```

```
    try:
```

```
        x = int(input())
```

```
        return x
```

```
    except Exception:
```

```
        return 0
```

```
soma = 0
```

```
while True:
```

```
    print('Digite 0 para SAIR')
```

```
    print('Digite 1 para SOMAR')
```

```
    lido = input()
```

```
    if lido == '0':
```

```
        break
```

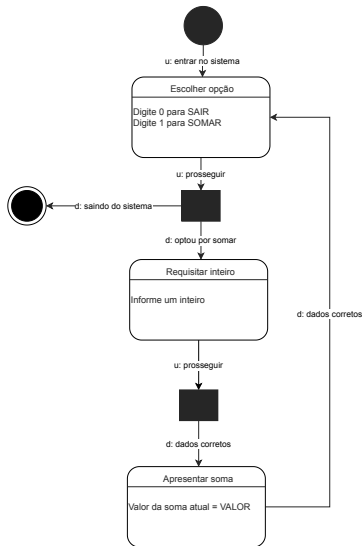
```
    elif lido == '1':
```

```
        print('Informe um inteiro:')
```

```
        soma = soma + leValor()
```

```
        print('Valor da soma atual = {}'.format(soma))
```

```
print('Encerrando...')
```



Atividade - Alternativa B

```
def leValor() -> int:
```

```
    try:
```

```
        x = int(input())
```

```
        return x
```

```
    except Exception:
```

```
        return 0
```

```
soma = 0
```

```
while True:
```

```
    print('Digite 0 para SAIR')
```

```
    print('Digite 1 para SOMAR')
```

```
    lido = input()
```

```
    if lido == '0':
```

```
        break
```

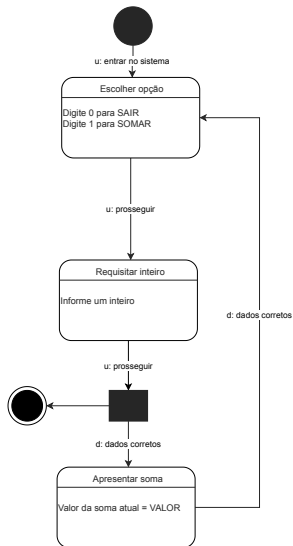
```
    elif lido == '1':
```

```
        print('Informe um inteiro:')
```

```
        soma = soma + leValor()
```

```
        print('Valor da soma atual = {}'.format(soma))
```

```
print('Encerrando...')
```



Atividade - Alternativa C

```
def leValor() -> int:
```

```
    try:
```

```
        x = int(input())
```

```
        return x
```

```
    except Exception:
```

```
        return 0
```

```
soma = 0
```

```
while True:
```

```
    print('Digite 0 para SAIR')
```

```
    print('Digite 1 para SOMAR')
```

```
    lido = input()
```

```
    if lido == '0':
```

```
        break
```

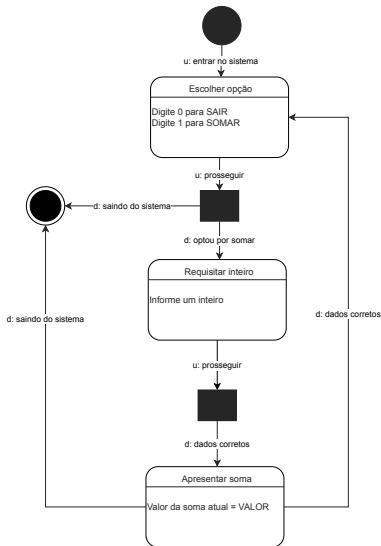
```
    elif lido == '1':
```

```
        print('Informe um inteiro:')
```

```
        soma = soma + leValor()
```

```
        print('Valor da soma atual = {}'.format(soma))
```

```
print('Encerrando...')
```





Attribution 4.0 International (CC BY 4.0)

Duvidas:

Matias Giuliano Gutierrez Benitez

Vinicius Takeo Friedrich Kuwaki

matiguti17@gmail.com

vtkwki@gmail.com