

# Notas de aula sobre ferramentas para modelagem e documentação de softwares (UML)

*Prof. Henrique Moura*

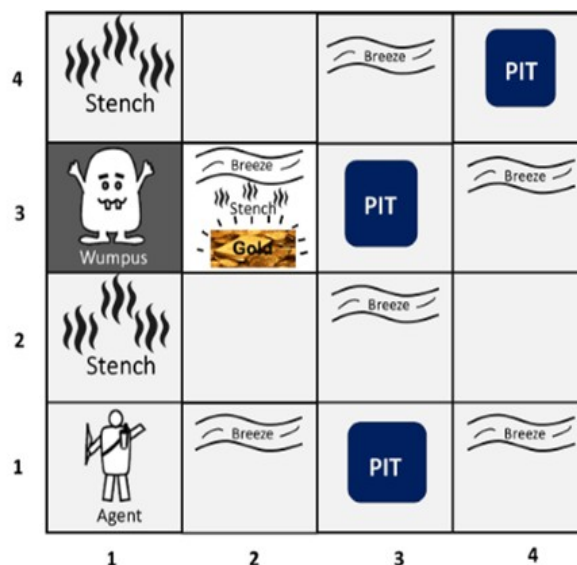
---

## 1) Definição de requisitos: lista de substantivos e verbos

A lista de substantivos e verbos é útil para o apontamento dos requisitos do projeto, de uma forma detalhada e sob o ponto de vista do cliente. Inicialmente, desejamos listar os substantivos que tenham potencial para classes, atributos ou dados. Em seguida, podemos listar os verbos com potencial para funções membros, associadas às classes anteriormente indicadas. Para o mini mundo Wumpus, a ferramenta em questão poderia ser aplicada da forma seguinte:

----->

Deseja-se construir um **universo** dentro do qual exista um monstro, denominado Wumpus, e um herói. Ambos, realizarão suas atividades dentro de uma **caverna**, composta por um **grid de 16 células** - cada célula representa uma sala, dentro da qual pode existir o **Wumpus**. O **herói** sempre começará a sua jornada, pela caverna, na primeira célula do grid (1, 1).



O Wumpus nunca se move. Inicialmente, a sua **posição** é definida pelo **programador da caverna**. Existirão, dentro da caverna, **poços** (PIT) capazes de matar o herói, caso o mesmo entre na sala específica. O herói não saberá onde ficam os poços, mas ele **perceberá** a existência de um poço nas proximidades da sala, através de uma **brisa**. Quando o herói se aproximar da sala do Wumpus o mesmo sentirá um **cheiro**. Dentro da caverna, o herói deverá encontrar o **ouro** e voltar para a primeira célula para finalizar com vida a sua jornada.

Sobre o herói:

- Trata-se de uma **inteligência artificial**, programada com **conhecimentos\_lógicos** capazes de operar os sinais de brisa e cheiro encontrados dentro da caverna, com o intuito de **descobrir** a posição de poços e do Wumpus.
- A cada **rodada**, o herói pode se movimentar em qualquer **direção**, para uma sala subjacente.
- O herói possui uma **flecha** que pode ser utilizada para atingir e matar o Wumpus. Neste caso, o herói escutará um **grito** indicando que o monstro morreu.
- O herói pode se **cansar** de movimentar pela caverna (**vida**). Se a “stamina” do herói acabar antes de sair da caverna o mesmo morre, imediatamente.
- O herói possui **memória** limitada dos lances que realizou, ou seja, das cavernas por onde ele passou. A memória também deve incluir o conhecimento sobre as salas que ele já vasculhou.
- O herói pode se mover-se nas direções Norte, Sul, Leste e Oeste;
- O herói pode **pegar** o ouro quando entrar na sala correta.
- O herói poderá **sair** da caverna se estiver portando o ouro, na câmara (1,1);

#### Sobre o Wumpus:

- O monstro não se movimenta na caverna.
- O monstro pode matar um herói que entre em sua sala.
- O monstro não possui inteligência, ou seja, não se **lembra** de suas ações.

#### Regras gerais:

- +1000 (vida) por sair da caverna com o tesouro;
- -1000 (vida) por cair em um poço ou ser devorado pelo Wumpus;
- -1 (medida do cansaço) para cada tomada de ação;
- -10 por usar a flecha;
- O jogo termina quando o agente morre ou sai da caverna;

#### Sobre a caverna:

- Um grid 4x4 de salas, com paredes ao redor;
- O agente começa no quadrado (1,1) voltado para o leste;
- O programador deve definir a posição do Wumpus, dos poços e do ouro.

#### Sobre os sinais:

- Nas câmaras diretamente adjacentes (não na diagonal) ao Wumpus, o agente percebe seu ODOR característico.
- Nas câmaras diretamente adjacentes (não na diagonal) a um poço, o agente percebe uma BRISA.
- Na câmara onde está o ouro, o agente percebe um BRILHO intenso;

#### Lista de Substantivos e Verbos:

| Substantivos                    | Proposta                 |
|---------------------------------|--------------------------|
| Caverna (universo)              | Classe                   |
| Grid 4x4                        | Atributo                 |
| Wumpus                          | Classe                   |
| Herói (inteligência artificial) | Classe (usuário interno) |

|  |                                      |
|--|--------------------------------------|
| Posição do Wumpus, poços e herói         | Atributo                             |
| Programador da caverna (usuário externo) | Usuário                              |
| Cheiro                                   | Atributo (conhecimento do herói)     |
| Brisa                                    | Atributo (conhecimento do herói)     |
| Brilho (ouro)                            | Atributo (conhecimento do herói)     |
| Rodada                                   | Atributo                             |
| Direção                                  | Atributo (Norte, Sul, Leste e Oeste) |
| Flecha                                   | Atributo                             |
| Grito (morte do Wumpus)                  | Atributo (conhecimento do herói)     |
| Pontos de vida                           | Atributo                             |
| Memória (salas ou movimentações)         | Atributo do herói                    |

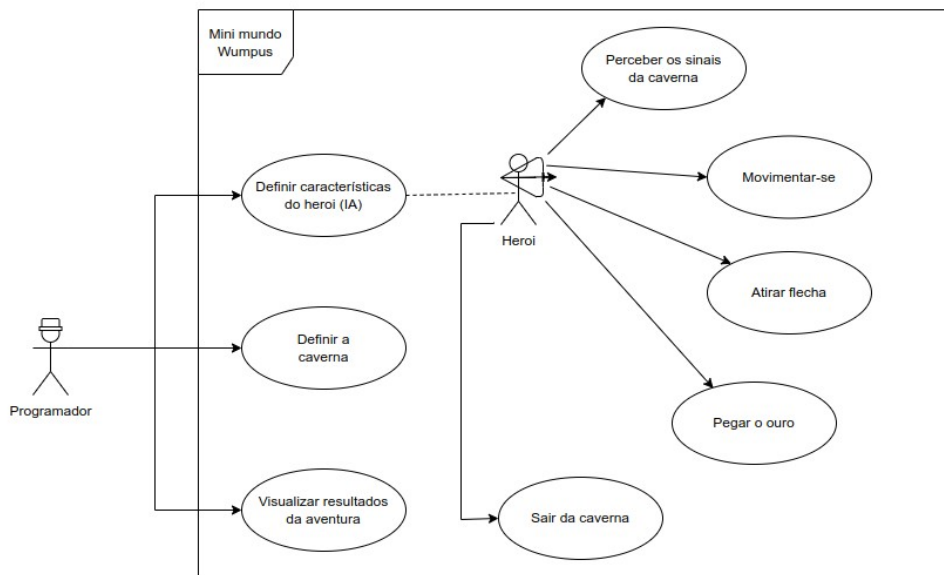
| Verbos                             | Proposta        |
|------------------------------------|-----------------|
| Mover                              | Membro do herói |
| Perceber (cheiro, brisas e brilho) | Membro do herói |
| Descobrir (relações lógicas)       | Membro do herói |
| Atirar flecha                      | Membro do herói |
| Lembrar                            | Membro do herói |
| Pegar o ouro                       | Membro do herói |
| Sair da caverna                    | Membro do herói |
| Cansar                             | Membro do herói |

Algumas “classes” poderão ser suprimidas durante a modelagem, da mesma forma que novas classes, bem como membros, poderão ser adicionadas.

## 2) Diagrama de caso de uso

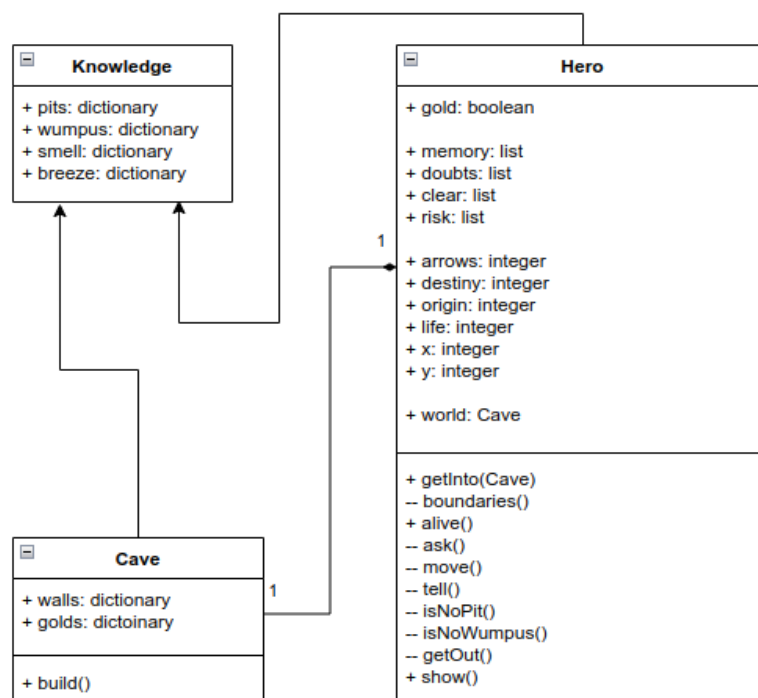
O diagrama de caso de uso é uma representação visual para os requisitos do sistema que se pretende modelar. Os requisitos de um software são as especificações das funcionalidades que o sistema deve ter, bem como as restrições e as expectativas dos usuários em relação ao software. Trata-se de uma perspectiva do usuário sobre o sistema. Seguindo o exemplo do mini mundo Wumpus, podemos gerar o resultado a seguir.

Observamos que deve existir um programador (agente externo) encarregado de construir a caverna (definir as posições e quantidades dos poços, do Wumpus e do ouro) e definir as características da IA. Se necessário, podemos solicitar uma outra conversa com o cliente para que o mesmo estabeleça detalhes da inteligência artificial do herói. Neste caso, estes detalhes ficaram em aberto, ou seja, ao critério do desenvolvedor de software.



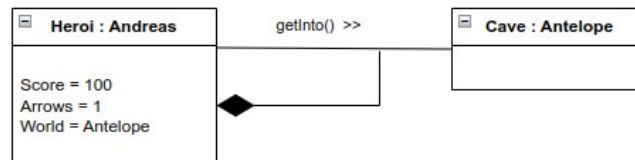
### 3) Diagrama de classes

Este é o próximo estágio após a definição das entidades com potencial para classes, por parte dos requisitos apontados pelas ferramentas anteriores. O diagrama de classes pode ser realizado em etapas, desde sua forma mais simples, contendo apenas as classes isoladas, até a sua forma mais completa, envolvendo as associações entre classes. Nesta etapa, estamos adicionando ao projeto a propriedade de ENCAPSULAMENTO, que é o primeiro pilar de um projeto OO. Para o mundo Wumpus, os resultados estão apresentados a seguir.



## 4) Diagrama de objetos

Diferentemente do diagrama de classes, o diagrama de objeto representa uma instância específica do diagrama de classes. Em outras palavras, o diagrama de objeto deve apresentar uma solução específica do problema, enquanto o diagrama de classes apresenta a modelagem necessária para tal. Por este motivo, o diagrama de objetos será diferente do diagrama de classes. A solução desenhada para o exemplo mini mundo Wumpus está apresentada a seguir.



O objeto Andreas, obtido da classe Heroi, foi definido com vida igual a 100 e com apenas 1 flecha (é possível ter mais de uma flecha). O diagrama indica que o Andreas chama a caverna através da função getInto() e esta chamada resulta em uma composição. Não é necessário indicar a multiplicidade, porque estamos tratando a comunicação entre dois objetos apenas – esta solução específica possui apenas dois objetos.

## 5) Diagrama de Fluxo de Dados

Modela uma perspectiva da solução que, geralmente, indique a simulação completa do fluxo de informações ao longo da operação do sistema. É necessário para a lapidação de requerimentos, módulos, operações, processos, dentre outros elementos necessários para a sustentação da solução. Abaixo, segue uma lista dos elementos gráficos constituintes de um DFD padrão:

- Processos: retângulos com três partes (pode ter cantos arredondados)
  - Representam partes do código relacionadas com o uso ou com a transformação de informações.
  - Deve ser indicado por nomes verbos imperativos
  - Não podem ser “fontes” ou “Destino” de informações
  - O processo não é, necessariamente, a representação de um único objeto!
- Fluxo de informações: setas
  - Determina a transferência das informações ao longo do sistema.
  - A informação pode ser descrita na seta, para maior clareza da solução (substantivos)

- Sempre flui para processos e entre processos, mas pode representar informações entre entidades externas também
- Não pode ser utilizada entre entidades de armazenamento de informações (banco de dados)
- Não pode ser utilizada entre banco de dados e entidades externas
- Indica o estado momentânea das informações, ao longo do sistema
- Entidades externas: elipses (podem ser círculos)
  - Determinam as fontes de informação (entrada de dados).
  - Pode enviar ou receber informações de um processo
  - Deve se comunicar com o sistema de alguma forma
- Entidade de armazenamento de informações (banco de dados): retângulo aberto com duas partes
  - Relacionado com o armazenamento de informações, temporárias ou não

Alguns modelos de DFD, e ferramentas gráficas, fazem uso diferente das formas geométricas apresentadas, indicando processos com círculos ou elípticos e entidades externas com retângulos (o mais importante é não misturar as formas no mesmo mapa). O DFD pode ser elaborado em níveis, com diferentes profundidades e detalhamento! Exemplos a seguir.

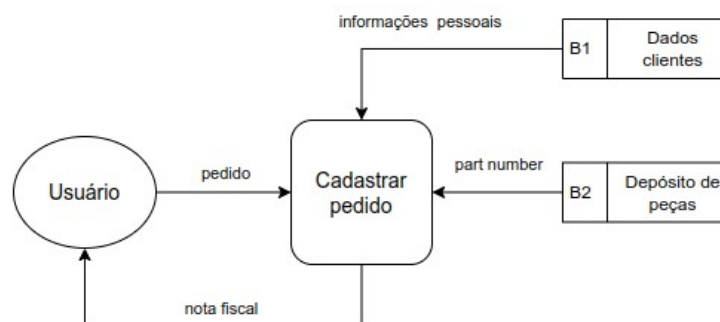


Diagrama de fluxo de dados de nível ZERO.

No DFD acima, temos a uma representação de nível ZERO, pois não temos o detalhamento do processo “Cadastrar Pedido”, apenas as entradas e saídas do mesmo. A seguir, este mesmo processo foi explodido em mais etapas, a fim de detalhar o fluxo das informações. A este novo diagrama podemos associar o rótulo “DFD de nível 1”. Mais níveis poderão ser desenhados com mais detalhes, se necessário.

