

Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio  
Departamento de Engenharia Civil e Ambiental  
Programa de Pós-Graduação em Engenharia Civil (Estruturas)

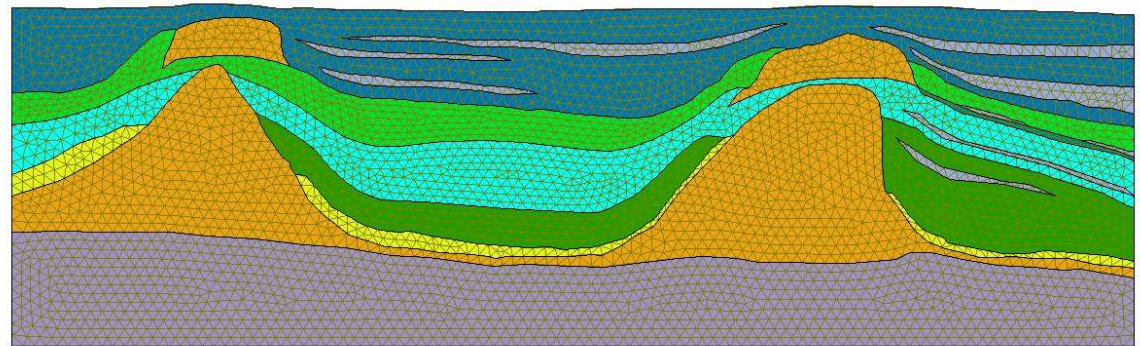
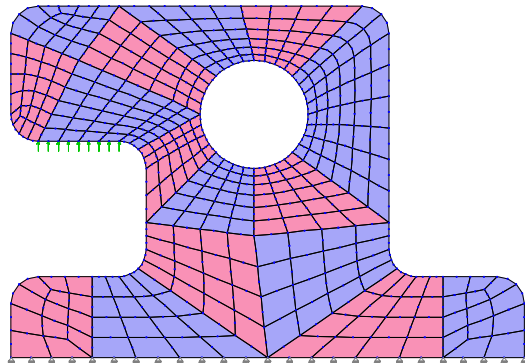
# UMA ESTRATÉGIA ABERTA E EXTENSÍVEL PARA CRIAÇÃO DE MODELOS DE SUBDIVISÕES PLANARES PARA MECÂNICA COMPUTACIONAL

Danilo Silva Bomfim

Orientador: Luiz Fernando Martha

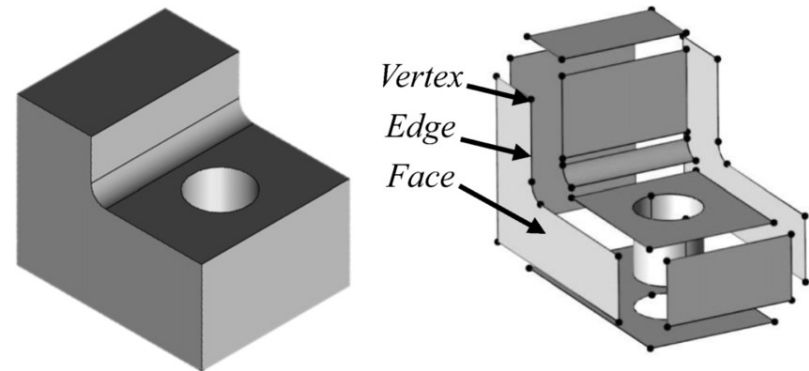
# Introdução

- Modelagem geométrica de sólidos
- Várias aplicações nas áreas da Engenharia, Geologia, Cartografia e Geografia



# Introdução

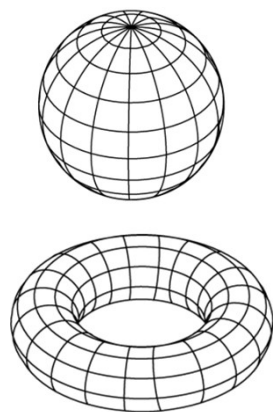
- Existem varias técnicas utilizadas para o armazenamento e gerenciamento das informações de uma subdivisão planar
- Representação de fronteira
  - Winged-Edge (Baumgart)
  - **Half-Edge (Mäntylä)**
  - Quad-Edge (Guibas e Stolf)
  - Radial-Edge (Weiler)



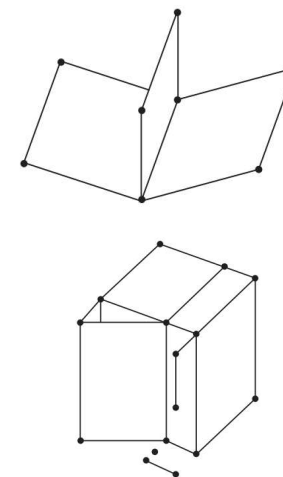
# Introdução

- Half-Edge foi desenvolvida para representar superfícies de sólidos *2-manifold*
- *Sólido 2-manifold*:
  - Sólido cuja as fronteiras são superfícies fechadas que são variedades de dimensão 2

2-manifold →

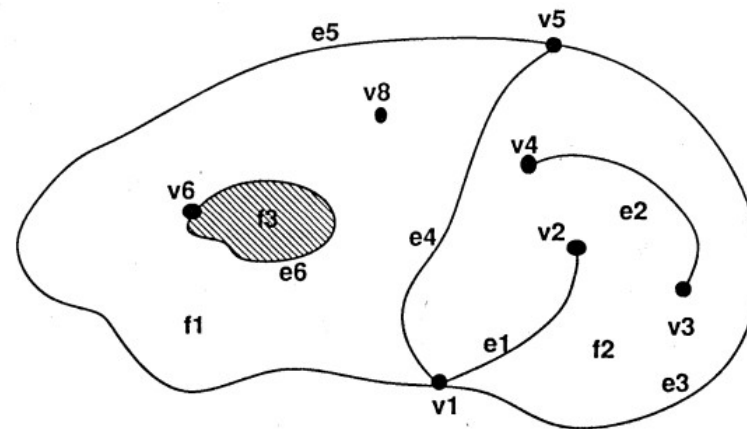


Não-manifold →



# Introdução

- As subdivisões planares podem ser modeladas como a superfície de um sólido *2-manifold* planificado. Isso justifica o aparecimento da face infinita (*fext*) que não apresenta uma fronteira externa definida

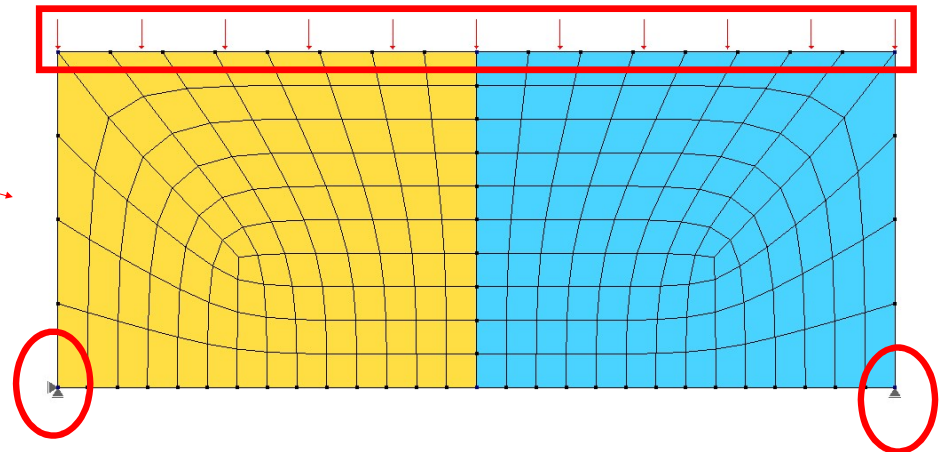


Face infinita  $\longrightarrow$  *fext*

# Introdução

- Estrutura de dados topológica pode ser aplicada em diversas problemas de modelagem
- A topologia e a geometria não são suficientes para modelar completamente um problema de Engenharia ou Geologia

- Atributos de modelagem
- Gerenciamento de atributos:
  - Configurável e extensível



# Introdução

- Quais são as vantagens da estrutura de dados topológica na modelagem de sólidos?
  - Sistema estruturado e organizado de informações
  - Velocidade e eficiência
  - Evita consultas globais (consultas locais)
  - Redução da quantidade de erros numéricos

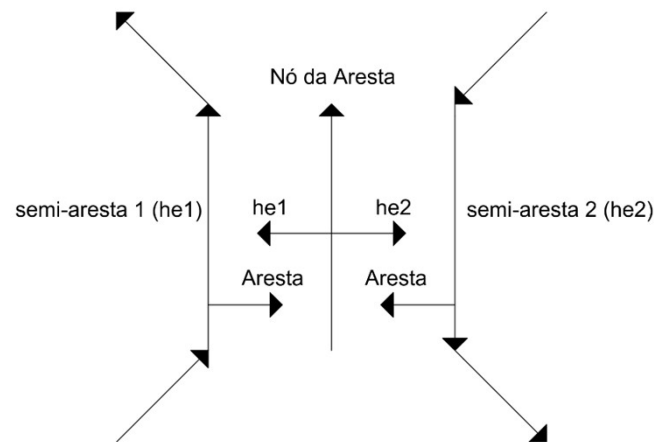
# Objetivos

- Implementação da biblioteca HETOOL
  - Python
  - Half-Edge
  - Arquitetura orientada a objetos
  - Mecânica computacional (atributos especiais)
  - Pontos e curvas poligonais
  - Interseção automática dos elementos geométricos modelados
  - Reconhecimento automático de regiões fechadas
  - Gerenciamento genérico de atributos
  - Não é preciso entender sobre a topologia para utilizá-la
  - Desenvolvimento de aplicativos
  - Código aberto e extensível



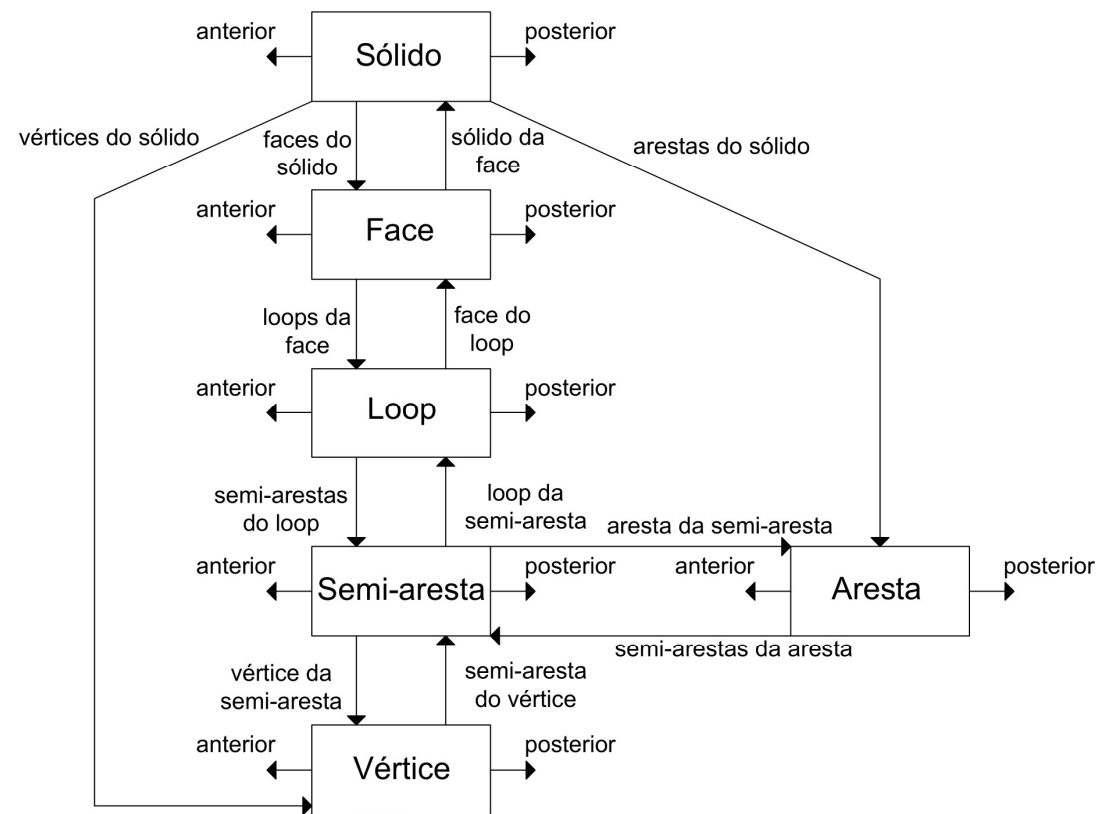
# Half-Edge

- Se baseia no fato de que cada aresta é delimitada por exatamente duas faces, permitindo a aresta ser separada em duas semi-arestas que são orientadas em direções opostas

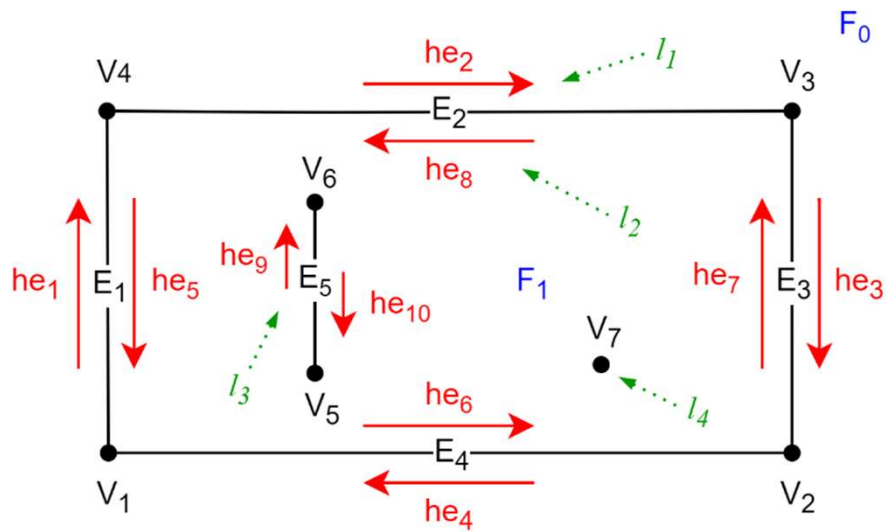


- Entidade abstrata *half-edge*

# Half-Edge



# Half-Edge



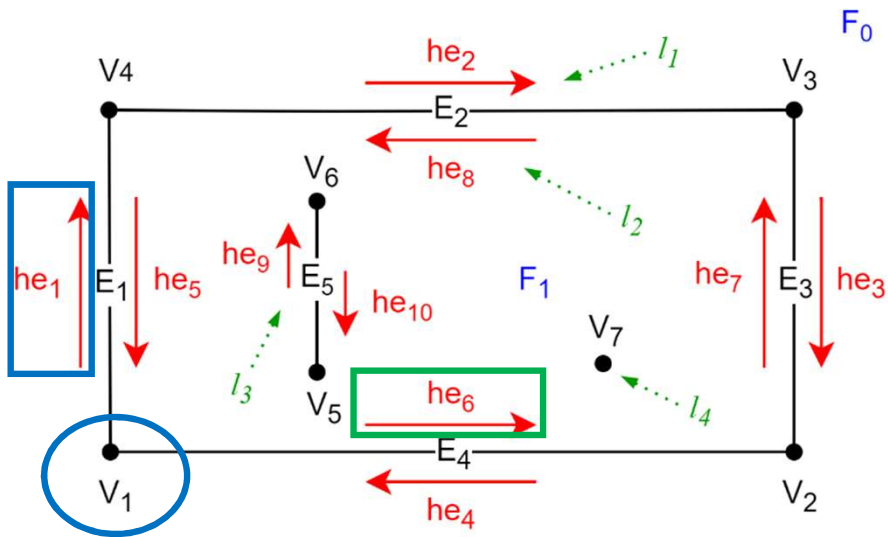
Vértices	Coordenadas	Semi-aresta
$V_1$	$x_1 y_1$	$he_1$
$V_2$	$x_2 y_2$	$he_4$
$V_3$	$x_3 y_3$	$he_3$
$V_4$	$x_4 y_4$	$he_2$
$V_5$	$x_5 y_5$	$he_9$
$V_6$	$x_6 y_6$	$he_{10}$
$V_7$	$x_7 y_7$	—

Arestas	Semi-arestas
$E_1$	$he_1$ e $he_5$
$E_2$	$he_2$ e $he_8$
$E_3$	$he_3$ e $he_7$
$E_4$	$he_4$ e $he_6$
$E_5$	$he_9$ e $he_{10}$

Loops	Semi-aresta	Face
$l_1$	$he_1$	$F_0$
$l_2$	$he_5$	$F_1$
$l_3$	$he_9$	$F_1$
$l_4$	—	$F_1$

Faces	Loop_externo	Loop_interno
$F_0$	—	$l_1$
$F_1$	$l_2$	$l_3$ e $l_4$

# Half-Edge



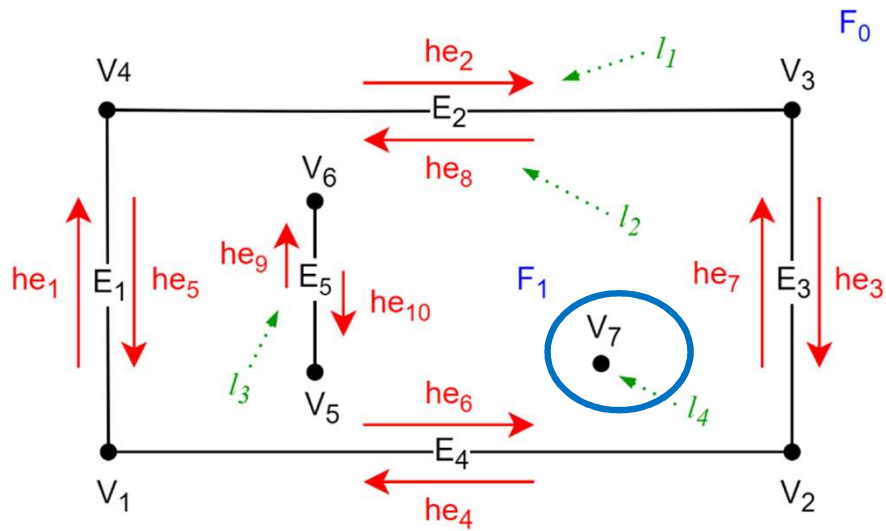
Vértices	Coordenadas	Semi-aresta
$V_1$	$x_1 y_1$	$he_1$
$V_2$	$x_2 y_2$	$he_4$
$V_3$	$x_3 y_3$	$he_3$
$V_4$	$x_4 y_4$	$he_2$
$V_5$	$x_5 y_5$	$he_9$
$V_6$	$x_6 y_6$	$he_{10}$
$V_7$	$x_7 y_7$	—

Arestas	Semi-arestas
$E_1$	$he_1$ e $he_5$
$E_2$	$he_2$ e $he_8$
$E_3$	$he_3$ e $he_7$
$E_4$	$he_4$ e $he_6$
$E_5$	$he_9$ e $he_{10}$

Loops	Semi-aresta	Face
$l_1$	$he_1$	$F_0$
$l_2$	$he_5$	$F_1$
$l_3$	$he_9$	$F_1$
$l_4$	—	$F_1$

Faces	Loop_externo	Loop_interno
$F_0$	—	$l_1$
$F_1$	$l_2$	$l_3$ e $l_4$

# Half-Edge



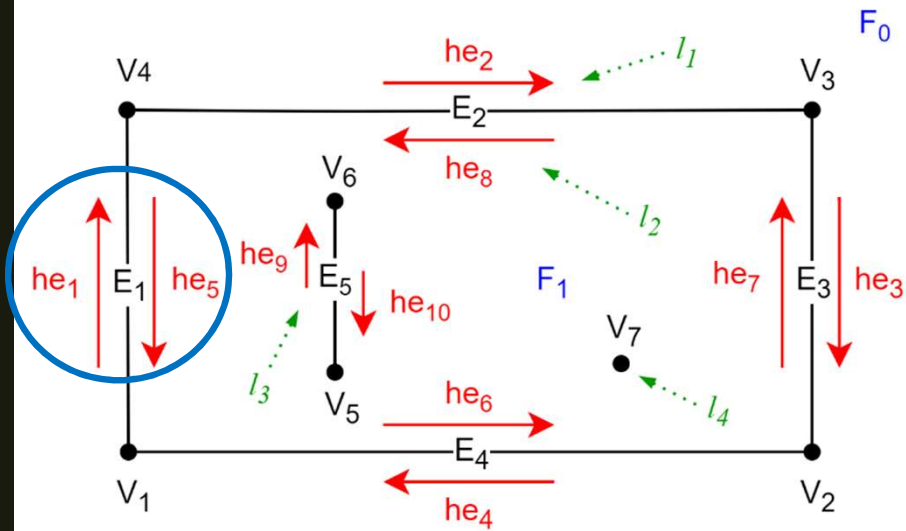
Vértices	Coordenadas	Semi-aresta
$V_1$	$x_1 y_1$	$he_1$
$V_2$	$x_2 y_2$	$he_4$
$V_3$	$x_3 y_3$	$he_3$
$V_4$	$x_4 y_4$	$he_2$
$V_5$	$x_5 y_5$	$he_9$
$V_6$	$x_6 y_6$	$he_{10}$
$V_7$	$x_7 y_7$	—

Arestas	Semi-arestas
$E_1$	$he_1$ e $he_5$
$E_2$	$he_2$ e $he_8$
$E_3$	$he_3$ e $he_7$
$E_4$	$he_4$ e $he_6$
$E_5$	$he_9$ e $he_{10}$

Loops	Semi-aresta	Face
$l_1$	$he_1$	$F_0$
$l_2$	$he_5$	$F_1$
$l_3$	$he_9$	$F_1$
$l_4$	—	$F_1$

Faces	Loop_externo	Loop_interno
$F_0$	—	$l_1$
$F_1$	$l_2$	$l_3$ e $l_4$

# Half-Edge



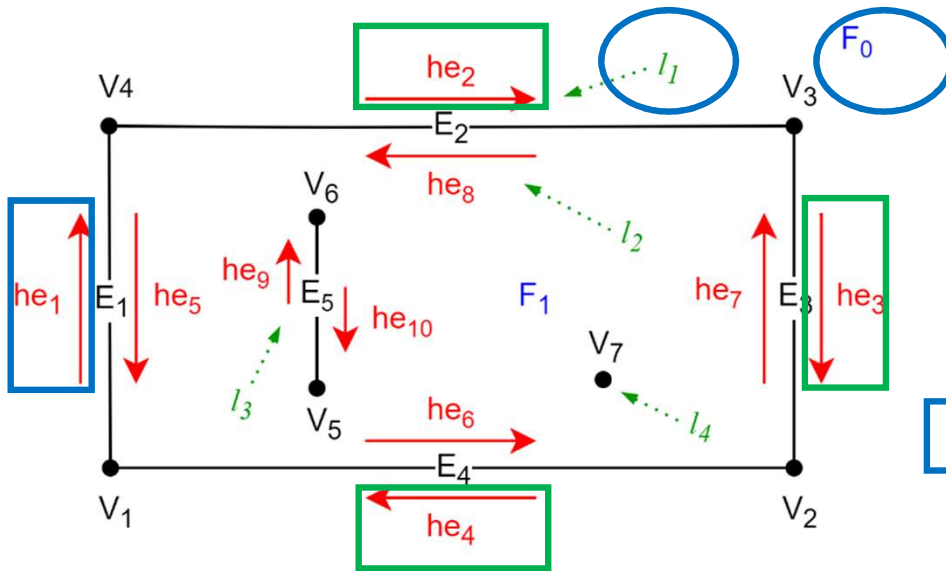
Vértices	Coordenadas	Semi-aresta
$V_1$	$x_1 y_1$	$he_1$
$V_2$	$x_2 y_2$	$he_4$
$V_3$	$x_3 y_3$	$he_3$
$V_4$	$x_4 y_4$	$he_2$
$V_5$	$x_5 y_5$	$he_9$
$V_6$	$x_6 y_6$	$he_{10}$
$V_7$	$x_7 y_7$	—

Arestas	Semi-arestas
$E_1$	$he_1$ e $he_5$
$E_2$	$he_2$ e $he_8$
$E_3$	$he_3$ e $he_7$
$E_4$	$he_4$ e $he_6$
$E_5$	$he_9$ e $he_{10}$

Loops	Semi-aresta	Face
$l_1$	$he_1$	$F_0$
$l_2$	$he_5$	$F_1$
$l_3$	$he_9$	$F_1$
$l_4$	—	$F_1$

Faces	Loop_externo	Loop_interno
$F_0$	—	$l_1$
$F_1$	$l_2$	$l_3$ e $l_4$

# Half-Edge



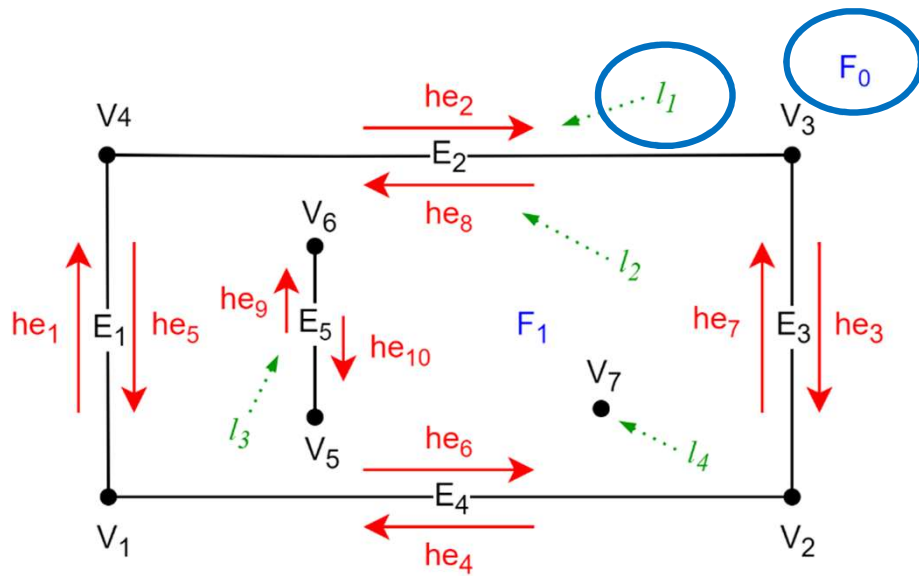
Vértices	Coordenadas	Semi-aresta
$V_1$	$x_1 y_1$	$he_1$
$V_2$	$x_2 y_2$	$he_4$
$V_3$	$x_3 y_3$	$he_3$
$V_4$	$x_4 y_4$	$he_2$
$V_5$	$x_5 y_5$	$he_9$
$V_6$	$x_6 y_6$	$he_{10}$
$V_7$	$x_7 y_7$	—

Arestas	Semi-arestas
$E_1$	$he_1$ e $he_5$
$E_2$	$he_2$ e $he_8$
$E_3$	$he_3$ e $he_7$
$E_4$	$he_4$ e $he_6$
$E_5$	$he_9$ e $he_{10}$

Loops	Semi-aresta	Face
$l_1$	$he_1$	$F_0$
$l_2$	$he_5$	$F_1$
$l_3$	$he_9$	$F_1$
$l_4$	—	$F_1$

Faces	Loop_externo	Loop_interno
$F_0$	—	$l_1$
$F_1$	$l_2$	$l_3$ e $l_4$

# Half-Edge



Vértices	Coordenadas	Semi-aresta
$V_1$	$x_1 y_1$	$he_1$
$V_2$	$x_2 y_2$	$he_4$
$V_3$	$x_3 y_3$	$he_3$
$V_4$	$x_4 y_4$	$he_2$
$V_5$	$x_5 y_5$	$he_9$
$V_6$	$x_6 y_6$	$he_{10}$
$V_7$	$x_7 y_7$	—

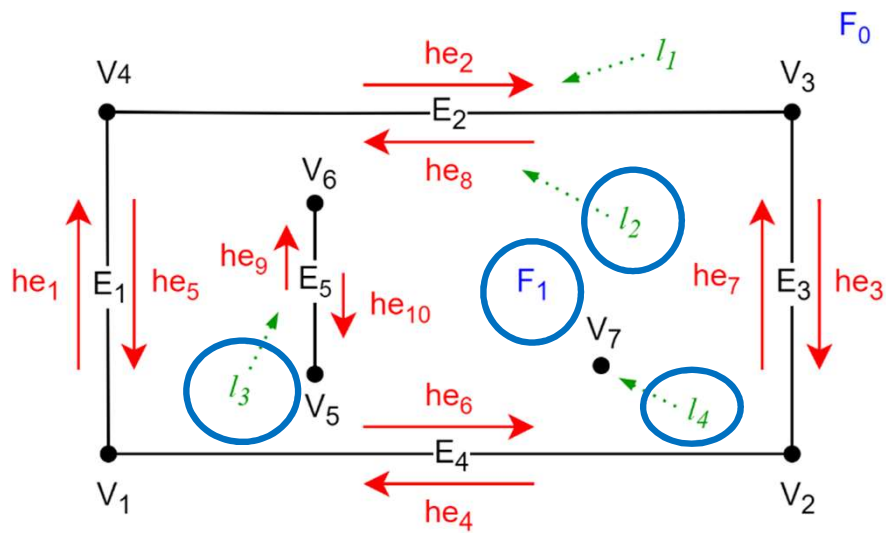
Arestas	Semi-arestas
$E_1$	$he_1$ e $he_5$
$E_2$	$he_2$ e $he_8$
$E_3$	$he_3$ e $he_7$
$E_4$	$he_4$ e $he_6$
$E_5$	$he_9$ e $he_{10}$

Loops	Semi-aresta	Face
$l_1$	$he_1$	$F_0$
$l_2$	$he_5$	$F_1$
$l_3$	$he_9$	$F_1$
$l_4$	—	$F_1$

Faces	Loop_externo	Loop_interno
$F_0$	—	$l_1$
$F_1$	$l_2$	$l_3$ e $l_4$



# Half-Edge



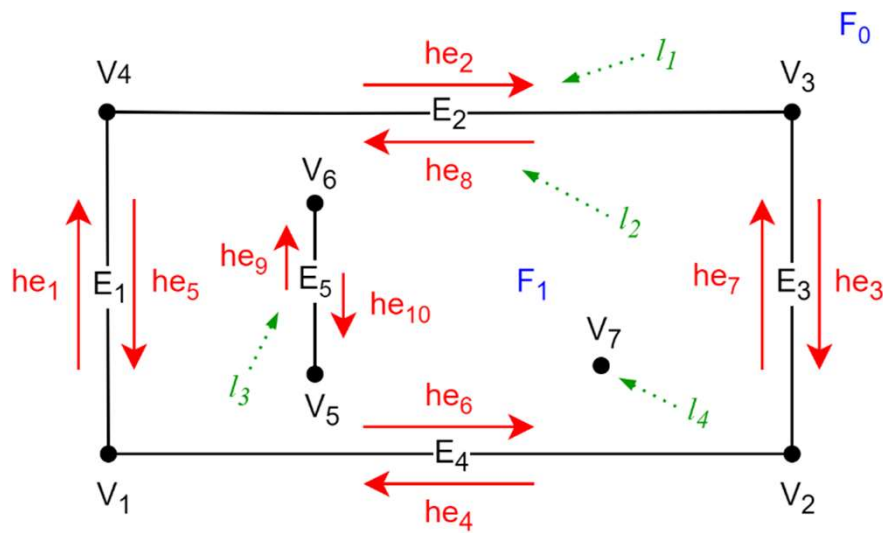
Vértices	Coordenadas	Semi-aresta
$V_1$	$x_1 y_1$	$he_1$
$V_2$	$x_2 y_2$	$he_4$
$V_3$	$x_3 y_3$	$he_3$
$V_4$	$x_4 y_4$	$he_2$
$V_5$	$x_5 y_5$	$he_9$
$V_6$	$x_6 y_6$	$he_{10}$
$V_7$	$x_7 y_7$	—

Arestas	Semi-arestas
$E_1$	$he_1$ e $he_5$
$E_2$	$he_2$ e $he_8$
$E_3$	$he_3$ e $he_7$
$E_4$	$he_4$ e $he_6$
$E_5$	$he_9$ e $he_{10}$

Loops	Semi-aresta	Face
$l_1$	$he_1$	$F_0$
$l_2$	$he_5$	$F_1$
$l_3$	$he_9$	$F_1$
$l_4$	—	$F_1$

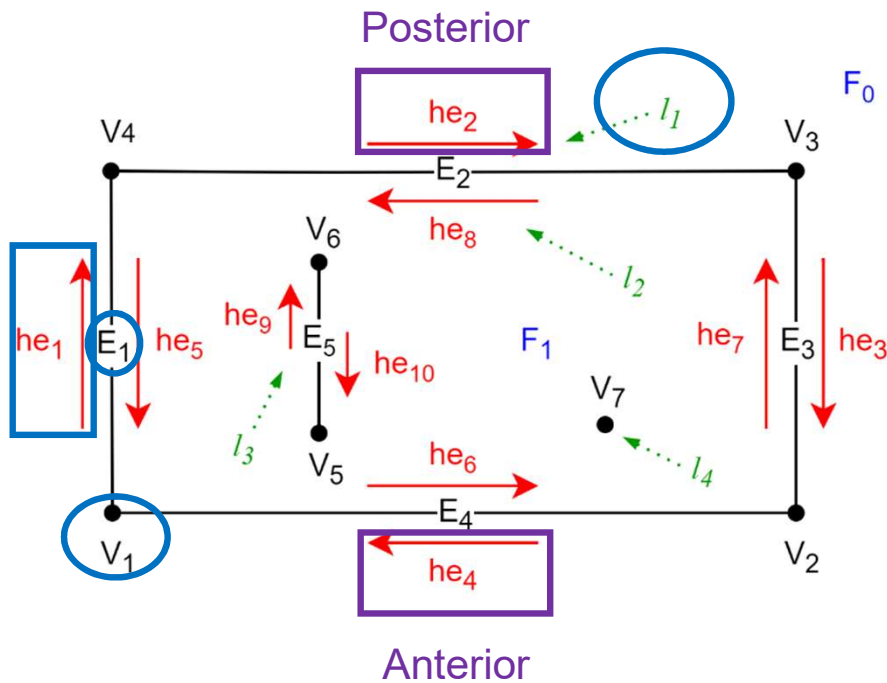
Faces	Loop_externo	Loop_interno
$F_0$	—	$l_1$
$F_1$	$l_2$	$l_3$ e $l_4$

# Half-Edge



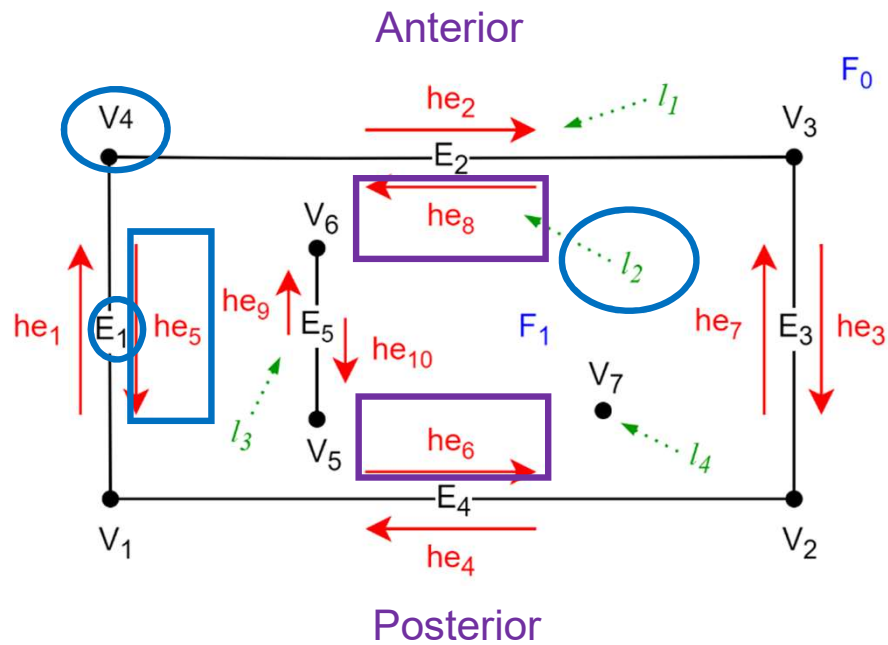
Semi-aresta	Aresta	Vértice	Loop	he_next	he_prev
$he_1$	$E_1$	$V_1$	$l_1$	$he_2$	$he_4$
$he_2$	$E_2$	$V_4$	$l_1$	$he_3$	$he_1$
$he_3$	$E_3$	$V_3$	$l_1$	$he_4$	$he_2$
$he_4$	$E_4$	$V_2$	$l_1$	$he_1$	$he_3$
$he_5$	$E_1$	$V_4$	$l_2$	$he_6$	$he_8$
$he_6$	$E_4$	$V_1$	$l_2$	$he_7$	$he_5$
$he_7$	$E_3$	$V_2$	$l_2$	$he_8$	$he_6$
$he_8$	$E_2$	$V_3$	$l_2$	$he_5$	$he_7$
$he_9$	$E_5$	$V_5$	$l_3$	$he_{10}$	$he_{10}$
$he_{10}$	$E_5$	$V_6$	$l_3$	$he_9$	$he_9$

# Half-Edge



Semi-aresta	Aresta	Vértice	Loop	he_next	he_prev
$he_1$	$E_1$	$V_1$	$l_1$	$he_2$	$he_4$
$he_2$	$E_2$	$V_4$	$l_1$	$he_3$	$he_1$
$he_3$	$E_3$	$V_3$	$l_1$	$he_4$	$he_2$
$he_4$	$E_4$	$V_2$	$l_1$	$he_1$	$he_3$
$he_5$	$E_1$	$V_4$	$l_2$	$he_6$	$he_8$
$he_6$	$E_4$	$V_1$	$l_2$	$he_7$	$he_5$
$he_7$	$E_3$	$V_2$	$l_2$	$he_8$	$he_6$
$he_8$	$E_2$	$V_3$	$l_2$	$he_5$	$he_7$
$he_9$	$E_5$	$V_5$	$l_3$	$he_{10}$	$he_{10}$
$he_{10}$	$E_5$	$V_6$	$l_3$	$he_9$	$he_9$

# Half-Edge



Semi-aresta	Aresta	Vértice	Loop	he_next	he_prev
$he_1$	$E_1$	$V_1$	$l_1$	$he_2$	$he_4$
$he_2$	$E_2$	$V_4$	$l_1$	$he_3$	$he_1$
$he_3$	$E_3$	$V_3$	$l_1$	$he_4$	$he_2$
$he_4$	$E_4$	$V_2$	$l_1$	$he_1$	$he_3$
$he_5$	$E_1$	$V_4$	$l_2$	$he_6$	$he_8$
$he_6$	$E_4$	$V_1$	$l_2$	$he_7$	$he_5$
$he_7$	$E_3$	$V_2$	$l_2$	$he_8$	$he_6$
$he_8$	$E_2$	$V_3$	$l_2$	$he_5$	$he_7$
$he_9$	$E_5$	$V_5$	$l_3$	$he_{10}$	$he_{10}$
$he_{10}$	$E_5$	$V_6$	$l_3$	$he_9$	$he_9$

# Operadores de Euler

- A estrutura de dados Half-Edge utiliza operadores que modificam as informações topológicas da estrutura de dados
- Criam superfícies fechadas e modificam as superfícies adicionando e excluindo faces, arestas e vértices
- Equação de Euler:

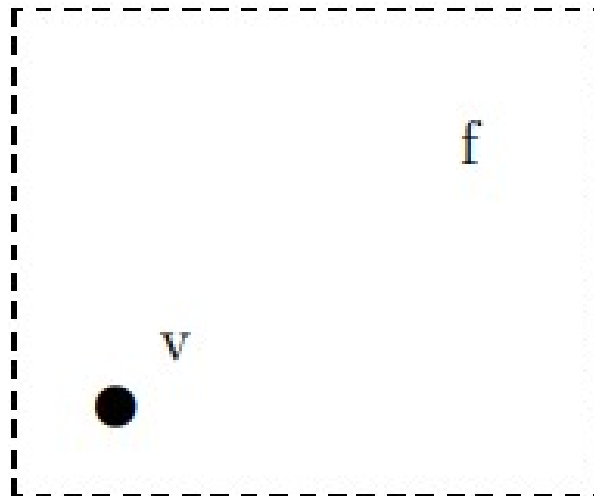
$$V - E + 2F - L - 2 = 0$$

# Operadores de Euler

Operadores	Significado	S	F	L	E	V	HE	Inverso
MVFS	make a vertex, a face, and a shell	+1	+1	+1		+1	+1	KVFS
MVR	make a vertex ring			+1		+1	+1	KVR
MEV	make an edge and a vertex				+1	+1	+2	KEV
MEF	make an edge and a face		+1		+1		+2	KEF
MEKR	make an edge kill a vertex			-1	+1		+2	KEMR
MVSE	make a vertex split an edge				+1	+1	+2	KVJE

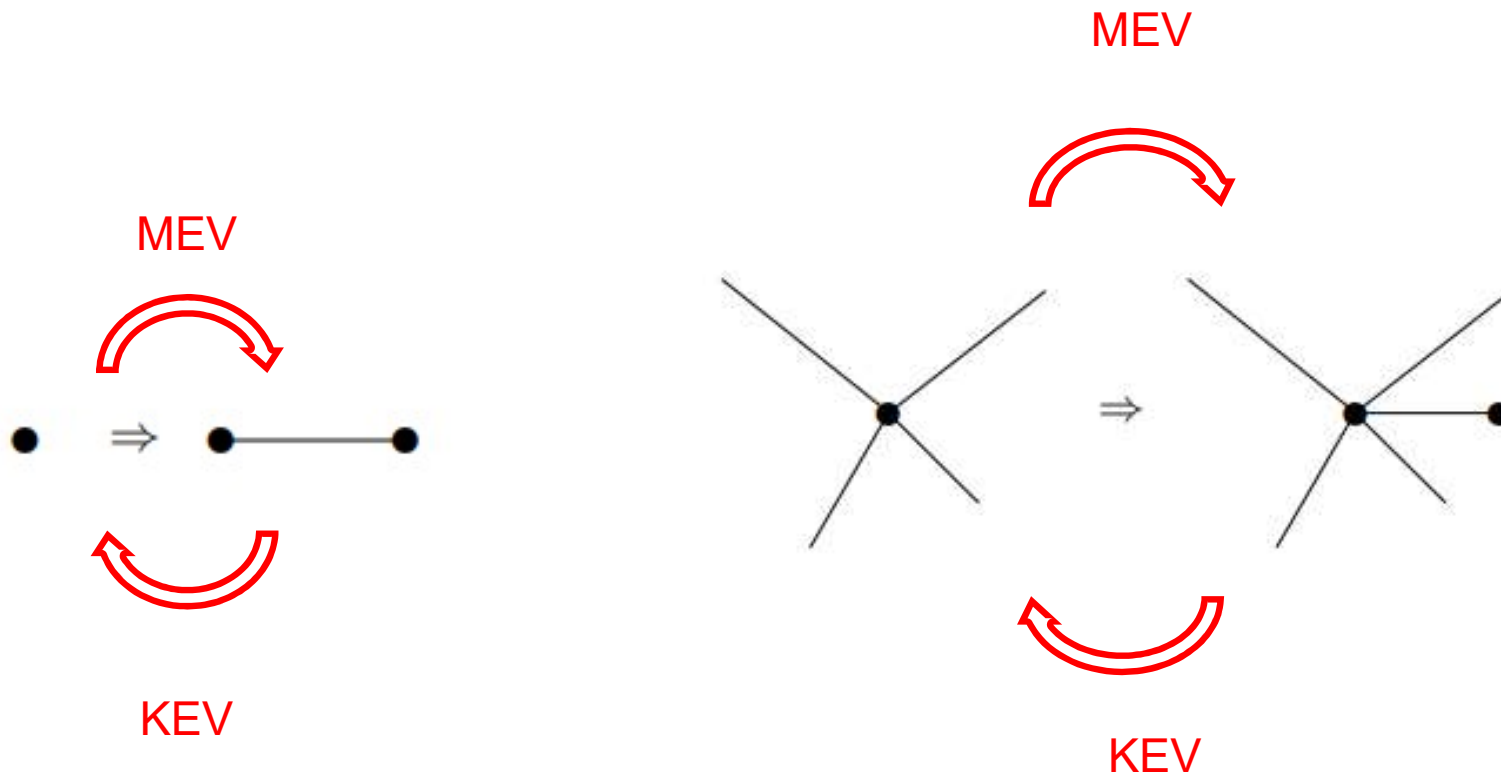
# Operadores MVFS e KVFS

- Cria um vértice, uma face e um sólido



# Operadores MEV e KEV

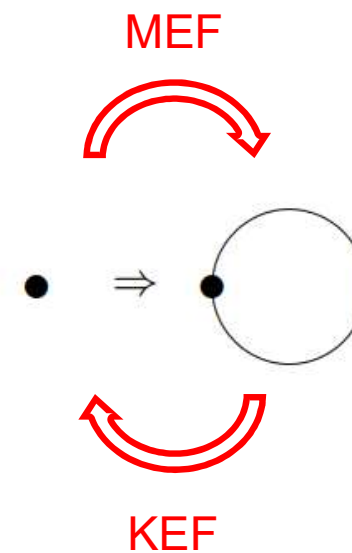
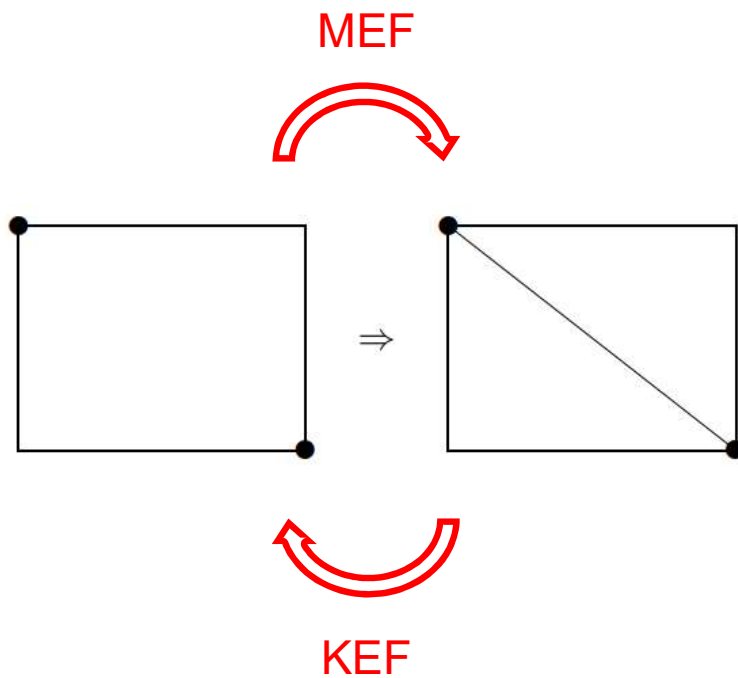
- Cria um vértice e uma aresta





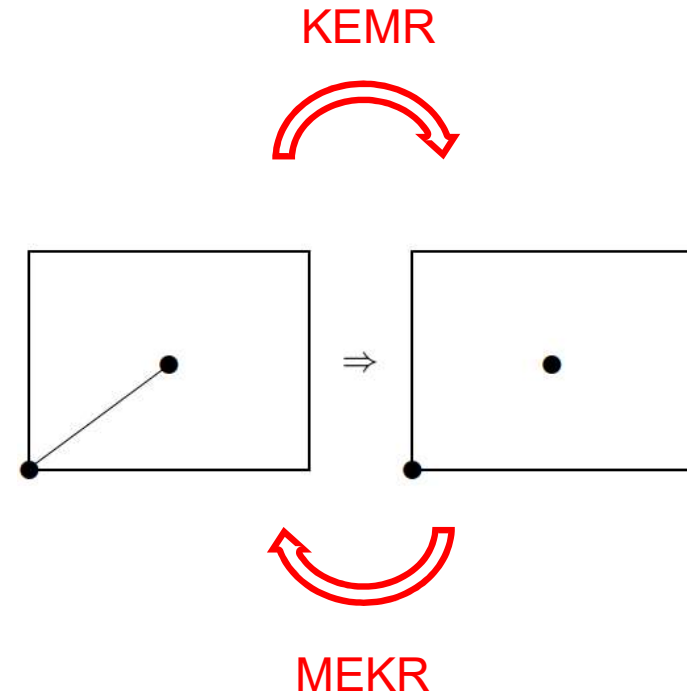
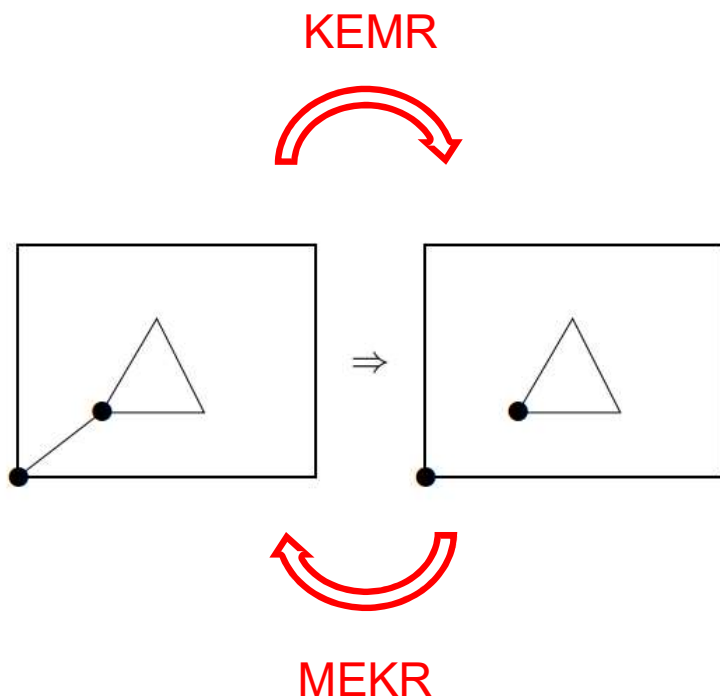
# Operadores MEF e KEF

- Cria uma aresta e uma face



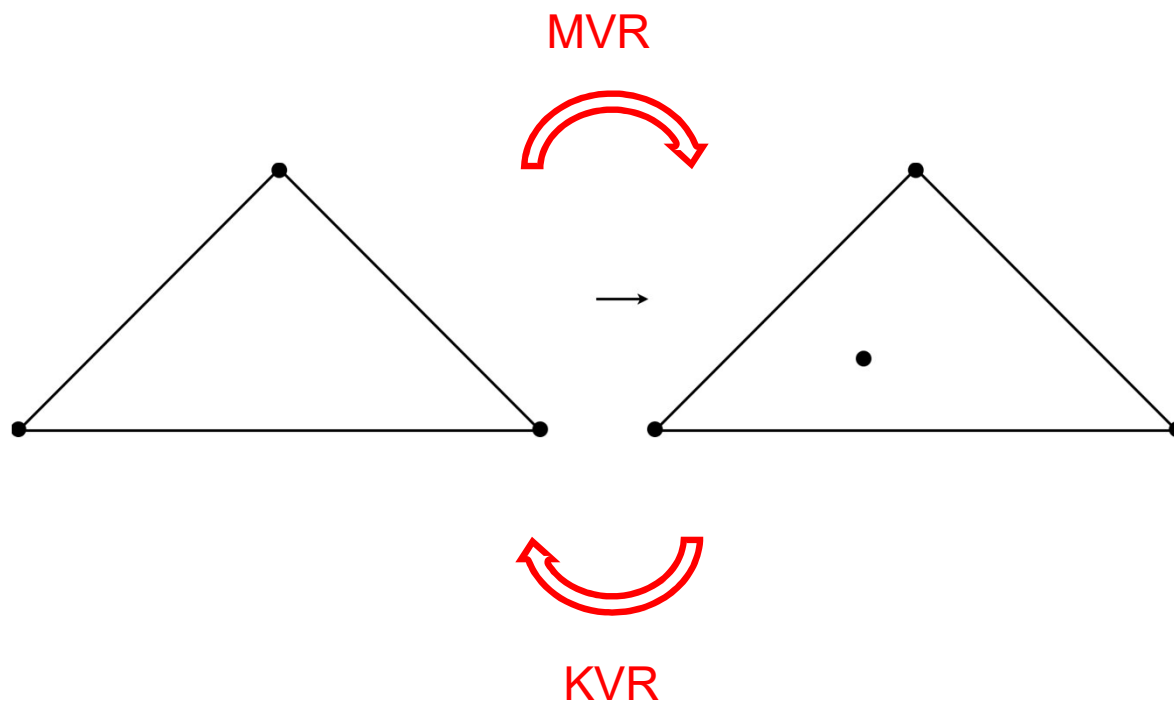
# Operadores KEMR e MEKR

- Elimina uma aresta e cria um loop



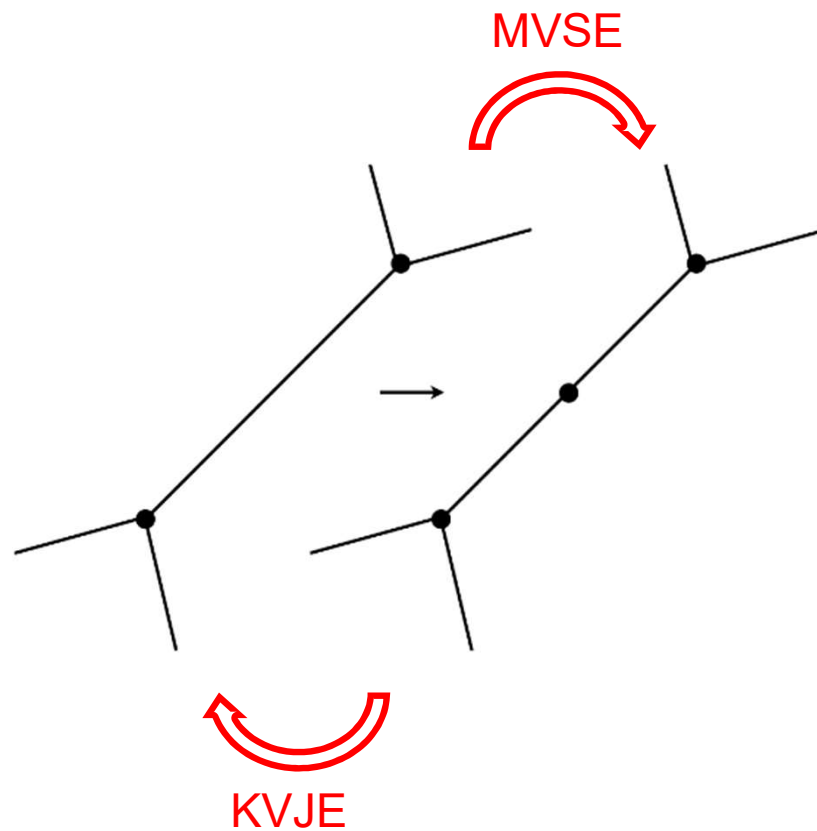
# Operadores MVR e KVR

- Cria um vértice e um novo loop



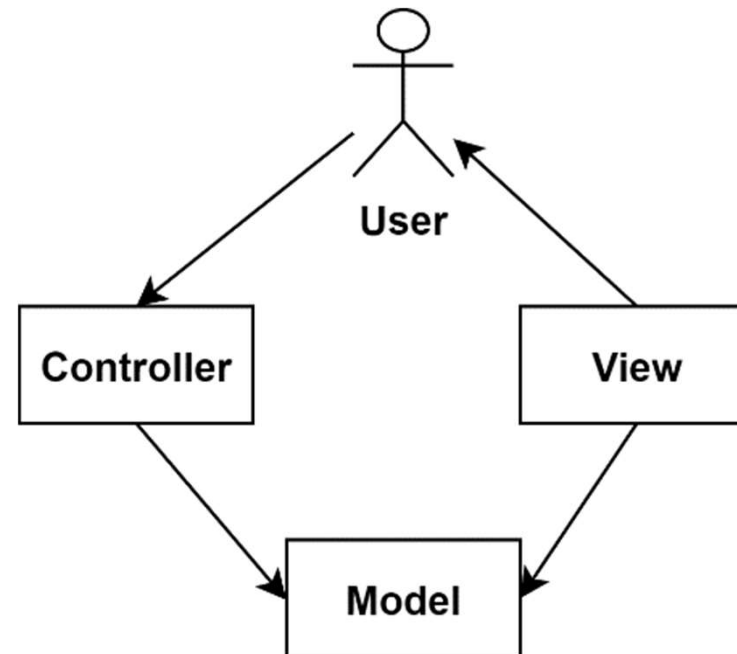
# Operadores MVSE e KVJE

- Cria uma vértice e divide uma aresta

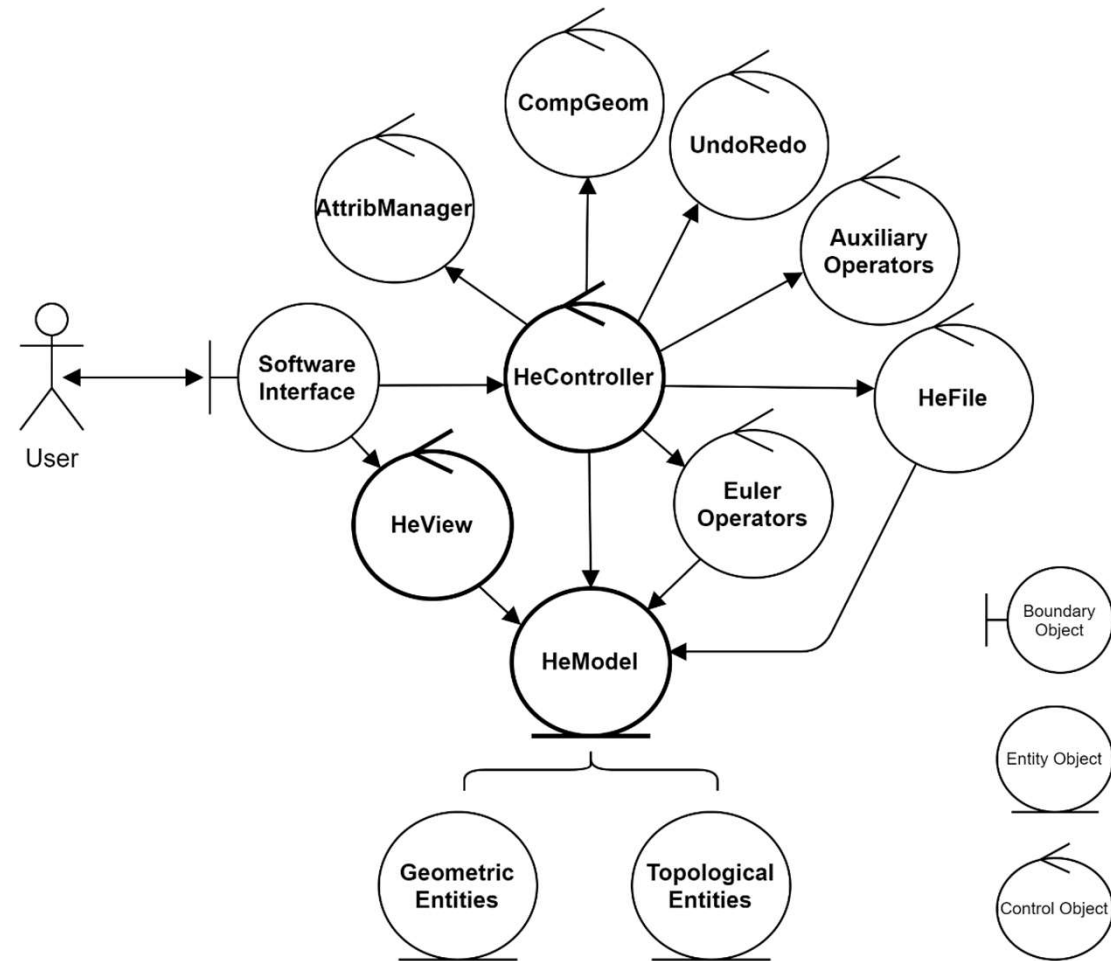


# Biblioteca HETOOL

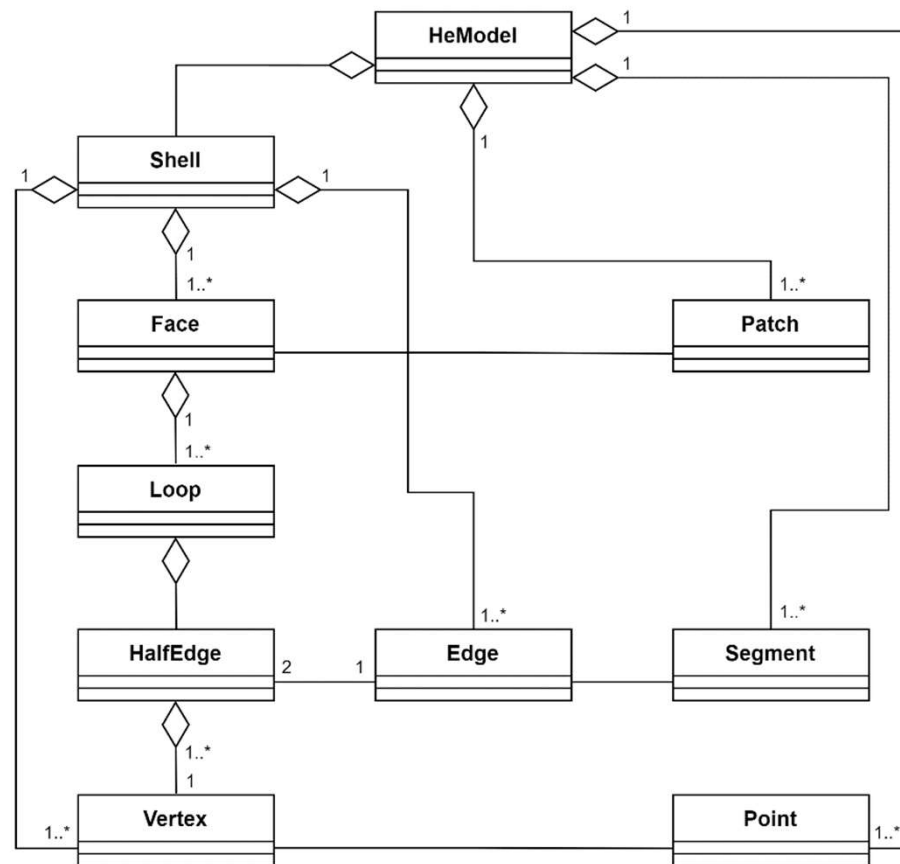
- Subdivisões planares
- Half-Edge
- Python
- Orientada a objetos
- Padrão MVC



# Diagrama de robustez da biblioteca HETOOL

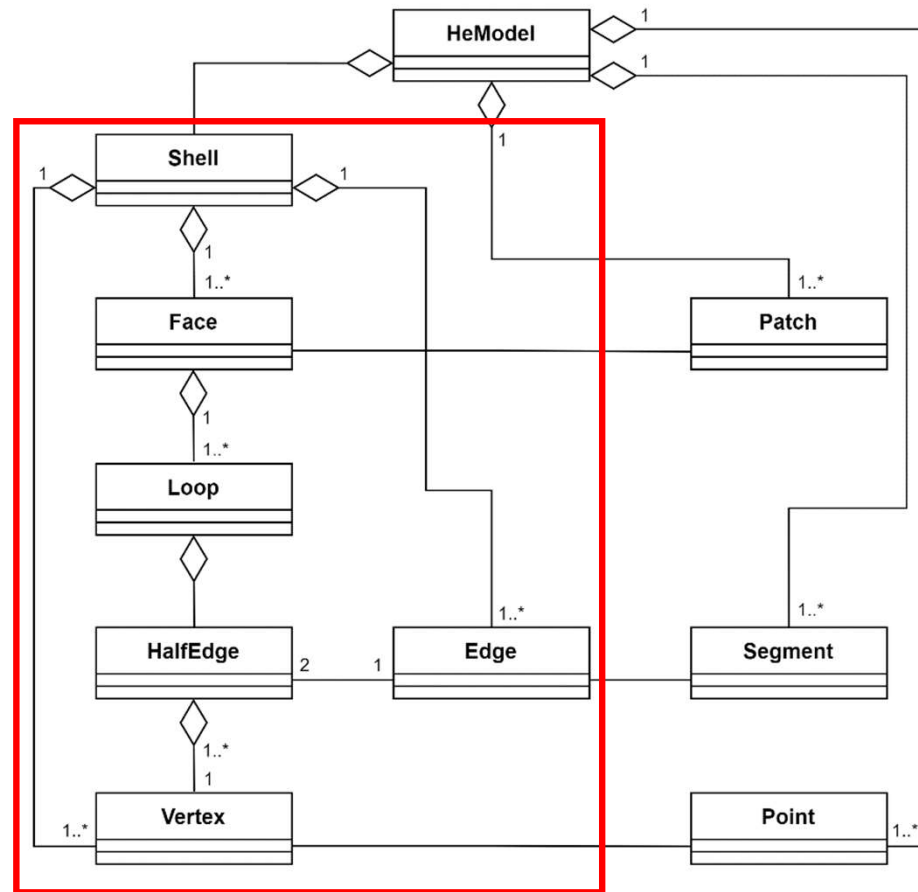


# Classe HeModel



# Classe HeModel

Classes topológicas

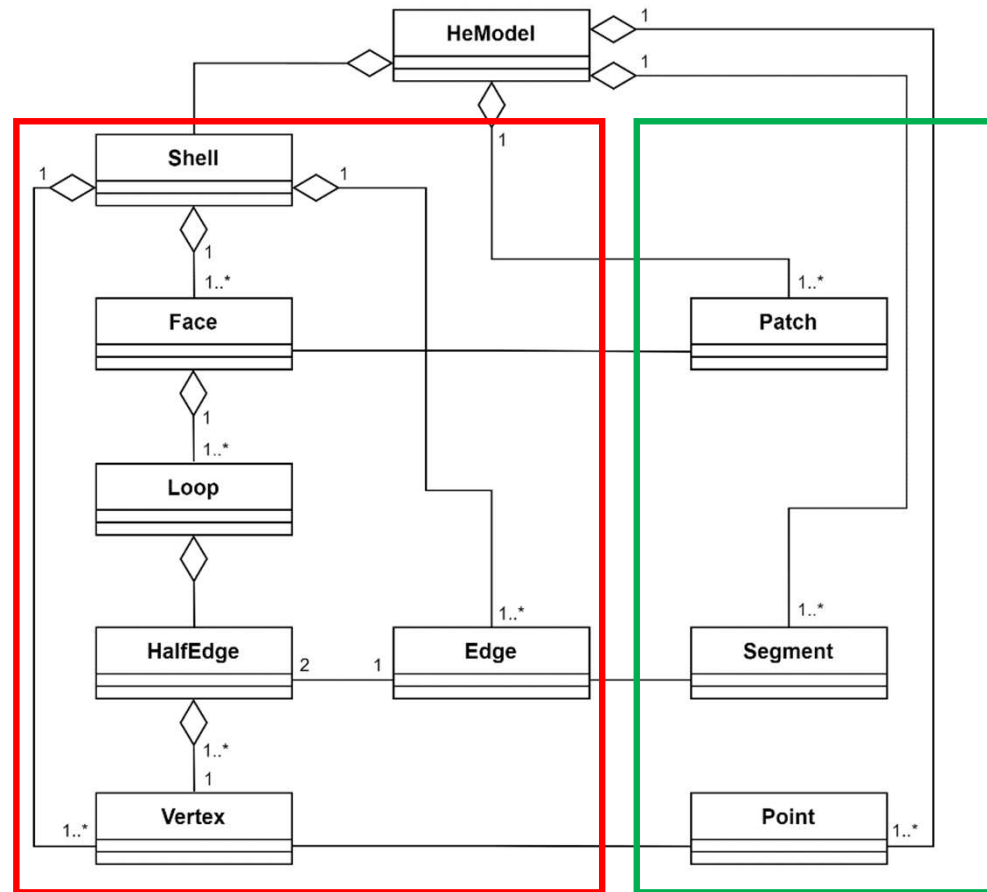




# Classe HeModel

Classes topológicas

Classes Geométricas

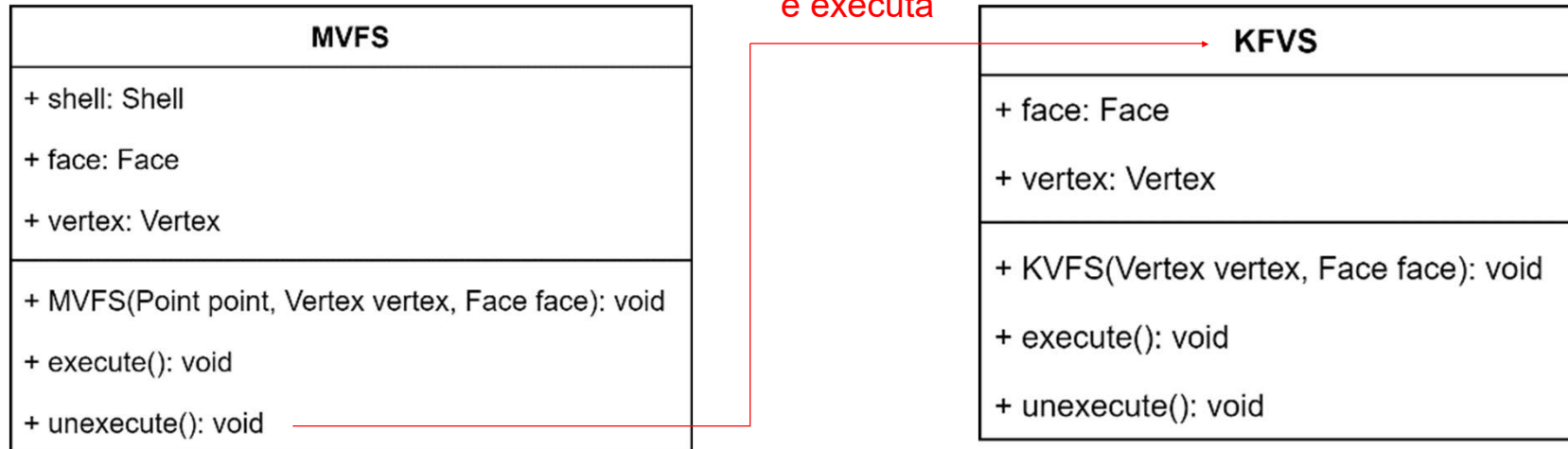


# Operadores de Euler

MVFS
+ shell: Shell
+ face: Face
+ vertex: Vertex
+ MVFS(Point point, Vertex vertex, Face face): void
+ execute(): void
+ unexecute(): void

KFVS
+ face: Face
+ vertex: Vertex
+ KVFS(Vertex vertex, Face face): void
+ execute(): void
+ unexecute(): void

# Padrão dos operadores de Euler



# Operadores Auxiliares

- Apresentam o mesmo padrão dos operadores de Euler
- Não realizam modificações topológicas
- Cumprem outras funções específicas na estrutura de dados:
  - *SetMesh* e *DelMesh*
  - *SetNumberOfSubdivisions*
  - *SetAttribute* e *UnSetAttribute*

# Classe UndoRedo

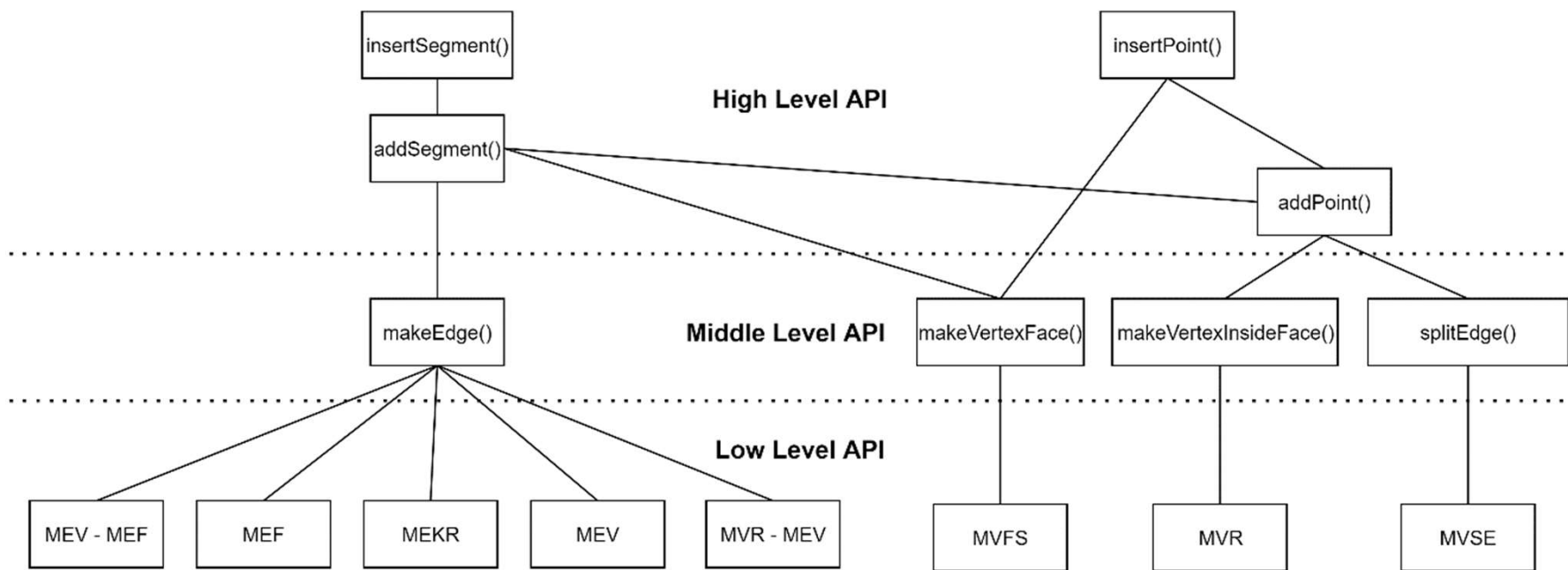
- Armazena e gerencia uma quantidade especifica de comandos de modelagem feitos pelo usuário
- Cada comando é representado por um conjunto de operações (Operadores de Euler e Operadores auxiliares)
- Realiza o gerenciamento de duas listas (comandos que podem ser desfeitos e comandos que podem ser refeitos)
  - Desfazer -> Unexecute()
  - Refazer - > Execute()

# Classe HeController

- Gerencia e processa as interações do usuário com a estrutura de dados
- Tem acesso a maioria dos módulos presentes na biblioteca
- Métodos de alto nível

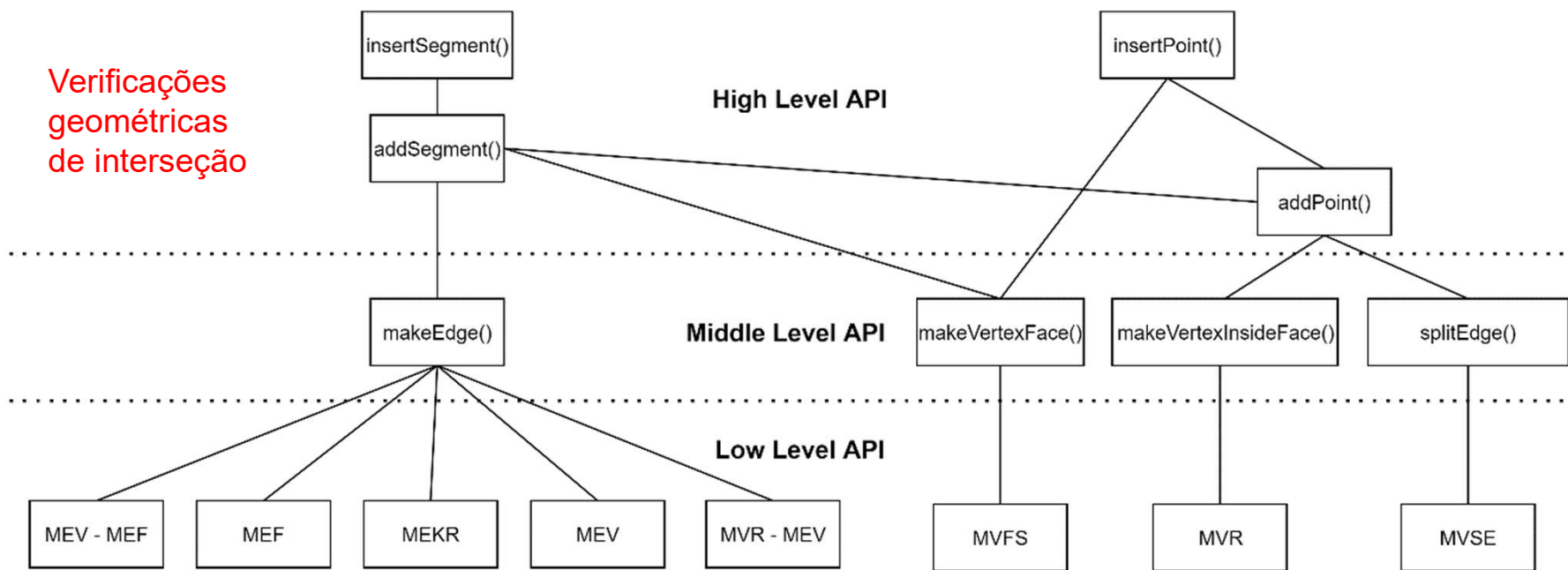
# Classe HeController

- API - *Application Programming Interface*



# Classe HeController

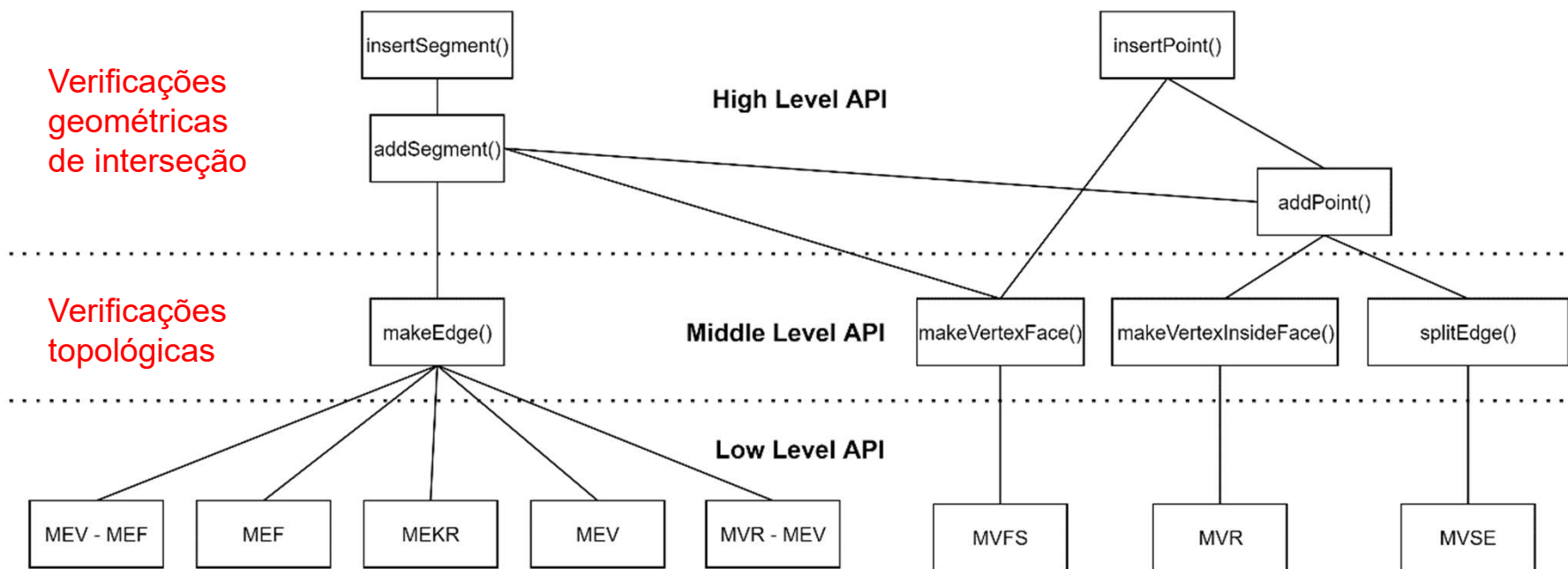
## ■ API - *Application Programming Interface*





# Classe HeController

## ■ API - *Application Programming Interface*



# Método delSelectedEntities

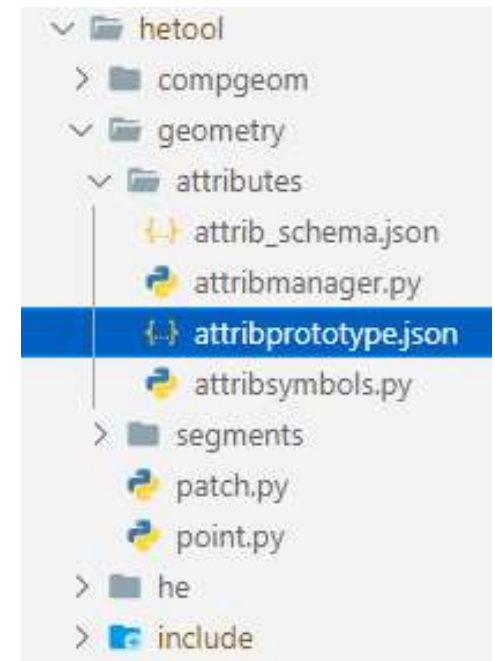
- Responsável pela remoção das entidades topológicas e geométricas do modelo que foram previamente selecionadas
- A função coleta do modelo todas as entidades selecionadas e em seguida elimina estes elementos em uma ordem predefinida
- É possível remover o visual de uma face para formar um buraco

# Classe AttribManager

- Responsável pelo gerenciamento dos atributos de modelagem
- Gerencia duas listas (protótipos de atributos e atributos)
- Protótipo de atributo
- Dicionário (dict):
  - Chave : valor

# Classe AttribManager

- Protótipos de atributos são criados a partir de um arquivo no formato JSON
- Esse arquivo apresenta uma estrutura similar ao dicionário em Python
- Objetos estruturados por:
  - chave : valor



# Classe AttribManager

```
{
  "type": "General Model",
  "symbol": "test",
  "name": "untitled",
  "properties": {
    "int": 0,
    "float": 0.0,
    "string": "Danilo Bomfim",
    "bool": true,
    "options": {
      "list": [
        "a",
        "b",
        "c",
        "d"
      ],
      "index": 0
    },
    "Color": [
      0,
      0,
      0
    ]
  ],
  "applyOnVertex": true,
  "applyOnEdge": true,
  "applyOnFace": true
}
```

# Classe AttribManager

Tipo do protótipo  
de atributo

```
{  
  "type": "General Model",  
  "symbol": "test",  
  "name": "untitled",  
  "properties": {  
    "int": 0,  
    "float": 0.0,  
    "string": "Danilo Bomfim",  
    "bool": true,  
    "options": {  
      "list": [  
        "a",  
        "b",  
        "c",  
        "d"  
      ],  
      "index": 0  
    },  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": true,  
  "applyOnEdge": true,  
  "applyOnFace": true  
}
```

# Classe AttribManager

Símbolo do atributo

```
{  
  "type": "General Model",  
  "symbol": "test",  
  "name": "untitled",  
  "properties": {  
    "int": 0,  
    "float": 0.0,  
    "string": "Danilo Bomfim",  
    "bool": true,  
    "options": {  
      "list": [  
        "a",  
        "b",  
        "c",  
        "d"  
      ],  
      "index": 0  
    },  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": true,  
  "applyOnEdge": true,  
  "applyOnFace": true  
}
```

# Classe AttribManager

Nome do atributo  
a ser criado pelo usuário

```
{
  "type": "General Model",
  "symbol": "test",
  "name": "untitled",
  properties: {
    "int": 0,
    "float": 0.0,
    "string": "Danilo Bomfim",
    "bool": true,
    "options": {
      "list": [
        "a",
        "b",
        "c",
        "d"
      ],
      "index": 0
    },
    "Color": [
      0,
      0,
      0
    ]
  },
  "applyOnVertex": true,
  "applyOnEdge": true,
  "applyOnFace": true
}
```



# Classe AttribManager

Propriedades dos atributos

```
{  
  "type": "General Model",  
  "symbol": "test",  
  "name": "untitled",  
  "properties": {  
    "int": 0,  
    "float": 0.0,  
    "string": "Danilo Bomfim",  
    "bool": true,  
    "options": {  
      "list": [  
        "a",  
        "b",  
        "c",  
        "d"  
      ],  
      "index": 0  
    },  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": true,  
  "applyOnEdge": true,  
  "applyOnFace": true  
}
```

# Classe AttribManager

Propriedades dos atributos

```
{  
  "type": "General Model",  
  "symbol": "test",  
  "name": "untitled",  
  "properties": {  
    "int": 0,  
    "float": 0.0,  
    "string": "Danilo Bomfim",  
    "bool": true,  
    "options": {  
      "list": [  
        "a",  
        "b",  
        "c",  
        "d"  
      ],  
      "index": 0  
    },  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": true,  
  "applyOnEdge": true,  
  "applyOnFace": true  
}
```

Número inteiro

# Classe AttribManager

Propriedades dos atributos

```
{  
  "type": "General Model",  
  "symbol": "test",  
  "name": "untitled",  
  "properties": {  
    "int": 0,  
    "float": 0.0,  
    "string": "Danilo Bomfim",  
    "bool": true,  
    "options": {  
      "list": [  
        "a",  
        "b",  
        "c",  
        "d"  
      ],  
      "index": 0  
    },  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": true,  
  "applyOnEdge": true,  
  "applyOnFace": true  
}
```

Número real

# Classe AttribManager

Propriedades dos atributos

```
{  
  "type": "General Model",  
  "symbol": "test",  
  "name": "untitled",  
  "properties": {  
    "int": 0,  
    "float": 0.0,  
    "string": "Danilo Bomfim",  
    "bool": true,  
    "options": {  
      "list": [  
        "a",  
        "b",  
        "c",  
        "d"  
      ],  
      "index": 0  
    },  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": true,  
  "applyOnEdge": true,  
  "applyOnFace": true  
}
```

Palavra

# Classe AttribManager

Propriedades dos atributos

```
{  
  "type": "General Model",  
  "symbol": "test",  
  "name": "untitled",  
  "properties": {  
    "int": 0,  
    "float": 0.0,  
    "string": "Danilo Bomfim",  
    "bool": true,  
    "options": {  
      "list": [  
        "a",  
        "b",  
        "c",  
        "d"  
      ],  
      "index": 0  
    },  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": true,  
  "applyOnEdge": true,  
  "applyOnFace": true  
}
```

Indicador de verdadeiro ou falso

# Classe AttribManager

Propriedades dos atributos

```
{  
  "type": "General Model",  
  "symbol": "test",  
  "name": "untitled",  
  "properties": {  
    "int": 0,  
    "float": 0.0,  
    "string": "Danilo Bomfim",  
    "bool": true,  
    "options": {  
      "list": [  
        "a",  
        "b",  
        "c",  
        "d"  
      ],  
      "index": 0  
    },  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": true,  
  "applyOnEdge": true,  
  "applyOnFace": true  
}
```

Lista de opções que limita a escolha do usuário

# Classe AttribManager

Propriedades dos atributos

```
{  
  "type": "General Model",  
  "symbol": "test",  
  "name": "untitled",  
  "properties": {  
    "int": 0,  
    "float": 0.0,  
    "string": "Danilo Bomfim",  
    "bool": true,  
    "options": {  
      "list": [  
        "a",  
        "b",  
        "c",  
        "d"  
      ],  
      "index": 0  
    },  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": true,  
  "applyOnEdge": true,  
  "applyOnFace": true  
}
```

Cor do atributo segundo padrão RGB

# Classe AttribManager

Indicadores que determinam  
se é possível aplicar o atributo  
em um vértice, uma aresta ou uma face

```
{  
  "type": "General Model",  
  "symbol": "test",  
  "name": "untitled",  
  "properties": {  
    "int": 0,  
    "float": 0.0,  
    "string": "Danilo Bomfim",  
    "bool": true,  
    "options": {  
      "list": [  
        "a",  
        "b",  
        "c",  
        "d"  
      ],  
      "index": 0  
    },  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": true,  
  "applyOnEdge": true,  
  "applyOnFace": true  
}
```



# Exemplos de protótipos de atributos

```
{
  "type": "Material",
  "symbol": "Material",
  "name": "untitled",
  "properties": {
    "YoungsModulus": 0.0,
    "PoisonsRatio": 0.0,
    "Color": [
      0,
      0,
      0
    ]
  },
  "applyOnVertex": false,
  "applyOnEdge": false,
  "applyOnFace": true
},
```

```
{
  "type": "Concentrated Load",
  "symbol": "Arrow",
  "name": "untitled",
  "properties": {
    "Fx": 0.0,
    "Fy": 0.0,
    "Mz": 0.0,
    "Color": [
      0,
      0,
      0
    ]
  },
  "applyOnVertex": true,
  "applyOnEdge": false,
  "applyOnFace": false
},
```

# Classe HeFile

- Salvar, ler e exportar
- JSON
- Identificador numérico (ID)

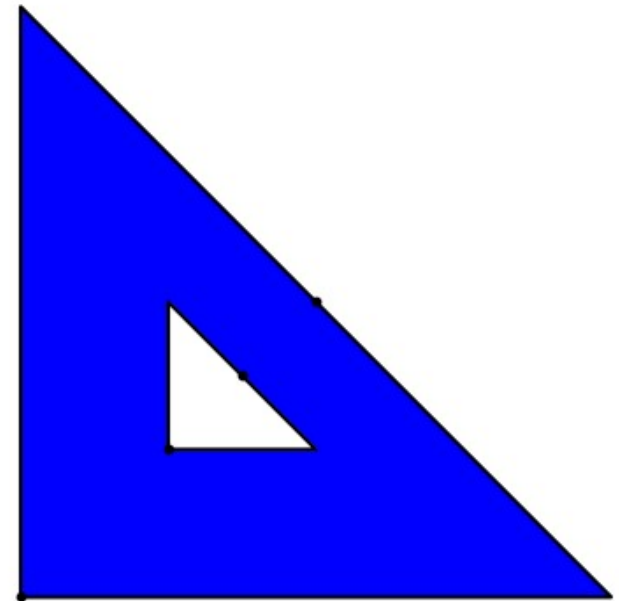
# Uso da biblioteca HETOOL

- Criação do modelo geométrico
- Criação e configuração de atributos de modelagem
- Aplicativos desenvolvidos

# Exemplos de uso

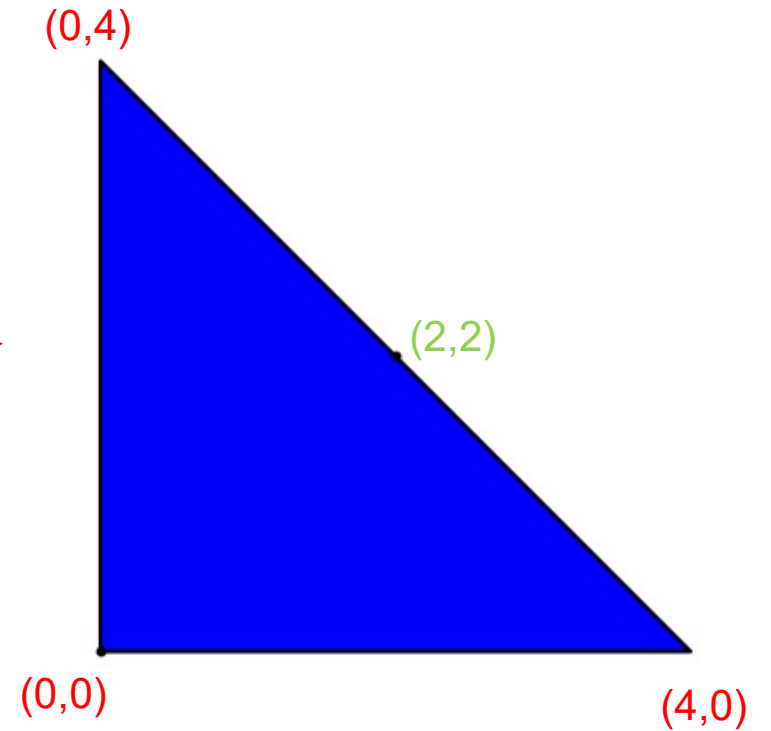
```
from hetool.include.hetool import Hetool  
Hetool.insertSegment([0,0,4,0,0,4,0,0])  
Hetool.insertSegment([1,1,2,1,1,2,1,1])  
Hetool.selectPick(1.1,1.1,0.01)  
Hetool.delSelectedEntities()
```

matplotlib



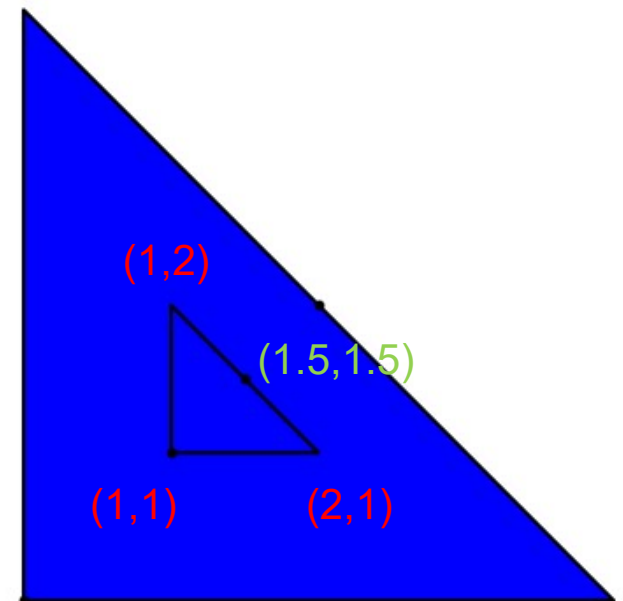
# Exemplos de uso

```
from hetool.include.hetool import Hetool  
Hetool.insertSegment([0,0,4,0,0,4,0,0])  
Hetool.insertSegment([1,1,2,1,1,2,1,1])  
Hetool.selectPick(1.1,1.1,0.01)  
Hetool.delSelectedEntities()
```



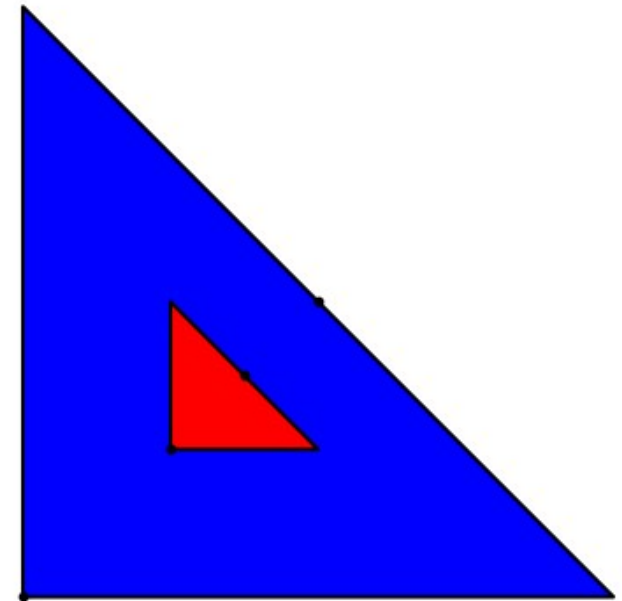
# Exemplos de uso

```
from hetool.include.hetool import Hetool
Hetool.insertSegment([0,0,4,0,0,4,0,0])
Hetool.insertSegment([1,1,2,1,1,2,1,1])
Hetool.selectPick(1.1,1.1,0.01)
Hetool.delSelectedEntities()
```



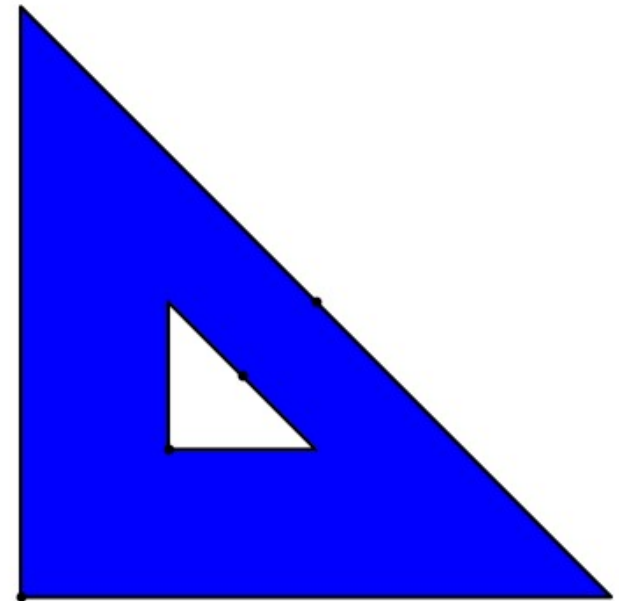
# Exemplos de uso

```
from hetool.include.hetool import Hetool
Hetool.insertSegment([0,0,4,0,0,4,0,0])
Hetool.insertSegment([1,1,2,1,1,2,1,1])
Hetool.selectPick(1.1,1.1,0.01)
Hetool.delSelectedEntities()
```



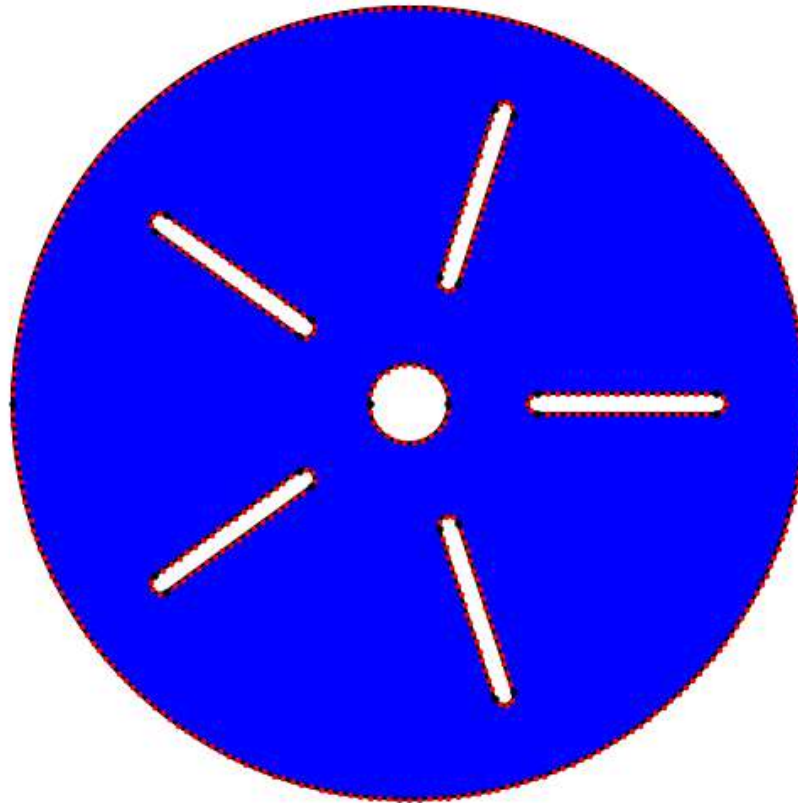
# Exemplos de uso

```
from hetool.include.hetool import Hetool
Hetool.insertSegment([0,0,4,0,0,4,0,0])
Hetool.insertSegment([1,1,2,1,1,2,1,1])
Hetool.selectPick(1.1,1.1,0.01)
Hetool.delSelectedEntities()
```





# Exemplos de uso



# Criação e configuração dos atributos

```
{  
  "type": "Material",  
  "symbol": "Material",  
  "name": "untitled",  
  "properties": {  
    "YoungsModulus": 0.0,  
    "PoissonsRatio": 0.0,  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": false,  
  "applyOnEdge": false,  
  "applyOnFace": true  
},
```

```
{  
  "type": "Concentrated Load",  
  "symbol": "Arrow",  
  "name": "untitled",  
  "properties": {  
    "Fx": 0.0,  
    "Fy": 0.0,  
    "Fz": 0.0,  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": true,  
  "applyOnEdge": false,  
  "applyOnFace": false  
},
```

Tipo do protótipo  
de atributo

Nome  
específico

```
Hetool.addAttribute("Material", "M1")  
Hetool.addAttribute("Material", "M2")  
Hetool.addAttribute("Support Conditions", "S1")  
Hetool.addAttribute("Concentrated Load", "CL1")  
Hetool.addAttribute("Uniform Load", "UL1")
```

Referência

Nome  
do atributo

```
material_1 = Hetool.getAttributeByName("M1")  
material_2 = Hetool.getAttributeByName("M2")  
support = Hetool.getAttributeByName("S1")  
concentratedLoad = Hetool.getAttributeByName("CL1")  
uniformLoad = Hetool.getAttributeByName("UL1")
```

# Criação e configuração dos atributos

```
{  
  "type": "Material",  
  "symbol": "Material",  
  "name": "untitled",  
  "properties": {  
    "YoungsModulus": 0.0,  
    "PoisonsRatio": 0.0,  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": false,  
  "applyOnEdge": false,  
  "applyOnFace": true  
},
```

```
material_1['properties']['YoungsModulus'] = 100000  
material_1['properties']['PoisonsRatio'] = 0.3
```

```
material_2['properties']['YoungsModulus'] = 500000  
material_2['properties']['PoisonsRatio'] = 0.25
```

```
concestratedLoad['properties']['Fy'] = -10.0
```

```
concestratedLoad['properties']['Mz'] = -2.0
```

```
{  
  "type": "Concentrated Load",  
  "symbol": "Arrow",  
  "name": "untitled",  
  "properties": {  
    "Fx": 0.0,  
    "Fy": 0.0,  
    "Mz": 0.0,  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": true,  
  "applyOnEdge": false,  
  "applyOnFace": false  
},
```

```
Hetool.selectPick(1.0, 0.25, 0.01)
```

```
Hetool.setAttribute("M1")
```

```
Hetool.selectPick(3, 0.25, 0.01)
```

```
Hetool.setAttribute("M2")
```

# Criação e configuração dos atributos

```
{  
  "type": "Material",  
  "symbol": "Material",  
  "name": "untitled",  
  "properties": {  
    "YoungsModulus": 0.0,  
    "PoisonsRatio": 0.0,  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": false,  
  "applyOnEdge": false,  
  "applyOnFace": true  
},
```

```
material_1['properties']['YoungsModulus'] = 100000  
material_1['properties']['PoisonsRatio'] = 0.3  
material_2['properties']['YoungsModulus'] = 500000  
material_2['properties']['PoisonsRatio'] = 0.25
```

```
concentratedLoad['properties']['Fy'] = -10.0  
concentratedLoad['properties']['Mz'] = -2.0
```

```
{  
  "type": "Concentrated Load",  
  "symbol": "Arrow",  
  "name": "untitled",  
  "properties": {  
    "Fx": 0.0,  
    "Fy": 0.0,  
    "Mz": 0.0,  
    "Color": [  
      0,  
      0,  
      0  
    ]  
  },  
  "applyOnVertex": true,  
  "applyOnEdge": false,  
  "applyOnFace": false  
},
```

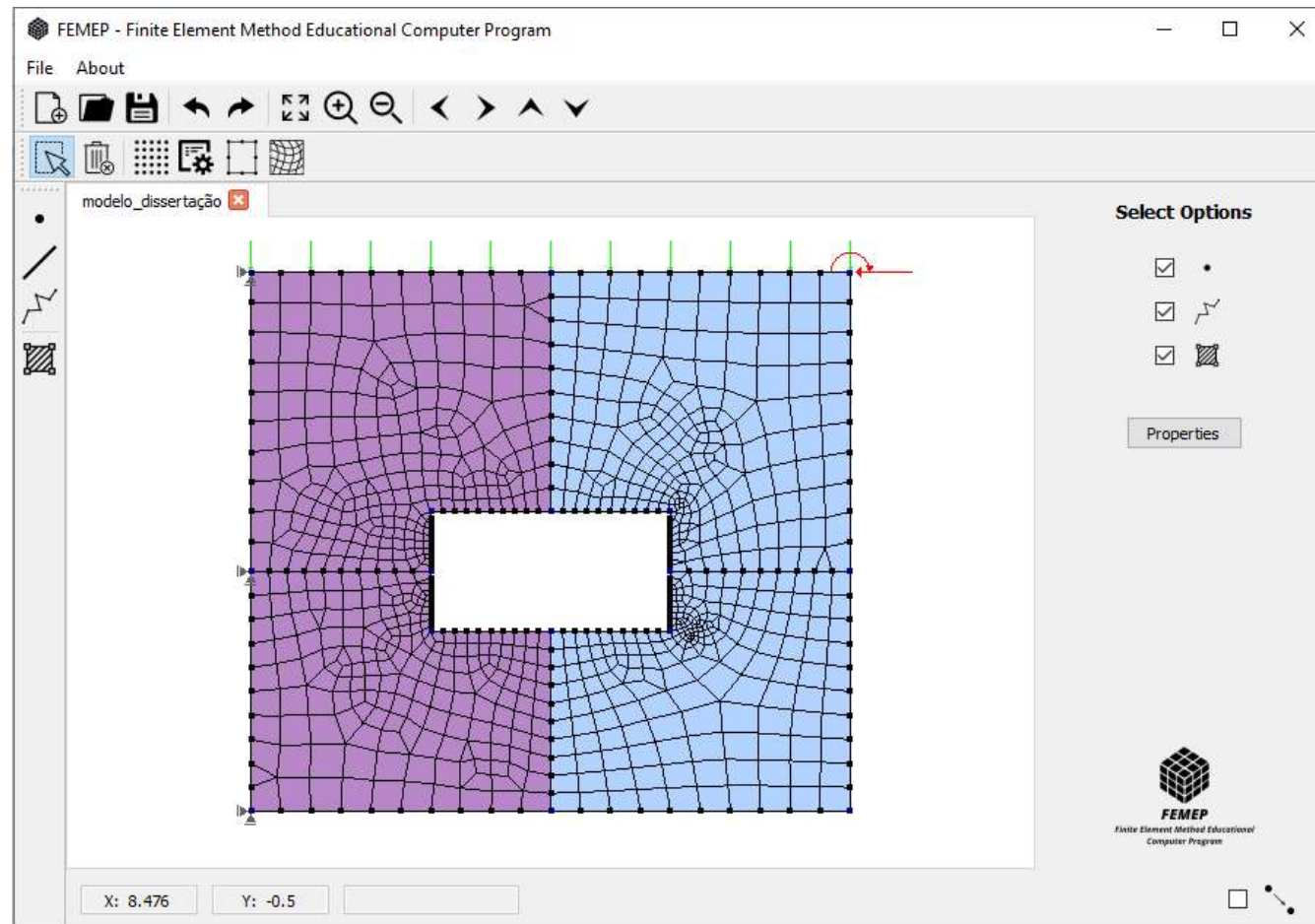
```
Hetool.selectPick(1.0, 0.25, 0.01)
```

```
Hetool.setAttribute("M1")
```

```
Hetool.selectPick(3, 0.25, 0.01)
```

```
Hetool.setAttribute("M2")
```

# Modelador de sólidos - FEMEP





Select Options

- 
- ↗
- ▨

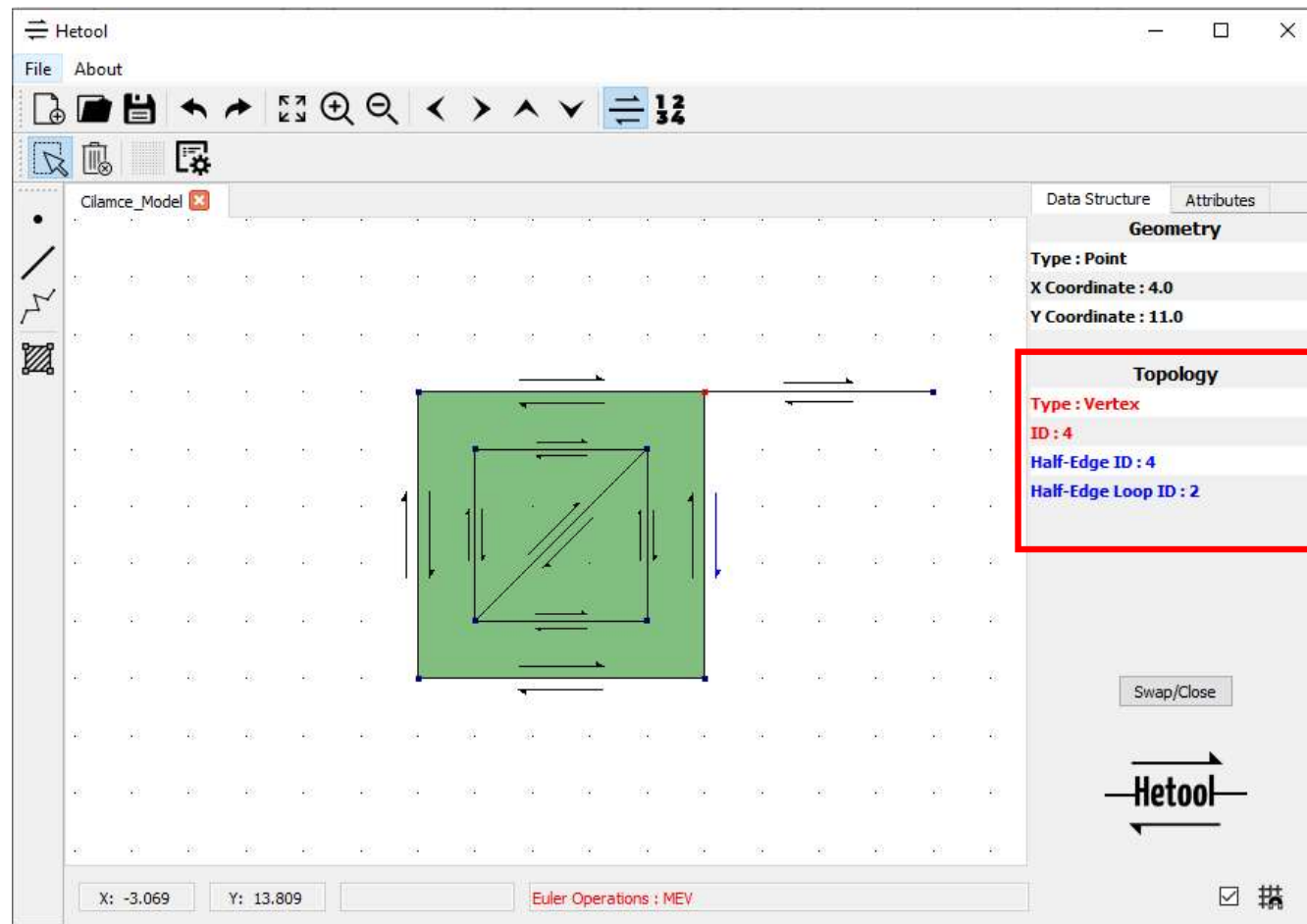
Properties



X: 20.773 Y: 3.676

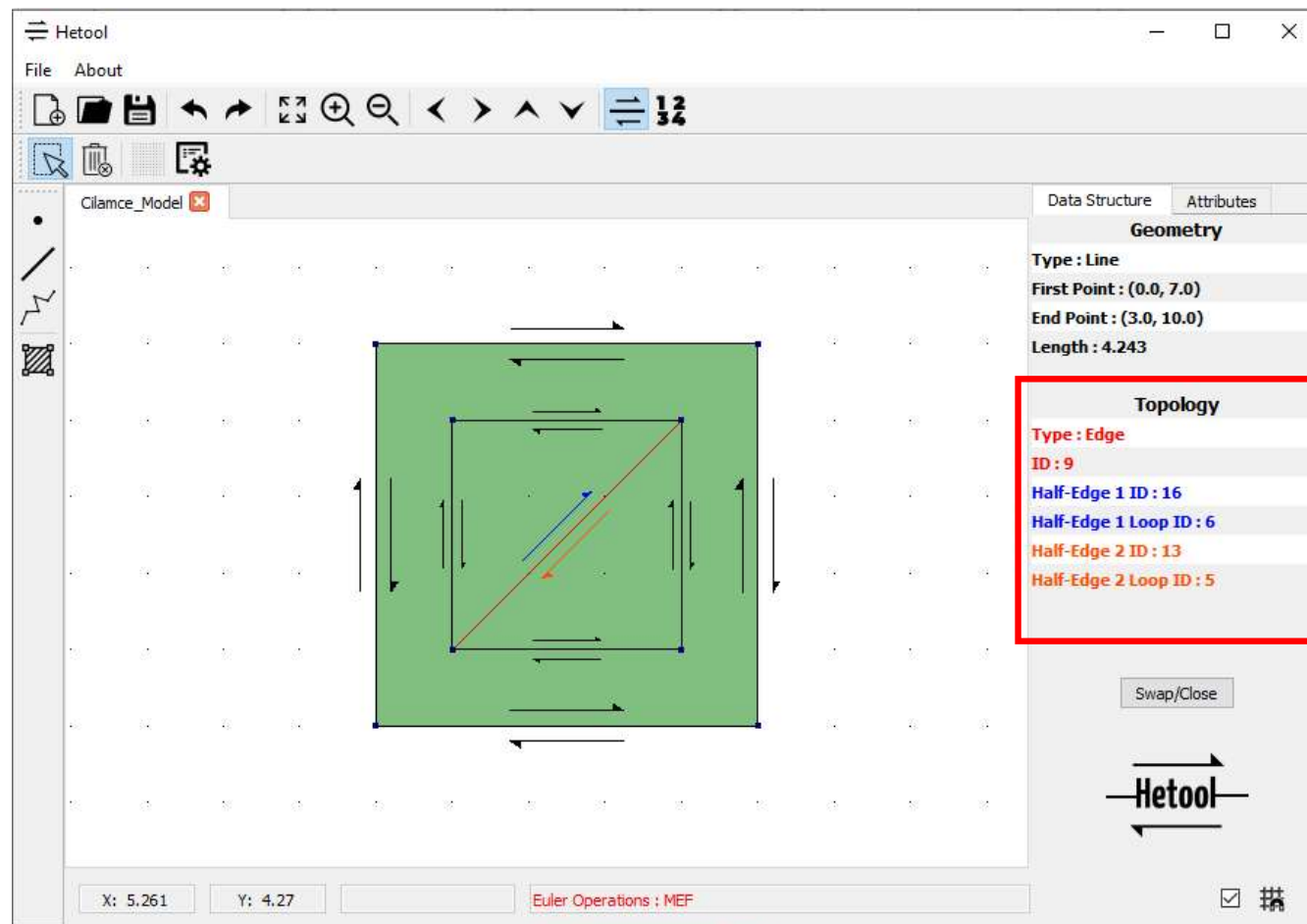


# Programa HETOOL



Informações topológicas

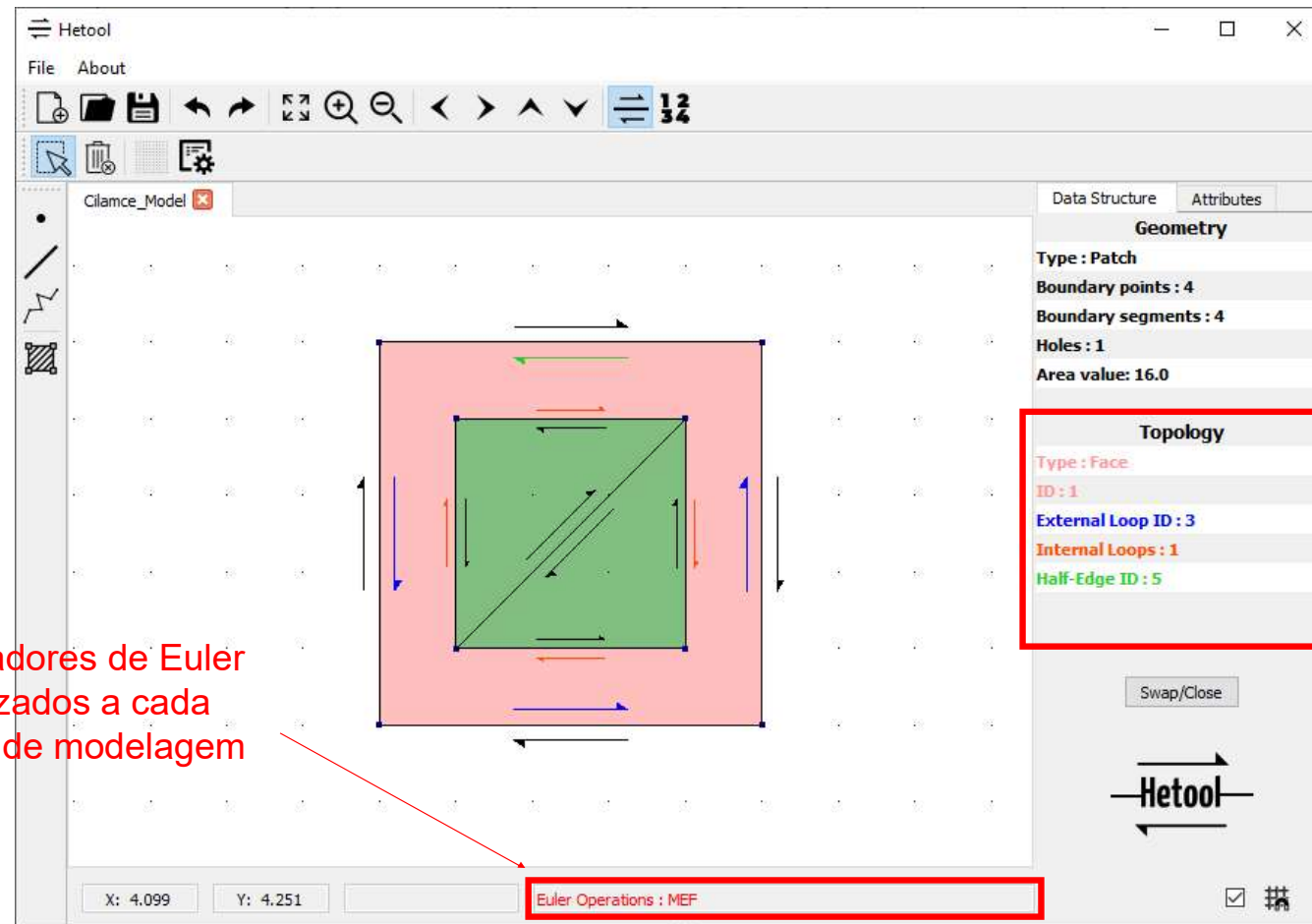
# Programa HETOOL



Informações topológicas



# Programa HETOOL



Operadores de Euler utilizados a cada ação de modelagem

Informações topológicas

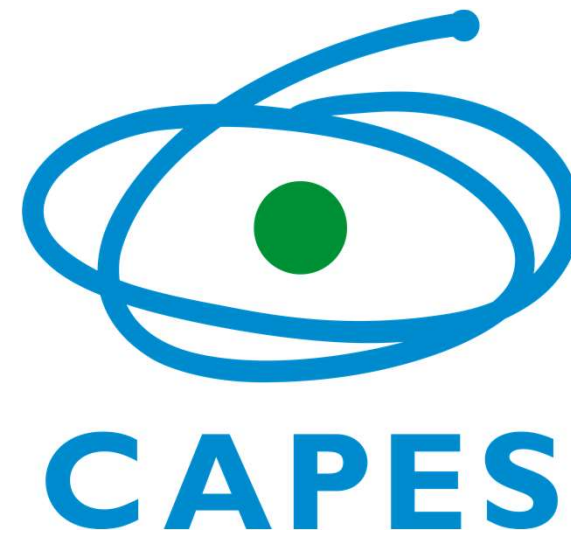
# Conclusões e principais contribuições

- Biblioteca HETOOL
- Estrutura de dados dinâmica e iterativa
- Simples e de fácil uso
- Gerenciamento genérico de atributos de modelagem

# Sugestões de trabalhos futuros

- Incrementos a biblioteca HETOOOL ou nos aplicativos
- Tratamento de outros tipos de curva com conversão para poligonais equivalentes adaptativo à curvatura
- Otimização dos processos
- Desenvolvimento de uma biblioteca para a criação de modelos de subdivisões espaciais tridimensionais
  - Radial-Edge (não-manifold)

# Agradecimentos



# Referências Bibliográficas

- BOGUSLAWSKI, P. **Modeling and Analysing 3D Building Interiors with The Dual Half-Edge Data Structure**. Doctor of Philosophy (PhD) - University of Glamorgan, 2011.
- BORTOLOSSI, Humberto J. **Notas de Aula de um Curso de Modelagem de Sólidos em Computação Gráfica**. Universidade Federal de Fluminense, 2017.
- CAVALCANTI, Paulo Roma. **Criação e Manutenção de subdivisões no espaço**. Orientador: Paulo Cezar Pinto Carvalho. Tese (Doutorado em Informática: Ciência da Computação) - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 1992.
- FILHO, Waldemar Celes. **Modelagem Configurável de Subdivisões Planares Hierárquicas**. Orientador: Marcelo Gattass. Tese (Doutorado em Informática: Ciência da Computação) - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 1995.
- HOFFMANN, C. M. **Geometric & Solid Modeling: An Introduction**. Purdue University, Indiana, 1989.
- MARTHA, L.F. **Análise Matricial de Estruturas com Orientação a Objetos, 1ª Edição**. Editora GEN LTC, ISBN (versão digital): 978-85-352-8798-1, 2018.
- MARTHA, L.F. et al. FE Adaptive Analysis of Multi-regions Models. **Proceedings of the VI International Conference on Adaptive Modeling and Visualization (ADMOS 2013)**, p. 456-467, Portugal, 2013.
- MÄNTYLÄ, M. **An Introduction to Solid Modeling Computer**. Science Press, Rockville, Maryland, 1988.
- WEILER, K. **Topological Structures for Geometric Modeling**. Doctor of Philosophy (PhD) - Rensselaer Polytechnic Institute, 1986.

Obrigado  
pela atenção!