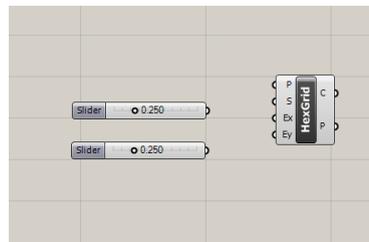


Exercício de implementação 1 | ATRATORES

Este exercício de implementação foi traduzido e adaptado tendo como base o tutorial *Attractors*, disponível na plataforma TOI-Pedia da Universidade Técnica de Delft (TU-Delft, Países Baixos). O tutorial original pode ser acessado em: <http://wiki.bk.tudelft.nl/toi-pedia/Attractors>. Explicações e comentários adicionais também se baseiam em outras fontes que estão listados no item 7.1. Sugestões de leitura.

Atratores na modelagem paramétrica podem ter múltiplas aplicações. Eles podem ser usados para a produção de padrões, mas talvez uma de suas aplicações mais interessantes seja em explorações de performance do *design*. Por exemplo, otimizando a insolação em uma fachada de acordo com o movimento do sol. Neste tutorial será realizado um exercício envolvendo uma curva de atração, essa curva define diferentes raios para círculos em um grid. Iremos agora implementar os passos desse algoritmo.

- 1) Criação do padrão hexagonal: crie um padrão hexagonal usando o componente correspondente e defina os tamanhos nos sentidos X e Y usando uma entrada numérica.
 - 1.1. Adicione um componente hexagonal na tela: **Vector > Grid > Hexagonal**;
 - 1.2. Defina os tamanhos X e Y com um Number Slider: **Params > Input > Number Slider**.

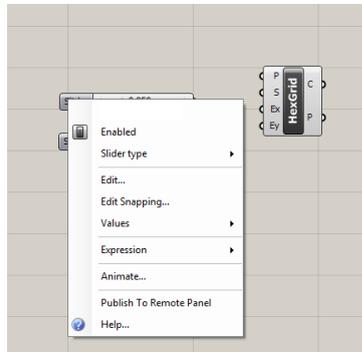


Neste momento, existem dois componentes Number Slider em tela. Um dos componentes será ligado ao parâmetro S do componente de grid hexagonal e o outro será conectado aos parâmetros Ex e Ey. O Parâmetro S (Size) é o parâmetro de tamanho do grid, ele define o tamanho do raio do hexágono e os parâmetros Ex (Extent X) e Ey (Extent Y) definem as quantidades de células do grid hexagonal nos sentidos X e Y, respectivamente.

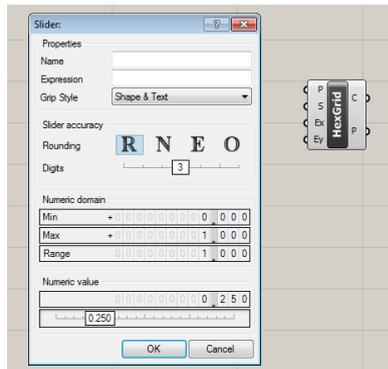
Antes de conectar cada um dos componentes Number Slider no componente de grid hexagonal, precisaremos configurá-los.

1.3. **Configuração do componente Number Slider** para conectá-lo aos parâmetros Ex e Ey.

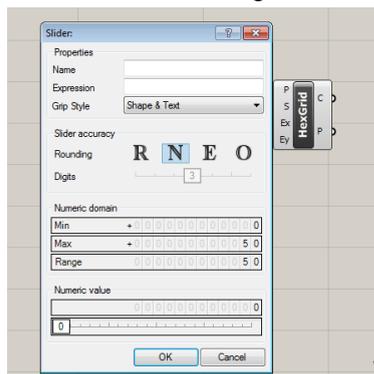
- a. Clique com o botão direito sobre a palavra **Slider**, abrirá o seguinte menu:



- b. Clique em **Edit** e será aberta uma janela para configurar o componente **Number Slider**.



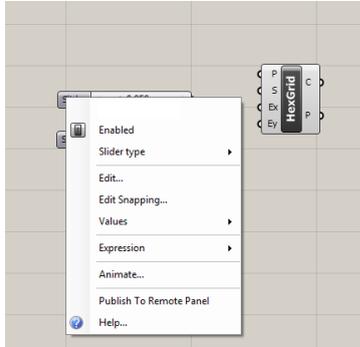
- c. Na janela aberta selecione a opção **N**, para que sejam usados números inteiros e digite **0** em **Min** e **50** em **Max**



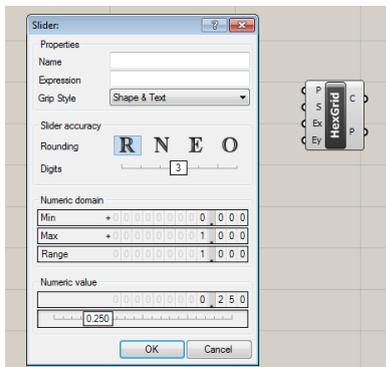
- d. Clique em **OK**

1.4. **Configuração do componente Number Slider** para conectá-lo ao parâmetro S.

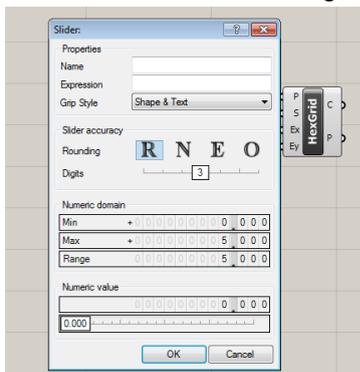
- a. Clique com o botão direito sobre a palavra **Slider**, abrirá o seguinte menu:



- b. Clique em **Edit** e será aberta uma janela para configurar o componente **Number Slider**.



- c. Na janela aberta mantenha selecionada a opção **R**, para que sejam usados números reais, e digite **0** em **Min** e **5** em **Max**



- d. Clique em **OK**

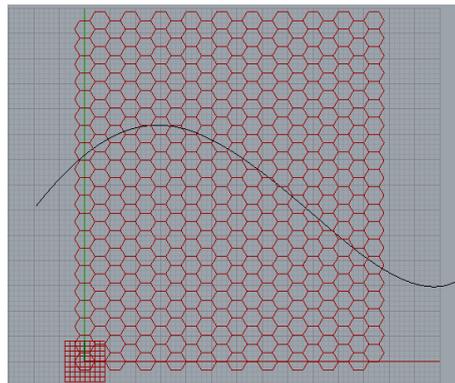
2. **Desenhar uma curva no grid hexagonal.** A curva pode cruzar os limites do grid, ela deve ser desenhada na interface do Rhinoceros e configurada no algoritmo do Grasshopper.

2.1. Desenhe, no Rhino, uma curva que ultrapasse os limites do grid hexagonal.

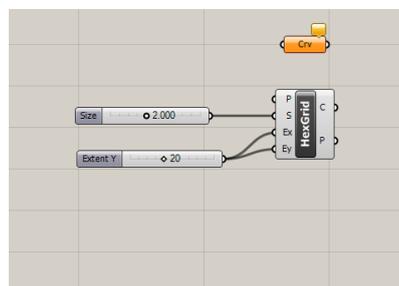
- a. Você pode desenhar a curva através da função Curva, no menu lateral esquerdo do Rhinoceros.



- b. Você tem a liberdade para criar a sua curva, o desenho final deve ficar parecido com a imagem a seguir.

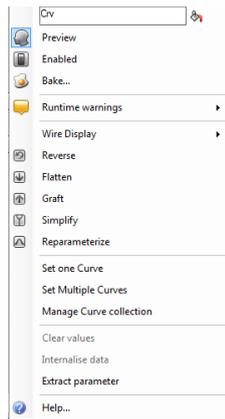


2.2. No Grasshopper, adicione um componente de curva indo em **Params > Geometry > Curve**

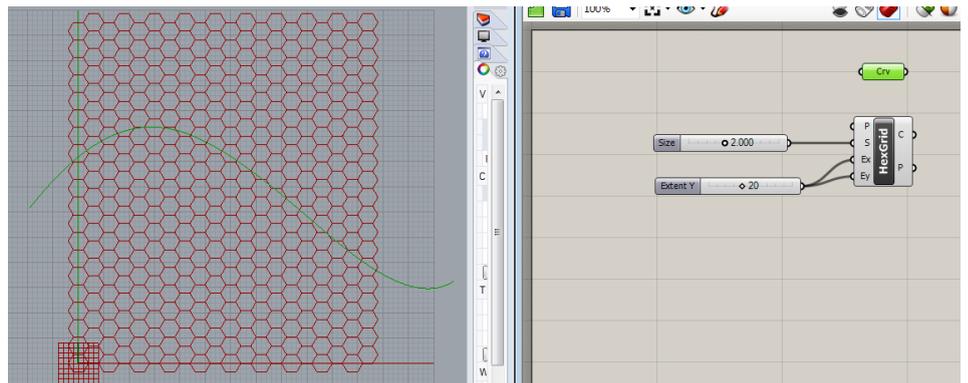
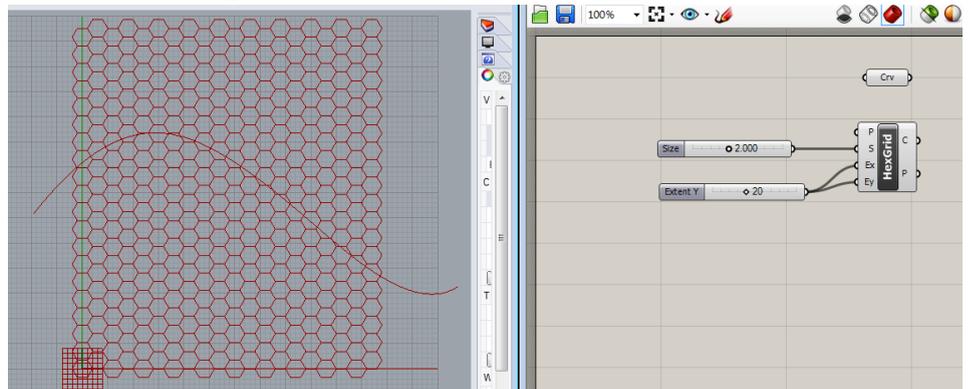


2.3. Configure no Grasshopper a curva que foi desenhada no Rhino.

- a. No Grasshopper, clique com o botão direito no componente **Curve** que acabou de ser adicionado, irá abrir um menu:

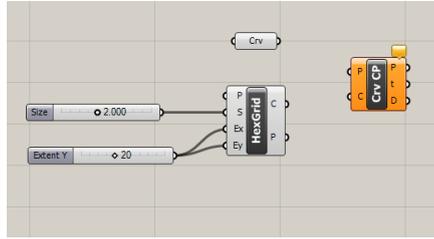


- b. No menu aberto, clique sobre a opção **Set one Curve**.
- c. No Rhino, clique sobre a curva que foi desenhada. Essa curva piscará rapidamente em amarelo e, em seguida, ficará vermelha. No Grasshopper, o componente **Curve** deverá ficar cinza e, ao clicar nele, a curva deverá ficar verde na tela do Rhino, isso indica que a curva foi corretamente configurada.

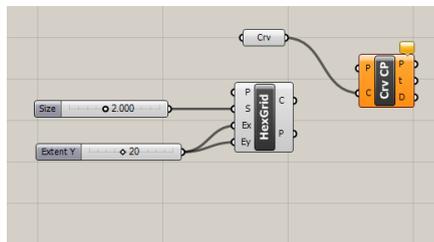


3. **Encontrar a distância entre o centro do hexágono e o ponto mais próximo da curva.**
Existe um componente específico para isso, o componente Curve Closest Point

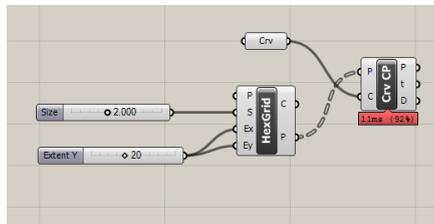
- 3.1. Encontrar a distância entre a curva e o centro dos hexágonos: **Curve > Analysis > Curve Closest Point**



- 3.2. Conecte a saída do componente **Curve** (que tem a curva desenhada no Rhino e configurada no Grasshopper) com o parâmetro **C (Curve)** do componente **Curve Closest Point**.



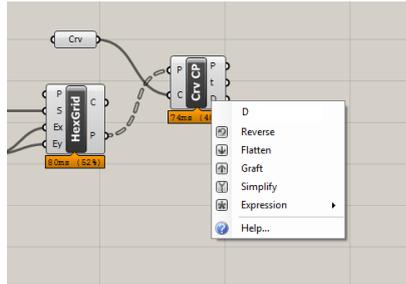
- 3.3. Conecte a saída **P (Points as tree)** do componente de grid hexagonal ao parâmetro **P (Point)** do componente **Curve Closest Point**



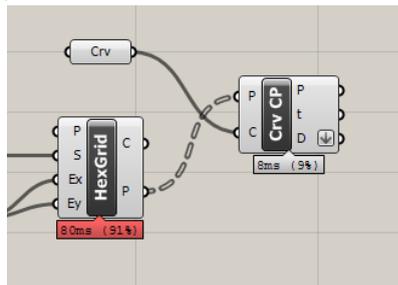
4. Remapeamento das distâncias para um outro intervalo. O remapeamento, é utilizado quando precisamos modificar um parâmetro em relação a outro. Esse recurso permite que nós consigamos criar relações entre diferentes propriedades de objetos. É um recurso poderoso que trabalha com a construção de domínios de conjuntos numéricos, uma boa explicação do conceito e de seu uso pode ser encontrada nas sugestões de leitura do item 7.1 deste tutorial.

- 4.1. No componente **Curve Closest Point**, precisaremos realizar a **operação flatten** no parâmetro de saída **D (Distance)**. A operação de flatten diz respeito ao conjunto de dados, ela modifica a estrutura de dados que estamos trabalhando. No nosso caso, a operação toma um conjunto de dados em diferentes listas e as une em uma única.

- a. No Componente **Curve Closest Point**, clique com o botão direito sobre o parâmetro de saída **D (Distance)**. Irá abrir o seguinte menu:

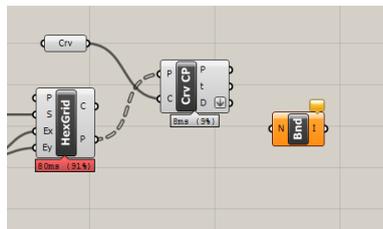


- b. Com o menu aberto, clique na opção **Flatten**. Após clicar, deverá aparecer o ícone da operação (uma seta para baixo) ao lado do parâmetro de saída.

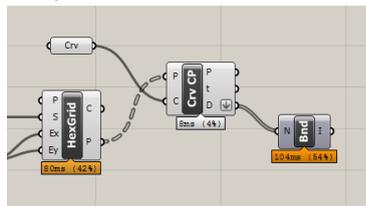


4.2. **Encontrar o domínio (intervalo de valores) das distâncias do centro de cada um dos hexágonos até a curva que definimos.** Esse domínio costuma ser chamado de **Domínio Fonte**, dele sairão valores que irão passar por uma transformação: a cada valor do Domínio Fonte, corresponderá um outro valor no nosso segundo domínio de interesse, o **Domínio Alvo**.

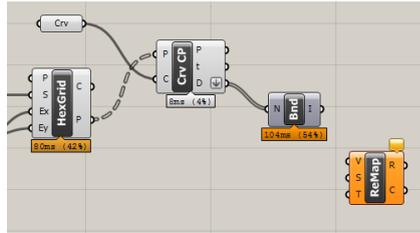
- a. Para encontrar esses limites, utilize o componente **Bounds**: [Maths > Domain > Bounds](#)



- b. Conecte o parâmetro de saída **D (Distance)** do componente **Curve Closest Point** no parâmetro de entrada **N (Numbers as list)** do componente **Bounds**



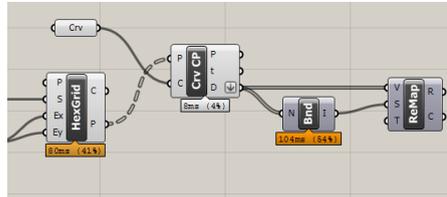
4.3. Remapear os valores através do componente Remap Numbers: [Maths > Domain > Remap Numbers](#)



Conecte:

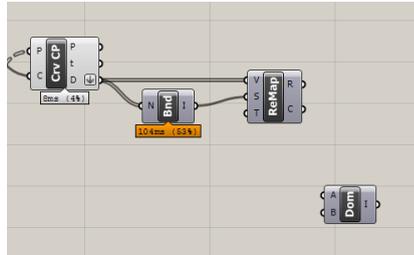
- o parâmetro de saída **D (Distance)** do componente **Curve Closest Point** ao parâmetro de entrada **V (Value)** do componente **ReMap**;

- o parâmetro de saída **I (Domain)** do componente **Bounds** no parâmetro de entrada **S (Source Domain)** do componente **ReMap**.

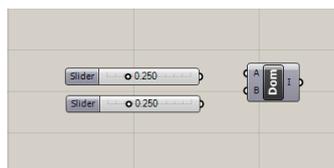


4.3. Definir o domínio alvo utilizando o componente Construct Domain

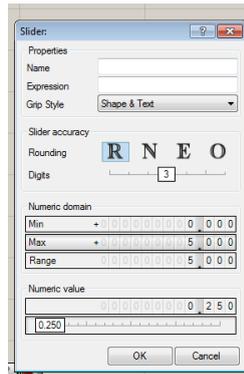
- Inserir o componente **Construct Domain**: **Maths > Domain > Construct Domain**



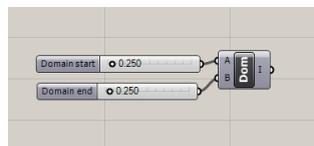
- Definir os limites do intervalo através de dois componentes **Number Slider** (**Params > Input > Number Slider**.)



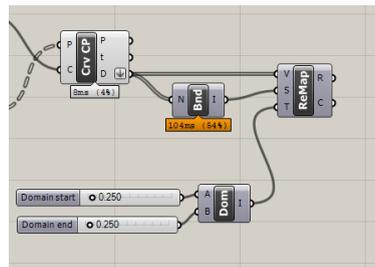
Os componentes **Number Slider** devem estar configurados para valores reais (R) e limites Min = 0 e Max = 5)



Conecte os parâmetros de saída de cada um dos componentes **Number Slider** aos parâmetros de entrada **A (Domain Start)** e **B (Domain End)** do componente **Construct Domain**

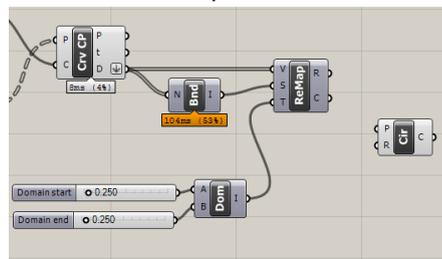


4.4. Conectar o Domínio Alvo ao Componente Remap: conectar o parâmetro de saída **I (Domain)** do componente **Construct Domain** no parâmetro de entrada **T (Target)** do componente **Remap**.

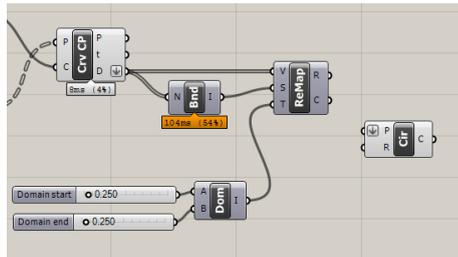


5. Baseados nas distâncias remapeadas, criaremos círculos. O raio desses círculos serão determinados pelas distâncias calculadas pelo componente Remap.

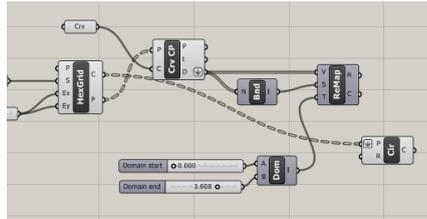
5.1. Adicione um componente de círculo: **Curve > Primitive > Circle**



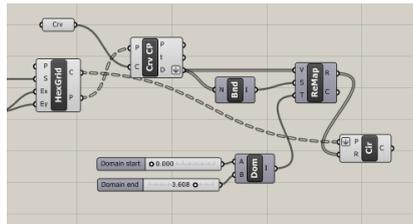
5.2. Aplique a **operação Flatten** ao parâmetro de **entrada P (Plane)** do componente **Circle**



5.3. Conecte o parâmetro de saída **C (Cells as tree)** do grid de hexágonos ao parâmetro de entrada **P (Plane)** do componente **Circle**

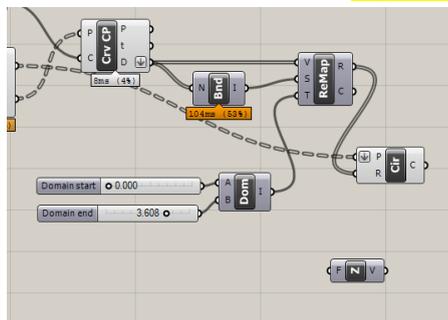


5.4. Defina os raios das circunferências com os números remapeados: conecte o parâmetro de saída **R (Mapped)** do componente **Remap Numbers** ao parâmetro de entrada **R (Radius)** do componente **Circle**

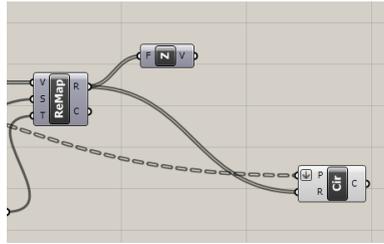


6. **Extrusão das circunferências:** para tornar o resultado um pouco mais visível em 3D, nós podemos extrudar os círculos baseados na distância com relação à curva. Embora possamos remapear os números novamente, também é possível usar uma simples multiplicação e vetor unitário na direção Z para definir a altura. Realizaremos essa segunda abordagem

6.1. Adicione um vetor unitário Z: **Vector > Vector > Unit Z**

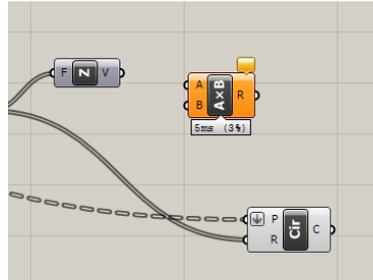


6.2. Conecte o parâmetro de saída **R (Mapped)** do componente **Remap** ao parâmetro de entrada **F (Factor)** do componente **Unit Z**

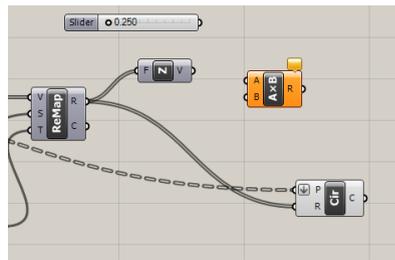


6.3. Multiplique o vetor unitário Z por um fator:

- a. Adicione um componente de multiplicação: **Maths > Operators > Multiplication**

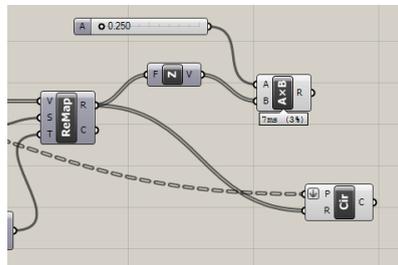


- b. Adicione um componente **Number Slider** com valores reais (R) e valores Min = 0 e Max = 10.



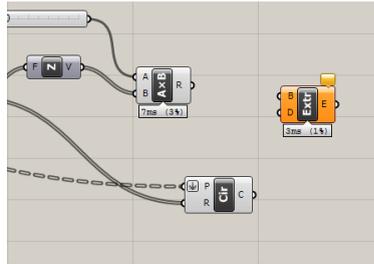
- c. Conecte:

- o parâmetro de saída do componente **Number Slider** no parâmetro de entrada **A** do componente **Multiplication**;
- o parâmetro de saída **V (Unit Vector)** do componente de vetor unitário ao parâmetro de entrada **B** do componente **Multiplication**.



6.4. Extrudar os círculos

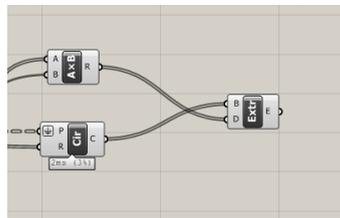
- a. Adicione o componente de extrusão: **Surface > Freeform > Extrude**



b. Conecte:

- o parâmetro **R (Result)** do componente de multiplicação no parâmetro de entrada **D (Direction)** do componente de extrusão;

- o parâmetro de saída **C (Circle)** do componente **Circle** ao parâmetro de entrada **B (Base)** do componente de extrusão.



7. Exploração do algoritmo:

Agora você pode alterar diferentes valores para os parâmetros definidos e também relações.

7.1. Sugestões de leitura: para aprofundar o entendimento do componente Remap, sugerimos a leitura de:

- a. o item 2.5. *Remapping Numbers/Attractors* de TEDESCHI (2014); e
- b. o artigo *online: How to use Remap in Grasshopper*, do arquiteto Thomas Tait do escritório Snohetta, disponível em: <https://hopific.com/remap-in-grasshopper/>