

Como criar seu primeiro componente com FMX

GABRIEL OLIVEIRA

Experiência Profissional

02

PORTFÓLIO CRIATIVO

PORQUE COMPONENTIZAR

HERANÇA

CASTELO

MÃO NA MASSA

MVP EMBARCADERO

Criador da comunidade Delphi na Veia

03



Herança

Herança

Não tem como criar um componente sem usar o conceito de Orientação Objeto e Herança

Escolher quem vai ser o “PAI” do componente é muito importante.

Este momento vai decidir o comportamento, e o tipo de componente que vamos gerar no final



05

Herança

TRectangle

Caso o componente escolhido para ser o “Pai” seja o Rectangle, vamos ter todas essas propriedades acessíveis pelo Self.

06

Properties Events

Property	Type	Value	
Align	None	[akLeft,akTop]	
Anchors	<input type="checkbox"/>	False	
ClipChildren	<input type="checkbox"/>	False	
ClipParent		[TopLeft,TopRight,BottomLeft,BottomRight]	
Corners		Round	
CornerType		crDefault	
Cursor		dmManual	
DragMode	<input checked="" type="checkbox"/>	True	
Enabled	<input checked="" type="checkbox"/>	True	
EnableDragHighlight		(Brush)	
Fill		50	
Height			
Hint	<input checked="" type="checkbox"/>	True	
HitTest		LiveBindings	
LiveBindings		LiveBindings Designer	
LiveBindings Designer		<input type="checkbox"/>	False
Locked		(TBounds)	
Margins		Rectangle1	
Name		1	
Opacity		(TBounds)	
Padding		<input checked="" type="checkbox"/>	True
ParentShowHint		(TPosition)	
PopupMenu		0	
Position		(TPosition)	
RotationAngle		(TPosition)	
RotationCenter		(TPosition)	
Scale		0	
ShowHint	<input type="checkbox"/>	False	
Sides		[Top,Left,Bottom,Right]	
Size		(TControlSize)	
Stroke		(Brush)	
StyleName		0	
Tag		(TTouchManager)	
Touch	<input checked="" type="checkbox"/>	True	
Visible		217	
Width		20	
XRadius		20	
YRadius		20	

Herança

Begin

07

Herança

System.TObject

Até pai: Propriedades dos métodos do sistema

Delfos

```
TObject = class
```

C++

```
class __DELPHICLASS_TOBJECT TObject
```

Propriedades

Tipo	Visibilidade	Fonte	Unidade	Pai
.class	público	System.pas systobj.h	Sistema	Sistema

Descrição

TObject é o ancestral final de todos os objetos e componentes.

TObject encapsula o comportamento fundamental comum aos objetos, introduzindo métodos que:

- Crie, mantenha e destrua instâncias do objeto alocando, inicializando e liberando a memória necessária.
- Responda quando instâncias de objeto são criadas ou destruídas.
- Retorna informações de tipo de classe e instância em um objeto e informações de tipo de tempo de execução (RTTI) sobre suas propriedades publicadas.
- Suporte ao tratamento de mensagens.
- Interfaces de suporte implementadas pelo objeto.

Herança

TComponent

A classe TComponent cria componentes não visuais

Herança

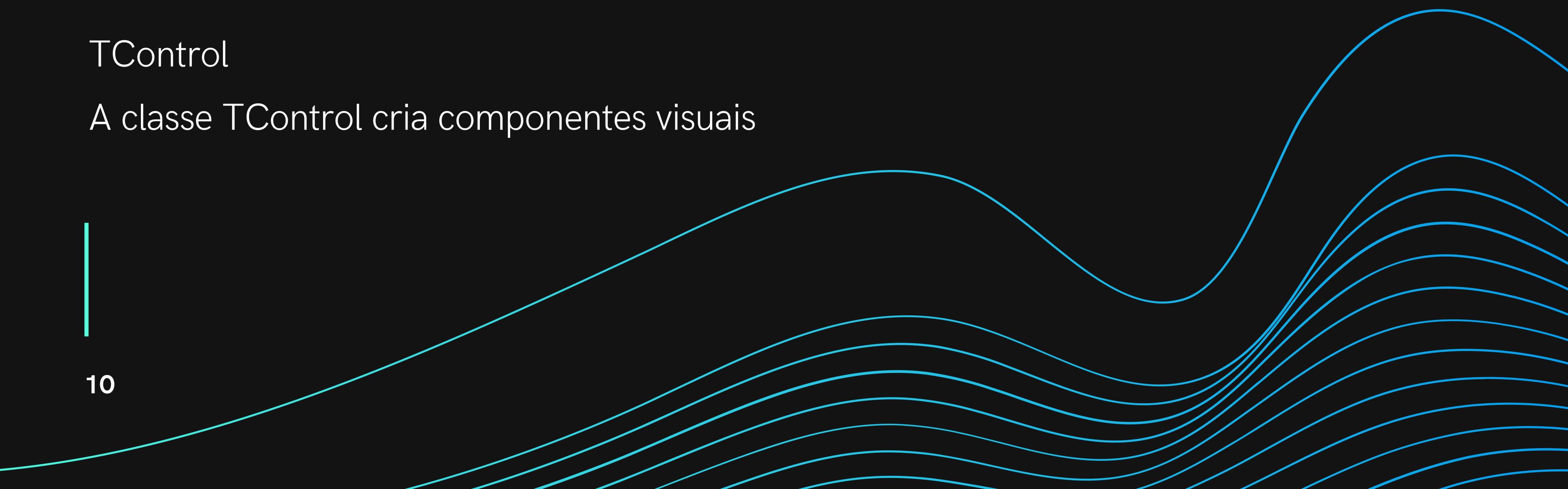
TComponent

A classe TComponent cria componentes não visuais

TControl

A classe TControl cria componentes visuais

10



Herança

FMX.Objects.TRectangle

Up to Parent: FMX.Objects

Methods

Properties

Events

Fields



Castelo



Properties

```
| type
| TBotao = class(TComponent)
| private
|   function GetText: string;
|   procedure SetText(const Value: string);
|   { Private declarations }
| protected
|   { Protected declarations }
| public
|   { Public declarations }
| published
|   { Published declarations }
|   property Text: string read GetText write SetText;
| end;
```

Properties

-
- **property** OnStart: TNotifyEvent **read** FOnStart **write** FOnStart;
- **property** OnStop: TNotifyEvent **read** FOnStop **write** FOnStop;
-

type System.Classes.TNotifyEvent = procedure(Sender: TObject) of object

Parameters

Sender

TObject

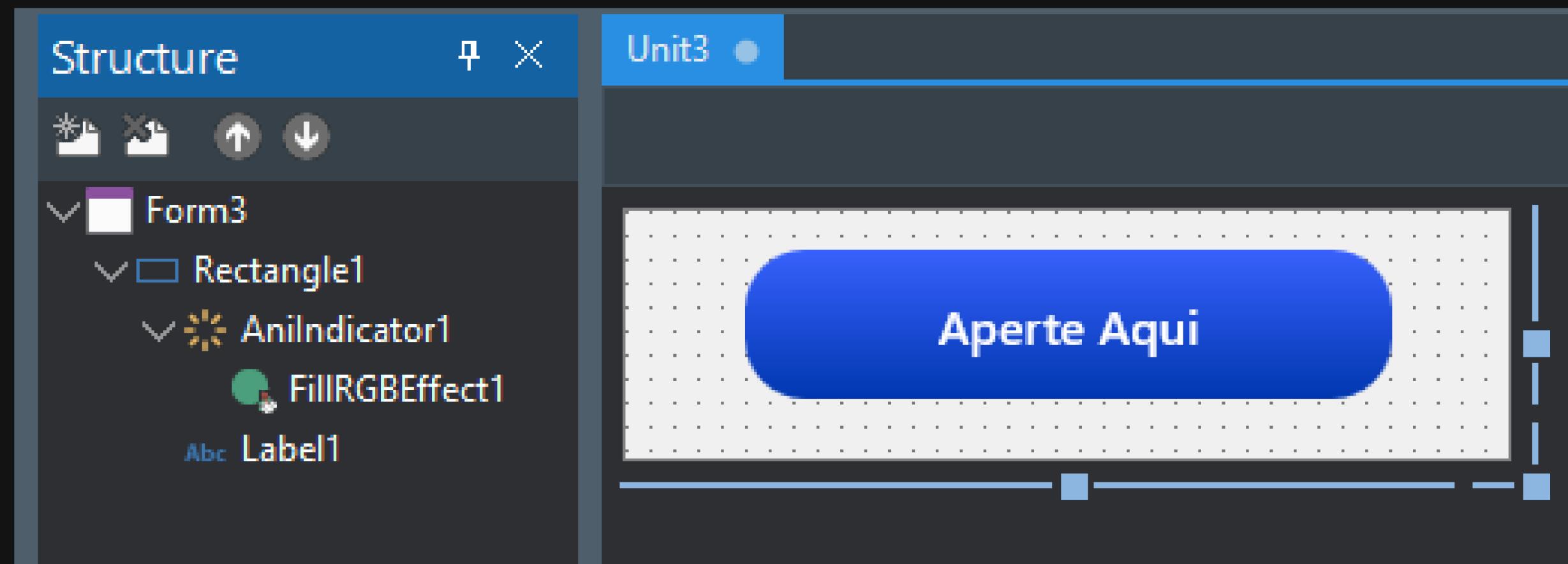
Register

```
procedure Register;
└── implementation
    └── procedure Register;
        ┌── begin
        │   RegisterComponents('EmbarcaderoConference', [TBotao]);
        └── end;
```

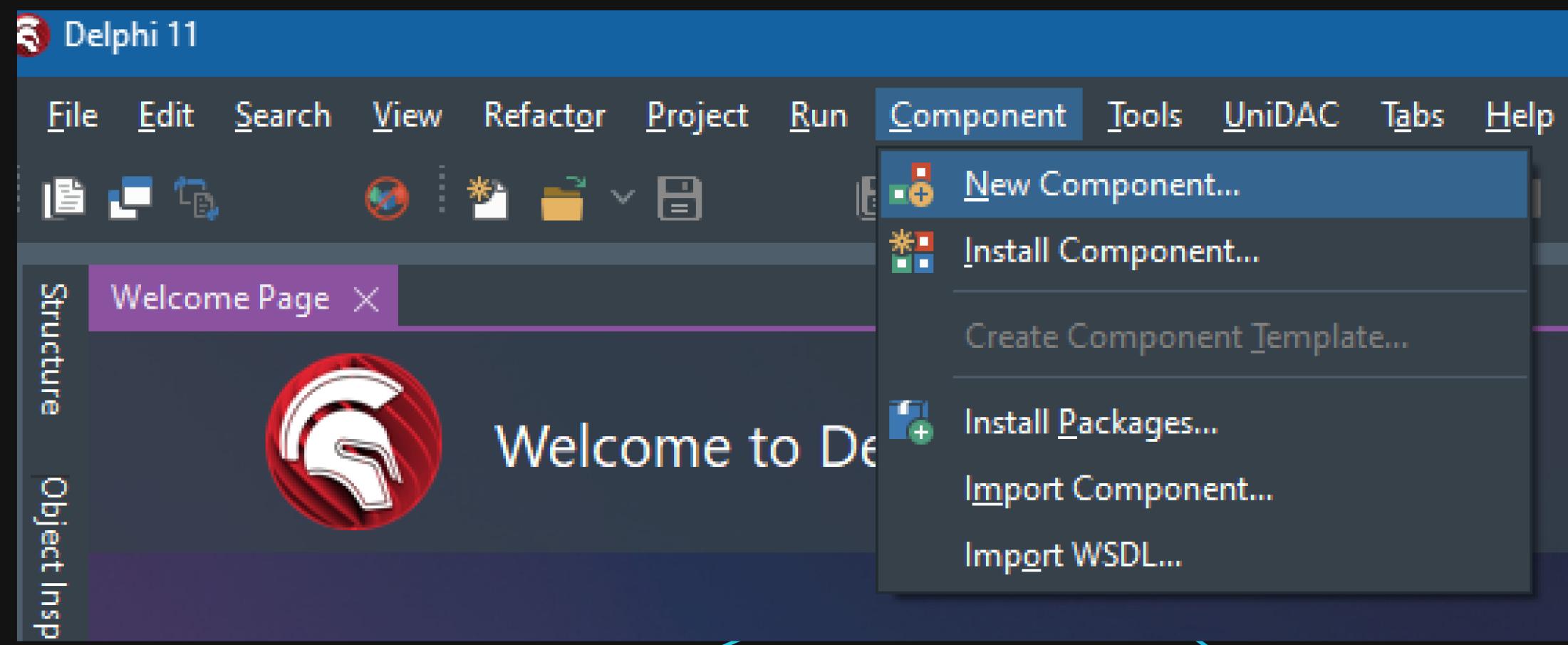
Register

```
25 | public  
· |   constructor Create(AOwner: TComponent); override;  
· |  
· |
```

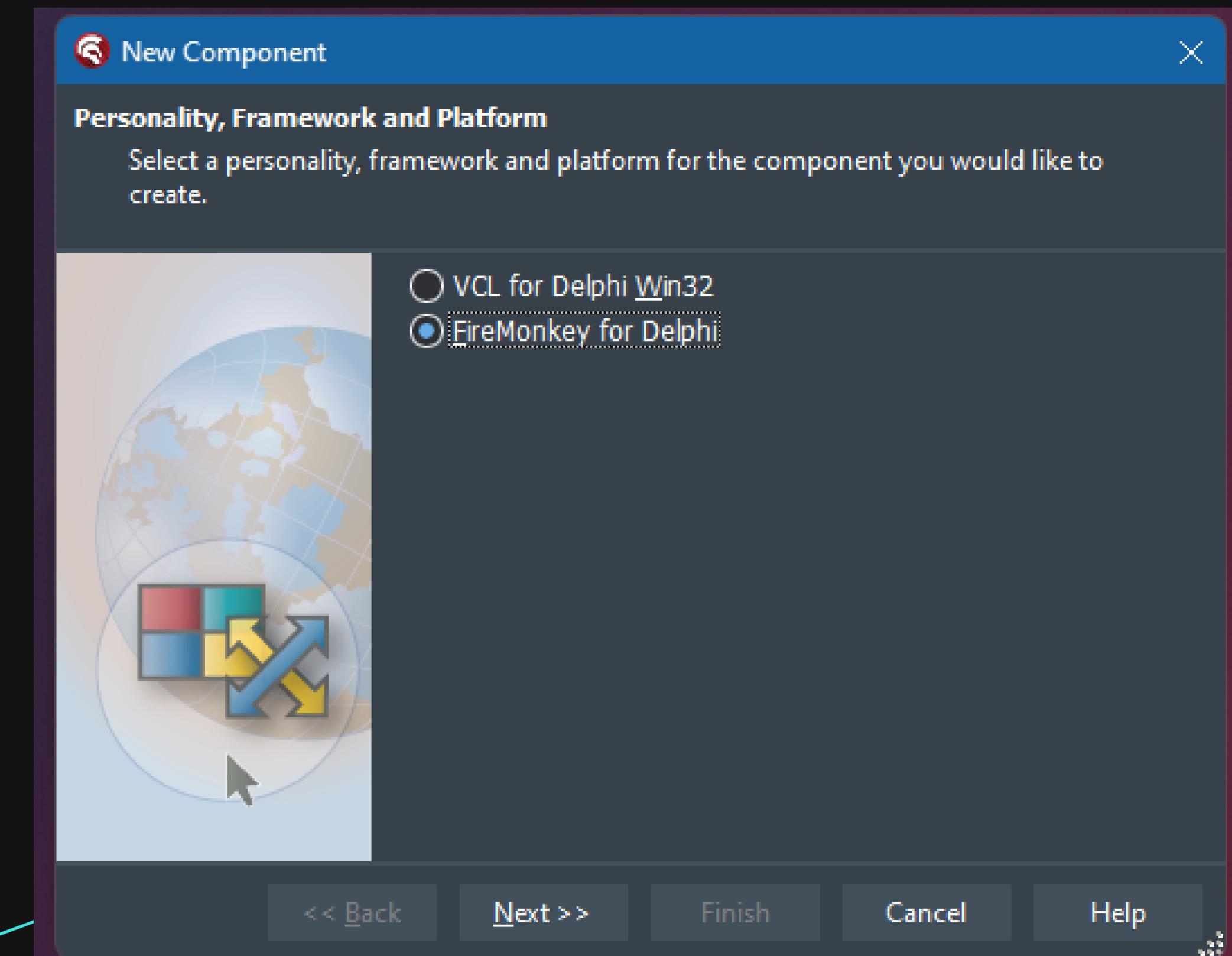
Jeito tradicional



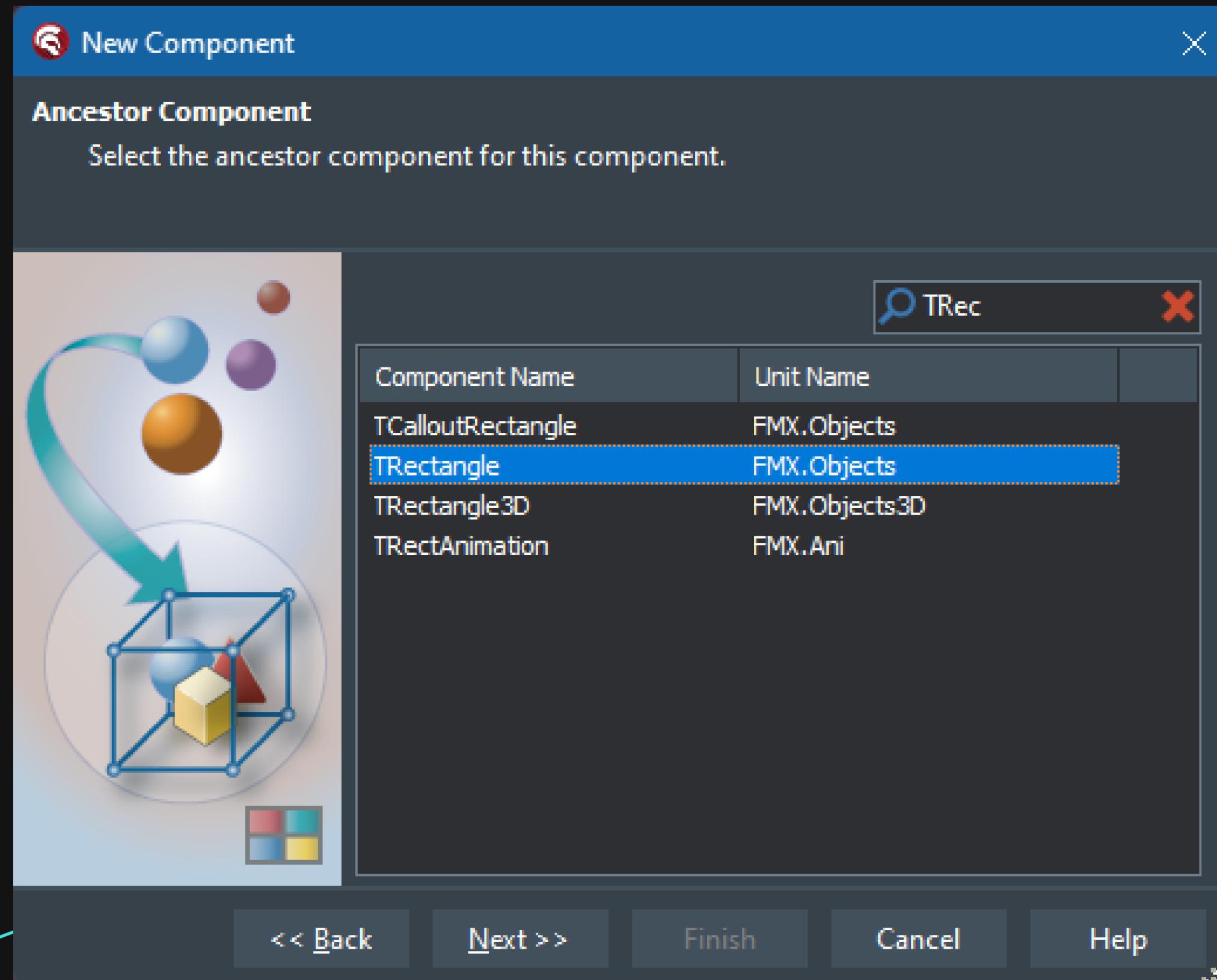
Componentizar



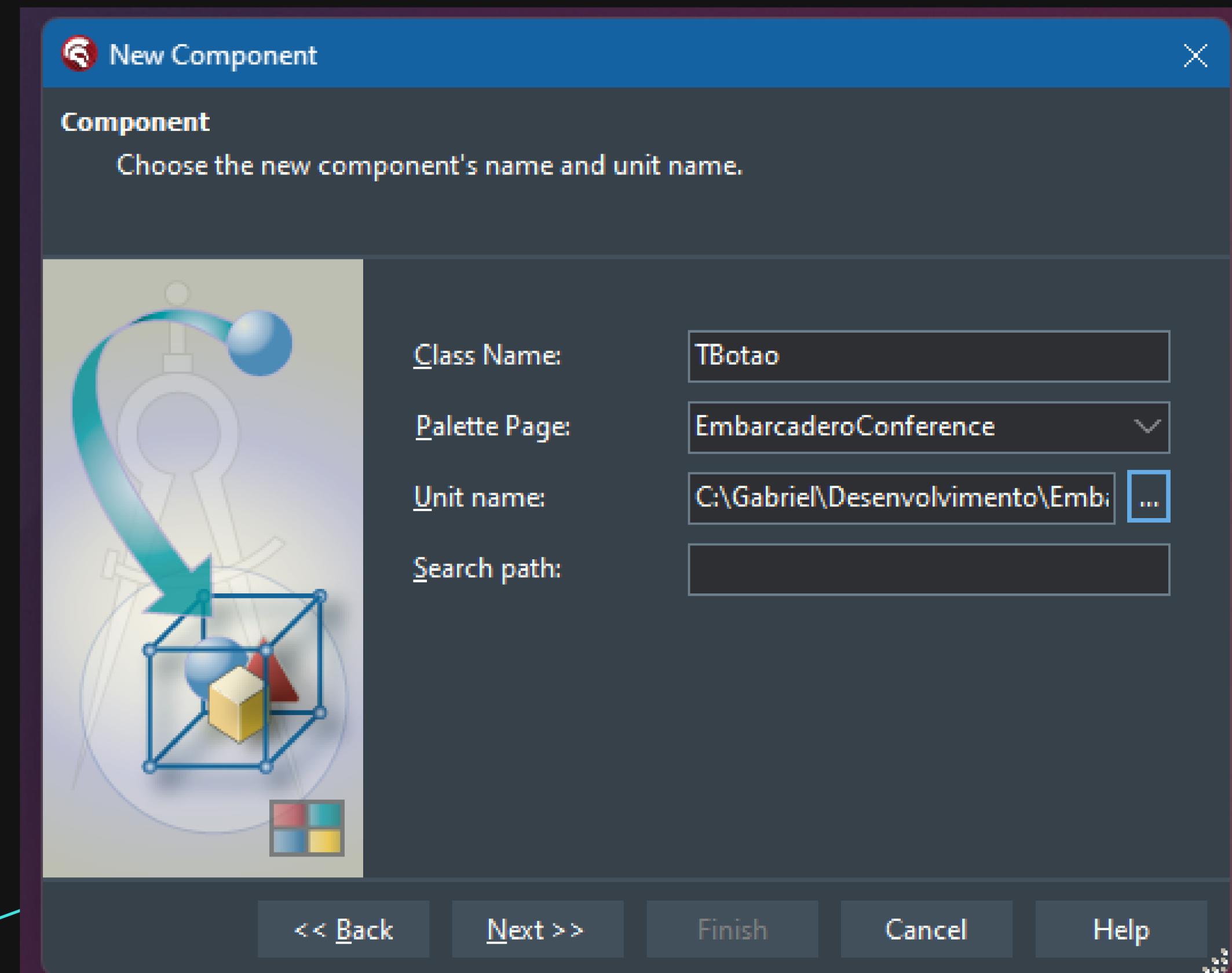
Componentizar



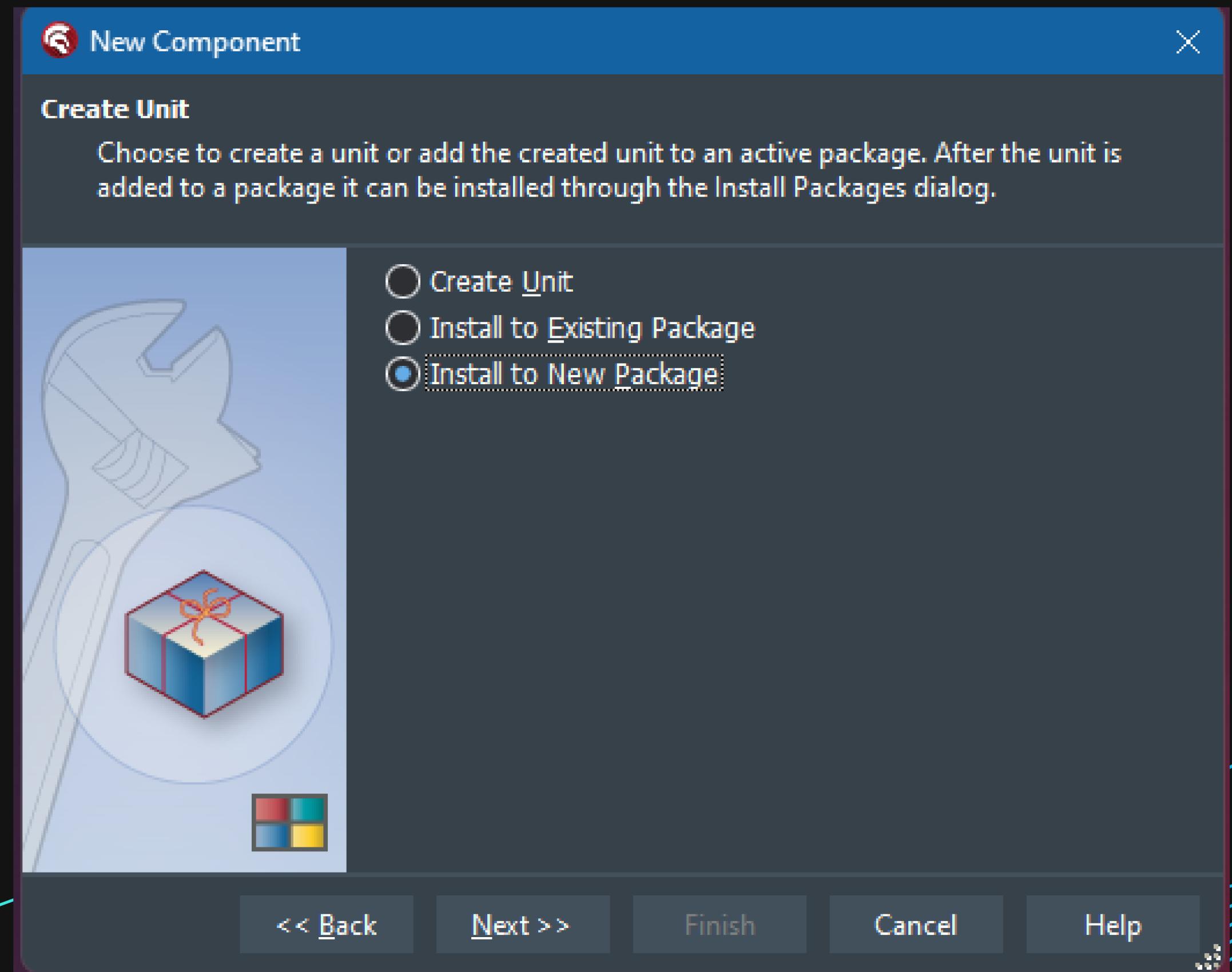
Componentizar



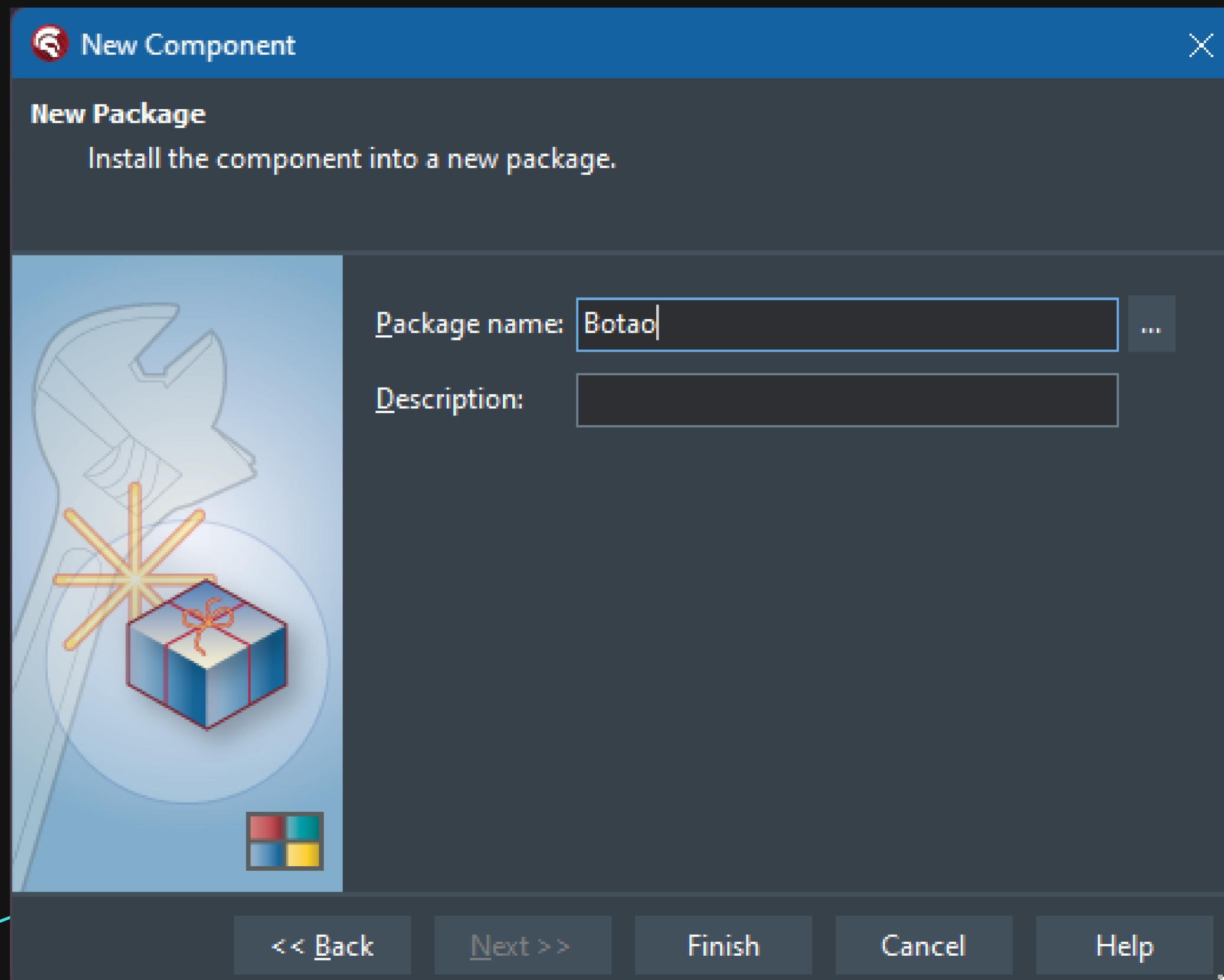
Componentizar



Componentizar



Componentizar



23

