

# O Ambiente App Designer do MATLAB

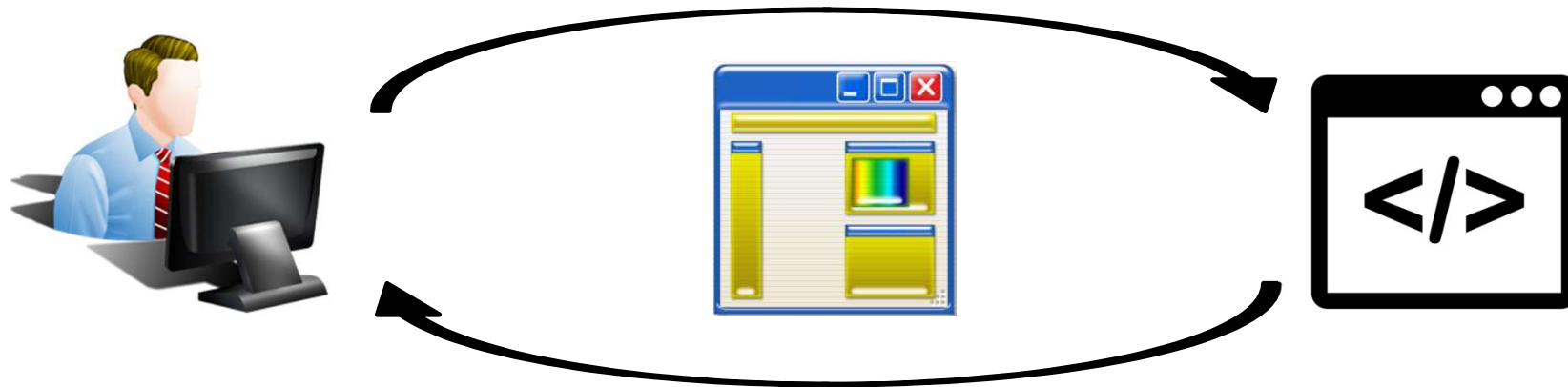
Pedro Cortez Lopes  
Rafael Lopez Rangel  
Luiz Fernando Martha



**CIV2801 – Fundamentos da Computação Gráfica Aplicada**  
2023.2

# Interface Gráfica do Usuário (GUI)

- Uma interface gráfica permite a interação do usuário do computador com um programa por meio de elementos gráficos (botões, menus, etc.)
- O ambiente App Designer permite a criação manual de uma interface gráfica do utilizador (GUI – Graphical User Interface) de forma rápida e interativa
- A comunicação entre as ações realizadas pelo usuário sobre os elementos gráficos e a resposta do programa se dá por meio do paradigma da Programação Orientada a Eventos
- Interfaces gráficas também podem ser criadas no MATLAB apenas com código de programação (criação programática), ou pelo ambiente GUIDE



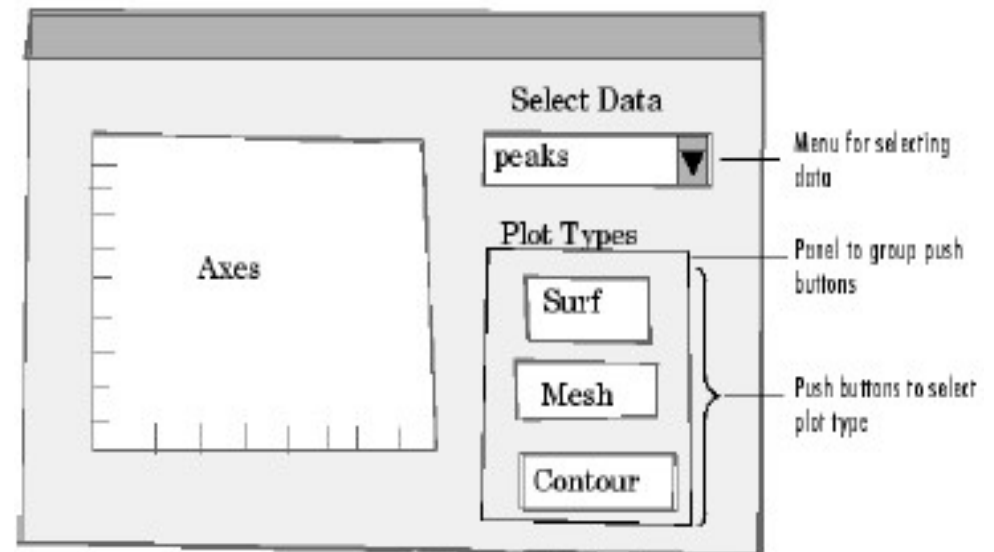
# Interface Gráfica do Usuário (GUI)

Antes de começar a montar uma interface gráfica é necessário projetá-la. Para isso, deve-se pensar em como se dará a interação do usuário com o programa:

- Quais funcionalidades estarão disponíveis e como serão executadas
- Quais componentes serão utilizados para permitir o usuário realizar essas tarefas
- Prever possíveis erros de uso do programa para cercar as ações do usuário
- Organização do layout

Para projetar interfaces com um grau de complexidade relativamente alto, recomenda-se esboçar o layout em uma folha de papel.

<http://asktog.com/atc/principles-of-interaction-design/>



# Interface Gráfica do Usuário (GUI)

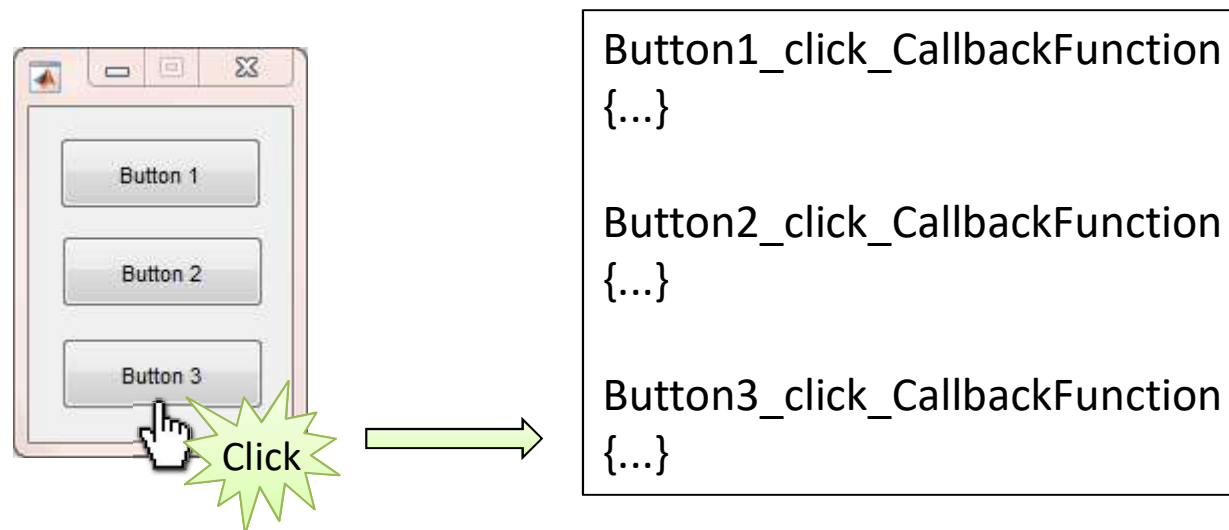
Após projetar a interface, deve-se decidir qual técnica será utilizada para criá-la.

No MATLAB é possível criar uma interface de três formas: Utilizando o ambiente App Designer, o ambiente GUIDE, ou fazendo programaticamente, apenas através de código de programação (pode ser feito baseando-se em manipular Figuras – ideia por trás do GUIDE – ou seguindo o padrão de código orientado a objeto do App Designer).

	Pontos positivos	Pontos negativos
App Designer	<ul style="list-style-type: none"><li>• Ambiente gráfico interativo integrado com código</li><li>• Formalização de código seguindo POO</li><li>• De manuseio fácil e intuitivo</li><li>• Perspectiva de manutenção e avanços no futuro</li></ul>	<ul style="list-style-type: none"><li>• Ainda em desenvolvimento, não está otimizado (aplicações de moderada complexidade podem ficar lentas)</li></ul>
GUIDE	<ul style="list-style-type: none"><li>• Ambiente gráfico interativo para manuseio de componentes gráficos</li><li>• Relativamente eficiente</li><li>• Maior extensão de referências e fóruns online para buscar ajuda</li></ul>	<ul style="list-style-type: none"><li>• Código menos formal para o contexto de um programa estruturado</li><li>• Menos opções de componentes gráficos</li><li>• Sem perspectiva de avanços</li></ul>
Código	<ul style="list-style-type: none"><li>• Maior controle de posição e comportamento de todos os componentes gráficos em cada diálogo</li><li>• Maior controle da interação entre diferentes GUIs</li><li>• Maior liberdade ao desenvolvedor</li></ul>	<ul style="list-style-type: none"><li>• Código deve ser fornecido para descrever todos os aspectos da interface gráfica</li><li>• Em geral, maior investimento de tempo</li></ul>

# Programação Orientada a Eventos

- O fluxo do código é guiado por indicações externas chamadas eventos.
- Eventos são as diferentes ações que usuários podem realizar sobre os componentes adicionados à interface.
- Cada evento está associado à uma função chamada *Callback*, disparada quando se verifica a ocorrência de tal evento, que define a reação do programa.
- No App Designer, as *Callbacks* são associadas a métodos, no contexto da POO, de uma classe que descreve a aplicação gráfica.



## App Designer – Quick Tutorial

- Um simples tutorial para o uso do App Designer é disponibilizado pela MathWorks:

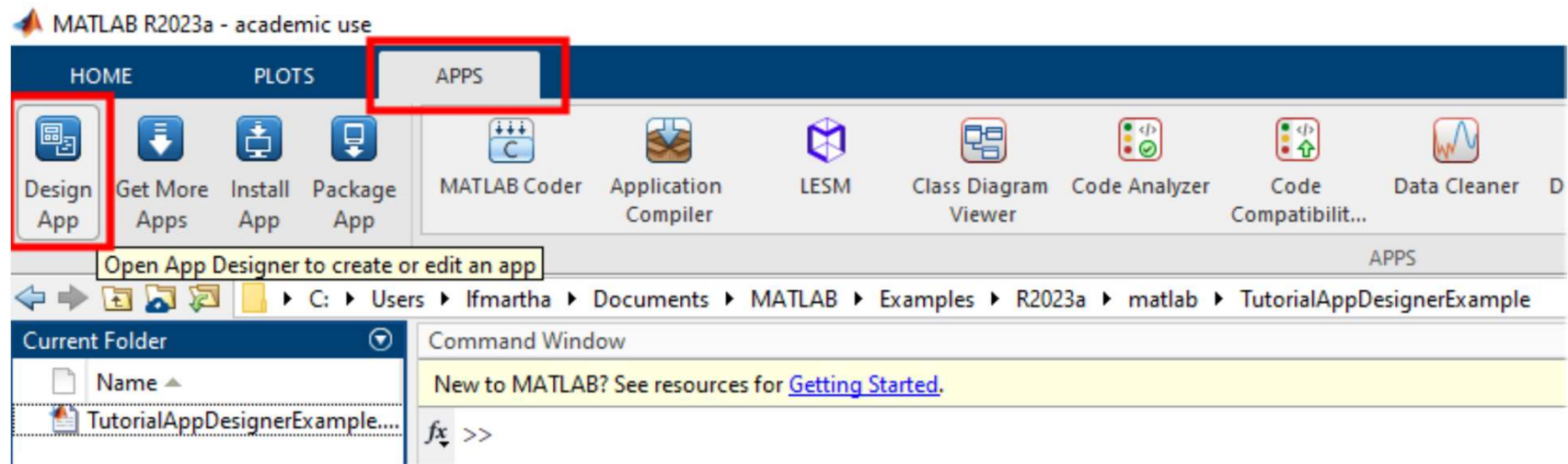
[https://www.mathworks.com/help/matlab/creating\\_guis/create-a-simple-app-or-gui-using-app-designer.html](https://www.mathworks.com/help/matlab/creating_guis/create-a-simple-app-or-gui-using-app-designer.html)

- Também está disponível em vídeo:

<https://www.mathworks.com/videos/app-designer-overview-1510748719083.html>

# App Designer – Quick Tutorial

- Para executar o tutorial do App Designer, abra App Designer Start Page

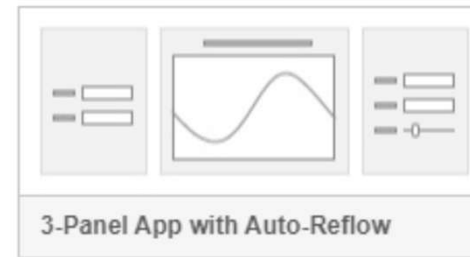
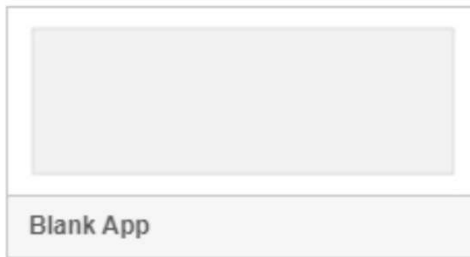


# App Designer – Quick Tutorial

- Clique em “Show examples” in the Apps section.

[Getting Started](#) | [How-to Videos](#) | [GUIDE Migration Strategies](#) | [Display Graphics in App Designer](#) | [Find and Create UI Components](#) | [Release Notes](#)

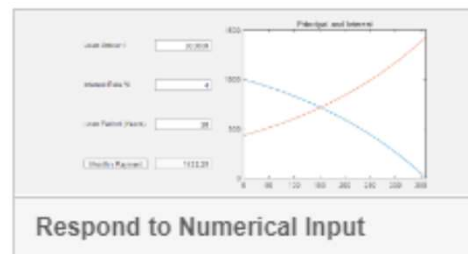
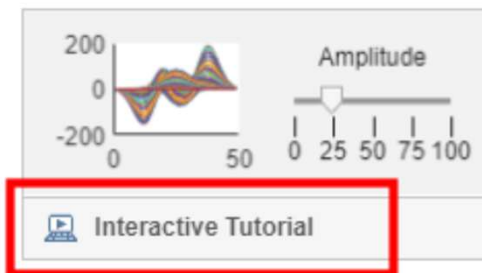
## Apps



[Show examples](#)

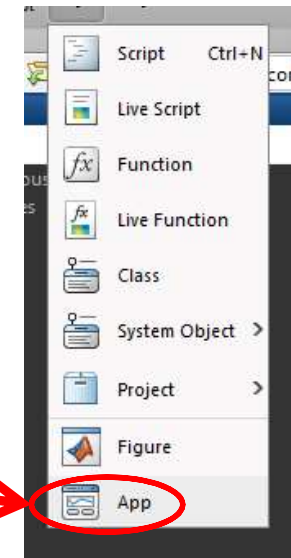
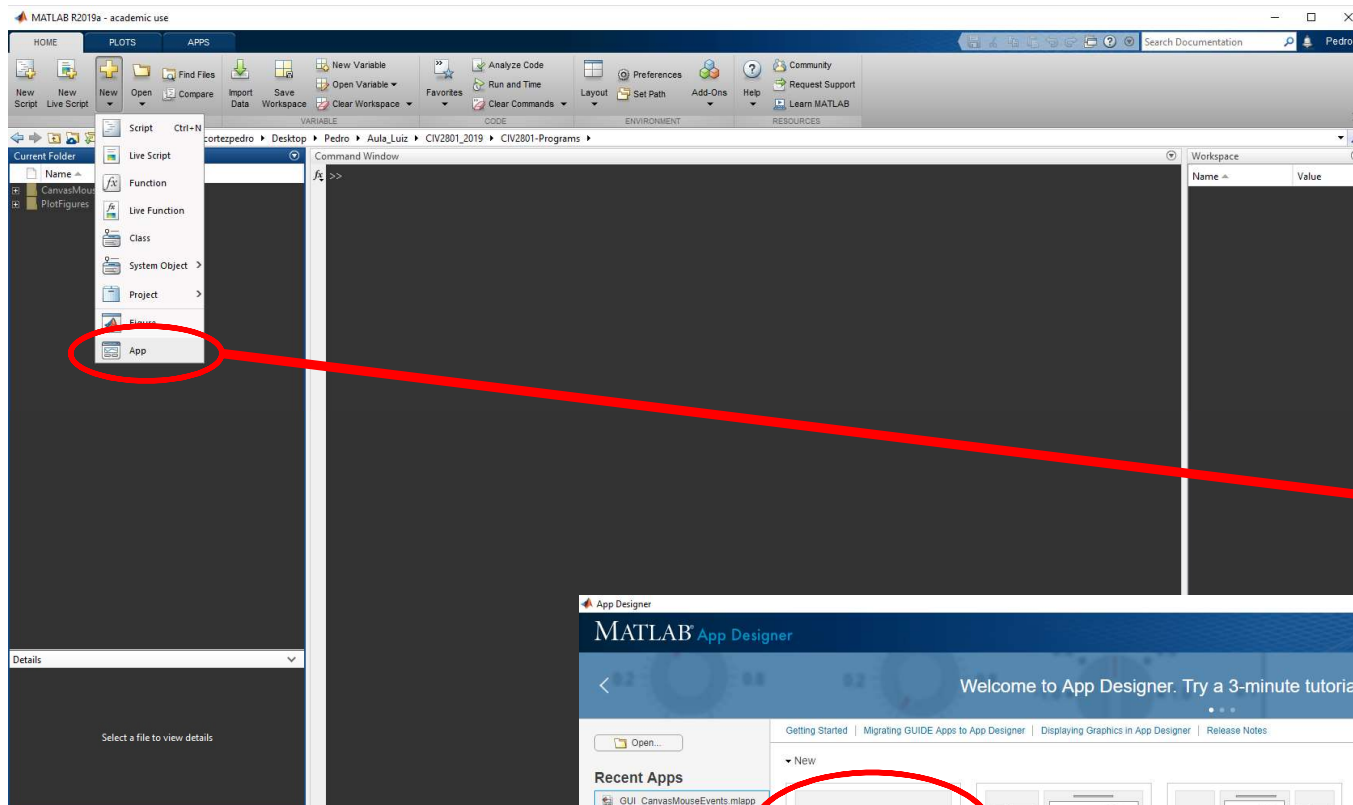
- Selecione “Interactive Tutorial”

## Examples: General

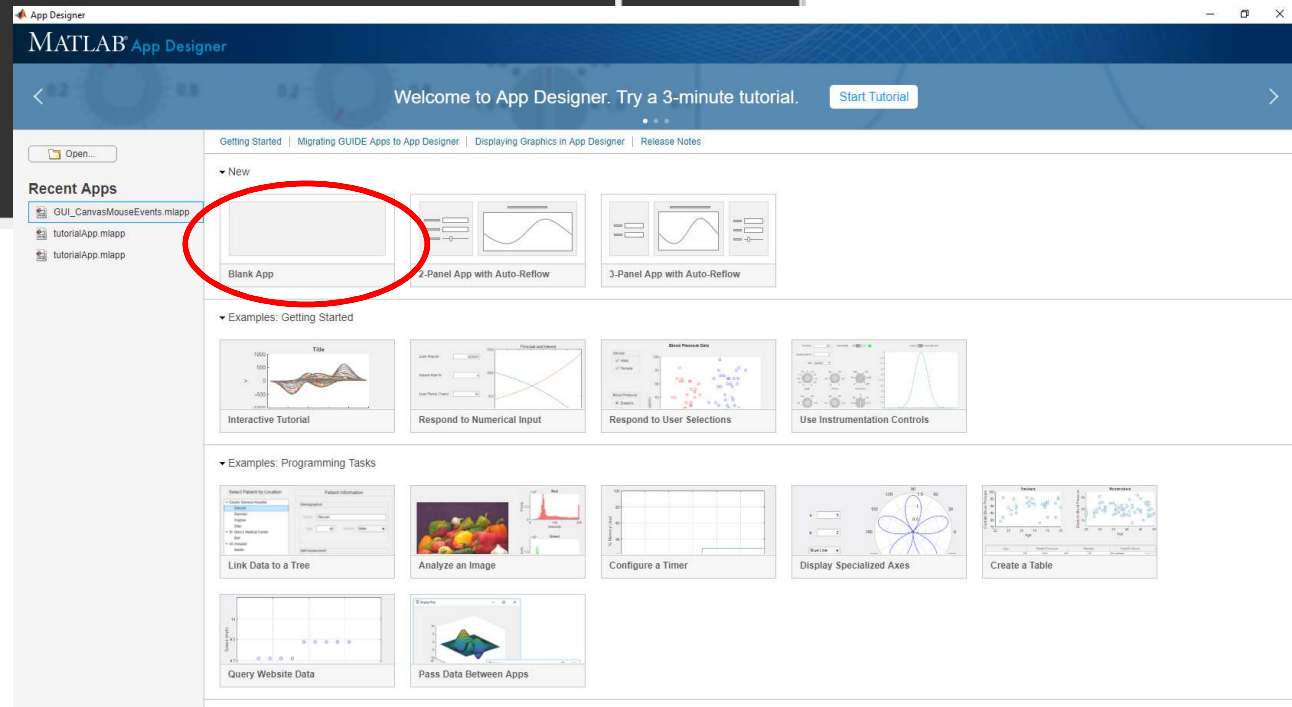




# Inicializando uma Janela de Interface Gráfica Vazia

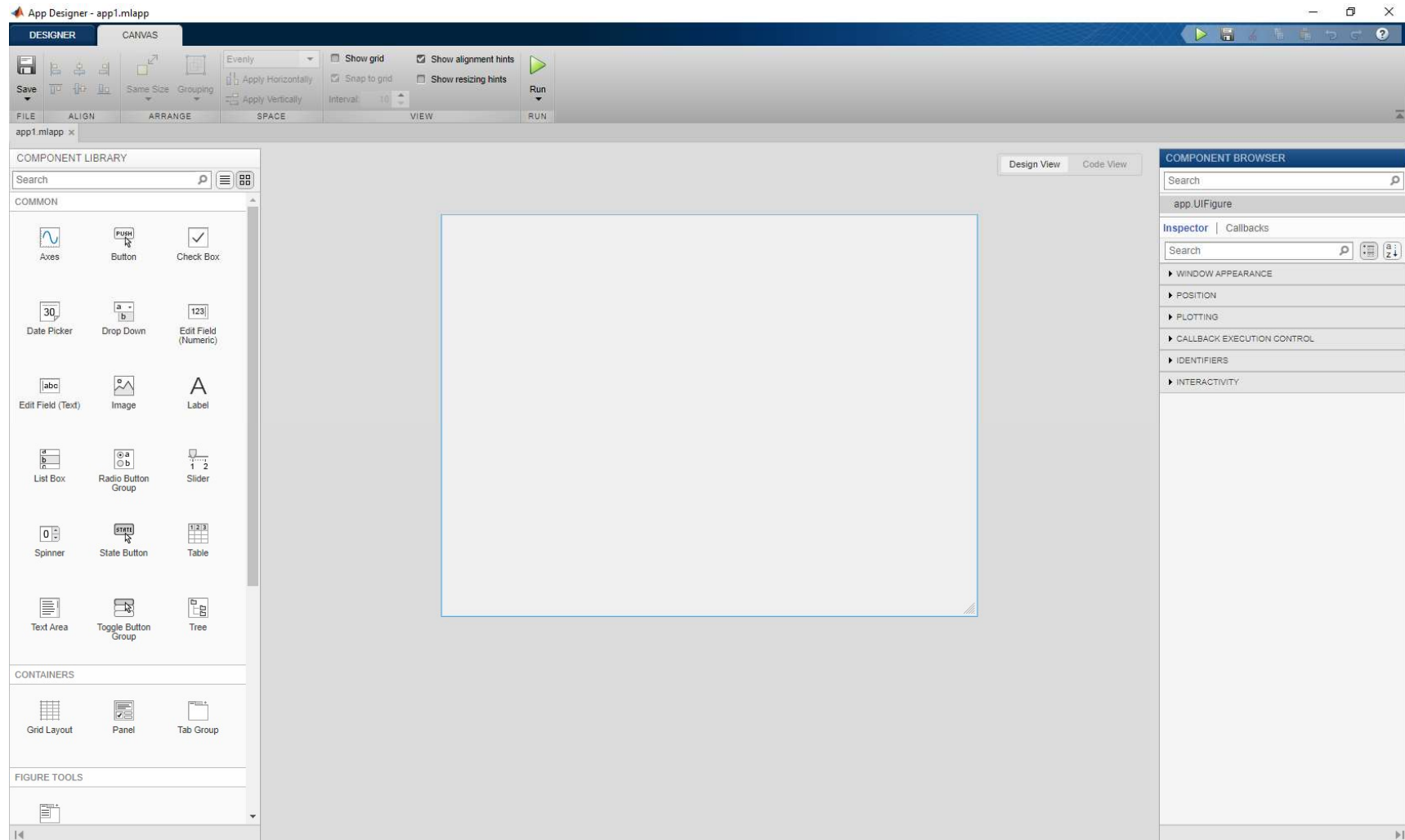


Também pode ser chamado usando comando *appdesigner* no prompt principal do MATLAB





















# App Designer

É possível elaborar o layout de uma GUI, atribuindo e modificando as propriedades dos componentes gráficos. É habilitado um sistema *drag and drop* para posicionamento interativo de componentes.






# App Designer – Componentes Gráficos


## COMMON

 Axes	 Button	 Check Box
 Date Picker	 Drop Down	 Edit Field (Numeric)
 Edit Field (Text)	 Image	 Label
 List Box	 Radio Button Group	 Slider
 Spinner	 State Button	 Table
 Text Area	 Toggle Button Group	 Tree











## CONTAINERS

 Grid Layout	 Panel	 Tab Group
--	--	--

## FIGURE TOOLS

 Menu Bar
---

## INSTRUMENTATION

 90 Degree Gauge	 Discrete Knob	 Gauge
 Knob	 Lamp	 Linear Gauge
 Rocker Switch	 Semicircular Gauge	 Switch
 Toggle Switch		

# App Designer – Componentes Gráficos

The screenshot displays the App Designer interface with various graphical components arranged on a light gray background. On the left, there is a plot titled "Title" with x and y axes ranging from 0 to 1. Below the plot are several input controls: an "Edit Field" containing the number 0, a "Slider" with a range from 0 to 100, a "Spinner" also containing 0, and a "Text Area". In the center, there is a "Node" tree with three sub-nodes: "Node2", "Node3", and "Node4", next to a blue 3D cube icon. To the right of the cube is a "Gauge" with a scale from 0 to 100 and a red needle pointing to 0. Below the gauge is a "Knob" with four positions: "Low", "Medium", "High", and "Off". At the top right, there are a "Drop Down" menu showing "Option 1", a "Date Picker" showing "mm/dd/yyyy", a "Button", and a "Check Box". On the far right, a vertical panel shows a tree view of the component hierarchy:

- ▼ app.UIFigure
  - app.UIAxes
  - app.DropDown
  - app.DatePicker
  - app.Button
  - app.CheckBox
  - app.EditField
  - app.Image
  - app.Slider
  - app.Spinner
  - app.UITable
  - ▼ app.Tree
    - ▼ app.Node
      - app.Node2
      - app.Node3
      - app.Node4
    - app.TextArea
    - app.Lamp
    - app.Gauge
    - app.Knob

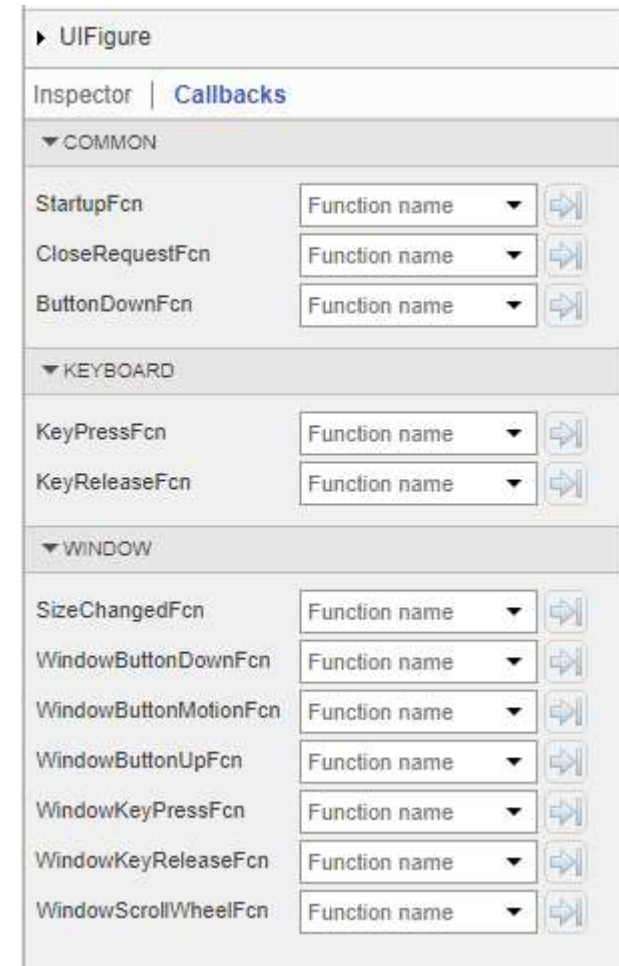
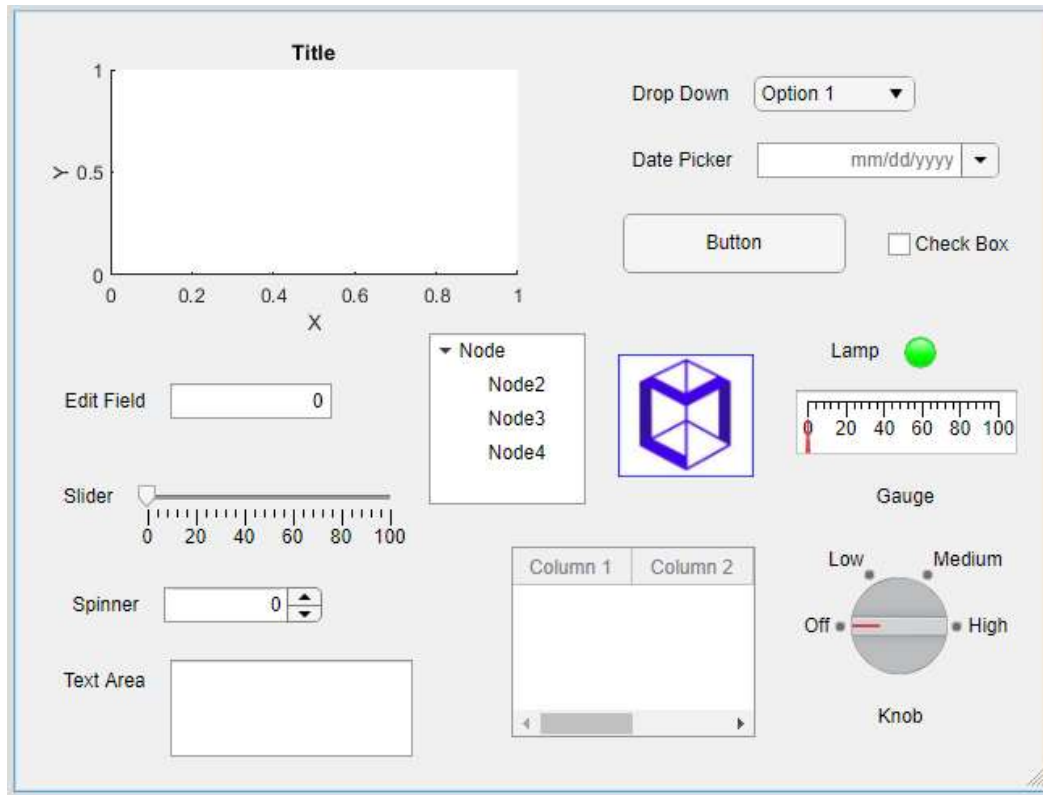
Árvore contendo *handles* para os componentes gráficos

# App Designer – Componentes Gráficos

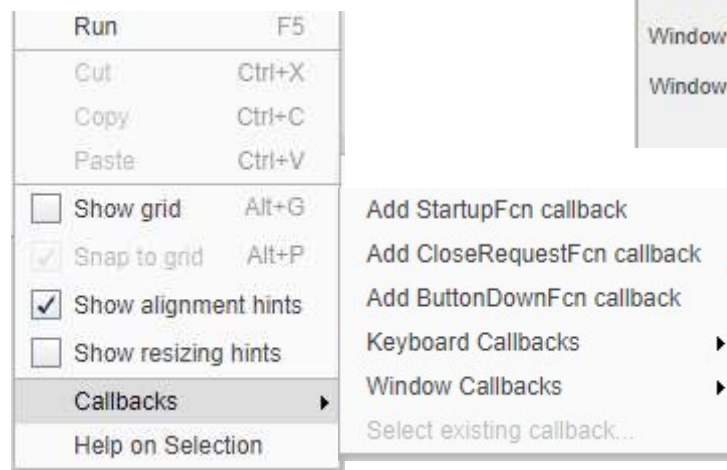
The screenshot displays the App Designer interface with various graphical components and their properties. On the left, a plot titled "Title" is shown with X and Y axes ranging from 0 to 1. Below the plot are several control components: an Edit Field with the value 0, a Slider with a range from 0 to 100, a Spinner with the value 0, and a Text Area. In the center, there is a "Node" dropdown menu with options Node2, Node3, and Node4, a 3D cube icon, a Date Picker set to mm/dd/yyyy, a Button, and a Check Box. On the right, there is a "Lamp" component (a green circle), a "Gauge" component (a scale from 0 to 100), and a "Knob" component (a circular dial with Low, Medium, High, and Off positions). On the far right, the "Inspector" panel is visible, showing the hierarchy of components (app.UIFigure, app.UIAxes, app.DropDown) and the "Inspector" tab. The Inspector panel lists properties for the selected component, including Title.String (Title), XLabel.String (X), and YLabel.String (Y). Other categories like FONT, TICKS, RULERS, GRIDS, MULTIPLE PLOTS, COLOR AND TRANSPARENCY MAPS, BOX STYLING, POSITION, VIEWING ANGLE, INTERACTIVITY, CALLBACK EXECUTION CONTROL, and PARENT/CHILD are also visible.

É possível inspecionar e modificar propriedades de componentes gráficos, na aba lateral direita.

# App Designer – Componentes Gráficos

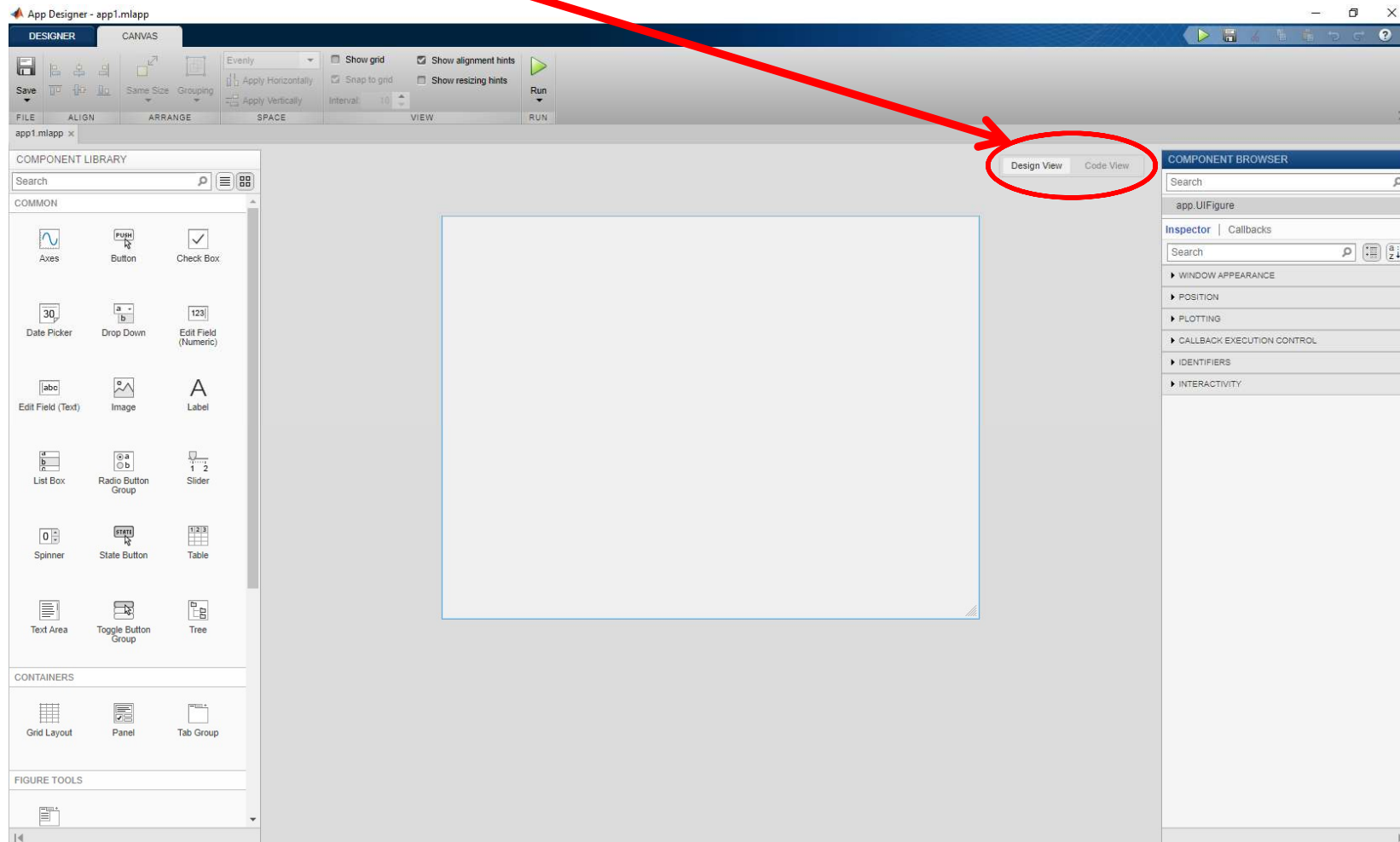


Clique com botão direito na UIFigure abre opções, incluindo de adicionar *callbacks*



# App Designer – Code View

A medida que a interface gráfica é modificada interativamente, o código é atualizado automaticamente. É possível visualizar e modificar partes do código neste mesmo ambiente, selecionando a aba *code view*.





# App Designer – Code View

The screenshot displays the App Designer interface in Code View for a file named `app1.mlapp`. The interface includes a menu bar with options like Save, Callback, Function, Property, App Input Arguments, Find, Indent, Comment, Enable app coding alerts, Show Tips, and Run. Below the menu bar is a toolbar with icons for these actions. The main workspace is divided into three panes:

- CODE BROWSER:** Located on the left, it shows a search bar and a list of components: Callbacks, Functions, and Properties. Below this is the APP LAYOUT pane, which is currently empty.
- Code Editor:** The central pane displays the MATLAB code for the `app1` class. The code defines a class `app1` that inherits from `matlab.apps.AppBase`. It includes properties for `UIFigure`, a `createComponents` function to initialize the UI, and a `delete` function to clean up the app.
- COMPONENT BROWSER:** Located on the right, it shows a search bar and a list of components. The `app.UIFigure` component is selected, and its Inspector pane is visible, showing various properties like WINDOW APPEARANCE, POSITION, PLOTTING, CALLBACK EXECUTION CONTROL, IDENTIFIERS, and INTERACTIVITY.

```
1 classdef app1 < matlab.apps.AppBase
2
3     % Properties that correspond to app components
4     properties (Access = public)
5         UIFigure matlab.ui.Figure
6     end
7
8     % Component initialization
9     methods (Access = private)
10
11         % Create UIFigure and components
12         function createComponents(app)
13
14             % Create UIFigure and hide until all components are created
15             app.UIFigure = uifigure('Visible', 'off');
16             app.UIFigure.Position = [100 100 640 480];
17             app.UIFigure.Name = 'UI Figure';
18
19             % Show the figure after all components are created
20             app.UIFigure.Visible = 'on';
21         end
22     end
23
24     % App creation and deletion
25     methods (Access = public)
26
27         % Construct app
28         function app = app1
29
30             % Create UIFigure and components
31             createComponents(app)
32
33             % Register the app with App Designer
34             registerApp(app, app.UIFigure)
35
36             if nargin == 0
37                 clear app
38             end
39         end
40
41         % Code that executes before app deletion
42         function delete(app)
43
44             % Delete UIFigure when app is deleted
```

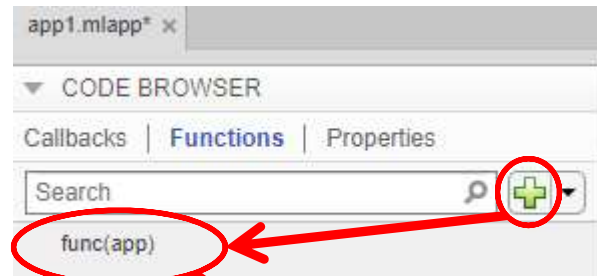


# App Designer – Code View



Na visualização do código, na aba esquerda, é possível adicionar *callbacks*, funções (métodos) e propriedades, públicos e privados, no contexto de POO. Estas modificações são automaticamente implementadas no código.

# App Designer – Code View



Código editável, dentro do ambiente App Designer (Code View)

```
methods (Access = public)

    function results = func(app)
        |
    end
end

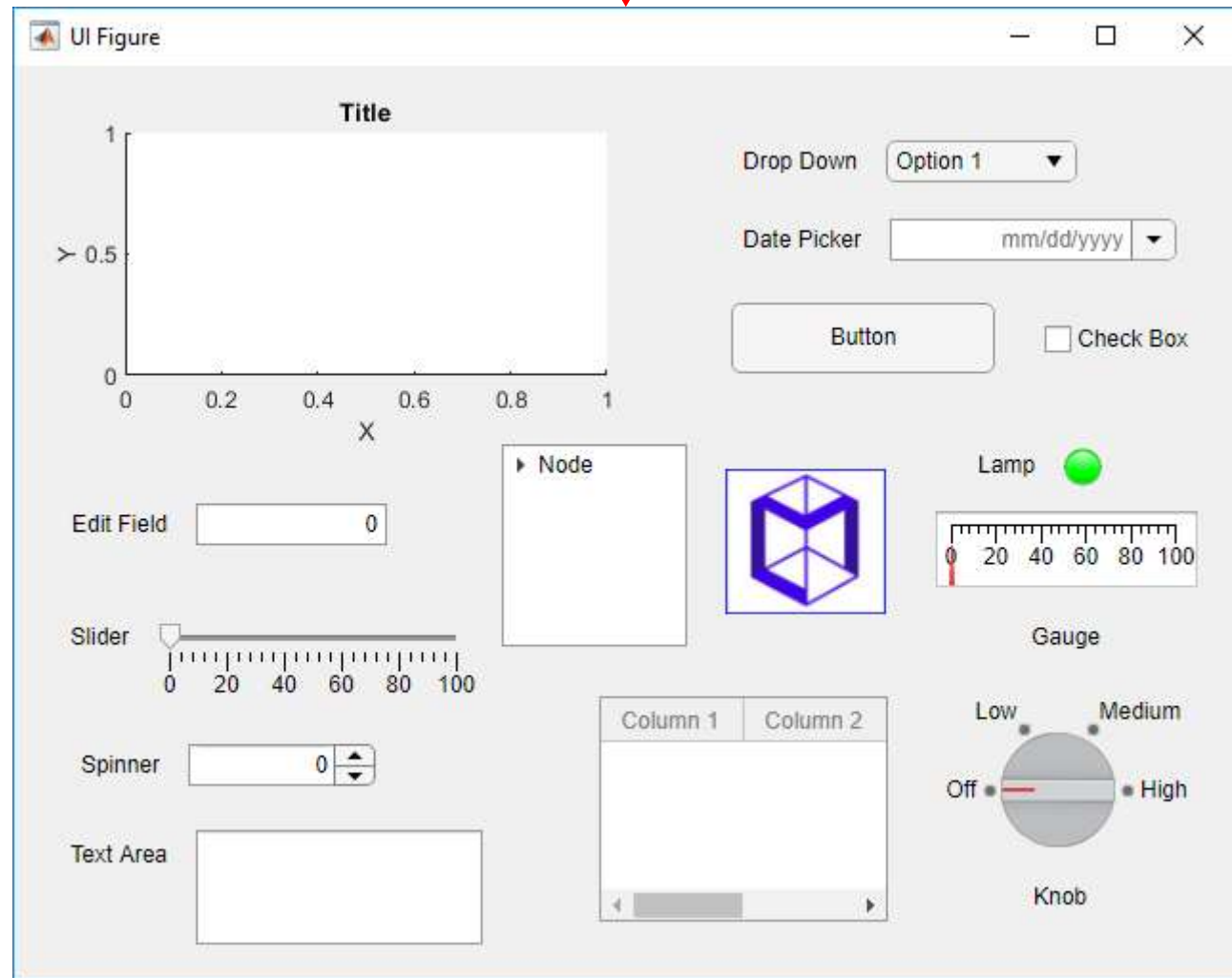
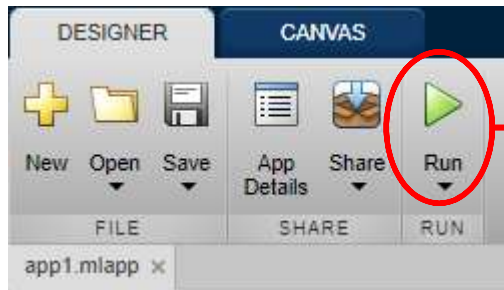
% Component initialization
methods (Access = private)

    % Create UIFigure and components
    function createComponents(app)

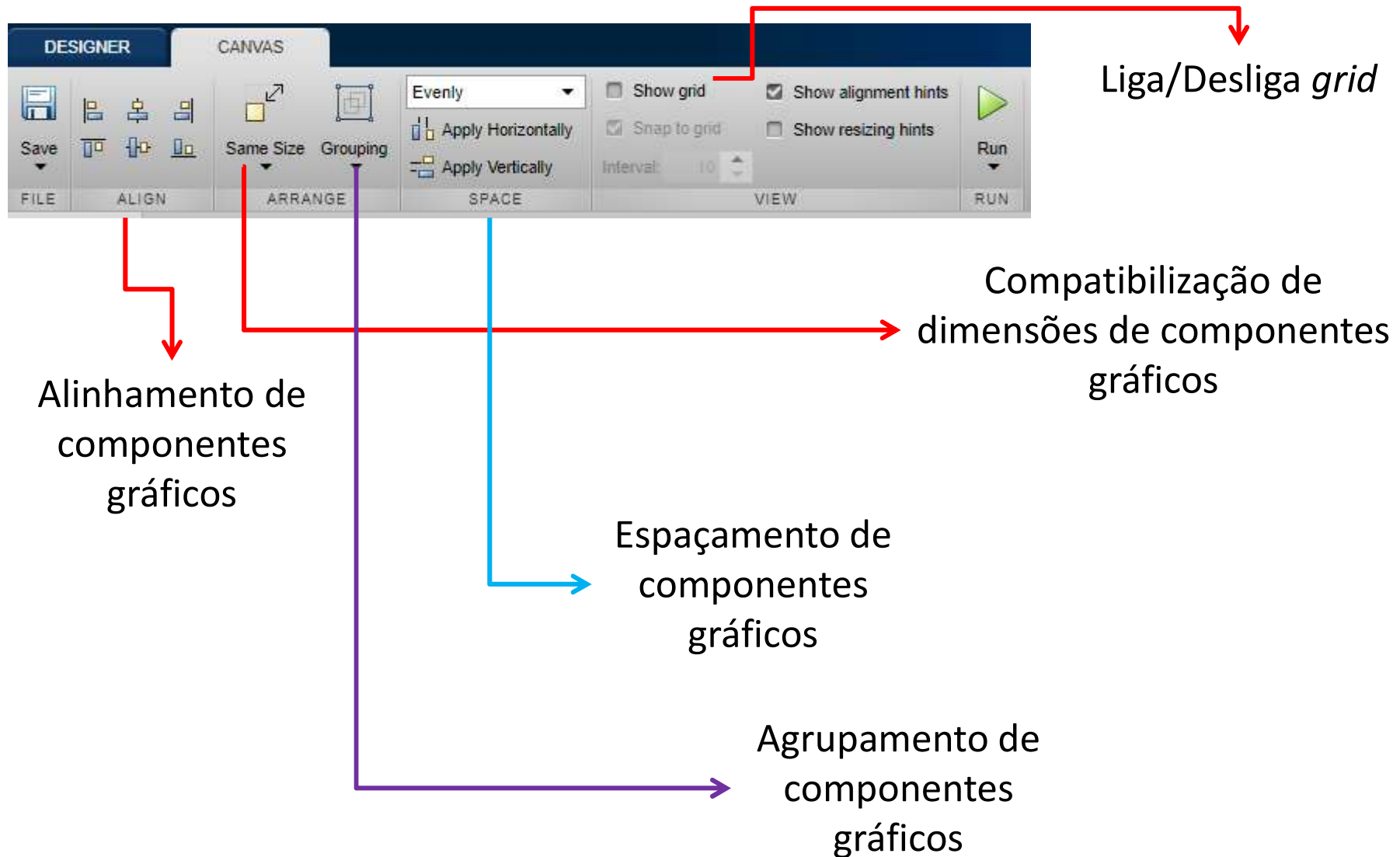
        % Create UIFigure and hide until all components are created
        app.UIFigure = uifigure('Visible', 'off');
        app.UIFigure.Position = [100 100 640 480];
        app.UIFigure.Name = 'UI Figure';

        % Create UIAxes
        app.UIAxes = uiaxes(app.UIFigure);
        title(app.UIAxes, 'Title')
        xlabel(app.UIAxes, 'X')
        ylabel(app.UIAxes, 'Y')
        app.UIAxes.Position = [19 282 300 185];
```

# App Designer



# App Designer – Ferramentas Auxiliares

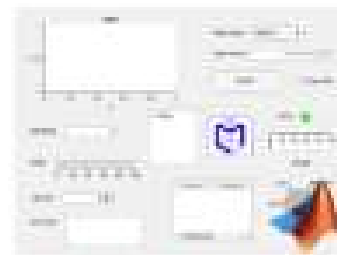


## App Designer – Arquivos

- Um arquivo .mlapp é gerado ao salvar projetos no App Designer.
- Este arquivo pode ser rodado diretamente pelo App Designer, ou chamado por código para inicializar a aplicação gráfica.
- É possível ainda copiar o código, na aba Code View do App Designer, e salvá-lo em um arquivo .m. Desta forma, pode-se trabalhar de forma puramente programática com a interface.



app1.m



app1.mlapp

## Alterando as Propriedades da Janela

Algumas propriedades básicas da janela de interface gráfica (UIFigure):

- **Name:** Define o nome que aparecerá na barra superior da janela.
- **Tag:** Propriedade presente em todos os componentes que define o nome pelo qual estes são identificados no código. É importante sempre modificar essa propriedade para cada componente adicionado para que o código não fique confuso, já que o nome padrão é a numeração na qual os componentes são adicionados.
- **Resize:** Indica se é permitido o redimensionamento da janela.

Descrição de todas as propriedades da janela de interface:

<https://www.mathworks.com/help/matlab/ref/matlab.ui.figureappd-properties.html>

# Componentes Gráficos

Algumas propriedades comuns à diversos tipos diferentes de componentes gráficos:

- **Enable:** Permite tornar um componente desabilitado ou inativo para o usuário. No primeiro caso, o componente fica com uma aparência sombreada, já no segundo continua com a mesma aparência de como se estivesse habilitado porém usuários não são capazes de utiliza-lo.
- **Estilos de Fonte:** Grupo de propriedades presente em qualquer componente que exiba algum texto e que permitem definir o tipo, tamanho, cor e orientação da fonte.
- **Position:** Permite definir as coordenadas X e Y do canto inferior direito do componente em relação à origem da janela de interface ou do painel no qual esteja inserido. Também inclui a possibilidade de alterar a largura e altura do componente. [X Y Largura Altura]

## Componentes Gráficos

- **String:** Propriedade também presente em componentes que exibam algum texto e que permite definir o texto exibido.
- **Tag:** Como já mencionado, essa propriedade é presente em todos os componentes e define o nome pelo qual estes são identificados no código. É importante sempre modificar essa propriedade para cada componente adicionado.
- **Visible:** Propriedade presente em todos os componentes que, se desativada, torna o componente invisível.

Descrição de todas as propriedades dos componentes gráficos:

<https://www.mathworks.com/help/matlab/ref/matlab.ui.figureappd-properties.html>



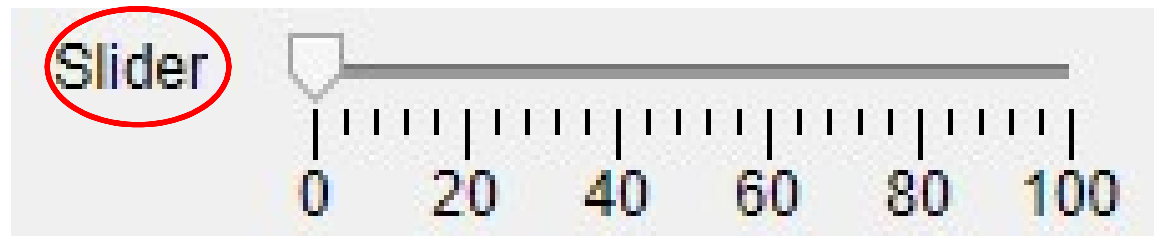
## Componentes Gráficos: Label

Caixa de texto para exibir strings que não pode ser modificada pelo usuário.

Este componente não está associado a nenhuma função *callback*, sendo utilizado apenas para rotular outros elementos e fornecer informações ao usuário.

Algumas vezes a informação exibida deve ser modificada, para isso a propriedade “Text” deve ser acessada em um função *callback* e modificada.

Ex.: *Label* que indica o nome de uma barra de escala (componente do tipo *Slider*)



## Componentes Gráficos: Edit Field (Numeric/Text)

Caixa de texto que pode ser alterada pelo usuário para inserir ou editar valores numéricos/texto (Numeric/Text).

Associado à *callback ValueChangedFcn*. Utilizado comumente para permitir a entrada de dados em programas.

Esses valores fornecidos pelos usuários nas caixas de texto podem ser obtidos acessando a propriedade “Value” do respectivo componente.

A propriedade “Value” só é alterada quando se clica fora da caixa de texto ou a tecla “Enter” é pressionada (“Enter” + Ctrl para caixas multilinha).



## Componentes Gráficos: Push Button

Botão simples para realizar alguma ação quando ativado por um clique do botão esquerdo do mouse.

Esta ação (resposta do programa ao evento de clique do mouse sobre o botão) deve ser programada em uma função callback que é disparada no instante em que se libera o clique no interior do botão.



## Componentes Gráficos: State Button

Semelhante ao *Push Button*, porém possui dois estados: ativado (pressionado) e desativado (não pressionado).

Quando ativado, a propriedade “Value” adquire o valor lógico *true* = 1, e quando desativado adquire o valor *false* = 0.

Para verificar se um *State Button* está ativado ou desativado, basta checar sua propriedade “Value”.



Botão Desativado



Botão Ativado

## Componentes Gráficos: Check Box

Assim como *State Buttons*, possui o estado ativado e desativado, associado a valores lógicos *true* e *false*.

Este componente é utilizado quando se deseja prover ao usuário opções independentes, que podem ter seleção múltipla.

Exemplo de Check Box

Caixa Desativada

Exemplo de Check Box

Caixa Ativada

## Componentes Gráficos: Radio Button Group

Componente similar à *Check Box*, porém a diferença é que este pode ser utilizado para dar opções mutuamente exclusivas ao usuário, ou seja, apenas um *Radio Button* de um grupo pode ser selecionado.

O nome de um *Button Group* pode ser alterado através da propriedade “Title”.

Ex.: Seleção de idioma:



## Componentes Gráficos: Drop Down

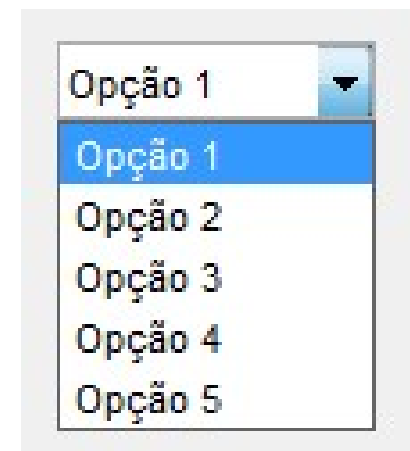
Componente que exibe uma lista de opções quando aberto, e quando fechado indica a opção corrente.

Geralmente é utilizado quando se deseja criar uma lista de opções mutualmente exclusivas, porém não se dispõe de espaço suficiente para apresentar tais opções em um grupo de *Radio Buttons* ou *State Buttons*.

Para editar as opções deve-se modificar a propriedade “Items”, em que cada linha corresponde a uma opção diferente.

Para determinar qual a opção que está selecionada, deve-se acessar a propriedade “Value”.

Uma função *callback* pode ser disparada sempre que a opção for trocada.

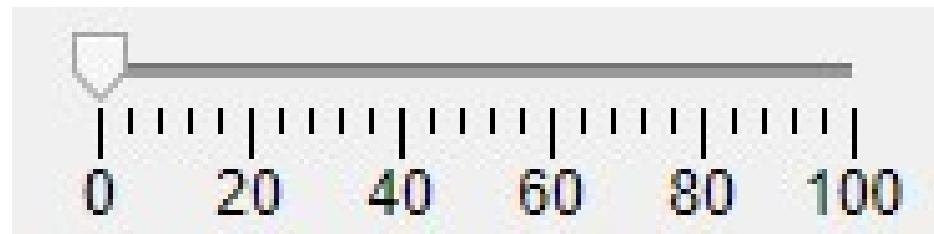


## Componentes Gráficos: Slider

*Sliders* são componentes geralmente utilizados para variar um valor numérico entre um valor mínimo e um máximo.

É atribuída à propriedade “Value” um valor que corresponde a posição do objeto deslizando que é proporcional aos valores mínimo e máximo atribuídos à propriedade “Limits” ([min,max]). Quando o objeto está no extremo esquerdo do *Slider*, a propriedade “Value” assume o valor da propriedade “Limits(1)”, e quando se encontra no extremo direito assume o valor da propriedade “Limits(2)”.

Uma função *callback* é disparada somente quando o objeto deslizando é solto, *ValueChangedFcn*, enquanto outra é chamada durante o movimento do *Slider*, *ValueChangingFcn*.





## **Componentes Gráficos: Table**

Componente para armazenar dados textuais ou numéricos em forma de tabela.

O número e formatação de linhas e colunas pode ser definido editando a propriedade “ColumnFormat”. Os dados podem ser acessados e editados através da propriedade “Data”, na qual é possível fornecer um vetor ou matriz de dados que serão exibidos na tabela.

## **Componentes Gráficos: Panel**

Tem apenas o objetivo de organizar a interface do programa em painéis que agrupam diferentes tipos de componentes.

O nome de um painel pode ser alterado através da propriedade “Title”.

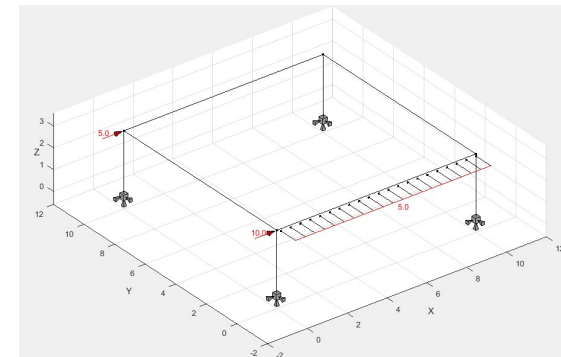
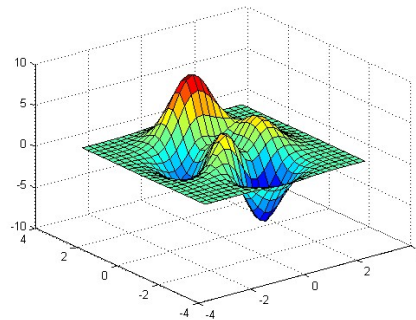
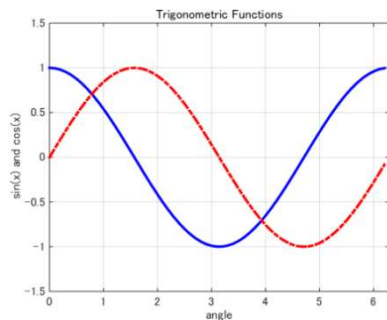
# Componentes Gráficos: UIAxes

Eixos são onde os objetos gráficos são plotados e geralmente são utilizados como o canvas da interface gráfica.

Quando um comando de plotagem é executado, este utiliza o eixo ativo que é o mais recentemente criado ou o último utilizado. Em interfaces com mais de um eixo, deve-se especificar qual eixo é o ativo antes de executar qualquer comando de plotagem para evitar que se utilize o eixo errado.

Os eixos são naturalmente tridimensionais e possuem diversas propriedades e comandos que permitem controlar a visualização dos objetos gráficos.

<https://www.mathworks.com/help/matlab/ref/matlab.ui.control.uiaxes-properties.html>



# Funções Úteis

## ➤ Caixas de Diálogo:

Diferentes tipos de caixas de diálogo predefinidas podem ser exibidas com comandos simples, disponíveis em:

<https://www.mathworks.com/help/matlab/develop-apps-using-the-figure-function.html>

## ➤ Manipulação dos Eixos

Especificar o estilo do eixo: `axis(UmHandleParaUIAxes,'equal');`

Limpar todos os objetos gráficos: `cla (UmHandleParaUIAxes);`

Objetos gráficos não sejam substituídos na próxima plotagem:

`hold(UmHandleParaUIAxes,'on');`

Especificar os limites de um eixo: `UmHandleParaUIAxes.Xlim = [a, b];`

Ativar a grade: `UmHandleParaUIAxes.XGrid = true;`

Definir as coordenadas do centro da câmera: `campos(UmHandleParaUIAxes,x, y, z);`

Obter as coordenadas do centro da câmera: `var = campos;`

Mover a câmera para uma visão 2D ou 3D: `view(UmHandleParaUIAxes,2);` ou `view(UmHandleParaUIAxes,3);`

## Funções Úteis

➤ Manipulação de Arquivos:

Criar um arquivo txt para escrita: `txt = fopen(fullname, 'wt');`


Criar um arquivo txt para leitura: `txt = fopen(fullname, 'rt');`

Abrir um arquivo na tela: `winopen('NomeDoArquivo.extensão');`

Imprimir dados: `fprintf(txt, 'texto ou número %f', valor);`

Ler dados: `fscanf(txt, '%f', NúmeroDeDados);`

# GUIDE vs APP DESIGNER (Fonte: Mathworks)

 *Support for interactive and programmatic app building*

 *Support for programmatic app building only*

Feature	GUIDE	App Designer
Component Library		
Property Inspector		
Component Browser		
Callback Sharing		
App Templates		
Single-File Management for Code and User Interface		
Modern Look & Feel		
Toolstrip-Based Desktop		
Rich Canvas Interactions		
Embedded Code Editor		
Rich Property Editors		
App Metadata Management		
Robust Programming Editor and Workflow		
Quick Start Interactive Tutorial		
App Testing Framework Support		

# GUIDE vs APP DESIGNER (Fonte: Mathworks)

✔ Support for interactive and programmatic app building

✔ Support for programmatic app building only


## Code Editor

Feature	GUIDE	App Designer
Code Debugger	✔	✔
Code Folding	✔	✔
Code Analyzer	✔	✔
App Input Arguments	✔	✔
Rich Programming Aids		✔
Quick Code Navigation		✔
Global Component Rename		✔
Utility Functions via Custom App Methods		✔


## Graphics Support

Feature	GUIDE	App Designer
2D & 3D Plotting	✔	✔
Axes Support (axes, polaraxes, geoaxes)	✔	✔
Axes Interactions (pan, zoom, rotate)	✔	✔
Figure Interactions (keyboard and mouse events)	✔	✔
Printing Support	✔	

# GUIDE vs APP DESIGNER (Fonte: Mathworks)

 Support for interactive and programmatic app building

1. Requires MATLAB Compiler.

 Support for programmatic app building only

2. MATLAB Online is available with select licenses.

## App Sharing & Access

Feature	GUIDE	App Designer
Share as MATLAB App Install File		
Share as MATLAB Standalone Desktop App <sup>1</sup>		
Share as MATLAB Web App <sup>1</sup>		
Create, Edit, and Run Apps in MATLAB Online <sup>2</sup>		
Export as Standalone MATLAB Class File		

## Containers & Layout Managers

Feature	GUIDE	App Designer
Panel		
Menu Bar		
Tab Group		
Scrollable Containers		
Grid Layout Manager		
Auto-Reflow		
Context Menu		
Toolbar		

# GUIDE vs APP DESIGNER (Fonte: Mathworks)

✔ Support for interactive and programmatic app building

✔ Support for programmatic app building only

## Common Components

Feature	GUIDE	App Designer
Button	✔	✔
Drop Down	✔	✔
Edit Field (Text)	✔	✔
Label	✔	✔
List Box	✔	✔
Radio Button Group	✔	✔
Slider	✔	✔
Table	✔	✔
Text Area	✔	✔
Toggle Button Group	✔	✔
Date Picker		✔
Edit Field (Numeric)		✔
Image		✔
Spinner		✔
State Button		✔
Tree		✔



# GUIDE vs APP DESIGNER (Fonte: Mathworks)

- ✔ Support for interactive and programmatic app building
- ✔ Support for programmatic app building only

3. Requires Aerospace Toolbox.

## Instrumentation

Feature	GUIDE	App Designer
Gauge		✔
90 Degree Gauge		✔
Linear Gauge		✔
Semicircular Gauge		✔
Knob		✔
Discrete Knob		✔
Lamp		✔
Switch		✔
Rocker Switch		✔
Toggle Switch		✔
Airspeed Indicator <sup>3</sup>		✔
Altimeter <sup>3</sup>		✔
Climb Rate Indicator <sup>3</sup>		✔
EGT Indicator <sup>3</sup>		✔
Heading Indicator <sup>3</sup>		✔
Artificial Horizon <sup>3</sup>		✔
RPM Indicator <sup>3</sup>		✔
Turn Indicator <sup>3</sup>		✔

# GUIDE vs APP DESIGNER (Fonte: Mathworks)

- ✔ Support for interactive and programmatic app building
- ✔ Support for programmatic app building only

## Dialogs

Feature	GUIDE	App Designer
Alert Dialog	✔	✔
Confirmation Dialog	✔	✔

## Component Extensibility

Feature	GUIDE	App Designer
ActiveX Control	✔	