

# **Desenvolvimento de Sistemas Orientados a Objetos I**

**Interface Gráfica**

---

Jean Carlo Rossa Hauck, Dr.

[jean.hauck@ufsc.br](mailto:jean.hauck@ufsc.br)

<http://www.inf.ufsc.br/~jeanhauck>

# Conteúdo Programático

- Técnicas de uso comum em sistemas orientados a objetos
  - Interface gráfica com o usuário



# Interface Gráfica com Usuário - *GUI*

A interface gráfica com o usuário (GUI – *Graphical User Interface*) fornece um conjunto de componentes facilitando primeiramente a utilização de determinada aplicação;

O browser é um exemplo de GUI. Na janela, há uma barra de menus que contém itens de menus específicos oferecendo funcionalidades aos usuários;

As GUIs são construídas a partir de componentes GUI;

Um componente GUI é um objeto com o qual o usuário interage através do mouse, do teclado ou de outra forma de entrada, como o reconhecimento de voz



# Interface Gráfica com Usuário em Python

Existem diversas bibliotecas que dão suporte a GUI em Python:

- Tkinter
- PyQt
- PyGTK
- PyGUI
- ✓ **PySimpleGUI**

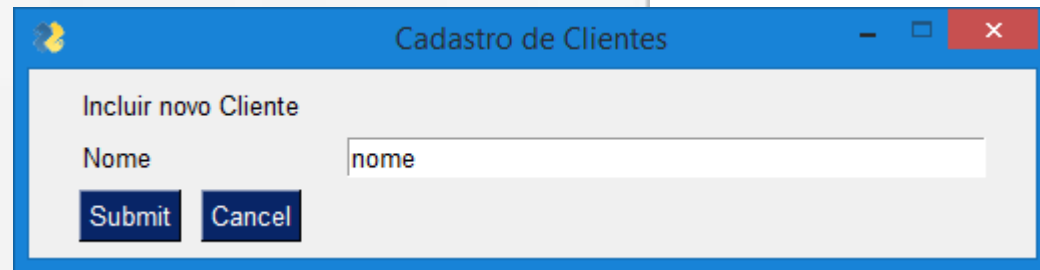
# Primeiro Exemplo

```
import PySimpleGUI as sg

layout = [
    [sg.Text('Incluir novo Cliente')],
    [sg.Text('Nome', size=(15,1)), sg.InputText('nome')],
    [sg.Submit(), sg.Cancel()]
]

window = sg.Window('Cadastro de Clientes').Layout(layout)

button, values = window.Read()
```



# Primeiro Exemplo

**Necessário instalar:**

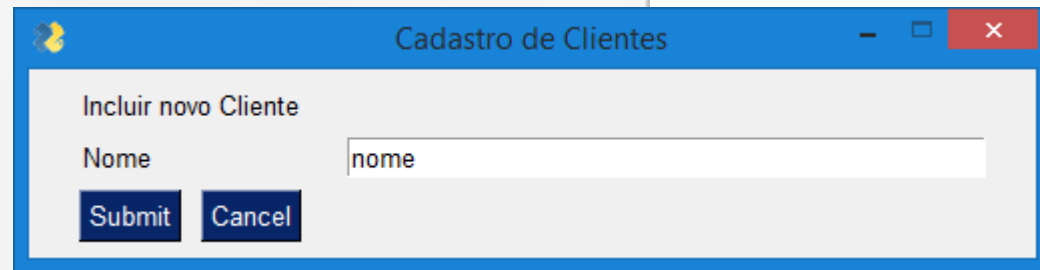
> pip install PySimpleGUI

```
import PySimpleGUI as sg

layout = [
    [sg.Text('Incluir novo Cliente')],
    [sg.Text('Nome', size=(15,1)), sg.InputText('nome')],
    [sg.Submit(), sg.Cancel()]
]

window = sg.Window('Cadastro de Clientes').Layout(layout)

button, values = window.Read()
```



# Como criar uma tela com componentes visuais

1. Criar uma classe que representa a Tela
2. Definir na classe um atributo `__window`
3. Definir layout com as “linhas” da tela
4. Adicionar componentes visuais às linhas da tela
5. Adicionar layout ao `__window`
6. Tratar retorno da Tela no Controlador

<https://github.com/PySimpleGUI/PySimpleGUI/blob/master/docs/tutorial.md>

ExemploView
- __window
+ __init__()
+ init_components()
+ open()
+ close()
+ show_message(titulo, mensagem)



# Um exemplo completo (1)

```
import PySimpleGUI as sg
```

```
class ExemploView:
```

```
    def __init__(self):  
        self.__window = None  
        self.init_components()
```

ExemploView
- __window
+ __init__()
+ init_components()
+ open()
+ close()
+ show_message(titulo, mensagem)

# Um exemplo completo (2)

ExemploView
- __window
+ __init__()
+ init_components()
+ open()
+ close()
+ show_message(titulo, mensagem)

```
def init_components(self):
    sg.ChangeLookAndFeel('Reddit')
    colunal = [[sg.Text('Coluna 1', background_color='#d3dfda', justification='center', size=(10, 1)),
                [sg.Spin(values=('1', '2', '3'), initial_value='selecione', key='sp_spin1')],
                [sg.Spin(values=('1', '2', '3'), initial_value='selecione', key='sp_spin2')],
                [sg.Spin(values=('1', '2', '3'), initial_value='selecione', key='sp_spin3')]]

    layout = [[sg.Text('Todos os componentes em uma tela!', size=(30, 1), font=("Helvetica", 25))],
               [sg.Text('Este e um texto e espaço para responder:')],
               [sg.InputText('Texto de resposta', key='it_nome')],
               [sg.Checkbox('Checkbox opção 1!', key='ck_opcao1'), sg.Checkbox('Checkbox opção 2!', default=True,
                                     key='ck_opcao2')],
               [sg.Radio('Radio opção 1!', "RD1", default=True, key='rd_opcao1'), sg.Radio('Radio opção 2!',
                                     "RD1", key='rd_opcao2')],
               [sg.Multiline(default_text='Texto padrão para o caso do usuário não digitar nada', size=(35, 3)),
                sg.Multiline(default_text='Outra coluna com multiplas linhas', size=(35, 3))],
               [sg.InputCombo(('Combobox 1', 'Combobox 2'), size=(20, 3), key='cb_opcoes'),
                sg.Slider(range=(1, 100), orientation='h', size=(34, 20), default_value=85, key='sl_slider1')],
               [sg.ListBox(values=('Listbox 1', 'Listbox 2', 'Listbox 3'), size=(30, 3), key='lb_itens'),
                sg.Slider(range=(1, 100), orientation='v', size=(5, 20), default_value=25, key='sl_slider2'),
                sg.Slider(range=(1, 100), orientation='v', size=(5, 20), default_value=75, key='sl_slider3'),
                sg.Slider(range=(1, 100), orientation='v', size=(5, 20), default_value=10, key='sl_slider4'),
                sg.Column(colunal, background_color='#d3dfda')],
               [sg.Text('_' * 80)], # linha divisoria
               [sg.Text('Selecione uma pasta', size=(35, 1))],
               [sg.Text('Pasta selecionada', size=(15, 1), auto_size_text=False, justification='right'),
                sg.InputText('Pasta Default'), sg.FolderBrowse(key='fb_pasta_selecionada')],
               [sg.Button('Gravar'), sg.Cancel('Cancelar')]]

    self.__window = sg.Window('Título da tela', default_element_size=(40, 1)).Layout(layout)
```



# Um exemplo completo (2)

ExemploView
- __window
+ __init__()
+ init_components()
+ open()
+ close()
+ show_message(titulo, mensagem)

```
def init_components(self):
    sg.ChangeLookAndFeel('Reddit')
    colunal = [[sg.Text('Coluna 1', background_color='#d3dfda', justification='center', size=(10, 1))],
               [sg.Spin(values=('1', '2', '3'), initial_value='selecione', key='sp_spin1')],
               [sg.Spin(values=('1', '2', '3'), initial_value='selecione', key='sp_spin2')],
               [sg.Spin(values=('1', '2', '3'), initial_value='selecione', key='sp_spin3')]]

    layout = [[sg.Text('Todos os componentes em uma tela!', size=(40, 1), font=('Helvetica', 25))],
              [sg.Text('Este e um texto e espaço para', size=(35, 1))],
              [sg.InputText('Texto de resposta', key='it_texto')],
              [sg.Checkbox('Checkbox opção 1!', key='ck_opcao1'),
               sg.Checkbox('Checkbox opção 2!', key='ck_opcao2')],
              [sg.Radio('Radio opção 1!', "RD1", default=True, key='rd_opcao1'), sg.Radio('Radio opção 2!',
               "RD1", key='rd_opcao2')],
              [sg.Multiline(default_text='Texto padrão para o caso do usuário não digitar nada', size=(35, 3)),
               sg.Multiline(default_text='Outra coluna com multiplas linhas', size=(35, 3))],
              [sg.InputCombo(('Combobox 1', 'Combobox 2'), size=(20, 3), key='cb_opcoes'),
               sg.Slider(range=(1, 100), orientation='h', size=(34, 20), default_value=85, key='sl_slider1')],
              [sg.ListBox(values=('Listbox 1', 'Listbox 2', 'Listbox 3'), size=(30, 3), key='lb_itens'),
               sg.Slider(range=(1, 100), orientation='v', size=(5, 20), default_value=25, key='sl_slider2'),
               sg.Slider(range=(1, 100), orientation='v', size=(5, 20), default_value=75, key='sl_slider3'),
               sg.Slider(range=(1, 100), orientation='v', size=(5, 20), default_value=10, key='sl_slider4'),
               sg.Column(colunal, background_color='#d3dfda')],
              [sg.Text('_' * 80)], # linha divisoria
              [sg.Text('Selecione uma pasta', size=(35, 1))],
              [sg.Text('Pasta selecionada', size=(15, 1), auto_size_text=False, justification='right'),
               sg.InputText('Pasta Default'), sg.FolderBrowse(key='fb_pasta_selecionada')],
              [sg.Button('Gravar'), sg.Cancel('Cancelar')]]

    self.__window = sg.Window('Título da tela', default_element_size=(40, 1)).Layout(layout)
```

Uma coluna com 4 linhas de componentes visuais (sendo 3 spins)

# Um exemplo completo (2)

ExemploView
- __window
+ __init__()
+ init_components()
+ open()
+ close()
+ show_message(titulo, mensagem)

Layout principal com diversos exemplos de componentes visuais

```
def init_components(self):
    sg.ChangeLookAndFeel('Reddit')
```

```
        justification='center', size=(10, 1)],
        'Selecione', key='sp_spin1']],
        'Selecione', key='sp_spin2']],
        'Selecione', key='sp_spin3']]]
```

```
layout = [[sg.Text('Todos os componentes em uma tela!', size=(30, 1), font=("Helvetica", 25))],
           [sg.Text('Este e um texto e espaço para responder:')],
           [sg.InputText('Texto de resposta', key='it_nome')],
           [sg.Checkbox('Checkbox opção 1!', key='ck_opcao1'), sg.Checkbox('Checkbox opção 2!', default=True,
                                   key='ck_opcao2')],
           [sg.Radio('Radio opção 1!', "RD1", default=True, key='rd_opcao1'), sg.Radio('Radio opção 2!',
                                   "RD1", key='rd_opcao2')],
           [sg.Multiline(default_text='Texto padrão para o caso do usuário não digitar nada', size=(35, 3)),
            sg.Multiline(default_text='Outra coluna com multiplas linhas', size=(35, 3))],
           [sg.InputCombo(('Combobox 1', 'Combobox 2'), size=(20, 3), key='cb_opcoes'),
            sg.Slider(range=(1, 100), orientation='h', size=(34, 20), default_value=85, key='sl_slider1')],
           [sg.Listbox(values=('Listbox 1', 'Listbox 2', 'Listbox 3'), size=(30, 3), key='lb_itens'),
            sg.Slider(range=(1, 100), orientation='v', size=(5, 20), default_value=25, key='sl_slider2'),
            sg.Slider(range=(1, 100), orientation='v', size=(5, 20), default_value=75, key='sl_slider3'),
            sg.Slider(range=(1, 100), orientation='v', size=(5, 20), default_value=10, key='sl_slider4'),
            sg.Column(columal, background_color='#d3dfda')],
           [sg.Text('_' * 80)], # linha divisoria
           [sg.Text('Selecione uma pasta', size=(35, 1))],
           [sg.Text('Pasta selecionada', size=(15, 1), auto_size_text=False, justification='right'),
            sg.InputText('Pasta Default'), sg.FolderBrowse(key='fb_pasta_selecionada')],
           [sg.Button('Gravar'), sg.Cancel('Cancelar')]]
```

```
self.__window = sg.Window('Título da tela', default_element_size=(40, 1)).Layout(layout)
```



# Um exemplo completo (2)

ExemploView
- __window
+ __init__()
+ init_components()
+ open()
+ close()
+ show_message(titulo, mensagem)

```
def init_components(self):
    sg.ChangeLookAndFeel('Reddit')
    colunal = [[sg.Text('Coluna 1', background_color='#d3dfda', justification='center', size=(10, 1))],
               [sg.Spin(values=('1', '2', '3'), initial_value='selecione', key='sp_spin1')],
               [sg.Spin(values=('1', '2', '3'), initial_value='selecione', key='sp_spin2')],
               [sg.Spin(values=('1', '2', '3'), initial_value='selecione', key='sp_spin3')]]

    layout = [[sg.Text('Todos os componentes em uma tela!', size=(30, 1), font=("Helvetica", 25))],
              [sg.Text('Este e um texto e espaço para responder:')],
              [sg.InputText('Texto de resposta', key='it_nome')],
              [sg.Checkbox('Checkbox opção 1!', key='ck_opcao1'), sg.Checkbox('Checkbox opção 2!', default=True,
                                     key='ck_opcao2')],
              [sg.Radio('Radio opção 1!', "RD1", default=True, key='rd_opcao1'), sg.Radio('Radio opção 2!',
                                     "RD1", key='rd_opcao2')],
              [sg.Multiline(default_text='Texto padrão para o caso do usuário não digitar nada', size=(35, 3)),
               sg.Multiline(default_text='Outra coluna com multiplas linhas', size=(35, 3))],
              [sg.InputCombo(('Combobox 1', 'Combobox 2'), key='ic_combo1'), sg.Slider(range=(1, 100), key='sl_slider1')],
              [sg.Listbox(values=('Lista 1', 'Lista 2', 'Lista 3'), key='lb_lista1'), sg.Slider(range=(1, 100), key='sl_slider2')],
              [sg.Slider(range=(1, 100), orientation='h', size=(5, 20), default_value=75, key='sl_slider3'),
               sg.Slider(range=(1, 100), orientation='v', size=(5, 20), default_value=10, key='sl_slider4')],
              sg.Column(colunal, background_color='#d3dfda'),
              [sg.Text('_' * 80)], # linha divisoria
              [sg.Text('Selecione uma pasta', size=(35, 1))],
              [sg.Text('Pasta selecionada', size=(15, 1), auto_size_text=False, justification='right'),
               sg.InputText('Pasta Default'), sg.FolderBrowse(key='fb_pasta_selecionada')],
              [sg.Button('Gravar'), sg.Cancel('Cancelar')]]

    self.__window = sg.Window('Título da tela', default_element_size=(40, 1)).Layout(layout)
```

Adicionando a coluna definida anteriormente

# Um exemplo completo (2)

ExemploView
- __window
+ __init__()
+ init_components()
+ open()
+ close()
+ show_message(titulo, mensagem)

```
def init_components(self):
    sg.ChangeLookAndFeel('Reddit')
    colunal = [[sg.Text('Coluna 1', background_color='#d3dfda', justification='center', size=(10, 1))],
                [sg.Spin(values=('1', '2', '3'), initial_value='selecione', key='sp_spin1')],
                [sg.Spin(values=('1', '2', '3'), initial_value='selecione', key='sp_spin2')],
                [sg.Spin(values=('1', '2', '3'), initial_value='selecione', key='sp_spin3')]]

    layout = [[sg.Text('Todos os componentes em uma tela!', size=(30, 1), font=("Helvetica", 25))],
              [sg.Text('Este e um texto e espaço para responder:')],
              [sg.InputText('Texto de resposta', key='it_nome')],
              [sg.Checkbox('Checkbox opção 1!', key='ck_opcao1'), sg.Checkbox('Checkbox opção 2!', default=True,
                                     key='ck_opcao2')],
              [sg.Radio('Radio opção 1!', "RD1", default=True, key='rd_opcao1'), sg.Radio('Radio opção 2!',
                                     "RD1", key='rd_opcao2')],
              [sg.Multiline(default_text='Texto padrão para o caso do usuário não digitar nada', size=(35, 3)),
               sg.Multiline(default_text='Outra coluna com multiplas linhas', size=(35, 3))],
              [sg.InputCombo(('Combobox 1', 'Combobox 2'), size=(20, 3), key='cb_opcoes'),
               sg.Slider(range=(1, 100), orientation='h', size=(34, 20), default_value=85, key='sl_slider1')],
              [sg.ListBox(values=('Listbox 1', 'Listbox 2', 'Listbox 3'), size=(30, 3), key='lb_itens'),
               sg.Slider(range=(1, 100), orientation='v', size=(5, 20), default_value=25, key='sl_slider2'),
               sg.Slider(range=(1, 100), orientation='v', size=(5, 20), default_value=75, key='sl_slider3'),
               sg.Slider(range=(1, 100), orientation='v', size=(5, 20), default_value=10, key='sl_slider4'),
               sg.Column(colunal, background_color='#d3dfda')],
              [sg.Text('_' * 80)], # linha de separação
              [sg.Text('Selecione uma pasta', size=(20, 1))],
              [sg.Text('Pasta selecionada', size=(20, 1))],
              [sg.InputText('Pasta Default', key='it_pasta_default')],
              [sg.Button('Gravar'), sg.Cancel('Cancelar')]]

    self.__window = sg.Window('Título da tela', default_element_size=(40, 1)).Layout(layout)
```

Instanciando a Window, com o título, layout e demais parâmetros

# Um exemplo completo (3)

ExemploView
- __window
+ __init__()
+ init_components()
+ open()
+ close()
+ show_message(titulo, mensagem)

```
def open(self):
    button, values = self.__window.Read()
    return button, values

def close(self):
    self.__window.Close()

def show_message(self, titulo: str, mensagem: str):
    sg.Popup(titulo, mensagem)
```



# Um exemplo completo (3)

ExemploView
- __window
+ __init__()
+ init_components()
+ open()
+ close()
+ show_message(titulo, mensagem)

```
def open(self):  
    button, values = self.__window.Read()  
    return button, values  
  
def close(self):  
    self.__window.Close()  
  
def show_message(self, titulo: str, mensagem:  
    sg.Popup(titulo, mensagem)
```

Abre/exibe a tela.  
Retorna o botão clicado e um  
dicionário com os valores dos  
componentes da tela

# Um exemplo completo (3)

ExemploView
- __window
+ __init__()
+ init_components()
+ open()
+ close()
+ show_message(titulo, mensagem)

```
def open(self):  
    button, values = self.__window.Read()  
    return button, values
```

```
def close(self):  
    self.__window.Close()
```

```
def show_message(self, titulo, mensagem):  
    sg.Popup(titulo, mensagem)
```

Fecha/oculta a tela

# Um exemplo completo (3)

ExemploView
- __window
+ __init__()
+ init_components()
+ open()
+ close()
+ show_message(titulo, mensagem)

```
def open(self):  
    button, values = self.__window.Read()  
    return button, values  
  
def close(self):  
    self.__window.Close()  
  
def show_message(self, titulo: str, mensagem: str):  
    sg.Popup(titulo, mensagem)
```

Exibe uma tela simples de mensagem. Bom para dar feedback para o usuário

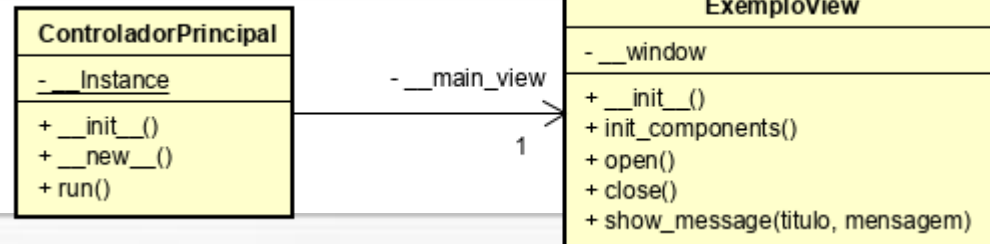
- 
- The diagram illustrates the user flow for a mobile application, with screens and transitions annotated in red:
- Home Screen:** Features a list of items. A red arrow points to the word **HOMES** in red.
  - Login Screen:** Contains fields for "E-mail" and "Senha", and buttons for "Entrar" and "Esqueci minha senha?". A red arrow points to the word **HOME** in red.
  - MAQA Screen:** Displays a map. A red arrow points to the word **MAQA** in red.
  - Bilanci Comparatif Screen:** A screen for comparing balances, with a red arrow pointing to the text "Bilanci Comparatif Voir no G maps".
  - Detail Screens:** Multiple screens showing details of items, with red arrows pointing to the word **HOMES** in red.
  - Add Local Screen:** A screen for adding a new location, with a red arrow pointing to the text "Add Local".
  - Settings/About Screen:** A screen for settings or about information, with a red arrow pointing to the text "Add Local".

# O Padrão Singleton

- **Nome:** Singleton
- **Classificação:** Creational
- **Problema**
  - Não devem ser permitidas duas instâncias da mesma classe no sistema
- **Solução**
  - Controlar o acesso ao construtor e controlar a criação de instâncias
- **Consequências**
  - Somente há um único ponto de acesso global à instância



# Controlador da Aplicação



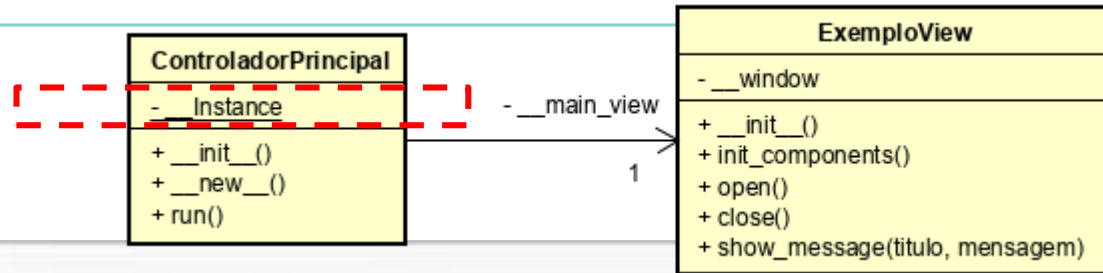
```
class ControladorPrincipal:
    __instance = None

    def __init__(self):
        self.__main_view = ExemploView()

    def __new__(cls):
        if ControladorPrincipal.__instance is None:
            ControladorPrincipal.__instance = object.__new__(cls)
        return ControladorPrincipal.__instance

    def run(self):
        (botao, dados) = self.__main_view.open()
        ...
```

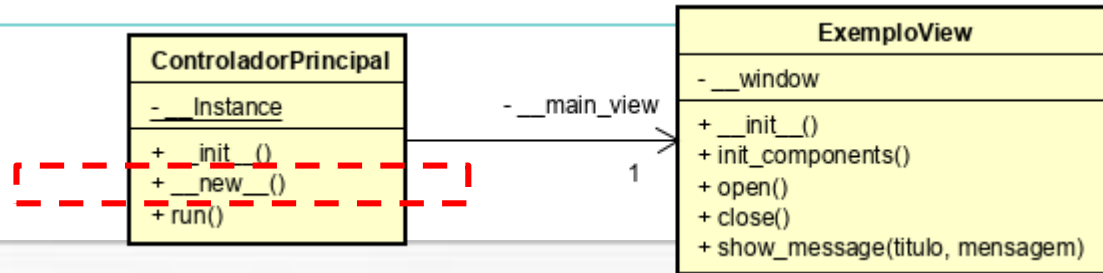
# Controlador da Aplicação



```
class ControladorPrincipal:
    | __instance = None
    |
    | def __init__(self):
    |     self.__main_view = ExemploView
    |
    | def __new__(cls):
    |     if ControladorPrincipal.__instance is None:
    |         ControladorPrincipal.__instance = object.__new__(cls)
    |     return ControladorPrincipal.__instance
    |
    | def run(self):
    |     (botao, dados) = self.__main_view.open()
    |     ...
```

Atributo de classe guarda a instância única

# Controlador da Aplicação



```
class ControladorPrincipal:
```

```
    __instance = None
```

```
    def __init__(self):
```

```
        self.__main_view = ExemploView()
```

```
    def __new__(cls):
```

```
        if ControladorPrincipal.__instance is None:
```

```
            ControladorPrincipal.__instance = object.__new__(cls)
```

```
        return ControladorPrincipal.__instance
```

```
    def run(self):
```

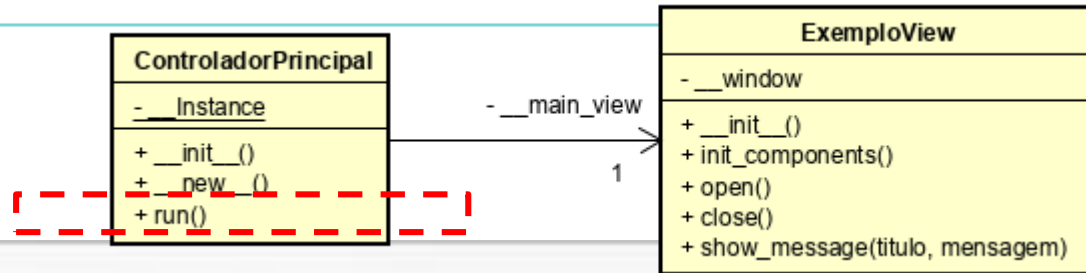
```
        (botao, dados) = self.__main_view.open()
```

```
        ...
```

Controla a criação de instâncias e garante **uma única instância**



# Controlador da Aplicação



```
class ControladorPrincipal:
    __instance = None

    def __init__(self):
        self.__main_view = ExemploView()

    def __new__(cls):
        if ControladorPrincipal.__instance is None:
            ControladorPrincipal.__instance = cls()
        return ControladorPrincipal.__instance

    def run(self):
        (botao, dados) = self.__main_view.open()
```

Inicia o sistema e abre a tela inicial

**Agora:**

- 1. Desenhe os protótipos de tela**
- 2. Implemente uma tela de cadastro no sistema do trabalho**

**Preparado  
para sofrer  
um pouquinho?**





## Atribuição-Uso-Não-Comercial-Compartilhamento pela Licença 2.5 Brasil

### ***Você pode:***

- copiar, distribuir, exhibir e executar a obra
- criar obras derivadas

### ***Sob as seguintes condições:***

**Atribuição** — Você deve dar crédito ao autor original, da forma especificada pelo autor ou licenciante.

**Uso Não-Comercial** — Você não pode utilizar esta obra com finalidades comerciais.

**Compartilhamento pela mesma Licença** — Se você alterar, transformar, ou criar outra obra com base nesta, você somente poderá distribuir a obra resultante sob uma licença idêntica a esta.

Para ver uma cópia desta licença, visite <http://creativecommons.org/licenses/by-nc-sa/2.5/br/> ou mande uma carta para Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.