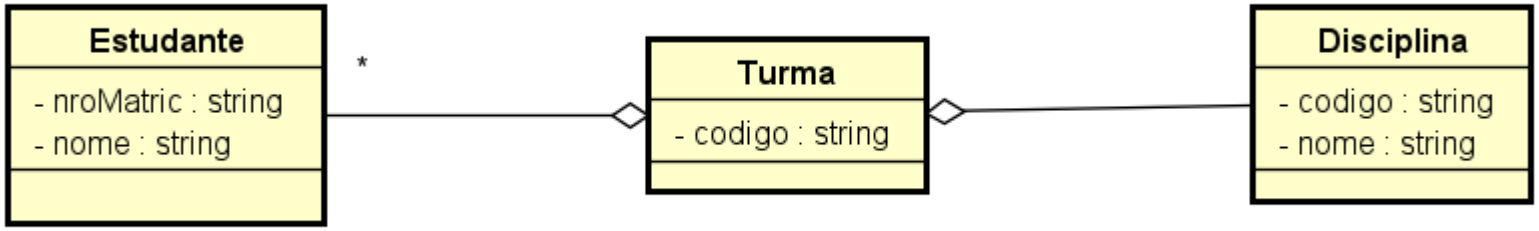
COM220
Computação Orientada a Objetos I

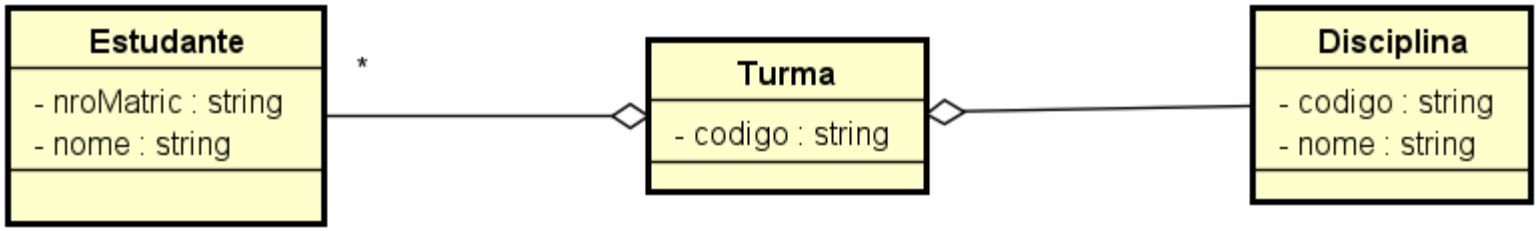
Aula 15: Prática com MVC

MVC

- Considere o modelo a seguir

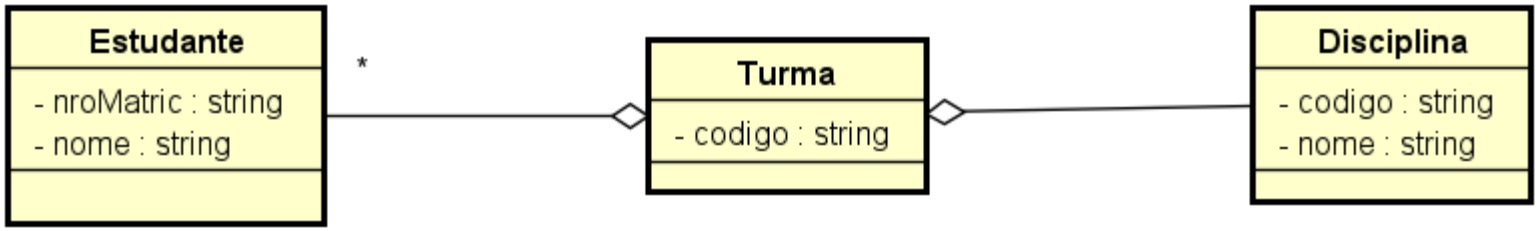


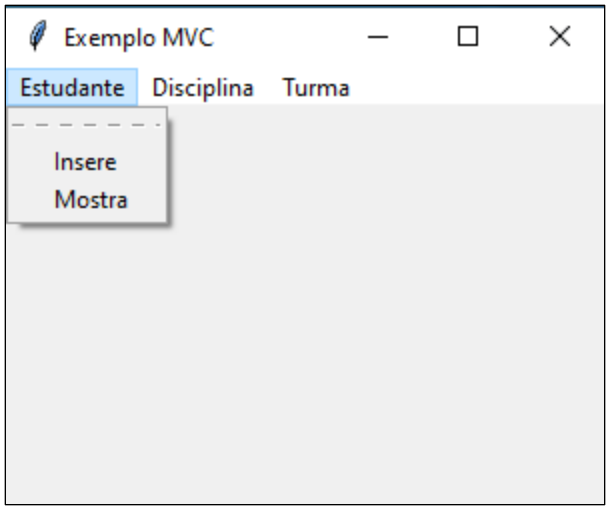
MVC

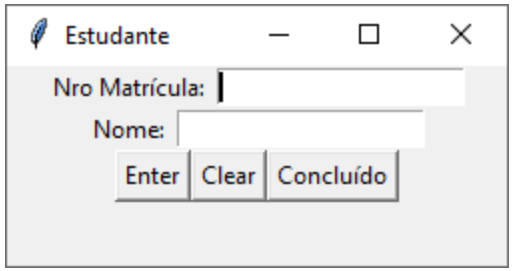


- Queremos desenvolver um sistema no qual seja possível
  - Cadastrar e listar estudantes e disciplinas
  - Criar turmas com as disciplinas e os estudantes cadastrados
  - Listar as turmas criadas

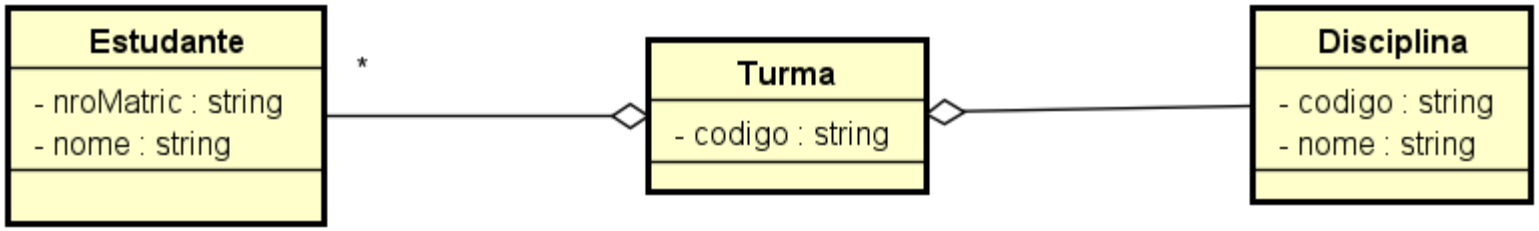
MVC

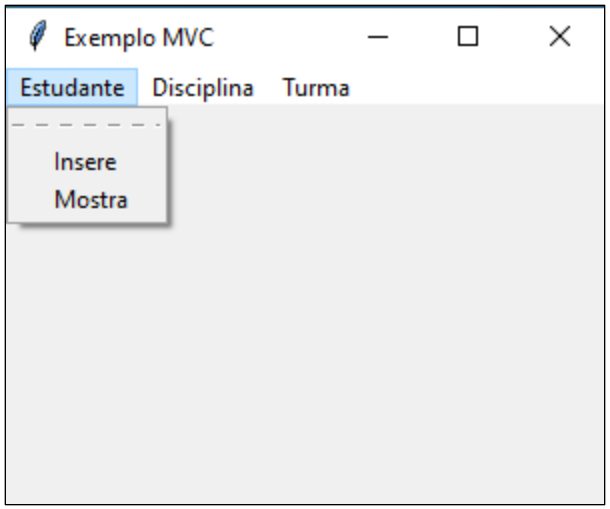

Estudante
- nroMatric : string
- nome : string
*
Turma
- codigo : string
Disciplina
- codigo : string
- nome : string

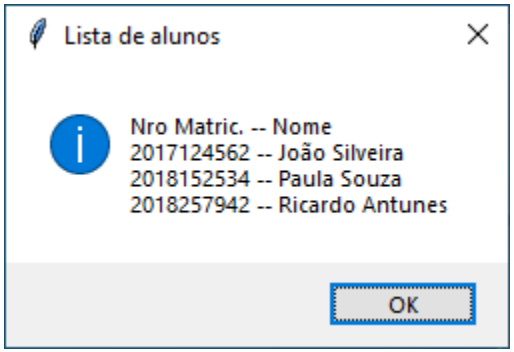

Exemplo MVC
Estudante
Disciplina
Turma
Insere
Mostra


Estudante
Nro Matrícula:
Nome:
Enter
Clear
Concluído

MVC

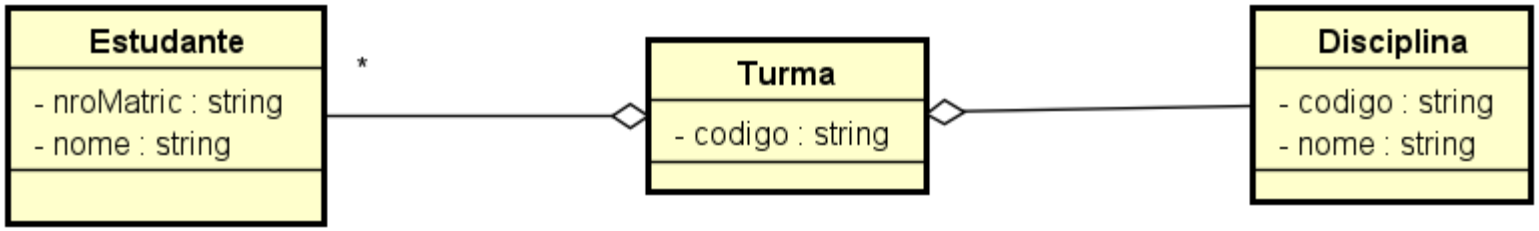

Estudante
- nroMatric : string
- nome : string
*
Turma
- codigo : string
Disciplina
- codigo : string
- nome : string


Exemplo MVC
Estudante
Disciplina
Turma
Insere
Mostra


Lista de alunos
Nro Matric. -- Nome
2017124562 -- João Silveira
2018152534 -- Paula Souza
2018257942 -- Ricardo Antunes
OK

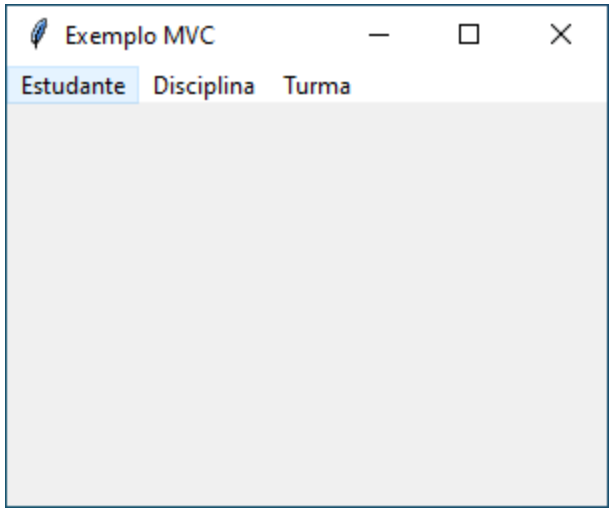
MVC

- Temos 3 classes de entidade, então:
  - Precisamos de no mínimo 3 controladores
  - Temos 2 operações para cada classe de entidade
    - 3 x 2 = 6 classes de limite (ou mais)
- Temos que exibir e controlar um menu, então precisamos de um controlador principal e um limite principal
- 3 + 3 + 6 + 2 = 14 classes (mínimo)



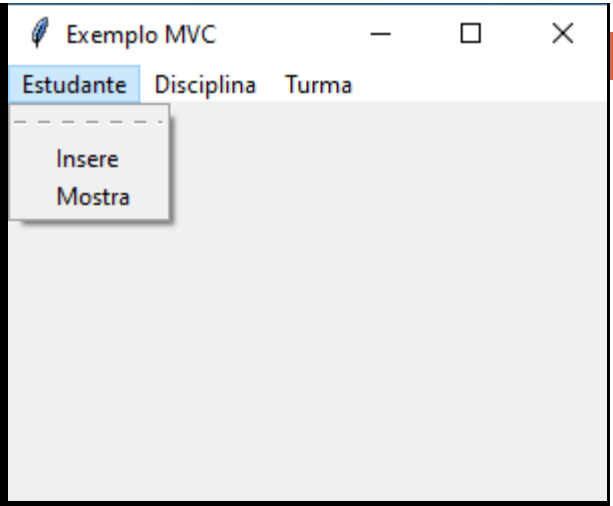
Criando um Menu

- Vamos começar nosso projeto criando um controle principal e um limite principal que exibe um Menu para a seleção das operações
- Tkinter oferece classes que facilitam a criação de diferentes tipos de Menu
- Vamos usar um menu que ficará acoplado à janela principal da nossa aplicação



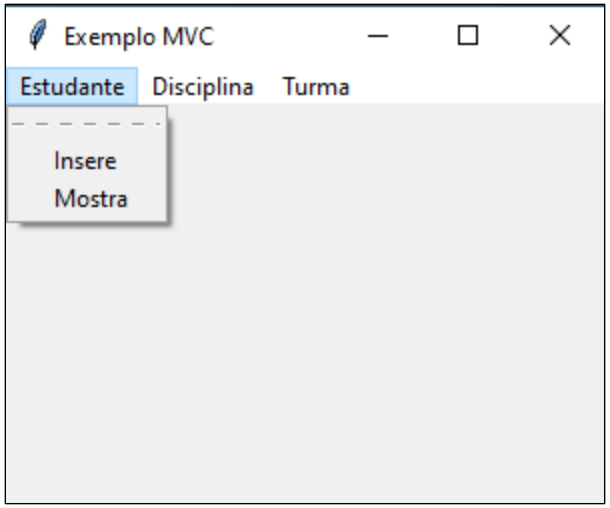
Criando um Menu

```
import tkinter as tk
from tkinter import messagebox
import estudante as est

class LimitePrincipal():
    def __init__(self, root, controle):
        self.controle = controle
        self.root = root
        self.root.geometry('300x200')
        self.menubar = tk.Menu(self.root)
        self.estudanteMenu = tk.Menu(self.menubar)
        self.discipMenu = tk.Menu(self.menubar)
        self.turmaMenu = tk.Menu(self.menubar)

        self.estudanteMenu.add_command(label="Insere", \
                    command=self.controle.insereEstudantes)
        self.estudanteMenu.add_command(label="Mostra", \
                    command=self.controle.mostraEstudantes)
        self.menubar.add_cascade(label="Estudante", \
                    menu=self.estudanteMenu)
```



Criando um Menu

```
        self.discipMenu.add_command(label="Insere")
        self.menubar.add_cascade(label="Disciplina", \
                    menu=self.discipMenu)

        self.turmaMenu.add_command(label="Insere")
        self.menubar.add_cascade(label="Turma", \
                    menu=self.turmaMenu)

        self.root.config(menu=self.menubar)
```

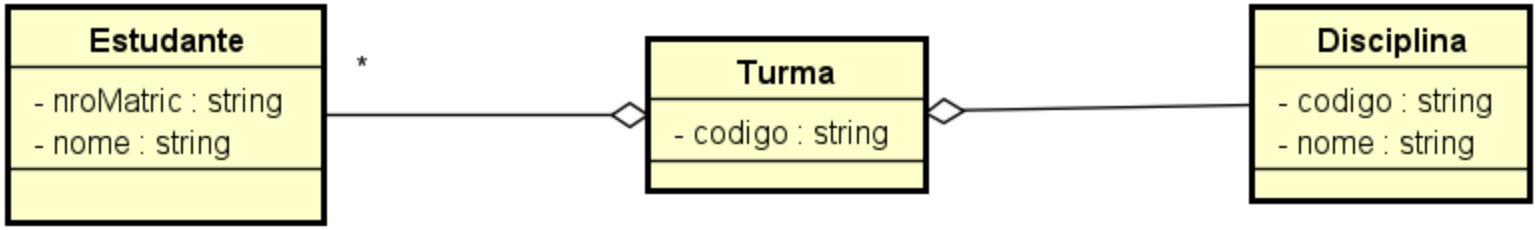


Estudante

- Agora precisamos criar as classes associadas ao Estudante
  - 1 Entidade
  - 1 Controle
  - 2 Limites

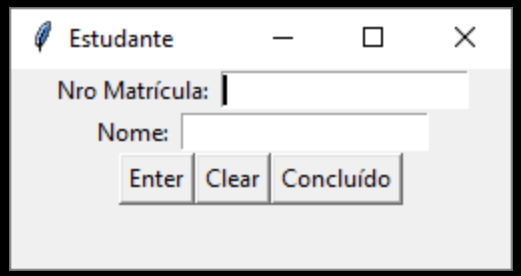
Estudante

```
import tkinter as tk
from tkinter import messagebox

class Estudante:

    # implementar
```

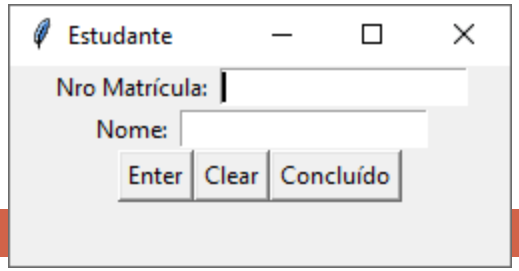


Estudante

```
class LimiteInsereEstudantes(tk.Toplevel):
    def __init__(self, controle):

        tk.Toplevel.__init__(self)
        self.geometry('250x100')
        self.title("Estudante")
        self.controle = controle

        self.frameNro = tk.Frame(self)
        self.frameNome = tk.Frame(self)
        self.frameButton = tk.Frame(self)
        self.frameNro.pack()
        self.frameNome.pack()
        self.frameButton.pack()

        self.labelNro = tk.Label(self.frameNro,\
                        text="Nro Matrícula: ")
        self.labelNome = tk.Label(self.frameNome,text="Nome: ")
        self.labelNro.pack(side="left")
        self.labelNome.pack(side="left")
```



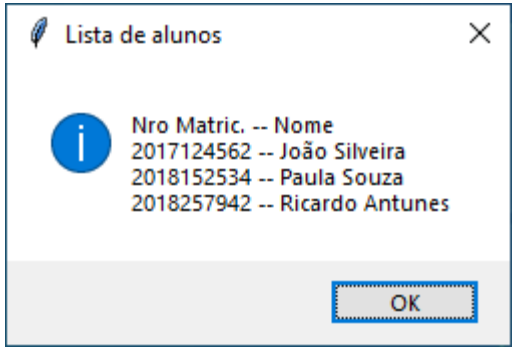
Estudante




```
self.inputNro = tk.Entry(self.frameNro, width=20)
self.inputNro.pack(side="left")
self.inputNome = tk.Entry(self.frameNome, width=20)
self.inputNome.pack(side="left")

self.buttonSubmit = tk.Button(self.frameButton ,text="Enter")

self.buttonSubmit.pack(side="left")
self.buttonSubmit.bind("<Button>", controle.enterHandler)

self.buttonClear = tk.Button(self.frameButton ,text="Clear")

self.buttonClear.pack(side="left")
self.buttonClear.bind("<Button>", controle.clearHandler)

self.buttonFecha = tk.Button(self.frameButton ,text="Concluído")

self.buttonFecha.pack(side="left")
self.buttonFecha.bind("<Button>", controle.fechaHandler)
```

Estudante

```
    def mostraJanela(self, titulo, msg):
        messagebox.showinfo(titulo, msg)

class LimiteMostraEstudantes():
    def __init__(self, controle):
        messagebox.showinfo('Lista de alunos', controle.getEstudantes())


class CtrlEstudante():

    # implementar
```



Exercício

- Implementar o cadastro e a listagem de disciplinas