

## Arquivo: classificacao\_orcamentaria.py

```
from PyQt6.QtWidgets import *
from PyQt6.QtCore import QDate
import pandas as pd

def create_classificacao_orcamentaria_group(data):
    """Cria o QGroupBox para a seção de Classificação Orçamentária."""
    classificacao_orcamentaria_group_box = QGroupBox("Classificação Orçamentária")
    classificacao_orcamentaria_group_box.setStyleSheet("""
        QGroupBox {
            border: 1px solid #3C3C5A;
            border-radius: 10px;
            font-size: 16px;
            font-weight: bold;
            color: white;
            margin-top: 13px;
        }
        QGroupBox:title {
            subcontrol-origin: margin;
            padding: 0 3px;
        }
    """)
    classificacao_orcamentaria_group_box.setFixedWidth(400)

    layout = QVBoxLayout()
    widgets_classificacao_orcamentaria = {} # Dicionário para armazenar os widgets

    # Criando cada campo de entrada com conversão de tipo
    valor_total_layout = QHBoxLayout()
    valor_total_label = QLabel("Valor Estimado:")
    valor_total_edit = QLineEdit(str(data.get('valor_total', '')))
    valor_total_layout.addWidget(valor_total_label)
    valor_total_layout.addWidget(valor_total_edit)
    layout.addLayout(valor_total_layout)

    acao_interna_layout = QHBoxLayout()
    acao_interna_label = QLabel("Ação Interna:")
    acao_interna_edit = QLineEdit(str(data.get('acao_interna', '')))
    acao_interna_layout.addWidget(acao_interna_label)
    acao_interna_layout.addWidget(acao_interna_edit)
    layout.addLayout(acao_interna_layout)

    fonte_recursos_layout = QHBoxLayout()
    fonte_recursos_label = QLabel("Fonte de Recurso (FR):")
    fonte_recursos_edit = QLineEdit(str(data.get('fonte_recursos', '')))
    fonte_recursos_layout.addWidget(fonte_recursos_label)
    fonte_recursos_layout.addWidget(fonte_recursos_edit)
    layout.addLayout(fonte_recursos_layout)

    natureza_despesa_layout = QHBoxLayout()
    natureza_despesa_label = QLabel("Natureza de Despesa (ND):")
```

```

natureza_despesa_edit = QLineEdit(str(data.get('natureza_despesa', '')))
natureza_despesa_layout.addWidget(natureza_despesa_label)
natureza_despesa_layout.addWidget(natureza_despesa_edit)
layout.addLayout(natureza_despesa_layout)

unidade_orcamentaria_layout = QHBoxLayout()
unidade_orcamentaria_label = QLabel("Unidade Orçamentária (UO):")
unidade_orcamentaria_edit = QLineEdit(str(data.get('unidade_orcamentaria', '')))
unidade_orcamentaria_layout.addWidget(unidade_orcamentaria_label)
unidade_orcamentaria_layout.addWidget(unidade_orcamentaria_edit)
layout.addLayout(unidade_orcamentaria_layout)

ptres_layout = QHBoxLayout()
ptres_label = QLabel("PTRES:")
ptres_edit = QLineEdit(str(data.get('ptres', '')))
ptres_layout.addWidget(ptres_label)
ptres_layout.addWidget(ptres_edit)
layout.addLayout(ptres_layout)

# Adicionando o rádio button de Atividade de Custeio
custeio_layout = QHBoxLayout()
custeio_label = QLabel("Atividade de Custeio?")
radio_custeio_sim = QRadioButton("Sim")
radio_custeio_nao = QRadioButton("Não")
custeio_group = QButtonGroup() # Grupo exclusivo para o conjunto de botões
custeio_group.addButton(radio_custeio_sim)
custeio_group.addButton(radio_custeio_nao)

# Define o estado inicial com base nos dados
atividade_custeio_value = data.get('atividade_custeio', 'Não')
radio_custeio_sim.setChecked(atividade_custeio_value == 'Sim')
radio_custeio_nao.setChecked(atividade_custeio_value == 'Não')

custeio_layout.addWidget(custeio_label)
custeio_layout.addWidget(radio_custeio_sim)
custeio_layout.addWidget(radio_custeio_nao)

# Adiciona o layout do rádio button ao layout principal
layout.addLayout(custeio_layout)

widgets_classificacao_orcamentaria = {
    'valor_total_edit': valor_total_edit,
    'acao_interna_edit': acao_interna_edit,
    'fonte_recursos_edit': fonte_recursos_edit,
    'natureza_despesa_edit': natureza_despesa_edit,
    'unidade_orcamentaria_edit': unidade_orcamentaria_edit,
    'ptres_edit': ptres_edit,
    'radio_custeio_sim': radio_custeio_sim,
    'radio_custeio_nao': radio_custeio_nao
}

classificacao_orcamentaria_group_box.setLayout(layout)

```

```
return classificacao_orcamentaria_group_box, widgets_classificacao_orcamentaria
```