

Arquivo: treeview_utils.py

```
# treeview_utils.py

from PyQt6.QtWidgets import *
from PyQt6.QtGui import *
from PyQt6.QtCore import *
import pandas as pd
import os
import subprocess
import sys

def create_button(text, icon, callback, tooltip_text, parent, icon_size=QSize(40, 40)): # Aumente
o tamanho padrão do ícone
    btn = QPushButton(text, parent)
    if icon:
        btn.setIcon(QIcon(icon))
        btn.setIconSize(icon_size) # Define o tamanho do ícone
    if callback:
        btn.clicked.connect(callback)
    if tooltip_text:
        btn.setToolTip(tooltip_text)

    # Aplica folhas de estilo para personalizar a aparência do botão, incluindo efeito de hover
    btn.setStyleSheet("""
QPushButton {
    background-color: #071661;
                font-size: 14pt;
    min-height: 30px;
    padding: 5px;
}
QPushButton:hover {
    background-color: white;
    color: #050f41;
    border: 2px solid #050f41;
}
QPushButton:pressed {
    background-color: #1c3664;
}
""")

    return btn

def create_button_2(text, icon, callback, tooltip_text, parent, icon_size=QSize(40, 40)): #
Aumente o tamanho padrão do ícone
    btn = QPushButton(text, parent)
    if icon:
        btn.setIcon(QIcon(icon))
        btn.setIconSize(icon_size) # Define o tamanho do ícone
    if callback:
        btn.clicked.connect(callback)
```

```

if tooltip_text:
    btn.setToolTip(tooltip_text)

# Aplica folhas de estilo para personalizar a aparência do botão, incluindo efeito de hover
btn.setStyleSheet("""
QPushButton {
    background-color: #071661
    font-size: 14pt;
    min-height: 30px;
    padding: 5px;
}
QPushButton:hover {
    background-color: white;
    color: #050f41;
    border: 2px solid #050f41;
}
QPushButton:pressed {
    background-color: #1c3664;
}
""")

return btn

def save_dataframe_to_excel(data_frame, file_path):
    try:
        data_frame.to_excel(file_path, index=False)
        print("DataFrame saved successfully.")
    except Exception as e:
        print(f"Error saving DataFrame: {e}")

def open_folder(path):
    if sys.platform == 'win32': # Para Windows
        os.startfile(path)
    elif sys.platform == 'darwin': # Para macOS
        subprocess.Popen(['open', path])
    else: # Para Linux e outros sistemas Unix-like
        subprocess.Popen(['xdg-open', path])

```