

## Explicação do Código

O objetivo deste código é construir uma árvore hierárquica a partir de dados armazenados em um DataFrame do pandas. Cada linha do DataFrame representa um nó em algum nível da hierarquia, e a árvore é construída de acordo com as colunas: Root, Branch, Sub-Branch, Leaf, Subleaf, e URL. Aqui está uma explicação detalhada do código:

### Classe `Node`

A classe `Node` representa um nó na árvore.

python

```
class Node:
    def __init__(self, name, url=None):
        self.name = name # Nome do nó
        self.url = url # URL associada ao nó, se houver
        self.children = [] # Lista de filhos do nó

    def add_child(self, child_node):
        self.children.append(child_node) # Adiciona um nó filho à lista de fi
```

- `__init__(self, name, url=None)`: Inicializa um novo nó com um nome e uma URL opcional. Também inicializa uma lista vazia para armazenar os nós filhos.
- `add_child(self, child_node)`: Adiciona um nó filho à lista de filhos do nó atual.

### Função `build\_tree\_from\_df`

Esta função constrói a árvore a partir de um DataFrame do pandas.

python

```
def build_tree_from_df(df):
    roots = {} # Dicionário para armazenar os nós raiz
    for _, row in df.iterrows():
        root_name = row['Root'] # Nome da raiz
        branch_name = row['Branch'] # Nome do ramo
        sub_branch_name = row['Sub-Branch'] # Nome do sub-ramo
        leaf_name = row['Leaf'] # Nome da folha
        subleaf_name = row['Subleaf'] # Nome da subfolha
        url = row['URL'] # URL associada ao nó

        # Se a raiz ainda não existir, crie-a
```