Fiz todas as alterações menos esta:

1 - Primeiro, o problema do zero não aparecer no triangle-grid e a influência nas combinações:

Voçe pediu pra alterar o app.py porem a função generateGrid esta no main.js

OO QUE EU FAÇO???

# No arquivo app.py, ajuste a função generateGrid para mostrar o zero:

def generateGrid(numbers):

const grid = document.querySelector('.triangle-grid');

grid.innerHTML = '';

const rows = [5, 4, 3, 2, 1];

let numberIndex = 0;

rows.forEach((cellCount) => {

const row = document.createElement('div');

row.className = 'grid-row';

for (let i = 0; i < cellCount; i++) {

const cell = document.createElement('div');

cell.className = 'grid-cell';

// Aqui está a correção para mostrar o zero

cell.textContent = numbers[numberIndex].toString();

numberIndex++;

}

grid.appendChild(row);

});

}

# E na função get\_valid\_pairs do app.py:

def get\_valid\_pairs(grid):

valid\_pairs = set()

rows = [

grid[0:5], # primeira linha

grid[5:9], # segunda linha

grid[9:12], # terceira linha

grid[12:14], # quarta linha

grid[14:15] # quinta linha

]

# Horizontal (esquerda para direita e direita para esquerda)

for row in rows:

for i in range(len(row)-1):

# Considerar combinações nos dois sentidos

num1 = combine\_digits(row[i], row[i+1])

num2 = combine\_digits(row[i+1], row[i])

# Zero à esquerda também é válido (06)

if 1 <= num1 <= 60:

valid\_pairs.add(num1)

if 1 <= num2 <= 60:

valid\_pairs.add(num2)

# Especial para o zero

if row[i] == 0:

zero\_combo = row[i+1]

if 1 <= zero\_combo <= 60:

valid\_pairs.add(zero\_combo)

if row[i+1] == 0:

zero\_combo = row[i]

if 1 <= zero\_combo <= 60:

valid\_pairs.add(zero\_combo)

