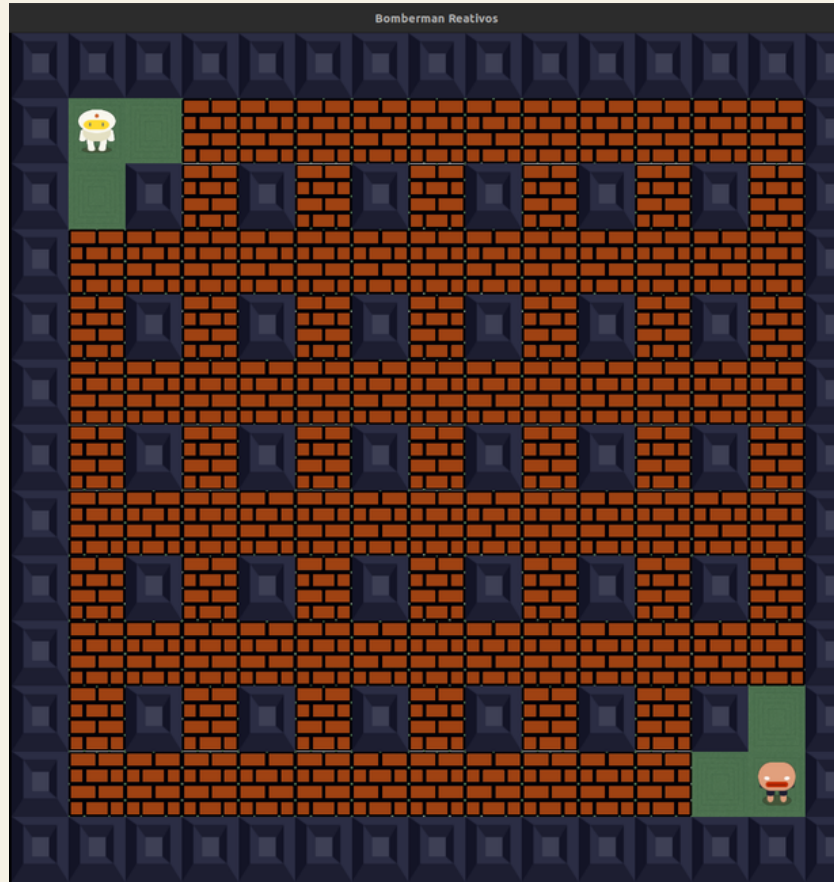

SISTEMAS REATIVOS

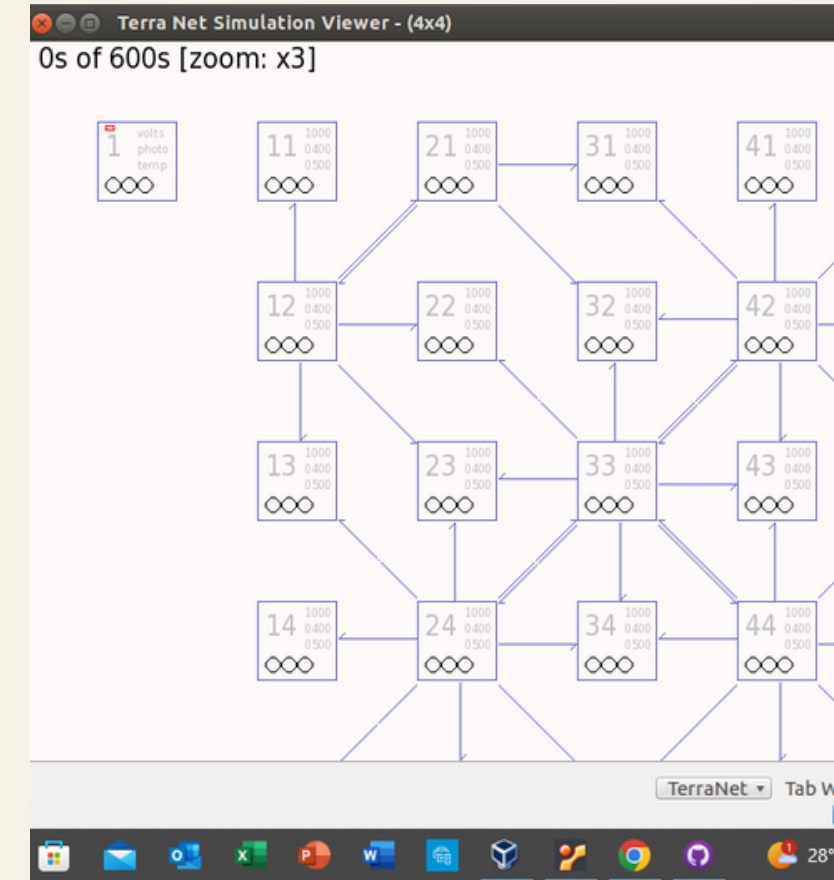
Jerônimo Augusto Soares
Paulo de Tarso

PROJETOS ANTERIORES

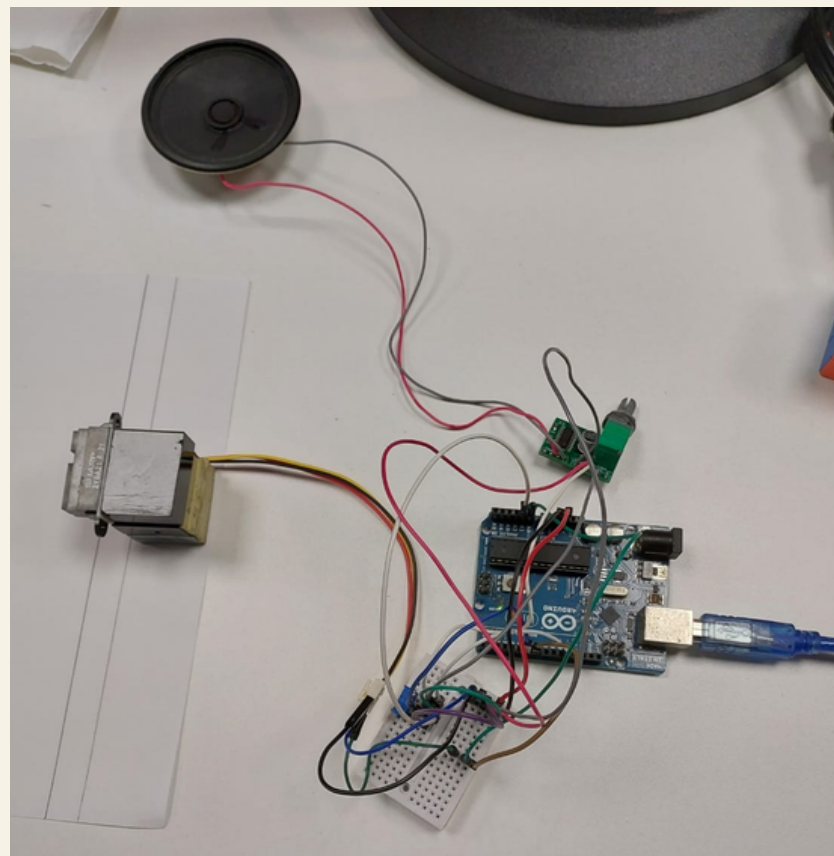
Bomberman
Reativos



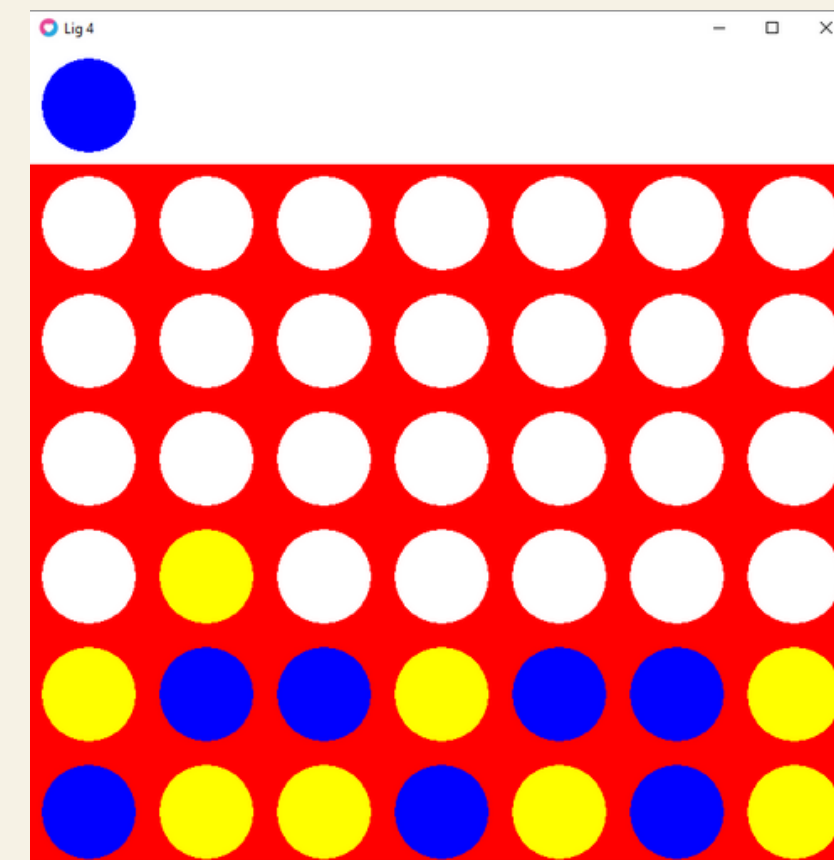
RSSF



Piano de
8 teclas

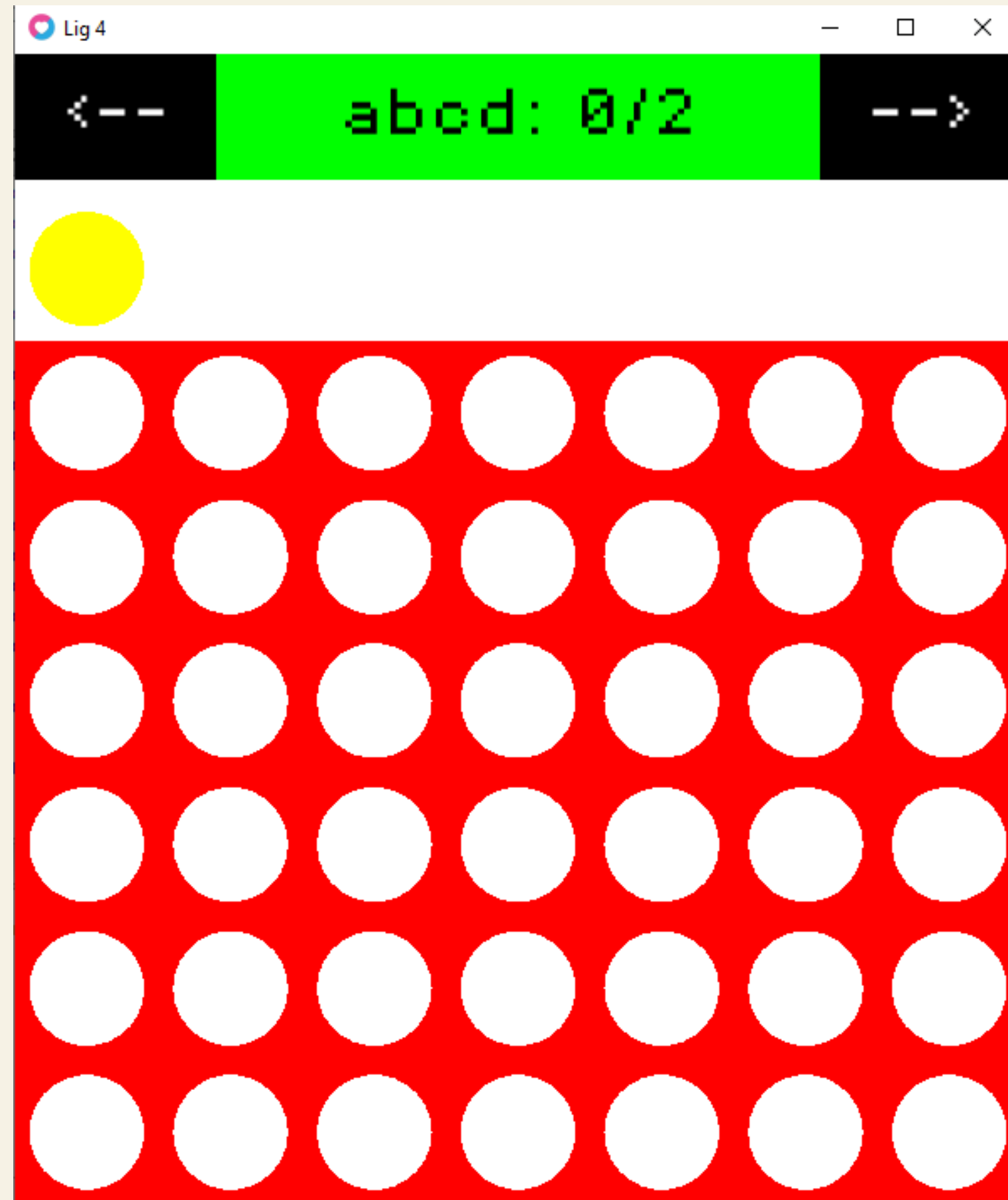


Lig-4



PROJETO

Lig-4 com
servidor
online para
vários
jogadores

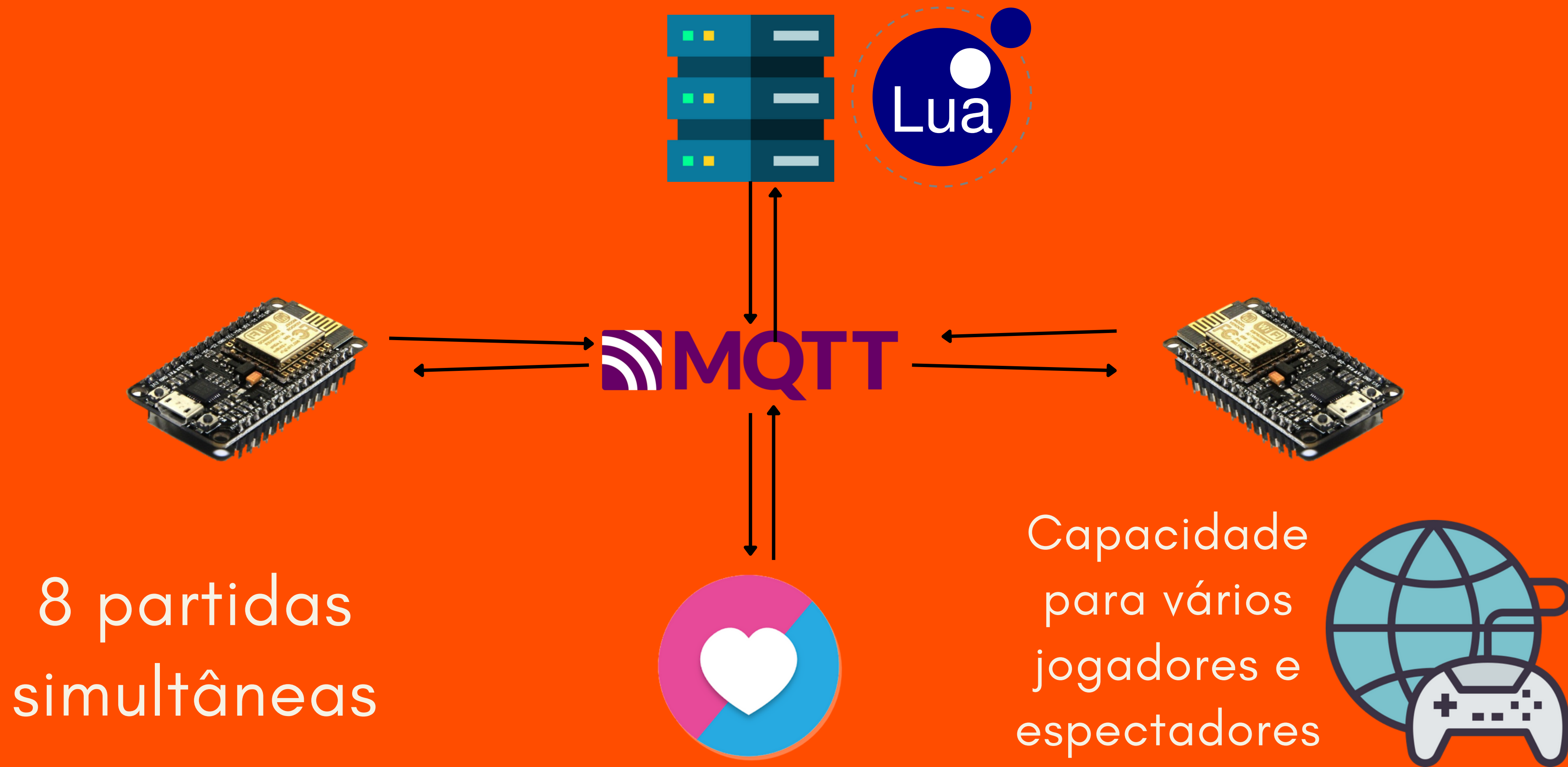


INCREMENTOS PREVIAMENTE CONSTATADOS

Tratar bounce dos botões.

Adicionar efeitos sonoros reproduzidos no buzzer do NodeMCU.

DESCRIÇÃO DAS NOVIDADES



NOVA ESTRUTURA DO CÓDIGO

- Adição do servidor
- Atualização da janela LÖVE
- Modularização dos arquivos do NodeMCU

Quantidade
de arquivos:

8



LOVE

9

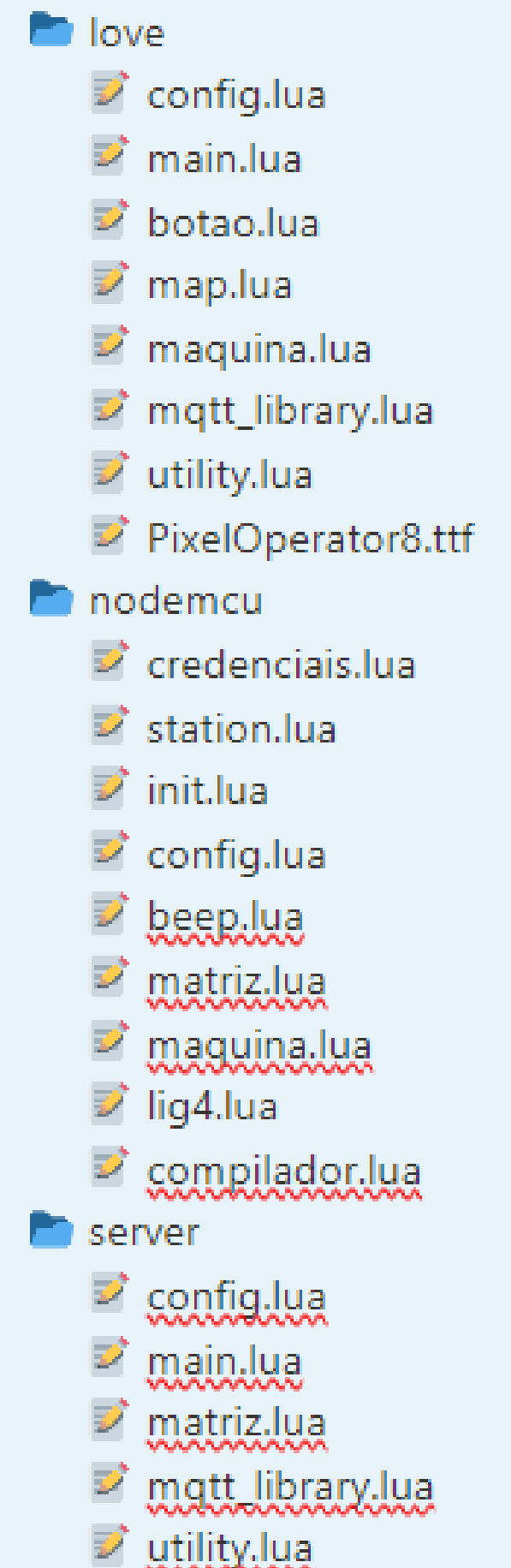


NODEMCU

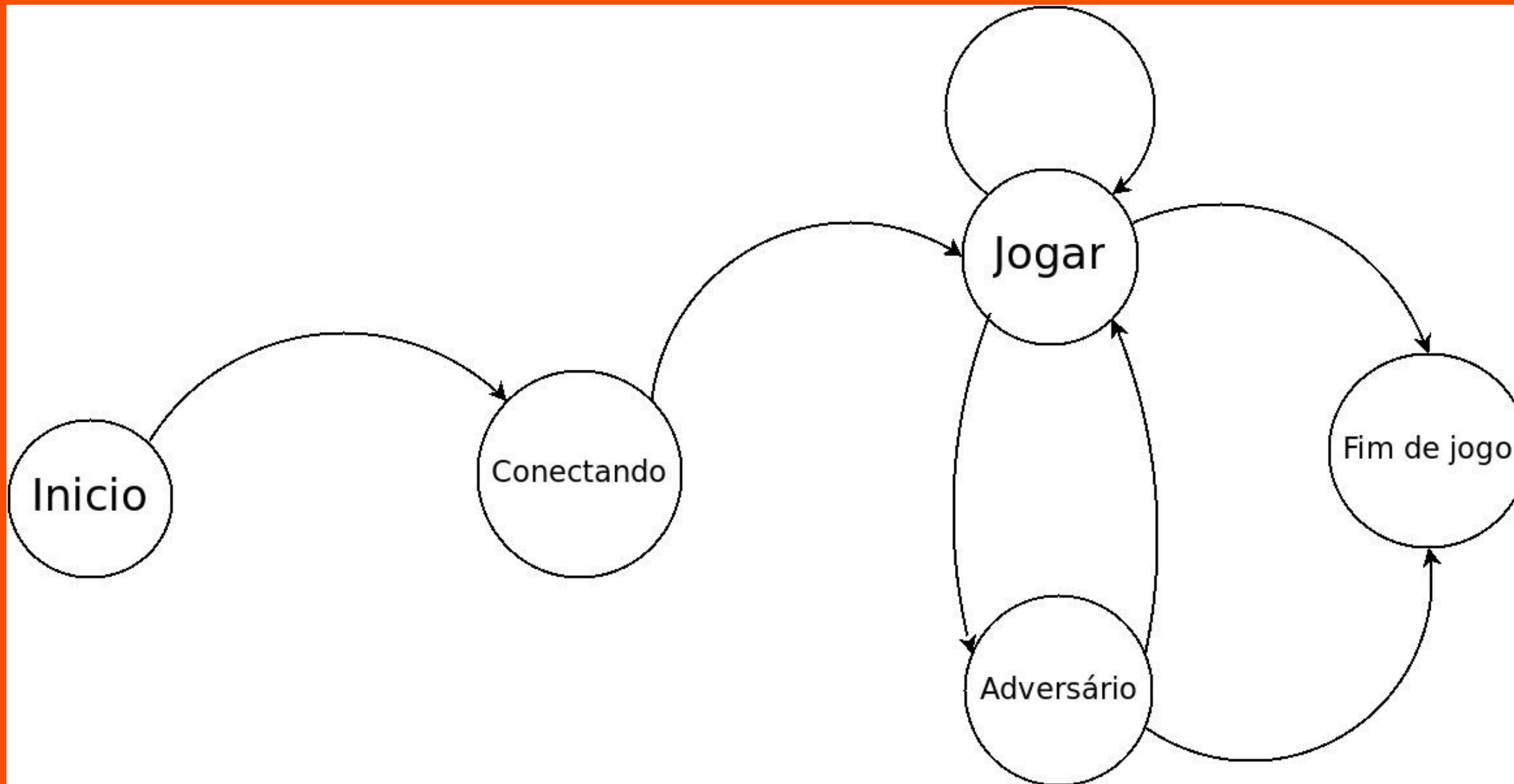
5



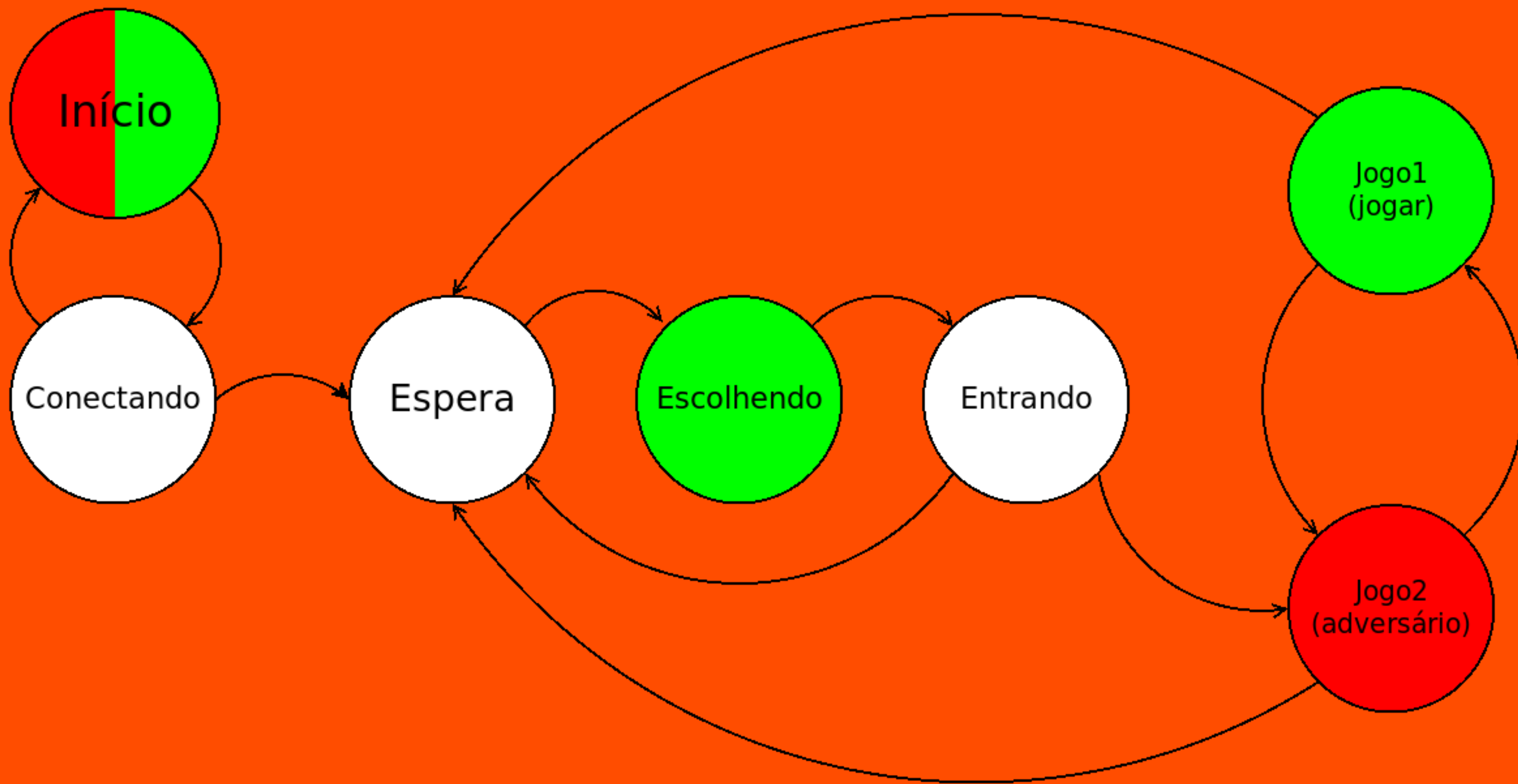
SERVER



MÁQUINA DE ESTADOS ANTERIOR



MÁQUINA DE ESTADOS ATUALIZADA



Cada estado
é indicado
pelos LEDs
dos
NodeMCUs

DIFICULDADES

Limitação de memória no NodeMCU

Cloud Build Service: <https://nodemcu-build.com>

LFS options (for release & dev branches) ^

The [Lua Flash Store \(LFS\)](#) allows NodeMCU Lua code and constants to be executed directly out of flash-memory. This enables Lua applications with up to 256 KB Lua code. All of the RAM is thus available for read-write data.

You must choose an explicit (non-default) LFS size below to enable the use of LFS.

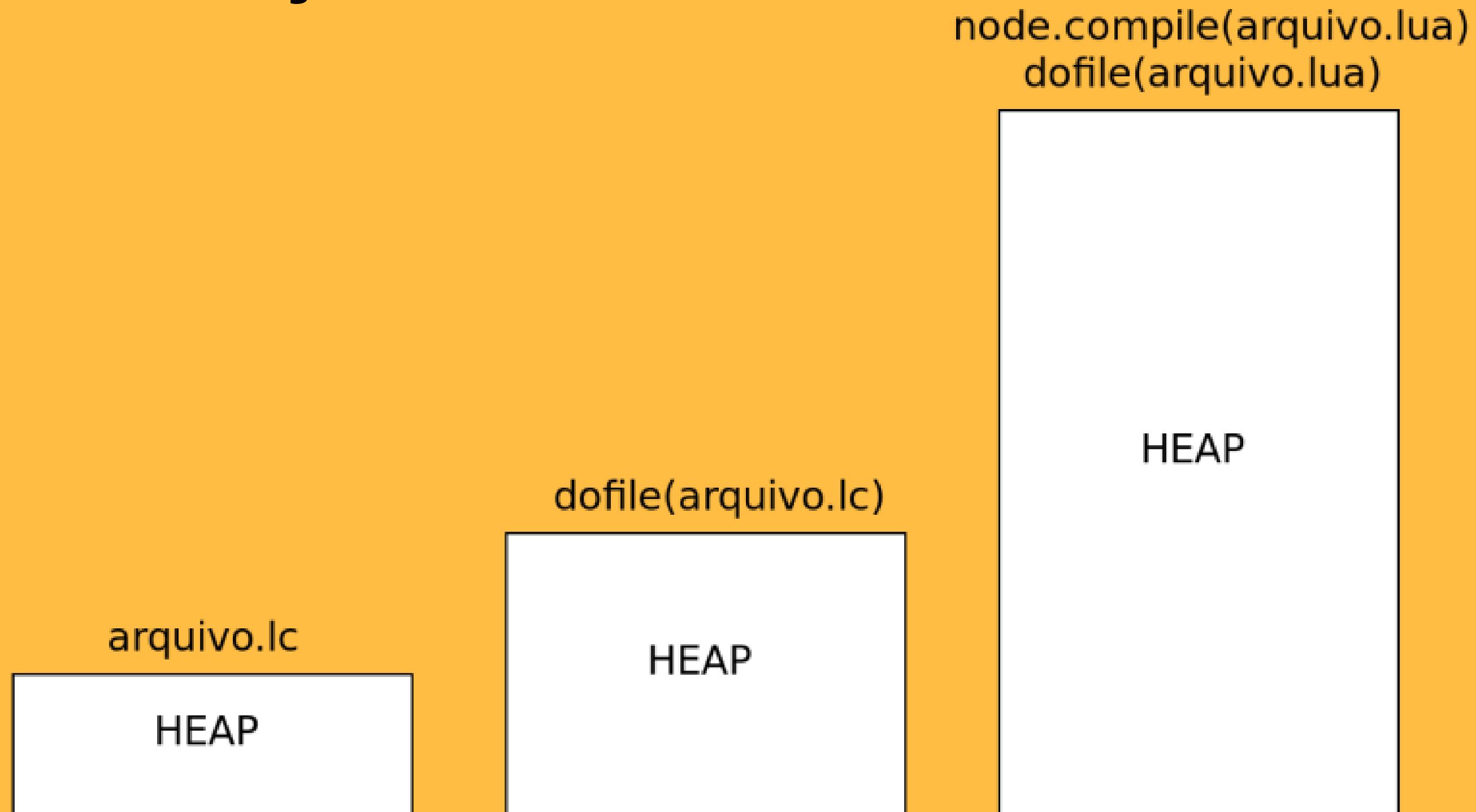
! It's *your* responsibility to select a valid combination of options below! No validation is performed by the build service.

LFS size	SPIFFS base (fixed location)	SPIFFS size
0, no LFS ▾	0, start right after firmwar ▾	all free flash ▾

Tentativa: alocação de Lua Flash Store no firmware

DIFICULDADES

Limitação de memória no NodeMCU



Solução: modularização e compilação

DIFICULDADES

Sons para diferenciar cada sala na seleção

```
local nomesSalas = {"a","ab","abc","abcd","g","gf","gfe","gfed"}
```

```
escolhendo = {  
  botao1 = function(l, t)  
    if consts.x > 0 then  
      consts.x = consts.x - 1  
      print("x=" .. consts.x)  
      print("sala=" .. (salas[consts.x] or "nil"))  
      if salas[consts.x] == nil then return end  
      local sala = salas[consts.x]  
      local notas = {}  
      for i = 1, sala:len() do  
        notas[i] = { sala:sub(i, i), 100 }  
      end  
      beep(notas)  
    end  
  end,  
}
```

Solução: variação na quantidade de notas musicais

DIFICULDADES

Controlar tempo de inatividade

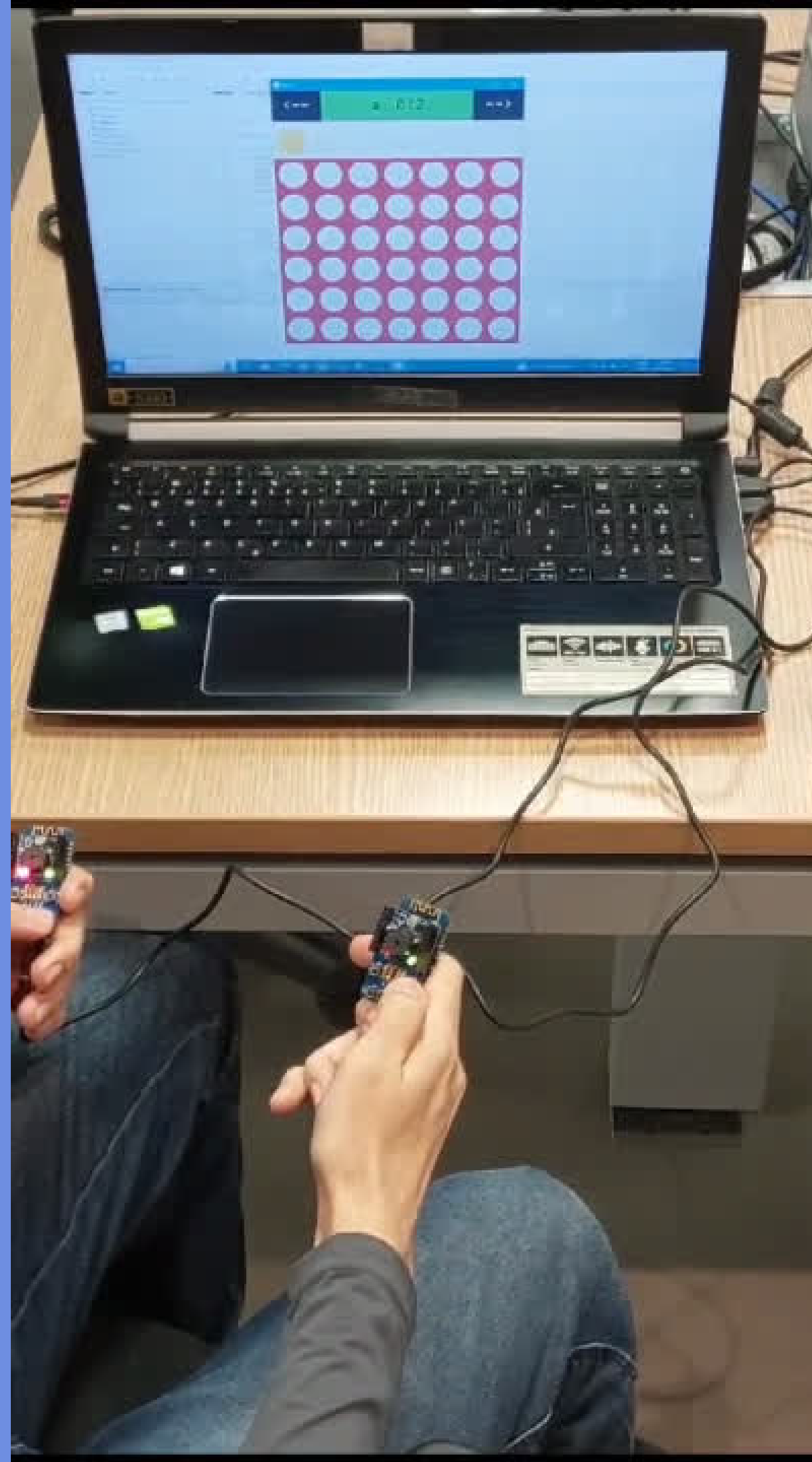
```
local sala = salas[tmsg[3]]
local qtdJog = sala.qtdJogadores

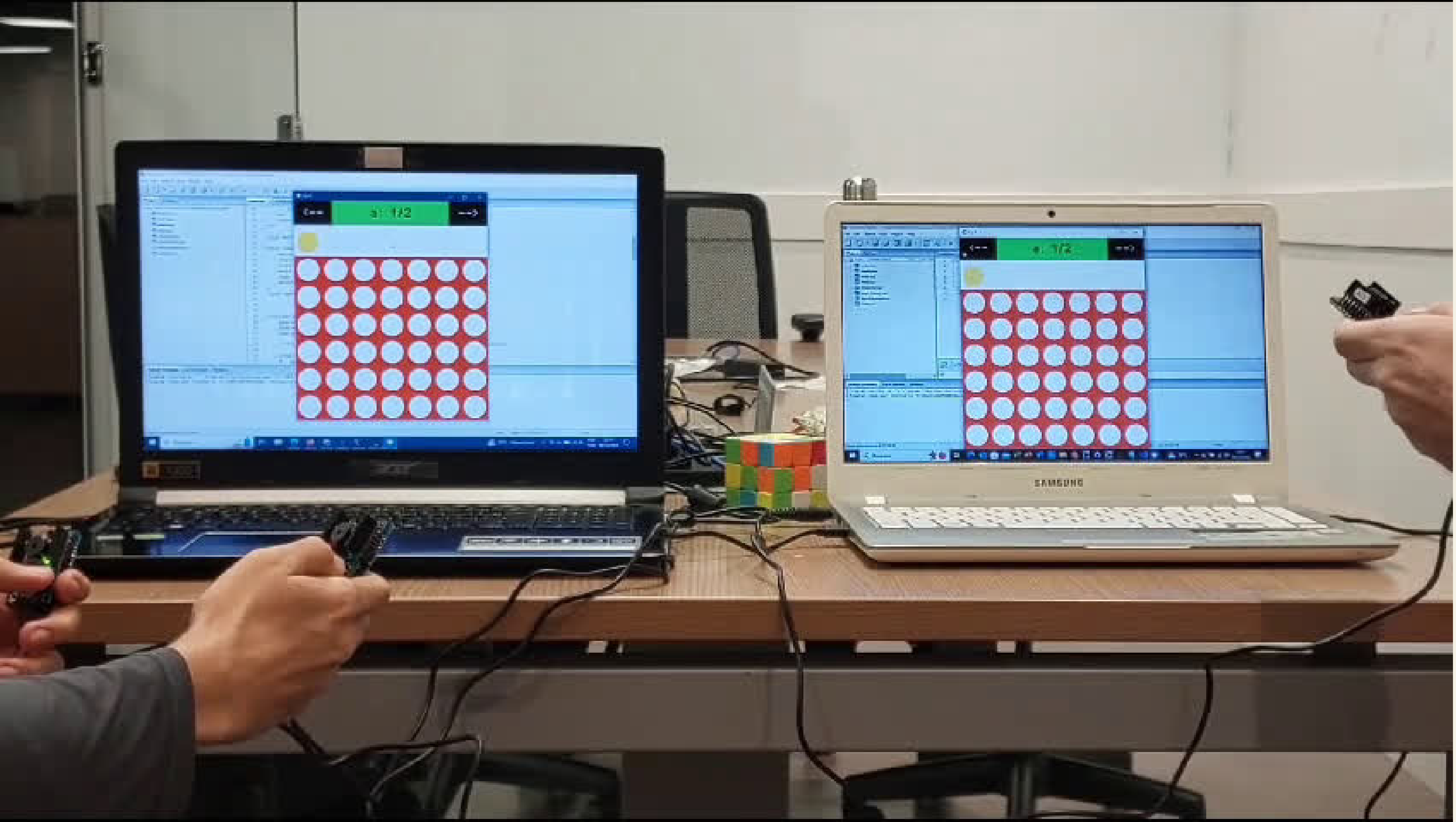
if tmsg[4] == "SUB" then
    -- node_id,BROADCAST,salax,SUB
    if qtdJog == 0 then
        -- salas[tmsg[3]] = 1
        sala.qtdJogadores = 1
        local msgSend = meuid .. "," .. tmsg[1] .. "," .. tmsg[3] .. ",JOG1"
        print(msgSend)
        mqtt_client:publish(topic, msgSend)
        sala.timer = coroutine.create(timer)
        coroutine.resume(sala.timer, timeout, function ()
            sala.matriz = criaMatrizVazia()
            sala.x = 1
            sala.qtdJogadores = 0
            sala.vez=1
            sala.timer=nil
            local msgSend = meuid .. ",BROADCAST," .. tmsg[3] .. ",RESET"
            print(msgSend)
            mqtt_client:publish(topic, msgSend)
        end)
    end
end
```

```
local timer = function(tempo, callback)
    local tempo1 = os.time()
    while true do
        local dt = os.time() - tempo1
        if dt >= tempo then
            callback()
            break
        end
        coroutine.yield()
    end
end
```

Solução: uso da corrotina para timer

DEMONSTRAÇÃO DO PROJETO





JOGUE VOCÊ MESMO!

PDTCCLF/INF1350— Projetos



2

Contributors



0

Issues



0

Stars



0

Forks



PDTCCLF/INF1350---Projetos

Contribute to PDTCCLF/INF1350---Projetos development by creating an account on GitHub.



OBRIGADO!