



# IO1

## Minecraft Pi & Blocos de Computação Física

---

Deliverable: IO1A2



### STEM4CLIM8

29 Setembro, 2021

---

HeartHands Solutions LTD

Autoria de: HeartHands Solutions LTD

Projeto Número: 2020-1-UK01-KA201-079141



Co-funded by the  
Erasmus+ Programme  
of the European Union

O apoio da Comissão Europeia à produção desta publicação não constitui um aval do seu conteúdo, que reflete unicamente o ponto de vista dos autores, e a Comissão não pode ser considerada responsável por eventuais utilizações que possam ser feitas com as informações nela contidas.

## Resumo Executivo

Num inquérito recente da OCDE (OECD- Education and Skills Today 2018), que envolveu 25 países europeus, quase todos os países relatam défices das competências necessárias aos professores para satisfazer as exigências das escolas, conjugadas com as dificuldades em atualizar as competências dos professores. Contudo, espera-se que a digitalização mude profundamente a forma como aprendemos e trabalhamos. É provável que muitas crianças que hoje entram na escola venham a trabalhar em empregos que ainda não existem. Preparar os estudantes para estes territórios desconhecidos, significa que não só temos de nos certificar de que eles têm as capacidades técnicas adequadas, mas também que temos de reforçar as suas competências emocionais e sociais. A resiliência, a capacidade individual de ultrapassar circunstâncias adversas e de as utilizar como fontes de desenvolvimento pessoal, está no cerne da capacidade de se adaptarem com sucesso à mudança e, assim, de se envolverem ativamente no nosso mundo digital. Ao mesmo tempo, precisamos de reconhecer a dependência da Internet e os comportamentos de risco de IAB (Internet Addiction Behaviour) como problemas emergentes para a nossa juventude. Uma abordagem STEM, que une a computação física com a consciência ambiental, ao mesmo tempo que se foca em atividades colaborativas, é uma excelente forma de melhorar as capacidades técnicas, ao mesmo tempo que reforça as capacidades emocionais e sociais.

A STEM4CLIM8 tem como objetivo principal produzir abordagens e ferramentas para ajudar todos aqueles que trabalham com crianças, com vista a ajudá-los a envolverem-se na programação e a desenvolverem competências relacionadas com as STEM. O objetivo não é aumentar o tempo em ecrãs, mas sim estimular o envolvimento no jogo através da criação de um mundo virtual personalizado, utilizando a modelação Minecraft e a execução de missões relacionadas com desastres naturais, utilizando blocos de computação física programados para interagir com o mundo virtual através do GPIO Raspberry. As missões vão mostrar a ciência por detrás dos fenómenos naturais, frequentemente associados às alterações climáticas, e inspirar a consciência ambiental, ao mesmo tempo que reforçam as competências STEM.

### Referências:

\* OCDE- Education and Skills Today, Succeeding with resilience-Lessons for schools, 29 de Janeiro de 2018, Obtido a 18 de Fevereiro de 2021 em:: <https://oecdeditoday.com/succeeding-with-resilience-lessons-for-schools/>

## Índice

<b>RESUMO EXECUTIVO .....</b>	<b>2</b>
<b>1. BLOCOS DE COMPUTAÇÃO FÍSICA EM MINECRAFT PI .....</b>	<b>4</b>
1.1 INSTALAÇÃO DA “FISHING TOWN” MUNDO MINECRAFT PI.....	4
1.2 CENÁRIO E INTERAÇÕES .....	5
1.3 SISMO .....	5
1.3.1 Equipamento e cenário .....	5
1.3.2 Circuito eletrônico e bloco físico.....	6
1.3.3 Programação Python .....	7
1.4 DOLINA (SINKHOLE) .....	8
1.4.1 Equipamento e cenário .....	8
1.4.2 Circuito eletrônico e bloco físico.....	9
1.4.3 Programação Python .....	10
1.5 ERUPÇÃO VULCÂNICA .....	12
1.5.1 Equipamento e cenário .....	12
1.5.2 Circuito eletrônico e bloco físico.....	12
1.5.3 Programação Python .....	14
1.6 ONDA DE CALOR.....	15
1.6.1 Equipamento e Cenário.....	15
1.6.2 Circuito eletrônico e execução da atividade.....	15
1.6.3 Programação Python .....	18
1.7 INUNDAÇÃO .....	19
1.7.1 Equipamento e Cenário.....	19
1.7.2 Circuito eletrônico e execução da atividade.....	19
1.7.3 Programação Python .....	23
1.8 CONCLUSÃO .....	23
<b>REFERÊNCIAS .....</b>	<b>23</b>

## 1. Blocos de Computação Física em Minecraft Pi

Um conjunto de blocos de computação física foram desenvolvidos para interagir com o Mundo Minecraft Pi, onde ocorrem uma série de desastres naturais. Estes blocos serão ligados através da GPIO Raspberry Pi e serão programados em Python, de modo a interagir com o mundo Minecraft Pi, simulando o desastre natural e os seus resultados catastróficos numa pequena “Fishing Town”.

Este guia, dirigido aos educadores, contém instruções para a montagem dos blocos de computação física, para a sua ligação à GPIO e para a sua programação utilizando Python e Minecraft Pi.

Os professores devem ter:

- Consola STEM4CLIM8
- Blocos de Computação Física
- Material didático para o ensino destes fenómenos naturais.

### 1.1 Instalação da “Fishing Town” Mundo Minecraft Pi

Para executar qualquer um dos cenários dos desastres naturais, é necessário instalar primeiro um mapa da cidade piscatória, que será utilizado como laboratório de ensaios dos diferentes desastres naturais. Para o fazer, é necessário ligar o Raspberry Pi na consola STEM4CLIM8.

No ambiente de trabalho, encontra uma pasta chamada "Fishing Town Map" (Mapa da Cidade Piscatória). Fazer duplo clique sobre ela para ver o conteúdo. Em seguida, seguir estes passos:

1. Abrir Minecraft Pi e criar um novo mundo, depois fechar Minecraft Pi
2. Abrir "Gestor de ficheiros".
3. Clicar em " View" e assinalar "Show Hidden".
4. Duplo clique sobre a pasta ".minecraft".
5. Duplo clique sobre a pasta "jogos".
6. Duplo clique sobre a pasta "com.mojang".
7. Duplo clique sobre a pasta "minecraftWorlds".
8. Copiar o conteúdo da pasta "Fishing Town Map" para a pasta já existente "world".
9. Renomear essa pasta para “Fishing Town”.
10. Abrir o Minecraft Pi e carregar o mundo “Fishing Town”.

A “Fishing Town” do mundo Minecraft Pi será carregada. Sempre que necessitar de um novo mapa sem os desastres naturais, é necessário repetir este procedimento.

Este mundo Minecraft Pi será utilizado para simular os seguintes desastres naturais:

- Sismo
- Erupção vulcânica
- Dolina (Sinkhole)
- Onda de calor
- Inundação

## 1.2 Cenário e Interações

Com a utilização da GPIO Raspberry Pi, sensores, eletrónica, periféricos e construção de caixas de cartão, os estudantes vão aprender mais sobre os desastres naturais, vão testar os seus resultados e discutir as suas consequências.

No início, os professores orientam os estudantes sobre a ligação dos sensores e outros componentes eletrónicos ao Raspberry Pi, bem como na construção das caixas de cartolina e na execução das atividades do Minecraft Pi em Python.

O mundo Minecraft Pi está configurado como uma pequena cidade piscatória na qual os desastres naturais vão ocorrer. As atividades estão divididas em cinco tópicos:

1. Sismo
2. Dolina (Sinkhole)
3. Erupção vulcânica
4. Onda de calor
5. Inundação

Foram desenvolvidas instruções específicas de como executar cada atividade. É recomendamos que a turma seja dividida em vários grupos, e cada grupo deve ter acesso a pelo menos uma consola STEM4CLIM8, um teclado e um rato, bem como uma coluna de som.

## 1.3 Sismo

### 1.3.1 Equipamento e cenário

Esta atividade vai apresentar um sismo no mundo da Minecraft Pi. Os estudantes vão ligar um botão à GPIO do Raspberry Pi e programá-lo para gerar um sismo após ser pressionado.

A ideia é criar um bloco de computação física utilizando os seguintes componentes eletrónicos e periféricos:

1. 1 x Placa de ensaio
2. 1 x Botão de pressão com tampa
3. 1 x Resistor 220 Ohm
4. 2 x Cabos de ligação (Macho/Fêmea)
5. 1 x Molde da caixa em cartolina

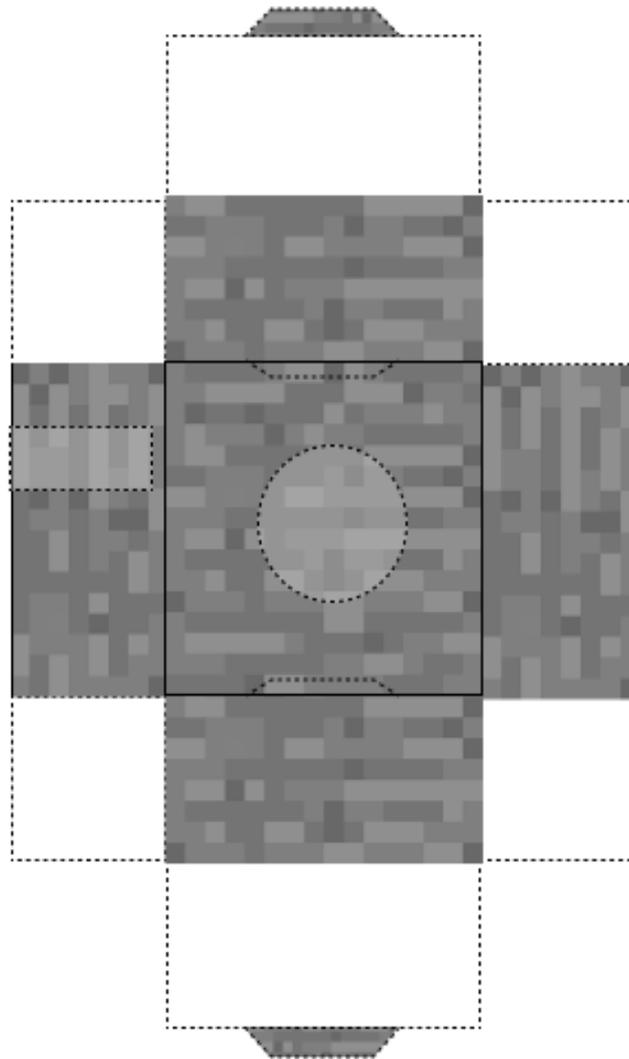
Depois de tudo ligado, os estudantes devem carregar o mapa "Fishing Town" e o script Python "earthquake.py" que pode ser encontrado no ambiente de trabalho da consola STEM4CLIM8.

Depois, cada vez que o botão for premido, vai ocorrer um sismo no mundo Minecraft Pi, que vai destruir edifícios e outros blocos da "Fishing Town", semelhante a um sismo real de grande magnitude.

Após a atividade, deverá seguir-se uma discussão com os estudantes sobre o impacto do sismo e o significado de uma construção bem estruturada.



Depois de criar o circuito, é necessário fazer a montagem de uma caixa e colocar o circuito na mesma. Para o efeito, imprima a imagem seguinte, em Cartolina A4 branca e depois recorte pelas margens indicadas.



**Figura 3 Molde para impressão, da caixa para o bloco de computação física com botão.**

### 1.3.3 Programação Python

Basta abrir Thonny Python no ambiente de trabalho Raspberry Pi, depois carregar o script "earthquake.py" que pode ser encontrado no ambiente de trabalho ou copiar e colar diretamente para o compilador Python à sua escolha. Se decidir ligar o botão de pressão a um pino GPIO diferente, não se esqueça de alterar o número GPIO no script abaixo (linhas 10 e 42).

```
import RPi.GPIO as GPIO
import mcpi.minecraft as minc
import mcpi.block as block
mc = minc.Minecraft.create()
import random, time

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

def earthquake(x, z):
    mc.postToChat('Earthquake!')
```

```

y = mc.getHeight(x, z)
endtime = time.time() + 60
nearthtime = time.time()
while endtime > time.time():
    if time.time() > nearthtime:
        nearthtime = time.time() + 5
    ppos = mc.player.getPos()
    if ppos.x < x+100 and ppos.x > x-100:
        if ppos.y < y+100 and ppos.y > -60:
            if ppos.z < z+100 and ppos.z > z-100:
                mc.player.setPos(ppos.x, ppos.y, ppos.z)
    bx = random.randint(x-100, x+100)
    by = y
    bz = random.randint(z-100, z+100)
    if mc.getHeight(bx, bz) > -50:
        by = mc.getHeight(bx, bz)
    if mc.getBlock(bx, by, bz) in [block.GLASS.id, block.GLASS_PANE.id]:
        mc.setBlock(bx, by, bz, block.AIR.id)
        continue
    mc.setBlock(bx, by, bz, block.GRAVEL.id)
    mc.setBlocks(bx, by-1, bz, bx, -60, bz, block.AIR.id)

disasters = [earthquake]
def main(disasters, mc):
    baseed = random.randint(1, 10000)
    mc.postToChat('Press button for earthquake!')
    while True:
        if GPIO.input(23) == GPIO.HIGH: #Look for button press
            t = random.randint(5, 60)
            t = 5
            time.sleep(t)
            random.seed(baseed + t)
            baseed = random.randint(1, 10000)
            random.shuffle(disasters)
            disaster = random.choice(disasters)
            ppos = mc.player.getTilePos()
            disaster(ppos.x, ppos.z)

try:
    import _thread as thread
except ImportError:
    import thread
thread.start_new_thread(main, (disasters, mc))

```

## 1.4 Dolina (Sinkhole)

### 1.4.1 Equipamento e cenário

Esta atividade vai apresentar uma dolina ao mundo Minecraft Pi, que vai surgir após um sismo. À semelhança da primeira atividade, os estudantes vão ligar um botão ao GPIO Raspberry Pi e programá-lo para criar um sismo e uma dolina depois de este ser pressionado.

Para esta atividade vai utilizar os seguintes componentes eletrónicos e periféricos:

1. 1 x Placa de ensaio
2. 1 x Botão de pressão com tampa
3. 1 x Resistor 220 Ohm
4. 2 x Cabos de ligação (Macho/Fêmea)
5. 1 x Molde da caixa em cartolina

Depois de tudo ligado, os estudantes devem carregar o mapa "Fishing Town" e o script Python "sinkhole.py" que pode ser encontrado no ambiente de trabalho da consola STEM4CLIM8.

Depois, cada vez que o botão for premido, vai ocorrer um sismo no mundo Minecraft Pi, seguido de uma dolina que vai destruir edifícios e afundar o solo como uma catástrofe da vida real.

Após a atividade, deverá seguir-se uma discussão com os estudantes sobre o impacto dos sismos e das dolinas.

### 1.4.2 Circuito eletrônico e bloco físico

Ligar um botão de pressão à GPIO do Raspberry Pi é um processo muito simples, como se pode ver na Figura 4:

1. Ligar o lado direito do botão de pressão ao pino de entrada 16 (GPIO23) usando um cabo de ligação macho-fêmea.
2. Ligar o lado esquerdo do botão de pressão ao resistor de 220 Ohm.
3. Ligar o outro lado do resistor ao pino 1 de 3,3V.
4. O circuito está pronto.

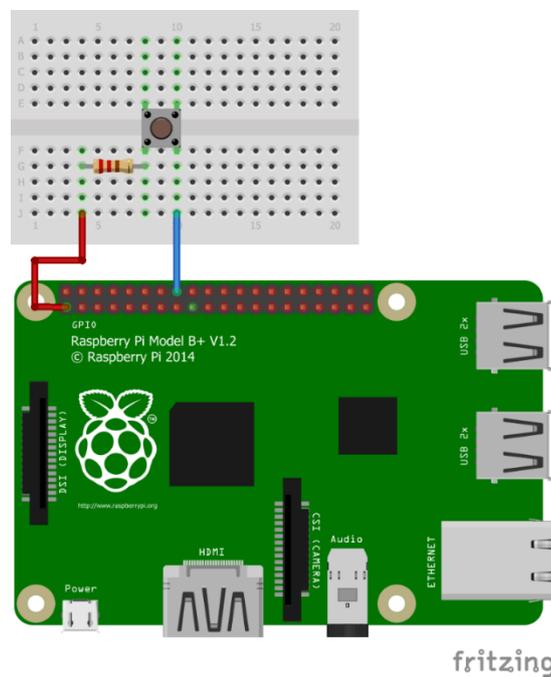


Figura 4 Ligação de um botão de pressão à GPIO utilizando uma placa de ensaio e dois cabos de ligação.

Para compreender os diferentes pinos GPIO, deve observar a Figura 5, que descreve as propriedades de cada pino.

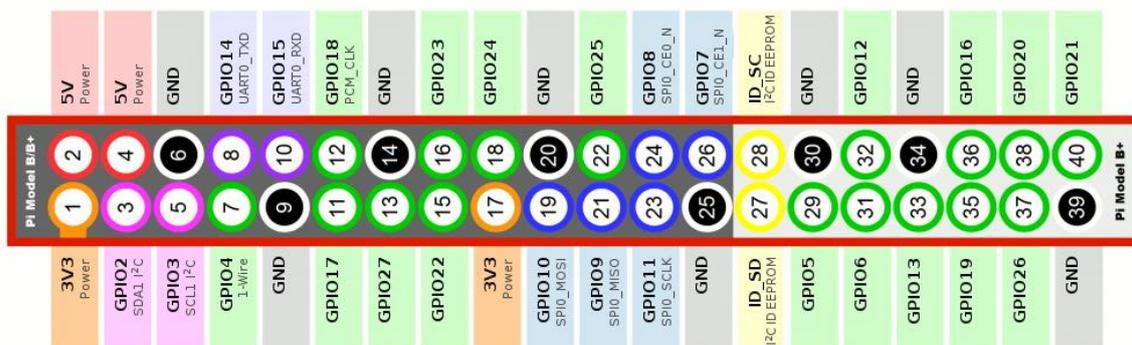
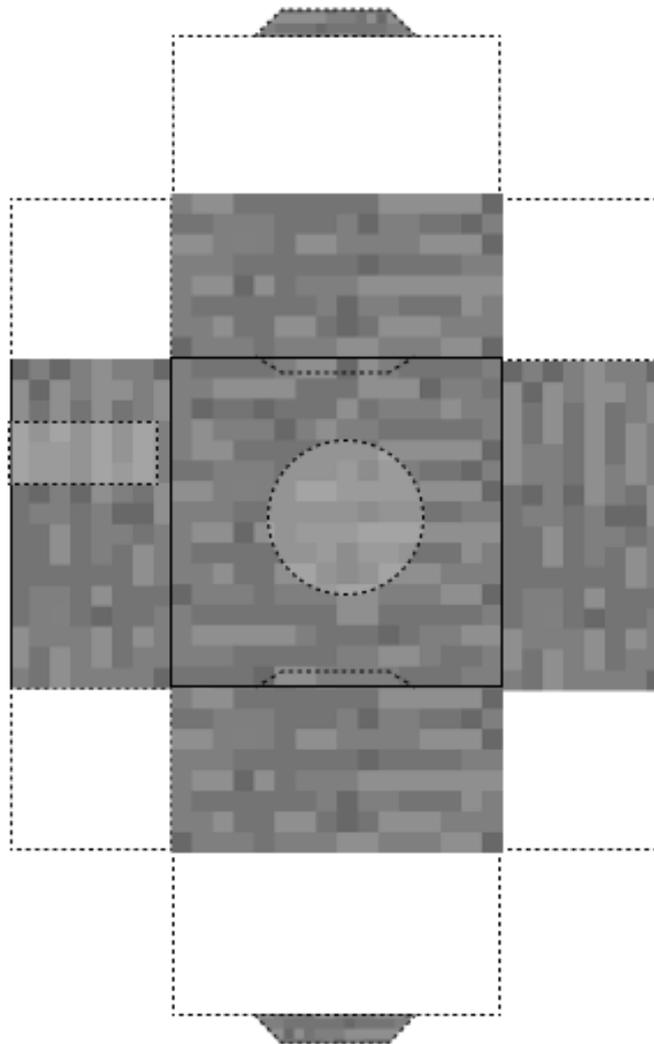


Figura 5 Pinos da GPIO Raspberry Pi

Depois de criar o circuito, é necessário fazer a montagem de uma caixa e colocar o circuito na mesma. Para o efeito, imprima a imagem seguinte, em Cartolina A4 branca e depois recorte pelas margens indicadas.



**Figura 6 Molde para impressão, da caixa para o bloco de computação física com botão.**

### 1.4.3 Programação Python

Basta abrir Thonny Python no ambiente de trabalho Raspberry Pi, depois carregar o script "sinkhole.py" que pode ser encontrado no ambiente de trabalho ou copiar e colar diretamente para o compilador Python à sua escolha. Se decidir ligar o botão de pressão a um pino GPIO diferente, não se esqueça de alterar o número GPIO no script abaixo (linhas 10 e 57).

```
import RPi.GPIO as GPIO
import mcpi.minecraft as minc
import mcpi.block as block
mc = minc.Minecraft.create()
import random, time

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

```

def sinkhole(x, z):
    mc.postToChat('Earthquake!')
    y = mc.getHeight(x, z)
    endtime = time.time() + 60
    nearthtime = time.time()
    while endtime > time.time():
        if time.time() > nearthtime:
            nearthtime = time.time() + 5
        ppos = mc.player.getPos()
        if ppos.x < x+50 and ppos.x > x-50:
            if ppos.y < y+50 and ppos.y > -60:
                if ppos.z < z+50 and ppos.z > z-50:
                    mc.player.setPos(ppos.x, ppos.y, ppos.z)
        bx = random.randint(x-50, x+50)
        by = y
        bz = random.randint(z-50, z+50)
        if mc.getHeight(bx, bz) > -50:
            by = mc.getHeight(bx, bz)
        if mc.getBlock(bx, by, bz) in [block.GLASS.id, block.GLASS_PANE.id]:
            mc.setBlock(bx, by, bz, block.AIR.id)
            continue
        mc.setBlock(bx, by, bz, block.GRAVEL.id)
        mc.setBlocks(bx, by-1, bz, bx, -60, bz, block.AIR.id)

    mc.postToChat('Sinkhole!')
    blks = []
    y = mc.getHeight(x, z)
    xdists = random.randint(1, 20)
    for bx in range(-xdists, xdists+1):
        zdists = random.randint(1, 20)
        for bz in range(-zdists, zdists+1):
            blks.append([x+bx, z+bz])
    for blk in blks:
        mc.setBlocks(blk[0], mc.getHeight(blk[0], blk[1]), blk[1], blk[0], -60, blk[1],
        block.AIR.id)
        mc.setBlocks(blk[0], -55, blk[1], blk[0], -60, blk[1], block.LAVA.id)
    for blk in blks:
        mc.setBlock(blk[0], y, blk[1], block.GRAVEL.id)

disasters = [sinkhole]
def main(disasters, mc):
    baseed = random.randint(1, 10000)
    mc.postToChat('Press button for disaster!')
    while True:
        if GPIO.input(23) == GPIO.HIGH: #Look for button press
            t = random.randint(5, 60)
            t = 5
            time.sleep(t)
            random.seed(baseed + t)
            baseed = random.randint(1, 10000)
            random.shuffle(disasters)
            disaster = random.choice(disasters)
            ppos = mc.player.getTilePos()
            disaster(ppos.x, ppos.z)

try:
    import _thread as thread
except ImportError:
    import thread
thread.start_new_thread(main, (disasters, mc))

```

## 1.5 Erupção vulcânica

### 1.5.1 Equipamento e cenário

Esta atividade vai apresentar uma erupção vulcânica ao mundo Minecraft Pi. Mais uma vez, os estudantes vão ligar um botão à GPIO Raspberry Pi e programá-lo para, depois de este ser premido, criar uma erupção vulcânica.

Para esta atividade vai utilizar os seguintes componentes eletrónicos e periféricos:

1. 1 x Placa de ensaio
2. 1 x Botão de pressão com tampa
3. 1 x Resistor 220 Ohm
4. 2 x Cabos de ligação (Macho/Fêmea)
5. 1 x Construção de caixa em cartolina

Depois de tudo ligado, os estudantes devem carregar o mapa "Fishing Town" e o script Python "vulcano.py" que pode ser encontrado no ambiente de trabalho da consola STEM4CLIM8.

Depois, cada vez que o botão for premido, um vulcão vai aparecer no mundo Minecraft Pi, e entrará em erupção, a lava vai começar a fluir em direção à cidade piscatória. Passado algum tempo, a lava transforma-se em pedra, semelhante a uma erupção vulcânica real. Os edifícios e outros blocos desaparecem, e as árvores e ervas pegam fogo.

Após a atividade, deve seguir-se uma discussão com os estudantes sobre o impacto nos edifícios e sobre a população das cidades, uma vez que as catástrofes naturais não têm limites e podem ameaçar as nossas vidas e as condições de vida.

### 1.5.2 Circuito eletrónico e bloco físico

Ligar um botão de pressão à GPIO do Raspberry Pi é um processo muito simples, como mostra a Figura 7:

1. Ligar o lado direito do botão de pressão ao pino de entrada 16 (GPIO23) usando um cabo de ligação macho-fêmea.
2. Ligar o lado esquerdo do botão de pressão ao resistor de 220 Ohm.
3. Ligar o outro lado do resistor ao pino 1 de 3,3V.
4. O circuito está pronto.

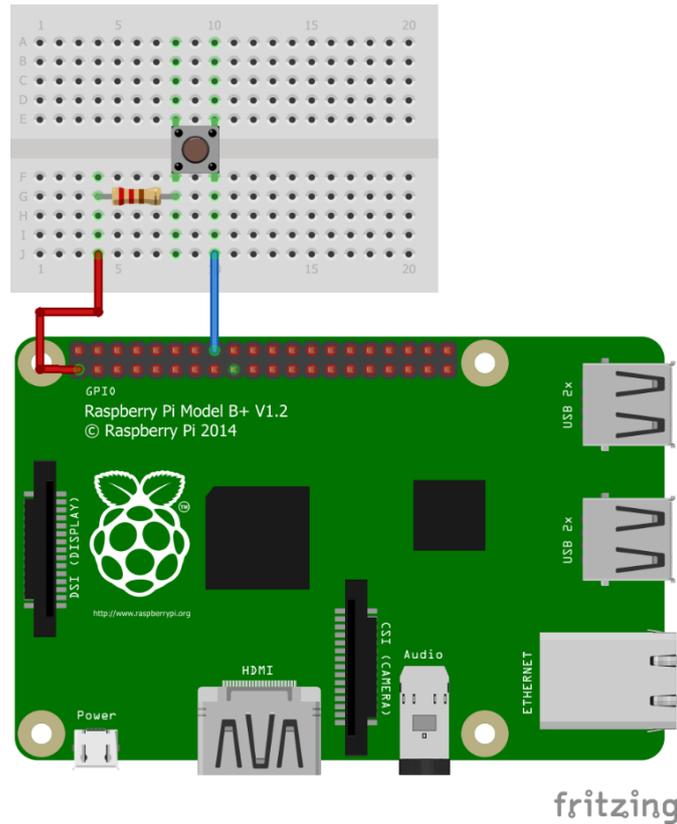


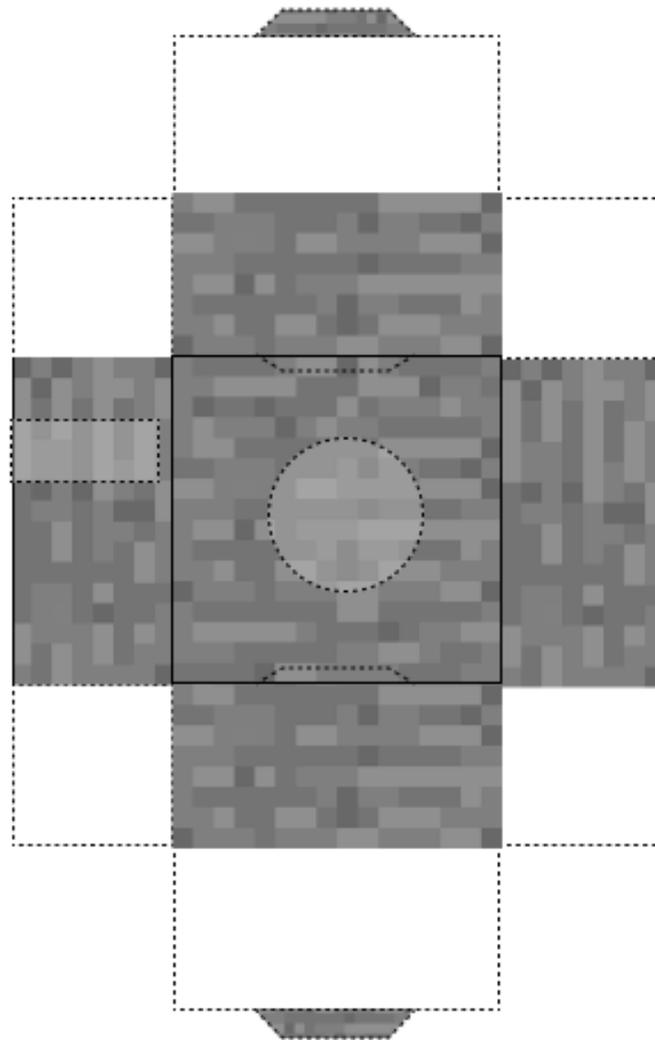
Figura 7 Ligação de um botão de pressão à GPIO utilizando uma placa de ensaio e dois cabos de ligação.

Para compreender os diferentes pinos GPIO, deve observar a Figura 8, que descreve as propriedades de cada pino.

	5V Power	5V Power	GND	GPIO14 UART0_TXD	GPIO15 UART0_RXD	GPIO18 PCM_CLK	GND	GPIO23	GPIO24	GND	GPIO25	GPIO8 SPI0_CEO_N	GPIO7 SPI0_CEL_N	ID_SD I <sup>2</sup> C ID EEPROM	GND	GPIO12	GND	GPIO16	GPIO20	GPIO21																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
3V3 Power	GPIO2 SDA1 I <sup>2</sup> C	GPIO3 SCL1 I <sup>2</sup> C	GPIO4 1-Wire	GND	GPIO17	GPIO27	GPIO22	3V3 Power	GPIO10 SPI0_MOSI	GPIO9 SPI0_MISO	GPIO11 SPI0_SCLK	GND	ID_SD I <sup>2</sup> C ID EEPROM	GPIO5	GPIO6	GPIO13	GPIO19	GPIO26	GND																				

Figura 8 Pinos do GPIO Raspberry Pi

Depois de criar o circuito, é necessário fazer a montagem da caixa e colocar o circuito na mesma. Para o efeito, imprima a imagem seguinte, em Cartolina A4 branca e depois recorte pelas margens indicadas.



**Figura 9 Molde para impressão, da caixa para o bloco de computação física com botão.**

### 1.5.3 Programação Python

Basta abrir Thonny Python no ambiente de trabalho Raspberry Pi, depois carregar o script "vulcano.py" que pode ser encontrado no ambiente de trabalho ou copiar e colar diretamente para o compilador Python à sua escolha. Se decidir ligar o botão de pressão a um pino GPIO diferente, não se esqueça de alterar o número GPIO no script abaixo (linhas 9 e 13).

```
import RPi.GPIO as GPIO
import mcpi.minecraft as minecraft
import mcpi.block as block
import random, time

mc = minecraft.Minecraft.create()

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

mc.postToChat('Press button for eruption!')
```

```
while True:
    if GPIO.input(23) == GPIO.HIGH: #Look for button press
        mc.postToChat("Eruption")
        x, y, z = mc.player.getPos()
        mc.setBlocks(x, y, z, x+20, y+15, z+20, block.AIR)

        height = 10
        center = x+15, y, z+15

        for i in range(height): #Create the base
            size = height - i
            mc.setBlocks(center[0] - size, center[1] + i, center[2] - size, center[0] +
size, center[1] + i, center[2] + size, block.STONE)

        while True:
            mc.setBlock(center[0], center[1]+height, center[2], block.LAVA_FLOWING)
            time.sleep(1)
```

## 1.6 Onda de calor

### 1.6.1 Equipamento e Cenário

Esta atividade vai apresentar uma onda de calor no mundo da Minecraft Pi. Os estudantes vão ligar um sensor ultrassónico à GPIO e programá-lo para criar uma onda de calor quando um objeto se aproximar do sensor.

A atividade necessita do equipamento seguinte:

1. 1 x Placa de ensaio
2. 1 x Sensor ultrassónico HC-SR04
3. 3 x Resistores 1k Ohm
4. 4 x Cabos de ligação (Macho/Fêmea)
5. 1 x Molde da caixa em cartolina

Depois de tudo ligado, os estudantes devem carregar o mapa "Fishing Town" e o script Python "heatwave.py" que pode encontrar no ambiente de trabalho da consola STEM4CLIM8.

À medida que o "sol" se aproxima da terra, vai aparecer uma onda de calor que vai causar a morte de árvores, plantas e ervas, semelhante a uma catástrofe real.

Após a atividade, deve seguir-se uma discussão com os estudantes sobre o fenómeno do aumento da temperatura e os resultados negativos para o ambiente.

### 1.6.2 Circuito eletrónico e execução da atividade

A primeira coisa a fazer é criar o circuito e ligar o sensor HC-SR04 aos pinos GPIO do Raspberry Pi. Antes de prosseguir, é necessário encerrar o Raspberry Pi e desligá-lo da tomada. O circuito completo está no diagrama esquemático que se segue.

O sensor de distância ultrassónico HC-SR04 possui 4 pinos: potência (VCC), ativador (TRIG), eco (ECHO), e terra (GND). O pino da potência será ligado ao pino de 5V da Raspberry Pi, o ativador será atribuído a um pino GPIO como saída (pino 4), ao eco será atribuído um pino GPIO como entrada (pino 18), e o pino terra será ligado a um pino terra da Raspberry Pi GPIO.

O programa vai operar de forma que, sempre que um objeto se aproxima do sensor, o sensor emite um pulso ultrassom e o programa verifica o tempo que demora a receber o eco de volta (ou seja, quanto tempo passou enquanto as ondas sonoras foram emitidas, atingem o objeto em frente do sensor, fazem ricochete e voltam ao sensor).

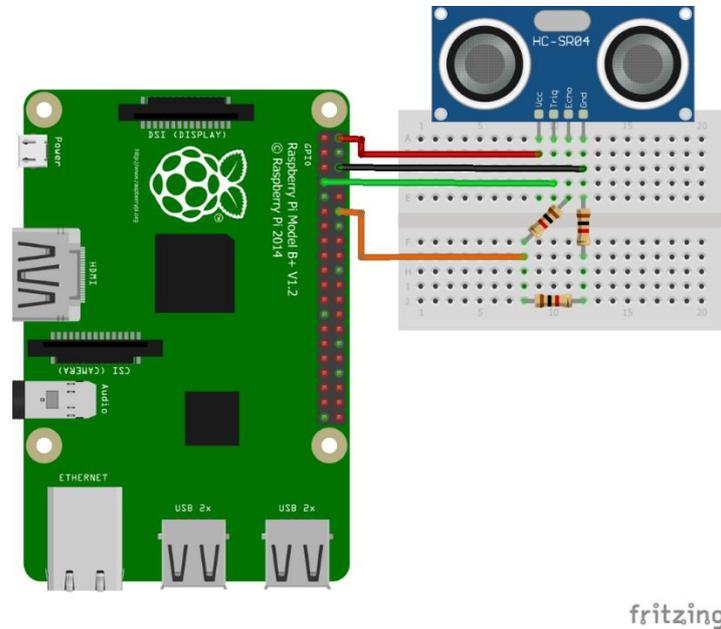


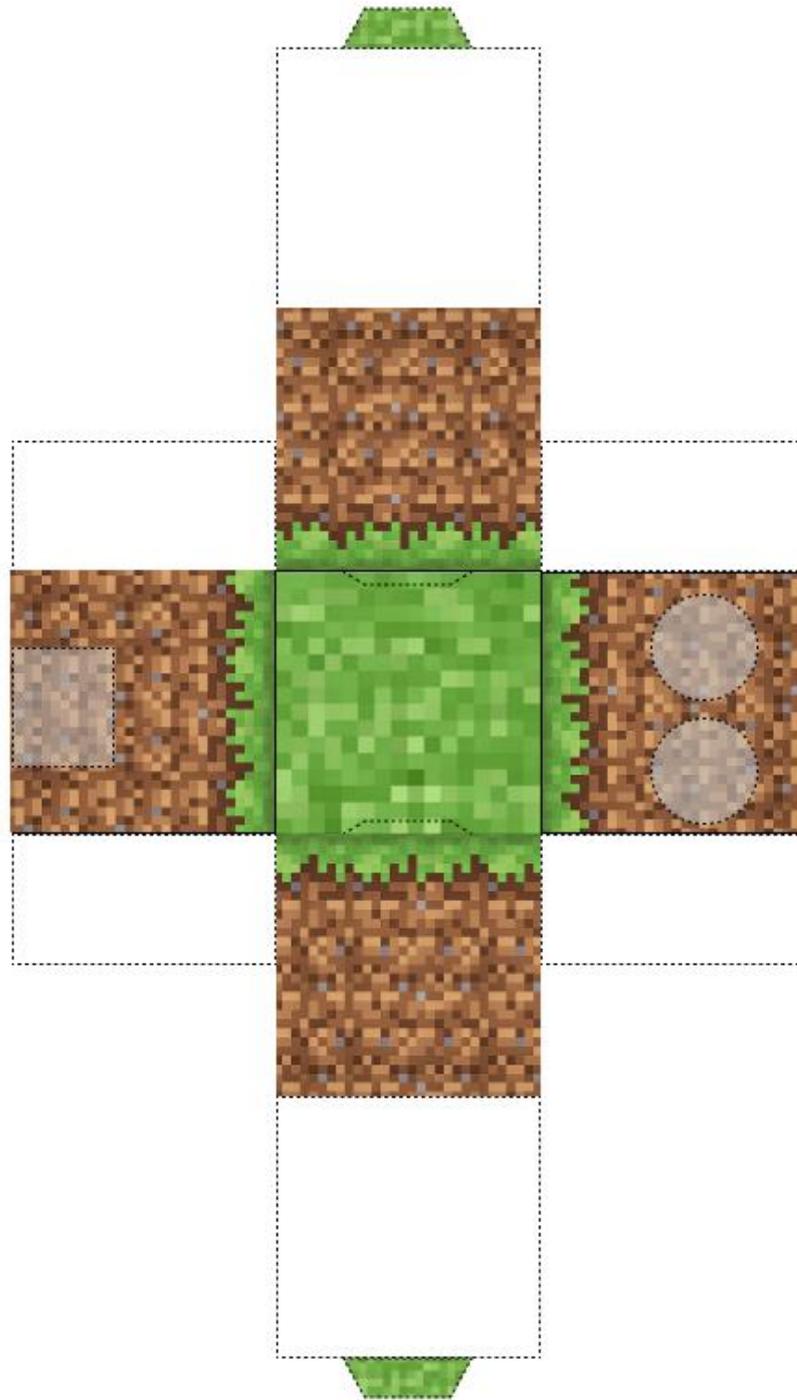
Figura 10 Diagrama esquemático do circuito com sensor de distância ligado aos pinos GPIO.

Para compreender os diferentes pinos GPIO, deve observar a Figura 11, que descreve as propriedades de cada pino.



Figura 11 Pinos GPIO Raspberry Pi

Depois de criar o circuito, é necessário fazer a montagem da caixa e colocar o circuito na mesma. Para o efeito, imprima a imagem seguinte, em Cartolina A4 branca e depois recorte pelas margens indicadas.



**Figure 12 Molde para impressão, da caixa para o bloco de computação física com sensor de distância.**

A caixa de cartolina representa a "Terra". Será necessário também, outro objeto de tamanho semelhante, que represente o "Sol". No pacote STEM4CLI8, vai encontrar uma esfera impressa em 3D que pode usar. Pode também imprimir em 3D o seu próprio "Sol" utilizando os ficheiros .stl que estão no ambiente de trabalho do Raspberry Pi.

### 1.6.3 Programação Python

Quando terminar o circuito, pode ligar o Raspberry Pi e iniciar o Thonny Python, depois carregar o script "heatwave.py" que pode ser encontrado no ambiente de trabalho ou copiar e colar diretamente para o compilador Python à sua escolha. Em suma, quando executar o programa, o sensor de distância HC-SR04 vai começar a emitir ultrassons. Cada vez que o "sol" se aproximar do sensor "Terra", o fenómeno da onda de calor irá acontecer no mundo Minecraft Pi. Iremos ver as árvores, flores e plantas incendiarem-se e a tornarem-se cinzas.

```
import RPi.GPIO as GPIO
import mcpi.minecraft as minc
import mcpi.block as block
mc = minc.Minecraft.create()
import random, time

GPIO.setwarnings(False)
GPIO.cleanup()
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN) #pin for push button

#defining TRIG and ECHO pins
TRIG = 4
ECHO = 18
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)

def heatwave(x, z):
    mc.postToChat('Heatwave!')
    y = mc.getHeight(x, z)
    endtime = time.time() + random.randint(50, 90)
    while time.time() < endtime:
        blkid = block.AIR.id
        while blkid == block.AIR.id:
            bx = random.randint(x-10, x+10)
            by = random.randint(y, y+10)
            bz = random.randint(z-10, z+10)
            blkid = mc.getBlockWithData(bx, by, bz).id
        blk = blkid
        blkd = mc.getBlockWithData(bx, by, bz).data
        if blkid == block.GRASS.id:
            blk = block.DIRT.id
            blkd = 0
        elif blkid in [block.WATER.id, block.WATER_FLOWING.id, block.WATER_STATIONARY.id]:
            blk = block.WATER.id
            blkd = 1
        elif blkid == block.LEAVES.id:
            blk = block.COBWEB.id
            blkd = 0
        elif blkid == block.WOOD.id:
            blk = block.LAVA_STATIONARY.id
            blkd = 1
        mc.setBlock(bx, by, bz, blk, blkd)

disasters = [heatwave]
def main(disasters, mc):
    baseed = random.randint(1, 10000)
    mc.postToChat('Push button for sensor to work!')
    mc.postToChat('Move "sun" closer to sensor, and wait for Heatwave!')
    #emit a burst of ultrasound
    GPIO.output(TRIG, True)
    time.sleep(0.00001)
    GPIO.output(TRIG, False)
    StartTime = time.time()
```

```
StopTime = time.time()
while True:
    if GPIO.input(23) == GPIO.HIGH:
        while GPIO.input(ECHO) == False:
            StartTime = time.time()
        while GPIO.input(ECHO) == True:
            StopTime = time.time()
        TimeElapsed = StopTime - StartTime
        distance = (TimeElapsed * 34300) / 2
        while distance < 15:
            print (distance)
            t = random.randint(5, 120)
            t = 5
            time.sleep(t)
            random.seed(baseed + t)
            baseed = random.randint(1, 10000)
            random.shuffle(disasters)
            disaster = random.choice(disasters)
            ppos = mc.player.getTilePos()
            disaster(ppos.x, ppos.z)
try:
    import _thread as thread
except ImportError:
    import thread
thread.start_new_thread(main, (disasters, mc))
```

## 1.7 Inundação

### 1.7.1 Equipamento e Cenário

Esta atividade irá apresentar uma inundação ao mundo da Minecraft Pi. Os estudantes vão ligar um sensor de chuva à GPIO e programá-lo para criar uma inundação cada vez que o sensor tocar em água.

A ideia é criar um bloco de computação física utilizando o seguinte equipamento:

1. 1 x Placa de ensaio
2. 1 x Sensor detetor de chuva
3. 3 x Cabos de ligação (Macho/Fêmea)
4. 2 x Cabos de ligação (Fêmea /Fêmea)
5. 1 x Molde da caixa em cartolina

Depois de tudo ligado, os estudantes devem carregar o mapa "Fishing Town" e o script Python "flood.py" que pode encontrar no ambiente de trabalho da consola STEM4CLIM8.

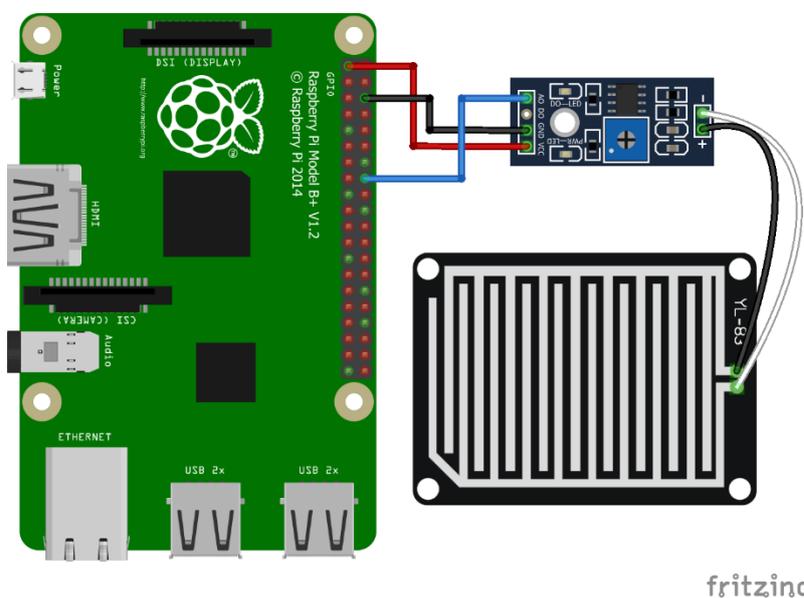
Depois, cada vez que for lançada água sobre o sensor, deverá ocorrer uma inundação no mundo Minecraft Pi, causando a submersão de tudo. A chuva forte será referenciada com uma mensagem no mundo Minecraft que irá criar um Geyser (inundação).

Após a atividade, deverá seguir-se uma discussão com os estudantes sobre as más práticas e as razões pelas quais a inundação ocorreu após a chuva forte.

### 1.7.2 Circuito eletrónico e execução da atividade

Ligar o sensor detetor de chuva à GPIO do Raspberry Pi é um processo relativamente simples que se pode ver na figura 9:

1. A placa de controlo do sensor tem 4 pinos que podem ser ligados à GPIO, ou seja, tensão (VCC), terra (GND), saída digital (DO) e saída analógica (AO).
2. É necessário fazer as ligações seguintes através de uma placa de ensaio ou diretamente na GPIO:
  - a) VCC a 5V (pino 2)
  - b) GND para GND (pino 4)
  - c) DO a GPIO23 (pin16)
3. De seguida, ligar a placa do sensor de deteção à placa de controlo, utilizando dois cabos de ligação, ligar da seguinte forma:
  - a. O + da placa de controlo com o + da placa de deteção.
  - b. O - da placa de controlo com o - da placa de deteção.
4. O circuito está pronto.



fritzing

Figura 13 Esquema do sensor detetor de chuva ligado aos pinos GPIO da Raspberry Pi.

Para compreender os diferentes pinos GPIO, deve observar a Figura 14, que descreve as propriedades de cada pino.

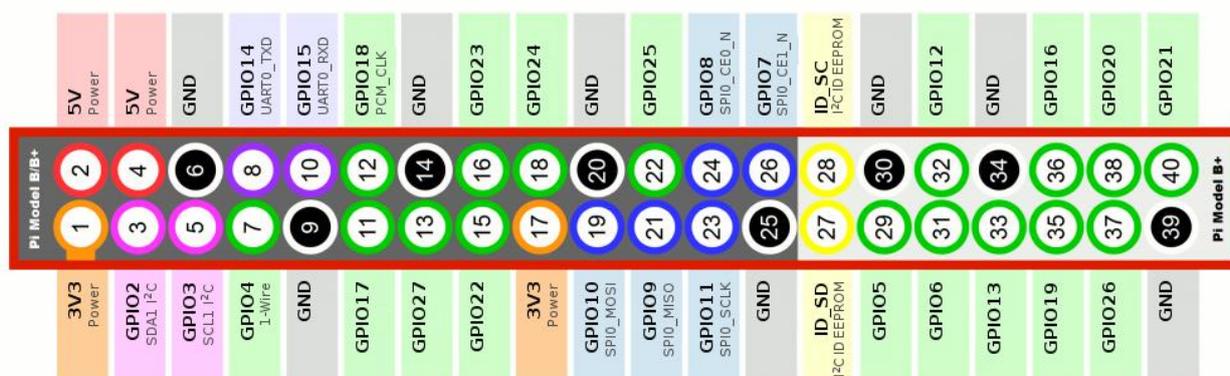


Figura 14 Pinos da GPIO do Raspberry Pi

O sensor detetor de chuva permite enviar, sinais analógicos e digitais, para o dispositivo de controlo. Neste caso, o computador Raspberry Pi apenas lê sinais digitais. Assim, para esta atividade, só é necessária uma saída digital, uma vez que o programa é executado cada vez que o sensor deteta uma gota de água, causando uma inundação no mundo Minecraft Pi.

Depois de criar o circuito, é necessário fazer a montagem da caixa e colocar o circuito na mesma. Para o efeito, imprima a imagem seguinte, em Cartolina A4 branca e depois recorte pelas margens indicadas.

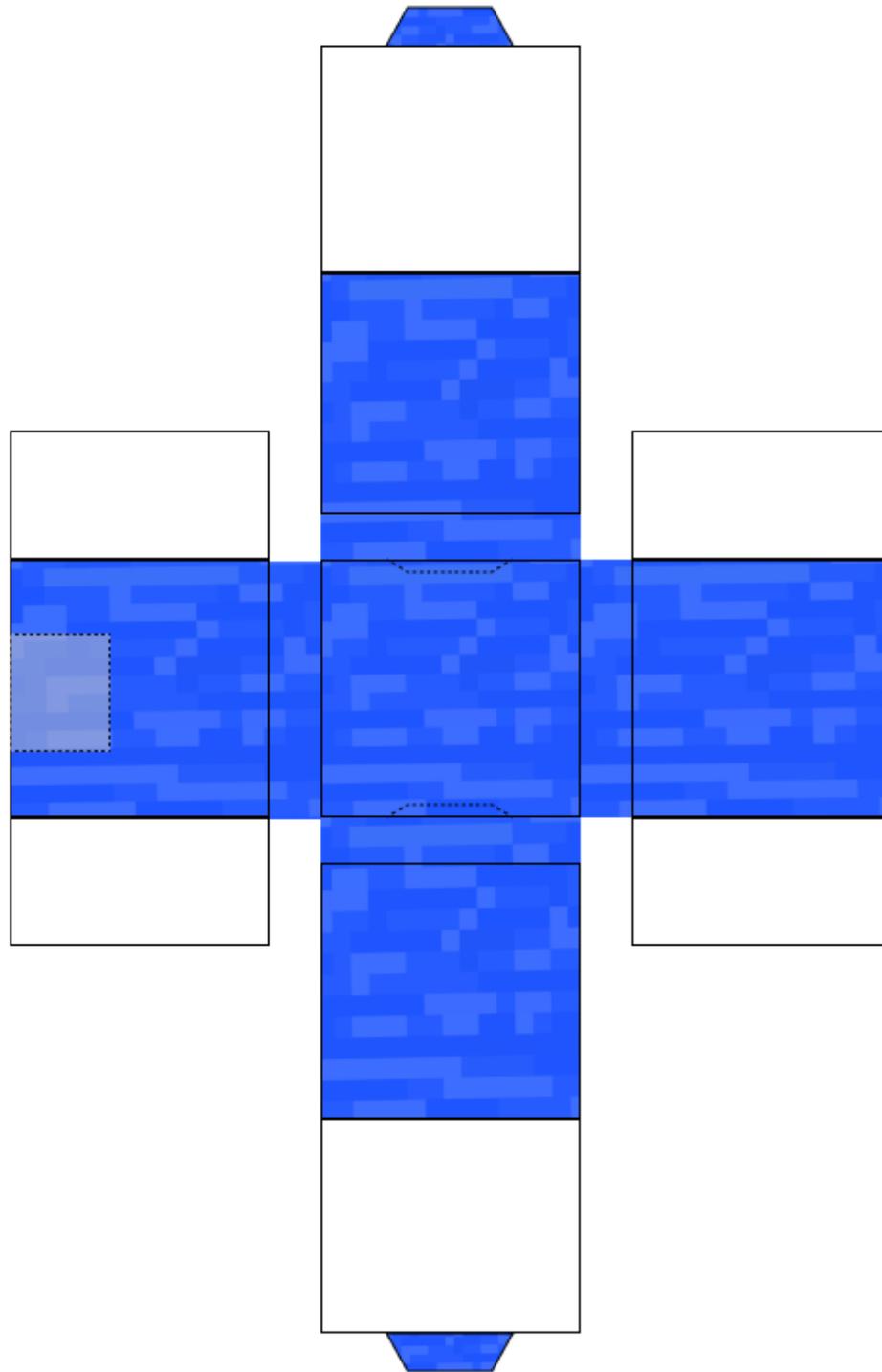


Figura 15 Molde para impressão, da caixa para o bloco de computação física com sensor detetor de chuva.

### 1.7.3 Programação Python

```
import RPi.GPIO as GPIO
import mcpi.minecraft as minc
import mcpi.block as block
mc = minc.Minecraft.create()
import random, time

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN)

def geyser(x, z):
    mc.postToChat('Flood!')
    y = mc.getHeight(x, z)
    mc.setBlocks(x-2, y+5, z-2, x+2, -60, z+2, block.WATER.id)
    time.sleep(25)
    mc.setBlocks(x-2, y+5, z-2, x+2, -60, z+2, block.AIR.id)

disasters = [geyser]
def main(disasters, mc):
    baseed = random.randint(1, 10000)
    mc.postToChat('Drop water on sensor for disaster!')
    while True:
        if GPIO.input(23) == GPIO.LOW: #Look for raindrop sensor activation
            t = random.randint(5, 120)
            t = 5
            time.sleep(t)
            random.seed(baseed + t)
            baseed = random.randint(1, 10000)
            random.shuffle(disasters)
            disaster = random.choice(disasters)
            ppos = mc.player.getTilePos()
            disaster(ppos.x, ppos.z)

try:
    import _thread as thread
except ImportError:
    import thread
thread.start_new_thread(main, (disasters, mc))
```

## 1.8 Conclusão

A utilização destas atividades vai permitir que às crianças experienciem as catástrofes naturais e compreender o papel das atividades humanas no agravamento destes fenómenos, nomeadamente as alterações climáticas, etc.

## Referências

Edição Minecraft Pi: SURVIVAL mode-ish. Obtido em: <https://teachwithict.weebly.com/computing-blog/minecraft-pi-edition-survival-mode-ish#sthash.S6XeUUDH.dpbs>

Como usar o Módulo Sensor de Chuva com Arduino & Raspberry Pi da SunFounder Maker Education. Obtido em: <https://www.youtube.com/watch?v=Xnisf0GP9bA&t=203s>

Utilizar um sensor de distância Raspberry Pi (sensor ultrassónico HC-SR04). Obtido em: <https://tutorials-raspberrypi.com/raspberry-pi-ultrasonic-sensor-hc-sr04/>