

FLUXOGRAMAS E MAPA MENTAL

Fellype

Cauã

Davi P.

Guilherme

Alexandre

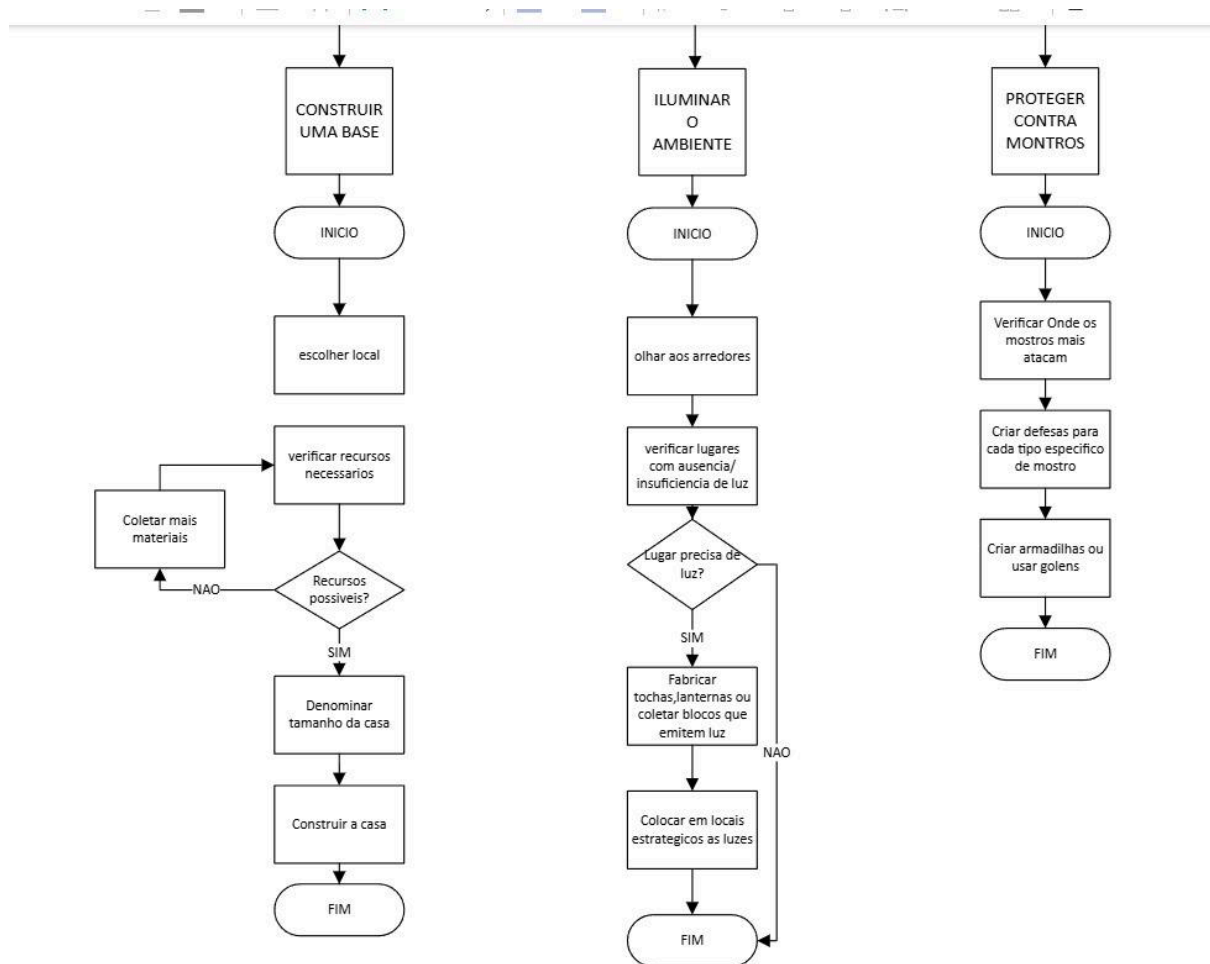
Wenso

Mapa Mental



FLUXOGRAMAS

.Abrigo e Segurança



CÓDIGOS

```
programa {
    funcao inicio() {

        inteiro madeira
        inteiro pedregulho

        escreva("Você precisa contruir uma base!", "\n", "É preciso 40
madeiras refinadas e 20 blocos de pedregulho.\n")
        escreva("\nDigite sua quantidade de madeira: ")
        leia(madeira)
```

```

    enquanto (madeira < 40){
        escreva("\nConseguindo madeira...\n")
        madeira = madeira + 1
        escreva("Madeira atual: ", madeira, "\n")
    }

    se(madeira >= 40){
        escreva("\nQuantidade de madeira suficiente!\n")
        escreva("\nDigite sua quantidade de pedregulho: ")
        leia(pedregulho)
    }

    enquanto (pedregulho < 20){
        escreva("\nConseguindo pedregulho...\n")
        pedregulho = pedregulho + 1
        escreva("Pedregulho atual: ", pedregulho, "\n")
    }

    se(pedregulho >= 20){
        escreva("\nQuantidade de pedregulho suficiente!\n")
    }

    se(madeira >= 40 e pedregulho >= 20){
        escreva("\n*Localizando o melhor lugar para base...*", "\n",
        "*Construindo a base...*", "\n", "*Estrutura feita!*\n")
    }
}
}

```

```

programa {
    funcao inicio() {

        inteiro madeira
        inteiro pedregulho

        escreva("Você precisa contruir uma base!", "\n", "É preciso 40
madeiras refinadas e 20 blocos de pedregulho.\n")
        escreva("\nDigite sua quantidade de madeira: ")
        leia(madeira)
    }
}

```

```

    enquanto (madeira < 40){
        escreva("\nConseguindo madeira...\n")
        madeira = madeira + 1
        escreva("Madeira atual: ", madeira, "\n")
    }

    se(madeira >= 40){
        escreva("\nQuantidade de madeira suficiente!\n")
        escreva("\nDigite sua quantidade de pedregulho: ")
        leia(pedregulho)
    }

    enquanto (pedregulho < 20){
        escreva("\nConseguindo pedregulho...\n")
        pedregulho = pedregulho + 1
        escreva("Pedregulho atual: ", pedregulho, "\n")
    }

    se(pedregulho >= 20){
        escreva("\nQuantidade de pedregulho suficiente!\n")
    }

    se(madeira >= 40 e pedregulho >= 20){
        escreva("\n*Localizando o melhor lugar para base...*", "\n",
        "*Construindo a base...*", "\n", "*Estrutura feita!*\n")
    }
}
}

```

```

programa {
    funcao inicio() {

        inteiro madeira

        escreva("Você precisa proteger sua base!", "\n", "Conte 3 blocos a
partir da porta e cave com 5 blocos de profundidade em volta dela.\n")
        escreva("\n*Contando blocos...*", "\n", "*Cavando em volta da
casa...*\n")
        escreva("\nPara proteger, precisará utilizar trapdoor.", "\n",
        "Digite sua quantidade de madeiras: ")
        leia(madeira)
    }
}

```

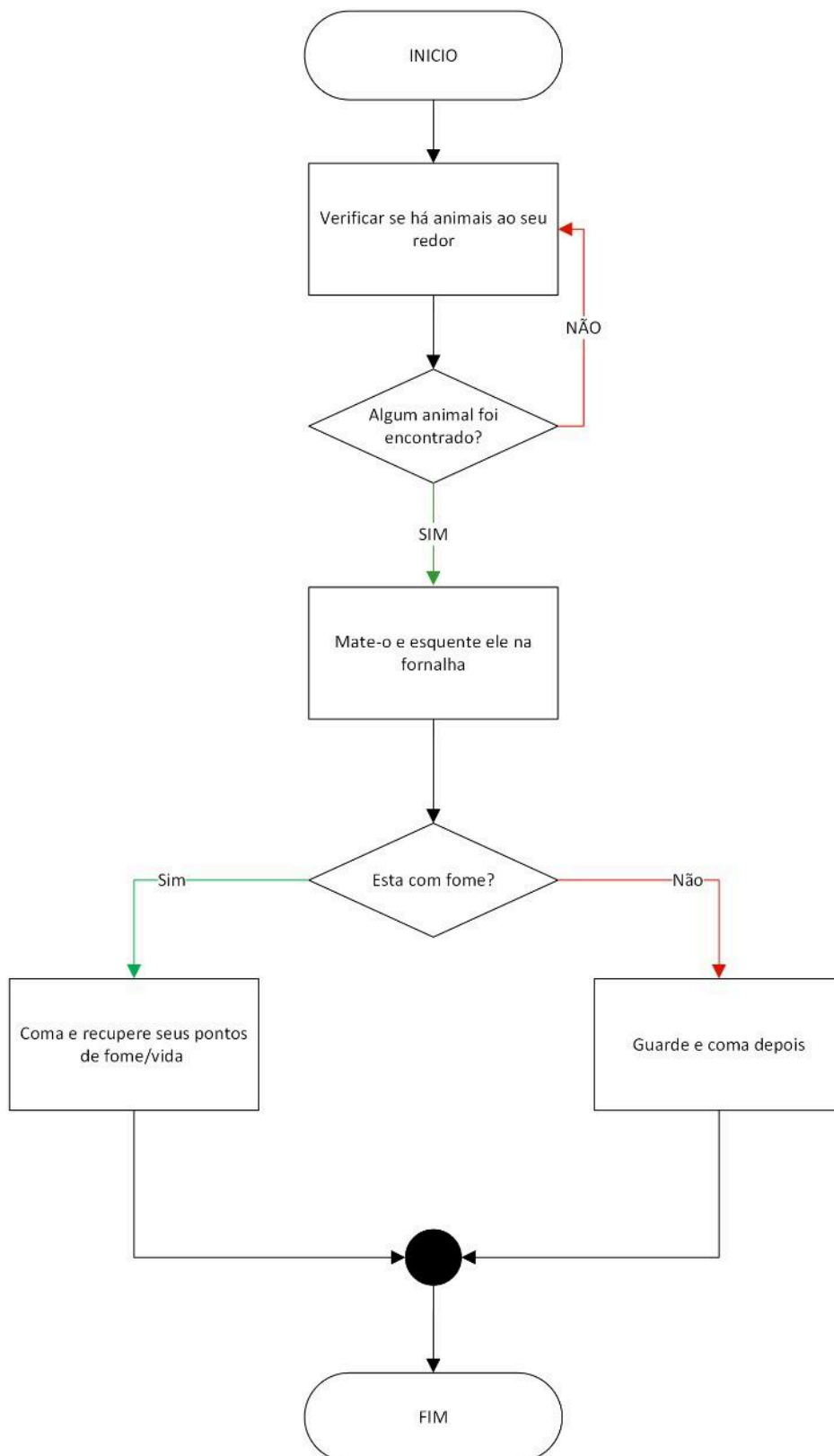
```
    enquanto(madeira < 42){
        escreva("\nConseguindo madeira...\n")
        madeira = madeira + 1
        escreva("Madeira atual: ", madeira, "\n")
    }

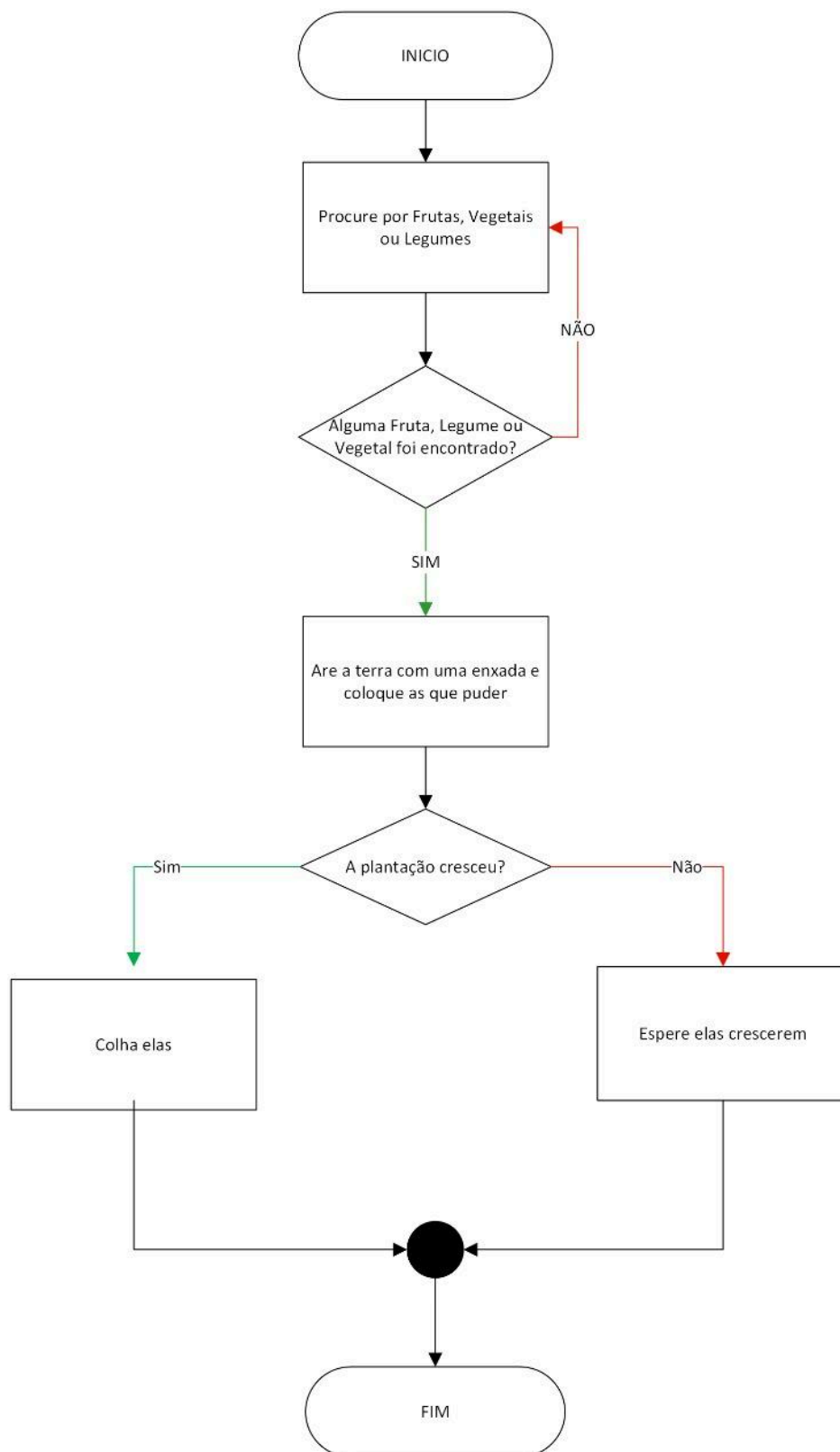
    se(madeira >= 42){
        escreva("\n*Quantidade de madeira suficiente!*\n")
        escreva("\nVocê agora precisa criar as trapdoors e colocá-las
\nna parte de cima dos buracos cavados.\n", "\n", "*Fazendo a criação
da trapdoor...*\n", "*Colocando em cima dos buracos...*\n", "*Proteção
feita!*\n")
    }

}

}
```

.Alimentação e saúde





CÓDIGOS

```
programa {
    funcao inicio() {

        logico resposta_alimento
        logico resposta_fome

        escreva("Você precisa coletar alimentos para sua fome e seus pontos
de vida.", "\n", "Procure por animais ao redor.\n")

        faca{
            escreva("\nProcurando por animais...", "\n", "Algun animal foi
encontrado? (sim ou não)\n")
            leia(resposta_alimento)
        } enquanto (resposta_alimento == falso)

        se (resposta_alimento == verdadeiro) {
            escreva("\nMate-o e esquente ele na fomalha.\n")
        }

        escreva("\nEstá com fome? (sim ou não)\n")
        leia(resposta_fome)

        se (resposta_fome == verdadeiro) {
            escreva("\nComa e recupere seus pontos de fome e/ou de vida\n")
        }

        senao {
            escreva("\nGuarde e coma depois.\n")
        }
    }
}
```

```
programa {
    funcao inicio() {

        logico resposta_plantacao
        logico resposta_crescer

        escreva("Você precisa começar uma plantação.\n", "Pegue sementes de
frutas, legumes ou vegetais.")
    }
}
```

```
faca{
    escreva("\nProcurando por frutas, legumes ou vegetais...", "\n",
"Alguma fruta, legume ou vegetal foi encontrado? (sim ou não)\n")
    leia(resposta_plantacao)
}enquanto(resposta_plantacao == falso)

se(resposta_plantacao == verdadeiro){
    escreva("\nAre a terra com uma enxada e coloque as que puder.\n")
}

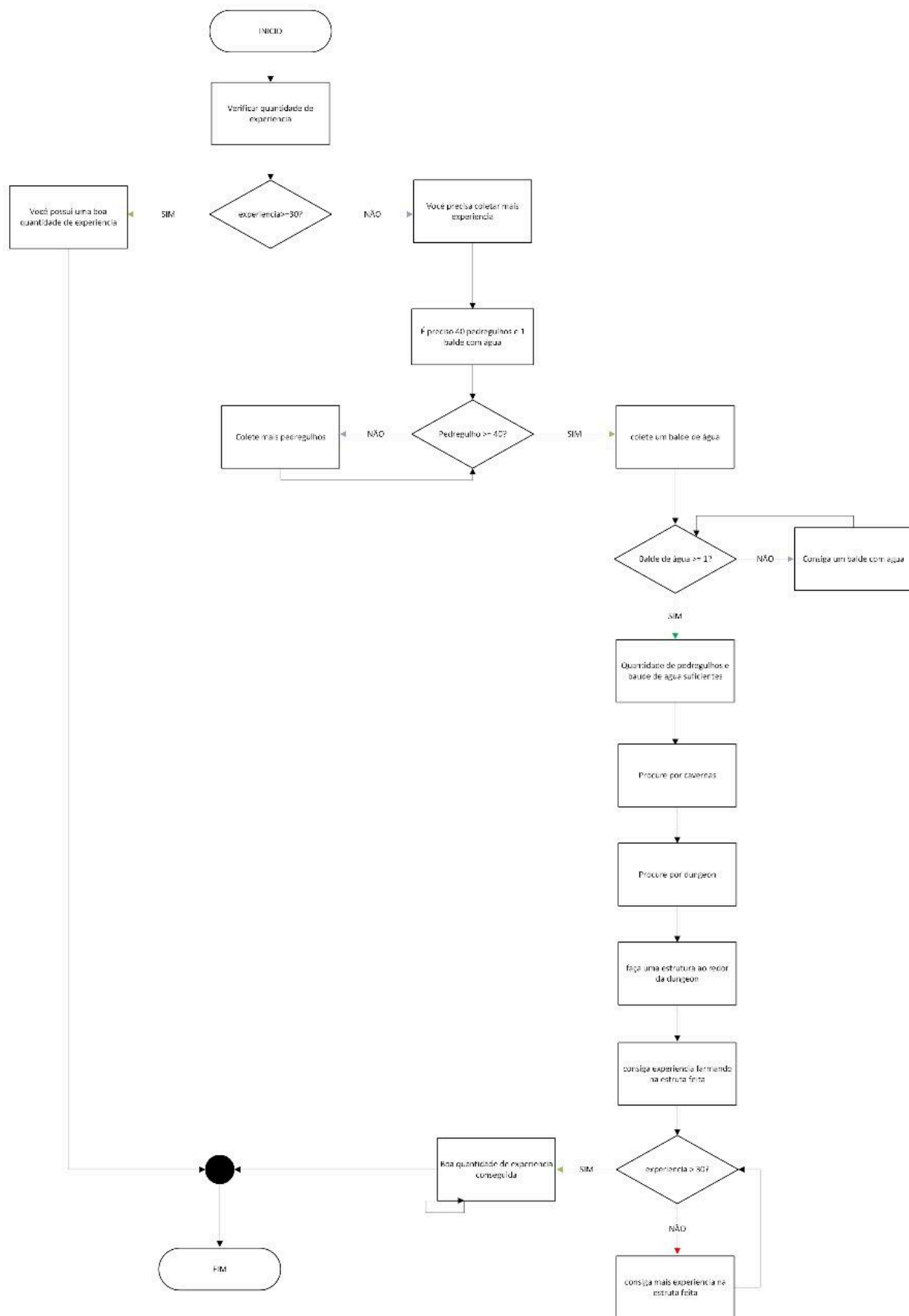
escreva("\nA plantação cresceu? (sim ou não)\n")
leia(resposta_crescer)

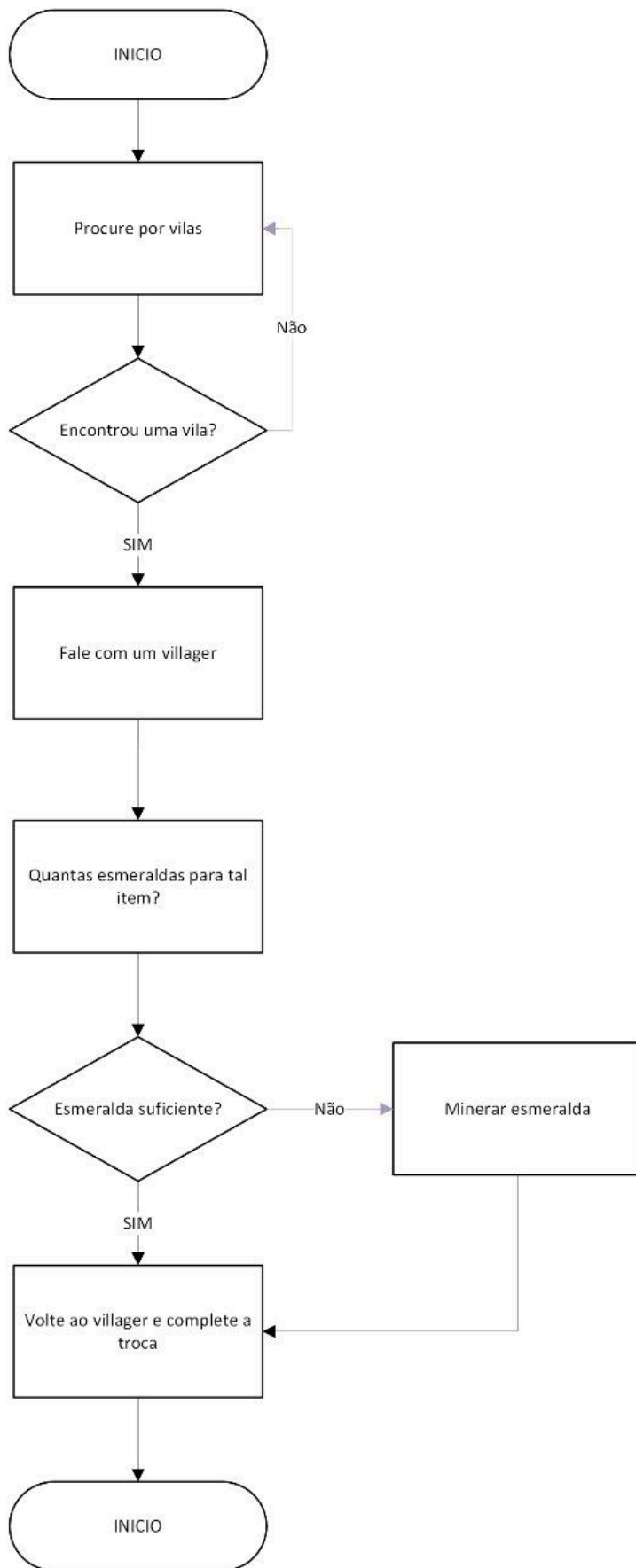
se(resposta_crescer == verdadeiro){
    escreva("\nColha elas.\n")}

senao{
    escreva("\nEspere elas crescerem\n")}

}
}
```

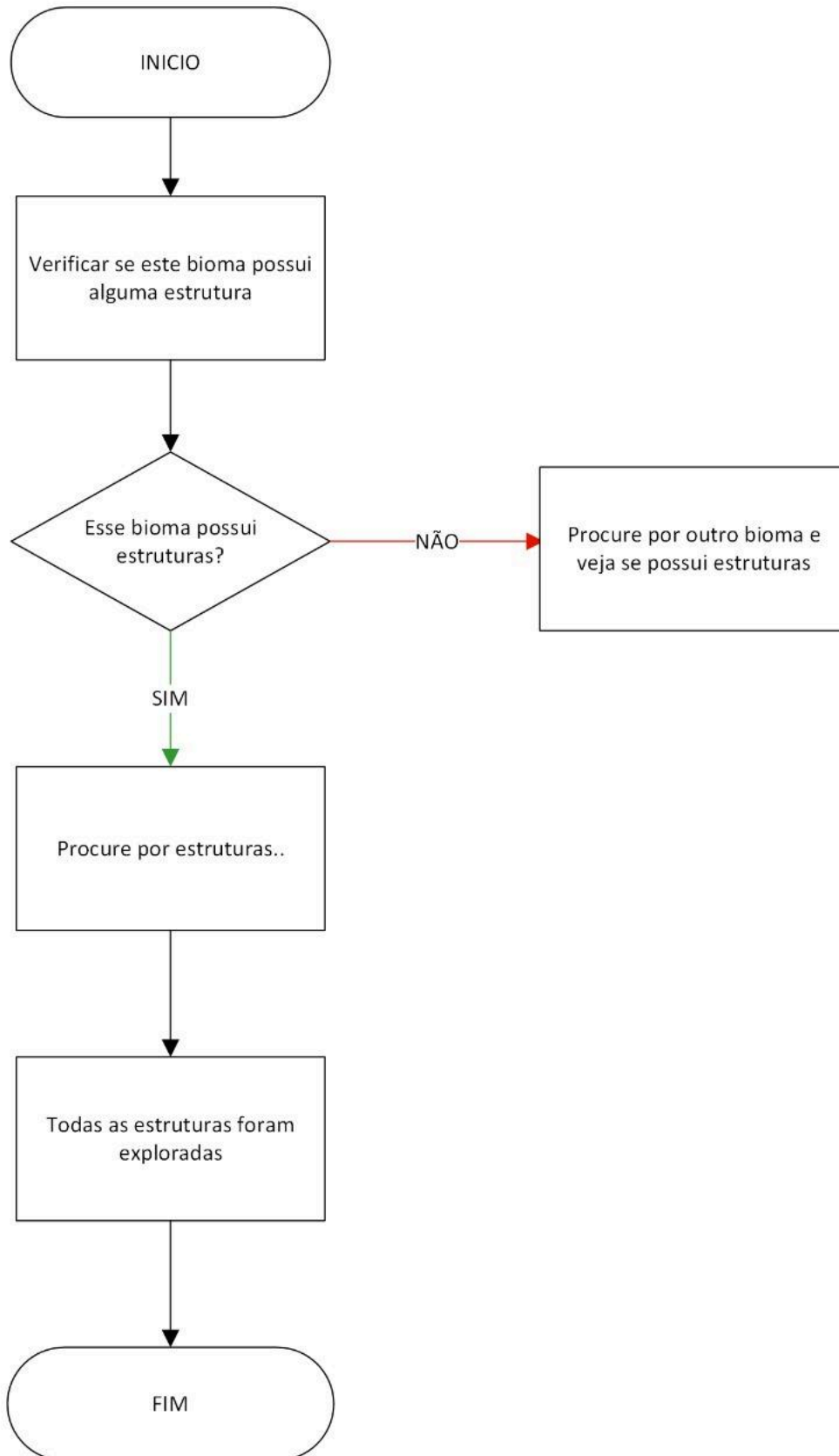
Exploração e Progresso





Coletar recursos raros





CÓDIGOS

```
programa {
    funcao inicio() {

        inteiro quantidade_diamante
        logico diamante_suficiente

        quantidade_diamante = 0
        diamante_suficiente = falso

        escreva("Procurando caverna...", "\n", "Descendo para as camadas 15 e
17...", "\n", "Procurando fontes de lava...", "\n", "\n")

        faca{
            escreva("Quebrando minério de diamante\n")
            quantidade_diamante = quantidade_diamante + 1
            escreva("Dimante atual: ", quantidade_diamante, "\n", "\n")
        } enquanto (quantidade_diamante < 18)

        se (quantidade_diamante == 18){
            diamante_suficiente = verdadeiro
        }

        escreva("Esmeralda suficiente coletado!! Quantidade total: ",
quantidade_diamante, "\n")

    }
}
```

```
programa {
    funcao inicio() {

        inteiro quantidade_esmeralda
        logico esmeralda_suficiente

        quantidade_esmeralda = 0
        esmeralda_suficiente = falso

        escreva("Procurando caverna...", "\n", "Descendo para as camadas 15 e
17...", "\n", "Procurando por fontes de lava...", "\n", "\n")

        faca{
            escreva("Quebrando minério de esmeralda\n")
```

```

    quantidade_esmeralda = quantidade_esmeralda + 1
    escreva("Esmeralda atual: ", quantidade_esmeralda, "\n", "\n")
} enquanto (quantidade_esmeralda < 12)

se (quantidade_esmeralda == 12 ){
    esmeralda_suficiente = verdadeiro
}

escreva("Esmeralda suficiente coletado!! Quantidade total: ",
quantidade_esmeralda, "\n")

}
}

```

```

programa {
    funcao_inicio() {

        inteiro quantidade_lasuli
        logico lasuli_suficiente

        quantidade_lasuli = 0
        lasuli_suficiente = falso

        escreva("Procurando caverna...", "\n", "Descendo para as camadas 15 e
17...", "\n", "Procurando fontes de lava...", "\n", "\n")

        faca{
            escreva("Quebrando minério de lápis lasuli\n")
            quantidade_lasuli = quantidade_lasuli + 1
            escreva("Lápis lasuli atual: ", quantidade_lasuli, "\n", "\n")
        } enquanto (quantidade_lasuli < 48)

        se (quantidade_lasuli == 48){
            lasuli_suficiente = verdadeiro
        }

        escreva("Lápis lasuli suficiente coletado!! Quantidade total: ",
quantidade_lasuli, "\n")

    }
}

```



```

programa {
    funcao_inicio() {

        inteiro quantidade_redstone
        logico redstone_suficiente

        quantidade_redstone = 0
        redstone_suficiente = falso

        escreva("Procurando caverna...", "\n", "Descendo para as camadas 15 e
17...", "\n", "Procurando fontes de lava...", "\n", "\n")

        faca{
            escreva("Quebrando minério de redstone\n")
            quantidade_redstone = quantidade_redstone + 1
            escreva("Redstone atual: ", quantidade_redstone, "\n", "\n")
        } enquanto (quantidade_redstone < 48)

        se (quantidade_redstone == 48){
            redstone_suficiente = verdadeiro
        }

        escreva("Redstone suficiente coletado!! Quantidade total: ",
quantidade_redstone, "\n")

    }
}

```

```

programa {
    funcao_inicio() {

        inteiro experiencia
        inteiro pedregulho
        inteiro balde_aqua

        escreva("Qual sua quantidade de experiência?\n")
        leia(experiencia)

        se(experiencia >= 30){
            escreva("\nVocê possui boa quantidade de experiência.\n")
        }

        senao se(experiencia < 30){
            escreva("\nVocê precisa de recursos para coletar experiência.",

```

```
"\n", "É preciso 40 pedregulhos e 1 balde com água.\n", "\n")
escreva("Digite sua quantidade de pedregulho: ")
leia(pedregulho)

enquanto(pedregulho < 40){
    escreva("\nConseguindo pedregulho...\n")
    pedregulho = pedregulho++
    escreva("Pedregulho atual: ", pedregulho, "\n")
}

se(pedregulho >= 40){
    escreva("\nQuantidade de pedregulho suficiente!\n")
    escreva("\nDigite sua quantidade de balde com água: ")
    leia(balde_agua){

        enquanto (balde_agua < 1){
            escreva("Conseguindo balde com água...\n")
            balde_agua = balde_agua + 1
        }

        se(balde_agua >= 1){
            escreva("\nQuantidade de balde suficiente!\n")}

        se(pedregulho >= 40 e balde_agua >= 1){
            escreva("\n*Procurando por cavernas...*", "\n", "*Procurando
por dungeon...*", "\n", "*Dungeon encontrada!*")
            escreva("\n", "\n*Fazendo estrutura em volta da dungeon*...",
"\n", "\n*Estrutura feita!\n")
        }

        faca{
            escreva("\nConseguindo experiência com a estrutura...")
            experiencia = experiencia + 1
            escreva("\nExperiência atual: ", experiencia, "\n")
        }enquanto(experiencia < 30)

    }
}
}
```

```

programa {
    funcao inicio() {
        logico vila, picareta
        inteiro esmeralda, quantidade

        faca{
            escreva("Procurando por uma vila...")
            escreva("\nEncontrou uma vila? (sim ou não) ")
            leia(vila)
        } enquanto (vila == falso)

        escreva("Fale com um villager ")
        escreva("\nQuantas esmeraldas para tal item? ")
        leia(esmeralda)
        escreva("Minerar ", esmeralda, " esmeraldas. ")
        escreva("\nVolte ao villager e complete a troca.")
    }
}

```

```

programa {
    funcao inicio() {
        logico resposta

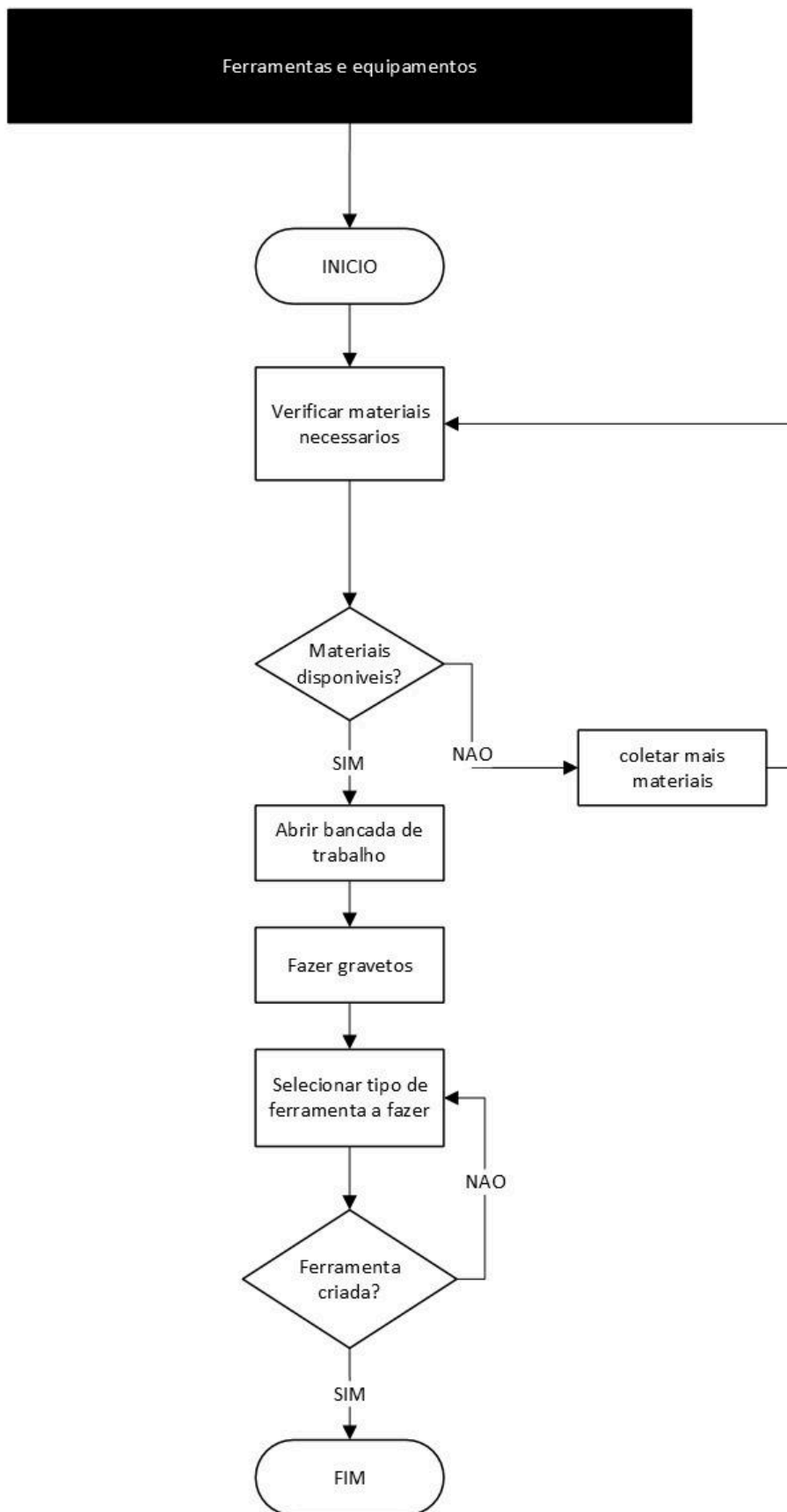
        escreva("Você precisa encontrar estruturas.", "\n", "Aperte 'F3' no teclado e veja se há estruturas no bioma.\n")
        escreva("\nEsse bioma possui estruturas?\n")
        leia(resposta)

        se (resposta == falso) {
            escreva("\nProcure por outro bioma e veja se possui estruturas.\n")
        }

        senao se (resposta == verdadeiro) {
            escreva("\n*Procurando por estruturas...*", "\n", "*Todas as estruturas foram exploradas!*", "\n")
        }
    }
}

```

Ferramentas e Progresso



CÓDIGO

```

programa {
    funcao_inicio() {

        cadeia_resposta
        logico_resposta_ferramenta
        logico_resposta Equipamento

        escreva("Você deseja criar um equipamento ou uma ferramenta? (digite
o quê deseja)\n")
        leia(resposta)

        se(resposta == "ferramenta" ou "Ferramenta"){

            faça{
                escreva("\nColete os recursos para construir sua ferramenta.\n")
                escreva("Já possui todos os recursos? (sim ou não)\n")
                leia(resposta_ferramenta)
            } enquanto(resposta_ferramenta == falso)
        }

        se(resposta_ferramenta == verdadeiro){
            escreva("\nFaça a sua ferramenta na bancada de trabalho.\n")
        }

        se(resposta == "equipamento" ou "Equipamento"){

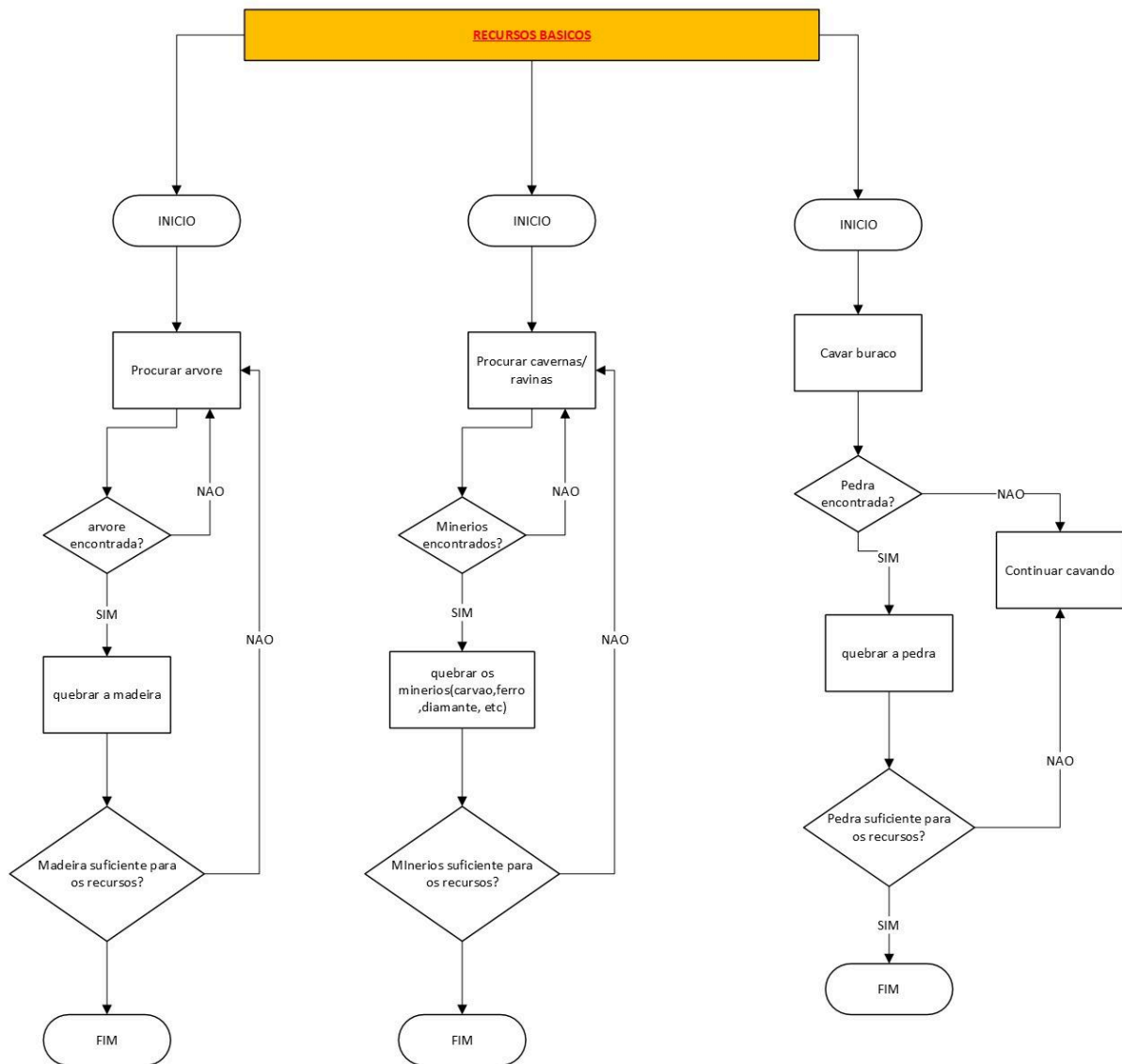
            faça{
                escreva("\nColete os recursos para construir seu equipamento.\n")
                escreva("Já possui todos os recursos? (sim ou não)\n")
                leia(resposta_equipamento)
            } enquanto(resposta_equipamento == falso)
        }

        se(resposta_equipamento == verdadeiro){
            escreva("\nFaça o seu equipamento na bancada de trabalho.\n")
        }

    }
}

```

Recursos Básicos



CODIGOS

```
programa {
```

```

funcao inicio() {

    inteiro quantidade_carvao
    logico carvao_suficiente

    quantidade_carvao = 0
    carvao_suficiente = falso

    escreva("Procurando caverna...")

    faca{
        escreva("Quebrando minério de carvão\n")
        quantidade_carvao = quantidade_carvao + 1
        escreva("Carvão atual: ", quantidade_carvao, "\n")
    } enquanto (quantidade_carvao <= 30)

    se (quantidade_carvao == 30){
        carvao_suficiente = verdadeiro
    }

    escreva("\nCarvão suficiente coletado!! Quantidade total: ",
quantidade_carvao, "\n")

}
}

```

```

programa {
    funcao inicio() {

        inteiro quantidade_ferro
        logico ferro_suficiente

        quantidade_ferro = 0
        ferro_suficiente = falso

        escreva("Procurando caverna...")

        faca{
            escreva("Quebrando minério de ferro\n")
            quantidade_ferro = quantidade_ferro + 1
            escreva("Ferro atual: ", quantidade_ferro, "\n", "\n")
        } enquanto (quantidade_ferro < 40)
    }
}

```



```

    se (quantidade_ferro == 40 ){
        ferro_suficiente = verdadeiro
    }
    escreva("Ferro suficiente coletado!! Quantidade total: ",
quantidade_ferro, "\n")

}
}

```

```

programa {
    funcao inicio() {

        inteiro quantidade_madeira
        logico madeira_suficiente

        quantidade_madeira = 0
        madeira_suficiente = falso

        escreva("Procurando árvore...\n")

        faca{
            escreva("Quebrando bloco de madeira\n")
            quantidade_madeira = quantidade_madeira + 1
            escreva("Madeira atual: ", quantidade_madeira, "\n", "\n")
        } enquanto (quantidade_madeira < 20)

        se (quantidade_madeira == 20 ){
            madeira_suficiente = verdadeiro
        }

        escreva("Madeira suficiente coletada!! Quantidade Total: ",
quantidade_madeira, "\n")

    }
}

```

```

programa {
    funcao inicio() {

```

```
inteiro quantidade_pedra
logico pedra_suficiente

quantidade_pedra = 0
pedra_suficiente = falso

escreva("Cavando buraco...\n")

faca{
    escreva("Quebrando bloco de pedra\n")
    quantidade_pedra = quantidade_pedra + 1
    escreva("Pedregulho atual: ", quantidade_pedra, "\n", "\n")
} enquanto (quantidade_pedra < 32)

se (quantidade_pedra == 32 ){
    pedra_suficiente = verdadeiro
}

escreva("\nPedregulho suficiente coletada!! Quantidade atual: ",
quantidade_pedra, "\n")

}

}
```