



# Wife`s Programing Language

Uma linguagem engraçada, baseada na minha esposa !

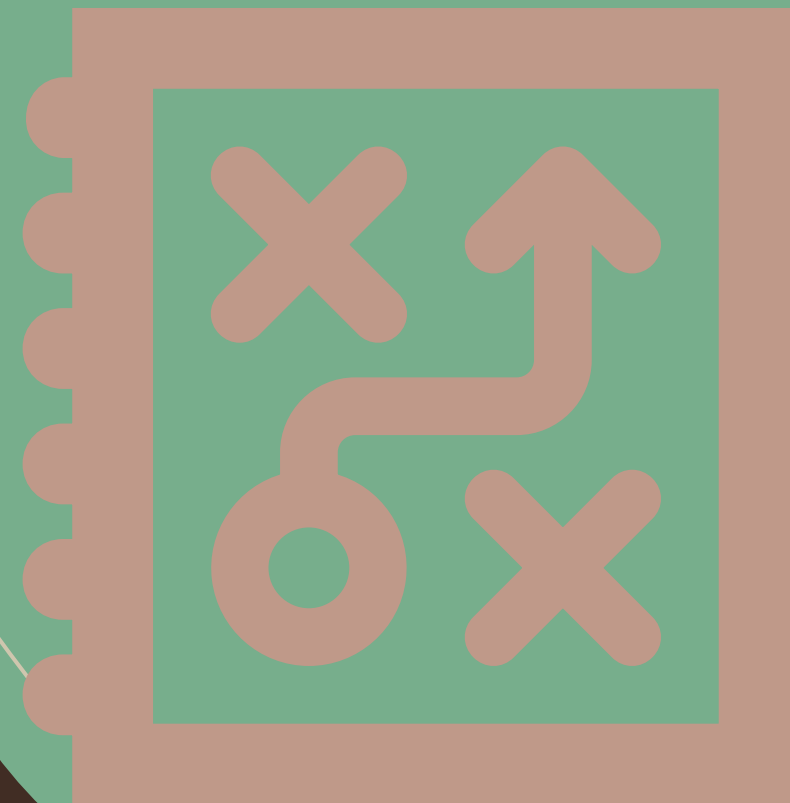
# Motivação

- Ser divertida ao criar
- Usar características da Leandra na linguagem
- Um jeito, de pelo menos no papel, controlar o tempo que ela gasta nas tarefas diárias



## Características da Linguagem

- Baseada em "programar" o dia ou uma sequência de dias
- Possui "tarefas", as variáveis, que são declaradas com o tempo que pode ser gasto
- Tem laço condicional, loop e print, tudo baseado no tempo



# Tokens Importantes

As palavras reservadas são todas em CAPSLOCK:

- HORA\_DE\_BRILHAR / HORA\_DE\_SEMPRE\_DIVA
- GRUNIDO / A\_MIMIR
- TAREFA / ACAO
- TO\_COM\_FOME / QUERO\_PITDAS
- HABLAR
- SE
- ENQUANTO\_ELA\_NAO\_MUDA\_DE\_IDEIA

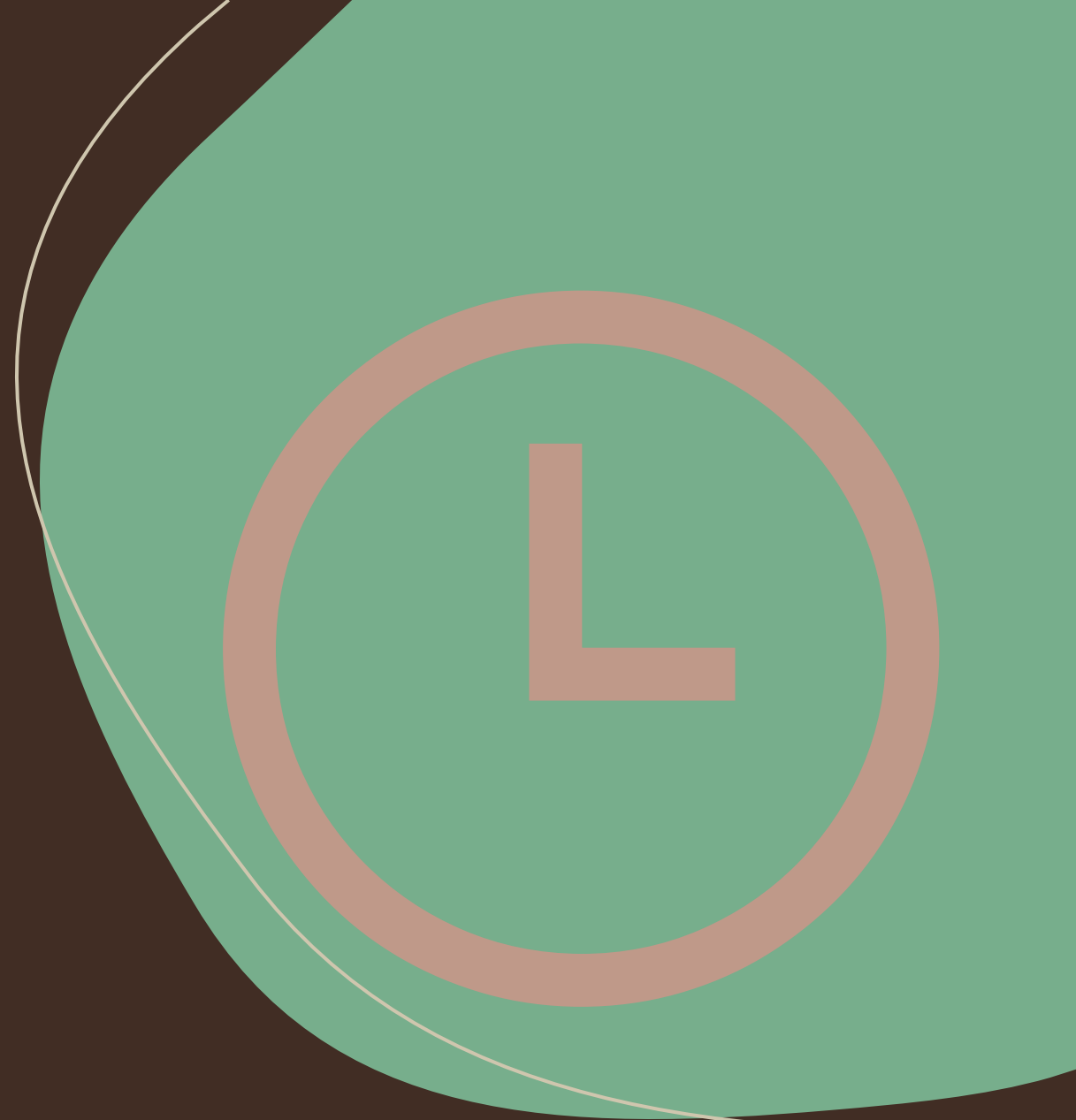
HORA\_DE\_BRILHAR / HORA\_DE\_SEMPRE\_DIVA

Esses tokens marcam o início e o fim  
do código, tudo precisa vir dentro  
desses tokens

HORA\_DE\_BRILHAR

aqui seu código

HORA\_DE\_SEMPRE\_DIVA



# GRUNIDO / A\_MIMIR

Esses tokens delimitam um dia de tarefas

HORA\_DE\_BRILHAR

GRUNIDO

aqui seu dia 1

A\_MIMIR

GRUNIDO

aqui seu dia 2

A\_MIMIR

HORA\_DE\_SEMPRE\_DIVA

# TAREFA / ACAO

- Tarefa funciona para a declaração da variável

TAREFA << (instagram, 60)

No exemplo acima, temos a tarefa instagram, que pode ser feita por até 60 min !

- ACAO indica que aquela tarefa irá ser executada, e o tempo restante da mesma irá diminuir

ACAO >> (tirando\_foto, 30)



# TO\_COM\_FOME / QUERO\_PITDAS

Esses tokens são para evitar ambiguidade, funcionam como {} de muitas linguagens, apenas para separar escopo. É usado nos condicionais e no loop

TO\_COM\_FOME

aqui seu código caso condicional ou caso loop

QUERO\_PITDAS





# HABLAR

Token para print, uma forma  
da esposa se comunicar e expressar  
o que pensa

HABLAR (vou\_sair, 30)



# SE

Token condicional, irá testar se tem tempo sobrando na tarefa, e executar o seu bloco

SE TAREFA.instagram

TO\_COM\_FOME

ACAO >> (tirando\_foto, 30)

ACAO >> (postando\_foto, 30)

QUERO\_PITDAS



## ENQUANTO\_ELA\_NAO\_MUDA\_DE\_IDEIA

Token do loop, o while de muitas linguagens. Ele irá executar tudo dentro do bloco enquanto sobrar tempo no condicional

```
ENQUANTO_ELA_NAO_MUDA_DE_IDEIA
```

```
  TAREFA.se_arrumando_para_sair
```

```
    TO_COM_FOME
```

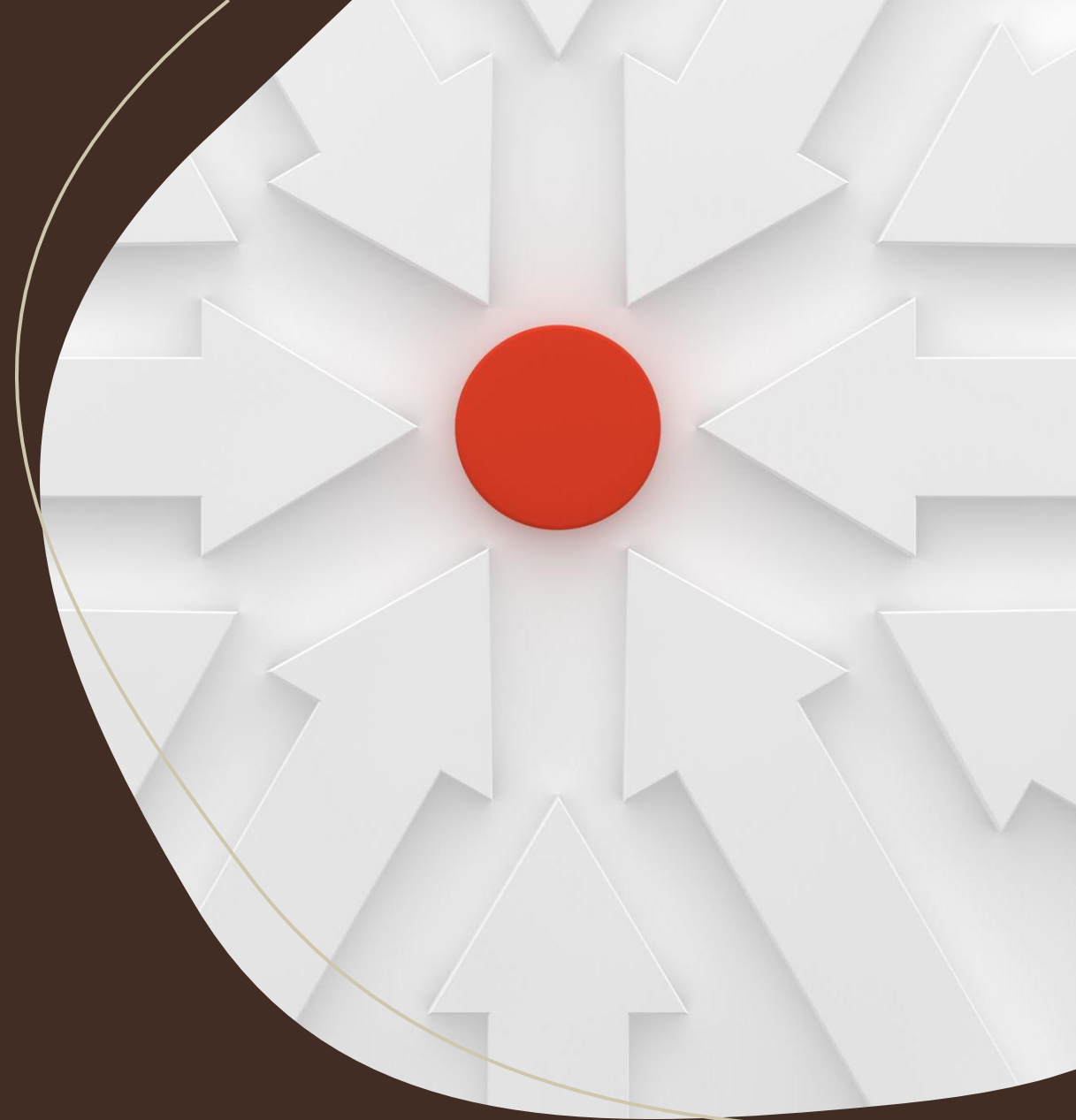
```
      ACAO >> (maquiagem, 60)
```

```
      ACAO >> (escolhendo_roupa, 60)
```

```
      ACAO >> (arrumando_cabelo, 60)
```

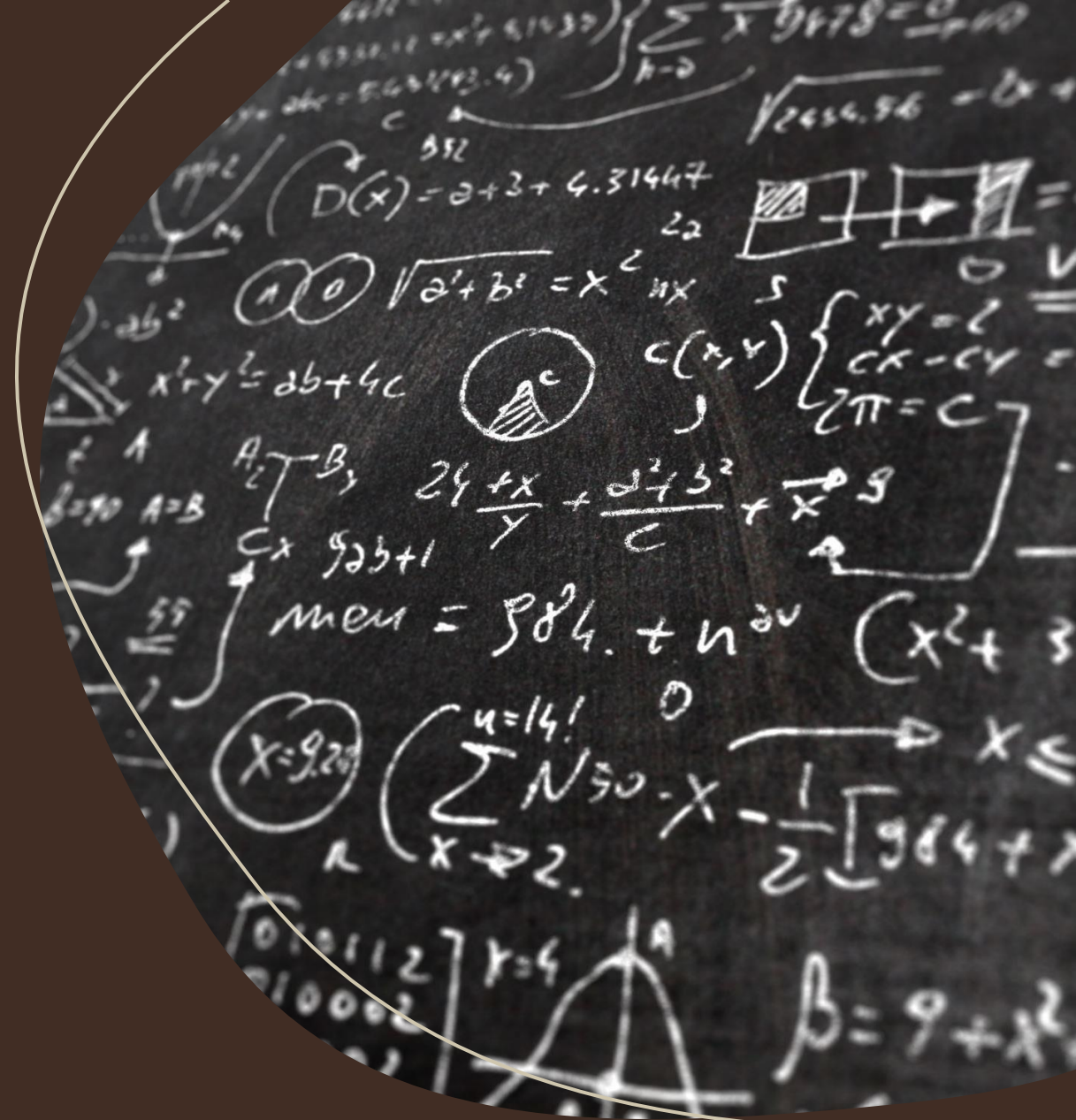
```
      HABLAR (vou_sair, 15)
```

```
    QUERO_PITDAS
```



## Outras características importantes

- Nos condicionais e loops, é preciso especificar o token "TAREFA", seguido de "." para especificar de qual tarefa irá testar o tempo restante
- "<<" usado para declaração de tarefa
- ">>" usado para executar uma ação







# Curiosidades

A linguagem como um todo foi baseada na minha esposa:

- 'GRUNIDO' é o token de começar um dia porque é a primeira coisa que ela faz quando acorda, e "A\_MIMIR" é o último porque ela sempre fala isso antes de dormir
- Ela gosta muito de pizza, e chama carinhosamente de "PITDAS", por isso o token de "QUERO\_PITDAS"
- Sempre que ela pensa muito fala "hummm", por isso é a maneira de inserir comentários no código