

# **HEALTHLANG**

**UMA LINGUAGEM DE PROGRAMAÇÃO PARA AUTOMAÇÃO DE  
RESPOSTAS EM AMBIENTES HOSPITALARES**

# HEALTHLANG

## LINGUAGEM DE PROTOCOLOS DE SAÚDE

**HealthLang é uma linguagem de alto nível simples, pensada para escrever protocolos clínicos básicos de forma clara e legível.**

**Ela foi projetada para que profissionais de saúde (médicos, enfermeiros) consigam ler e compreender os protocolos escritos sem precisar conhecer programação tradicional.**

### Objetivo

- **Simular respostas automáticas de uma máquina virtual (VitalsVM).**
- **Permitir que protocolos clínicos sejam escritos em linguagem quase natural.**

### Público-alvo

**Usuário da linguagem (médico/enfermeiro): Vê apenas sensores e ações clínicas.**

# ORGANIZAÇÃO DO REPOSITÓRIO

```
HealthLang/
├── compiler/
│   ├── src/          # Flex + Bison → healthlang
│   └── examples/     # 6 protocolos .hl
├── vm/
│   ├── implementacao/ # VitalsVM (vm.py)
│   └── vmasm_files/   # Assembly gerado
├── saidas/           # Logs de execução
├── grammar.ebnf       # Gramática formal
├── README.md          # Documentação
└── TUTORIAL.md        # Guia de uso
```

# CARACTERÍSTICAS DA **LINGUAGEM**

Compilada para assembly - **VitalsVM**

## **Três pilares:**

1. Sensores
2. Ações
3. Lógica

- 
- 1
  - 2
  - 3

Leitura de sinais vitais  
(oxigenio, batimento,  
intravenoso)

Comandos clínicos  
(ajustar\_O2, ajustar\_soro,  
alerta)

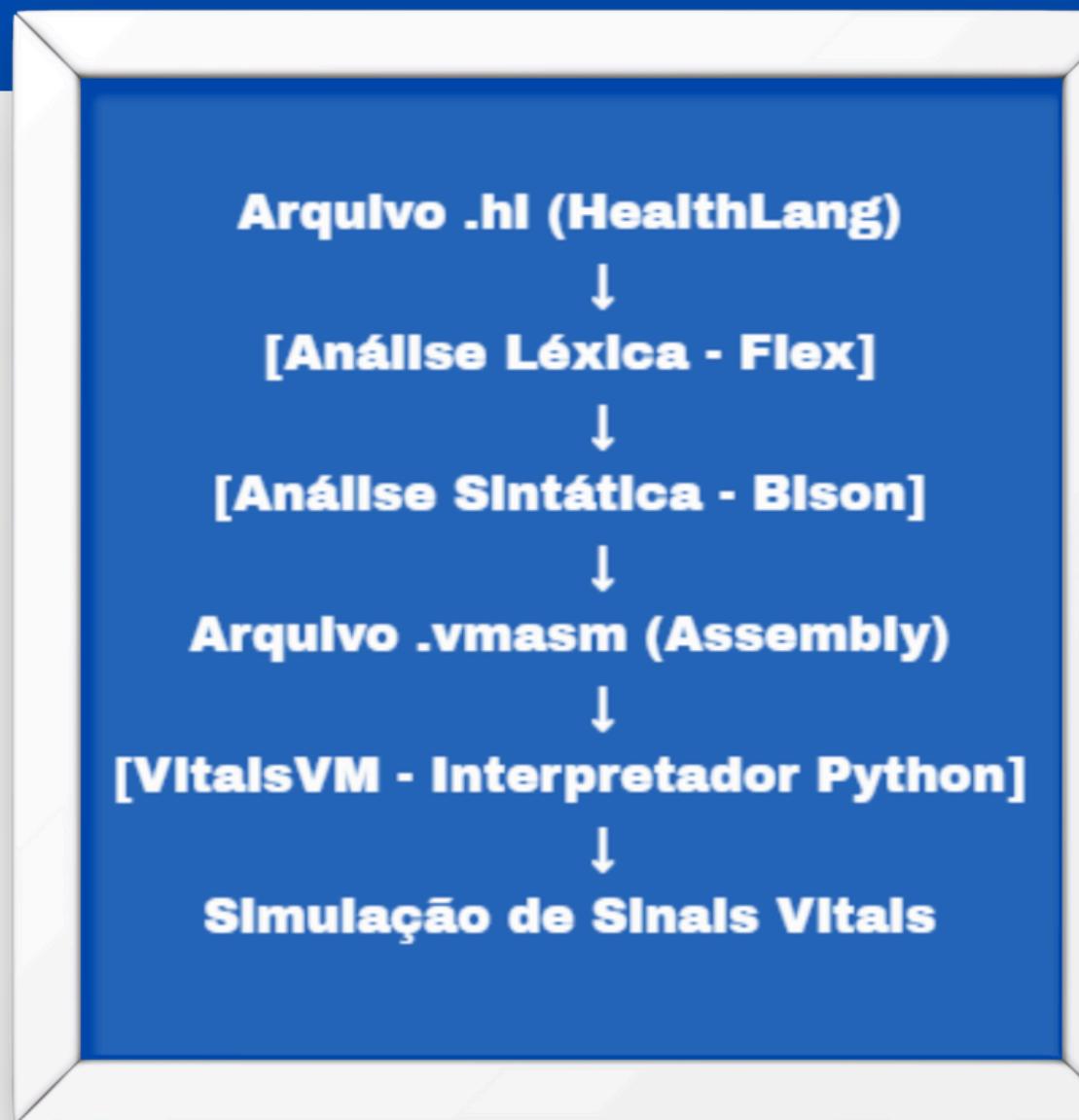
Estruturas de controle  
(if/else, while)

# COMO É UM PROTOCOLO EM **HEALTHLANG?**

- Se a frequência cardíaca passar de 120 bpm
- Ajustar soro para 50 ml/h
- Disparar alarme
- Registrar valor do batimento

```
// Protocolo de Taquicardia
if (batimento > 120) {
    ajustar_soro(50);
    alerta();
    log(batimento);
} else {
    silenciar();
}
```

# COMO FUNCIONA O COMPILADOR?



## Tecnologias utilizadas:

- **Flex: Geração do analisador léxico**
- **Bison: Geração do parser sintático**
- **C: Geração do compilador**
- **Python: Implementação da VitalsVM**

# EBNF

## SIMPLIFICADA

Destaque:

- Sintaxe simples e clara
- Estruturas familiares (if, while)
- Vocabulário do domínio médico

A versão completa está no README do repositório

```
PROGRAM      = { STATEMENT } ;
STATEMENT    = IF | WHILE | ADJUST | ACTION | WAIT | LOG ;
IF           = "if" "(" BOOL ")" BLOCK [ "else" BLOCK ] ;
ADJUST       = "ajustar_02" "(" NUMBER ")" ";" |
                "ajustar_soro" "(" NUMBER ")" ";" ;
SENSOR        = "oxigenio" | "batimento" | "intravenoso" ;
```

## **Registradores mutáveis:**

- O2 - Fluxo de oxigênio (0-100)
- IV - Taxa de soro intravenoso (0-100)

## **Sensores readonly (atualizam a cada tick):**

- S\_SPO2 - Saturação de oxigênio 80-100%
- S\_BPM - Batimentos por minuto 40-160 bpm
- S\_IVLV - Nível de fluido corporal 0-100%

## **ISA Turing-Completa:**

- SET, INC, DECJZ (estilo Minsky Machine)
- GOTO, CJMP (controle de fluxo condicional)
- PUSH/POP (pilha para memória temporária)
- PRINT, HALT (debug e controle)

# SIMULAÇÃO DE SINAIS VITAIS

A VM simula como os sinais vitais respondem às intervenções

## Exemplos:

- $S\_SPO2$  sobe quando  $O_2$  aumenta (com inércia)
- $S\_BPM$  cai quando  $IV$  aumenta (hidratação)
- $S\_IVLV$  sobe gradualmente com infusão de soro

## Fórmulas simplificadas:

```
 $S\_SPO2 += clamp((O_2 - 50) / 50, -2, +2)$ 
```

```
 $S\_BPM += clamp((90 - IV) / 20, -3, +1)$ 
```

```
 $S\_IVLV += clamp(IV / 10 - 1, -1, +2)$ 
```

# PROTOCOLOS EXEMPLOS PRÁTICOS

Esses foram os exemplos práticos testados durante o desenvolvimento do projeto.

Todos estão disponíveis no repositório.

Protocolo	Descrição
<a href="#">taquicardia.hl</a>	Detecção de FC elevada
<a href="#">oxigenio.hl</a>	Ajuste progressivo de O <sub>2</sub>
<a href="#">soro.hl</a>	Reposição volêmica
<a href="#">desidratacao.hl</a>	Simulação de perda de volume
<a href="#">choque_hipovolemico.hl</a>	Múltiplas condições críticas
<a href="#">monitoramento_continuo.hl</a>	Loop infinito (UTI 24/7)

# EXEMPLO: MONITORAMENTO CONTÍNUO

**monitoramento\_continuo.hl**

```
while (batimento >= 0) {  
    if (oxigenio < 92) {  
        ajustar_o2(70);  
        alerta();  
    }  
    if (batimento > 120) {  
        aumentar_soro(5);  
        alerta();  
    }  
    esperar(50);  
}
```

**Saída do programa:**

```
--- ciclo 100 ---  
S_BPM=120  
S_SP02=87  
--- ciclo 200 ---  
S_BPM=115  
S_SP02=91
```

# DESTAQUES E CURIOSIDADES

1

## 1. Suporte a Aninhamento

- Pilha de labels para ifs/whiles aninhados
- Gestão correta de escopo

2

## 2. Modelo Fisiológico com Inércia

- Sensores não mudam instantaneamente
- Limites fisiológicos realistas
- SpO2: 80-100%, BPM: 40-160, IVLV: 0-100
- Atualização a cada instrução executada

3

## 3. VM Implementada do Zero

- Sistema de labels e jumps
- Proteção contra loops infinitos (--steps) → podem ser usados mas está limitado afim de testes

# COMO USAR O HEALTHLANG

Do código HealthLang até a execução na VitalsVM

## 1. COMPILE O COMPILADOR

```
cd compiler/src && make
```

## 2. COMPILE UM PROTOCOLO CLÍNICO

```
./healthlang ..//exemples/taquicardia.hl \
..//..//vm/vmasm_files/taquicardia.vmasm
```

## 3. EXECUTAR NA VITALSVM

```
cd ..//vm/implementacao
python3 vm.py ..//vmasm_file/taquicardia.vmasm --steps 500
```

# LIMITAÇÕES E PRÓXIMOS PASSOS

## Limitações atuais

- Sem operadores lógicos (AND/OR)
- Modelo fisiológico simplificado
- Sem variáveis temporárias
- Não possui interface gráfica
- Sem persistência de estado

## Futuras melhorias

- Novos sensores (pressão, temperatura)
- Suporte a funções e escopos
- Variáveis locais
- Dashboard web em tempo real
- Integração com IoT hospitalar

# ONDE O HEALTHLANG PODE SER USADO?

## Educação Médica

- Simulação de urgências
- Treinamento de equipes

## Prototipagem de Protocolos

- Testes seguros antes da aplicação real
- Validação de condutas

## Pesquisa

- Modelagem fisiológica
- Estudos sobre automação clínica

## Sistemas Embarcados

- Monitores inteligentes
- Equipamentos de UTI automáticos



# OBRIGADA