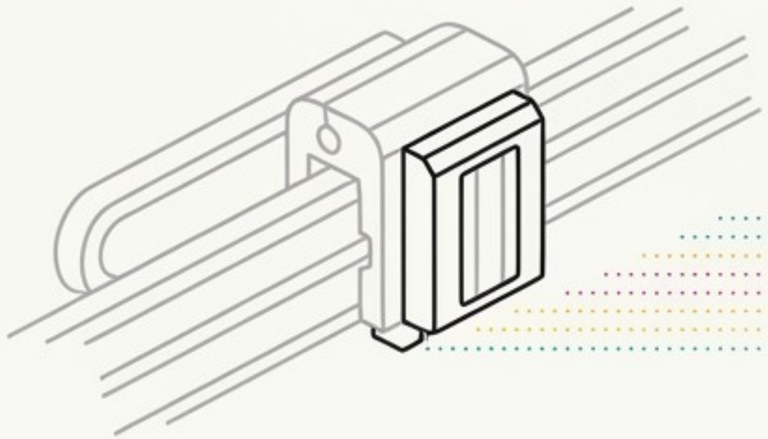
Impressora

Uma linguagem de programação para
automatizar cenários do dia a dia.



Por que uma linguagem dedicada para impressoras?

Tarefas de impressão, embora comuns, frequentemente exigem scripts complexos ou intervenção manual repetitiva. A linguagem **Impressora** nasceu da necessidade de simplificar e automatizar esse processo com uma sintaxe dedicada e intuitiva, focada exclusivamente no domínio da impressão.

Script Genérico

```
# Script complexo para uma tarefa simples
if [ $(get_ink) -lt 20 ]; then
  echo "Tinta baixa!"
fi
lp -d HP_LaserJet -o media=A4 -o
quality=normal -n 30 prova.pdf
```

Linguagem Impressora

```
// Simples, legível e direto ao ponto
se (nivel_tinta() < 20) {
   imprimir_texto("Tinta baixa!");
}
imprimir("prova.pdf", 30);
```

Na Prática: Imprimindo uma Prova

Um cenário real: uma professora precisa imprimir 30 cópias de uma prova, em preto e branco e com qualidade média, mas antes quer garantir que a impressora tem tinta suficiente.

```
// Professora imprimindo prova

int alunos = 30;

se (nivel_tinta() < 20) {
    imprimir_texto("Tinta baixa!");
}

definir_cor(falso);
definir_qualidade(MEDIA);

imprimir("prova.pdf", alunos);

imprimir_texto("Concluído!");
```

Define o modo de impressão para preto e branco.

Ajusta a qualidade para economizar tinta.

Envia o arquivo para a fila de impressão com o número de cópias.

A Caixa de Ferramentas: Comandos Principais

A linguagem oferece um conjunto completo de comandos, organizados para cobrir todas as etapas do processo de impressão.

Configuração

```
definir_cor(verdadeiro/falso)
definir_qualidade(BAIXA |
MEDIA | ALTA)
definir_papel(A4 | CARTA |
A3)
```

Define os parâmetros da impressão antes do envio.

Ação e Verificação

```
imprimir("arquivo", copias)
imprimir_texto("mensagem")
verificar_tinta()
verificar_papel()
aguardar_pronta()
```

Executa as tarefas de impressão e verifica o estado.

Sensores

```
nivel_tinta()
qtd_papel()
esta_pronta()
paginas_impressas()
modo_cor()
```

Retorna valores em tempo real dos sensores da impressora.

Impressão Inteligente com Lógica Condicional

‘Impressora’ permite criar scripts que reagem ao estado do hardware, evitando falhas e desperdício de papel e tinta.

```
// Verifica papel e tinta ANTES de imprimir 50 cópias

int copias = 50;

se (qtd_papel() < copias) {
    imprimir_texto("Papel insuficiente!");
} senao {
    se (nivel_tinta() < 15) {
        imprimir_texto("Tinta baixa!");
    } senao {
        imprimir("trabalho.pdf", copias);
        imprimir_texto("Sucesso!");
    }
}
```

O Poder da Automação com Loops

Para trabalhos em lote, como imprimir provas para várias turmas, a linguagem suporta loops que, combinados com verificações, criam fluxos de trabalho automatizados e resilientes.


```
// Imprime provas para 3 turmas, verificando a tinta a cada lote

int turma = 1;

enquanto (turma <= 3) {
    se (nivel_tinta() < 20) {
        imprimir_texto("Reabastecer tinta!");
        verificar_tinta(); // Pausa e aguarda ação
    }

    imprimir("prova_turma.pdf", 35);
    turma = turma + 1;
}
```

Por Baixo do Capô: Arquitetura do Compilador à VM

‘Impressora’ é um sistema completo: o código-fonte é traduzido por um compilador (Flex/Bison) para um assembly customizado, que é então executado por uma Máquina Virtual (Python) que simula uma impressora real.

O DNA da Linguagem: A Gramática Formal (EBNF)

A sintaxe da linguagem é rigorosamente definida por uma gramática formal, garantindo que o código seja consistente e previsível.

Estrutura `se/senao`

```
ifStmt = "se" "(" expr ")" stmt [ "senao" stmt ] ;
```

Comando de Impressão de Documento

```
printDoc = "imprimir" "(" string "," expr ")" ";" ;
```

Expressões Aritméticas

```
additive = term { ("+" | "-" ) term } ;
term     = factor { ("*" | "/" ) factor } ;
```

Chamada de Sensor

```
primary = "nivel_tinta" "(" ")" | "qtd_papel" "(" ")" ... ;
```

Experimente Você Mesmo

O projeto é open-source e pronto para ser compilado e executado em um ambiente Linux.

Requisitos e Instalação

Requisitos: Flex, Bison, GCC, Make, Python 3

Instalação (Debian/Ubuntu):

```
sudo apt-get install flex bison gcc make python3
```

Compilação e Execução

1. Compilar o compilador:

```
make
```

2. Gerar o código Assembly a partir de um arquivo `.imp`:

```
./impressora saida.asm < seu_programa.imp
```

3. Executar o Assembly na VM:

```
python3 impressoravm.py saida.asm
```

A pasta `testes/´ no repositório contém diversos exemplos prontos para execução.

Impressora

Simples. Focada. Poderosa.

OBRIGADO!