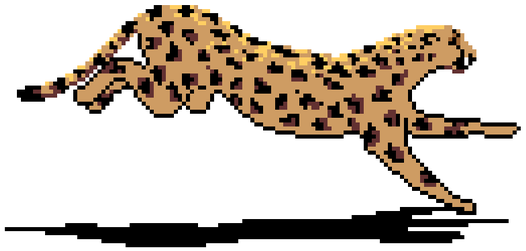
PUC Minas

Seminário - Linguagem de Programação

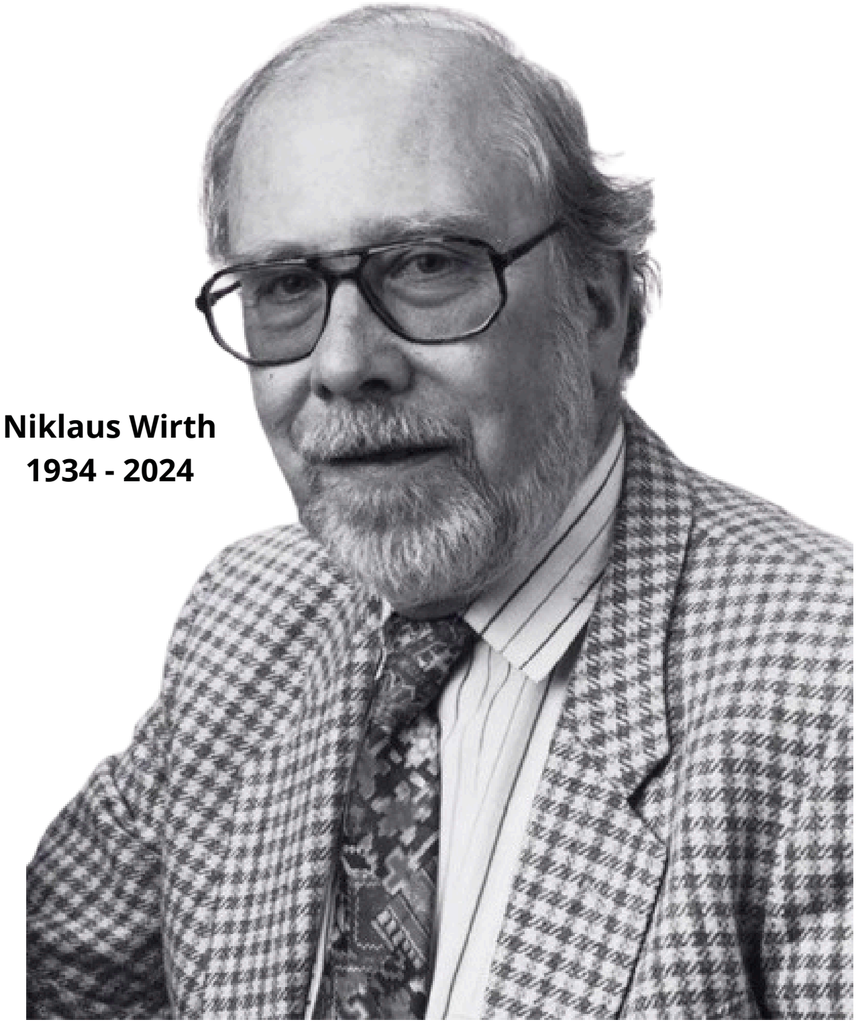
Pascal



Caio César, Diego Feitosa, Felipe Guerzoni, Guilherme Silva e Mateus Diniz

Sumário

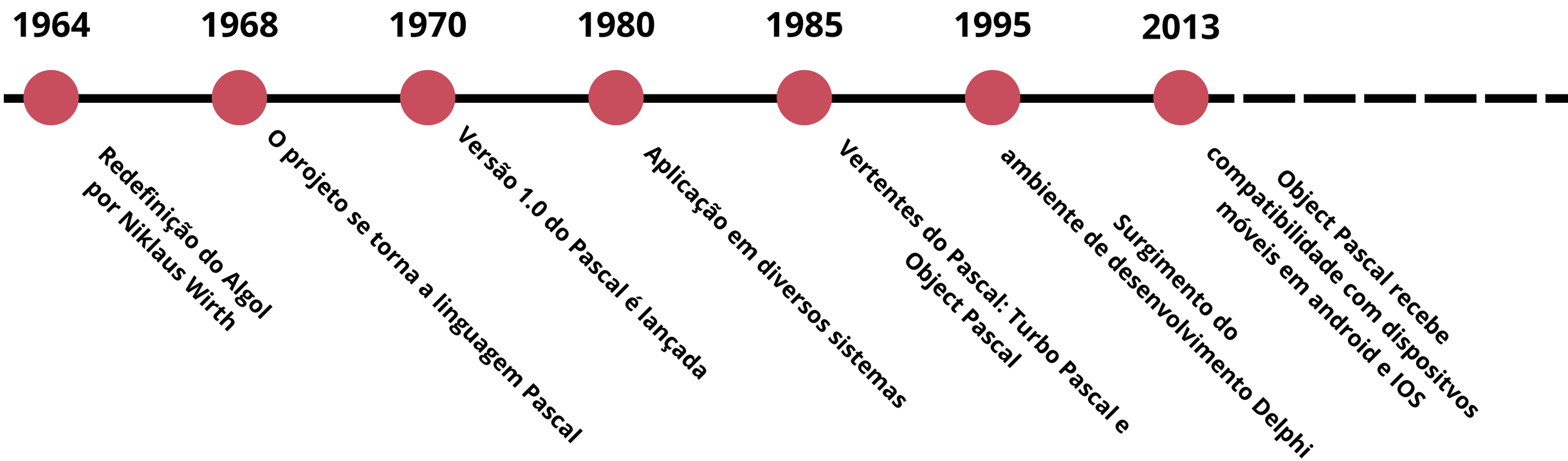
Introdução



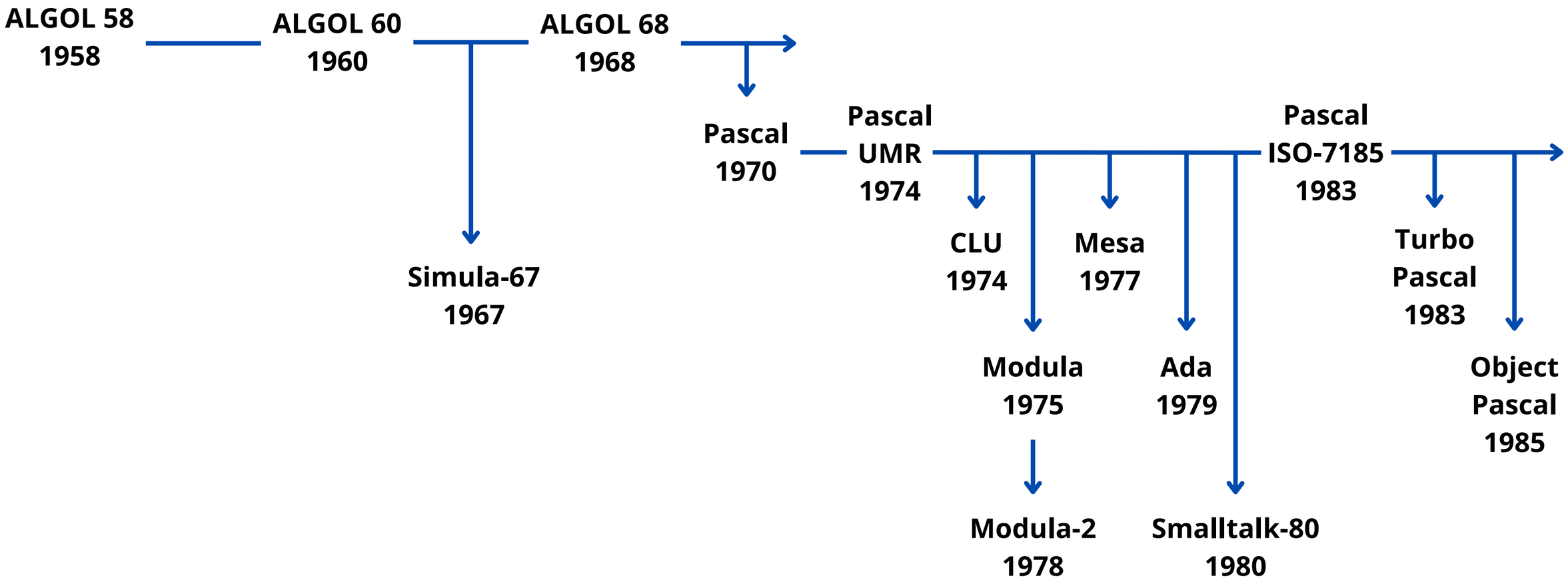
Niklaus Wirth
1934 - 2024

- Desenvolvida no final da década de 1960.
- Linguagem de tipagem forte e sintaxe simples.
- Uma das principais concorrentes da linguagem C por anos.

Histórico


1964
Redefinição do Algol por Niklaus Wirth
1968
O projeto se torna a linguagem Pascal
1970
Versão 1.0 do Pascal é lançada
1980
Aplicação em diversos sistemas
1985
Vertentes do Pascal: Turbo Pascal e Object Pascal
1995
Surgimento do ambiente de desenvolvimento Delphi
2013
Object Pascal recebe compatibilidade com dispositivos móveis em android e IOS

Histórico


ALGOL 58
1958
ALGOL 60
1960
ALGOL 68
1968
Simula-67
1967
Pascal
1970
Pascal
UMR
1974
CLU
1974
Mesa
1977
Modula
1975
Ada
1979
Modula-2
1978
Smalltalk-80
1980
Pascal
ISO-7185
1983
Turbo
Pascal
1983
Object
Pascal
1985

Paradigma

Imperativo

Execução de comandos que atualizam variáveis, junto

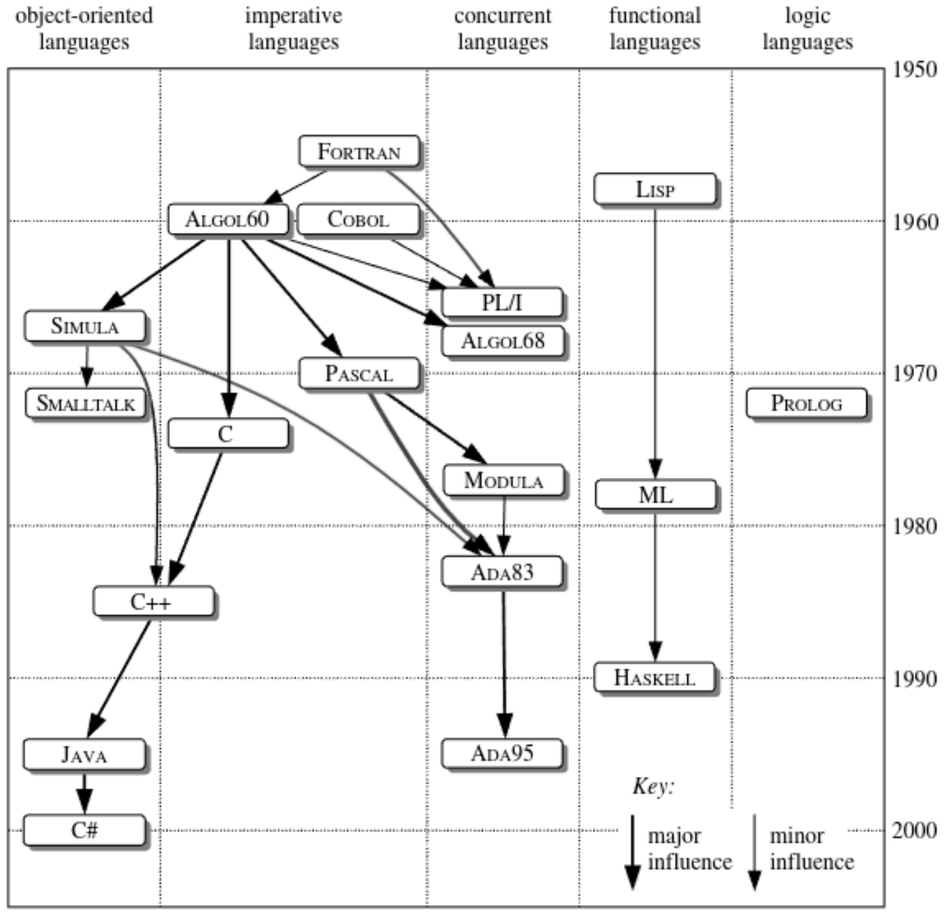


Figure 1.1 Dates and ancestry of major programming languages.

Características

Linguagem educacional

- Foco no ensino
- Incentivo a boas práticas
- Linguagem clara e objetiva

Características

Tipagem estática

```
var
  x: Integer; y: Real; a: Char; name: String; v: Boolean;
begin
  x:=10; y:=3.14; a:='n'; name='Niklaus'; v:=false;
end.
```

```
type
  day: Integer; name: String;
var
  d1: day; n1: name;
begin
  d1:=19; n1:='Niklaus Wirth';
end.
```

Características

Sintaxe

```
function factorial(const a: integer) : integer;
begin
    if a ⩾ 1 then
        factorial := a * factorial(a-1)
    else
        factorial := 1;
end;

procedure factorial(const a: integer; result: PInteger) : integer;
begin
    if a ⩾ 1 then
    begin
        factorial(a - 1 , result);
        result^ := a * result^;
    end
    else
        result^ := 1;
end;
```

Linguagens Relacionadas

Ada
In Strong Typing We Trust

Uma linguagem baseada em pascal criada pelo departamento de defesa americano com o objetivo de possuir alta segurança e confiabilidade para sistemas críticos como aeronaves

Algol-68

Sendo o precursor para muitas linguagens como C, Python e Pascal, Algol 68 trouxe conceitos avançados que revolucionaram a indústria na época como o uso de estruturas dinâmicas e tipagem forte, evitando a ocorrência de erros

Modula-2

Estabelecida pelo mesmo desenvolvedor do pascal, modula foi criada com o propósito de ser mais modular e segura, além de possuir compatibilidade com sistemas embarcados

Pascal

```pascal
program PrimoCheck;
uses crt;

function EhPrimo(n: Integer): Boolean;
var
  i: Integer;
begin
  if (n < 2) then
    EhPrimo := False
  else
  begin
    EhPrimo := True;
    for i := 2 to n div 2 do
    begin
      if (n mod i = 0) then
      begin
        EhPrimo := False;
        Break;
      end;
    end;
  end;
end;

var
  num: Integer;
begin
  clrscr;
  Write('Digite um número: ');
  ReadLn(num);

  if EhPrimo(num) then
    WriteLn('O número ', num, ' é primo.')
  else
    WriteLn('O número ', num, ' não é primo.');

  ReadLn;
end.
```

Oberon

```oberon
MODULE PrimoCheck;
IMPORT In, Out;

PROCEDURE EhPrimo(n: INTEGER): BOOLEAN;
VAR i: INTEGER;
BEGIN
  IF n < 2 THEN RETURN FALSE END;
  FOR i := 2 TO n DIV 2 DO
    IF (n MOD i = 0) THEN RETURN FALSE END;
  END;
  RETURN TRUE;
END EhPrimo;

VAR num: INTEGER;
BEGIN
  Out.String("Digite um número: ");
  In.Int(num);

  IF EhPrimo(num) THEN
    Out.String("O número "); Out.Int(num, 0); Out.String(" é primo.");
  ELSE
    Out.String("O número "); Out.Int(num, 0); Out.String(" não é primo.");
  END;
  Out.Ln;
END PrimoCheck.
```

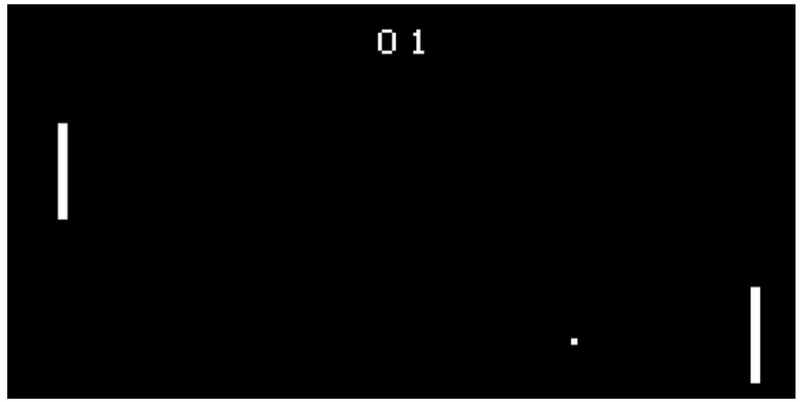
Pascal

```pascal
program Fibonacci;
uses crt;

function Fibonacci(n: integer): integer;
    if (n = 0) then
        Fibonacci := 0
    else if (n = 1) then
        Fibonacci := 1
    else
        Fibonacci := Fibonacci(n - 1) + Fibonacci(n - 2); //
end;

var
    i, num: integer;
begin
    clrscr;
    write('Digite um número: ');
    readln(num);
    writeln('Sequência de Fibonacci até ', num, ' termos:');
    for i := 0 to num - 1 do
        write(Fibonacci(i), ' ');
    writeln;
    readln;
end.
```

Haskell

```haskell
fibonacci :: Int -> Int
fibonacci 0 = 0
fibonacci 1 = 1
fibonacci n = fibonacci (n - 1) + fibonacci (n - 2)

main :: IO ()
main = do
putStrLn "Digite um número: "
input <- getLine
let num = read input :: Int
putStrLn $ "Sequência de Fibonacci até " ++ show num ++ " termos:"
print [fibonacci n | n <- [0..(num-1)]]
```

Exemplos práticos



Considerações finais

- Origem e evolução
- Características e paradigmas
- Comparativo com outras linguagens
- Influência
- Educação e formação

https://github.com/flp2113/lp-pascal-pong

Bibliografia

Free Pascal - Advanced open source Pascal compiler for Pascal and Object Pascal - Home Page. Disponível em: <https://www.freepascal.org/>.

MATTHIAS. Free Pascal meets SDL. Disponível em: <https://www.freepascal-meets-sdl.net/>.

WATT, D. A.; FINDLAY, W. Programming Language Design Concepts. [s.l.] John Wiley & Sons, 2004.

Pascal - Quick Guide. Disponível em: <https://www.tutorialspoint.com/pascal/pascal_quick_guide.htm>.

REISER, M.; WIRTH, N. Programming in Oberon. [s.l.] Addison-Wesley Longman, 1992.

JENSEN, K.; WIRTH, N. PASCAL User Manual and Report. [s.l.] Springer, 2013.

A Gentle Introduction to Haskell, Version 98. Disponível em: <https://www.haskell.org/tutorial/>.

Code Examples. Disponível em: <https://oberon-lang.github.io/>.

Obrigado.

PUC Minas