



APS Lógica da Computação

Beni Kracochansky

Junho 2023

Contribuindo para inclusão de novos programadores no mercado

Brasil é um dos maiores polos tecnológicos no mundo

Github

Estimado que mais de 3 milhões dos 100 milhões de usuários ativos são brasileiros

Escalabilidade

Crescimento ano a ano de usuários brasileiros no Github foi de 40%, configurando como o quarto país com maior crescimento de desenvolvedores

Crescimento

Brasscom indicou que o setor de TI no Brasil deverá crescer a uma taxa de 5,8% ao ano até 2024.



A Lingua Nova contribui para a democratização do acesso e aprendizado à programação pelos brasileiros

Democratização

A linguagem busca tornar a programação acessível para todos, independentemente da formação educacional ou experiência.

Acelerada

Sendo direta e intuitiva, facilita a rápida assimilação de conceitos fundamentais de programação para novos entrantes no campo.

Simples

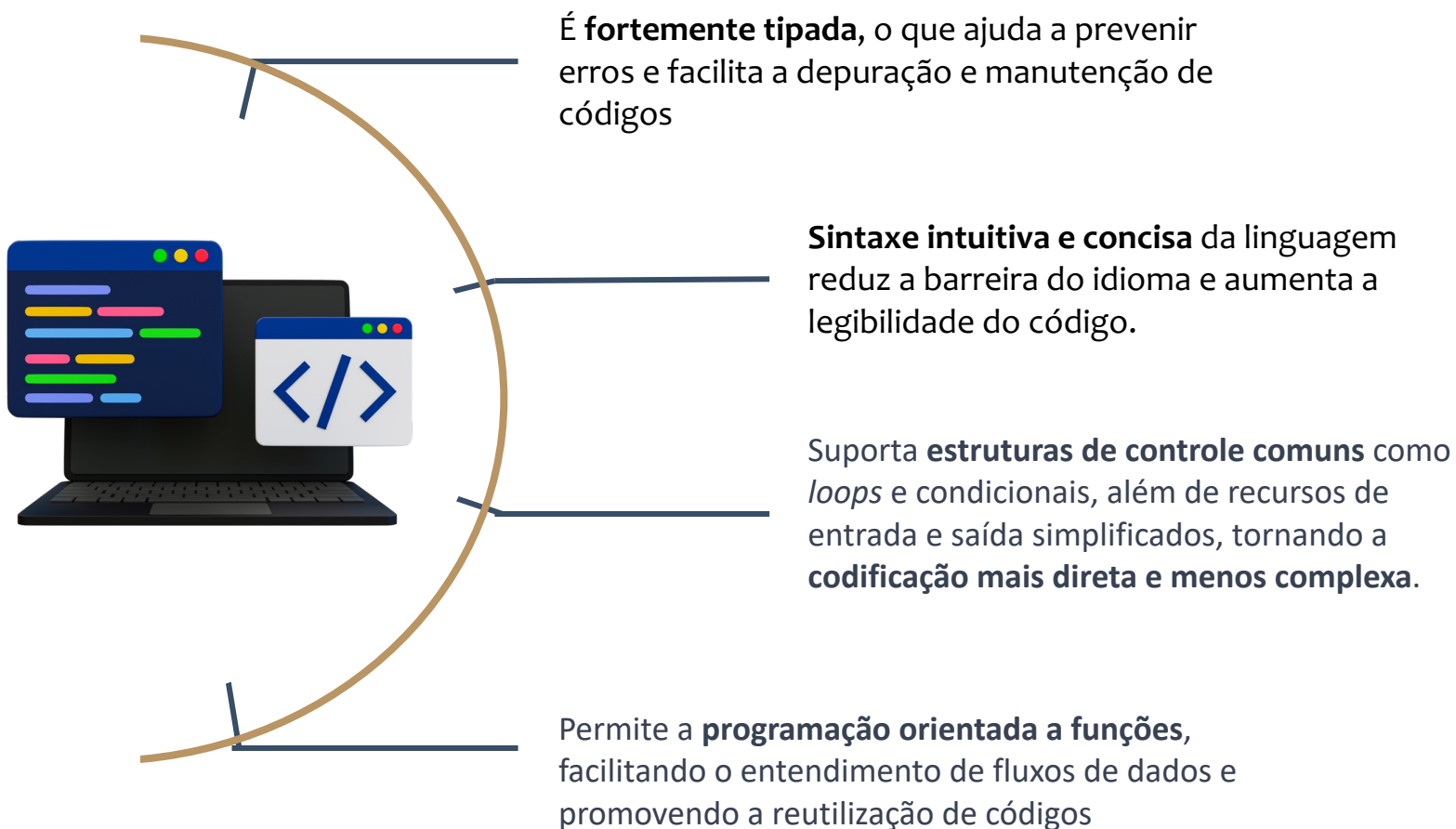
Desenvolvida com uma sintaxe clara e concisa, minimizando a complexidade desnecessária.

Inclusão

Contribui a construir uma comunidade inclusiva e acolhedora para todos os novos programadores.



Focamos em uma linguagem simples, que seja de aprendizado fácil e inclusiva



Destaques



Acessibilidade: democratiza a programação ao torná-la acessível a uma população mais ampla, usando o Português como idioma base.



Ferramenta Educacional: Excelente para introduzir conceitos de programação, de maneira intuitiva, em salas de aula ou plataformas de ensino online.



Potencial de Inovação: fomenta a inovação tecnológica, possibilitando a emergência de novas soluções e projetos.

A Lingua Nova segue as principais características da linguagem Julia, mas traduzido para o português

Linguagem Julia

```
# Definindo uma função de adição
function adiciona(val1::Int, val2::Int)::Int
    return val1 + val2
end

# Definindo uma variável com tipagem
numPrincipal::Int = 2
numPrincipal = adiciona(1, numPrincipal)

println("Por favor, insira um número: ")
numPrincipal = parse{Int, readline()}

if ((numPrincipal > 1) && !(numPrincipal < 1))
    numPrincipal = 3
else
    numPrincipal = (-20+30)*4*3/40
end

println(numPrincipal)

println("Por favor, insira outro número: ")
numPrincipal = parse{Int, readline()}

if ((numPrincipal > 1) && !(numPrincipal < 1))
    numPrincipal = 3
else
    numPrincipal = (-20+30)*12/40
end

println(numPrincipal)

while ((numPrincipal > 1) || (numPrincipal == 1))
    numPrincipal = numPrincipal - 1
    println(numPrincipal)
end
```



Lingua Nova

```
# Definindo uma função de adição
funcao adiciona(val1::Int, val2::Int)::Int
    retorne val1 + val2
fim

# Definindo uma variável com tipagem
numPrincipal::Int
numPrincipal = 2
numPrincipal = adiciona(1, numPrincipal)

numPrincipal = leia()

se ((numPrincipal > 1) && !(numPrincipal < 1))
    numPrincipal = 3
senao
    numPrincipal = (-20+30)*4*3/40
fim

imprima(numPrincipal)

numPrincipal = leia()
    You, 8 minutes ago • Uncommitted changes
se ((numPrincipal > 1) && !(numPrincipal < 1))
    numPrincipal = 3
senao
    numPrincipal = (-20+30)*12/40
fim

imprima(numPrincipal)

enquanto ((numPrincipal > 1) || (numPrincipal == 1))
    numPrincipal = numPrincipal - 1
    imprima(numPrincipal)
fim
```