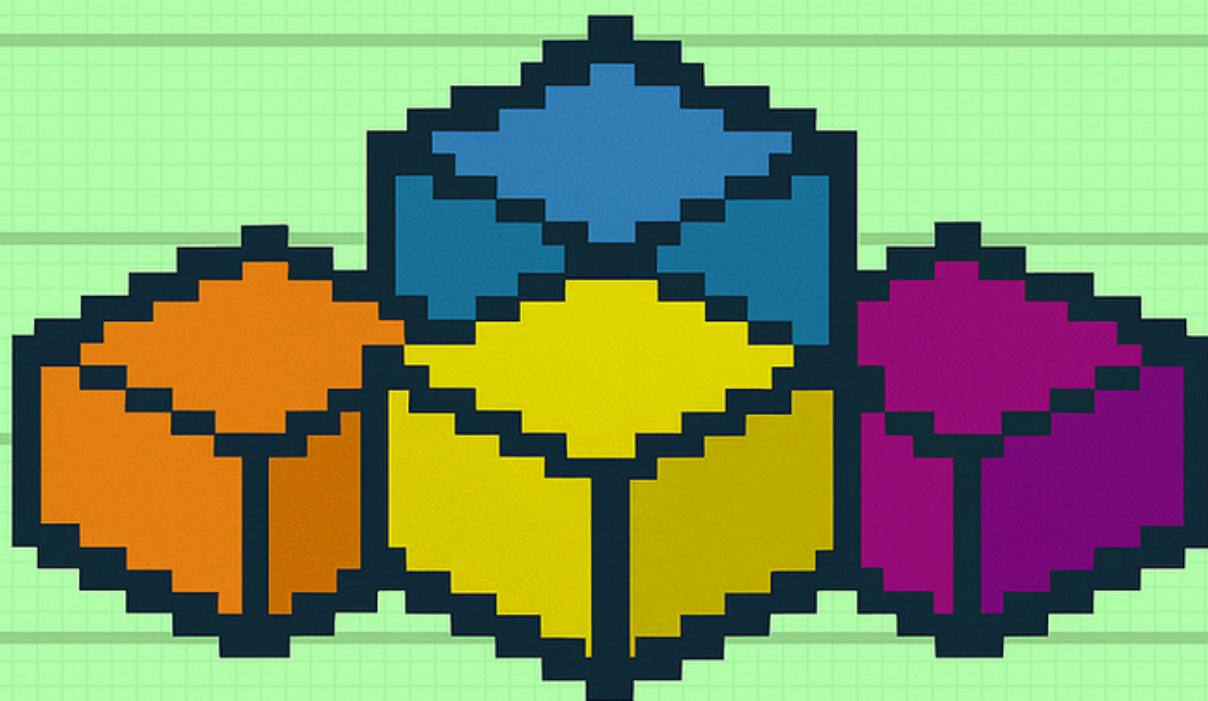


# VBA



## NEXT LEVEL

### DOMINE O JOGO DAS PLANILHAS



# VBA



# Sobre a Autora



Oi! Eu sou a Nayara, apaixonada por automação e pela ideia de transformar tarefas chatas em soluções inteligentes. Trabalho de forma autônoma e há pouco mais de um ano mergulhei no mundo do VBA, descobrindo como algumas linhas de código podem mudar completamente o ritmo de um dia de trabalho.

Este eBook nasceu durante o curso de Fundamentos de IA Generativa da DIO, como um projeto para compartilhar o que aprendi e mostrar que programar no Excel não precisa ser complicado. Gosto de criar conteúdos que simplificam o aprendizado e ajudam outras pessoas a enxergar que automatizar é mais fácil (e divertido!) do que parece.

Introdução

**VBA**

O que é VBA?

# O que é VBA?

O Excel é uma ferramenta incrível. Mas, por trás das planilhas e fórmulas, existe um recurso que poucos exploram: o VBA (Visual Basic for Applications).

O VBA é uma linguagem de programação que vive dentro do Excel e permite que você automatize tarefas, crie relatórios e desenvolva soluções sob medida. Em vez de repetir as mesmas ações todos os dias, você ensina o Excel a fazer isso por você — de forma rápida, precisa e inteligente.

Aprender VBA é como desbloquear um novo nível dentro do Excel. Você deixa de apenas preencher planilhas e passa a comandá-las. Um simples código pode atualizar relatórios, enviar e-mails, formatar tabelas e até conversar com outros programas.

E o melhor: você não precisa ser programador para começar. A linguagem é simples, direta e fala a língua de quem já usa o Excel no dia a dia.



Pense no VBA como um copiloto: enquanto você decide o que precisa ser feito, ele cuida do resto.

# Capítulo 01

**Desbloqueando o poder do  
VBA**



# Desbloqueando o poder do VBA

O VBA é o motor que faz o Excel ganhar vida. Ele observa o que você faz e é capaz de repetir essas ações em segundos. Tarefas manuais, como copiar dados ou formatar relatórios, podem ser automatizadas com poucos comandos.

Quando você cria uma macro, está literalmente programando o Excel para pensar como você.

Veja um exemplo simples:

```
VBA

Sub AtualizarRelatorio()
    Sheets("Base").Range("A1:D100").Copy
    Sheets("Relatório").Range("A1").PasteSpecial xlPasteValues
    MsgBox "Relatório atualizado com sucesso!"
End Sub
```

Esse código copia dados da planilha “Base”, cola no “Relatório” e avisa quando termina.

Com apenas três linhas, você economiza minutos — ou até horas — de trabalho repetitivo.



Automatizar é ganhar tempo para o que realmente importa.

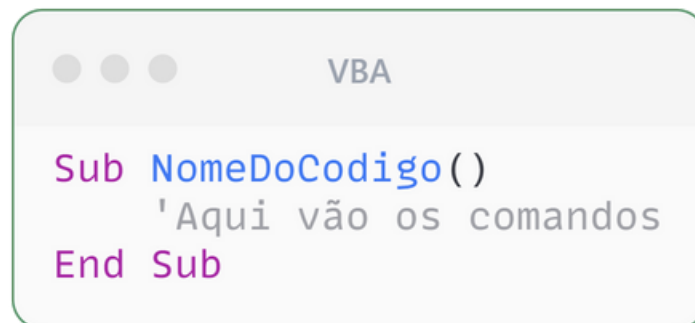
# Capítulo 02

Como funciona um código  
VBA

# Como funciona um código VBA

Todo código VBA segue uma lógica muito parecida: começa, executa uma sequência de ações e termina.

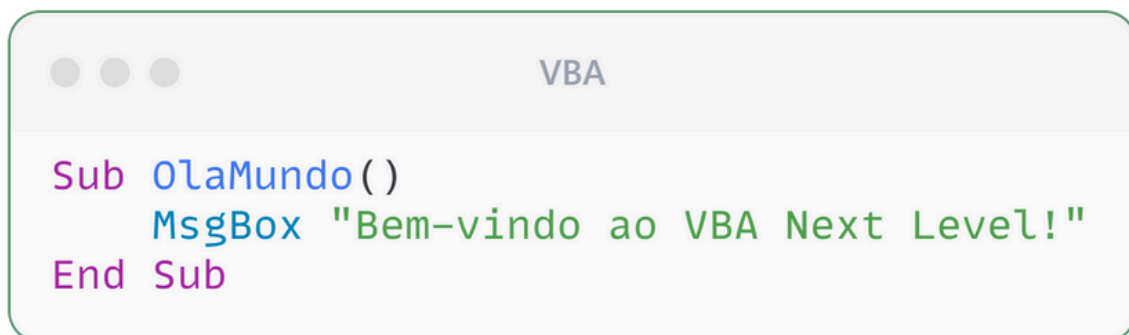
A estrutura básica é simples:

A diagrama de uma janela de código VBA com o título "VBA". O código dentro da janela é: 

```
Sub NomeDoCodigo()  
    'Aqui vão os comandos  
End Sub
```

```
Sub NomeDoCodigo()  
    'Aqui vão os comandos  
End Sub
```

Dentro desses blocos, você escreve o que o Excel deve fazer. Veja o exemplo clássico do “Olá Mundo”, o primeiro código de quase todo programador:

A diagrama de uma janela de código VBA com o título "VBA". O código dentro da janela é: 

```
Sub OlaMundo()  
    MsgBox "Bem-vindo ao VBA Next Level!"  
End Sub
```

```
Sub OlaMundo()  
    MsgBox "Bem-vindo ao VBA Next Level!"  
End Sub
```

Quando você aperta F5, o Excel mostra uma mensagem na tela.

Simples assim. É o primeiro passo para entender como o VBA “fala” com você.

A partir disso, você pode criar módulos com várias macros diferentes — cada uma automatizando uma parte do seu trabalho.

Com o tempo, o VBA se torna o seu atalho para tudo que o Excel não faz sozinho.



Capítulo

03

Variáveis e decisões  
inteligentes

# Variáveis e decisões inteligentes

As variáveis são como a memória do seu código.

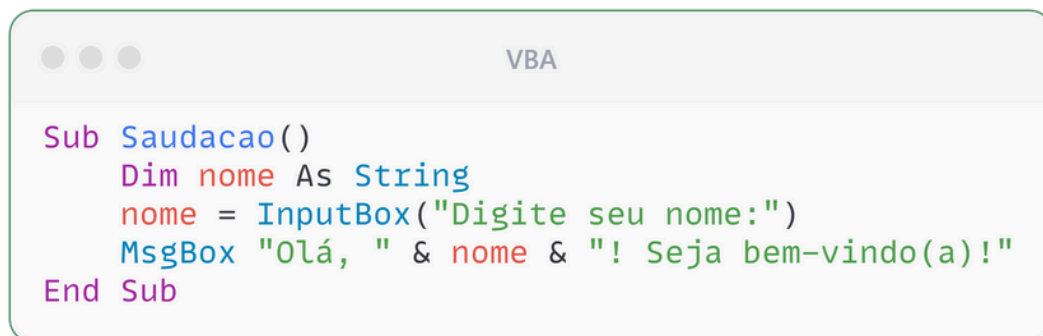
Elas guardam informações temporariamente enquanto o VBA executa as ações.

Pense nelas como pequenas caixas nomeadas, onde você pode guardar números, textos, datas e resultados para usar quando quiser.

Quando você trabalha no Excel, cada célula pode armazenar um valor.

No VBA, quem faz esse papel são as variáveis — só que dentro do código.

Vamos ver um exemplo simples:

A screenshot of a VBA code editor window. The window has a title bar with three colored circles (red, yellow, green) on the left and the text 'VBA' in the center. The code inside is as follows:

```
Sub Saudacao()  
    Dim nome As String  
    nome = InputBox("Digite seu nome:")  
    MsgBox "Olá, " & nome & "! Seja bem-vindo(a)!"  
End Sub
```

Aqui, a variável nome foi criada para guardar o texto que o usuário digita.

O VBA então usa esse valor para exibir uma saudação personalizada.

A palavra Dim significa dimensionar — é ela que “cria” o espaço de memória para o dado.

Existem diferentes tipos de variáveis, e escolher o tipo certo ajuda o código a rodar melhor:

- *String*: guarda textos
- *Integer*: números inteiros
- *Double*: números decimais
- *Date*: datas
- *Boolean*: valores Verdadeiro ou Falso

Mas o VBA também pode tomar decisões, usando estruturas como o If.

Esse comando é o que dá lógica ao código, permitindo que ele reaja aos dados em vez de apenas seguir ordens fixas.

É como ensinar o Excel a analisar uma situação e escolher o que fazer a partir disso.

```
VBA

Sub VerificarNota()
    Dim nota As Double
    nota = Range("B2").Value

    If nota ≥ 7 Then
        MsgBox "Aprovado!"
    Else
        MsgBox "Reprovado!"
    End If
End Sub
```

Agora o Excel é capaz de “pensar”. Ele avalia a nota e responde automaticamente.



Com variáveis e condições, você ensina o Excel a se adaptar a cada situação.

Capítulo

04

Repetição e automação em  
massa

# Repetição e automação em massa

Um dos maiores poderes do VBA é a repetição.

Com os loops, você ensina o Excel a percorrer linhas, colunas e até planilhas inteiras, aplicando as mesmas ações em segundos.

Veja um exemplo simples:

```
VBA

Sub ColorirLinhas()
    Dim i As Integer
    For i = 2 To 50
        If Cells(i, 1).Value <> "" Then
            Cells(i, 1).Interior.Color = vbGreen
        End If
    Next i
End Sub
```

Esse código verifica todas as células da coluna A e pinta de verde as que têm conteúdo.

Você pode ajustar o intervalo para centenas ou milhares de linhas — e o VBA faz tudo sozinho.



Na prática, isso é útil para validar dados, gerar relatórios ou formatar planilhas inteiras de uma vez.

*Quando você domina os loops, o Excel deixa de ser apenas uma planilha e vira um verdadeiro motor de automação.*

# Capítulo 05

**Trabalhando com planilhas e  
funções personalizadas**



# Trabalhando com planilhas e funções personalizadas

O VBA permite acessar qualquer parte da planilha e fazer cálculos automáticos.

Você pode atualizar células, somar valores e até criar funções novas, como se estivesse expandindo o Excel.

## ***Exemplo de cálculo automático:***

```
Sub AtualizarCusto()  
    Dim ws As Worksheet  
    Set ws = Sheets("Financeiro")  
    ws.Range("D2").Value = ws.Range("B2").Value * ws.Range("C2").Value  
End Sub
```

## ***Exemplo de função personalizada:***

```
Function SomaComBonus(a As Double, b As Double) As Double  
    SomaComBonus = (a + b) * 1.1  
End Function
```

Depois de criar essa função, basta digitar na planilha:  
**=SomaComBonus(A1;B1)**



Criar funções próprias é como adicionar superpoderes ao Excel. Você constrói fórmulas sob medida para o seu trabalho.

Capítulo

06

Montando seu primeiro  
sistema

# Montando seu primeiro sistema

Agora que você já conhece o básico, é hora de juntar tudo. Com variáveis, loops e condições, dá para montar um pequeno sistema que lê dados, calcula e mostra resultados.

```
VBA

Sub GerarRelatorioFinal()
    Dim ws As Worksheet
    Dim i As Long, total As Double

    ' Define qual planilha será usada
    Set ws = ThisWorkbook.Sheets("Relatório")

    For i = 2 To 20
        If ws.Cells(i, 1).Value = "" Then Exit For
        ws.Cells(i, 5).Value = ws.Cells(i, 2).Value * ws.Cells(i, 3).Value
        total = total + ws.Cells(i, 5).Value
    Next i

    ws.Range("F2").Value = total
    MsgBox "Relatório concluído. Total: " & Format(total, "R$ #,##0.00")
End Sub
```

Esse código lê uma tabela, calcula valores e soma o total automaticamente.

Em segundos, você tem um relatório completo, formatado e pronto para análise.



O VBA transforma planilhas simples em sistemas inteligentes — feitos por você, para o seu jeito de trabalhar.

# Considerações **FINAIS**

# Considerações Finais

Você acaba de conhecer a base do VBA:

- Como funciona o código;
- Como usar variáveis e condições;
- Como repetir tarefas e criar funções personalizadas.

Esses são os primeiros blocos do que pode se tornar uma ferramenta poderosa nas suas mãos.

Com um pouco de prática, você vai automatizar relatórios, conectar dados e criar planilhas que praticamente se gerenciam sozinhas.

Mais do que uma linguagem de programação, o VBA é uma nova forma de enxergar o Excel.

Ele permite que você vá além das fórmulas e comece a construir soluções completas, feitas sob medida para o seu trabalho.

Com o tempo, você vai perceber que o que antes levava horas pode ser feito em segundos — e que cada linha de código escrita é um passo em direção à produtividade real.

Não tenha medo de errar, testar ou experimentar. Todo programador começou com um simples MsgBox "Olá Mundo".

O segredo é praticar um pouco a cada dia, observar o que o Excel faz e tentar transformar essas ações em código.

Logo, você estará criando automações que resolvem problemas reais e economizam tempo de verdade.



Automatizar é mais do que economizar tempo. É abrir espaço para pensar, criar e evoluir.