

## Referências Bibliográficas

1. Amram, M., Kulatilaka, N. (1999). Real Options. Managing Strategic Investment in an Uncertain World. *Harvard Business School Press*.
2. Barone-Adesi, G.; Whaley, R. (1987) "Efficient analytic approximation of American option values." *Journal of Finance*, 42(2). 301-320.
3. Capozza, D.; Sick, G. (1994). "The Risk Structure of Land Markets". *Journal of Urban Economics*, 35. 297-319.
4. Dixit, A.; Pindyck R. S. (1994). Investment Under Uncertainty, *Princeton University Press*.
5. Hull, J. C. (2003): Options, Futures and Other Derivatives, Fifth Edition, *Prentice Hall*.
6. Majd, S. e Robert Pindyck (1987), Time to Build, Option Value, and Investment Decision, *Journal of Financial Economics* 18:1, p. 7-27
7. Medeiros, Priscilla Yung (2001), Aplicação de Opções Reais no Mercado Imobiliário residencial com enfoque na cidade do Rio de Janeiro, Dissertação de Mestrado, *Departamento de Economia, PUC-Rio*.
8. Titman, S. (1985). "Urban Land Prices Under Uncertainty". *American Economic Review*, 75 (3). 505-514.
9. Trigeorgis, L. (1996) Real Options: Managerial Flexibility and Strategy in Resource Allocation. *MIT Press*.
10. Williams, J. (1991). "Real Estate Development as an Option". *Journal of Real Estate Finance and Economics*, 4. 191-208.
11. Rubinstein, R. Y. (1981), "Simulation and The Monte Carlo Method". John Wiley & Sons, Inc.
12. Ross, S. M. (2002), "Simulation". Third Edition, Academic Press.

## Apêndice 1 – Resumo de utilização do programa desenvolvido

1º passo: Preencher os dados referentes ao lançamento do 1º empreendimento

- Eventos
  - ✓ Aquisição do Terreno
  - ✓ Lançamento
  - ✓ Início de Construção
  - ✓ Prazo da Obra
- Entrar no quadro de áreas e calcular a área privativa e a área equivalente de construção
- Determinar o percentual das despesas de venda
  - ✓ Corretagem
  - ✓ Publicidade
  - ✓ Pis/Cofins
  - ✓ Gestão
- Determinar o valor do terreno
  - ✓ Dinheiro
  - ✓ Permuta em VGV no local
  - ✓ Permuta de unidades no local
  - ✓ Permuta fora
- Estimar o projeto de arquitetura como um percentual do custo de construção
- Determinar o Custo/m<sup>2</sup> de construção, assim como o BDI
- Determinar um percentual sobre o VGV em relação às despesas legais e jurídicas
- Determinar a taxa de desconto do fluxo de caixa, ou seja, o custo de oportunidade do empreendimento
- Preenchido estes dados encontramos todos os indicadores do fluxo de caixa e do estático

2º passo: Preencher os dados referentes à expansão

- No caso do lançamento dinâmico o custo/m<sup>2</sup> será maior do que o custo/m<sup>2</sup> do 1º lançamento

3º passo: Entrar na planilha simulação

- Determinar a média da velocidade de vendas do lançamento
- Clicar na Macro: Calculo com Opções Reais
  - ✓ Guardar o valor do empreendimento -  $VPL_{EXPANDIDO}$
  - ✓ Analisar a distribuição de probabilidade do VPL e o seu respectivo risco
- Clicar na Macro: Calculo Lançamento Simultâneo
  - ✓ Guardar o valor do empreendimento -  $VPL_{TRADICIONAL}$
  - ✓ Analisar a distribuição de probabilidade do VPL e o seu respectivo risco
- Calcular o valor da opção através da diferença dos VPL's
  - ✓ Opção =  $VPL_{EXPANDIDO} - VPL_{TRADICIONAL}$ 
    - Se valor da Opção > Direito de Exclusividade
      - Incorporador compra o direito de exclusividade

## Apêndice 2 - Código do Programa

O código abaixo foi escrito para implementar a teoria das opções reais no Excel. O código abaixo foi escrito em VBA (Visual Basic for Applications).

---

```
'=====
' BOTAO PARA RODAR A SIMULACAO COM OPCOES REAIS
'=====

Private Sub CommandButton1_Click()
    'Informa se a espera na expansão foi um sucesso
    Dim esperaExpansao
    'Variavel que guarado o numero de sucessos
    Dim NumSucesso
    'Vetor de VPL1s
    Dim VetorVPL1s(10000)
    'Vetor de VPL2s
    Dim VetorVPL2s(10000)
    'Incia com ZERO sucessos
    NumSucesso = 0
    'imprime que esta rodando a simulação
    Plan1.Cells(8, 6).Value = "Rodando Simulação"
    'Calcula a curva de gatilho
    CalcularGatilho
    'Calcula a quantidade de Sucessos
    'sorteando a velocidade de vendas
    For i = 1 To Plan1.Cells(3, 7).Value
        'Imprime o quantos por cento da simulacao
        'Ja foi executada
        Plan1.Cells(8, 7).Value = (i / Plan1.Cells(3, 7).Value)
        'sortea a velocidade de vendas
```

CalcularVelocidadesVendas

'Considera o preco por metro quadrado na expansao 10% maior

'(sucesso)

Plan10.Cells(6, 9).Value = Plan10.Cells(5, 9).Value \* 1.1

'verifica se empreendimento foi um sucesso

If (Plan1.Cells(40, 2) = "SUCESSO") Then

NumSucesso = NumSucesso + 1

Plan9.Cells(5 + i, 5).Value = "S"

'Guarda o VPL calculado nesta etapa

VetorVPL1s(i) = Plan1.Cells(28, 2).Value

VetorVPL2s(i) = Plan1.Cells(29, 2).Value

Else

esperaExpansao = TratarFracasso

Plan9.Cells(5 + i, 5).Value = "F"

'Se VP do VGV bateu na curva de gatilho

'entao esperaExpansao foi um sucesso (True)

'senao esperaExpansao foi um fracasso (False)

If (esperaExpansao = True) Then

'Guarda o VPL calculado nesta etapa

VetorVPL1s(i) = Plan1.Cells(28, 2).Value

VetorVPL2s(i) = Plan1.Cells(29, 2).Value

Else

'Guarda o VPL calculado nesta etapa

'Mas VPL2 = 0

VetorVPL1s(i) = Plan1.Cells(28, 2).Value

VetorVPL2s(i) = 0

End If

End If

Next i

'imprime o numero de sucessos

Plan9.Cells(1, 2).Value = NumSucesso

'limpa o resultado da ultima simulacao

For i = 1 To 100

Plan9.Cells(5 + i, 1).Value = ""

```

        Plan9.Cells(5 + i, 2).Value = ""
        Plan9.Cells(5 + i, 3).Value = ""
    Next i
    'imprime todos os VPLs calculados
    For i = 1 To Plan1.Cells(3, 7).Value
        Plan9.Cells(5 + i, 1).Value = i
        Plan9.Cells(5 + i, 2).Value = VetorVPL1s(i)
        Plan9.Cells(5 + i, 3).Value = VetorVPL2s(i)
    Next i
    'Calcula a distribuicao de probabilidade
    CalculaDistribuicaoProbabilidade
    'imprime que esta rodando a simulacao
    Plan1.Cells(8, 6).Value = "Simulação Terminou!"
End Sub

```

---

```

'=====
' BOTAO PARA RODAR A SIMULACAO COM LANÇAMENTO
'SIMULTANEO
'=====

Private Sub CommandButton2_Click()
    'Variavel que guarado o numero de sucessos
    Dim NumSucesso
    'Vetor de VPL1s
    Dim VetorVPL1s(10000)
    'Vetor de VPL2s
    Dim VetorVPL2s(10000)
    'Incia com ZERO sucessos
    NumSucesso = 0
    'imprime que esta rodando a simulacao
    Plan1.Cells(8, 6).Value = "Rodando Simulação"
    For i = 1 To Plan1.Cells(3, 7).Value
        'Imprime o quantos por cento da simulacao
        'Ja foi executada
        Plan1.Cells(8, 7).Value = (i / Plan1.Cells(3, 7).Value)
    Next i

```

```

'sorteia a velocidade de vendas
CalcularVelocidadesVendas
'Considera o preco por metro quadrado na expansao igual da
'primeira fase
Plan10.Cells(6, 9).Value = Plan10.Cells(5, 9).Value
'verifica se empreendimento foi um sucesso
If (Plan1.Cells(40, 2) = "SUCESSO") Then
    NumSucesso = NumSucesso + 1
    Plan9.Cells(5 + i, 5).Value = "S"
    'Guarda o VPL calculado nesta etapa
    VetorVPL1s(i) = Plan1.Cells(28, 2).Value
    VetorVPL2s(i) = Plan1.Cells(29, 2).Value
Else
    Plan9.Cells(5 + i, 5).Value = "F"
    'Guarda o VPL calculado nesta etapa
    VetorVPL1s(i) = Plan1.Cells(28, 2).Value
    VetorVPL2s(i) = Plan1.Cells(29, 2).Value
End If
Next i
'imprime o numero de sucessos
Plan9.Cells(1, 2).Value = NumSucesso
'limpa o resultado da ultima simulacao
For i = 1 To 100
    Plan9.Cells(5 + i, 1).Value = ""
    Plan9.Cells(5 + i, 2).Value = ""
    Plan9.Cells(5 + i, 3).Value = ""
Next i
'imprime todos os VPLs calculados
For i = 1 To Plan1.Cells(3, 7).Value
    Plan9.Cells(5 + i, 1).Value = i
    Plan9.Cells(5 + i, 2).Value = VetorVPL1s(i)
    Plan9.Cells(5 + i, 3).Value = VetorVPL2s(i)
Next i
'Calcula a distribuicao de probabilidade

```

CalculaDistribuicaoProbabilidade

'imprime que esta rodando a simulação

Plan1.Cells(8, 6).Value = "Simulação Terminou!"

End Sub

---

Function TRIANINV(ByVal probability As Single, ByVal lowerbound  
As Single, ByVal mostlikely As Single, ByVal upperbound As Single)

On Error GoTo 18

Dim x As Single

If probability > 1 Or probability < 0 Then GoTo 18

If lowerbound >= upperbound Then GoTo 18

x = (mostlikely - lowerbound) / (upperbound - lowerbound)

If (x > 1 Or x < 0) Then GoTo 18

If probability <= x Then

    TRIANINV = lowerbound +

    (upperbound - lowerbound) \* ((probability \* x) ^ 0.5)

If probability > x Then

    TRIANINV = upperbound -

    (upperbound - lowerbound) \* (((1 - probability) \* (1 - x)) ^ 0.5)

Exit Function

18 TRIANINV = CVErr(xlErrNum)

End Function

---

Function CalcularVelocidadesVendas()

'Var q guarda o número da dist. uniforme

Dim uniNum

'Var q guarda o valor min da dist triangular

Dim min

'Var q guarda o valor max da dist triangular

Dim max

'Var q guarda a média da dist triangular

Dim media

'=====



'CALCULANDO A VELOCIDADE DE VENDAS NO  
'LANCAMENTO

'=====

'Gerar o número da dist. uniforme

Randomize

uniNum = Rnd

'Pegar os valores na planilha (min, max e média)

min = Plan1.Cells(4, 2).Value

max = Plan1.Cells(5, 2).Value

media = Plan1.Cells(6, 2).Value

'Colocar o valor de uniNum na planilha

Plan1.Cells(7, 2).Value = uniNum

'Gerar o número da dist. triangular e colocar o valor na planilha

Plan1.Cells(8, 2).Value = TRIANINV(uniNum, min, media, max)

'=====

'CALCULANDO A VELOCIDADE DE VENDAS ENTRE O  
'LANCAMENTO E O TERMINO DA OBRA

'=====

'Gerar o número da dist. uniforme

Randomize

uniNum = Rnd

'Pegar os valores na planilha (min, max e média)

min = Plan1.Cells(13, 2).Value

max = Plan1.Cells(14, 2).Value

media = Plan1.Cells(15, 2).Value

'Colocar o valor de uniNum na planilha

Plan1.Cells(16, 2).Value = uniNum

'Gerar o número da dist. triangular e colocar o valor na planilha

Plan1.Cells(17, 2).Value = TRIANINV(uniNum, min, media, max)

'=====

'CALCULANDO A VELOCIDADE NO TERMINO DA OBRA

'=====

'Gerar o número da dist. uniforme

Randomize

```

uniNum = Rnd
'Pegar os valores na planilha (min, max e média)
min = Plan1.Cells(22, 2).Value
max = Plan1.Cells(23, 2).Value
media = Plan1.Cells(24, 2).Value
'Colocar o valor de uniNum na planilha
Plan1.Cells(25, 2).Value = uniNum
'Gerar o número da dist. triangular e colocar o valor na planilha
Plan1.Cells(26, 2).Value = TRIANINV(uniNum, min, media, max)
End Function

```

---

```

Function CalculaDistribuicaoProbabilidade()
'Variavel que guarda o menor dos VPLs
Dim minVPL
minVPL = 1E+29
'Variavel que guarda o maior dos VPLs
Dim maxVPL
maxVPL = -1E+29
'varre todos os VPLS para calcular o menor e o maior
For i = 1 To Plan1.Cells(3, 7).Value
    If (minVPL > Plan9.Cells(5 + i, 4).Value) Then
        minVPL = Plan9.Cells(5 + i, 4).Value
    End If
    If (maxVPL < Plan9.Cells(5 + i, 4).Value) Then
        maxVPL = Plan9.Cells(5 + i, 4).Value
    End If
Next i
'Imprime o valor do menor e maior VPL
Plan9.Cells(24, 7).Value = minVPL
Plan9.Cells(43, 7).Value = maxVPL
'Zera os Contadores dos intervalos
For i = 1 To 19
    Plan9.Cells(24 + i, 8).Value = 0

```

Next i

'Conta Quantos Numeros tem entre os intervalos

For i = 1 To Plan1.Cells(3, 7).Value

If (Plan9.Cells(5 + i, 4).Value < Plan9.Cells(25, 7).Value) Then

Plan9.Cells(25, 8).Value = Plan9.Cells(25, 8).Value + 1

End If

If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(26, 7).Value

And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(25, 7).Value)

Then

Plan9.Cells(26, 8).Value = Plan9.Cells(26, 8).Value + 1

End If

If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(27, 7).Value

And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(26, 7).Value)

Then

Plan9.Cells(27, 8).Value = Plan9.Cells(27, 8).Value + 1

End If

If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(28, 7).Value

And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(27, 7).Value)

Then

Plan9.Cells(28, 8).Value = Plan9.Cells(28, 8).Value + 1

End If

If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(29, 7).Value

And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(28, 7).Value)

Then

Plan9.Cells(29, 8).Value = Plan9.Cells(29, 8).Value + 1

End If

If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(30, 7).Value

And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(29, 7).Value)

Then

Plan9.Cells(30, 8).Value = Plan9.Cells(30, 8).Value + 1

End If

If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(31, 7).Value

And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(30, 7).Value)

Then

```

    Plan9.Cells(31, 8).Value = Plan9.Cells(31, 8).Value + 1
End If
If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(32, 7).Value
    And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(31, 7).Value)
    Then
        Plan9.Cells(32, 8).Value = Plan9.Cells(32, 8).Value + 1
    End If
If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(33, 7).Value
    And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(32, 7).Value)
    Then
        Plan9.Cells(33, 8).Value = Plan9.Cells(33, 8).Value + 1
    End If
If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(34, 7).Value
    And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(33, 7).Value)
    Then
        Plan9.Cells(34, 8).Value = Plan9.Cells(34, 8).Value + 1
    End If
If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(35, 7).Value
    And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(34, 7).Value)
    Then
        Plan9.Cells(35, 8).Value = Plan9.Cells(35, 8).Value + 1
    End If
If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(36, 7).Value
    And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(35, 7).Value)
    Then
        Plan9.Cells(36, 8).Value = Plan9.Cells(36, 8).Value + 1
    End If
If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(37, 7).Value
    And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(36, 7).Value)
    Then
        Plan9.Cells(37, 8).Value = Plan9.Cells(37, 8).Value + 1
    End If
If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(38, 7).Value
    And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(37, 7).Value)

```

```

    Then
        Plan9.Cells(38, 8).Value = Plan9.Cells(38, 8).Value + 1
    End If
    If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(39, 7).Value
        And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(38, 7).Value)
        Then
            Plan9.Cells(39, 8).Value = Plan9.Cells(39, 8).Value + 1
        End If
    If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(40, 7).Value
        And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(39, 7).Value)
        Then
            Plan9.Cells(40, 8).Value = Plan9.Cells(40, 8).Value + 1
        End If
    If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(41, 7).Value
        And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(40, 7).Value)
        Then
            Plan9.Cells(41, 8).Value = Plan9.Cells(41, 8).Value + 1
        End If
    If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(42, 7).Value
        And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(41, 7).Value)
        Then
            Plan9.Cells(42, 8).Value = Plan9.Cells(42, 8).Value + 1
        End If
    If (Plan9.Cells(5 + i, 4).Value <= Plan9.Cells(43, 7).Value
        And Plan9.Cells(5 + i, 4).Value > Plan9.Cells(42, 7).Value)
        Then
            Plan9.Cells(43, 8).Value = Plan9.Cells(43, 8).Value + 1
        End If
    Next i
End Function

```

---

Function TratarFracasso() As Boolean

'Informa se o maximo de iteracoes foram alcncadas

Dim maxIteracoes

```

maxIteracoes = False
'Var q guarda o número da dist. uniforme
Dim uniNum
'Contador de Meses
Dim meses
meses = 0
'Var q guarda o número da dist. normal padrao
Dim normalPadraoNum
'limpa a planilha com resultados antigos
For i = 1 To (Plan11.Cells(9, 2).Value * 12) + 1
    Plan10.Cells(9 + i, 3).Value = ""
    Plan10.Cells(9 + i, 6).Value = ""
Next i
'Calcula a curva de gatilho para cada novo preco (metro
'quadrado)
'O loop para se o numero de iteracoes (meses) for maior do que N
'ou VPL(VGV) for maior do que a curva de gatilho
Do
    'incrementa o numero de meses
    meses = meses + 1
    'Gerar o número da dist. uniforme
    Randomize
    uniNum = Rnd
    'Coloca na planilha o valor desse numero aleatorio
    Plan10.Cells(9 + meses, 3).Value = uniNum
    'Coloca o novo preco do metro quadrado como entrada
    'para a planilha que calcula a expansao
    Plan10.Cells(6, 9).Value = Plan10.Cells(9 + meses, 2).Value
    'Copia o valor do VPL do VGV calculado na planilha da
    'Expansao
    Plan10.Cells(9 + meses, 6).Value = Plan10.Cells(7, 9).Value
    'Verifica se o VPL(VGV) e maior do que a curva de gatilho
    'se VERDADE => sai do loop
    If (Plan10.Cells(9 + meses, 6).Value >

```

```

        Plan10.Cells(9 + meses, 7).Value) Then
            maxIteracoes = True
            Exit Do
        End If
        'Verifica se o numero de meses e maior do que o MAX definido
        'pela curva de gatilho
        'se VERDADE => sai do loop
        If (meses > Plan11.Cells(9, 2).Value * 12) Then
            maxIteracoes = False
            Exit Do
        End If
    Loop
    TratarFracasso = maxIteracoes
End Function

```

---

```

Function CalcularGatilho()
    'limpa a planilha da curva de gatilho
    For i = 1 To 60
        Plan10.Cells(9 + i, 7).Value = ""
    Next i
    'A tela da Curva de Gatilho precisa estar ativa
    'para o solver funcionar
    Plan11.Activate
    For i = 1 To Plan11.Cells(9, 2).Value * 12
        'Calcular a curva de gatilho usando o Solver
        'Primeiro vou colocar o valor de "t" na equacao da curva de
        'gatilho
        Plan11.Cells(12, 2).Value = Plan10.Cells(7, 3).Value *
        Plan10.Cells(9 + i, 1).Value
        'Roda o Solver do Excel para calcular a curva de gatilho
        SolverSolve (True)
        SolverFinish
        'Guarda o valor de S* (Resultado da Curva de Gatilho)
        Plan10.Cells(9 + i, 7).Value = Plan11.Cells(16, 2).Value
    Next i
End Function

```

```
Next i
' Quando t=5 entao S* = K
Plan10.Cells(70, 7).Value = Plan11.Cells(6, 2).Value
'Volta a tela da simulacao
Plan1.Activate
End Function
```