

**e**mbarcadero®

# EMBARCADERO CONFERENCE 2025

A JORNADA DO HERÓI | 30 ANOS DE INOVAÇÃO





# Aplicações Win64 como posso evoluir do Win32

Juliomar Marchetti



Quer me ver na  
**#ECON26?**  
Acesse o QRCode pelo App  
e avalie minha palestra!



@juliomarmarchetti

juliomarmarchetti

juliomarmarchetti@gmail.com

(49) 9 8426-8589

<https://www.juliomarmarchetti.com.br>

# Agenda:

1. Arquitetura x86: A Base de Tudo

- 1. Arquitetura x86: A Base de Tudo
- 2. Arquitetura x64: A Evolução Necessária
- 3. Delphi XE2: O Marco do Win64
- 4. VCL Win64: Performance Nativa
- 5. FMX Win64: Multiplataforma Poderoso
- 6. Comparativo VCL vs FMX
- 7. Alocação de Memória: 32 vs 64 bits
- 8. Performance de Processadores
- 9. Comparação de Cálculos: Código na Prática
- 10. Componentes Nativos Win64
- 11. Bibliotecas de Terceiros Compatíveis
- 12. Últimas Melhorias do Compilador
- 13. IDE Delphi Win64: O é o Futuro





## Arquitetura x86: Onde Tudo Começou

- Nasceu em 1978 com o Intel 8086 - o "avô" dos PCs modernos
- Evolução: 8086 → 80286 → 80386 → 80486 → Pentium
- Registradores de 32 bits (EAX, EBX, ECX, EDX...)
- Limite de memória: 4 GB (na prática, 2-3 GB por aplicação)
- Dominou o mercado por mais de 20 anos
- "Se funcionava no x86, funcionava em qualquer lugar!"
- Problema: O mundo cresceu, mas a memória não acompanhou

# Arquitetura x64: A Revolução dos 64 Bits

- AMD64 (2003) - AMD foi pioneira com o Opteron
- Intel 64 (2004) - Intel seguiu com o Xeon e Core
- Criadores: Chuck Moore (AMD) e equipe Intel liderada por Pat Gelsinger
- Motivação: "4 GB não é mais suficiente para nada!"
- Registradores de 64 bits (RAX, RBX, RCX, RDX...)
- 16 registradores vs 8 do x86
- Limite teórico: 18 Exabytes de memória
- Compatibilidade com x86: "O melhor dos dois mundos"
- Resultado: Performance e capacidade sem precedentes

# Delphi XE2: O Marco do Win64

- Lançado em setembro de 2011 - "Finally, 64 bits!"
- Primeiro Delphi com compilador nativo Win64
- Equipe liderada por Allen Bauer e Marco Cantù
- Slogan: "Compile once, run everywhere"
- Suporte simultâneo: Win32, Win64, macOS
- FireMonkey (FMX) estreou junto
- "De 32 para 64 bits em um clique!"
- Revolução: Mesmo código, múltiplas plataformas
- Desenvolvedores: "Isso é mágica ou tecnologia?"
- Resposta: "É Delphi!" 😊





# VCL Win64: Performance Nativa Turbinada

- VCL = Visual Component Library (a "veterana confiável")
- Performance nativa do Windows - "Rápido como um raio!"
- Acesso direto às APIs Win64
- Migração simples: Recompile e pronto!
- Suporte a mais de 4 GB de RAM por aplicação
- Componentes otimizados para 64 bits
- Estabilidade comprovada em produção
- "Se você quer velocidade, VCL é a resposta"
- Ideal para: Aplicações desktop Windows críticas
- Motto: "Nasceu no Windows, vive no Windows!"
- Resultado: Aplicações que "voam" no Win64

# FMX Win64: O Camaleão Multiplataforma

- FMX = FireMonkey (o "jovem ambicioso")
- Um código → Windows, macOS, iOS, Android, Linux
- Interface moderna com GPU acceleration
- Animações suaves e efeitos visuais
- Componentes responsivos e adaptativos
- "Escreva uma vez, rode em qualquer lugar!"
- Ideal para: Apps modernos e multiplataforma
- Performance excelente em Win64
- Suporte a 4K, multi-monitor, touch
- "O futuro é multiplataforma, e o futuro é agora!"
- Resultado: Aplicações bonitas que impressionam



# VCL vs FMX: O Duelo dos Titãs

## VCL Win64:

- Performance máxima no Windows
- Componentes nativos e maduros
- Migração fácil de projetos legados
- Acesso direto às APIs do Windows
- "O Flash da velocidade!"

## FMX Win64:

- Multiplataforma (Windows + outros)
- Interface moderna e atraente
- GPU acceleration e animações
- Componentes responsivos
- "O Homem de Ferro da versatilidade!"

## Veredito: Depende da missão!

- Só Windows + Performance máxima = VCL
- Multiplataforma + Interface moderna = FMX
- "Por que não os dois?" 😊





# Alocação de Memória: 32 vs 64 bits

Win32 (x86):

- Limite: 4 GB de espaço de endereçamento
- Por processo: ~2 GB (3 GB com /3GB)
- Ponteiros: 4 bytes (32 bits)
- "Memória? Que memória?" 🤔

Win64 (x64):

- Limite teórico: 18 Exabytes
- Limite prático: Depende do hardware/SO
- Windows 10/11: Até 2 TB por processo
- Ponteiros: 8 bytes (64 bits)
- "Memória infinita para aplicações!"

Impacto Real:

- Arrays gigantes: Possíveis em Win64
- Processamento de imagens: Sem limitações
- Bancos em memória: Muito maiores
- "De formiguinha para elefante!" 🐘





# Performance de Processadores: A Diferença Real


x86 (32 bits):

- 8 registradores de propósito geral
- Instruções de 32 bits
- Cache otimizado para 32 bits
- "Trabalhador dedicado, mas limitado"

x64 (64 bits):

- 16 registradores de propósito geral
- Instruções de 64 bits + compatibilidade x86
- Mais cache e pipeline otimizado
- Operações matemáticas mais rápidas
- "Trabalhador turbinado!"

Ganhos Reais:

- Cálculos intensivos: 20-40% mais rápido
- Operações com ponto flutuante: Significativo
- Acesso a dados grandes: Muito superior
- "De Fusca para Ferrari!" 

# Comparação de Cálculos

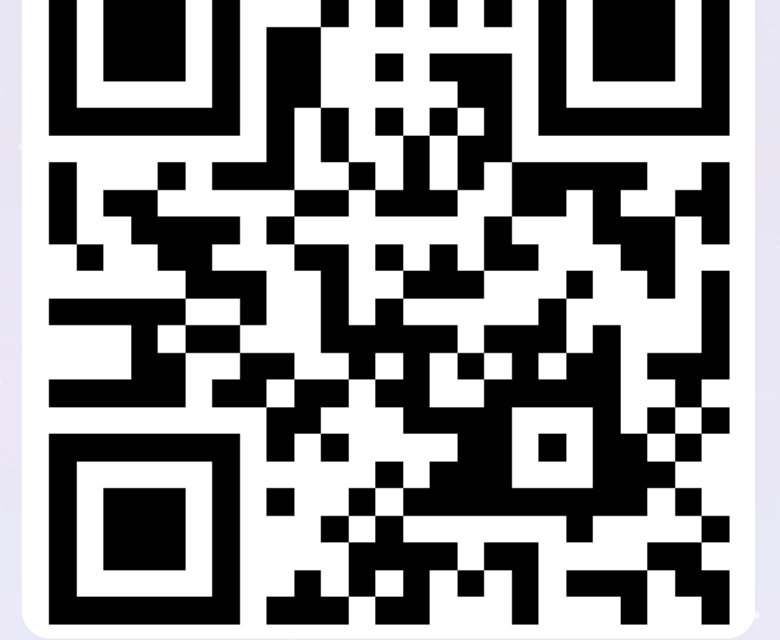
Teste de Performance - Cálculos Intensivos:

```
program PerformanceTest;
{$APPTYPE CONSOLE}
uses System.SysUtils, System.Diagnostics;

procedure TesteMatematicoIntensivo;
var
    i, j: Integer;
    resultado: Double;
    stopwatch: TStopwatch;
begin
    resultado := 0;
    stopwatch := TStopwatch.StartNew;

    for i := 1 to 10000000 do
        for j := 1 to 10 do
            resultado := resultado + Sqrt(i * j) + Sin(i) + Cos(j);
        end;
    end;

    stopwatch.Stop;
    WriteLn('Tempo: ', stopwatch.ElapsedMilliseconds, ' ms');
    WriteLn('Resultado: ', resultado:0:2);
end;
```





# Comparação de Cálculos

Teste de Alocação de Memória:

```
procedure TesteMemoriaGigante;
var
  lista: TArray<Double>;
  i: Integer;
  stopwatch: TStopwatch;
  tamanho: Integer;
begin
  {$IFDEF WIN64}
  tamanho := 1000000000; // 100 milhões em Win64
  {$ELSE}
  tamanho := 100000000; // 10 milhões em Win32
  {$ENDIF}

  stopwatch := TStopwatch.StartNew;
  SetLength(lista, tamanho);

  for i := 0 to High(lista) do
    lista[i] := Sqrt(i) * Pi;

  stopwatch.Stop;
  WriteLn('Alocou ', tamanho, ' elementos em ',
        stopwatch.ElapsedMilliseconds, ' ms');
end;
```

# Comparação de Cálculos

Resultados Típicos dos Testes:

Cálculos Matemáticos Intensivos:

- Win32: ~8.500 ms
- Win64: ~6.200 ms
- Ganho: ~27% mais rápido! 🚀

Alocação de Memória:

- Win32: 10 milhões → ~450 ms
- Win64: 100 milhões → ~850 ms
- Win64 processa 10x mais dados em 2x o tempo!

Uso de NativeInt vs Integer:

var

ponteiro: Pointer;

endereco32: Integer; // ❌ Problema em Win64!

endereco64: NativeInt; // ✅ Correto para ambos!

begin

endereco64 := NativeInt(ponteiro); // Funciona sempre!

end;

"Win64: Onde a matemática fica mais rápida!" 📊



# Componentes Nativos: Funcionam de Primeira!

Componentes VCL/FMX que funcionam sem ajustes:

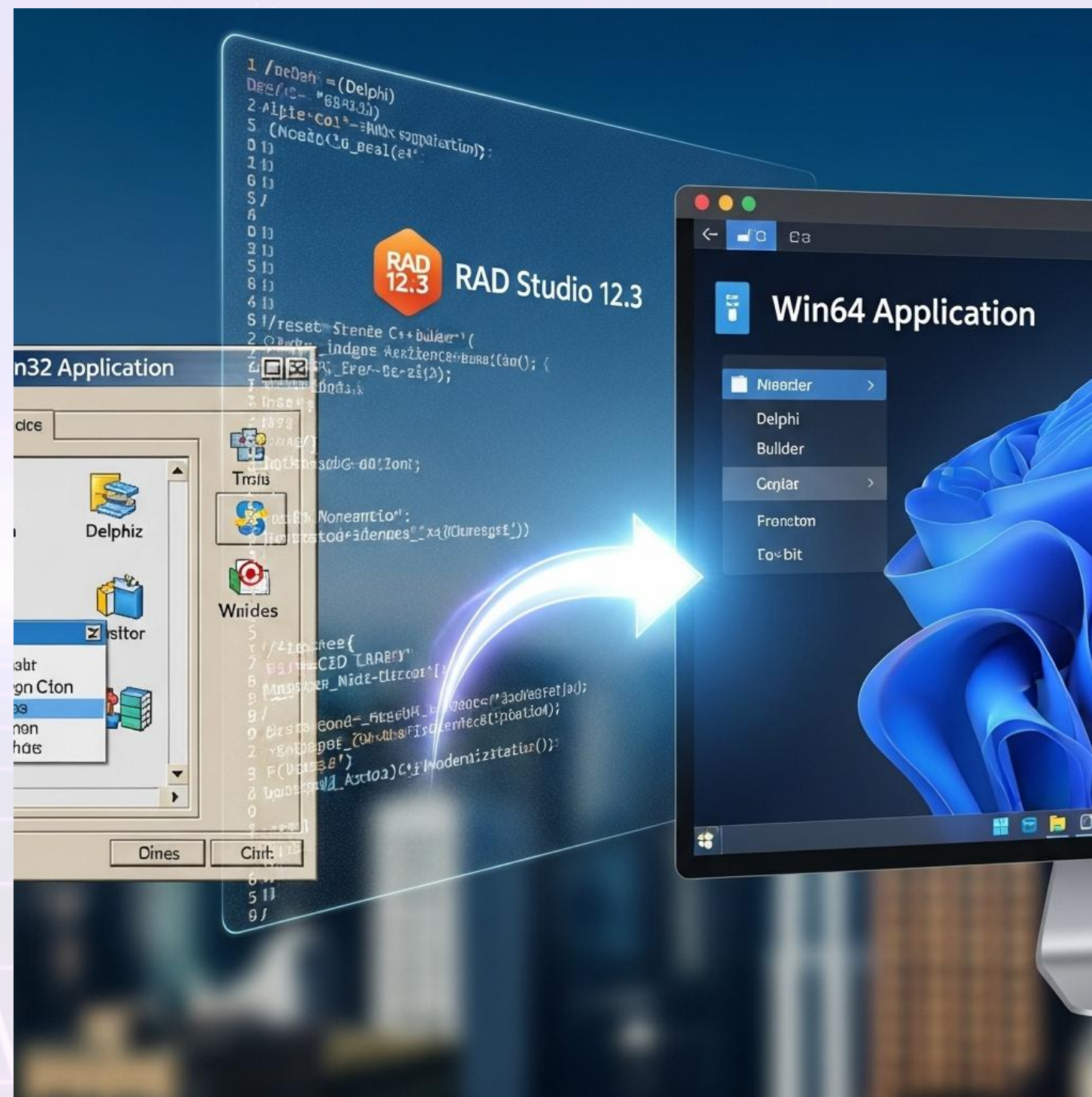
VCL Win64 - "Plug and Play":

- TButton, TEdit, TLabel, TMemo
- TListBox, TComboBox, TCheckBox
- TPanel, TGroupBox, TScrollBar
- TMainMenu, TPopupMenu, TToolBar
- TDataSource, TClientDataSet
- TTimer, TImageList, TStatusBar
- "99% dos componentes VCL funcionam!"

FMX Win64 - "Multiplataforma Nativo":

- TButton, TEdit, TLabel, TMemo
- TListBox, TComboBox, TCheckBox
- TLayout, TPanel, TScrollBar
- TMainMenu, TPopupMenu, TToolBar
- TBindingsList, TPrototypeBindSource
- TTimer, TImageList, TStatusBar

"Se compila, funciona!" 🤓



# Ajustes Necessários: Os Pequenos Detalhes

Situações que precisam de atenção:

Ponteiros e Handles:

// ❌ Problemático

var Handle: Integer;

Handle := Integer(GetWindowHandle);

// ✅ Correto

var Handle: NativeInt;

Handle := NativeInt(GetWindowHandle);

DLLs Externas:

- DLLs de 32 bits NÃO funcionam em Win64
- Precisa da versão 64 bits da DLL
- Exemplo: sqlite3.dll, mysql.dll, etc.

Assembly Inline:

- Não suportado em Win64
- Migrar para Pascal puro ou DLL externa

"Pequenos ajustes, grandes resultados!" 🔧



# Bibliotecas de Terceiros: O Time dos Campeões

Bibliotecas compatíveis com Win64:

ACBr (Automação Comercial Brasil):

- NFe, NFCe, SAT, TEF, ECF
- Totalmente compatível com Win64
- "O canivete suíço do comércio brasileiro!"

JEDI (Joint Endeavour of Delphi Innovators):

- JCL (JEDI Code Library)
- JVCL (JEDI Visual Component Library)
- Suporte completo Win64

DevExpress:

- Componentes visuais premium
- Grids, editores, relatórios
- Performance otimizada para Win64

FastReport:

- Gerador de relatórios profissional
- Versão Win64 nativa
- "Relatórios que impressionam!"

TMS Software:

- Componentes diversos e modernos
- Suporte completo Win64
- "Qualidade premium garantida!"

"A comunidade Delphi não abandona ninguém!" 🤝



# Últimas Melhorias: Compilador Sempre Evoluindo

Melhorias recentes no compilador Win64:

Delphi 11 Alexandria (2021):

- Otimizações de performance significativas
- Melhor geração de código para x64
- Suporte a instruções AVX2
- "Mais rápido que café expresso!"

Delphi 12 Athens (2023):

- Compilação mais rápida (até 30%)
- Melhor otimização de loops
- Suporte aprimorado para multi-threading
- Correções de bugs críticos

Novidades Contínuas:

- Melhor integração com Windows 11
- Otimizações para processadores modernos
- Suporte a novas APIs do Windows
- "Sempre melhorando, nunca parando!"

Resultado:

- Aplicações mais rápidas
- Compilação mais eficiente
- Menos bugs, mais estabilidade
- "O compilador que não para de evoluir!" 🚀



# IDE Delphi Win64: O Futuro já Chegou!

Projeto em desenvolvimento pela Embarcadero:

Status Atual:

- IDE ainda é Win32
- Projeto de migração para Win64 em andamento
- Testes já ocorrendo
- "Paciência, o melhor está por vir!"

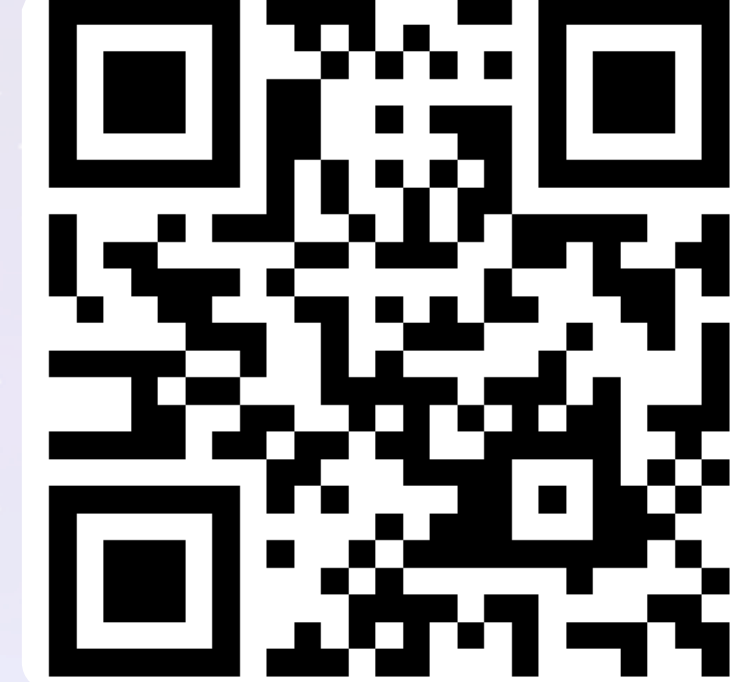
Benefícios Esperados:

- Suporte a mais memória para projetos gigantes
- Performance melhorada da IDE
- Estabilidade aprimorada
- Melhor integração com Windows moderno
- "IDE turbinada para desenvolvedores exigentes!"

Enquanto isso:

- IDE Win32 atual é super estável
- Compila perfeitamente para Win64
- "Se não está quebrado, não mexe!" 😊

"O futuro da IDE será 64 bits de pura potência!" 💪





# Conclusão: Delphi Win64 é o Futuro!

Por que migrar para Win64 com Delphi:

Performance:

- Cálculos 20-40% mais rápidos
- Acesso a memória ilimitada (praticamente)
- Melhor aproveitamento do hardware moderno
- "Velocidade que impressiona!"

Facilidade:

- Migração simples: recompila e pronto!
- Componentes nativos funcionam
- Bibliotecas de terceiros disponíveis
- "Fácil como trocar de canal!"

Futuro:

- Windows é 64 bits por padrão
- Aplicações modernas exigem Win64
- Suporte contínuo da Embarcadero
- "Quem não migra, fica para trás!"

Delphi Win64: Onde performance encontra simplicidade!

"Do Win32 ao Win64: Uma jornada de sucesso!" 🎉

Obrigado! Dúvidas? 🤔

embarcadero®

# EMBARCADERO CONFERENCE

A JORNADA DO HERÓI | 30 ANOS DE INOVAÇÃO



Quer me ver na

#ECC

Acesse o QRCode pelo App  
e avalie minha palestra!

Estarei do mesmo jeito!



**Juliomar Marchetti**

📷 @juliomarmarchetti

📧 juliomarmarchetti

✉️ juliomarmarchetti@gmail.com

☎️ (49) 9 8426-8589

<https://www.juliomarmarchetti.com>