

Embarcadero Delphi e o Desenvolvimento de Aplicações Windows



An Idera, Inc. Company

Por David Intersimone "David I"

Tabela de Conteúdo

1	Introdução	6
2	Desenvolvimento para Windows Atualmente	8
3	Por que Delphi?	10
	3.1 Antes do Delphi	10
	3.2 Delphi 1 e 2	11
	3.3 Delphi 3 até 10	12
	3.4 Delphi Continua a Evoluir	13
4	Tipos de Projetos e Aplicações para Windows	16
5	Comparação de Ferramentas para Windows	18
	5.1 Embarcadero Delphi	18
	5.2 Microsoft Visual Studio	19
	5.3 JetBrains IntelliJ IDEA	19
	5.4 Eclipse IDE	19
	5.5 Microsoft Xamarin	20
	5.6 Visual Studio Code	20
	5.7 Matriz de Comparações das Ferramentas	20
6	Principais Recursos e Benefícios do Delphi para Windows	24
	6.1 FireDAC	25
	6.2 Visual Live Bindings	27
	6.3 Ferramentas Visuais Two Way	28
	6.4 Estilos e Temas	29
	6.5 Windows API e WinRT Units	31
	6.6 Biblioteca de Programação Paralela (PPL)	31
	6.7 Componentes Windows 10, High DPI e suporte Multi-Monitor	32
	6.7.1 Componentes para Windows 10	32
	6.7.2 Suporte High DPI no Windows 10	33
	6.7.3 Suporte a vários Monitores para VCL e FMX	33
	6.8 RAD Server	34
	6.9 App Tethering	35
	6.10 Bluetooth, Beacons, Beacon Fencing e IoT	36
	6.11 InterBase	40
	6.1 Enterprise Connectors	40

7	Produtividade e Performance no Delphi	40
	7.1 Produtividade	42
	7.2 Performance	44
8	Recursos Adicionais, Benefícios e Recursos	46
	8.1 Multiplataforma	46
	8.2 Virtualização e Distribuição de Aplicativos Delphi	46
	8.2.1 Cybele Software Thinfinity® VirtualUI	46
	8.2.2 Amazon AppStream 2.0	47
	8.2.3 O Gerenciador de Implantação da IDE	47
	8.2.4 Ferramentas de Instalação de Terceiros	48
	8.2.5 Submetendo para uma Loja Online	48
	8.3 Ecossistema de Parceiros de Tecnologia	49
	8.4 Educação – Aprenda, Receba Ajuda, Encontre Exemplos	51
	8.4.1 LearnDelphi.org	51
	8.4.2 Embarcadero Academy	51
	8.4.3 Developer Skill Sprints	52
	8.4.4 Webinar Replays e outros Vídeos no YouTube	52
	8.4.5 How-To Videos	52
	8.4.6 Embarcadero Developers Blog	52
	8.4.7 Embarcadero nas Redes Sociais	52
	8.4.8 Feito com Delphi	53
	8.4.9 Exemplos de Código, Tutoriais e Exemplos	53
	8.4.10 RAD Studio IDE Tools API	53
	8.4.11 GetIt Package Manager	54
	8.4.12 GitHub Repositórios para Delphi	54
	8.4.13 Respostas no StackOverflow	54
	8.4.14 Página de Status de Plataforma	54
9	Vídeos sobre Windows 10 e VCL	56
	9.1 Modernização de Aplicativos para Windows 10	56
	9.2 Delphi IDE Características e Produtividade	57

Delphi 1 - 15 de Fevereiro de 1995	59
Delphi 2 - 10 de Fevereiro de 1996	59
Delphi 3 - 5 de Agosto de 1997	60
Delphi 4 - 17 de Junho de 1998	60
Delphi 5 - 10 de Agosto de 1999	60
Delphi 6 - 21 de Maio de 2001	60
Delphi 7 - 9 de Agosto de 2002	61
Delphi 8 - 22 de Dezembro de 2003	61
Delphi 2005 - 12 de Outubro de 2004	61
Delphi 2006 - 23 de Novembro de 2005	61
Delphi 2007 - 16 de Março de 2007	62
Delphi 2009 - 25 de Agosto de 2008	63
Delphi 2010 - 15 de Agosto de 2009	63
Delphi XE - 30 de Agosto de 2010	63
Delphi XE2 - 2 de Setembro de 2011	64
Delphi XE3 - 3 de Setembro de 2012	64
Delphi XE4 - 22 de Abril de 2013	64
Delphi XE5 - 11 de Setembro de 2013	65
Delphi XE6 - 15 de Abril de 2014	65
Delphi XE7 - 2 de Setembro de 2014	65
Delphi XE8 - 7 de Abril de 2015	66
Delphi 10 Seattle - 31 de Agosto de 2015	66
Delphi 10.1 Berlim - 20 de Abril de 2016	67
Delphi 10.2 Tóquio - 22 de Março de 2017	67
Delphi 10.3 Rio - 21 de Novembro de 2018	67
Delphi 10.3.1 Rio - 14 de Fevereiro de 2019	68
Delphi 10.3.2 Rio - 18 de Julho de 2019	68
Delphi 10.3.3 Rio - 21 de Novembro de 2019	68



Introdução

1 Introdução

O Microsoft Windows ainda é o sistema operacional líder em computação desktop, de acordo com vários relatórios de analistas do setor e muitas empresas de rastreamento de uso de computador. O relatório do [StatCounter.com](https://www.statcounter.com) de janeiro de 2020 mostra que o Microsoft Windows tem uma participação de mercado de 77,7% no sistema operacional desktop. Desde a introdução do Windows, milhões de aplicativos foram construídos por desenvolvedores usando diferentes ferramentas de desenvolvimento e linguagens de programação. Este white paper é para desenvolvedores experientes com qualquer linguagem ou ferramenta para o desenvolvimento Windows, incluindo versões anteriores e atuais do Delphi, bem como desenvolvedores novos para o desenvolvimento Windows.



Desenvolvimento Windows Atualmente

2 Desenvolvimento para Windows Atualmente

Vi pela primeira vez uma prévia do Windows 1 em uma reunião com Bill Gates quando estava trabalhando na Softsel Computer Products (agora Merisel). A demo inicial mostrava aplicativos abrindo em áreas não utilizadas da tela e parecia lado a lado (versus janelas sobrepostas no Xerox Star, Apple Lisa e Macintosh). Eu questionei a abordagem lado a lado ao gerente de produto e Bill Gates retrucou de volta para mim e me disse que cada aplicativo estava sendo executado em uma janela!

Com o lançamento do Windows 1 e 2, os desenvolvedores ainda estavam construindo aplicativos para DOS e consoles de texto. A IBM estava trabalhando com a Microsoft para criar uma substituição de sistema operacional baseada em GUI para o Dos chamado OS/2. Como o OS/2 estava sendo lançado, a Microsoft continuou o trabalho no Windows e lançou o Windows 3.0 que, em minha opinião, foi a primeira versão "utilizável" do Windows para desenvolvedores. O desenvolvimento inicial foi feito usando C, Pascal, Basic e outros para criar aplicativos que chamavam diretamente a API do Windows. Alguns desenvolvedores se lembrarão das 100 linhas de código C necessárias para criar um aplicativo simples do Windows.



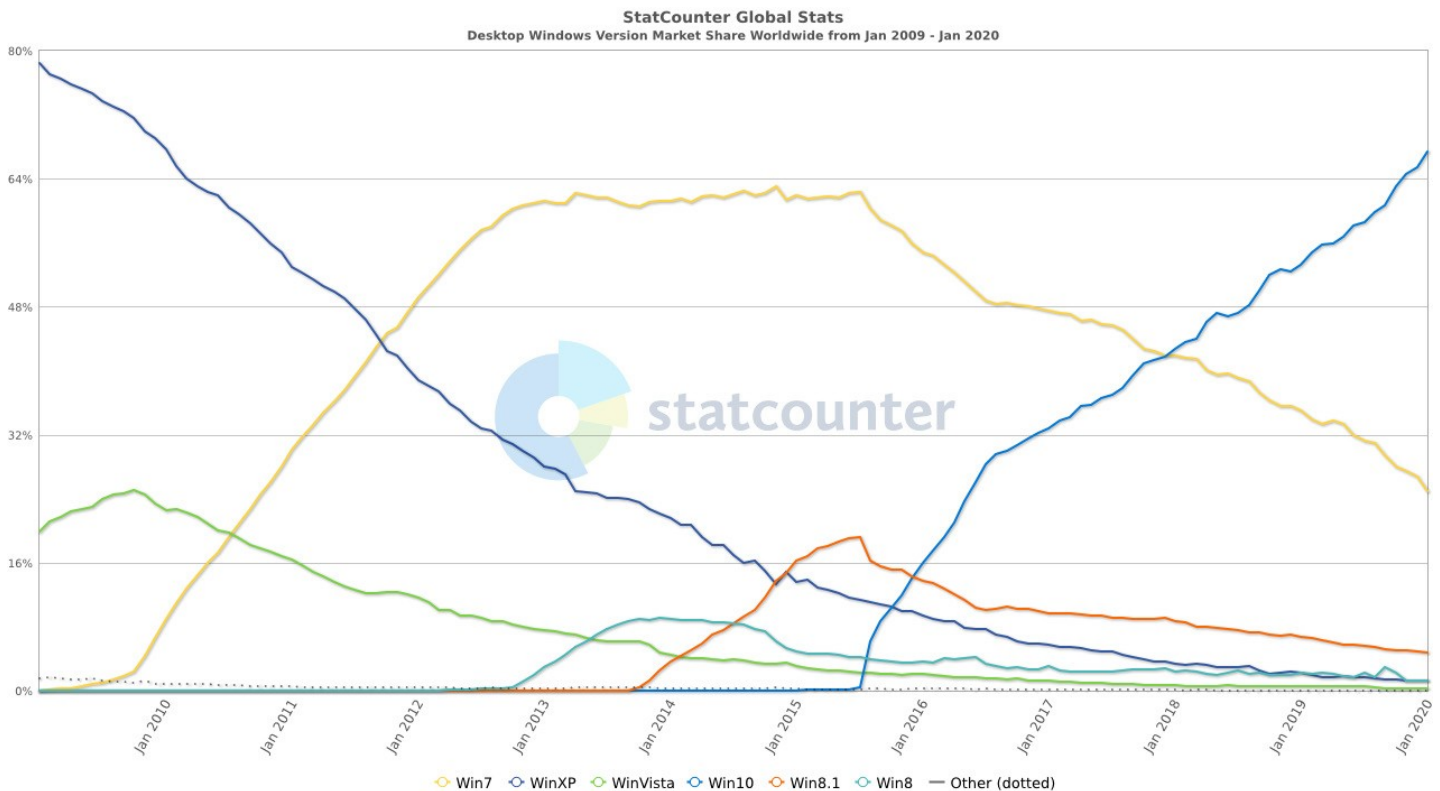
Com o advento de linguagens orientadas a objetos como C++, Object Pascal, Actor e outros, a próxima onda de desenvolvimento de aplicativos do Windows começou a tornar a programação do Windows mais simples. Enquanto a maioria dos desenvolvedores ainda usava diretamente a API do Windows, surgiram bibliotecas de classe e componentes que encapsulavam a API fornecendo uma camada de abstração mais alta e suporte para múltiplas versões do Windows.

À medida que o Windows se tornou mais popular, os desenvolvedores tiveram que lutar com o desafio de suportar versões distintas do Windows em seus aplicativos. Com o passar do tempo, a Microsoft começou a encerrar oficialmente o suporte para versões mais antigas do Windows. Mais recentemente, a Microsoft "oficialmente" encerrou o suporte para o Windows 7 em 14 de janeiro de 2020. Ainda assim, os desenvolvedores têm que suportar vários releases populares do Windows, incluindo Windows 7, 8.x e 10.

O StatCounter.com possui um [gráfico que mostra a participação de mercado](#) da versão para desktop do Windows em todo o mundo e como ela mudou nos últimos onze anos (janeiro de 2009 a janeiro de 2020). Este gráfico mostra que o Windows 7 e o Windows 10 ainda são os dois principais lançamentos mais populares em uso.



Por que Delphi?



3 Por que Delphi?

Os desenvolvedores têm uma gama de opções para plataformas, ferramentas de desenvolvimento, linguagens de programação, bibliotecas, arquiteturas, etc. que usam para construir aplicativos. Por que um desenvolvedor deve usar Delphi? Que tipos de aplicativos eles podem construir? Será que eles serão capazes de atender aos requisitos e cronogramas de um usuário e cliente exigentes? Vamos dar uma olhada rápida em algumas das inovações e características-chave que mostram como os desenvolvedores podem ter sucesso em responder às necessidades dos negócios, às demandas dos clientes e permanecer na vanguarda da engenharia de software e das tecnologias de ponta usando Delphi.

3.1 Antes do Delphi

Antes do Delphi ser lançado, as linguagens de programação orientadas a objetos e objetos eram usadas para criar uma ampla gama de aplicativos Windows. Na Embarcadero, ferramentas e bibliotecas de desenvolvimento forneceram a seguinte migração de suporte para o desenvolvimento de aplicativos DOS e Windows, passando do uso direto de APIs do sistema operacional para bibliotecas de classes:

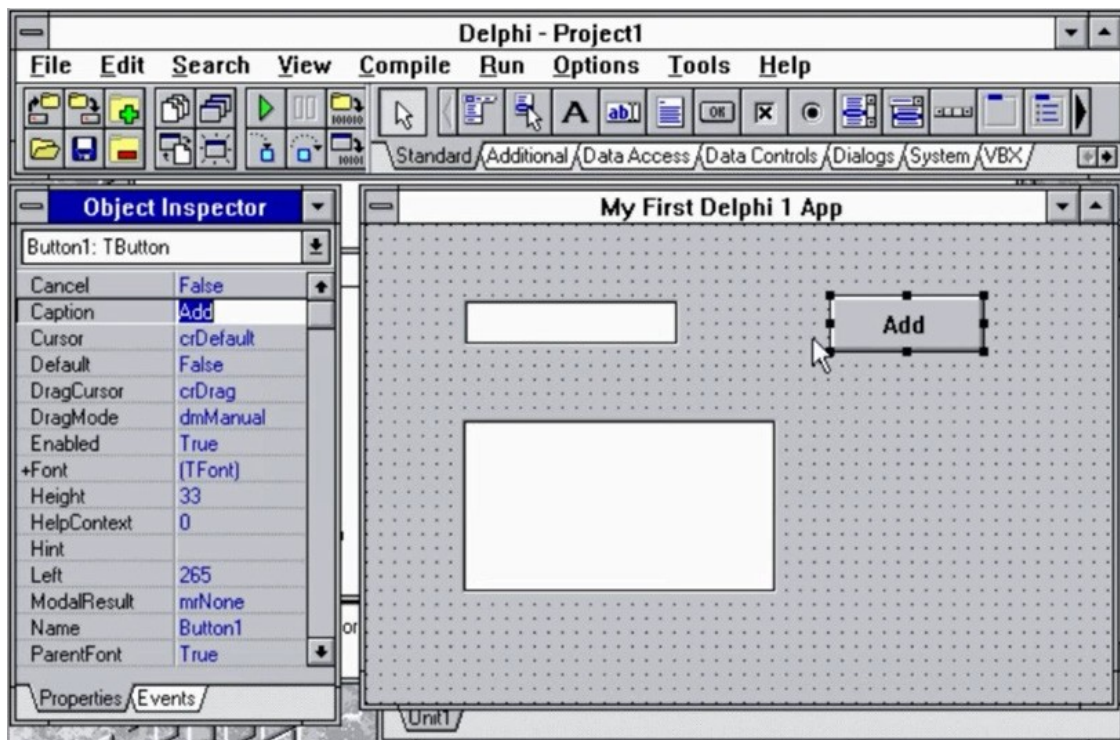
- Turbo Pascal para DOS com Turbo Vision
- Turbo Pascal para Windows com Object Windows Library (OWL)

Enquanto essas soluções de desenvolvimento aumentavam a produtividade para os desenvolvedores, uma próxima geração de tecnologia e implementação era necessária para acelerar o sucesso do Microsoft Windows. A solução foi combinar programação visual, tecnologia de compilador e programação de banco de dados em uma experiência de desenvolvimento perfeita para avançar rapidamente no estado da arte para o desenvolvimento de aplicativos Windows.

3.2 Delphi 1 e 2

Em 14 de fevereiro de 1995 na Conferência de Desenvolvimento de Software no *Moscone Convention Center* em São Francisco, Delphi 1 foi revelado para uma audiência de mais de 2.000 desenvolvedores e especialistas da indústria de PCs. Anders Heilsberg, arquiteto-chefe do Delphi, demonstrou naquela noite e recebeu uma ovação de pé dos presentes.

Anders Heilsberg disse: "O Delphi é o primeiro produto que combina programação visual, tecnologia de compilador e programação de banco de dados em uma solução única. A corrente predominante difundida em Delphi é de abertura e extensibilidade. Por exemplo, quando adicionamos a capacidade de ter experts em Delphi, não apenas adicionamos um certo conjunto de experts pré-fabricados, também demos a você uma maneira de instalar seus próprios experts, nós lhe demos uma maneira de instalar seus próprios modelos em vez de apenas incluir alguns padrões. Basicamente, o Delphi foi projetado do zero para ser portátil para 32 bits. Seu código fonte será portado sem alterações, nós já verificamos isso em casa. Estamos constantemente procurando novas e emergentes tecnologias para adotar, por exemplo, controles OLE2, OCX e todas as novas APIs disponíveis no Windows 95 serão incluídas no Delphi. O interessante deste produto é que é divertido de usar. Nós realmente gostamos de usar o produto nós mesmos. Nós construímos o produto em si, o que também significa que ao longo dos dois anos que trabalhamos neste produto, nos deparamos com todos os problemas que você poderia de outra forma encontrar, e nós realmente resolvemos esses problemas. E hoje, é simplesmente incrível a rapidez com que você pode montar um aplicativo com aparência profissional, pressionando alguns controles em um formulário e escrevendo algumas linhas de código e pronto! "



O Delphi 1 foi lançado suportando a API de 16 bits do Windows 3, um compilador de código nativo de 16 bits, a Biblioteca de Componentes Visuais (VCL) que forneceu objetos de código nativo para interfaces de GUI do Windows, depuração integrada, acesso ao banco de dados local e suporte ao banco de dados SQL Client/Server através de um conjunto comum de componentes chamado de Borland Database Engine (BDE).

Utilizando a IDE os desenvolvedores poderiam arrastar e soltar componentes visuais e não visuais em um formulário, definir algumas das propriedades dos componentes usando o inspetor de objetos e selecionar eventos de componentes para escrever código adicional de algoritmos e lógica de negócios. Tudo o que um desenvolvedor poderia fazer no IDE com componentes e propriedades também poderia ser feito usando código. O design de interface do usuário de um aplicativo aparece na definição do formulário visualmente e também no modo texto, e modificando qualquer visualização você pode evoluir o aplicativo. No IDE, os desenvolvedores poderiam, pela primeira vez, ver dados ao vivo em tempo de projeto para as conexões com bancos de dados locais (Paradox e dBase, por exemplo) e uma grande variedade de bases de dados SQL client/server.

Enquanto o Delphi 1 estava sendo desenvolvido, a Microsoft estava trabalhando no próximo lançamento do Windows. Como mencionado na citação acima, a equipe Delphi estava testando o suporte à tecnologia Windows de 16 e 32 bits ao mesmo tempo para garantir que o código escrito para Windows de 16 bits também pudesse ser compilado para Windows de 32 bits sem necessidade de alterações. Demonstrei uma versão de pré-lançamento do Delphi 2 no lançamento do Windows 95 no gramado do campus da Microsoft.

A VCL continuou a evoluir para suportar versões subsequentes do Windows, preservando os investimentos em arquitetura baseada em componentes e os investimentos em código criados pelo desenvolvedor.

3.3 Delphi 3 ao 10

O Delphi 3 e 4 adicionaram novos recursos de linguagem, desenvolvimento de vários níveis e suporte ao Windows 98. Os recursos de linguagem adicionados incluem interfaces, sobrecarga de métodos, matrizes dinâmicas e pacotes de componentes.

Delphi 5, 6, 7 introduziu funcionalidades de Internet e Web, frameworks adicionais, suporte ao XML, serviços web SOAP, suporte a UML e Windows XP.

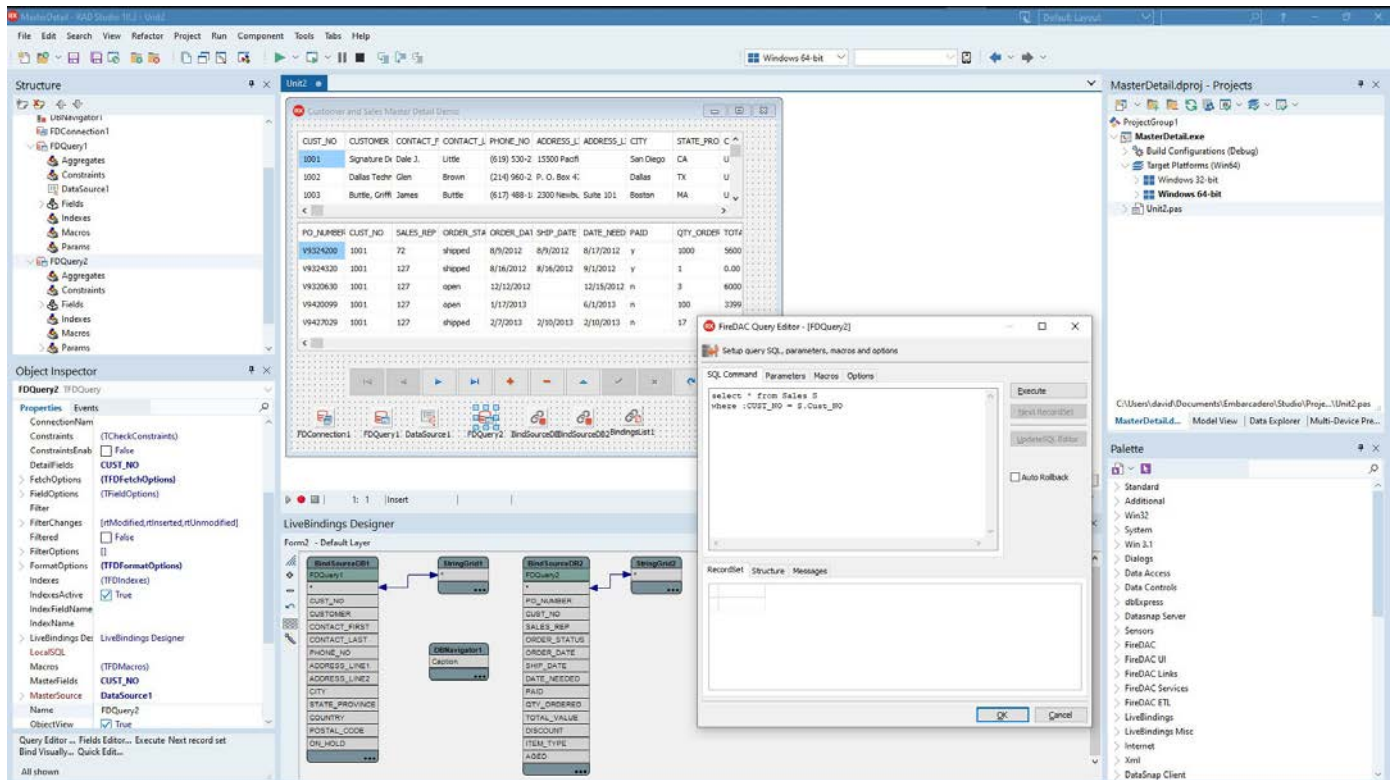
Delphi 2005 a 2010 adicionou inlining de função, for..in, generics, métodos anônimos, atributos, toque/gestos, multi-unit name spaces, sobrecarga do operador, métodos/propriedades estáticas, numerosos aprimoramentos do editor de código, o gerenciador de memória FastMM, suporte ao Unicode e muito mais para Windows.

Delphi XE para XE8 adicionou desenvolvimento multiplataforma (Windows, macOS, iOS e Android) com FireMonkey (FMX), FireDAC, suporte a REST, Live Bindings, Styles, App Tethering, Suporte Cloud/BaaS, Parallel Programming Library (PPL), RAD Server, etc.

Delphi 10.0 a 10.3 adicionou suporte para declaração de variável inline, inferência de tipo, uma ampla gama de novos recursos do Windows 10, suporte ao FireDAC para MongoDB, Bluetooth Classic e LE, Beacons, componentes IoT, estilos Windows 10, novo suporte a plataformas de 64 bits (macOS, iOS, Android, Linux), suporte a Windows High-DPI, aprimoramentos do RAD Server e muito mais.

3.4 Delphi continua a evoluir

A Delphi e o VCL continuaram a evoluir para suportar novos recursos e mudanças no sistema operacional Windows e APIs. O Delphi também aproveitou muitos dos mesmos recursos para plataformas não-Windows adicionais, incluindo iOS, macOS,



Android e Linux. A linguagem Object Pascal da Delphi continuou a avançar junto com o RAD Studio IDE, suporte a banco de dados SQL e no-SQL, novas arquiteturas (web, nuvem, multicamadas e micro serviços), dispositivos Bluetooth, IoT e muito mais.

O Delphi também inclui edições da biblioteca de componentes FireMonkey (FMX) para a construção de aplicativos GUI multiplataforma para Win32, Win64, macOS, iOS, Android e Linux. Todo o código-fonte é fornecido para as bibliotecas VCL, FMX e run-time para ajudar os desenvolvedores a entender como os componentes e bibliotecas são construídos, e também como fonte final de documentação.

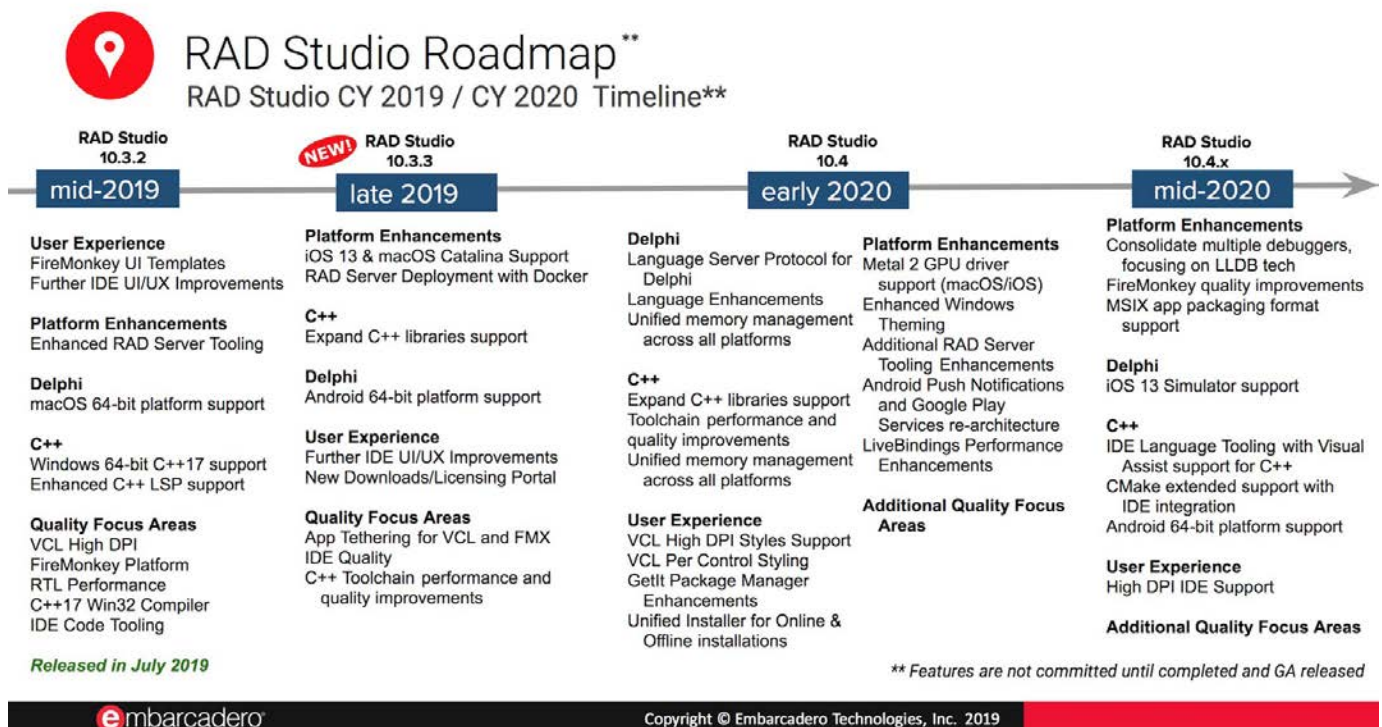
Uma pergunta repetida que os desenvolvedores têm feito ao longo dos anos é: "Qual

biblioteca de componentes devo usar, VCL ou FMX?" A resposta depende de quais aplicativos um desenvolvedor criará hoje para um sistema operacional em particular, e para onde o desenvolvimento está indo no futuro. Se o aplicativo for direcionado apenas ao Windows, a VCL seria a melhor opção. Se um aplicativo precisar oferecer suporte ao Windows, macOS e dispositivos móveis hoje ou no futuro próximo, o FMX será a melhor opção.

Jim McKeeth disse bem em um post em seu [blog](#): "Use o VCL quando você está apenas mirando o Windows e não precisa do 3D, efeitos ou flexibilidade do FireMonkey. Use FireMonkey quando você está indo para multiplataforma, ou você quer usar um pouco da flexibilidade do FireMonkey especialmente quando trabalhar em torno de gráficos. Ambas as estruturas estarão por aí por um longo tempo. À medida que você usa ambos, você terá uma melhor sensação para que usar em cada situação." Se um aplicativo é construído com VCL e precisa ser migrado para FMX, confira a ferramenta da parceira de tecnologia *Mida Converter* que converte aplicativos VCL em FMX.

Uma lista das principais características e inovações adicionadas a cada lançamento do Delphi é incluída no final deste documento. Planos para melhorias futuras estão disponíveis no roteiro público online do [RAD Studio](#).

Updated - August 2019





Tipos de Projetos e Aplicações para Windows


4 Tipos de Projetos e Aplicações para Windows

As ferramentas de desenvolvimento do Windows geralmente fornecem uma variedade de modelos e assistentes que ajudam os desenvolvedores a configurar um projeto inicial de aplicativo para Windows. Os tipos de projetos Windows incluem:

- Aplicações de GUI
- Aplicativos de Console
- Aplicativos de Serviço
- Aplicações do Servidor – Web, SOAP e REST
- Aplicações em Nuvem
- Aplicativos do Painel de Controle
- Bibliotecas Dinâmicas (DLLs)
- Bibliotecas ActiveX
- Pacotes
- Bibliotecas Estáticas
- Testes unitários

Para oferecer suporte ao Windows 10, as ferramentas de desenvolvimento precisam fornecer integração tecnológica com os modelos de aplicativos Windows API, COM, WinRT e APPX.





Comparação de Ferramentas para Desenvolvimento Windows

5 Comparação de Ferramentas para Desenvolvimento Windows

Existem várias ferramentas para a construção de aplicativos Windows. Algumas dessas ferramentas são ambientes completos de desenvolvimento integrados que suportam múltiplas linguagens de programação. Alguns desenvolvedores também usam editores de programação e sistemas de build que trabalham com interfaces de compilação via comando de linha para criar seus aplicativos. O Delphi também fornece compiladores de linha de comando para integração com editores e sistemas de build. Algumas das seguintes ferramentas geram aplicativos de código nativo para Windows. Outros compilam código fonte para um formato intermediário, ou interpretam o código em tempo de execução usando compiladores Just in Time.

A maioria das ferramentas de desenvolvimento suporta depuração integrada, desenvolvimento em equipe, testes automatizados, refatoração, implantação, análise de código, métricas de código, criação de perfil e muito mais. As ferramentas suportam o desenvolvimento para várias plataformas, incluindo Windows, macOS, iOS e Android, e fornecem acesso a APIs nativas em cada plataforma, frameworks criados em cima de APIs e reutilização de parte de todo o código nas plataformas suportadas.

Ao comparar ferramentas multiplataforma, os desenvolvedores vão querer explorar como cada ferramenta faz uma distinção entre a plataforma nativa (geralmente relacionada a interfaces de UI e SO) e o código de máquina nativa (compilado para o processador de cada plataforma).

Aqui está uma lista rápida de algumas das ferramentas de desenvolvedor populares que fornecem suporte para a construção de aplicativos Windows. Observe que algumas das linguagens suportadas, por exemplo JavaScript/Type Script, são mais usadas para criar aplicativos baseados na Web que serão executados em um navegador ou em um container browser em Windows.

5.1 Embarcadero Delphi

O Delphi está disponível separadamente e também como parte da suíte de desenvolvimento RAD Studio que suporta as linguagens de programação Delphi (Object Pascal) e C++ Builder (C++). Os desenvolvedores podem projetar e construir visualmente consoles, GUI desktop, cliente/servidor e vários tipos de projetos para Windows usando os componentes da VCL e bibliotecas de tempo de execução. Tanto o Delphi quanto o C++ Builder também podem usar o mesmo código-fonte para construir em multiplataforma aplicativos compilados nativamente usando os componentes do FireMonkey (FMX) e bibliotecas de tempo de execução para Windows, macOS, iOS e Android. O Delphi também pode ser usado para criar aplicativos de código nativo de console, desktop e servidor com foco em Linux de 64 bits em processadores Intel.

O Delphi está disponível em **quatro edições**:

- Delphi Community (grátis) – Todas as características do Professional, mas licenciados para não profissionais e estudantes
- Delphi Professional – Desenvolvimento mobile, desktop e banco de dados local
- Delphi Enterprise – Banco de dados remoto, cliente/servidor e soluções multicamadas, e também introduz a plataforma Linux
- Delphi Architect – Empacotado com outras ferramentas adequadas para um arquiteto de sistemas

Página inicial do produto – <https://www.embarcadero.com/products/delphi/>

5.2 Microsoft Visual Studio

O Visual Studio auxilia a construção de aplicativos Windows usando as linguagens de programação C++, C#, Visual Basic, JavaScript/Type Script, Python, F#. Um compilador de código nativo é fornecido apenas para a linguagem de programação C++. O foco principal do Visual Studio é a construção de aplicativos de console, desktop e servidor usando APIs e frameworks do Windows, incluindo o MFC e o WinForms. A linguagem C# também suporta a construção de aplicativos multiplataforma para Windows, iOS e Android usando Formulários .NET e WinForms/Xamarin com interfaces para APIs nativas da plataforma.

As edições do Visual Studio (<https://visualstudio.microsoft.com/vs/compare/>) incluem:

- Visual Studio Community (grátis)
- Visual Studio Professional
- Visual Studio Enterprise

Página inicial do produto – <https://visualstudio.microsoft.com/>

5.3 JetBrains IntelliJ IDEA

O IntelliJ IDEA suporta a construção de aplicativos usando Java, Kotlin, JavaScript/Type Script, Scala, Groovy, C/C++ (<https://www.jetbrains.com/cpp/>) e outras linguagens de programação para a construção de aplicativos desktop, servidor, web e multiplataforma. IntelliJ IDEA é principalmente um IDE para desenvolvimento Java. O IDE também sabe sobre a sintaxe de línguas adicionais. Java, Kotlin, Scala e Groovy criam aplicativos que usam JVMs e que estão disponíveis em multiplataforma. O IDE implementa a mesma codificação Android e conjunto de ferramentas integradas que está disponível no Android Studio do Google.

Edições do IntelliJ IDEA (https://www.jetbrains.com/idea/features/editions_comparison_matrix.html):

- IntelliJ IDEA Community Edition – para desenvolvimento JVM e Android
- IntelliJ IDEA Ultimate – para desenvolvimento web e

corporativos

- Página inicial do produto – <https://www.jetbrains.com/idea/>

5.4 Eclipse IDE

Este ambiente de desenvolvimento integrado de código aberto da Fundação Eclipse suporta várias linguagens de programação, incluindo Java, C++, FORTRAN e JavaScript. O Eclipse IDE possui uma API aberta para permitir integrações de ferramentas de desenvolvimento, ferramentas de build, bibliotecas, etc. Desde o início, o uso principal do Eclipse é a construção de aplicativos Java para aplicativos cliente e do servidor onde houver um runtime JVM e Java. As integrações de linguagens C++ e FORTRAN melhoraram ao longo do tempo fornecendo compiladores e depuradores de código para o desenvolvimento de aplicativos.

Página inicial do produto – <https://www.eclipse.org/eclipseide/>

5.5 Microsoft Xamarin

Xamarin é uma ferramenta de desenvolvimento multiplataforma de código aberto usando C# e o código aberto da plataforma .NET. O Xamarin se baseia na plataforma de desenvolvimento .NET, fornecendo ferramentas e bibliotecas para a construção de aplicativos Windows, Android, iOS, macOS, tvOS e watchOS. Para criar, desenvolvedores de aplicativos multiplataforma podem usar o programa/lógica de negócios em C# compartilhado, código de interface de usuário em C# compartilhado e código C# específico da plataforma. A Microsoft/Xamarin afirma que pelo menos 75% do código C# pode ser compartilhado entre plataformas. Os aplicativos Xamarin.Forms GUI podem ser construídos para iOS, Android e Windows 10 Universal Platform.

Página inicial do produto <https://dotnet.microsoft.com/apps/xamarin>

5.6 Visual Studio Code

Visual Studio Code é um ambiente de programadores de código aberto que suporta C++, C#, Python e outras linguagens. O Visual Studio Code fornece um IDE multiplataforma que pode ser executado no Windows, macOS e Linux. O desenvolvimento é focado no editor de programas, ambiente de build e depuração. Os colaboradores de código aberto também podem criar designers visuais para linguagens e plataformas de programação específicas.

Página inicial do produto – <https://code.visualstudio.com/>

5.7 Matriz de comparação das ferramentas

Esta tabela resume e compara os recursos das ferramentas de desenvolvimento Windows listadas acima. Os desenvolvedores devem verificar a matriz de recursos de cada produto para obter detalhes adicionais e informações de releases mais recentes.

Feature/ Tool	RAD Studio	Visual Studio	IntelliJ IDEA	Eclipse	Xamarin	VS Code
Editions	Architect, Enterprise, Professional, Community (Free)	Enterprise, Professional, Community (Free)	Ultimate, Community (Free)	Free (open source)	Free (open source)	Free (open source)
Development Environment	IDE	IDE	IDE	IDE	IDE	Source Code Editor
Windows Project Targets	Win32, Win64	Win32, Win64	JVM	JVM	.NET	Various
Languages	Object Pascal, C++	C++, C#, VB, TypeScript/ JavaScript, Python, F#	Java, Kotlin, JavaScript/ TypeScript, Scala, Groovy, C/C++, Ruby, etc.	Java, C++, Fortran, and JavaScript, etc.	C#	C++, C#, Python, Java, Go, Rust, VB,

Feature/ Tool	RAD Studio	Visual Studio	IntelliJ IDEA	Eclipse	Xamarin	VS Code
Native Machine Code Compilers	Y (Object Pascal and C++)	Y (C++)	Y (C++)	Y (C++ and Fortran)	.NET Byte Code	Y (C++)
Windows UI Framework and Components	Y	C++ (MFC)	N	N	N	N
Cross platform UI Framework and Components	Y	C# (Xamarin Forms)	N	N	N	N
Integrated Visual UI Designer	Y	N	N	Window Builder Pro plug-in	Y	IDE API for designers
Components for desktop. SQL and NoSQL databases	Y	N	N	N	N	N
Build Web Server Applications	IIS and Apache	IIS, Azure App Service	Apache Tomcat, WebLogic, WebSphere	Apache Tomcat	IIS, ASP.NET	Plug-ins for Apache and Jetty
Cloud Integration	Azure, AWS	Azure	Cloud Foundry, Google App Engine, OpenShift, AWS	Cloud Foundry, AWS, Apache	Azure	Azure
Component and Run Time Library Source Code	Y	Y	Y	Y	.NET Core	Y
Team Development	Subversion, Mercurial, Git	Subversion, Mercurial, Git, VS Team Services (VSTS)	Subversion, Mercurial, Git, Perforce, Team Foundation Server (plug-in)	Subversion, Mercurial, Git	Subversion, Mercurial, Git	Subversion, Mercurial, Git
Static Code Analysis and Code Metrics	Y	Y	Y	Y	Plug-ins	Plug-ins
Integrated Unit Testing	Y	Y	Y	Y	Y	Plug-ins

Feature/ Tool	RAD Studio	Visual Studio	IntelliJ IDEA	Eclipse	Xamarin	VS Code
Component and Run Time Library Source Code	Y	Y	Y	Y	.NET Core	Y
Team Development	Subversion, Mercurial, Git	Subversion, Mercurial, Git, VS Team Services (VSTS)	Subversion, Mercurial, Git, Perforce, Team Foundation Server (plug-in)	Subversion, Mercurial, Git	Subversion, Mercurial, Git	Subversion, Mercurial, Git
Static Code Analysis and Code Metrics	Y	Y	Y	Y	Plug-ins	Plug-ins
Integrated Unit Testing	Y	Y	Y	Y	Y	Plug-ins
Component and Run Time Library Source Code	Y	Y	Y	Y	.NET Core	Y
Team Development	Subversion, Mercurial, Git	Subversion, Mercurial, Git, VS Team Services (VSTS)	Subversion, Mercurial, Git, Perforce, Team Foundation Server (plug-in)	Subversion, Mercurial, Git	Subversion, Mercurial, Git	Subversion, Mercurial, Git
Static Code Analysis and Code Metrics	Y	Y	Y	Y	Plug-ins	Plug-ins
Integrated Unit Testing	Y	Y	Y	Y	Y	Plug-ins
Localized IDE	English, French, German, Japanese	English, French, German, Chinese, Japanese, Czech, Italian, Korean, Polish, Portuguese (Brazil), Russian, Spanish	N	N	N	Plug-in language packs
Turnkey Microservices Framework	RAD Server	Azure Service Fabric, .NET Microservices	Micronaut, Quarkus, and Helidon	Plug-ins	.NET Microservices	N

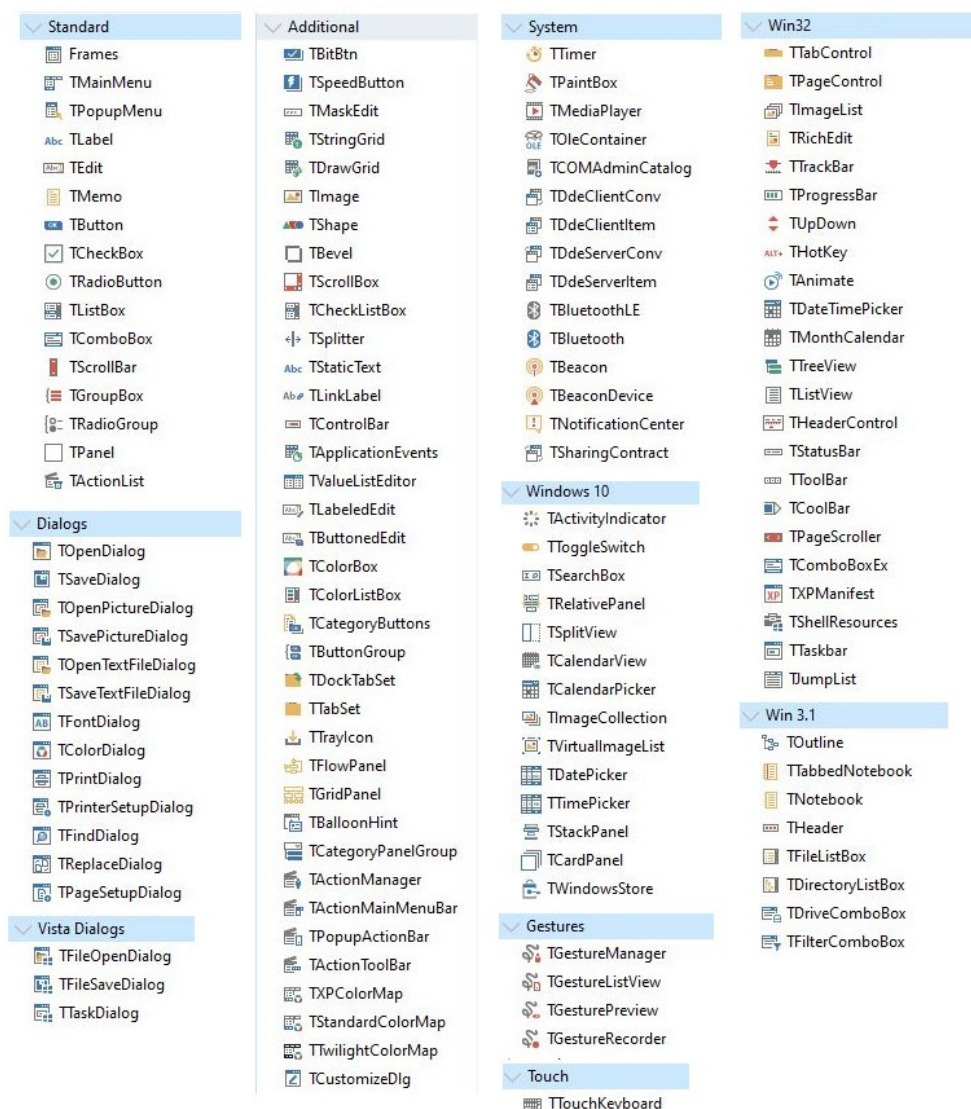


Principais Características e Benefícios do Delphi para Desenvolvedores Windows

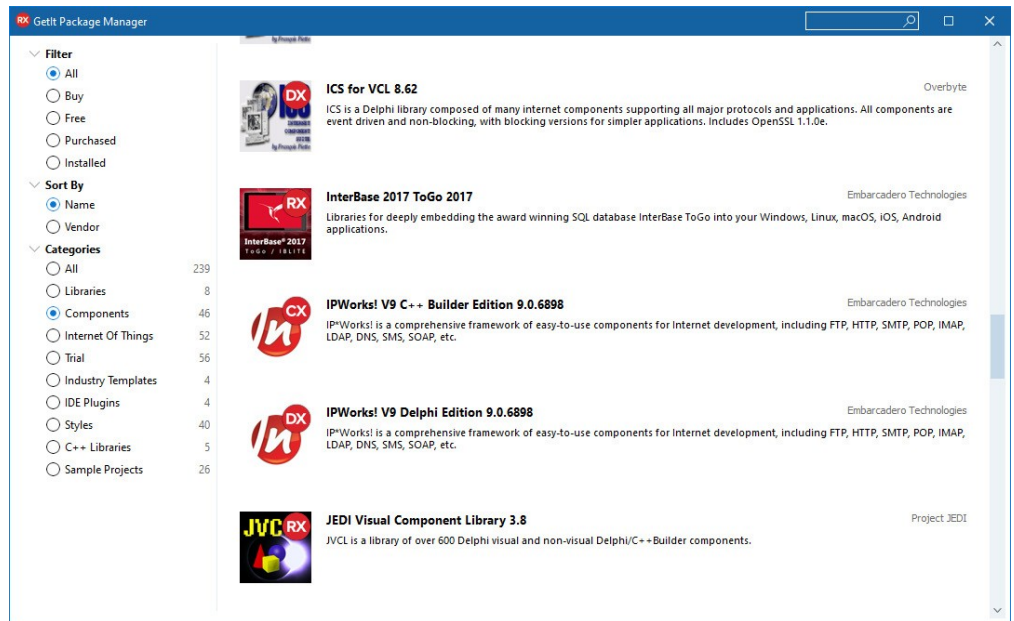
6 Principais Características e Benefícios do Delphi para Desenvolvedores Windows

Desde o início, o foco da Delphi tem sido fornecer o melhor ambiente de desenvolvimento visual baseado em componentes suportando compilação de código de máquina nativo, acesso a banco de dados e a ampla gama de APIs e tecnologias disponíveis para a plataforma Windows. Esta seção cobrirá alguns dos principais recursos e benefícios que fazem do Delphi a solução de desenvolvimento Windows mais bem avaliada. Os desenvolvedores podem encontrar uma [matriz de recursos completa](#) cobrindo Delphi e C++Builder no site da Embarcadero. O Delphi fornece compiladores otimizados de 32 bits e 64 bits e cadeias de ferramentas que suportam várias versões do Windows, incluindo Windows 7, 8.x e 10. A compatibilidade com o código-fonte permite que o mesmo código Delphi seja reutilizado em várias versões do Windows e versões do processador Intel para dar suporte aos seus usuários.

Os componentes da VCL (biblioteca de componentes visuais) e não-visuais suportam várias versões do Windows e funcionam com processadores de 32 bits e 64 bits. O desenvolvimento baseado em componentes permite que os desenvolvedores criem Interfaces visuais, acesso ao banco de dados, arquiteturas e outras lógicas de negócios definindo propriedades e escrevendo a menor quantidade de código possível. Os componentes podem ser usados em design e criados em tempo de execução. As propriedades podem ser definidas no tempo de projeto e de execução. Os componentes podem ser estendidos por meio de herança e novos componentes podem ser criados a partir de classes base. Há também é um grande ecossistema que fornece componentes e ferramentas adicionais que estão disponíveis gratuitamente e para compra quem atendem a maioria dos requisitos de negócios e aplicativos. O Delphi fornece inúmeros componentes para interface de usuário e recursos do sistema Windows.



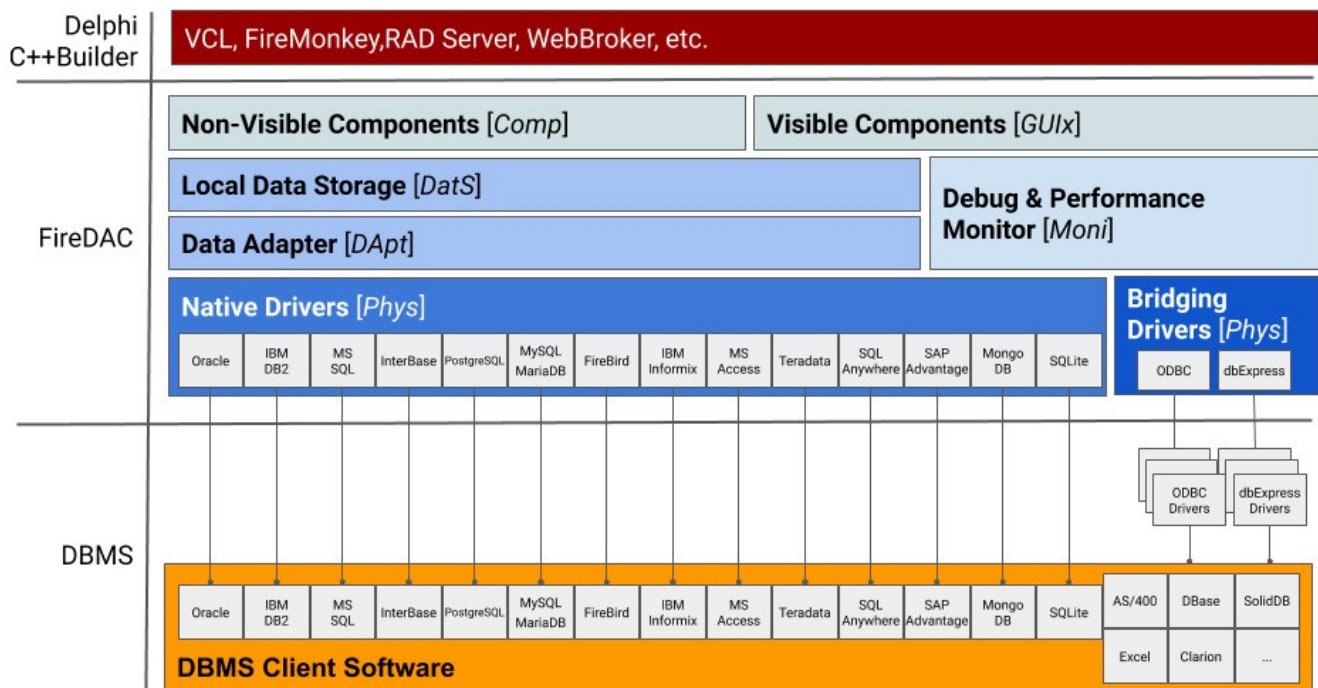
Os desenvolvedores também podem baixar e instalar componentes adicionais usando o GetIt Package Manager no IDE:



Informações adicionais sobre a VCL pode ser encontrado no Embarcadero DocWiki em [http://docwiki.embarcadero.com/RADStudio/en/Introducing_the_Visual_Component_Library_\(VCL\)](http://docwiki.embarcadero.com/RADStudio/en/Introducing_the_Visual_Component_Library_(VCL))

6.1 FireDAC

O FireDAC inclui um conjunto de componentes de acesso a dados para o desenvolvimento de aplicativos de banco de dados Windows e multidispositivos para Delphi e C++ Builder. O FireDAC permite acesso direto nativo de alta velocidade a InterBase, SQLite, MySQL, SQL Server, Oracle, PostgreSQL, IBM DB2, SQL Anywhere, Access, Firebird, Informix, MongoDB e mais. FireDAC tem uma arquitetura flexível, poderosa e extensível, conforme ilustrado no diagrama abaixo.

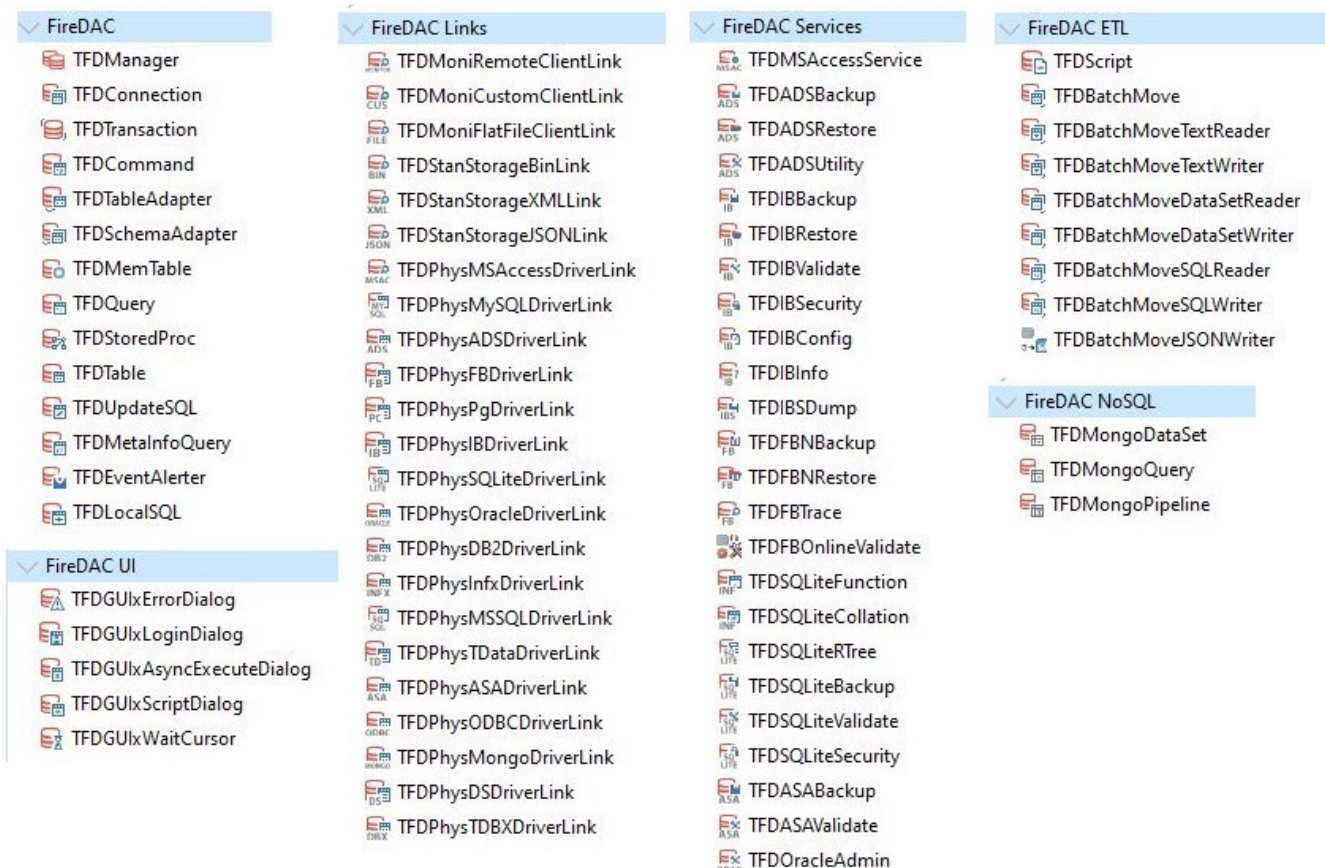


Usando o mesmo código-fonte e componentes, os desenvolvedores podem acessar recursos gerais e específicos de todos os bancos de dados (diretamente ou através de drivers) que são suportados pelo FireDAC. O FireDAC também inclui componentes para trabalhar com bancos de dados MongoDB No-SQL. O FireDAC também fornece interfaces de depuração e monitoramento de desempenho, que permitem monitorar e rastrear interações entre o aplicativo e as bases de dados conectadas.

Os benefícios do FireDAC incluem:

- FireDAC é o fruto de 10 anos de experiência no desenvolvimento de aplicações de banco de dados do mundo real.
- O FireDAC permite que o desenvolvedor se concentre no desenvolvimento do aplicativo, não nas complexidades da interação com o banco de dados.
- O FireDAC fornece um moderno conjunto de componentes ricos em recursos para atender a todos os principais ambientes RDBMS.
- O FireDAC usa um conjunto de componentes para atender a todos os tipos de RDBMS suportados.
- O FireDAC reduz o Custo Total de Propriedade, fornecendo:
 - Menor custo inicial, em comparação com bibliotecas externas caras.
 - Menos tempo necessário para aprender e se familiarizar com bibliotecas poderosas únicas.
 - Projeto e desenvolvimento de aplicativos unificados ao suportar diferentes plataformas e banco de dados.

Os componentes do FireDAC incluem:



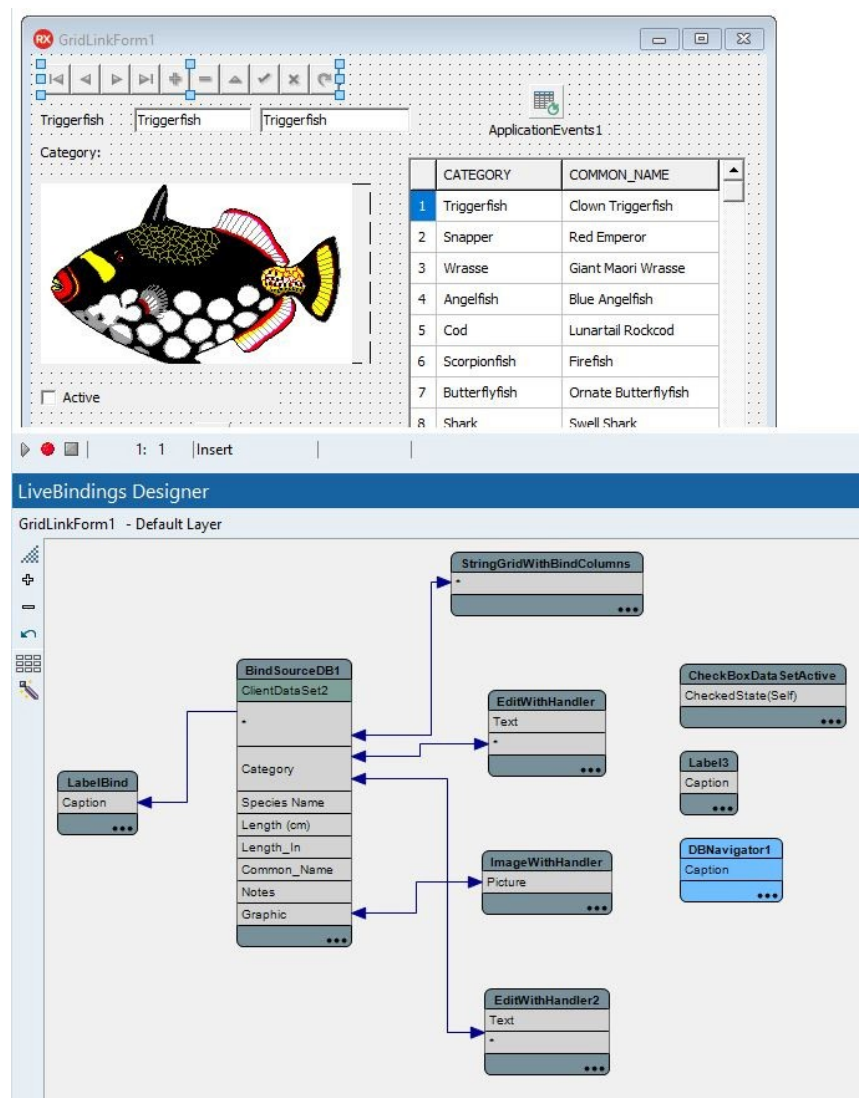
6.2 Visual LiveBindings

LiveBindings, uma estrutura baseada em expressão, é um recurso de vinculação de dados suportado pelos frameworks VCL e FireMonkey. LiveBindings é baseado em expressões relacionais, chamadas expressões vinculantes, que podem ser unidirecionais ou bidirecionais. Por meio de expressões vinculantes, qualquer objeto pode ser vinculado a qualquer outro objeto, simplesmente definindo uma expressão vinculante envolvendo um ou mais propriedades dos objetos a serem unidos.

Por exemplo, um desenvolvedor pode vincular um controle TEdit a um TLabel de modo que, quando o texto muda na caixa de edição, a legenda do rótulo é automaticamente ajustada ao valor avaliado pela expressão de vinculação. Outro exemplo é vincular um controle trackbar a uma barra de progresso de modo que o progresso aumente ou diminua à medida que o usuário move a barra. Da mesma forma, os campos de conjunto de dados podem ser conectados a objetos, uma ou mais propriedades de diferentes objetos podem ser alteradas, e assim por diante. Como o LiveBindings se propaga, um desenvolvedor pode até alterar propriedades de objetos que estão conectados a outros objetos que estão vinculados a um objeto de controle.

O LiveBindings Designer acelera o desenvolvimento conectando visualmente componentes e dados sem ter que escrever nenhum código. As vinculações ao vivo também podem ser usadas para gerar dados de protótipos antes que a conexão seja feita a um banco de dados real (talvez a equipe do banco de dados não tenha implementado o projeto do banco de dados quando o desenvolvimento é iniciado). As vinculações ao vivo podem, então, substituir facilmente os dados do protótipo por dados reais por um clique de mouse.

À medida que uma visão de LiveBindings Designer cresce, a janela de design pode ser ampliada para dentro e para fora e uma imagem dos LiveBindings atuais pode ser salva. Os elementos visuais na janela LiveBindings também podem ser escondidos com um clique do mouse. Camadas podem ser criadas para subconjuntos de objetos na janela LiveBindings para melhor organizar e ocultar conjuntos de elementos vinculados. Tools | Options suportam o LiveBindings Designer para permitir a exibição e ocultação de certos componentes da visualização do design.



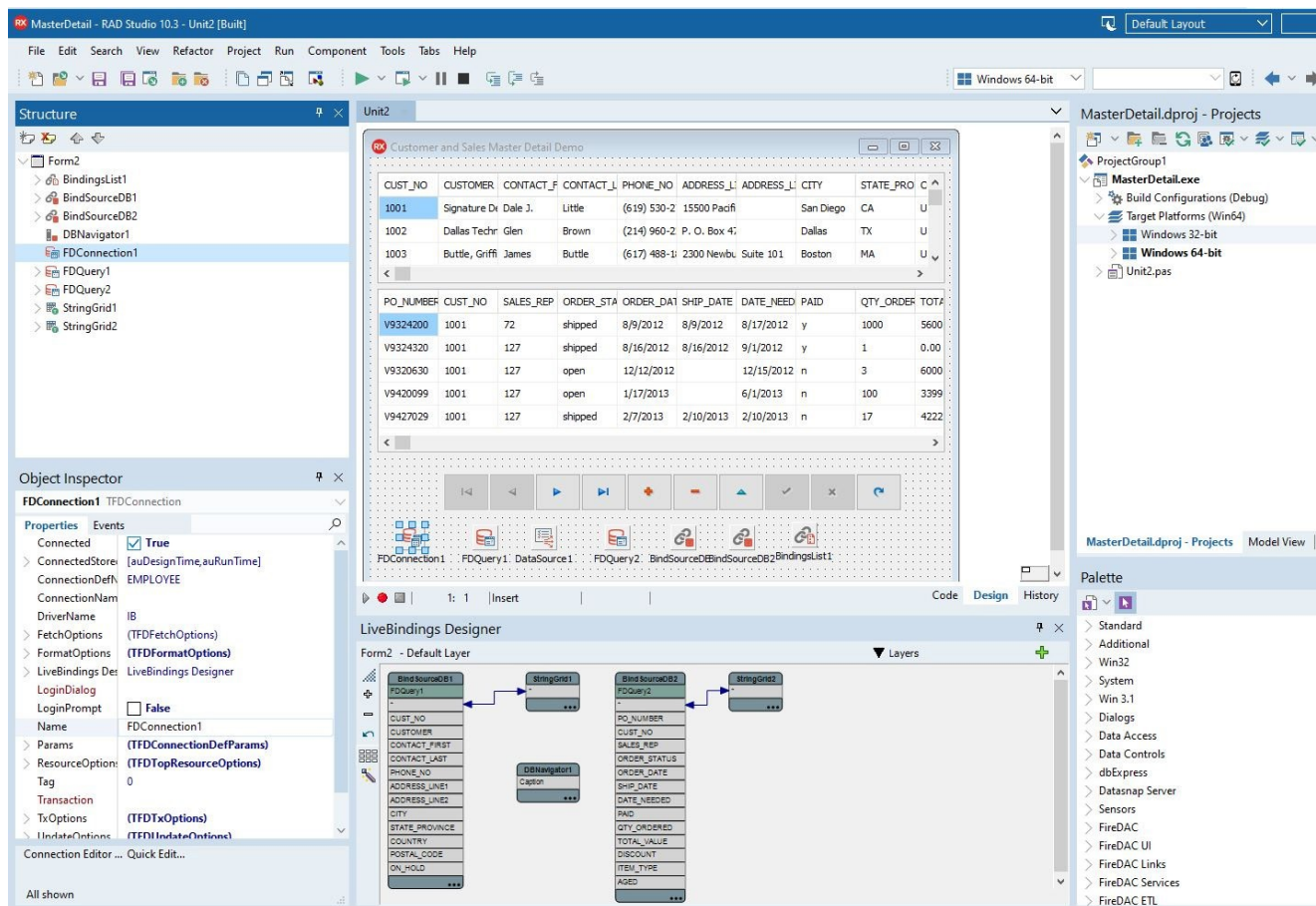
Para obter informações adicionais, confira o tutorial "Using LiveBinding in VCL Applications" on-line em

http://docwiki.embarcadero.com/RADStudio/en/Tutorial:_Using_LiveBinding_in_VCL_Applications

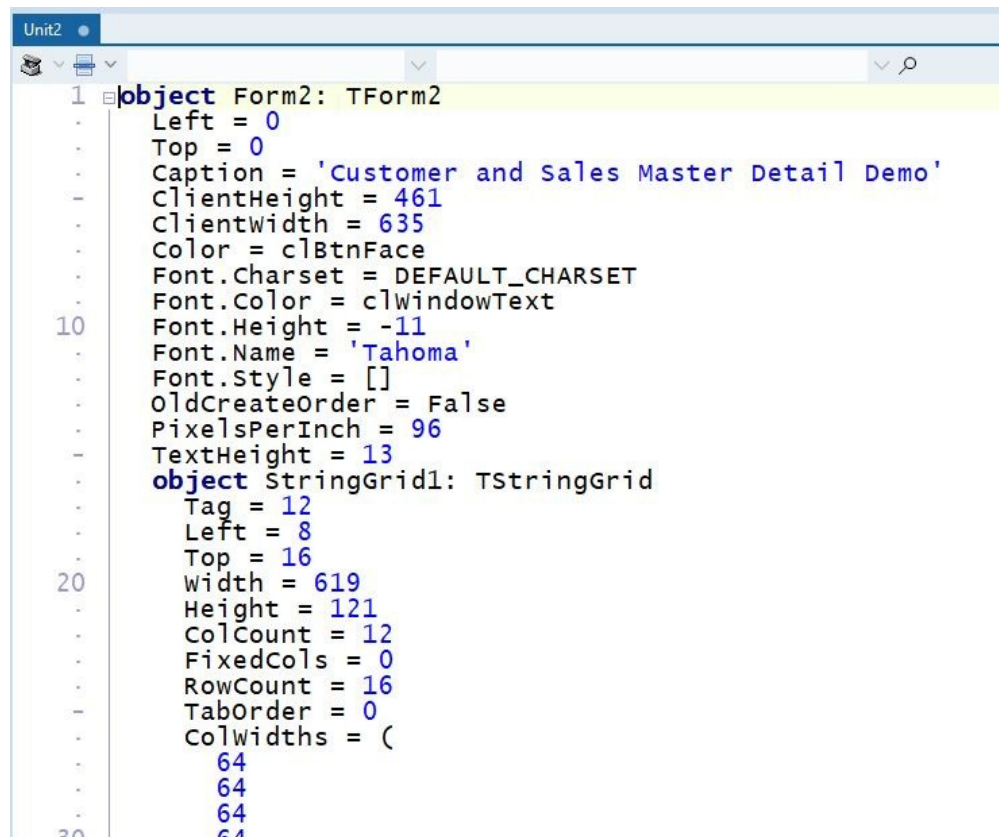
6.3 Ferramentas Visuais Two Way

O Designer de Formulários do IDE permite que os desenvolvedores construam visualmente aplicativos nativos do Windows, com dicas de ajuste ao vivo e diretrizes de layout. O recurso de ferramentas visuais de duas vias do Form Designer acelera o desenvolvimento da interface do usuário de um aplicativo do Windows e facilita a criação de menos código usando componentes, propriedades e eventos.

Alterações podem ser feitas na interface do aplicativo usando o inspetor de objetos. As alterações também podem ser feitas numa versão de modo de texto de cada formulário (o texto é uma representação do fluxo de componentes, propriedades e configurações de eventos). As alterações feitas no código-fonte também são refletidas na forma, componentes, propriedades e eventos. O bitmap abaixo mostra um conjunto completo de detalhes sem código, cliente e vendas no Form Designer com LiveBindings e dados ao vivo em tempo de design.



O botão direito do mouse pode ser usado no designer de formulários para alterar a exibição para "Form as a text". Faça quaisquer alterações nas configurações da propriedade (por exemplo, altere a consulta SQL e a legenda de formulário) e, em seguida, mude de volta para o design de formulário para ver o resultado.



O Inspetor de Objetos incluído permite que os desenvolvedores definam rapidamente propriedades e eventos de componentes. Os designers QuickEdit são fornecidos no designer de formulários com itens adicionais ao menu de designer, e um novo painel é fornecido para editar propriedades de controle comuns mais rápido do que em versões anteriores do Delphi.

Os desenvolvedores também podem usar a herança de formulário visual e a vinculação de formulários para reduzir a codificação e simplificar a manutenção. Frames também são fornecidos para construir e reutilizar componentes compostos. O Delphi também fornece um Repositório de Objetos para armazenamento e reutilização de formulários, Data Modules, além de experts. Assim como os dados do banco de dados ao vivo podem ser vistos no designer de formulários, os resultados das chamadas de serviço REST podem ser vistos ainda em design. Esses recursos podem economizar tempo de desenvolvimento e depuração mostrando dados que não se encaixam em componentes da UI, erros em declarações de SQL, problemas com a arquitetura do banco de dados e conexões de serviço REST.

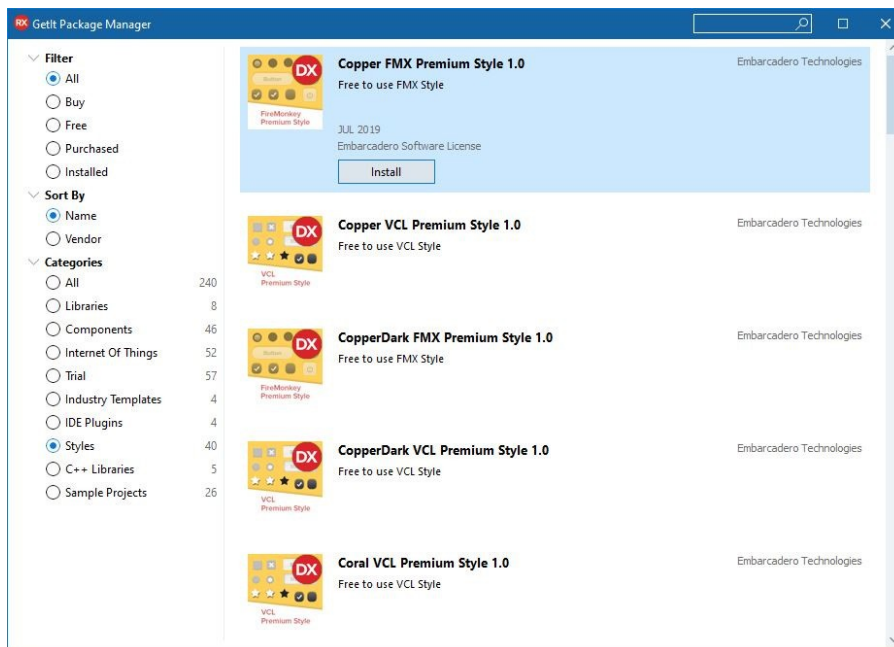
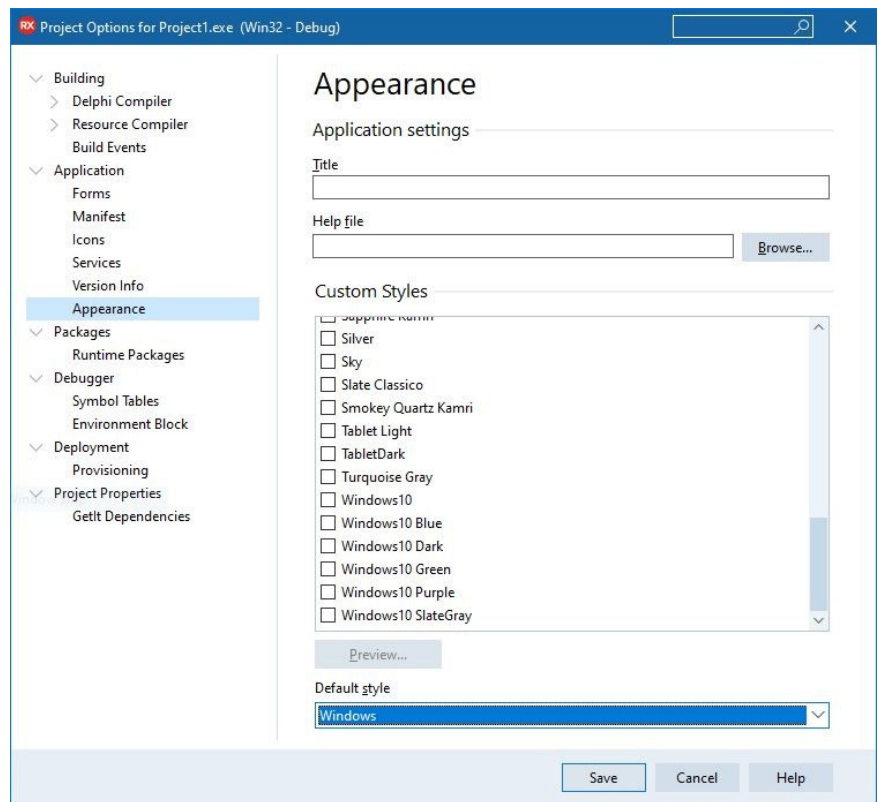
6.4 Estilos e Temas

O sistema operacional Windows forneceu um conjunto padrão de temas desde o Windows 98 onde os usuários podem optar por definir a aparência padrão e a sensação para o desktop e aplicativos. Os componentes VCL e FMX também podem ser estilizados para adaptar-se aparência do Windows 10 e o conjunto Metrópolis. E, no FMX, também existem estilos padrão para cada plataforma multidispositivos suportada. Os desenvolvedores também podem criar estilos próprios para personalizar seus aplicativos.

Delphi inclui um conjunto padrão de estilos que podem ser selecionados usando o Project | Options | Appearance. Código também pode ser escrito para selecionar estilos em tempo de execução.

Há também um conjunto de estilos Premium disponíveis para download através do GetIt Package Manager.

Para a VCL, os desenvolvedores podem criar estilos personalizados com o VCL Style Designer. A VCL também inclui suporte a estilos para diálogos comuns, o componente TWebBrowser e para descendentes de TGrid. Um estilo permite que os desenvolvedores mudem a aparência de cada peça e estado de um controle. Os controles VCL são compostos por partes e estados. Um estilo VCL consiste em um conjunto de valores para cada uma dessas partes e estados. Por exemplo, uma barra de rolagem tem as seguintes partes: quadro, controle deslizante e os dois botões laterais para cada direção. Os botões laterais, por exemplo, têm os seguintes estados: pressionado,



desabilitado, hot e normal.

No FireMonkey, cada classe de controle tem um estilo padrão, codificado por plataforma. Uma cópia do estilo interno para uma classe de controle é criada quando o comando Edit Default Style é clicado no menu pop-up do controle.

O componente TStyleBook contém as informações sobre o estilo padrão ou personalizado. Os desenvolvedores podem usar a janela do FireMonkey Style Designer para editar o estilo padrão ou um novo estilo. As informações de estilo são armazenadas em um componente TStyleBook.

No Structure Panel da janela de Style Designer, um desenvolvedor pode selecionar o componente do controle a ser alterado. Faça as edições necessárias, salve-as e feche a exibição. Agora, o StyleBook que foi criado inicialmente contém todas as alterações. Para ver as edições, clique duas vezes no StyleContainer.

6.5 Windows API e WinRT Units

O Delphi inclui o código-fonte da biblioteca de tempo de execução que contém wrappers para as APIs do Windows e também as APIs do WinRT. Embora a maioria dos desenvolvedores use os componentes para acessar propriedades, métodos e eventos para operações do Windows, haverá momentos no código onde pode haver a necessidade de chamar diretamente as APIs do Windows.

O código-fonte de todas as unidades está disponível nos seguintes diretórios de instalação:

- \Embarcadero\Studio\20.0\source\rtl\win
- \Embarcadero\Studio\20.0\source\rtl\win\winrt

Nota: 20.0 é a designação para a versão 10.3 Rio. Para uma versão anterior do Delphi, o caminho da pasta terá um número menor.

6.6 Biblioteca de Programação Paralela (PPL)

Incluído no RTL está a Biblioteca de Programação Paralela (PPL), proporcionando aos desenvolvedores a capacidade de ter tarefas em execução em paralelo. O PPL inclui uma série de recursos avançados para executar tarefas, juntar tarefas, esperar por grupos de tarefas, etc. O PPL inclui um pool de threads que faz o balanço automático (com base na carga da CPU) evitando o cuidado necessário para criar ou gerenciar threads.

Os desenvolvedores podem usar PPL incluindo a unidade `System.Threading` em seus aplicativos. Esta unidade é composta de vários recursos que podem ser incluídos em projetos novos e existentes. A unidade também inclui uma série de argumentos sobrecarregados para torná-lo adequado tanto para C++ bem como Delphi. O PPL funciona em dispositivos Windows (32 e 64 bits), macOS, Android e iOS.

Usando o PPL, o código do aplicativo pode facilmente:

- Faça loops mais rápido com `TParallel.For`
Tutorial: Using the For Loop from the Parallel Programming Library
- Execute várias tarefas em paralelo usando `TTask` e `Itask`.
Tutorial: Using Tasks from the Parallel Programming Library
- Deixe um processo em execução, foque em outras tarefas e, em seguida, obtenha o resultado desse processo no ponto desejado. O `IFuture` permite que o aplicativo estabeleça uma prioridade para os blocos de código em execução e ainda retorne os resultados quando necessário.

Tutorial: Using Futures from the Parallel Programming Library

Incluídos com Delphi estão aplicativos de exemplo VCL e FMX que implementam uma versão paralelizada do **Conways Game of Life**.

Para obter informações adicionais e tutoriais, acesse o Embarcadero DocWiki online em http://docwiki.embarcadero.com/RADStudio/en/Using_the_Parallel_Programming_Library

6.7 Componentes Windows 10, High DPI e suporte Multi-Monitor

O Delphi continua a evoluir para suportar as últimas versões do Windows com suporte para novos recursos do Windows 10. Aqui estão alguns dos importantes recursos do Delphi fornecidos para garantir que os aplicativos fiquem ótimos no Windows 10 e suportem o novo sistema operacional.

6.7.1 Componentes para Windows 10

Na paleta de componentes, os desenvolvedores encontrarão vários componentes do Windows 10 que também funcionam no Windows 7 e 8.x.

O componente `TWindowsStore` conecta um aplicativo do Windows com a Windows Store e fornece recursos, incluindo obter uma lista de aplicativos de propriedade do usuário, complementos disponíveis, complementos comprados e para suportar aplicativos no modo de teste. O componente `TWindowsStore` é suportado apenas para plataformas Windows e inclui uma versão VCL e uma versão FireMonkey.

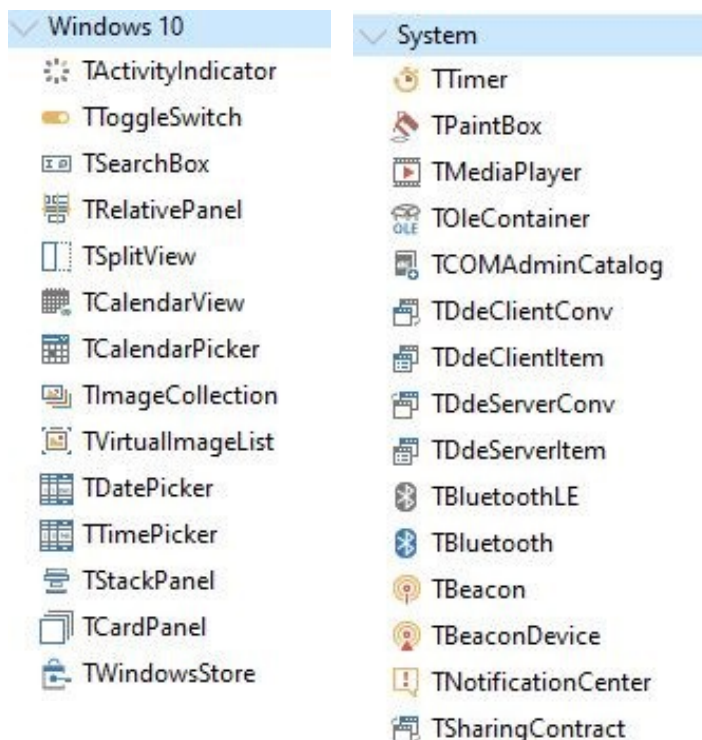
O componente `TNotificationCenter` pode ser usado para gerenciar notificações de vários dispositivos. Um aplicativo pode usar a central de notificação para enviar mensagens para informar o usuário. Usando notificações, as mensagens são exibidas na área de notificação designada para cada plataforma suportada.

Notificações podem ser usadas no Windows usando VCL e FMX. O `TNotificationCenter` funciona no Windows 8.x, 10, macOS, iOS e Android.

O `TSharingContract` permite que um aplicativo compartilhe conteúdo com outro aplicativo do Windows usando a funcionalidade de um Contrato de Compartilhamento. O Delphi implementa a API do Contrato de Compartilhamento que permite que um configure um aplicativo como fonte de compartilhamento.

O alvo compartilhado tem que ser um aplicativo nativo do Windows. A implementação da funcionalidade Sharing Contract suporta o uso do componente `TSharingContract` ou implementando um componente `TCustomSharingContract`. As propriedades dos componentes do `TSharingContract` incluem:

- O `DataTitle` e a `Description` especificam o título e a descrição do título que aparece na barra lateral.
- `ImageFile` permite que um desenvolvedor especifique uma imagem para compartilhar. Os desenvolvedores também podem fornecer um manipulador de eventos personalizado para o evento `OnTransferImage`, que permite que o aplicativo passe qualquer fluxo para o aplicativo de destino.
- Outras propriedades permitem o compartilhamento de muito mais informações com o aplicativo de destino. Consulte o [Windows Dev Center – Diretrizes](#) para compartilhamento de conteúdo e [Windows Dev Center - DataPackage](#) para obter mais informações sobre o compartilhamento de contratos.



6.7.2 Suporte High DPI no Windows 10

Jim McKeeth, Marco Cantu e David Millington escreveram posts no blog sobre suporte a High DPI para aplicativos VCL no Windows 10. Em seu post, Jim disse: "Quando você tem um monitor de DPI alto (>96 DPI), o Windows aplica um fator de escala para que, em vez de seu conteúdo ficar muito pequeno e difícil de ler, torna as fontes mais nítidas e as imagens mais detalhadas. O VCL lida automaticamente com a maioria dos o redimensionamento para você, mas há uma nova função no VCL.Classes chamada `GetSystemMetricsForWindow` que aproveita a nova API `GetSystemMetricsForDPI` (como alternativa para o `GetSystemMetrics` antigo). Ele lhe dará métricas, resolução e DPI para cada janela do seu aplicativo, com compreensão baseada no monitor em que ele aparece atualmente."



O suporte a High DPI está incluído na versão 10.3 Rio do Delphi com aprimoramentos para:

- **Suporte a lista de imagens e coleções de imagens**
- **Suporte por Monitor V2** (use o Project | Options | Application | Manifest page)
- Novos estilos que suportam High DPI
- APIs atualizadas do Windows para High DPI

Veja informações adicionais nas postagens do blog em:

- <https://community.idera.com/developer-tools/b/blog/posts/high-dpi-on-windows-10>
- <https://community.idera.com/developer-tools/b/blog/posts/windows-10-and-modern-high-dpi-display-support-in-delphi-and-c-builder>

6.7.3 Suporte a vários Monitores para VCL e FMX

A VCL fornece uma maneira de lidar com vários **monitores**. Uma variável global `Screen` (do tipo `TScreen`) é criada quando um projeto é iniciado. `TScreen` encapsula o estado da tela na qual um aplicativo está sendo executado. As tarefas comuns executadas pela `TScreen` incluem:

- Manuseio da lista e parâmetros de vários monitores acessíveis pelo aplicativo.
- O tamanho da janela em que o aplicativo está sendo executado.
- Uma lista de fontes disponíveis para o dispositivo de tela.
- O estilo do cursor.

Se o aplicativo VCL for executado em vários monitores, a VCL gerenciará efetivamente o layout da interface de usuário do aplicativo em vários monitores. Por exemplo, menus, caixas de diálogo e outros controles pop-up abrem no mesmo monitor que o formulário pai. Use a variável Screen para obter uma lista de monitores e suas dimensões. A propriedade Monitors fornece acesso a todos os monitores usados para compor a área de trabalho, e o MonitorCount retorna o número de monitores. Todas as coordenadas são relativas ao monitor primário.

O FMX suporta automaticamente a execução de um aplicativo em **vários monitores**. Os desenvolvedores não precisam implementar nenhuma programação para suportar a execução de uma aplicação FireMonkey em vários monitores. O FireMonkey gerencia efetivamente o layout da interface de usuário do aplicativo em vários monitores. Por exemplo, menus, caixas de diálogo e outros controles pop-up abrem no mesmo visor do formulário pai. O suporte a vários monitores na estrutura FireMonkey é semelhante ao suporte a vários monitores em aplicativos VCL. A classe TScreen define métodos e propriedades que fornecem acesso a parâmetros de vários monitores. O aplicativo cria automaticamente uma variável de Screen global, do tipo TScreen. A Screen encapsula o estado da tela em que o aplicativo está sendo executado.

Use a Screen para obter:

- Uma lista de displays (usado para compor a área de trabalho)
- Coordenadas, dimensões e outras propriedades desses displays

A propriedade DisplayCount retornar o número de telas e Displays fornece acesso às propriedades de todos os displays. Todas as coordenadas são coordenadas de tela virtual. As coordenadas da tela virtual são relativas ao display principal e medidas em pixels. A tela virtual é o retângulo delimitador de todas as telas usadas para compor a área de trabalho.

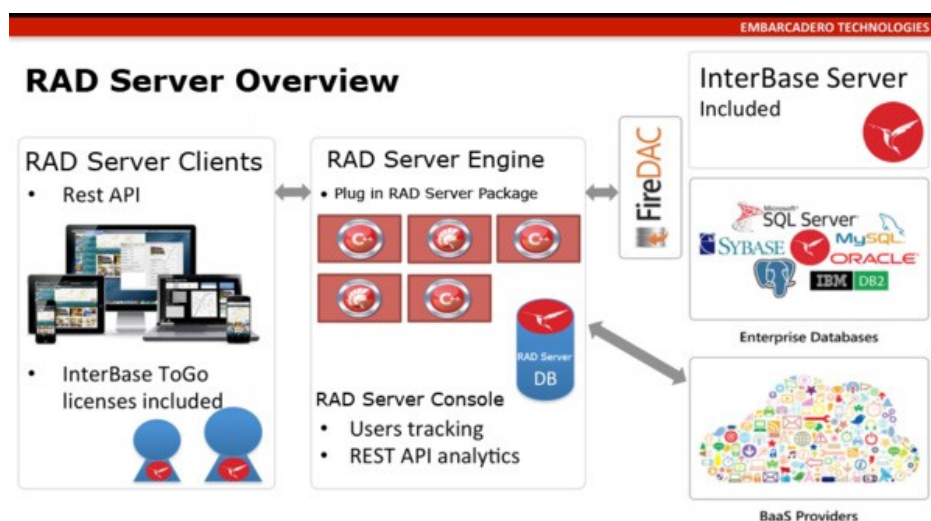
6.8 RAD Server

O RAD Server da Embarcadero fornece uma base de aplicativos para a construção rápida e implantação de serviços baseados em serviços usando RAD Studio, Delphi e C++Builder.

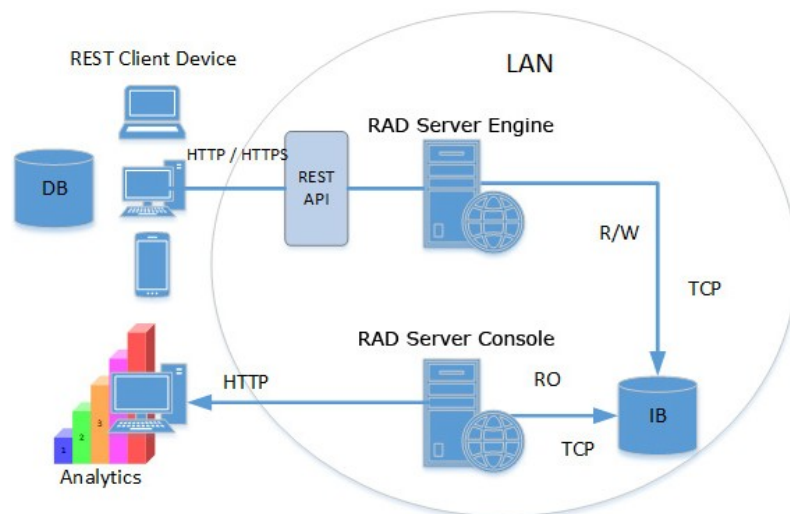
O RAD Server suporta o estilo arquitetônico REST (Representational State Transfer) com passagem de parâmetros e retornos em JSON (ou XML).

Os desenvolvedores podem publicar APIs, gerenciar usuários e dispositivos

que estão conectados ao RAD Server, capturam análises sobre o uso e usuários dos aplicativos, conectam-se a bancos de dados locais e corporativos usando os componentes FireDAC e se conectam com dispositivos Internet of Things (IoT). O RAD Server também suporta autenticação do usuário, notificações push, geolocalização e armazenamento de dados.



Com os assistentes, componentes e ferramentas do RAD Server, os desenvolvedores podem criar rapidamente novos aplicativos de middleware e back-end ou migrar os aplicativos cliente/servidor existentes Delphi e C++ Builder para um aplicativo baseado no RAD Server para executar em um servidor ou na nuvem. Os desenvolvedores podem publicar endpoints para chamadas REST de desktop, mobile, console, web e outros tipos de aplicativos. O RAD Server vem com um conjunto completo de ferramentas, componentes, conectividade de banco de dados e interfaces que os desenvolvedores podem aproveitar na construção de aplicativos baseados em serviço.



Os aplicativos RAD Server podem ser implantados nos servidores web Microsoft Windows IIS e Apache. Os desenvolvedores também podem implantar serviços baseados em Delphi para servidores Linux Intel de 64 bits. O RAD Server também é fornecido como um contêiner **Docker** de fácil implantação.

Informações adicionais do RAD Server estão disponíveis:

- The Complete Guide to RAD Server eBook –
<http://altd.embarcadero.com/releases/studio/general/The%20Complete%20Guide%20to%20RAD%20Server%20eBook.pdf>
- RAD Server Webinar Replay –
https://www.youtube.com/watch?v=eCyTiDq_jBY
- RAD Server Embarcadero Academy Course for Delphi –
<https://www.embarcaderoacademy.com/courses/author/226326>
- RAD Server Docker Deployment –
http://docwiki.embarcadero.com/RADStudio/Rio/en/RAD_Server_Docker_Deployment

6.9 App Tethering

A RTL fornece componentes de tethering de aplicativos, dando aos aplicativos a capacidade de interagir com outros aplicativos em execução na mesma máquina ou em uma máquina remota. Usando o recurso de tethering, os aplicativos podem facilmente:

- Descubra outros aplicativos que estão usando App Tethering no mesmo dispositivo ou em outros dispositivos conectados.
- Executar ações remotamente. Um aplicativo pode publicar ações usando o App Tethering. Em seguida, outros aplicativos podem invocar remotamente qualquer uma dessas ações no aplicativo anterior.
- Compartilhar dados entre aplicativos com tipos de dados padrão e streams.

O recurso de tethering de aplicativo não depende de um transporte ou protocolo específico, e novos transportes e protocolos podem ser implementados usando a API de tethering do aplicativo. O RTL oferece suporte integrado para conexões IP e Bluetooth Classic.

O App Tethering fornece dois componentes:

- **TTetheringManager** – Use este componente para descobrir outros aplicativos que estão usando tethering de aplicativos, executando no mesmo dispositivo ou em outros dispositivos conectados.
- **TTetheringAppProfile** – Use este componente para definir as ações e dados que um aplicativo compartilha com outros aplicativos anteriormente emparelhados usando o **TTetheringManager**.

Existem vários exemplos de tethering de aplicativos incluídas para Delphi no `\Documents\Embarcadero\Studio\20.0\Samples\Object Pascal\RTL\Tethering`. Estas amostras incluem:

- **BDSshoppingList** –
http://docwiki.embarcadero.com/CodeExamples/en/RTL.BDSshoppingList_Sample
- **DesktopCast** –
http://docwiki.embarcadero.com/CodeExamples/en/RTL.DesktopCast_Sample
- **MediaPlayer** –
http://docwiki.embarcadero.com/CodeExamples/en/RTL.MediaPlayer_Sample
- **PhotoWall** –
http://docwiki.embarcadero.com/CodeExamples/en/RTL.PhotoWall_Sample

6.10 Bluetooth, Beacons, Beacon Fencing e IoT

O Delphi fornece classes e componentes para implementar suporte Bluetooth (Classic e LE) em aplicativos. Quando um aplicativo está sendo executado em um dispositivo habilitado para Bluetooth, o aplicativo pode usar os recursos Bluetooth do dispositivo para se conectar a aplicativos em dispositivos remotos e trocar dados com esses aplicativos remotos.

Para implementar o suporte Bluetooth Classic em um aplicativo:

- Adicione um componente **TBluetooth** ao aplicativo.
- Descubra dispositivos remotos e faça par com eles.
- Conecte-se aos dispositivos emparelhados.
- Troque dados com os dispositivos conectados.

Para implementar o suporte Bluetooth LE para serviços padrão em um aplicativo:

- Coloque um componente **TBluetoothLE** em um formulário
- Descubra dispositivos **BLUETOOTH LE**
- Descubra e obtenha os serviços para o dispositivo
- Descubra e obtenha as características do serviço
- Ler, escrever ou assinar uma característica
- Subtraia as informações dos dados brutos

Platform	Classic Bluetooth	Bluetooth Low Energy	
		Client	Server
Windows*	✓	8+	
OS X	✓	10.7+	10.9+
iOS		5+	6+
Android	✓	4.3+	5+

Um beacon é um dispositivo Bluetooth de baixa energia que inclui informações em seus dados de advertising, especificamente na área Manufacturer Specific Data. Essas informações permitem que qualquer dispositivo Bluetooth de baixa energia na área circundante identifique o dispositivo e calcule a distância até ele sem ser pareado ou estabelecendo uma conexão.

* A API RTL para beacons está disponível apenas para plataformas Windows 10+. Os dois formatos disponíveis são iBeacon e AltBeacon. As versões anteriores do Windows não suportam anúncios BLE.

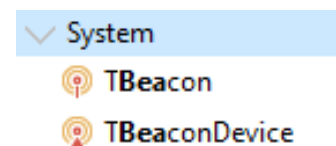
Platform	Supported
Windows(*)	10+
OS X	10.7+
iOS	5+
Android	4.3+

Para as outras plataformas suportadas, três formatos de beacon estão disponíveis: iBeacon, AltBeacon e Eddystone. A API RTL para beacons os diferencia usando os tipos TBeaconScanMode e TKindofBeacon. Os desenvolvedores precisam saber o formato de beacon que deseja monitorar:

- iBeacon – formato definido pela Apple. As especificações completas estão disponíveis [aqui](#).
- AltBeacon – formato aberto. As especificações completas estão disponíveis [aqui](#).
- Eddystone – formato aberto definido pelo Google. As especificações completas estão disponíveis [aqui](#).

Os componentes e classes do beacon incluem:

- TBeacon – TBeacon é a classe para o componente do beacon.
- TBeaconDevice – TBeaconDevice é a classe para o componente beacon device.
- TBeaconRegionCollection – TBeaconRegionCollection é uma coleção que tem uma lista de TBeaconRegionItem.
- TBeaconRegionItem – TBeaconRegionItem é a classe para conter as informações sobre uma região registrada.
- TCustomBeacon – TCustomBeacon é a classe base do componente TBeacon.
- TCustomBeaconDevice – TCustomBeaconDevice é a classe base do componente TBeaconDevice.



Informações adicionais do Beacon estão disponíveis em

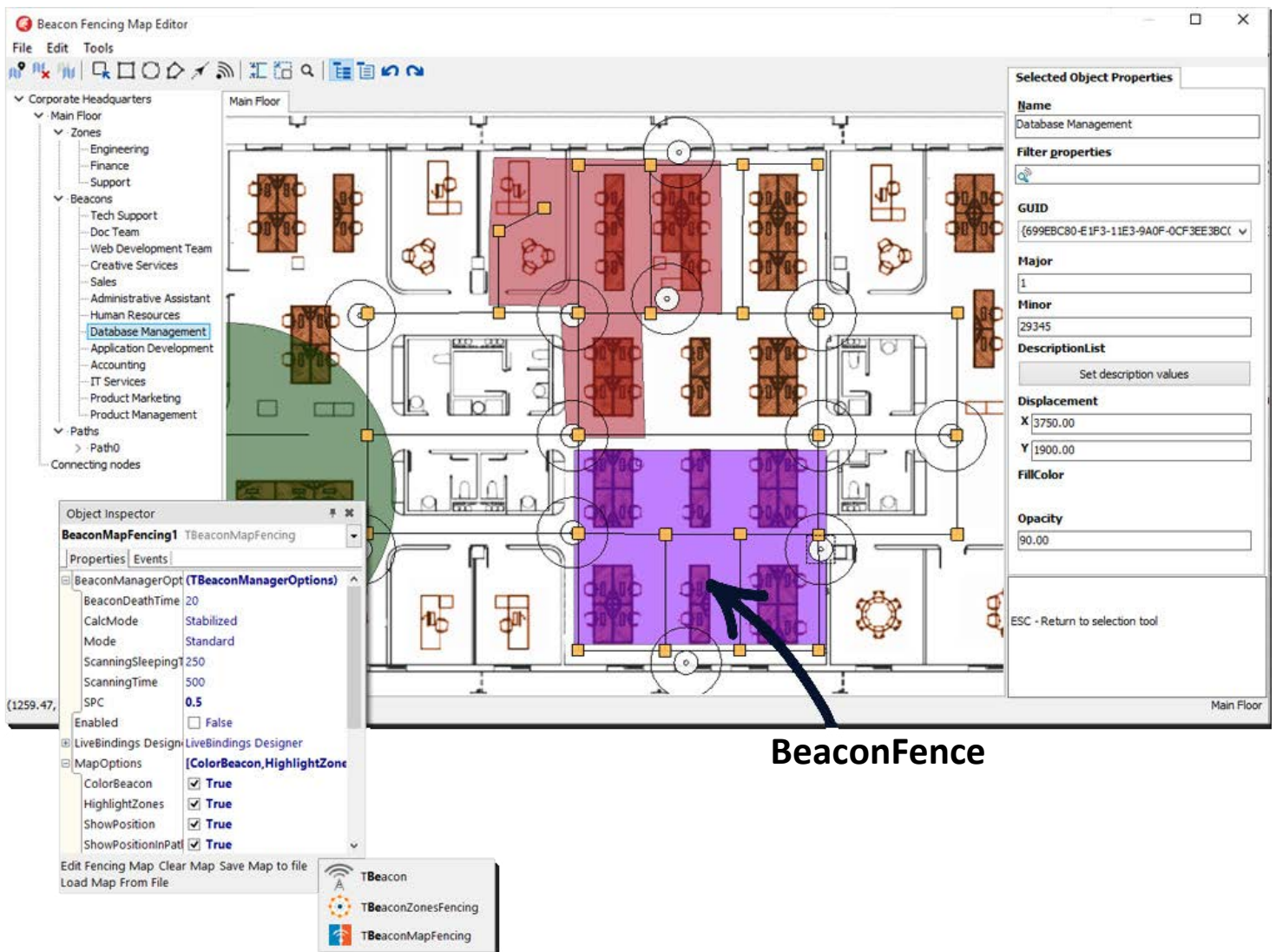
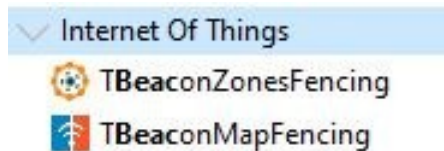
http://docwiki.embarcadero.com/RADStudio/en/Using_Beacons

O **BeaconFence** permite que os desenvolvedores usem recursos de geofencing em seus aplicativos. BeaconFence melhora o System.Beacon.TBeaconManager ao incluir o BeaconFence Map Editor para ajudar um desenvolvedor a projetar um mapa de um espaço físico de interesse, como uma área geográfica, um edifício ou, em geral, qualquer ambiente real ou virtual. BeaconFence está disponível para download no GetIt Package Manager para desenvolvimento e implantação limitada.



A implantação do BeaconFence está incluída com **as licenças RAD Server Enterprise** e Architect. O BeaconFence inclui propriedades e eventos adicionais, incluindo o evento OnZoneEnter para desencadear um evento no aplicativo cliente do usuário à medida que uma região predefinida é inserida.

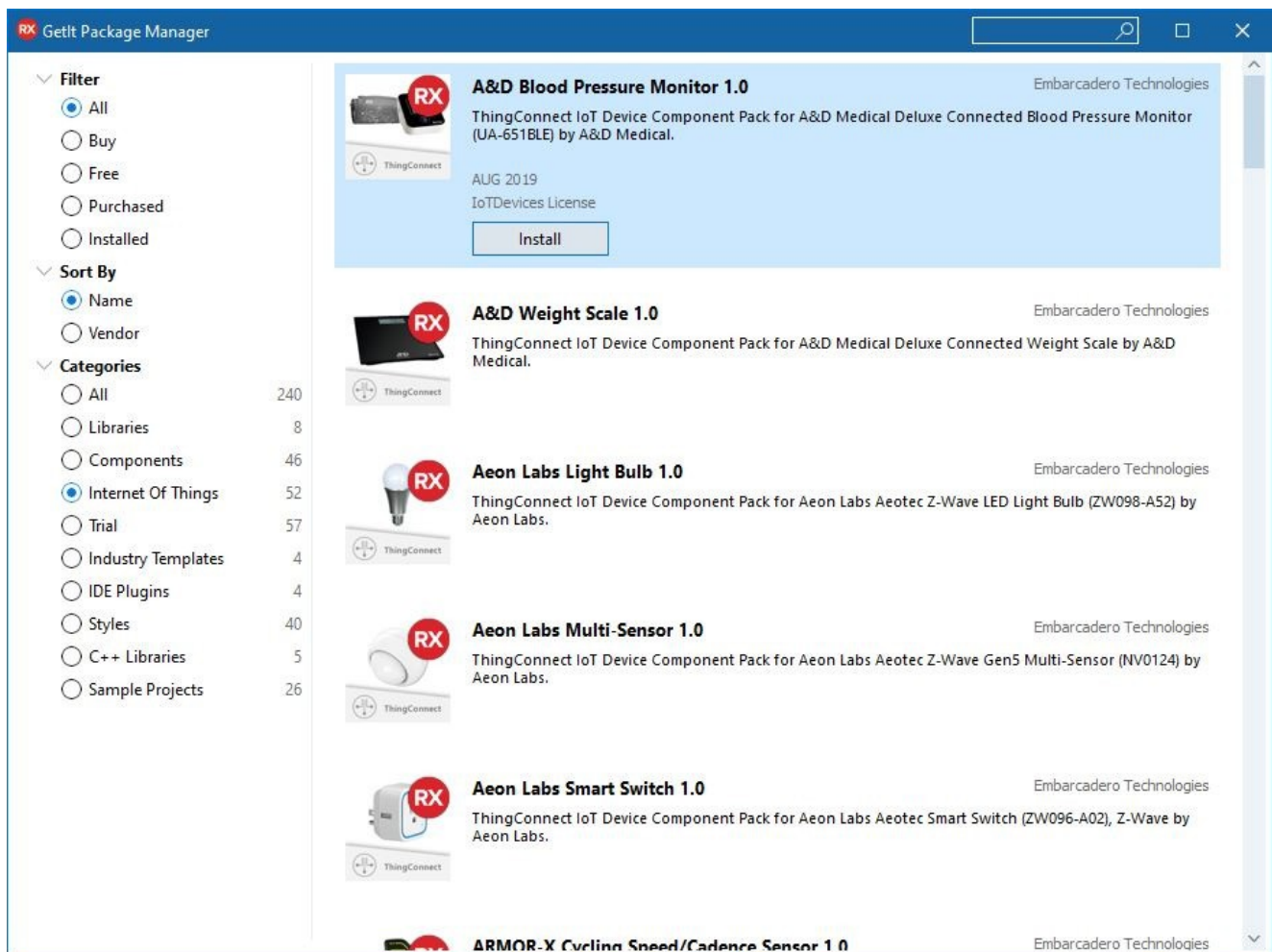
O componente também inclui um Beacon Fencing Map Editor para criar um bitmap usando este editor visual, ou carregar um bitmap existente, como um plano de piso de um escritório, para usar como parte das informações do mapa de beacon-fencing.



O BeaconFence fornece um aplicativo com os seguintes recursos:

- Monitore os beacons e acione eventos de proximidade.
- Defina zonas e acione eventos de proximidade para: entrar na zona, sair da zona e se aproximar de uma zona.
- Calcule a posição dos dispositivos móveis em um mapa.
- Calcule rotas entre pontos de interesse em um mapa.

O Delphi inclui suporte para uma ampla gama de **dispositivos IoT** que suportam Bluetooth. Os componentes IoT cobrem dispositivos desde cuidados de saúde até fitness e automação residencial. Estão incluídos os componentes do dispositivo BluetoothLE (BLE) e Z-Wave. Os dispositivos IoT suportados incluem: monitores de pressão arterial, monitores de frequência cardíaca, balanças de peso, lâmpadas, interruptores inteligentes, sensores de temperatura/umidade, sensores de água/inundação, sensores de óxido de fumaça e carbono, termostatos programáveis, fechaduras de portas, termômetros, etc. Os desenvolvedores podem baixar mais de 50 componentes de IoT e aplicativos de amostragem usando o GetIt Package Manager.



6.11 Interbase

InterBase é um banco de dados compatível com padrões SQL. O InterBase segue rigorosamente os padrões da indústria para o SQL. O InterBase suporta Unicode e é ideal para qualquer conjunto de texto globalmente. O InterBase oferece vários alertas de eventos em tempo real, criptografia e o rastreamento de alterações baseado em Change Views – tecnologia patenteada.

Os recursos do InterBase incluem:

Change Views acelera aplicativos e reduz o tráfego e os custos de rede identificando e buscando apenas as alterações feitas no banco de dados.

- A Arquitetura Multigeracional se conecta a instantâneos consistentes do banco de dados, garantindo que os readers não impeçam os writers de fazer alterações e aumentem a escalabilidade.
- Os Alertas de Eventos permitem um alerta em tempo real para clientes conectados quando uma alteração é cometida. Isso economiza na pesquisa intensiva de banco de dados e aumenta a escalabilidade.
- Os desenvolvedores podem facilmente personalizar o InterBase com tipos de campo personalizados, procedimentos armazenados, exceções personalizadas e triggers. A personalização ajuda a automatizar a lógica do banco de dados, da transformação de dados e a fácil manutenção.
- Os Generators permitem que as aplicações busquem números sequenciais que são normalmente usados para chaves primárias. O InterBase não atribuirá chaves duplicadas. Os Generators permitem que o InterBase produza valores-chave centralmente no banco de dados.
- O InterBase fornece criptografia *over-the-wire* e *at-rest*, login de segurança separado e segurança do usuário baseada em papéis. A criptografia do InterBase adiciona muito pouca sobrecarga à velocidade e desempenho do banco de dados, mantendo-se completa na criptografia de disco.

O InterBase inclui a **edição do desenvolvedor e 4 edições distribuíveis**:

- Edição do Desenvolvedor
- IBLite Embedded
- IBToGo Embedded
- Edição Desktop
- Edição Server

6.12 Enterprise Connectors

A Embarcadero anunciou em novembro passado que uma licença de assinatura de 1 ano do Enterprise Connectors (CData FireDAC Professional) num valor estimado de US\$ 999, está agora incluída como parte das compras das edições 10.3 Delphi, C++Builder e RAD Studio Enterprise e Architect.

Os benefícios chave do Enterprise Connectors incluem:

- Mova, integre e analise dados com facilidade utilizando nossos conectores corporativos para FireDAC. Esses componentes incomparáveis permitem a integração de mais de 70 aplicativos corporativos, simplificando a conectividade em um modelo padrão usando SQL.
- Incluem-se componentes para QuickBooks Desktop, MailChimp, Salesforce, YouTube, SugarCRM, Jira, SurveyMonkey, Amazon DynamoDB, Couchbase, PayPal, eBay, Google Sheets, Facebook, Twitter, Slack, Dropbox e muito mais.

Para obter informações adicionais sobre os Conectores Corporativos, confira o

[Página do produto CData Enterprise Connectors](#).

The background is a low-angle, upward-looking shot of a modern glass skyscraper. The building's facade is composed of a grid of windows, reflecting the sky. The entire image is overlaid with a semi-transparent blue filter. Several white geometric shapes are scattered across the image: a small square in the upper right, a larger square below it, a horizontal line with a vertical drop on the right side, a vertical line with a horizontal drop on the left side, and three small squares in the lower left area.

Produtividade e Performance no Delphi

7 Produtividade e Performance no Delphi

Uma pesquisa global de 2018 da base de clientes do RAD Studio da Embarcadero incluiu feedback altamente positivo para suas ferramentas Delphi e C++ Builder. Das mais de 1.200 organizações, 89% obtiveram melhoria na redução dos ciclos de desenvolvimento usando o RAD Studio IDE.

Conduzida pela [TechValidate](#), uma organização de pesquisa independente, a pesquisa capturou feedback de pequenas a grandes organizações em manufatura industrial, software/serviços de computador, governança, eletrônica, saúde, educação, serviços financeiros e aeroespacial e defesa, entre outras categorias.

De acordo com o TechValidate, os principais desafios que os clientes resolveram com o RAD Studio ao desenvolver aplicativos incluíram criar uma aplicação de desempenho rápido (50%), acompanhar as últimas plataformas e dispositivos (50%), cumprir prazos de projeto (37%), implantar em várias plataformas (36%) e design responsivo para vários tipos de dispositivos (27%).

Os principais recursos procurados pelos clientes que selecionaram o RAD Studio para seu IDE de aplicação incluem design visual (82%), bibliotecas de componentes abrangentes (69%), um compilador de nativo (58%) e suporte multiplataforma (55%).

Quarenta e quatro por cento dos clientes do RAD Studio usaram o Delphi e/ou o C++ Builder para escrever, compilar, empacotar e implantar 20 aplicativos ou mais, demonstrando a usabilidade, escalabilidade e agilidade das ferramentas. Os principais casos de uso incluíram empresas (60%), indústria (36%), engenharia (32%), produtividade (29%) e científico (18%) aplicações.

Os benefícios incluíram redução do custo de desenvolvimento, tempo para o mercado mais rápido, acesso a componentes pré-configurados e pré-testados, implementação de interfaces modernas do Windows 10 e a capacidade de alcançar públicos mais amplos através do suporte a plataformas adicionais.

7.1 Produtividade

Quando gestores e desenvolvedores pensam em desenvolvimento e produtividade da equipe, as seguintes características e benefícios são consideradas:

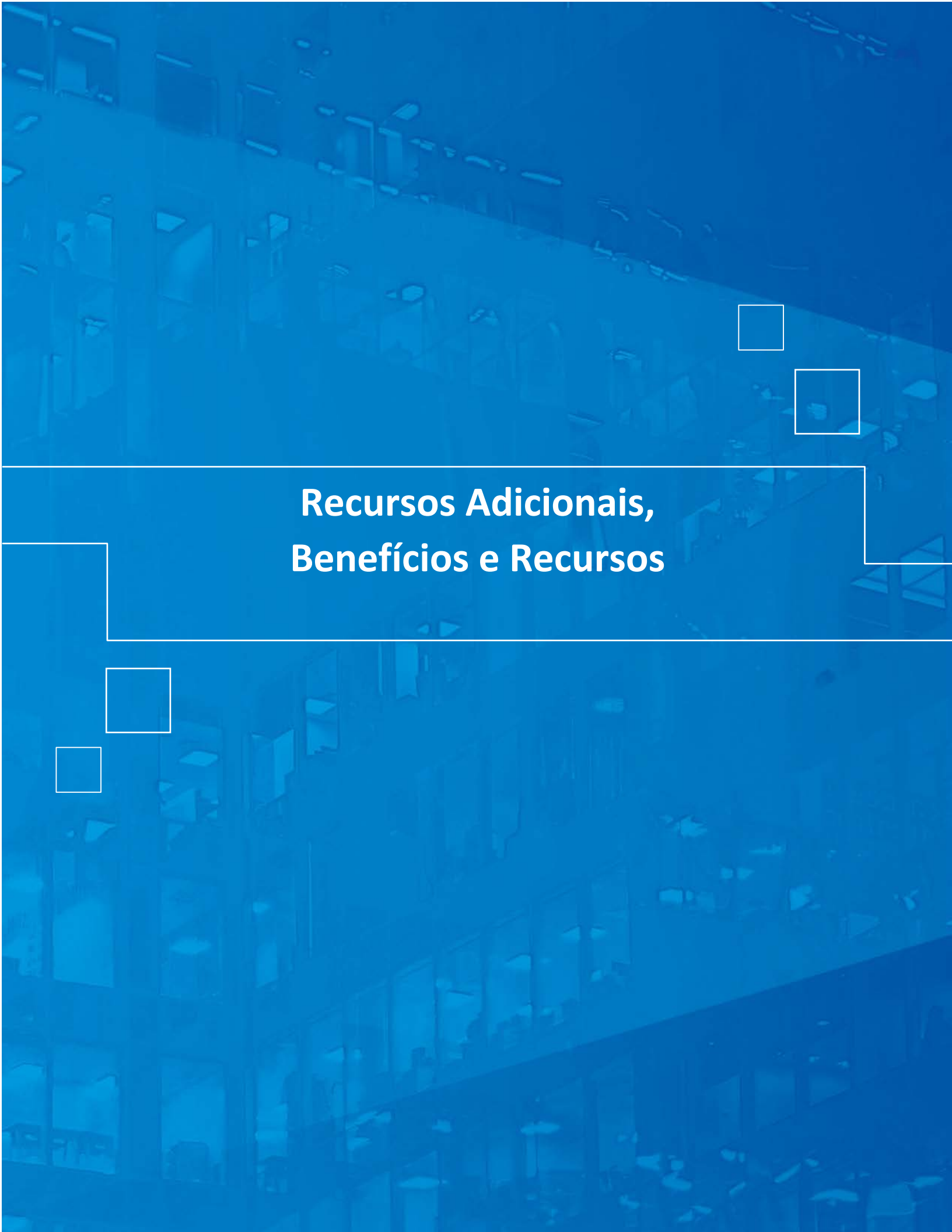
- A compatibilidade de projetos e código-fonte para múltiplas versões do Windows de 32 e 64 bits (7, 8, x e 10) e múltiplas plataformas (Windows, macOS, iOS, Android e Linux) permite que os desenvolvedores construam aplicativos para as plataformas-alvo exigidas por seus usuários e clientes atuais e futuros.
- **Os componentes** fornecem funcionalidade e extensibilidade pré-construídas que aceleram o desenvolvimento da interface do usuário e da lógica de negócios sem ter que escrever tanto código, enquanto ainda fornecem as interfaces para acessar os bastidores dos componentes e plataformas quando é necessária alguma codificação.
- O Visual LiveBindings permite que os desenvolvedores conectem visualmente componentes na hora do design sem necessidade de código.
- Ver dados ao vivo (para conexões de banco de dados e REST) em tempo de projeto acelera o desenvolvimento da interface do usuário e também garante que as instruções e conexões SQL estejam corretas.

- Os componentes e a implementação do FireDAC permitem o uso do mesmo código para múltiplos bancos de dados ao mesmo tempo em que fornece componentes adicionais para recursos específicos de cada plataforma.
- **O Code Insight** no editor acelera o desenvolvimento fornecendo conclusão de código, conclusão de parâmetros, conclusão de blocos, insight de erro, modelos de código e dicas de avaliação de expressão. Confira o tutorial do Code Insight na documentação online.
- **FMX** fornece uma única fonte, multiplataforma, multidispositivos e visualização de dispositivos para acelerar o desenvolvimento para uma ampla gama de dispositivos com diferentes tamanhos de tela e resoluções.
- Os modelos setorizados mostram aos desenvolvedores como projetar, construir e implantar aplicativos complexos. Estes modelos estão disponíveis na pasta \Documents\Embarcadero\Studio\20.0\Samples\Industry Templates\Object Pascal (nota: 20.0 é a designação para a versão do Rio 10.3. Uma pasta com numeração mais baixa será usada para versões anteriores).
- Uma ampla gama de código-fonte e exemplos estão incluídas na instalação e disponíveis on-line para acelerar o desenvolvimento de aplicações. O código-fonte do VCL, FMX e RTL para todas as plataformas está incluído para ler, aprender e também como documentação final.
- **Auditorias e Métricas** são **integradas** ao IDE e disponíveis como ferramentas de linha de comando para ajudar a impor padrões e convenções da empresa, capturar métricas reais, identificar possíveis problemas e melhorar a qualidade do código.
- O RAD Studio integra uma estrutura de **teste unitário** de código aberto para o desenvolvimento e execução de casos de teste automatizados para aplicativos. As estruturas DUnit e DUnitX estão disponíveis para uso em Delphi e C++Builder. Esses frameworks simplificam o processo de desenvolvimento de testes para classes e métodos para recursos e funcionalidades de aplicativos. O uso de testes unitários em combinação com a refatoração pode ajudar a melhorar a aplicação. Usar um conjunto padrão de testes toda vez que uma pequena mudança é feita torna mais provável que quaisquer problemas sejam pegos no início do ciclo de desenvolvimento.
- As soluções de teste automatizadas estão disponíveis para a Delphi através de parceiros de tecnologia, incluindo **TestRail**, **Ranorex** e **TestComplete**.
- Sistemas de Integração Contínua também estão disponíveis para Delphi e são fáceis de usar porque o formato do arquivo de projeto utiliza o padrão MS Build. As soluções de CI incluem:
 - Travis CI - <https://travis-ci.com/>
 - FinalBuilder – der.com/finalbuilder <https://www.finalbuilder.com/>
 - CONTINUA CI – <https://www.finalbuilder.com/continua-ci>
 - Jenkins CI- <http://thundaxsoftware.blogspot.com/2011/07/continuous-integration-for-delphi.html>
 - MSBuild - <https://docs.microsoft.com/en-us/visualstudio/msbuild/msbuild?view=vs-2019>
 - Amazon Code Pipeline - <https://aws.amazon.com/codepipeline/>
 - Controle de Cruzeiros – <http://cruisecontrol.sourceforge.net/>

7.2 Performance

O código nativo será sempre mais rápido do que o bytecode ou o código interpretado. Alguns desenvolvedores usam JavaScript, Python, Java e outras linguagens não compiladas, que muitas vezes sofrem com tempo de inicializar mais lento e múltiplas dependências de framework. Aqui estão alguns dos benefícios do uso de compiladores de código nativo, componentes pré-compilados e bibliotecas de tempo de execução para criar aplicativos pequenos e rápidos. Ferramentas adicionais também são listadas, as quais podem ajudar a melhorar o desempenho do aplicativo:

- Todos os compiladores otimizados do Delphi produzem aplicativos nativos de código de máquina para processadores Intel e ARM (dependendo da plataforma). Os componentes e bibliotecas de tempo de execução também são compilados para o código da máquina para inicialização e execução rápida.
- O compilador Delphi permite que funções de um procedimento sejam marcadas com a **diretiva inline** para melhorar o desempenho. Se a função ou procedimento atender a determinados critérios, o compilador inserirá o código diretamente, em vez de gerar uma chamada. Inlining é uma otimização de desempenho que pode resultar em código mais rápido.
- Considere usar a Parallel Programming Library (PPL) em partes da lógica do aplicativo para aproveitar mais núcleos de processador e também proporcionar uma interface de usuário mais responsiva.
- O VCL oferece uma integração forte com APIs e recursos do Windows.
- O Delphi pode ser usado para suportar o processamento de cargas de trabalho em infraestruturas multicamada e em nuvem.
- O acesso ao banco de dados FireDAC é otimizado internamente para cada banco de dados suportado e inclui features adicionais para aproveitar o acesso e as operações do banco de dados. O FireDAC também inclui suporte para recursos de cache e monitoramento que permitem aos desenvolvedores garantir que eles estão aproveitando o melhor desempenho de cada banco de dados.
- Ferramentas estão disponíveis para criar um profile do código, encontrar os pontos de acesso e identificar a cobertura do código para otimizar ainda mais os aplicativos. O uso de um profile pode resolver problemas de execução, estrutura de dados e algoritmos antes que seja tarde demais. As soluções de criação de profile para Delphi incluem:
 - **Smarbear AQTime** também inclui uma extensão da IDE para melhor integração com o RAD Studio).
 - **O registro do Raize Software CodeSite** para obter informações mais profundas sobre a execução do aplicativo.
 - **EurekaLog** que rastreia exceções, vazamentos de memória e muito mais.
 - **ProDelphi32 e ProDelphi64** fornecem profile para aplicativos Delphi de 32 bits e 64 bits. Assista a uma demonstração no YouTube em https://www.youtube.com/watch?v=cJ_G8XUWSOI.
 - **O DeLeaker** fornece profile para aplicativos Delphi. Uma demonstração está disponível em <https://www.youtube.com/watch?v=rnq44Td9opQ>.
- Os desenvolvedores, que desejam o melhor desempenho de aplicativos compilados nativamente e também precisam usar Python para ciência de dados ou outros requisitos, devem considerar o uso dos componentes livres como **Python4Delphi** que integram o Python em aplicativos Delphi.



Recursos Adicionais, Benefícios e Recursos

8 Recursos Adicionais, Benefícios e Recursos

Esta seção cobre uma lista de recursos e benefícios adicionais do Delphi que ajudarão os desenvolvedores a criar aplicativos Windows e multiplataforma. Incluídos nesta seção estão links para recursos adicionais que estão disponíveis para aumentar a produtividade do desenvolvedor.

8.1 Multiplataforma

O Delphi inclui compiladores de código nativo otimizados e o suporte nativo à API/Dispositivos permite que os mesmos recursos de linguagem do Object-Pascal sejam suportados em todas as plataformas— Win32, Win64, iOS, macOS, Android e Linux.

Os aplicativos VCL podem usar o mesmo código, componentes e bibliotecas nas versões Win32 e Win64 do Windows testadas para Windows 7, Windows 8.x e Windows 10.

Os aplicativos FMX podem usar o mesmo código, componentes e bibliotecas no Win32, Win64, iOS, macOS e Android. Os aplicativos de console, servidor e GUI também podem ser construídos para Linux usando o Delphi.

Suportar a linguagem, componentes e bibliotecas enquanto ainda fornece interfaces para as APIs e plataformas subjacentes aumenta a produtividade dos desenvolvedores, ao mesmo tempo em que oferece suporte de distribuição e compatibilidade de interface de usuário em cada plataforma nativa.

8.2 Virtualização e Distribuição de Aplicativos Delphi

Alguns usuários Delphi estão optando por implantar seus aplicativos desktop em servidores Windows ou plataformas em nuvem, em vez de instalar os aplicativos em vários desktops dos usuários. Esta seção abrange várias opções para os desenvolvedores considerarem.

8.2.1 Cybele Software Thinfinity® VirtualUI

A Cybele Software, parceira de tecnologia Embarcadero, oferece a capacidade de virtualizar aplicativos Delphi adicionando uma linha de código e uma biblioteca de tempo de execução para criar e implantar facilmente aplicativos em servidores Windows e permitir que vários usuários usem navegadores para operar estes aplicativos.

O Thinfinity® VirtualUI, da Cybele Software, permite que os aplicativos de desktop baseados no Windows funcionem como aplicativos web entre navegadores e dispositivos multiplataforma. Os aplicativos baseados no Thinfinity VirtualUI podem ser executados em um ambiente Windows ou serem acessados remotamente a partir de qualquer navegador Web HTML5. Desde que a VirtualUI foi lançada em 2015, ela ajudou milhares de organizações a abraçar a transformação digital, ampliando o ciclo de vida dos aplicativos críticos para os negócios, com custo efetivo e confiabilidade.

Adicionar uma única linha de código a um aplicativo Delphi permite a funcionalidade na dupla plataforma. Os aplicativos Delphi podem estar em funcionamento em um navegador literalmente em poucos minutos. Todos os navegadores HTML5 são cobertos, incluindo Internet Explorer, Edge, Firefox, Google Chrome e Safari.

Para obter mais informações, acesse o site do produto Cybele Software usando as seguintes URLs:

- Product page
<https://www.cybelesoft.com/thinfinity/virtualui/>
- Platform requirements
<https://www.cybelesoft.com/thinfinity/virtualui/requirements/>
- Delphi information page
<https://www.cybelesoft.com/delphi-web/>
- Delphi 5 minute tutorial YouTube video
<https://youtu.be/13ZANj8sUJE>

8.2.2 Amazon AppStream 2.0

Amazon AppStream 2.0 é um serviço de streaming de aplicativos totalmente gerenciado. As empresas podem gerenciar centralmente seus aplicativos de desktop no AppStream 2.0 e entregá-los com segurança a qualquer computador. O AppStream pode facilmente dimensionar aplicativos para qualquer número de usuários em todo o mundo sem adquirir, provisionar e operar hardware ou infraestrutura. AppStream 2.0 é construído sobre o AWS, para que os desenvolvedores se beneficiem de um data center e arquitetura de rede projetado para as organizações mais sensíveis à segurança. Cada usuário tem uma experiência de aplicação fluida e responsiva, incluindo projetos 3D com uso intensivo de GPU e os de engenharia, porque os aplicativos são executados em máquinas virtuais (VMs) otimizadas para casos específicos de uso e cada sessão de streaming se ajusta automaticamente às condições da rede.

As empresas podem usar o AppStream 2.0 para simplificar a entrega de aplicativos e concluir sua migração para a nuvem. As instituições de ensino podem fornecer a todos os alunos acesso aos aplicativos necessários para a aula em qualquer computador. Os fornecedores de software podem usar o AppStream 2.0 para fornecer testes, demonstrações e treinamento para seus aplicativos sem downloads ou instalações. Eles também podem desenvolver uma solução completa de software como serviço (SaaS) sem reescrever seu aplicativo.

8.2.3 O Gerenciador de Implantação do IDE

O RAD Studio **Deployment Manager** oferece a capacidade de implantar aplicativos Windows e multidispositivos a partir do IDE para qualquer uma das plataformas suportadas. Para Windows, macOS e iOS, o Deployment Manager da IDE funciona com o PAServer (os arquivos de instalação do PAServer estão localizados na pasta \Embarcadero\Studio\20.0\PAServer. O aplicativo que está sendo implantado em um dispositivo de destino deve usar o PAServer e um perfil de conexão. Para Android, os desenvolvedores só precisam adicionar um SDK Android ao RAD Studio para implantar aplicativos para um dispositivo.

O Deployment Manager permite que os desenvolvedores habilitem, visualizem, adicionem, excluam ou editem os arquivos que estão sendo implantados. Os desenvolvedores também podem usar o Deployment Manager para adicionar outros arquivos necessários à implantação, como drivers de banco de dados para a plataforma destino.

8.2.4 Ferramentas de Instalação de Terceiros

Depois que um aplicativo é criado, os desenvolvedores podem escolher entre uma série de ferramentas que auxiliam na criação de instalações personalizadas. Aqui está uma lista de algumas das ferramentas de instalação disponíveis para o RAD Studio:

- InstallAware
<https://www.installaware.com/>
- Advanced Installer
<https://www.advancedinstaller.com/>
- InstallAnywhere
<https://www.flexerasoftware.com/install/products/installanywhere.html>
- Inno Setup
<http://www.jrsoftware.org/isinfo.php>
- Wix Toolset
<https://wixtoolset.org/>
- InstallShield
<https://www.flexerasoftware.com/install/products/installshield.html>
- NSIS
https://nsis.sourceforge.io/Main_Page
- InstallForge
<https://installforge.net/home/>

8.2.5 Submetendo para uma Loja Online

Os desenvolvedores podem empacotar e implantar aplicativos para Windows, macOS, iOS e Android em suas várias lojas online. Aqui estão os links do DocWiki que definem as etapas necessárias para preparar, empacotar e enviar aplicativos para cada uma das lojas virtuais:

- [Submitting Your App to the WindowsStore](#)
- [Submitting Your App to the App Store\(iOS\)](#)
- [Submitting Your Android App to GooglePlay](#)
- [Submitting Your App to the Mac App Store](#)
- [Submitting Your Android App to theAmazon Appstore](#)

8.3 Ecossistema de Parceiros de Tecnologia

Desde o início, a equipe do Delphi sabia que não seria capaz de construir todos os componentes, bibliotecas, ferramentas, experts, assistentes, modelos, documentos, tutoriais, livros, etc. É por isso que o Delphi foi projetada para fornecer extensibilidade no IDE, biblioteca de componentes e muito mais. Ao longo dos anos, um grande ecossistema global de parceiros de tecnologia foi criado e apoiado para fornecer uma ampla gama de soluções adicionais de desenvolvimento. Aqui estão apenas alguns dos muitos links para parceiros de tecnologia e repositórios contendo soluções adicionais, componentes e tecnologias:

- **A TMS Software**, fundada em 1995, é uma empresa de desenvolvimento de software especializada em componentes, frameworks, ferramentas, desenvolvimento de projetos personalizados, consultorias e treinamentos. Alguns da família de produtos TMS para Delphi incluem:
 - Componentes **VCL**, FMX, FNC, Web e IntraWeb para toolbars, grids, menus, instrumentação, nuvem, gráficos, mapas, sincronização e muito mais.
 - **TMS Aurelius** é um framework ORM para Delphi com suporte total para manipulação de dados, consultas complexas e avançadas, herança, polimorfismo e muito mais.
 - **O TMS Data Modeler** é a melhor ferramenta para modelar bancos de dados com uma interface fácil e simples. Ele fornece recursos poderosos, incluindo diagramação, engenharia reversa, geração de script SQL e controle de versão. Possui integração com ferramentas de desenvolvimento como Delphi e C++ Builder.
 - **TMS Echo** é um framework Delphi para replicação de dados.
 - **TMSFixInsight** é um analisador de código para desenvolvedores Delphi que detecta problemas diretamente no código-fonte de projetos.
- **Woll2Woll**, fundada em 1993, ganhou sua reputação por fornecer complementos sólidos e profissionais para o RAD Studio com seus premiados conjuntos de componentes, InfoPower, 1stClass e FirePower. Estes conjuntos de componentes visuais foram projetados especificamente para dar aos desenvolvedores profissionais de banco de dados um poder inigualável em seus aplicativos móveis e desktop Delphi e C++ Builder
 - **InfoPower** para Delphi VCL é o conjunto mais popular de componentes para construir interfaces de aplicativos de banco de dados.
 - **O Firepower** tem enormes vantagens pois vem com os componentes mais intuitivos e poderosos do RAD Studio FireMonkey.
- **Steema Software**, TeeChart tornou-se líder indiscutível em Delphi, renomado pela indústria e ganhador da melhor ferramenta Delphi para Charting and Mapping do Informant Group por sete anos consecutivos. Ainda é a única ferramenta de gráficos VCL nativa que foi escolhida pela Embarcadero (anteriormente Borland) para ser agrupada ao IDE Delphi e C++ Builder.

- **TeeChart Pro** biblioteca de componentes de gráficos para VCL e FMX oferece centenas de estilos gráficos em 2D e 3D para visualização de dados, matemáticos, estatísticos e funções financeiras, juntamente com um número ilimitado de eixos e 30 componentes na paleta.
- **Componente TeeGrid** para Delphi VCL & FMX oferece uma grid muito rápido para projetos RAD Studio.
- A **FastReport** desenvolve componentes de geração de relatórios e cubos de decisão para Delphi VCL e FMX. Uma versão de desenvolvedor do FastReport está incluída no Delphi.
 - FastReport é um componente para gerar relatórios de forma rápida e eficiente. O FastReport fornece todas as ferramentas necessárias para o desenvolvimento de relatórios, incluindo um designer de relatório visual, classes e componentes base e uma janela de preview. As versões Delphi estão disponíveis para **VCL** e FMX.
 - FastCube é um conjunto de componentes OLAP que permite aos desenvolvedores analisar dados e construir tabelas de resumo (fatias de dados), bem como criar uma variedade de relatórios e gráficos, de forma fácil e instantânea. É uma ferramenta útil para a análise eficiente de matrizes de dados. As versões Delphi estão disponíveis para **VCL** e FMX.
- **IPWorks** é o pacote mais vendido e abrangente de componentes programáveis da Internet em todo o mundo. O IPWorks elimina grande parte da complexidade do desenvolvimento de aplicativos conectados, fornecendo componentes programáveis e fáceis de usar que facilitam tarefas como enviar e-mails, transferir arquivos, navegar na Web, consumir serviços da Web XML, e muito mais.
 - A edição de desenvolvedores para Delphi está disponível no GetIt Package Manager.
 - Conjunto completo de **componentes Delphi VCL** nativos com criptografia, BluetoothLE, WebSockets, IPC, MQ, IoT, suporte OFX e muito mais.
- **O Atozed Software** (pronuncia-se A-to-zed) desenvolve o conjunto IntraWeb de componentes que permite aos desenvolvedores criar projetos como um aplicativo desktop, mas implantar na Web como HTML padrão e JavaScript (sem plugins necessários).
- **Scooter Software** Beyond Compare permite que os desenvolvedores comparem rapidamente e facilmente arquivos e pastas. Usando comandos simples e poderosos, os desenvolvedores podem se concentrar nas diferenças em que estão interessados e ignorar o restante. Os desenvolvedores podem então mesclar as alterações, sincronizar arquivos e gerar relatórios.
- **A Raize Software** cria softwares inovadores e de alta qualidade desde 1995. Os controles VCL Konopka Signature é um conjunto de mais de 200 controles Windows e que agora está incluído no pacote de bônus do RAD Studio e disponível para download no GetIt Package Manager. A Raize Software também fornece o sistema de log CodeSite que fornece aos desenvolvedores uma visão mais profunda sobre como seu código está sendo executado, permitindo que eles localizem problemas de forma mais rápida e monitorem o desempenho do código.
- **Project Indy** é um projeto de código aberto que fornece componentes da Internet para aplicações VCL e FMX. A Indy oferece componentes de cliente e servidor para o uso de protocolos de internet, como TCP, UDP, Echo, FTP, HTTP, Telnet e muitos outros. A Indy também fornece componentes para manuseio de I/O, interceptações, SASL, UUE, MIME, codificadores XXE, entre outros. Os componentes Indy são pré-instalados a cada versão do Delphi.

- **Project JEDI** (Joint Endeavor of Delphi Innovators) é um projeto de código aberto que fornece componentes VCL, bibliotecas de tempo de execução, ferramentas e unidades de interface da API do Windows.
 - A **JEDI Visual Component Library (JVCL)** consiste em uma grande coleção (atualmente com 600+) de componentes visuais e não visuais que podem ser reutilizados em projetos Delphi e C++ Builder. A biblioteca JEDI VCL está disponível para instalação no GetIt Package Manager.
 - A **JEDI Code Library (JCL)** consiste em um conjunto de funções e utilidades completamente testadas e documentadas de classes não visuais que podem ser reutilizadas em projetos Delphi e C++ Builder.
- **Torry's Delphi Pages** contêm links para dezenas de milhares de fornecedores, componentes, bibliotecas, livros, exemplos e muito mais.

8.4 Educação – Aprenda, Receba Ajuda, Encontre Exemplos

Existem muitos recursos e sites disponíveis para aprender Delphi, descobrir como resolver um problema, como criar um aplicativo e muito mais. Aqui estão apenas alguns (dos muitos) sites que fornecem conhecimento do Delphi e ajudam a aumentar a produtividade do desenvolvimento.

8.4.1 LearnDelphi.org

LearnDelphi.org é o lugar para aprender a codificar com o Object Pascal Delphi. Pascal foi projetado para ensinar boas práticas de programação, e o Delphi leva o melhor de Pascal e agrega todos os recursos e poder necessários para enfrentar as aplicações mais exigentes em todas as plataformas. Este site está cheio de recursos e conteúdo gratuito para ensinar e aprender desenvolvimento de software.

8.4.2 Embarcadero Academy

O **Embarcadero Academy** oferece cursos no formato EAD e ao vivo diretamente de alguns dos desenvolvedores mais experientes do setor. A academia não discrimina linguagens, ferramentas e frameworks de programação. Atualmente, os desenvolvedores encontrarão cursos na Embarcadero Academy que se concentram principalmente em linguagens e habilidades em produtos Embarcadero como RAD Studio, Delphi, C++Builder, InterBase e Sencha. E enquanto a maior parte do nosso conteúdo é orientado para Delphi e C++, à medida que o tempo passa desenvolvedores encontrarão mais cursos e palestras sobre uma ampla variedade de habilidades de programação e linguagens. Os cursos são vendidos individualmente, embora haja também pacotes de cursos que combinam vários cursos em um único pacote. A maioria dos cursos são centrados em torno de palestras em streaming.

O Embarcadero Academy também posta bootcamps gratuito e replays de conferências. Além disso, os instrutores podem optar por oferecer cupons de desconto para seus cursos. Se oferecido por um instrutor, os alunos podem obter seu código de cupom diretamente do instrutor através de blogs, redes sociais ou seus sites, e de tempos em tempos as ofertas também serão postadas aqui em nossa página de [Deals](#).

8.4.3 Developer Skill Sprints

Desenvolvedor Skill Sprints são dicas rápidas de programação, truques e técnicas que os desenvolvedores podem usar agora! Esses vídeos online (20 minutos ou menos) contêm tutoriais (gravados ao vivo) sobre desenvolvimento de software para Windows, macOS, Android e iOS.

<http://embt.co/skill-sprints-youtube>

8.4.4 Webinar Replays e outros Vídeos no YouTube

Embarcadero tem centenas de replays de webinars, vídeos de lançamento de produtos e uma ampla gama de vídeos educacionais e informativos categorizados em várias listas de reprodução.

<https://www.youtube.com/user/EmbarcaderoTechNet/playlists>

8.4.5 How-To Videos

Os vídeos de How-To contêm tutoriais curtos focados em aprender como usar recursos e tecnologias específicas para Embarcadero RAD Studio, Delphi, C++Builder e InterBase.

<https://embt.co/TutorialVideos>

8.4.6 Embarcadero Developers Blog

Os gerentes de produto da Embarcadero e os líderes da comunidade mantêm blogs sobre uma ampla gama de tópicos de desenvolvimento Delphi no blog de desenvolvedores Embarcadero.

<https://community.idera.com/developer-tools/b/blog>

8.4.7 Embarcadero nas Redes Sociais

Os membros da equipe Embarcadero são muito ativos nas redes sociais. Aqui estão os links para acompanhar as notícias, dicas, eventos e muito mais:

- Twitter
<https://twitter.com/EmbarcaderoTech>
- Facebook
<https://www.facebook.com/embarcaderotech>
- LinkedIn
<https://www.linkedin.com/company/7548>

8.4.8 Feito com Delphi

Os desenvolvedores estão sempre perguntando quem está usando Delphi e que tipos de aplicações eles estão construindo. A seguir, alguns sites que catalogam aplicativos "Build with Delphi".

- Good Quality Applications Built With Delphi
https://delphi.fandom.com/wiki/Good_Quality_Applications_Built_With_Delphi
- Famous software made with Delphi
<https://jonlennartaasenden.wordpress.com/2014/11/06/famous-software-made-with-delphi/>
- Awesome Applications Built in Delphi
<https://beyondvelocity.blog/2019/05/19/awesome-applications-built-in-delphi/>
<https://beyondvelocity.blog/2019/07/28/delphi-mobile-its-all-components/>
- Cool Apps Winners – YouTube
[https://www.youtube.com/playlist?list= PLwUPJvR9mZHhHEPxc8i9RCNX4jTyBZOfw](https://www.youtube.com/playlist?list=PLwUPJvR9mZHhHEPxc8i9RCNX4jTyBZOfw)

8.4.9 Exemplos de Código, Tutoriais e Exemplos

Existem numerosos exemplos de código Delphi no wiki de documentação online. Os desenvolvedores também encontrarão muitos projetos de exemplo na pasta de instalação em \Documentos\Embarcadero\ Studio\ 20.0\ Samples (nota: 20.0 é a designação da versão 10.3 Rio; uma pasta numerada menor será usada para versões anteriores)

http://docwiki.embarcadero.com/CodeExamples/en/Code_Examples_Index

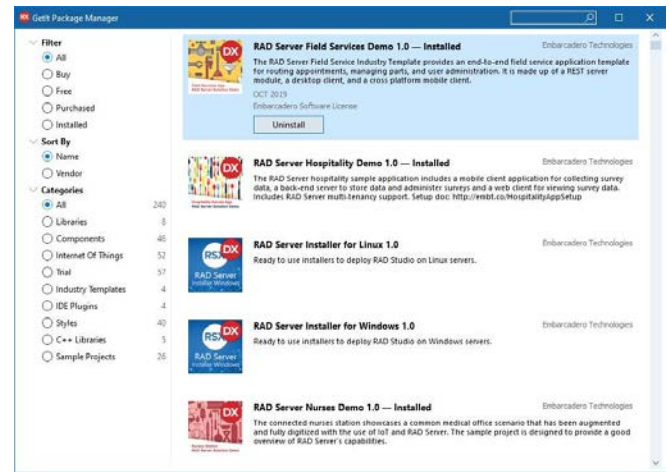
8.4.10 RAD Studio IDE Tools API

Os desenvolvedores podem estender e personalizar o RAD Studio IDE com itens de menus, botões de barra de ferramentas, assistentes dinâmicos de criação de formulários, novos componentes e muito mais usando a Tools API. A **Tools API** é um conjunto de mais de 100 interfaces que interagem e controlam o IDE, incluindo o menu principal, as barras de ferramentas, a lista de ação principal e a lista de imagens, os buffers internos do editor, macros e vinculações do teclado, formulários e seus componentes no editor, o depurador e o processo que está sendo depurado, a conclusão do código, a exibição de mensagens e a lista de to-do. Procurando informações adicionais sobre a Tools API do IDE? Tem a necessidade de adicionar funcionalidade para aumentar a produtividade da equipe? Quer aprender como criar extensões da IDE, experts, modelos e assistentes? Confira os seguintes sites que contêm informações adicionais sobre estas APIs:

- <http://www.gexperts.org/open-tools-api-faq/>
- <https://www.embarcadero.com/creating-ide-extensions-with-the-delphi-open-tools-api>
- https://www.davidghoyle.co.uk/WordPress/?page_id=1110

8.4.11 GetIt Package Manager

O GetIt Package Manager permite que os desenvolvedores naveguem, baixem e instalem pacotes no RAD Studio. Os pacotes podem fornecer bibliotecas, componentes, extensões da IDE e SDKs. O GetIt está disponível dentro do IDE usando o menu Tools | GetIt Package Manager.



8.4.12 Repositórios do GitHub para Delphi

<https://github.com/search?q=delphi>

Embarcadero, parceiros de tecnologia e membros da comunidade hospedam projetos no GitHub. Confira os mais de 9.600 resultados do repositório contendo soluções, componentes, projetos e outras informações.

8.4.13 Respostas no StackOverflow

Use a tag Delphi para encontrar as perguntas e respostas mais recentes no StackOverflow. Membros da equipe Delphi, MVPs e membros da comunidade estão respondendo perguntas de desenvolvedores em tempo real.

<https://stackoverflow.com/questions/tagged/delphi>

8.4.14 Página de Status de Plataforma

Não tem certeza se uma plataforma é suportada para uma versão específica do Delphi? Existem problemas conhecidos e soluções para uma plataforma ou versão específica? A Embarcadero tem uma página no DocWiki online que mostra as plataformas suportadas e sistemas operacionais para diferentes versões do RAD Studio. Um sinal * (estrela) ao lado de um sistema operacional indica que há um problema conhecido com esse sistema operacional e uma versão correspondente do RAD Studio. Para ver a solução para esse problema em particular, clique no nome do sistema operacional ou role até a seção apropriada.

http://docwiki.embarcadero.com/PlatformStatus/en/Main_Page

RAD Studio Version	Windows	macOS	iOS	Android	Linux
10.3 Rio	<ul style="list-style-type: none">Windows 10Windows 8.1Windows 7 (SP1+)Windows Server 2016Windows Server 2012	<ul style="list-style-type: none">macOS Catalina^[1]macOS MojavemacOS High SierramacOS Sierra	<ul style="list-style-type: none">iOS 13^[2]iOS 12iOS 11iOS 10, including iOS 10.3	<ul style="list-style-type: none">Android 10^[3]Android 9Android 8Android 7Android 6Android 5.1	<ul style="list-style-type: none">Ubuntu 18.04 LTSUbuntu 16.04 LTSUbuntu 14.04 LTSRedHat Enterprise Linux (version 7)



Vídeos sobre Windows 10 e VCL

9 Vídeos sobre Windows 10 e VCL

Existem centenas de vídeos focados em usar o Delphi para criar aplicativos Windows, modernizar aplicativos com foco em Windows 10, aprender Delphi e uma ampla gama de habilidades de desenvolvimento Windows.

9.1 Modernização de Aplicativos para Windows 10

- Top Windows 10 Features for Your VCL Applications –
<https://www.youtube.com/watch?v=7KuzjODgR3c>
- Building Native Apps for Windows: Why It Matters, with Serge Pilko (Embarcadero MVP)
<https://www.youtube.com/watch?v=4SL01nQbYKM>
- Move Your Windows Apps Forward from Windows 7 to Windows 10 with Microsoft and Embarcadero
<https://www.youtube.com/watch?v=3xKMGBUmwUs>
- VCL Integration with WinAPI, COM & ShellAPI, and WinRT
<https://www.youtube.com/watch?v=mM6sN6KS12U>
- Fast Migration to Windows 10 High DPI, with Denis Zubov
<https://www.youtube.com/watch?v=niCFI8kgoOY>
- New UX Design Principles for RAD Studio Developers in Windows 10, with Sarina DuPont
<https://www.youtube.com/watch?v=Qx71ytekamU>
- Working with Windows 10's Windows Runtime (WinRT)
<https://www.youtube.com/watch?v=bQI9GQB5y54>
- Windows 10 – Modernization with RAD Studio and TMS Software
https://www.youtube.com/watch?v=sU0x3sn-m_o
- FireDAC for Windows 10 Database Access
<https://www.youtube.com/watch?v=rq16Lc1ZZOY>
- Moving Your App to the Microsoft Store, with Olaf Monien
<https://www.youtube.com/watch?v=ODVHne9GOFY>
- Windows Subsystem for Linux (WSL) with Delphi
<https://www.youtube.com/watch?v=AD9XG4Y7MwA>
- High DPI User Interface Design with the VCL
<https://www.youtube.com/watch?v=Q5mu8yigcK4>
- Windows 10 VCL
<https://www.youtube.com/watch?v=LbAEWketmMg>
- Leverage Windows 10 APIs to Attract More Users –
https://www.youtube.com/watch?v=lvbQ9Hmfu_4

9.2 Delphi IDE Características e Produtividade

- Creating your First VCL Application for Windows (Delphi)
<https://www.youtube.com/watch?v=7oBTZ16dbc4>
- LiveBinding for VCL Developers
<https://www.youtube.com/watch?v=SVpbY4et2ug>
- Using LiveBinding in VCL Applications
https://www.youtube.com/watch?v=i9_6CJi5EIM
- Working with VCL Styles to Create Great Looking Windows Applications
<https://www.youtube.com/watch?v=m3RyP6AW9y0>
- Cool Tricks for VCL Styles
<https://www.youtube.com/watch?v=c6E3DZrEdtQ>
- VCL – the best bits
<https://www.youtube.com/watch?v=tH61SOVBfW8>





25 anos de Inovações Delphi



10 25 anos de Inovações Delphi

As seções a seguir fornecem uma rápida olhada em alguns dos principais recursos introduzidos e aprimorados nos últimos 25 anos de releases do Delphi. Você pode dar uma olhada na história do Delphi, nas memórias de desenvolvimento, nas histórias dos desenvolvedores sobre o uso do Delphi e muito mais em <https://delphi.embarcadero.com/>

Delphi 1 – February 15, 1995

- Native Code optimizing compiler
- Integrated Debugging
- 16-bit Windows 3.1 support
- Visual Two-Way tools
- Components / VCL
- Database support via BDE and SQL Links
- Database data live at design time

Delphi 2 – February 10, 1996

- 32-bit Windows 95 support
- Database Grid
- OLE automation
- Visual form inheritance
- Long strings
- Included Delphi 1 for 16-bit development

Delphi 3 – August 5, 1997

- Interfaces (COM based)
- Code Insight
- Component Templates
- DLL Debugging
- WebBroker
- ActiveForms
- Component Packages
- MIDAS multi-tier architecture

Delphi 4 – June 17, 1998

- Docking, Anchors and Constraints
- Method Overloading
- Dynamic arrays
- Windows 98 support

Delphi 5 – August 10, 1999

- Desktop layouts
- Frames
- XML support
- DBGo for ADO
- UI Language Translations

Delphi 6 – May 21, 2001

- Structure Window
- SOAP Web Services
- dbExpress
- BizSnap
- WebSnap
- DataSnap

Delphi 7 – August 9, 2002

- RAD Web application development
- Web Services
- UML Designer
- Model Driven Development
- .NET interoperability
- Windows XP themes

Delphi 8 – December 22, 2003

- .NET support

Delphi 2005 – October 12, 2004

- Multi-unit namespaces
- Error Insight
- History Tab
- for..in
- Function inlining
- Theme-enabled IDE
- Refactorings
- Wild-card in uses statement
- Data Explorer
- Integrated Unit Testing

Delphi 2006 – November 23, 2005

- Operator overloading
- Static methods and properties
- Designer Guidelines
- Form positioner view
- Live code templates
- Block Completion

- Line numbers
- Change Bars
- Sync-edit
- Code Folding and method navigation
- Debugging Tool-Tips
- Searchable Tool Palette
- FastMM memory manager
- Support for MySQL
- Unicode support in dbExpress
- TTrayIcon
- TFlowPanel
- TGridPanel

Delphi 2007 – March 16, 2007

- MS Build, Build Events, Build Configurations
- Windows Vista support – glassing, theming
- dbExpress 4 – connection pooling, delegate drivers
- CPU viewer windows
- FastCode enhancements
- IntraWeb / AJAX support
- Welcome Page
- Simultaneous-ship of English, French, German, Japanese editions

Delphi 2009 – August 25, 2008

- Unicode
- Generics
- Anonymous Methods
- Ribbon Controls
- DataSnap
- Build Configurations
- Class Explorer
- Type Library Editor Window
- PNG support

Delphi 2010 – August 15, 2009

- IDE Insight
- Attributes
- Enhanced RTTI
- Direct2D canvas
- Windows 7 support
- Touch/Gestures
- Source Code Formatter
- Incremental search
- Thread Specific Breakpoints
- Debugger Visualizers
- IOUtils unit for files, paths and directories
- Source Code Audits and Metrics
- Background Compilation
- Source code for MIDAS.DLL

Delphi XE – August 30, 2010

- Regular Expression Library
- Version Insight – Subversion Integration
- dbExpress – Filters, Authentication, Proxy Generation, JavaScript Framework
- REST support
- Indy WebBroker
- Cloud – Amazon EC2, Microsoft Azure
- Build Groups
- Named Threads in the Debugger
- Command line Audits, Metrics
- Documentation Generation
- Toolset and Component Additions:
- AQTime, Beyond Compare
- CodeSite, Documentation Insight
- FastReport, FinalBuilder, IP*Works!

Delphi XE2 – September 2, 2011

- 64-Bit Windows compiler
- Mac OSX
- FireMonkey (FMX)
- Live Bindings – FMX and VCL
- VCL Styles
- Unit Scope Names
- Platform Assistant
- DataSnap – Connectors for Mobile Devices, Cloud API, HTTPS support, TCP monitoring
- dbExpress support for ODBC drivers
- Deployment Manager

Delphi XE3 – September 3, 2012

- Metropolis UI for Windows 8, 7, Vista and XP
- OS X Mountain Lion
- Visual Live Bindings
- FMX 2:
 - actions, touch/gestures, layouts and anchors
- bitmap styles
- TMaterial source for FM 3D components
- audio/video
- Location sensor component
- VCL/FMX support for sensor devices
- Virtual keyboard support
- DirectX 10 support

Delphi XE4 – April 22, 2013

- FireDAC universal data access components
- Notifications
- Platform Services
- Location, Motion and Orientation sensor components
- TListView component
- Mobile form designer

- TWebBrowser component
- iOS support
- Mac OSX full screen support
- InterBase – IBLite and IBToGo

Delphi XE5 – September 11, 2013

- Android support
- Notification component
- iOS 7 style support
- Configurable form designer for mobile devices
- Deployment manager for Android devices
- REST Services client access and authentication components

Delphi XE6 – April 15, 2014

- App Tethering Components
- Cloud Service (BaaS) RESTful WEB Services integration (Parse, Kinvey, App42)
- Taskbar component
- New VCL Styles, VCL sensor components
- Mobile In-app purchase and in-app advertising
- Create apps for Google Glass
- Accessibility for FireMonkey desktop
- Database Explorer for FireDAC
- Apache support (WebBroker)
- DirectX 11 support
- Windows 7 and 8.1 styles
- Acces to Cloud base RESTful WEB Services
- FireDAC Compatible with more databases
- Fully integrated InterBase support

Delphi XE7 – September 2, 2014

- Multi-Device Designer – standard and custom views
- FireMonkey Multi-Device Applications Support Both Desktop and Mobile Platforms
- FireMonkey Behavior Service
- Parallel Programming Library (PPL)
- IBLite embeddable database for Windows, Mac, Android and iOS

- Multi-Display Support
- TMultiView component
- Multi-Touch Support and Gestures Changes
- Full-Screen Immersive Mode for Android
- FMX Pull-to-Refresh Feature for TListView on iOS and Android
- FMX Save State Feature
- Interbase IBLite for Windows and OSX
- Enterprise Mobility Services (RAD Server)
- Bluetooth and BluetoothLE run time library
- TBluetoothLE component
- App Tethering – Bluetooth and IP Address
- OmniXML Library
- Java2Pas interfaces converter
- Git version control support

Delphi XE8 – April 7, 2015

- GetIt Package Manager
- FireDAC Improvements
- New Embarcadero Community Toolbar
- Native Presentation of TListView, TSwitch, TMemo, TCalendar, TMultiView, and TEdit on iOS
- Interactive Maps
- New Options for Media Library
- InputQuery now Supports Masking Input Fields

Delphi 10 Seattle – August 31, 2015

- Android Background Services Support
- FireDAC support for the NoSQL MongoDB database
- FireMonkey controls zOrder support for Windows
- New TBeaconDevice class for turning a device on one of the supported platforms into a “beacon”
- Windows 10 styles
- StyleViewer for Windows 10 Style in Bitmap Style Designer
- High-DPI Awareness and 4K monitors support
- Support for Android Services in the IDE
- Support for calling WinRT APIs

Delphi 10.1 Berlin – April 20, 2016

- Android 6.0 Support
- Windows Desktop Bridge Support
- Address Book for iOS and Android
- New ListView Item Designer
- New CalendarView control
- QuickEdits for VCL
- High DPI Support on Windows
- Hint Property Changes
- RAD Server Apache Server Support
- GetIt based Web installer

Delphi 10.2 Tokyo – March 22, 2017

- 64-bit Linux support for Delphi
- FireDAC provides Linux support for all Linux-capable DBMS
- MariaDB support (v5.5)
- MySQL support for v5.7
- Firebird support for Direct I/O
- QuickEdits for FMX
- New VCL Controls for Windows 10
- Updated IDE Look & Feel (Dark Theme)
- RAD Server Deployment License Included

Delphi 10.3 Rio – November 21, 2018

- New Delphi Language Features – Inline Variable Declaration, Type inference, AnsiString / AnsiChar on Linux
- FMX Android zOrder, Native Controls and API Level 26
- iOS 12 support
- Windows 10, VCL, and HighDPI improvements
- Per Monitor V2 support for VCL
- Extensive IDE UI modernization – includes a new light theme and a revision of the dark theme
- RAD Server architecture extensions:
- New EndPoint Attributes for Content-Type and Accept based mapping
- HTTP Verb to Custom Method Name Mapping

- Ability to Delegate Processing of a Request to a Custom Class or Component
- Quality and Performance improvements
- C++/Delphi ABI compatibility

Delphi 10.3.1 Rio – February 14, 2019

- Expanded support for iOS 12 and iPhone X series devices
- RAD Server Console UI redesign and migration to the Ext JS framework (available via GetIt)
- RAD Server includes expanded push notification support to target more devices.
- Improved FireDAC support for Firebird 3.0.4 and Firebird embedded
- HTTP and SOAP Client Library Enhancements on Windows – support for compression decoding
- Two new IDE productivity tools: Bookmarks and Navigator
- 15 new custom VCL Windows and Multi-Device FireMonkey styles

Delphi 10.3.2 Rio – July 18, 2019

- Delphi macOS 64-bit
- RAD Server Wizards and Deployment Improvements
- Enhanced Firebase Push Notifications Android Support
- Delphi Linux Client Application Support
- Runtime Library Performance improvements
- VCL Significant Quality Improvements
- Delphi 10.3.3 Rio – November 21, 2019
- Delphi Android 64-bit support
- iOS 13 and macOS Catalina Support
- RAD Server Docker Deployment
- Enterprise Connectors in Enterprise & Architect Edition
- FireDAC drivers improvements for PostgreSQL, SQLite, DB2, and Oracle

Delphi 10.3.3 Rio – November 21, 2019

- Delphi Android 64-bit support
- iOS 13 and macOS Catalina Support
- RAD Server Docker Deployment
- Enterprise Connectors in Enterprise & Architect Edition
- FireDAC drivers improvements for PostgreSQL, SQLite, DB2, and Oracle



Sobre o Autor

Sobre o Autor

David Intersimone, conhecido por muitos como "David I", é um apaixonado e inovador veterano do software que exalta e educa o mundo sobre ferramentas, desenvolvimento e arquiteturas de software.

David se juntou à Borland Software em 1985, onde ele praticamente inventou o departamento de relações com desenvolvedores. Durante os 47 anos de David I como engenheiro de software, gerente de desenvolvimento, executivo da comunidade de desenvolvedores, líder e advogado dos desenvolvedores, ele criou uma próspera e global comunidade de desenvolvedores, milhares de artigos, vídeos e posts em blogs.



Antes da Embarcadero adquirir os negócios de ferramentas da Borland Software, David passou mais de 20 anos na Borland em várias posições como de evangelização, engenharia e desenvolvimento, incluindo a criação do programa de relações com desenvolvedores da empresa. David continuou a trabalhar na Embarcadero por 11 anos como vice-presidente de relações com desenvolvedores.

Hoje, David compartilha suas visões e insights como pioneiro em relações de desenvolvedores com gerentes de programas, diretores e desenvolvedores onde ministra workshops, webinars, orientações e conselhos sobre comunidades, advocacia de desenvolvedores e desenvolvimento de software.

David também é um MVP da Embarcadero (<https://www.embarcadero.com/embarcadero-mvp-program>) participando da comunidade de desenvolvedores, escrevendo, ensinando, blogando e participando de eventos online.

David I recebeu recentemente o Prêmio Spirit of Delphi 2019. Os vencedores do prêmio são selecionados pelos membros internos da equipe do Delphi. Os candidatos são considerados com base em suas contribuições para o espírito do Delphi e para a comunidade de desenvolvedores.

Embarcadero Delphi e o Desenvolvimento de Aplicações Windows



An Idera, Inc. Company

Por David Intersimone "David I"

Copyright © 2020 Embarcadero Technologies, Inc.