



Presentando el RAD Studio XE5 con soporte iOS y Android

Fernando Rizzato

Lead Software Consultant, Latin America

Fernando.rizzato@embarcadero.com

Agenda

- Bienvenida e Introducción
- Vivimos en un Mundo Múltiples Dispositivos
- Varados en las Aplicaciones de Escritorio
- 5 errores al migrar a dispositivos móviles
- RAD Studio XE5: Desarrollo Nativo para Múltiples Dispositivos
- Resumen, Preguntas y Respuestas.

Embarcadero Technologies

Founded 1993

3.2 Million Customers

500+ Employees in 29 Countries

5 R&D Centers of Excellence

World Wide Sales & Support

Embarcadero Products

Application Development

-  RAD Studio
-  C++Builder
-  Delphi
-  HTML5 Builder

Data Architecture and Modeling

-  ER/Studio Data Architect
-  ER/Studio Portal
-  ER/Studio Business Architect
-  ER/Studio Software Architect

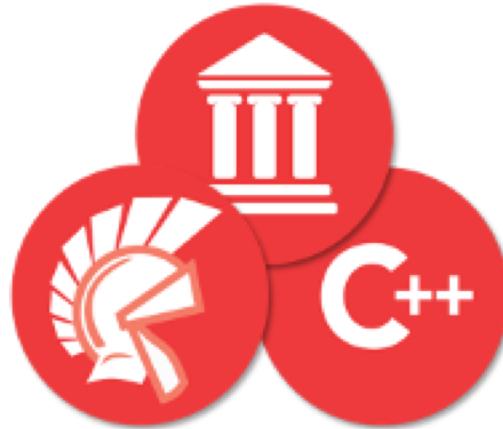
Database Development, Administration and Performance Optimization

-  DB Power Studio
-  DBArtisan
-  Rapid SQL
-  DB Optimizer
-  DB Change Manager

Embedded and Workgroup Databases

-  InterBase

3 Million Delphi/C++ Developers Coding ...



- 9.7 Billion lines of code per year
- 100,000 Windows Apps
- 80% less code required – effective \$1.60 per line of code
- \$62 Billion in annual developer savings



**Vivimos en un mundo multi
dispositivos...**

Análisis de la situación

2013: La Revolución del Cliente

Mil Millones



Windows

65 Millones



Mac

+ de mil millones

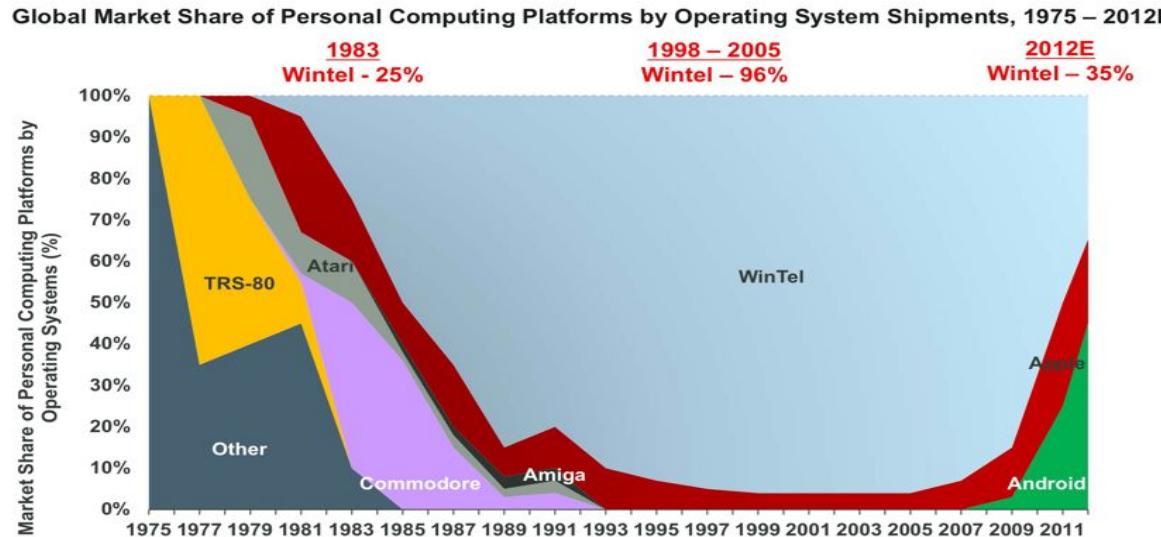


Mobile

Hoy, múltiples dispositivos sin precedentes

La Revolución del Cliente

Re-Imagination of Computing Operating Systems -
iOS + Android = 45% Share vs. 35% for Windows



KPCB

Source: Asymco.com (as of 2011), Public Filings, Morgan Stanley Research, Gartner for 2012E data. 2012E data as of Q3:12.

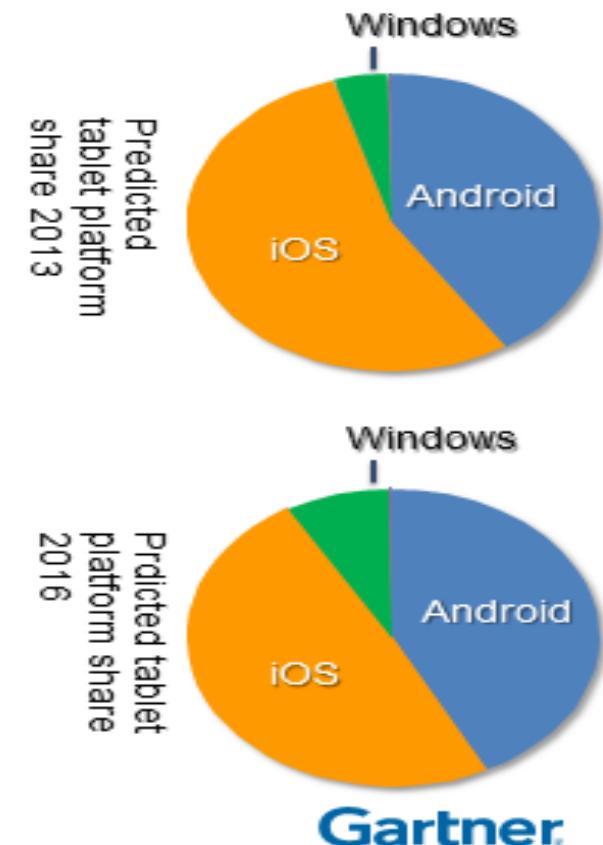
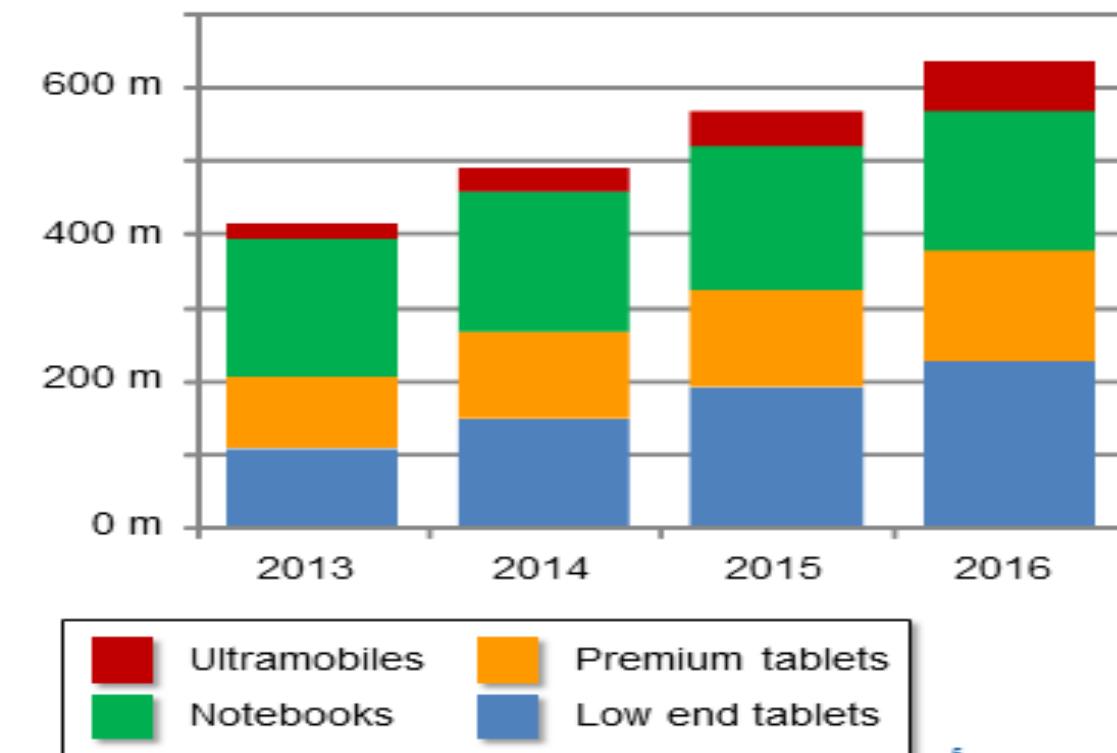
24

Múltiples Dispositivos sin precedentes

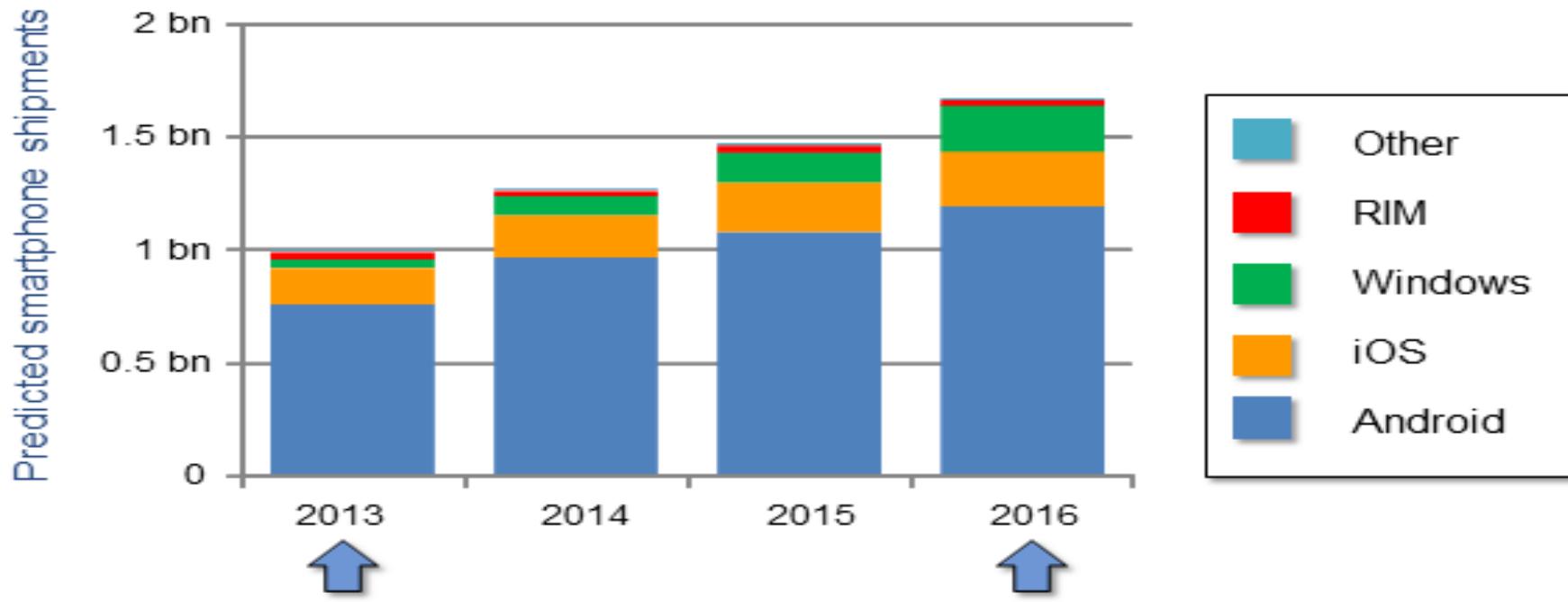
embarcadero®

Mobile PC and tablet shipments

Predicted mobile PC and tablet shipments 2013 to 2016
Gartner predictions published Q4 2012.



Smartphones Will Dominate Handsets



The rise of
smartphones

Smartphones will
be 52% total global
handset shipments

Smartphones will
be 78% total global
handset shipments

2013+



Continúa la expansión de dispositivos



Varados en las aplicaciones de Escritorio

¿Qué significan estos cambios para...?

- Usted
- Su Equipo
- Su Negocio

Estos cambios y... Usted

- Sus opciones ...
 - Amplían enormemente
 - Si no se actualiza, quedará fuera del negocio.
- Su participación en el proyecto
 - Administración de mejores proyectos
 - Oportunidad de ser el desarrollador estrella del negocio.
 - Innovación no imaginada.

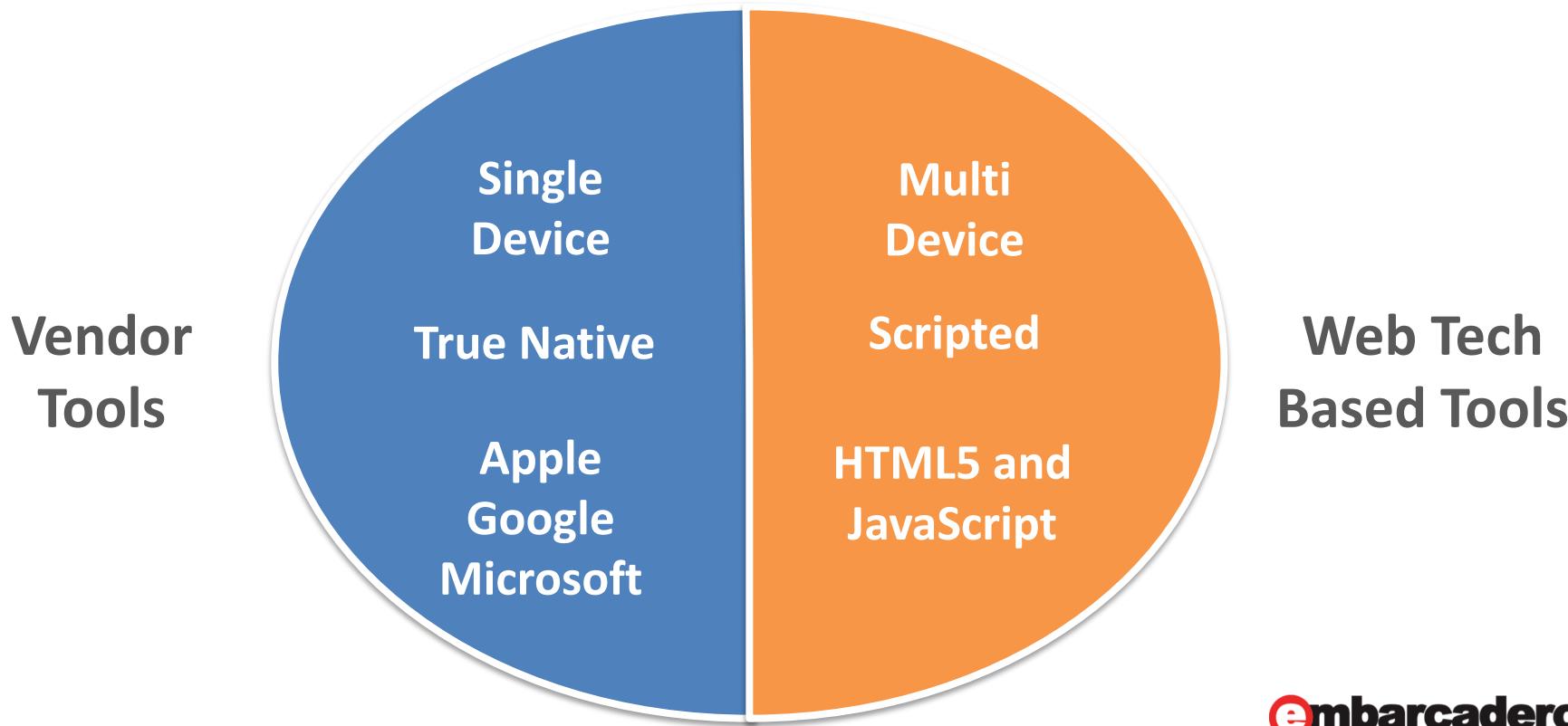
Estos cambios y... Su Equipo

- Los mismos criterios que para Usted
- División en función de la cantidad de plataformas
 - Actualmente esto sería tres o cuatro equipos (Win, Android, iOS, OS X)
 - Cómo te administrarías sin tener
 - 3x ó 4x de presupuesto y 3x ó 4x desarrolladores
 - Los miembros desmotivados dejarían los mejores proyectos.

Estos cambios y.. Su Negocio

- Los clientes esperan que todo sea para “ayer”
 - 24/7, la interacción personalizada de su negocio es demandada por las empresas de todos los tamaños
 - Those that are WIN, those that aren’t LOSE
- Costo...
 - Costo Transaccional y compra compulsiva
 - Mantenimiento y Lealtad del Cliente
 - Pérdida total de oportunidad
 - Exclusión total de un gran número de clientes

Opciones Actuales de Herramientas



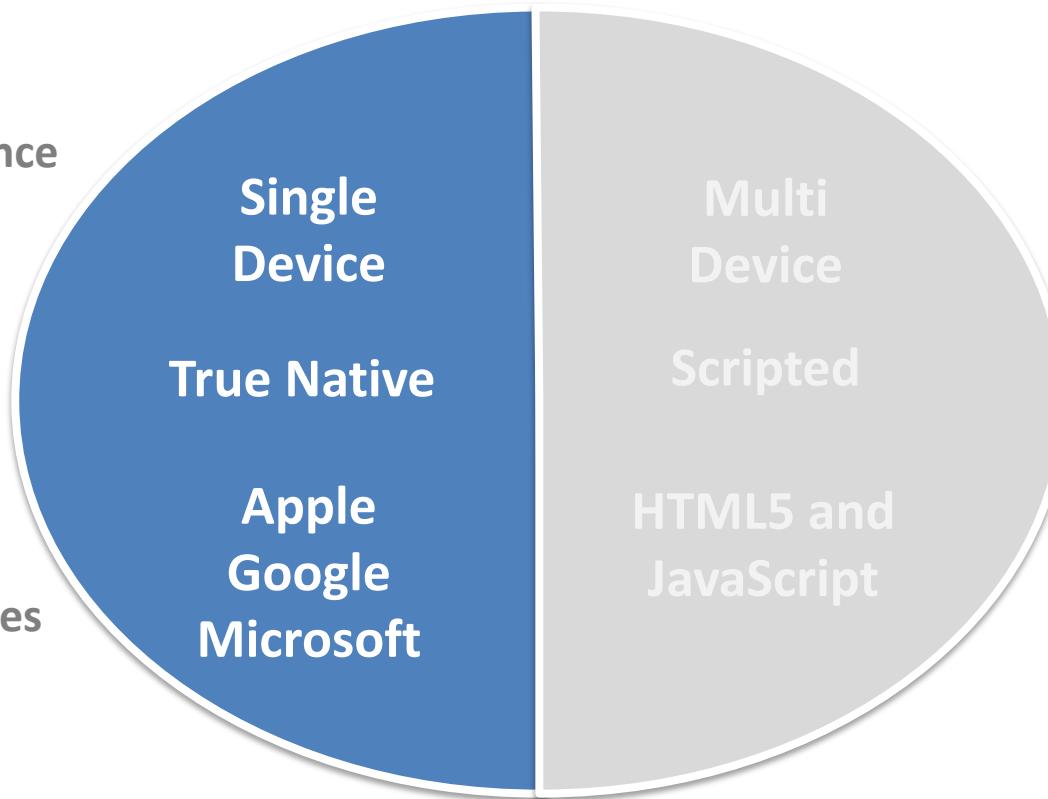
Pros & Cons for Multi-Device – Vendor Tools vs Web Tech Tools

Pros

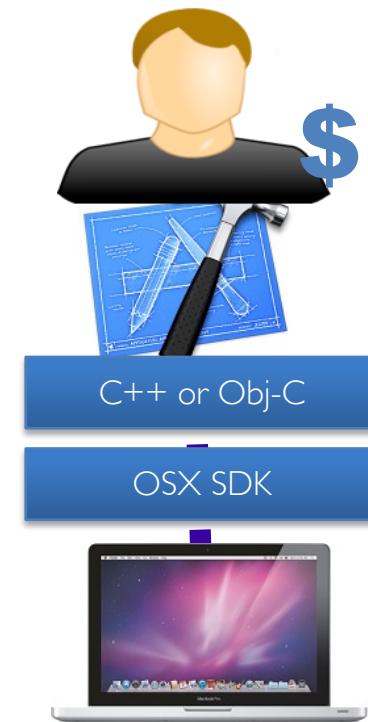
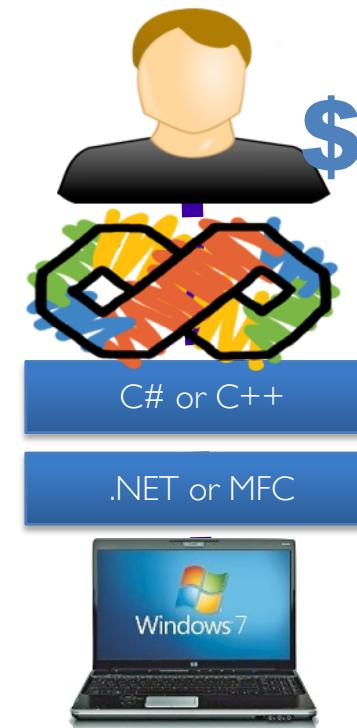
Native UX
Native performance
Secure

Cons

Multiple code bases
Multiple teams
Higher dev cost

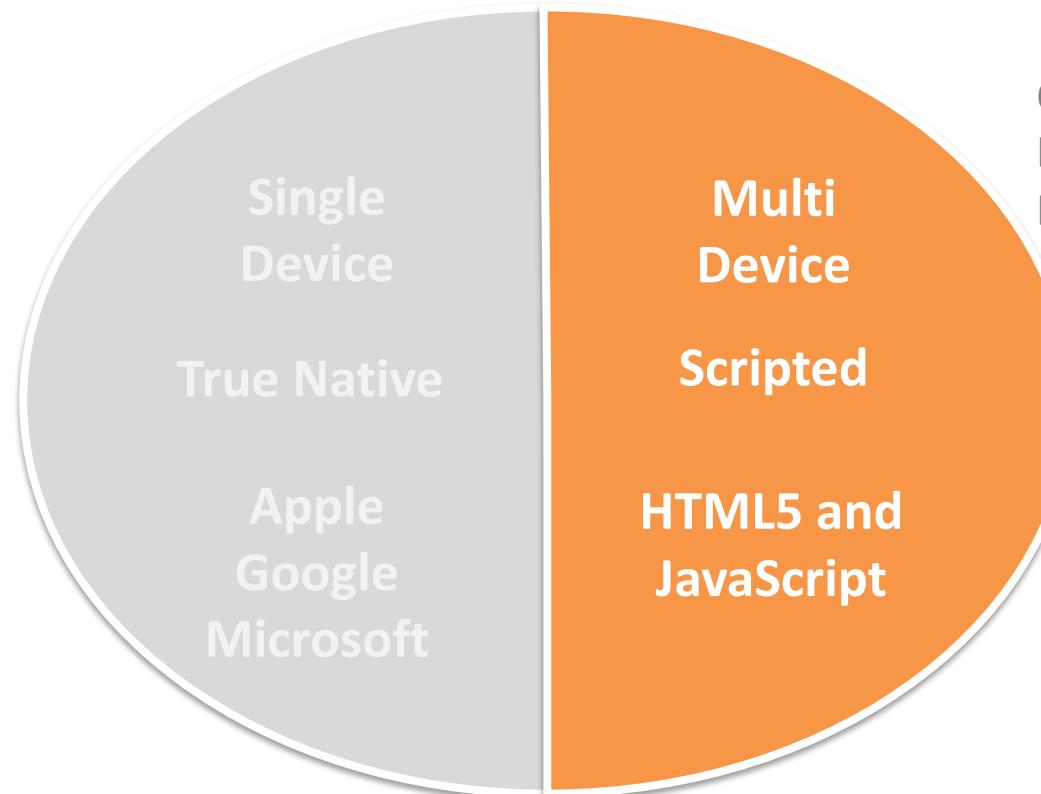


Enfoque del “Vendor Tool” para Múltiples Dispositivos



Múltiples lenguajes de programación

Pros & Cons for Multi-Device – Web Tech Based Tools



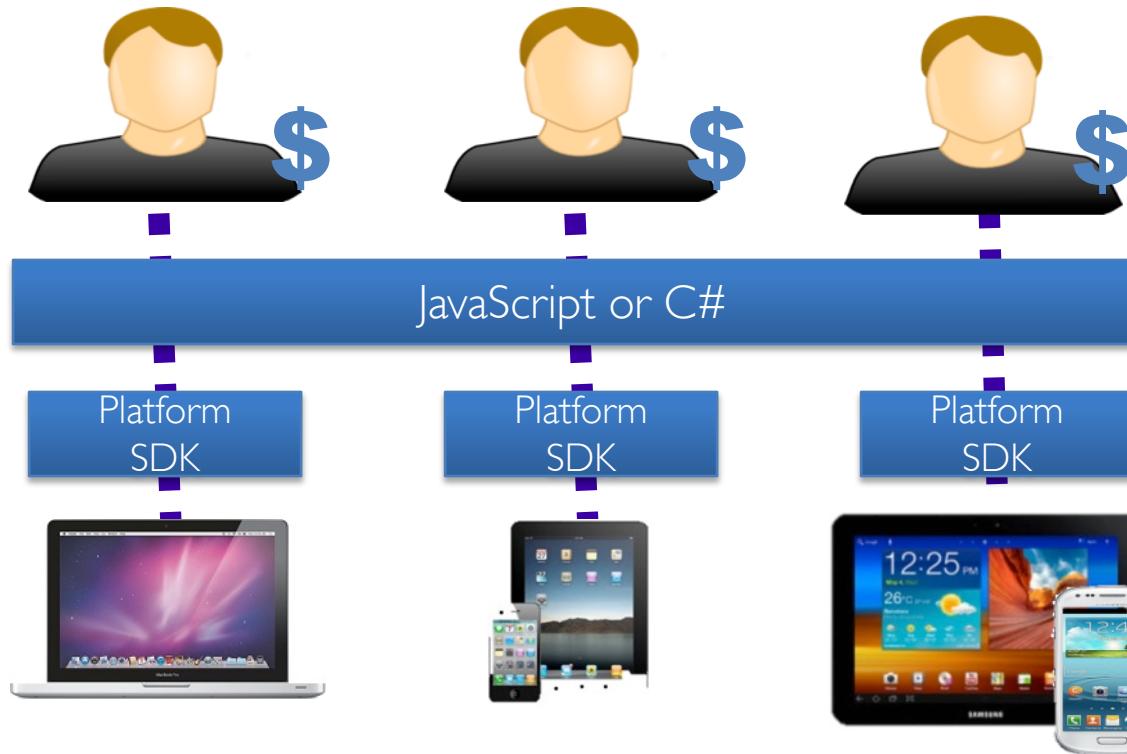
Pros

- One team
- Lower dev costs
- Fast time to market

Cons

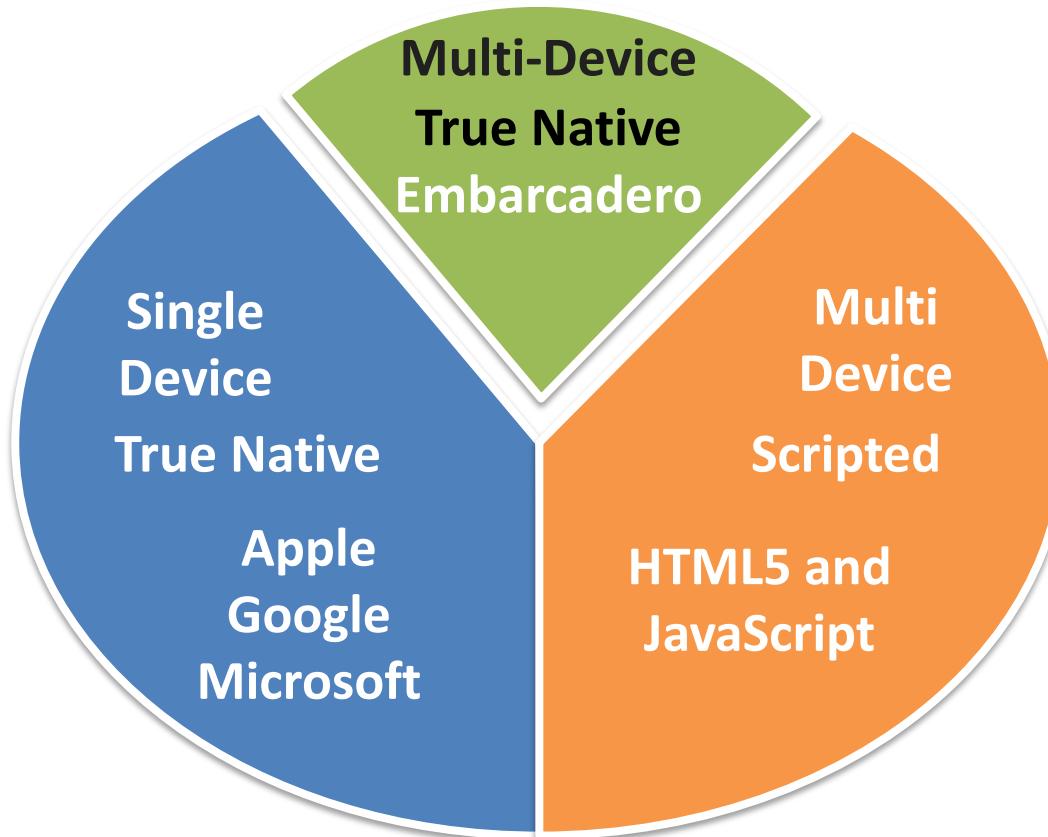
- Script Performance
- Non-Native UX
- Unsecure

Código Virtual para Soluciones Móviles



one language, multiple codebases

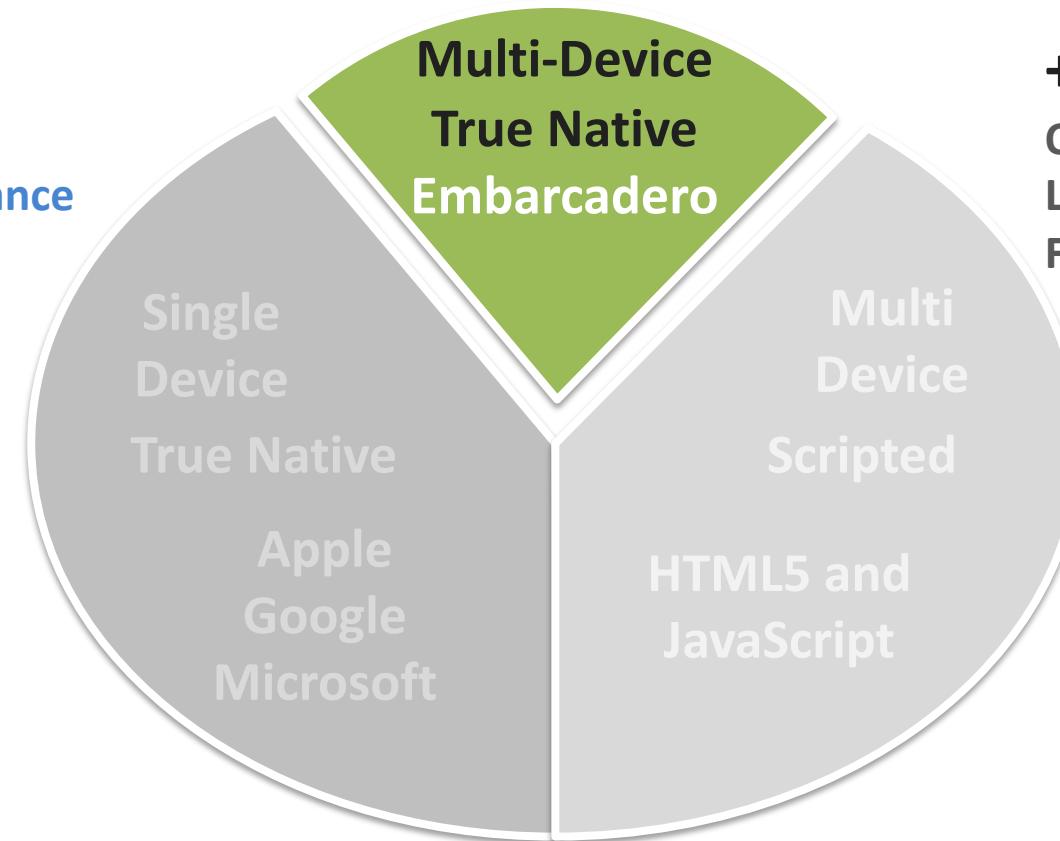
Múltiple Dispositivos Realmente Nativo



Multiple Dispositivos Realmente Nativo

Pros

Native UX
Native performance
Secure



+Pros

One team
Lower dev costs
Fast time to market



Los 5 errores que los desarrolladores comenten al migrar a móviles...

Error #1

Tratar de duplicar el desktop en el dispositivo móvil.



Dispositivos Móviles no cuenta con los recursos de un PC como:

- grandes monitores, que facilitan la visualización
- mucha RAM y procesadores rápidos
- teclados que facilitan la escritura

Tratar de desarrollar procesos complejos sobre el dispositivo que:

- saturan la pantalla
- consumen mucho recurso de CPU/Memoria
- afecta la usabilidad para el usuario final

Cómo evitarlo

- Identificar las acciones de mayor valor para ser implementadas en el dispositivo
 - diseñar sólo para estos requerimientos
 - asegurarse que los usuarios completen las acciones en pocos pasos
 - establecer un máximo de números de pasos para completar una acción
- Aprenda de las aplicaciones más comunes (incluyendo las propias del dispositivo)

Error #1: Cómo evitarlo con RAD Studio

- Plantillas para comenzar su aplicación
 - Diseños comunes para dispositivos móviles (teléfonos y tabletas)
 - Plantillas Maestro - Detalle
- Fácil uso de componentes, para un desarrollo rápido y consistente
- Soporte a diferentes tipos de dispositivos (formularios, resolución, orientación, etc).

Error #2

No hacer la experiencia del usuario su prioridad máxima



- cientos de millones de apps con una cantidad de opciones
- si su aplicación no es fácil de usar, el usuario no la usa
- los usuarios generalmente no regresan si su experiencia fue mala

Cómo evitarlo

- enfocarse en un buen diseño de interface gráfica de usuario
- verificar la experiencia del usuario de las aplicaciones con usuarios reales
- asegurarse que se completen tareas con pasos mínimos
- asegurase de haber cumplido los requerimientos de GUI antes de liberar la aplicación

Error #2: Como evitarlo con RAD Studio

- Creación rápida de prototipos para que Usted (y sus clientes) puedan verificar en el dispositivo cada vez que avanzamos en el proyecto
- Utilize los servicios del dispositivo (sensor, etc), para conocer de manera dinámica que acciones tomar

Error #3

Letting too much get between the app, the OS and the Hardware



Debería reducir el tiempo de respuesta lo mas que pueda y aprovechar todas las capacidades que ofrece el dispositivo

A mobile app development platform that abstracts away device details delivers only the common denominator of functionality across devices

Como evitarlo

- construir una verdadera aplicación nativa, no sólo contruir para una sistema operativo en particular
- una aplicación nativa está optimizada para un hardware particular
- evitar “Wrappers” como JVMs, con la finalidad de aprovechar las características del dispositivo al máximo.

Error #3: Como evitarlo con RAD Studio

- Creando aplicaciones nativas
- Implementar cuidando los temas de rendimiento

Error #4

Construir aplicaciones separadas para cada plataforma



Construir aplicaciones separadas para cada plataforma nos genera los siguientes problemas:

- muchos de los usuarios finales tienen múltiples dispositivos
- su aplicación se puede comportar diferente en cada dispositivo
- aplicaciones con características diferentes, obligan a reparar los errores en momentos diferentes
- mantener múltiples aplicaciones usando varios lenguajes de programación genera retrabajo.

Como evitarlo

Aproveche un único código fuente listo y optimizado para diferentes plataformas.

- Obtendrá una experiencia de usuario mas consistente en todos los dispositivos
- Reduzca los costos relacionados con desarrollo, control de calidad, corrección de errores usando un único código fuente.

Error #4: Como evitarlo con RAD Studio

- Un sólo lenguaje de programación
- One Codebase
- Una sola plataforma de desarrollo

Error #5

No olvidar la seguridad



Basarse en las medidas de seguridad de los dispositivos o de la red para proteger los datos puede ser peligroso

- tomar medidas activas para proteger los datos que residen en el dispositivo
- tomar medidas activas para proteger los datos trasmítidos y las invocaciones a servicios que se ejecutan dentro del dispositivo
- Java y JavaScript son blancos fáciles de piratas informáticos

Cómo evitarlo

- interacción directa del código con el dispositivo
- incluir precauciones propias para reducir el riesgo de ataques de terceros
- encriptar los datos sensibles almacenados en el dispositivo

Error #5: Cómo evitarlo con RAD Studio

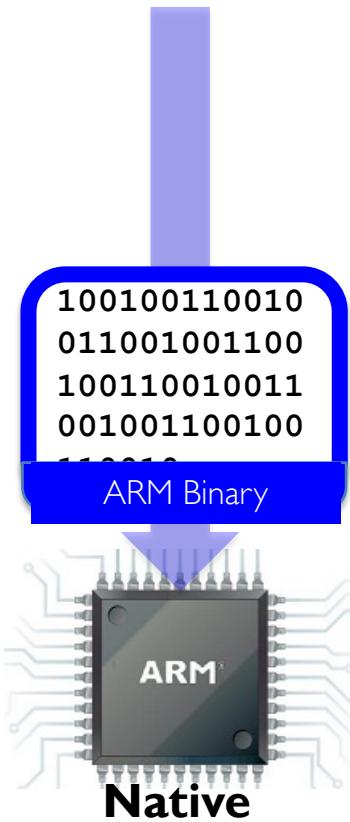
- Código verdaderamente nativo hace que su aplicación menos expuesta
- Cifrado de datos "punta a punta" con cliente DataSnap Mobile
- InterBase Togo con datos cifrados directamente a su dispositivo



True Native Code vs Virtual Code

Other considerations

Embarcadero, XCode



Appcelerator, HTML5, Sencha, Kendo, Edge, PhoneGap, Corona



JavaScript
Runtime
Interpreter

Dalvik

This diagram shows the VM compilation path. It starts with a blue box containing bytecode IL (Intermediate Language) code. This code is shown being loaded into an ARM processor chip, which is labeled "VM". A vertical red bar connects the bytecode IL box to the ARM chip.

```
0  aload_0
1  invokestatic #199; <Method initializePrimordial()
2  ldc           #25; "Ti.Geolocation.distanceFilter"
3  ldc           #26; "Callbacks Are Your Friend"
4  aload_1
5  aload_2
6  invokestatic #205; <Method initialize()
7  ldc           #210; <Method initialize()
8  ldc           #214; <Method initialize()
9  ldc           #217; <Method initialize()
10 ldc          #220; <Method initialize()
11 ldc          #223; <Method hostOrTarget()
12 ldc          #186; <Field PRIMORDIAL
13 ldc          #227; <Method initializeSchemes()
14 ldc          #230; <Method hostOrTarget()
15 ldc          #190; <Field PRISTINE
```

Bytecode IL

Virtual Machine
(VM) Runtime

True Native Code vs Virtual/Scripting Code

“Anyone remember VB?”

C, C++, Obj-C, Delphi

True
Native
Code

ARM CPU

vs

HTML5/JavaScript,
Java, .NET, Mono

Virtual Code

Software Virtual CPU aka Virtual
Machine (VM)

ARM CPU

Tunability: App is limited by the preset capabilities of the VM.

Performance:
App runs within a software machine process running on the device.

Predictability:
Memory is automatically garbage collected - when full and when the VM decides it's time. Not the dev.

User Experience: Virtual Apps are oblivious to device specs. Device vendors (Apple, Samsung, etc) spend countless engineering & dollars tuning & squeezing great UX into a device based on limited CPU, battery, and memory.

True Native Code

Native Device Applications
Intel/ARM Machine Code
Maximum Performance
Highly tunable
Smallest Possible Footprint
Low Latency
Developer sched mem mgmt

Languages:

C++, Obj-C, C, Delphi

Best Suited for:

User/Client Apps
Embedded Applications
Real-time Applications

Virtual/Scripting Code

Virtual Device Application
Virtual Machine Code
Medium to Slow Performance
Limited tunability
Large Runtime Footprint
Med to High Latency
Runtime scheduled mem mgmt

Languages:

HTML5/JavaScript, C#, Oxygen/Prism,
Java, Mono C#

Best Suited for:

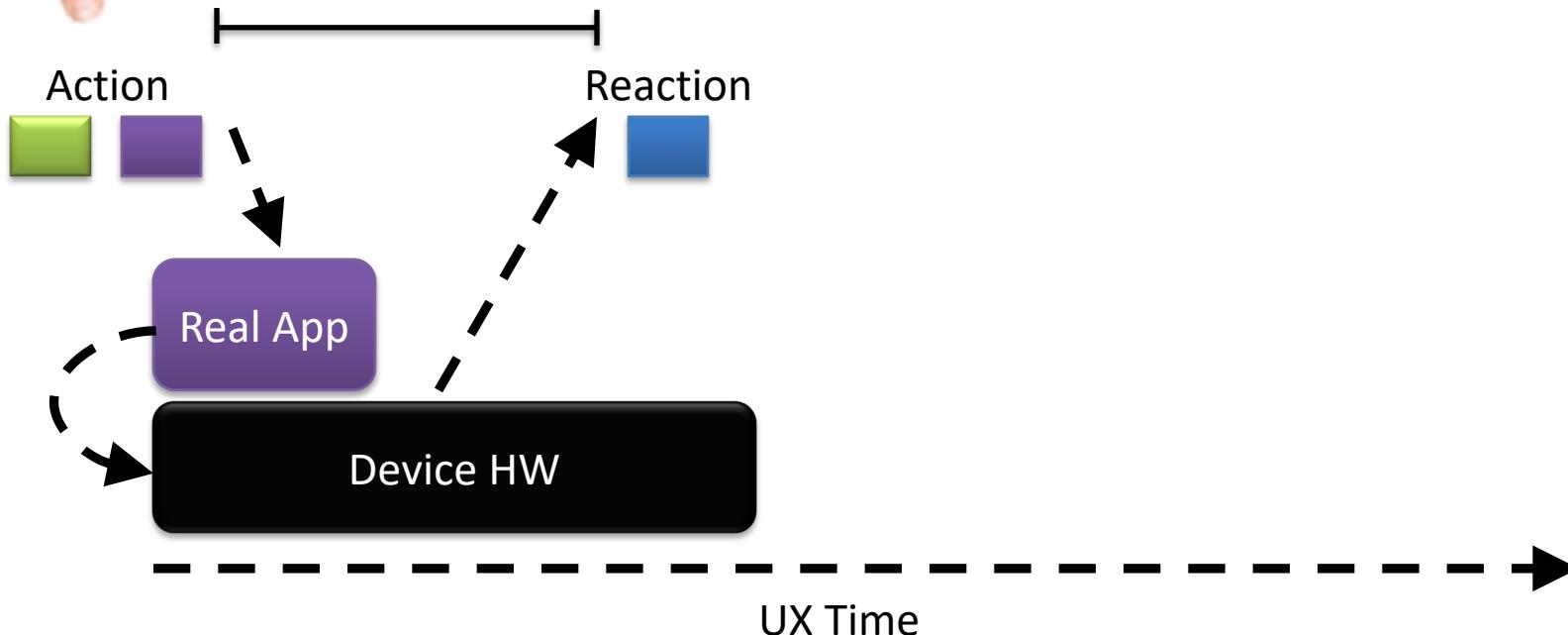
Web Server Applications
Browser based Applications

Truly Native or Scripted?

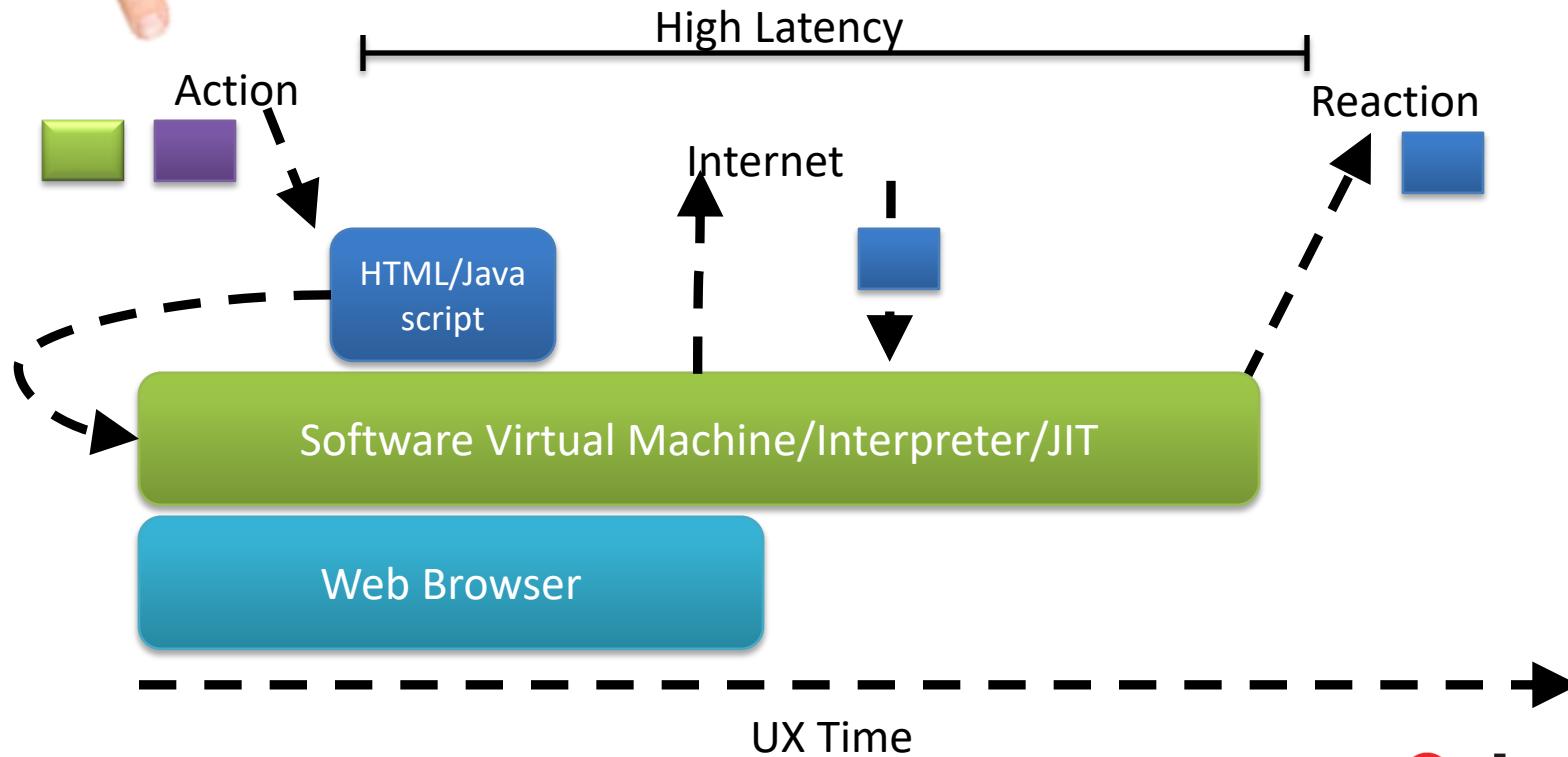
	True Native	Scripted	VM	
Platform Vendor Tools	Embarcadero	HTML5 Cross Platform	Appcelerator	Xamarin
Tools	XCode, Visual Studio, Eclipse	RADStudio	Adobe, Sencha, Kendo, HTML5Builder	Titanium
Platforms	Win <u>or</u> Mac <u>or</u> iOS <u>or</u> Android*	Mac/Win/iOS & Android* (2013)	iOS/Android	iOS/Android
Single Source / Multi-Device	No	Yes	Yes	Partial
Runtime	ARM	ARM	JavaScript	JavaScript
				Bytecode VM

*Android NDK is True Native – Android ADK with Java/Davlik VM is not

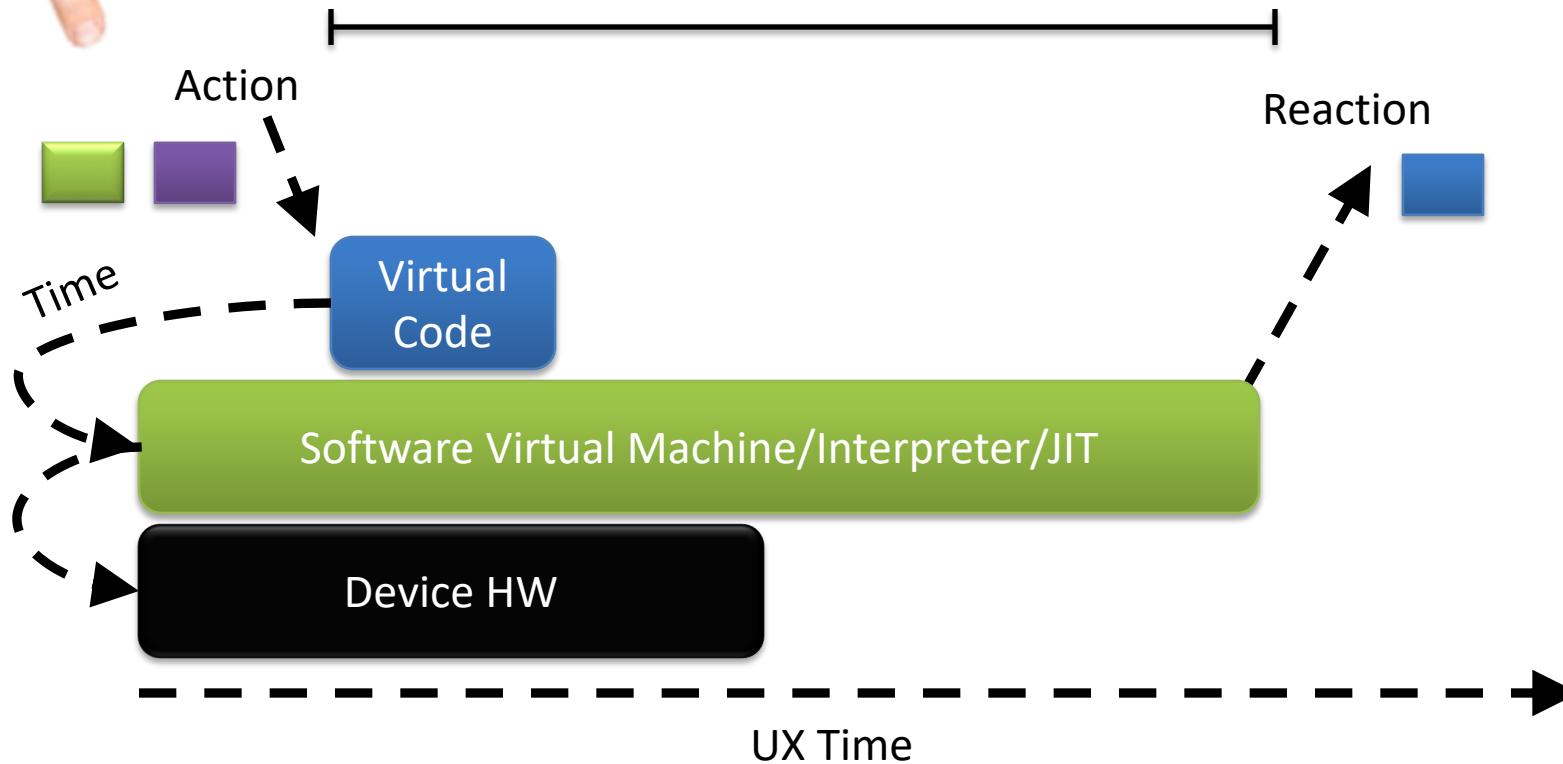
Low Latency = Great UX



High Latency UX is accepted in Browser



Unacceptable in Apps



"The biggest mistake we made as a company was betting too much on HTML5 as opposed to native"

Mark Zuckerberg - Facebook CEO

"Apple unbundling the (Java) runtime will erase a large number of security vulnerabilities"

AppleInsider 2011

"One of the biggest advantages we've gained from building on native iOS has been the ability to make the app fast."

Jonathan Dann – Facebook 2012

"We are currently unaware of a practical solution to this (Java vulnerability) problem"

U.S. Department of Homeland Security 2013

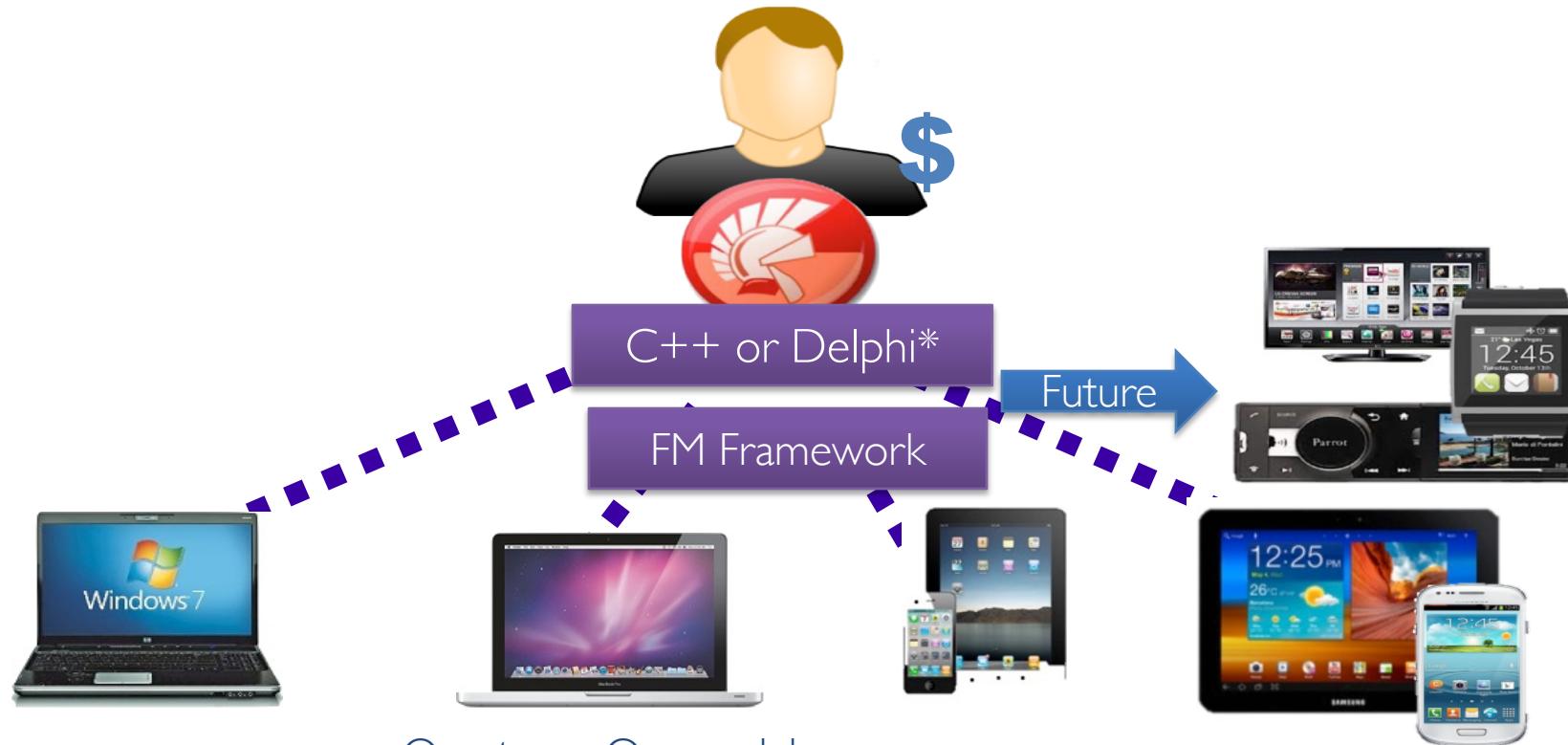
Code Nativo

- Las aplicaciones nativas en RAD Studio son código limpio y se ejecutan directamente sobre el CPU obteniendo las siguientes ventajas:
 - **Rápida** – Sin comprometer el rendimiento del dispositivo y con control total sobre las API y el CPU cuando sea necesario. No está limitado por lo lento de los intérpretes de código y VMs.
 - **Confiable** – Las aplicaciones se ejecutan directamente sobre el CPU del dispositivo, según lo establecido por los fabricantes. No es ralentizada por capas de software o “Garbage Collection”
 - **Mejor Experiencia al Usuario (UX)** – Las aplicaciones aprovechan al máximo las capacidades del dispositivo y su rendimiento



RAD Studio XE5: Desarrollo Nativo para Múltiples Dispositivos

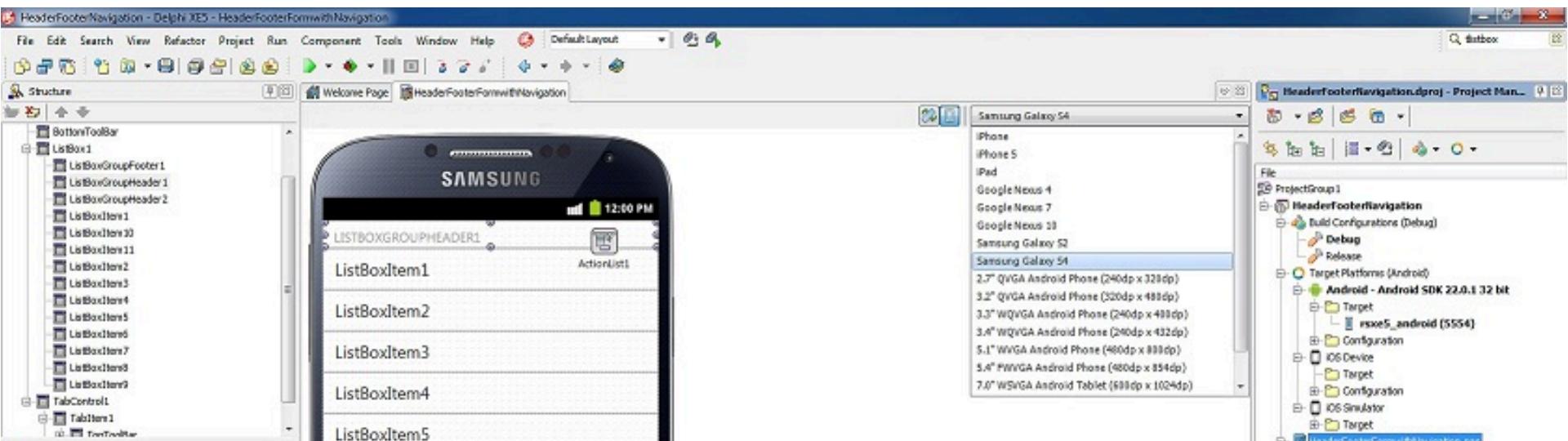
RAD Studio XE5



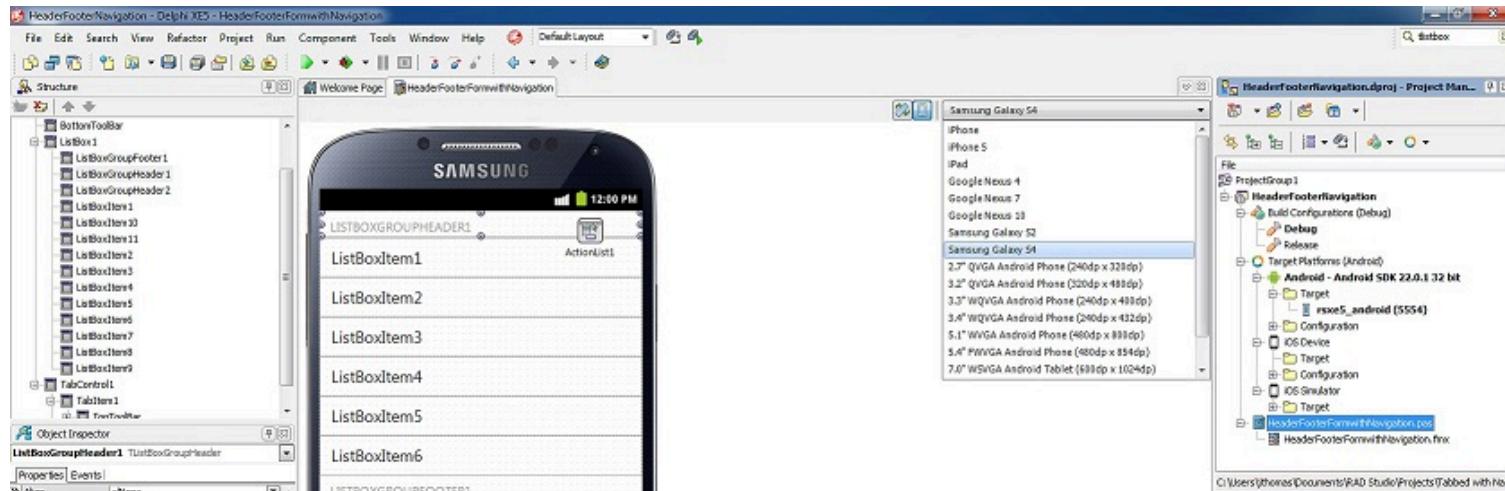
One team, One codebase

Delphi FireMonkey para Android y iOS

- Estilos Nativos y Personalizados
- Aplicaciones Nativas
- Modelo de desarrollo basado en componentes

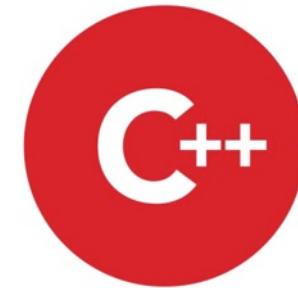


Desarrollo Visual



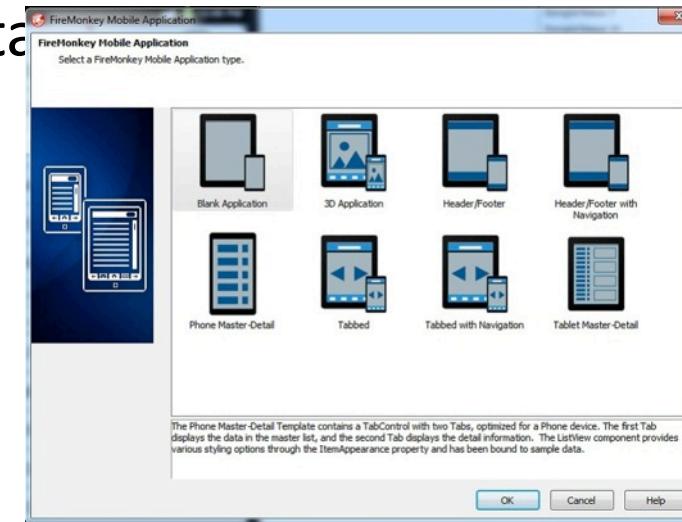
Características de Lenguajes Modernos

- Property-Method-Event
- Automatic Reference Counting
- Interfaces
- Attributes
- Generics/Collections
- Métodos Anónimos (Lambda functions/closures)
- Rich RTTI (introspection/reflection)



Ayudante para crear aplicaciones Móviles

- Inicie con una aplicación en blanco HD o 3D FireMonkey
- Seleccione modelos de aplicación (Tabbed, Header/Footer y Master/Detail)



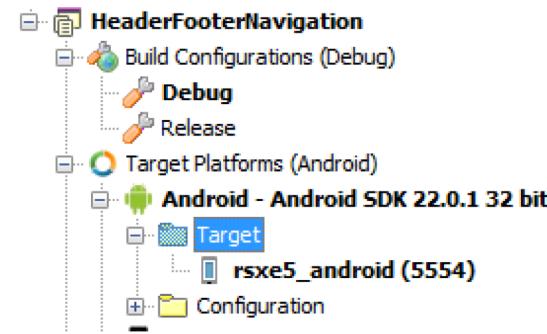
Opciones del IDE para el deployment de Android

Deploy apl Android al Emulador

- AVD device

Deploy apl. Android al Dispositivo

- Debug
 - debug/deploy to device
- Ad-hoc
 - distribute APK anywhere
- App Store
 - deploy apl Android a tiendas (Play Store, Amazon)



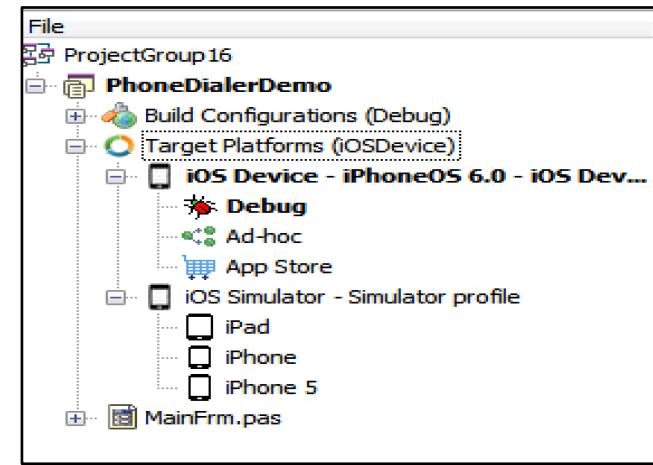
Opciones del IDE para el deployment iOS

Deploy apls iOS para el Simulador

- iPhone (Retina/non-Retina)
- iPhone 5 (Retina/non-Retina)
- iPad (Retina/non-Retina)

Deploy apls iOS para el Dispositivo

- Debug (debug/deploy to device)
- Ad-hoc (distribute within own enterprise)
- App Store (deploy to the App Store)

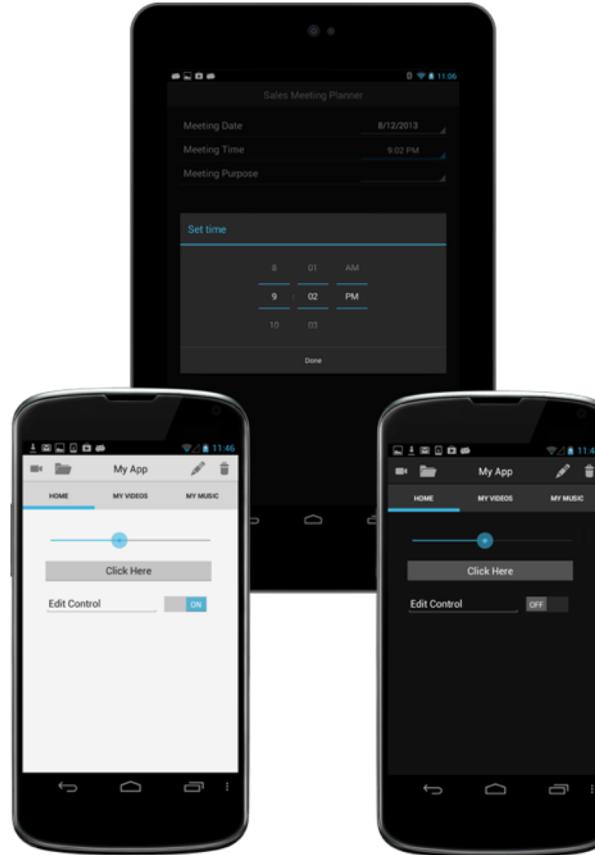


Configurando la Aplicación

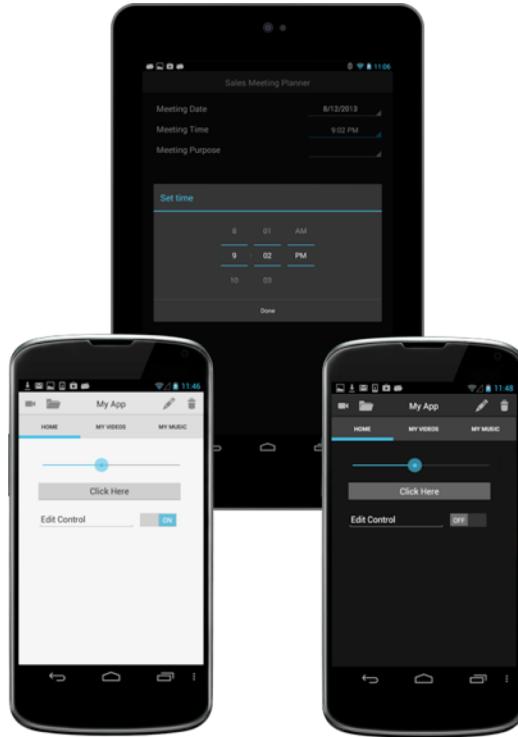
- A través de Project->Options
- Seleccione diferentes iconos (para multiple resoluciones) en Project-> Options



Estilos Nativos



Estilos para Android



- Holo Light (Android por defecto)
- Holo Dark (Estilo personalizado)
- Ambos estilos incluyen soporte a todas las resoluciones (1x – 3x)

Controles Nativos

- Message alerts
- Custom Picker
- Date Picker
- Phone Dialer
- iOS Keyboards
- Text Editing for TMemo and TEdit
 - Cut/Copy/Paste/Zoom



Soporte Multimedia y Redes Sociales

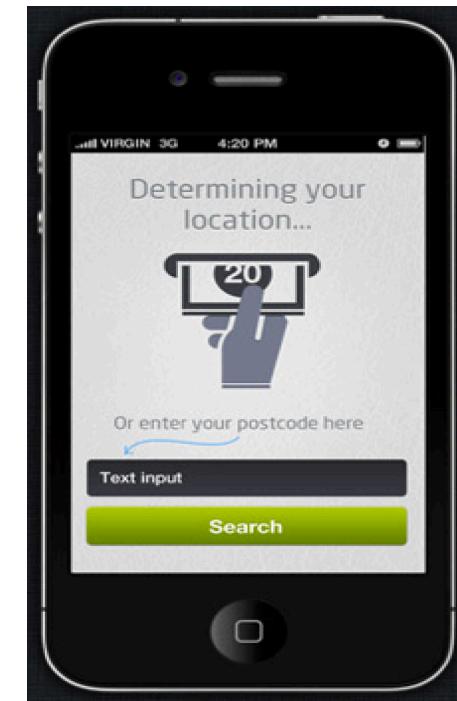
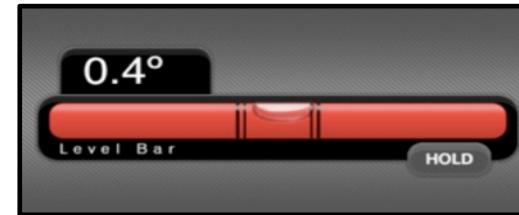


- Acceso a la Cámara (App, Roll).
- Compartir contenido (p.e. Fotos) via Message (SMS), Mail, Facebook, Twitter etc.
- Slide Transitions for Tabs



Componentes para el soporte de los sensores

- Localización (GPS)
- Movimiento (Accelerometer)
- Orientación (Gyroscope)



Servicios de los Dispositivos Móviles

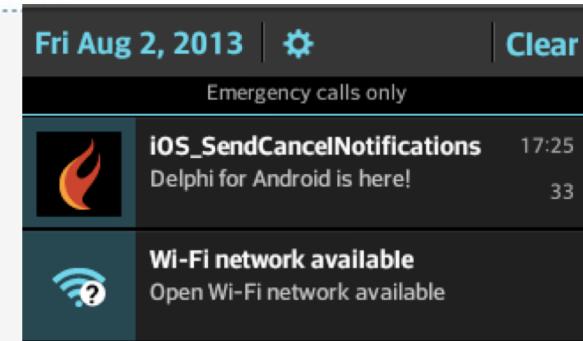
- Notification Center
- ... y mas



Notification Center

- Nuevo componente FireMonkey en XE5
- Soporte a diferentes tipos de notificaciones para iOS, Android y MacOSX

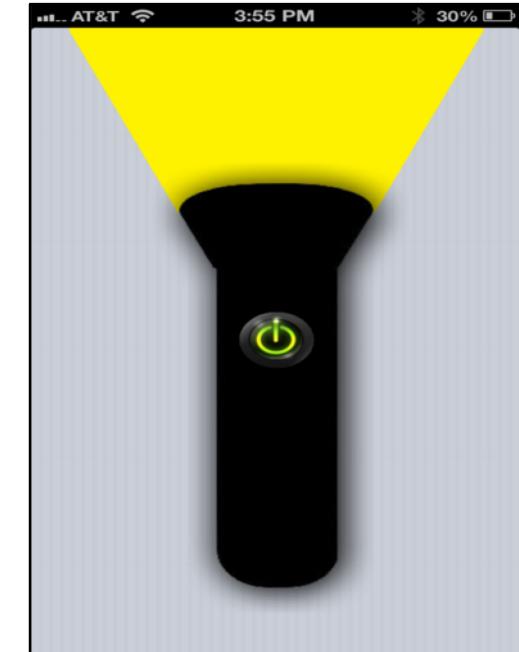
```
procedure TForm1.ScheduleNotification;
var
  MyNotification: TNotification;
begin
  MyNotification := NotificationCenter1.CreateNotification;
  try
    MyNotification.Name := 'MyNotification';
    MyNotification.AlertBody := 'Delphi for your mobile device is here!';
    // Fired in 10 seconds
    MyNotification.FireDate := Now + EncodeTime(0, 0, 10, 0);
    // Send notification to Notification Center
    NotificationCenter1.ScheduleNotification(MyNotification);
  finally
    MyNotification.DisposeOf;
  end;
end;
```



Cámara

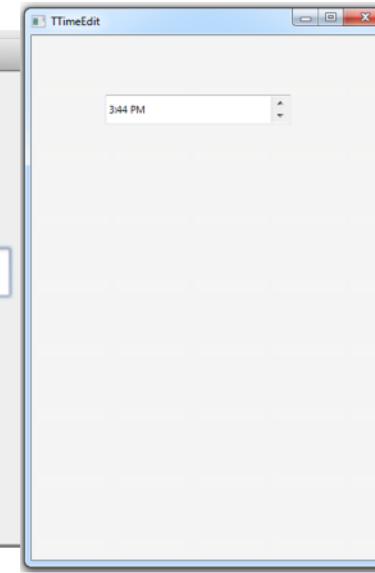
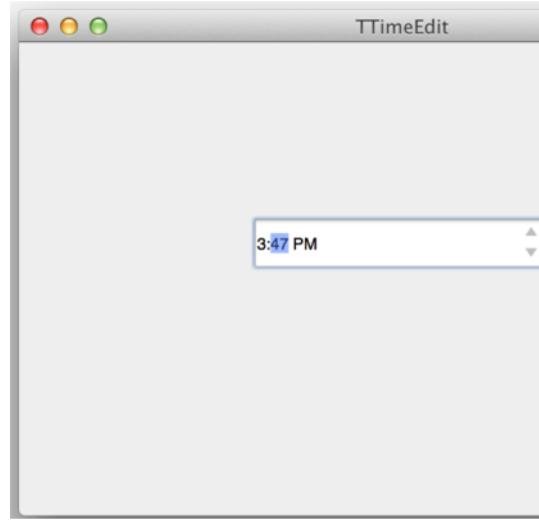
Provee acceso a los sensores de la cámara:

- activación de flash
- obtener la posición.
- acceso a la cámara de atrás/adelante
- tomar fotos



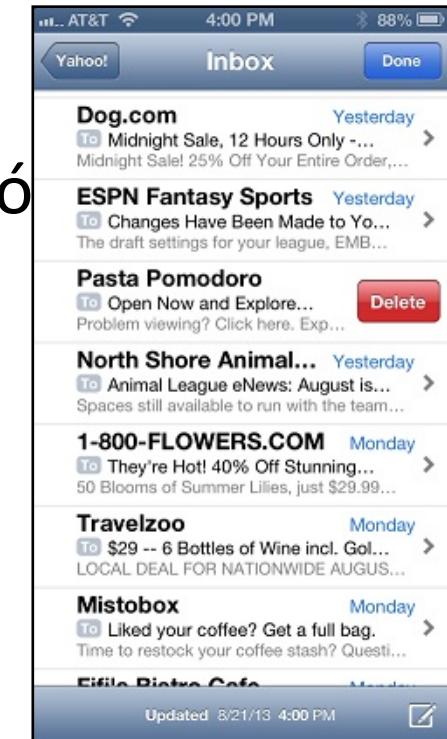
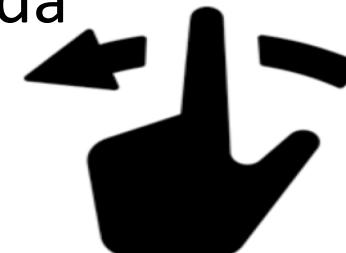
TTimeEdit

- Nuevo componente FireMonkey en XE5
- Soporte a Windows, Mac, iOS y Android
 - Native pickers para iOS y Android



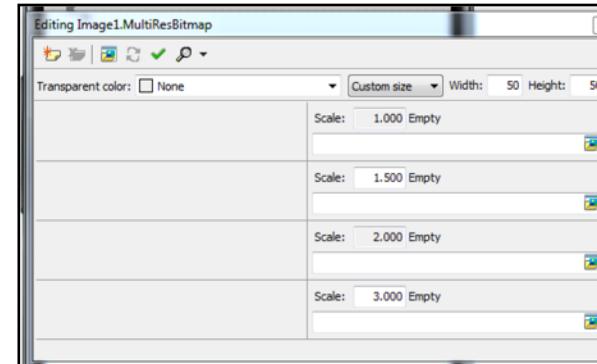
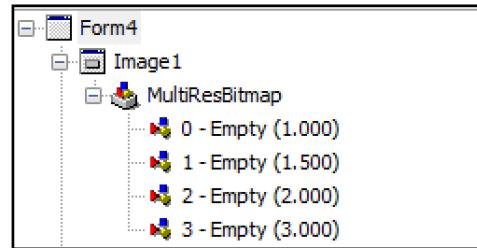
Características del TListView en XE5

- Swipe para eliminar elementos
- Múltiples modos de Visualización y Edición
- Diferentes “Accessory Styles”
- Animación en modo de edición
- Búsqueda por Filtro habilitada
- Soporte a LiveBindings



Soporte de images para múltiples resoluciones

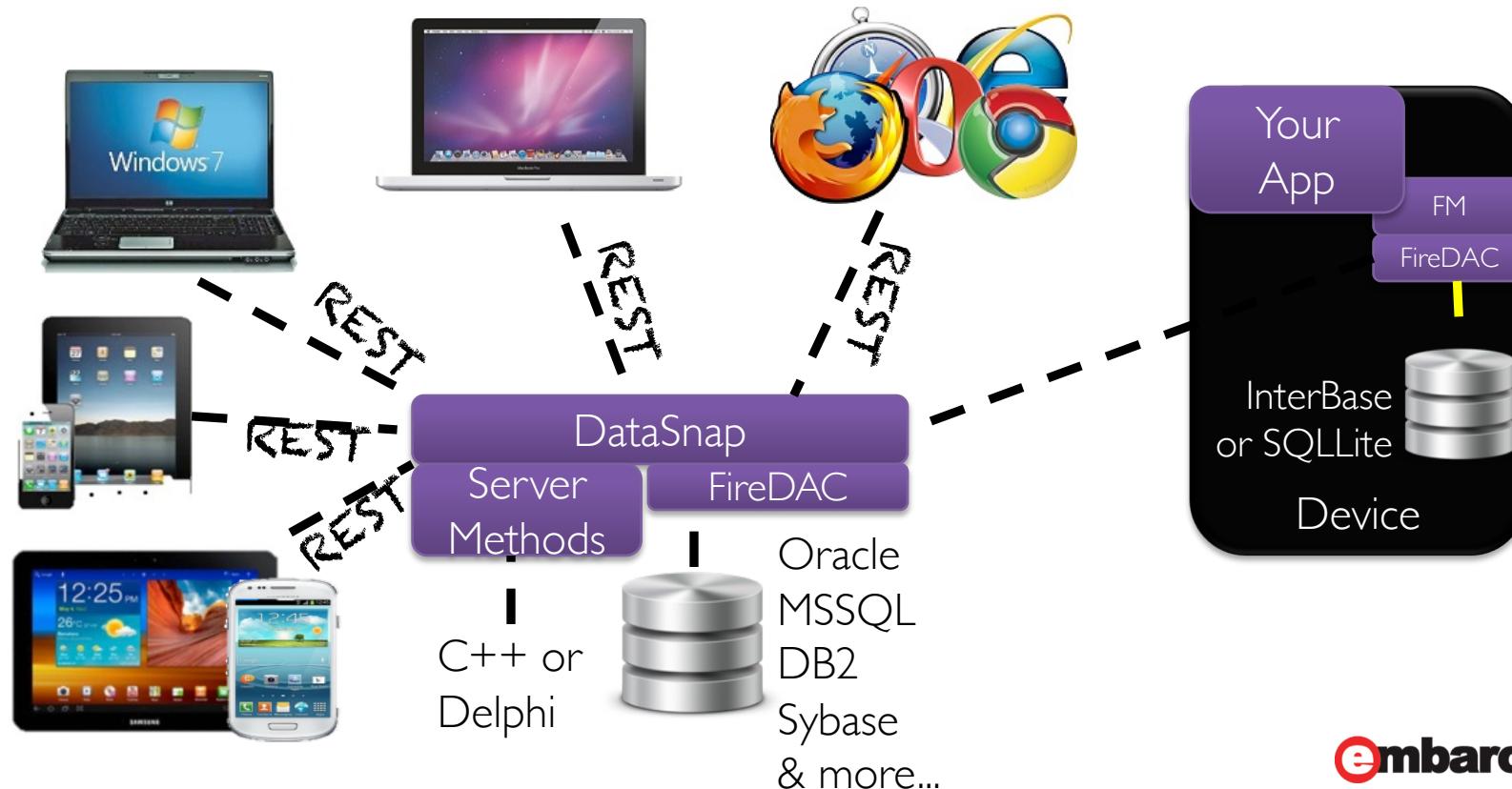
- TImage actualizado
- Propiedad MultiResBitmap adicionada
- 1x y 2x mostradas por defecto
- Se puede seleccionar sus propias resoluciones, i.e. 1.5x, 3x etc.



Mas componentes para Dispositivos Móviles

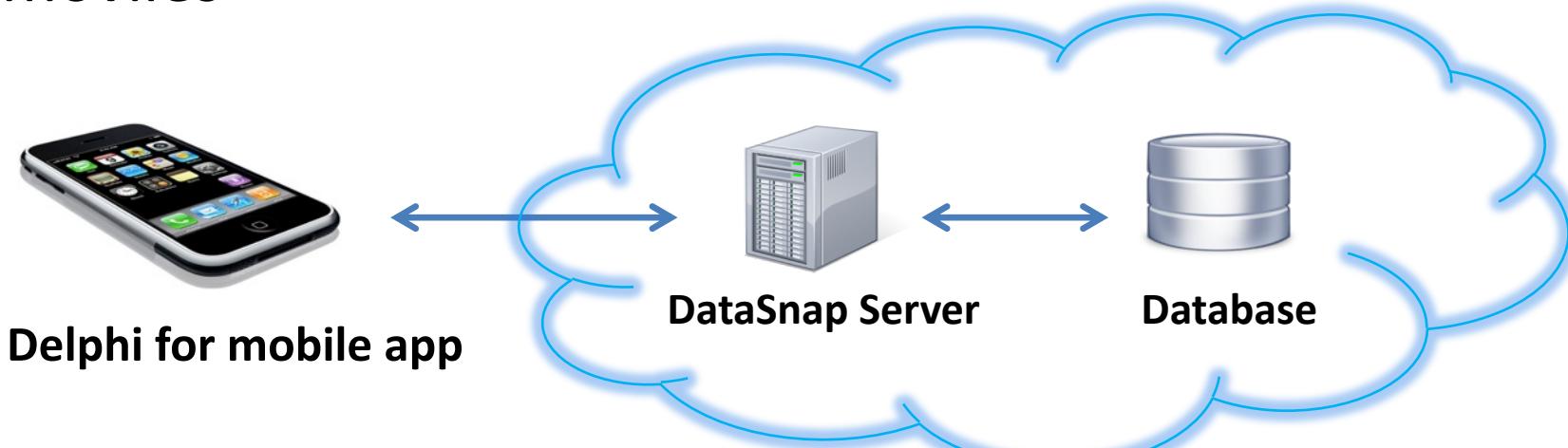
- **TWebBrowser** for embedded web browsing
- **TMagnifier** used with text editing, but can be used with other components as well
- **TCamera** - access front/back camera, flash etc.
- **TListBox** for short lists with no or minimal scrolling, header/footer/group/search styling
- **TListView** for long, scrollable, data-bound lists
- **TSwitch** for on/off selections
- **TProgressBar** for flexible progress control visualization
- **TSpeedButton** with variety of iOS-like icon options
- ... and more!

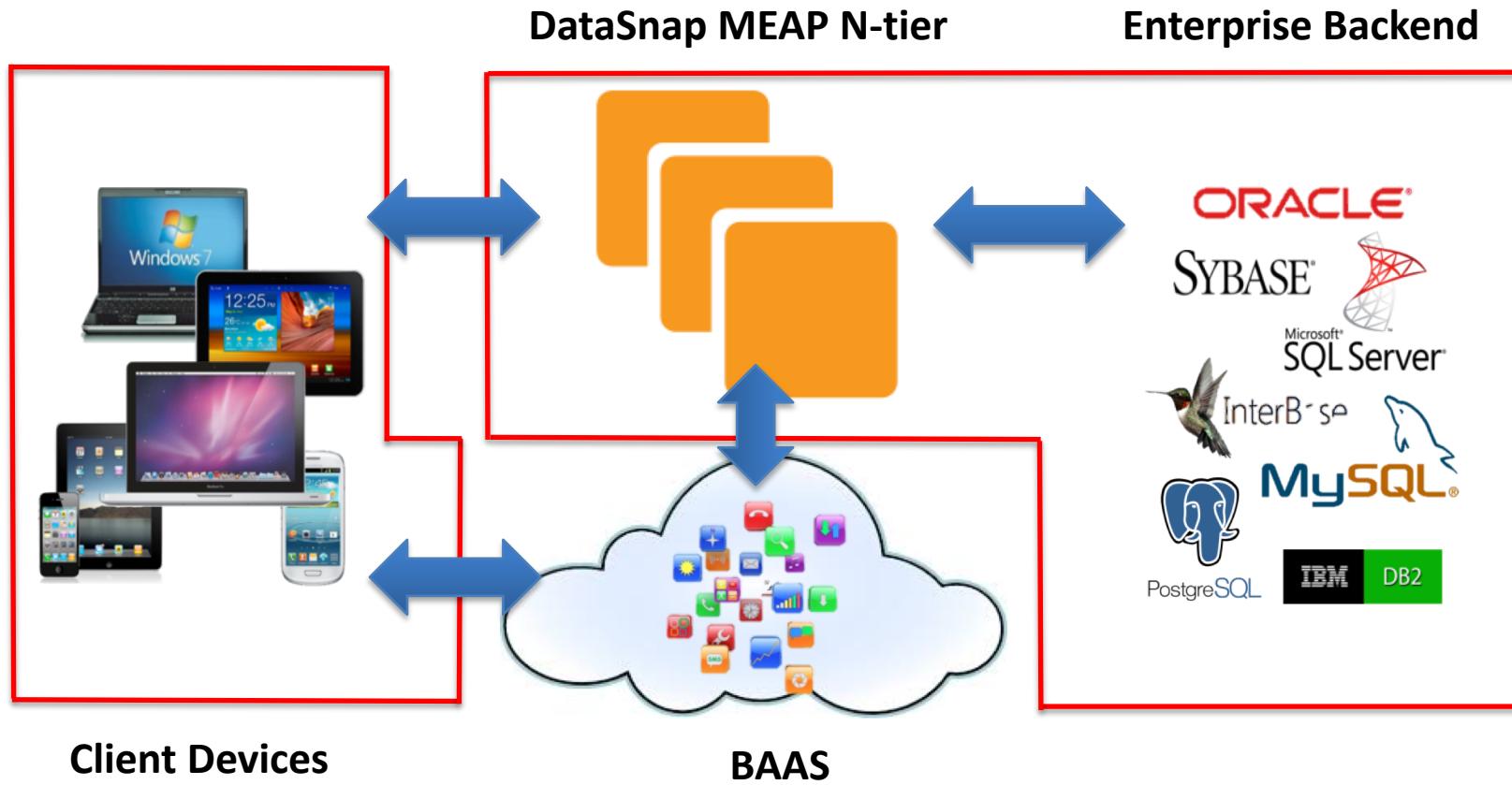
Preparado para Desarrollo Empresarial



Multicapas con DataSnap

- Acceso remoto a través de TCP/IP, REST/JSON o SOAP
- Connectándose a data Enterprise desde dispositivos móviles





FireDAC

- A set of Universal Data Access Components
 - for developing any database application
 - for Delphi and C++Builder
- High-performance, easy-to-use, enterprise connectivity
- Universal Data Access
 - But with many database specific features

Local Databases

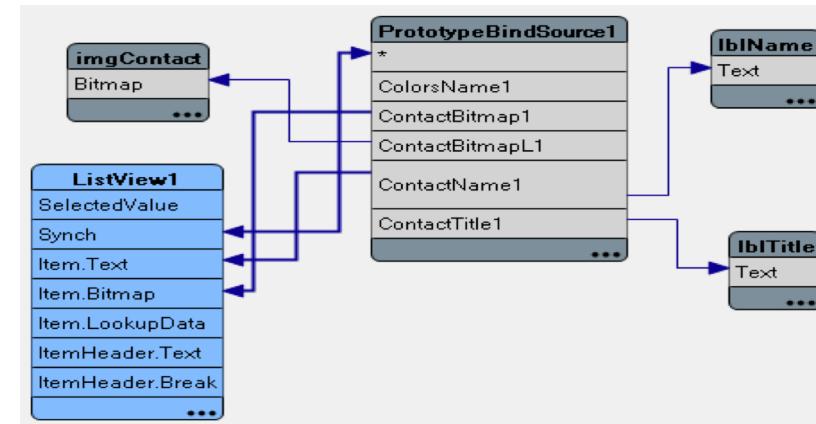
SQLite	IBLite	InterBase ToGo
Free	Free	Commercial
Feature light	Feature light	Fully featured
No security	No security	Secure Encryption
Simple Data Storage	Full SQL-92 RDBMS	Full SQL-92 RDBMS
Single read/write	Fast multi read/write	Fast multi read/write



Visual LiveBindings



- Bind controls to data
- Rapid Prototyping

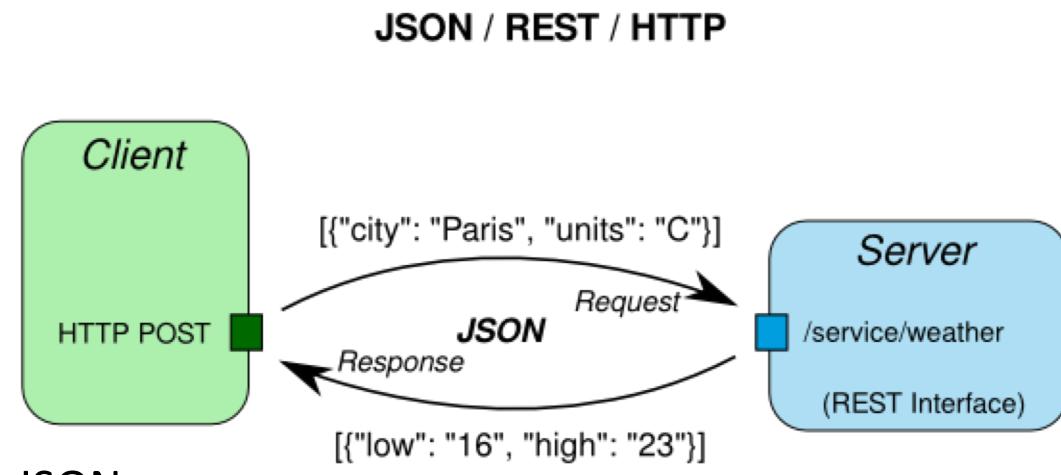


XE5 REST Client Stack

- REST components
 - For developing REST client application
- RESTDemos.exe
 - Uses REST components to access a few different providers
- RESTDebugger.exe
 - Uses REST components to execute ad hoc requests

REST component features

- Comprehensive HTTP client
 - Asynchronous execution
 - Proxy connection
 - HTTPS
- Authentication
 - Basic, OAuth1, OAuth2
- JSON
 - Parsing, Formatting
 - JSON to TObject, TObject to JSON
- Rapid Prototyping
 - LiveBindings
 - Design time execution





Resumen

Resumen

Multi-Dispositivo significa que usted no tiene que tener múltiples proyectos de desarrollo para entregar su aplicación nativa en múltiples plataformas (Android, iOS, Windows y Mac).

Desarrollo Nativo permite desplegar aplicaciones optimizadas para el hardware del dispositivo de una forma rápida, y atractiva para el usuario final.

La Forma mas rápida de crear aplicaciones visualmente para iOS y Android, usando un framework completo y optimizado.

Seguridad Implícita es construir aplicaciones nativas que se ejecutan directamente en el dispositivo sin el uso de interpretes ni Máquinas Virtuales..

Plataforma Enterprise para Múltiples Dispositivos significa que con la obtención de RAD Studio XE5, usted ya tendrá acceso a conectividad con la mayoría de las bases de datos Empresariales, desarrollo Multi-Capas, interacción con Servicios Clouds, Web Services, REST/JSON y mucho mas...



Muchas Gracias!

Fernando Rizzato

Fernando.rizzato@embarcadero.com