



# Embarcadero Conference

Um único esforço, uma única base de código, múltiplas  
plataformas, múltiplos dispositivos



Victory Fernandes

# Monetização de Aplicações iOS e Android





# Victory Fernandes

- Delphi desde 1998
  - Autor da ActiveDelphi, TheClub e ClubeDelphi
  - Embarcadero MVP e Consultor de Treinamentos
- 
- Engenheiro, Diretor Técnico da TKS Software
  - Professor de Engenharia da UNIJORGE
  - Mestrando em Medicina - Neurociências





Já implementou apps iOS ou  
Android com RAD Studio?





Já subiu apps para AppleStore ou  
Android-Market?







**LOJAS SEM FRONTEIRAS!**  
**70/30**



# Modelo Tradicional

- Usuários compram o app
  - App tem apelo de compra
  - Usuário compra app apenas uma vez
  - Novos usuários precisam ser conquistados



# Modelo tradicional

## #1 Angry Birds

The Angry Birds phenomenon would be nowhere if not for the original version which premiered in December of 2009 and captured our angry, angry hearts ever since. The game has since expanded its franchise to various other platforms, including apparel, food, and even a theme park.

**Price:** \$0.99



# Modelo tradicional

## #2 Fruit Ninja

In Fruit Ninja you slice fruit that jumps on the screen.

**Price:** \$0.99



# Modelo tradicional

## #6 WhatsApp

WhatsApp wasn't even in the top 10 last year when Apple put out its best selling apps list. Now it's number six. An incredible success for this messaging app.

**Price:** \$0.99



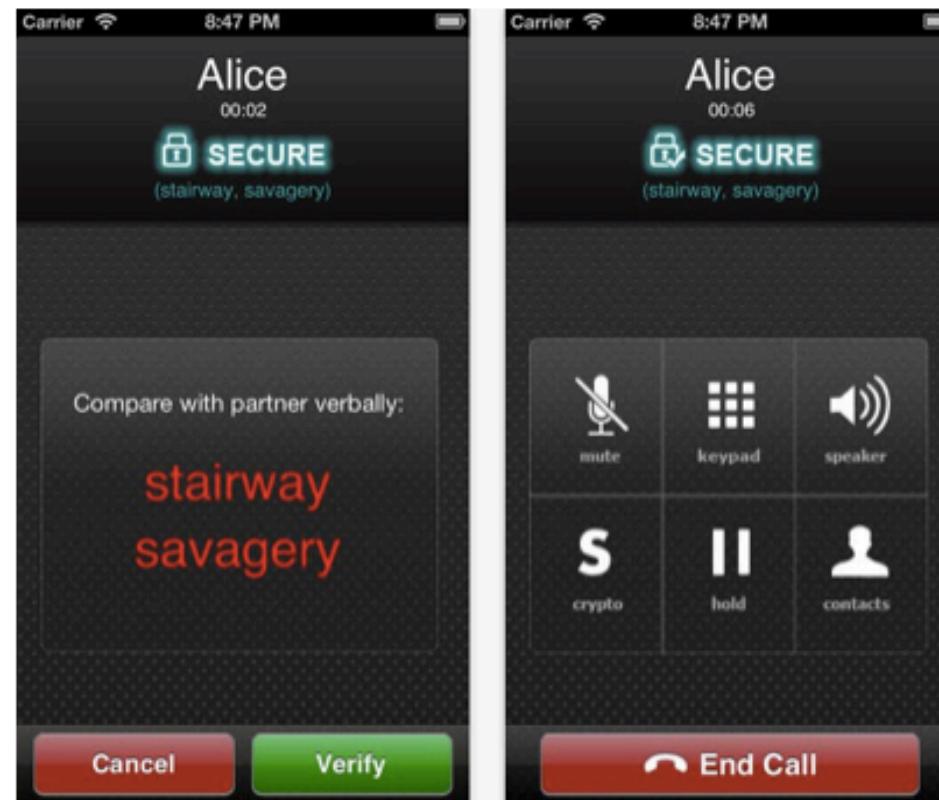


# Modelo tradicional

## \$299.99 – SafeSession Voice

SafeSession Voice encryption is an app that lets you make secure VoIP calls over the internet. The company says your calls are protected by voice encryption algorithms. The person you're talking to also needs to have the app installed for you to have a secure conversation.

**Price:** \$299.99

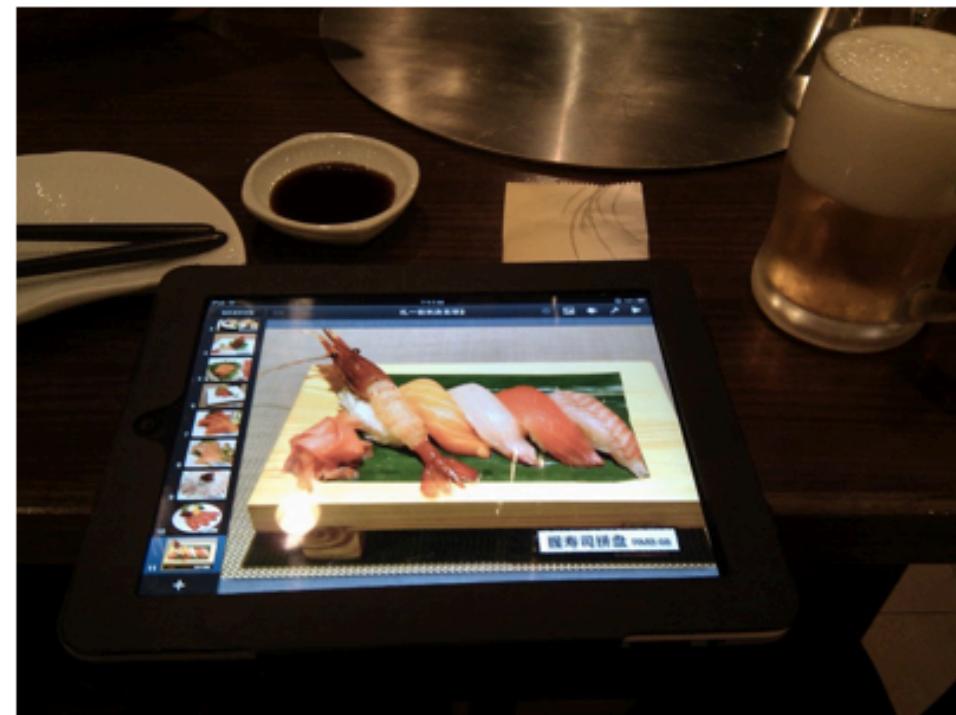


# Modelo tradicional

**\$399.99 – Tap Menu**

Tap Menu is a picture-based app for restaurants, hotels, shops, and other companies that makes your menu, book, or any other medium digitally visual.

**Price:** \$399.99

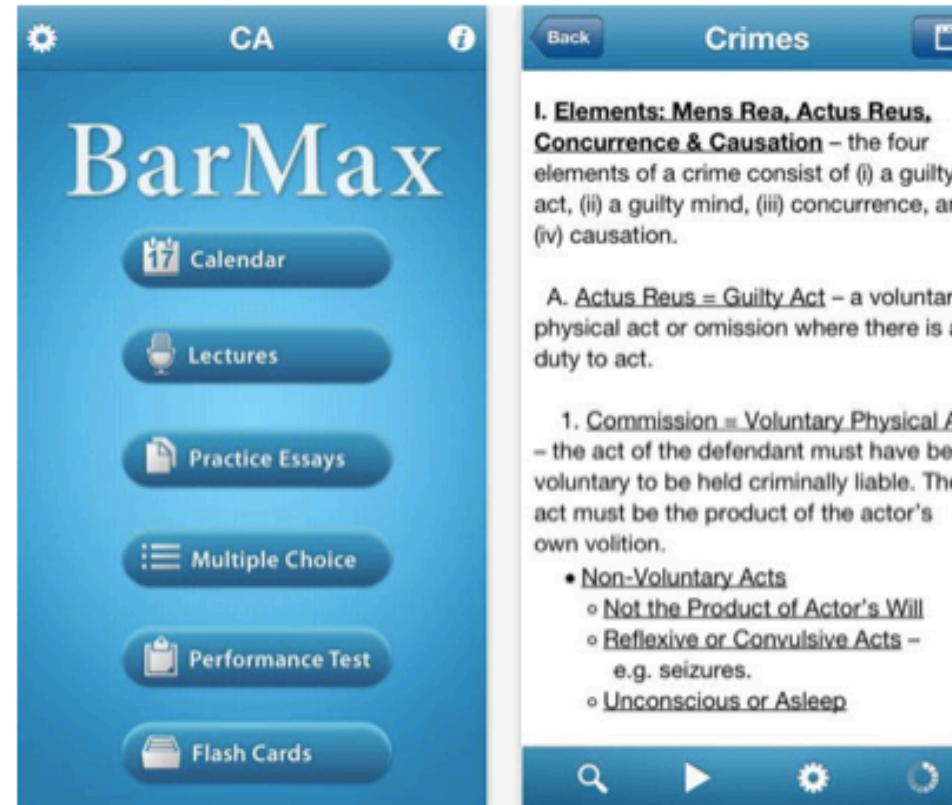


# Modelo tradicional

**\$999.99 – BarMax CA**

BarMax CA was created by a Harvard Law School alumni to help aspiring lawyers pass the California bar exam. There are more than 50 hours of audio lectures from Harvard Law School-educated law professors, and a ton more information to help students ace the test.

**Price:** \$999.99





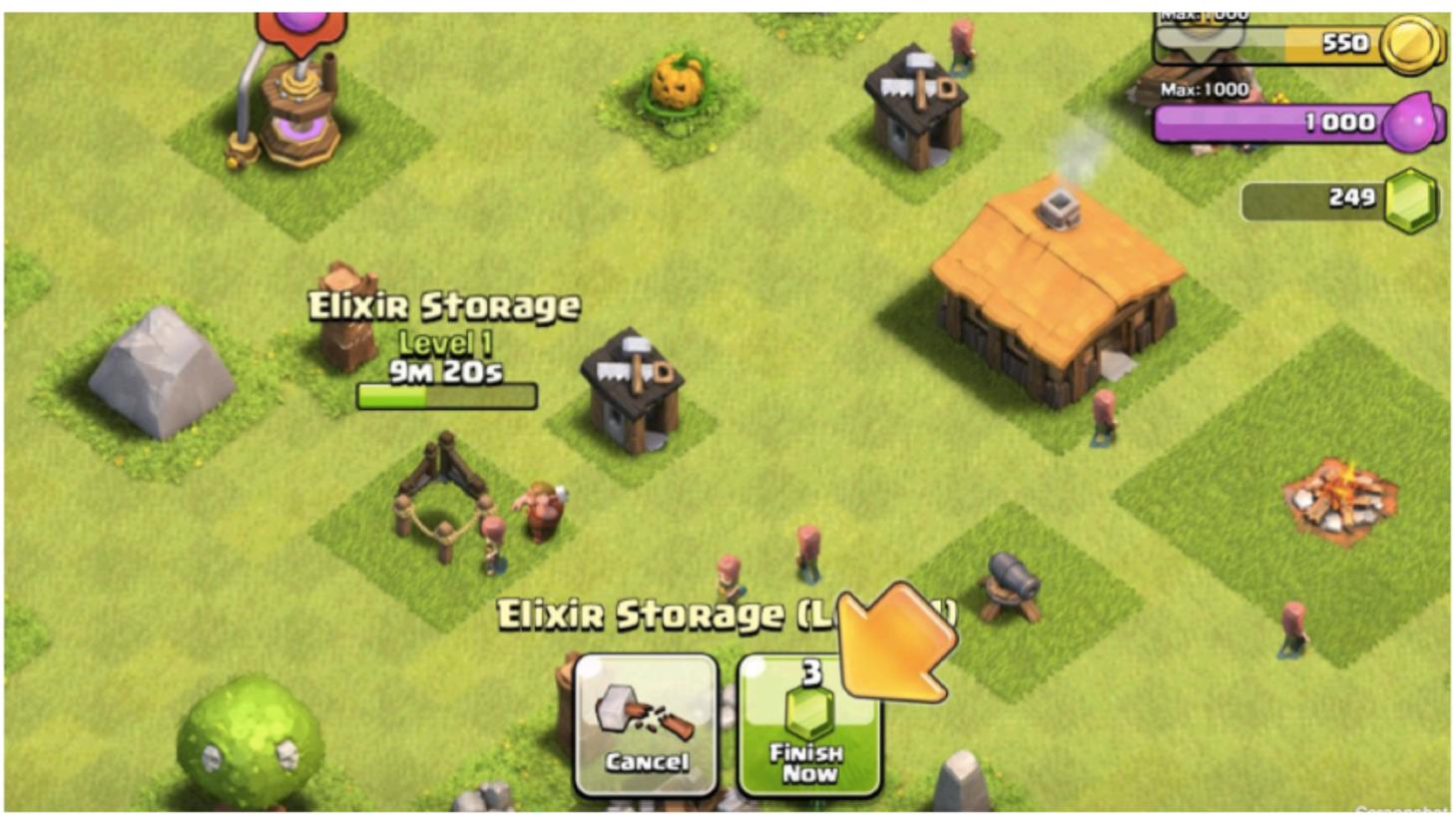
# Modelo in-App (freemium)

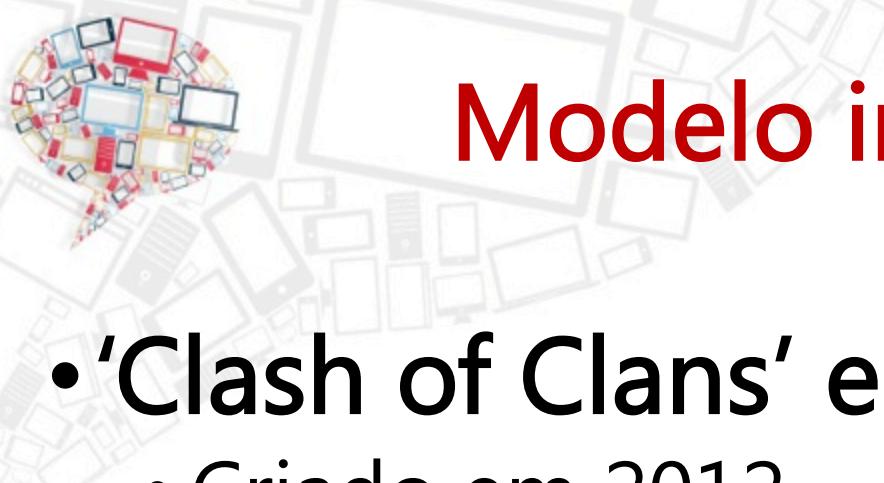
- Usuários podem fazer compras dentro do app
  - Desbloquear recursos ou funcionalidades
  - Remover anúncios
  - Itens podem ser duráveis ou consumíveis
  - Usuário se mantém comprando





# CLASH of CLANS

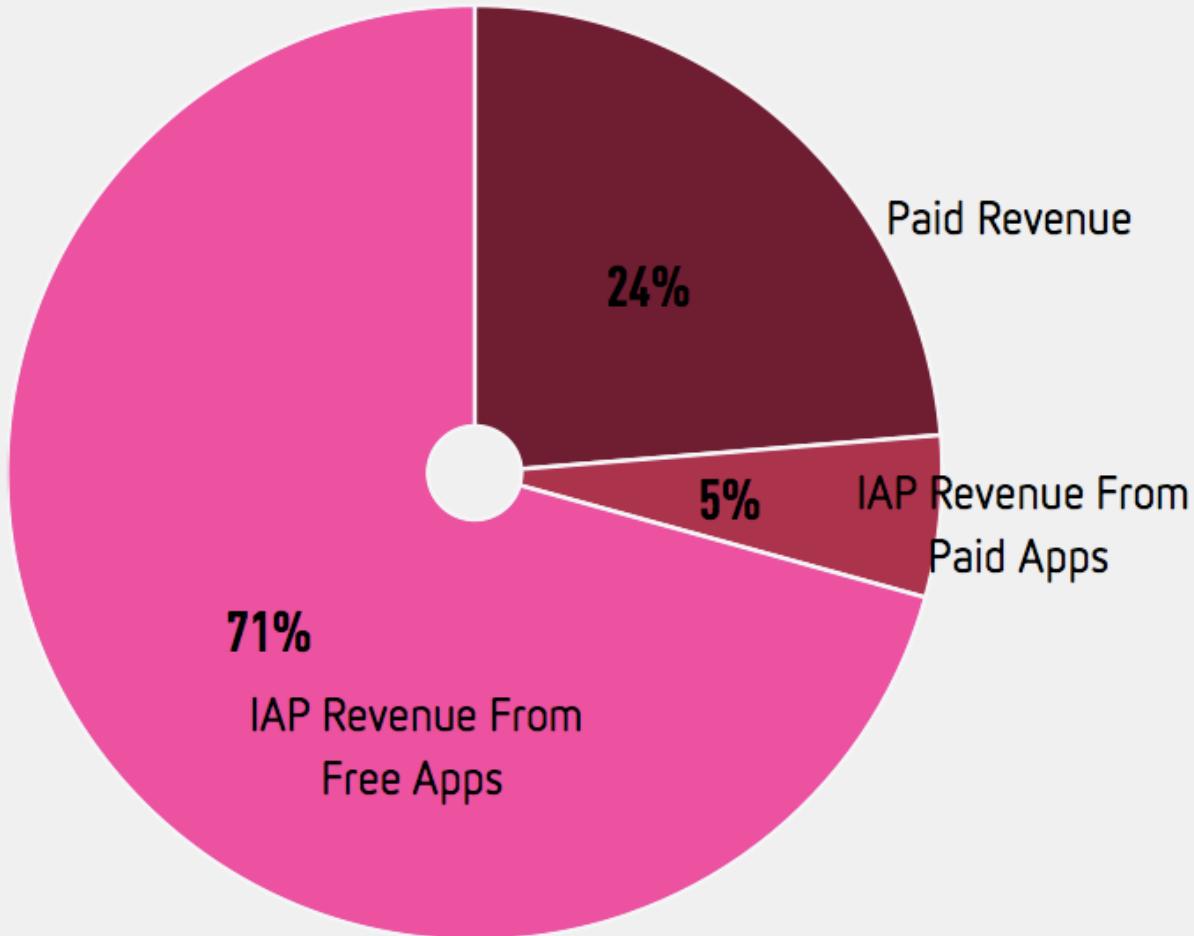




# Modelo in-App (freemium)

- 'Clash of Clans' em números
  - Criado em 2012
  - Chegou rapidamente ao topo da lista
  - Em 2013, faturamento \$892 milhões
  - Em 2013, Até \$5 milhões em um único dia

# Modelo in-App (freemium)



Top most grossing iPhone applications in the U.S. in February, which were released since January 2012

HOW THE MOST SUCCESSFUL APPS MONETIZE THEIR USER BASE, By Hendrik Koekkoek, Analyst at Distimo; Mar, 2013

# Modelo in-App (freemium)

## Top Fastest Growing Countries In Terms Of Revenue In The Apple App Store for iPhone

Based on revenue of top 200 highest grossing apps, top 30 countries are analyzed, May 2011 - May 2012

Rank	Country	Year-on-year growth
1	Japan	560%
2	Russia	115%
3	China	109%
4	Taiwan	101%
5	Thailand	84%
6	Brazil	83%
7	Mexico	63%
8	Korea	59%
9	Turkey	54%
10	Canada	47%

DISTIMO



# Modelo Banner



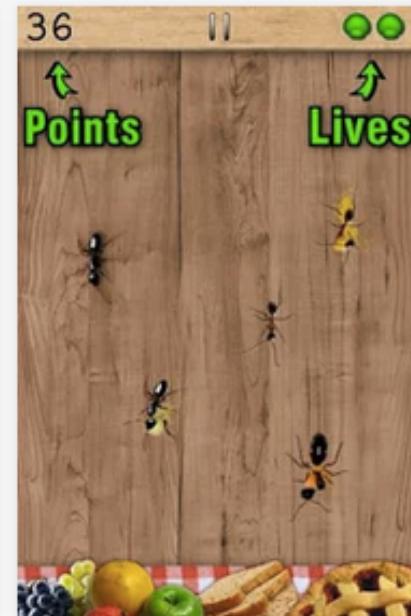
## Ant Smasher, Best Free Game

Best Cool & Fun Free Games - October 1, 2014  
Arcade

Install



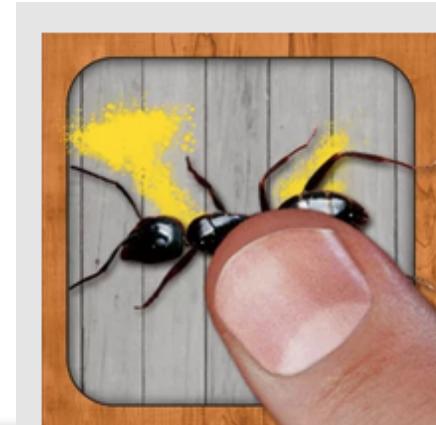
Add to Wishlist



- Milionário Mundo dos Aplicativos – por Mariana Queiroz Barbosa
- ISTOÉ - ECONOMIA & NEGÓCIOS | N° Edição: 2209 | 09.Mar.12

# Modelo Banner

- “Ant Smasher” - Desenvolvido em 2010 pelo paulista Guilherme Schvartsman (24 anos)
  - <https://www.linkedin.com/in/schvartsman>
- App gratuito
- 30 milhões de downloads
- Receita de \$ 3milhões/ano
- Em 2011 recusou proposta de compra de \$ 50 milhões





# Modelo Sob-Demandra

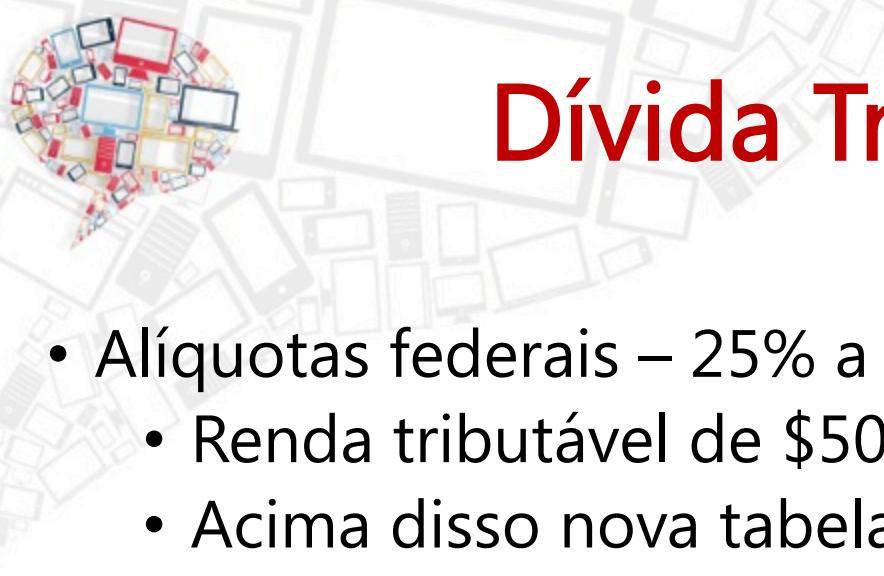
- Empresas pagam por seus apps
  - Desenvolvimento e atualização do app
  - Geração e administração de conteúdo
  - Modelo mais próximo do “convencional”!





# Dívida Tributária nos EUA

- Intrincado problema tributário
    - A quem pagar os impostos sobre faturamento?
    - No seu país, no país do servidor da loja ou no país onde o cliente fez o download?
  - Apple e Google reportam ao IRS-EUA desenvolvedores com faturamento > \$20mil
  - Passivo tributário junto ao IRS-EUA
    - Equivalente a nossa Receita Federal
  - Empresas descobrem o passivo ao abrir estritório nos EUA
- 
- Desenvolvedores brasileiros de apps desconhecem dívida tributária nos EUA – por Fernando Paiva
  - Mobile Time - 29/09/2014 às 10h03



# Dívida Tributária nos EUA

- Alíquotas federais – 25% a 34%
  - Renda tributável de \$50mil a \$10milhões por ano
  - Acima disso nova tabela com início em 35%
- Alíquotas municipais e estaduais de 0% a 15%
  - Dependem do local onde o download foi feito



# in-App no Delphi e C++Builder

uses

```
System.SysUtils, System.Types, System.UITypes,  
FMX.Types, FMX.Controls, FMX.Forms, FMX.Dialog  
FMX.Advertising, FMX.InAppPurchase, System.Gen
```

private

```
FWest: Boolean;  
FEast: Boolean;  
FCentral: Boolean;  
FEurope: Boolean;  
FBannerAd: TBannerAd;  
FInAppPurchase: TInAppPurchase;
```

# in-App no Delphi e C++Builder

```
TInAppPurchase = class(TCustomInAppPurchase)
published
  property ApplicationLicenseKey;
  property ProductIDs;
  property OnProductsRequestResponse;
  property OnError;
  property OnVerifyPayload;
  property OnDownloadProgress;
  property OnDownloadCompleted;
  property OnPurchaseCompleted;
  property OnRecordTransaction;
  property OnConsumeCompleted;
  property OnConsumeFailed;
end;
```

# in-App no Delphi e C++Builder

```
TCustomInAppPurchase = class(TComponent, IInAppPurchaseCallbacks)
private
  FProductIDs: TStrings;
  FIIsSetupComplete: Boolean;
  FInAppPurchaseService: IFMXInAppPurchaseService;
  FOnSetupComplete: TNotifyEvent;
  FOnProductsRequestResponse: TIAPPProductsRequestResponseEvent;
  FOnError: TIAPErrorEvent;
  FOnVerifyPayload: TIAPVerifyPayloadEvent;
  FOnDownloadProgress: TIAPDownloadProgressEvent;
  FOnDownloadCompleted: TIAPDownloadCompletedEvent;
  FOnPurchaseCompleted: TIAPPurchaseCompletedEvent;
  FOnRecordTransaction: TIAPRecordTransactionEvent;
  FOnConsumeCompleted: TIAPConsumeCompletedEvent;
  FOnConsumeFailed: TIAPConsumeFailedEvent;
  //Ensures IAP service has been setup first
  procedure CheckInAppPurchaseIsSetup;
  procedure SetProductIDs(const Value: TStrings);
  function GetApplicationLicenseKey: string;
  procedure SetApplicationLicenseKey(const Value: string);
  function GetTransactionPayload: string;
  procedure SetTransactionPayload(const Value: string);
protected
  { IInAppPurchaseCallbacks }
  procedure DoSetupComplete;
  procedure DoProductsRequestResponse(const Products: TIAPPProductList; const InvalidProductIDs: TStrings);
  procedure DoError(FailureKind: TFailureKind; const ErrorMessage: string);
  procedure DoVerifyPayload(const Payload: String; var PayloadOK: Boolean);
  procedure DoPurchaseCompleted(const ProductID: string; NewTransaction: Boolean);
  procedure DoRecordTransaction(const ProductID, TransactionID: string; TransactionDate: TDateTime);
  procedure DoDownloadProgress(const ProductID, ContentID: string; TimeRemaining: Double; Progress: Single);
  procedure DoDownloadCompleted(const ProductID, ContentID, FilePath: string);
  procedure DoConsumeCompleted(const ProductID: string);
```

# in-App no Delphi e C++Builder

```
{$IF Defined(iOS)}  
FMX.InAppPurchase.iOS,  
{$ELSEIF Defined(ANDROID)}  
FMX.InAppPurchase.Android,
```

- Referência iOS
  - [http://developer.apple.com/library/ios/#documentation/StoreKit/Reference/StoreKit\\_Collection/\\_index.html](http://developer.apple.com/library/ios/#documentation/StoreKit/Reference/StoreKit_Collection/_index.html)
- Referência Android
  - <http://d.android.com/google/play/billing>

# in-App no Delphi e C++Builder

- C:\Program Files\Embarcadero\Studio\15.0\source\fmx



`FMX.InAppPurchase.Android`



`FMX.InAppPurchase.iOS`



`FMX.InAppPurchase`

# in-App no Delphi e C++Builder

- <http://embt.co/xe6-in-app-payments>

The screenshot shows a web browser displaying a wiki page from [docwiki.embarcadero.com/RADStudio/XE6/en/Adding\\_In-App\\_Payments\\_to\\_Your\\_Mobile\\_Apps](http://docwiki.embarcadero.com/RADStudio/XE6/en/Adding_In-App_Payments_to_Your_Mobile_Apps). The page title is "Adding In-App Payments to Your Mobile Apps". The left sidebar contains a navigation menu for RAD Studio XE6, including sections like Examples and Samples, Libraries, Tutorials, FireMonkey Application Platform, Multi-Device Applications, Getting Started, Steps in Developing a Project, Key Application Types, FireDAC, Windows Developer's Guide, Modeling Tools, IDE Reference and Command-Line Utilities, Delphi Reference, and C++ Reference. The main content area starts with a note about adding support for in-app payments to an application. It includes links to Google Play In-app Billing (Android) and iOS In-App Purchase, with the Android link highlighted by a red box. Below this, it discusses in-app payment services and the TInAppPurchase component. A section titled "Establishing a Connection to Your In-App Payment Service" is shown at the bottom.

docwiki.embarcadero.com/RADStudio/XE6/en/Adding\_In-App\_Payments\_to\_Your\_Mobile\_Apps

Preferences Watchlist Contribut...

## Adding In-App Payments to Your Mobile Apps

[Go Up to Creating an Android App](#)

[Go Up to Creating an iOS App](#)

**Note:** This topic describes how to add support for in-app payments to an application, regardless of the target in-app payment service. For service-specific documentation prerequisites, testing, deployment and other aspects of adding in-app payments support to your application, read the following pages:

> [Google Play In-app Billing \(Android\)](#)

> [iOS In-App Purchase](#)

In-app payment services let you sell digital content directly within your applications. **Google** and **Apple** provide in-app payment services for **Google Play (Android)** and **iOS** respectively.

FireMonkey provides the `TInAppPurchase` component that abstracts your applications from the particularities of different in-app payment services.

You need to add an instance of this component to your application, and establish a connection to the target in-app payment service either when your application starts or point. Then you can use the service. Once you establish a connection to your in-app payment service, you can:

> Request information about the products of your application. Your application should do this at least once after you connect to your in-app payment service.

> Purchase products.

> Consume products. Process consumable products in your application, and let the in-app payment service know that your application consumed them.

> Restore products that your user purchased from a different device on the current device. You should do this when your user requests it; for example, your application "Restore" button for this.

`TInAppPurchase` also provides some service-specific features that you can use as well.

### Establishing a Connection to Your In-App Payment Service

[edit]

Before you can use your in-app payment service, you must configure `TInAppPurchase` to connect to your in-app payment service and establish a connection.

**Contents [hide]**

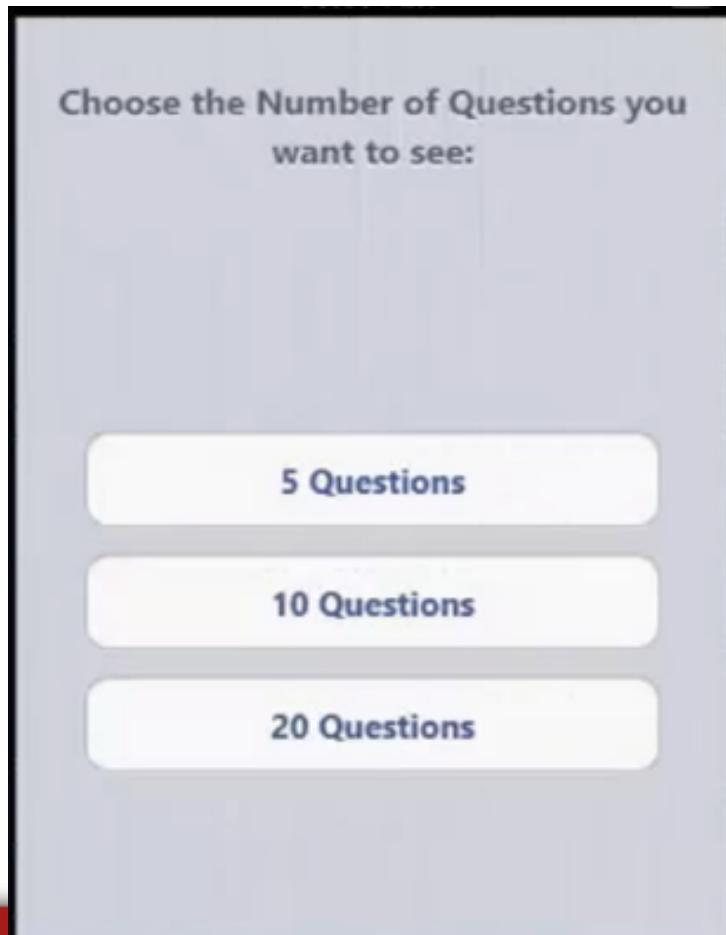
1 Establishing a Connection to Your In-App Payment Service

1.1 Configuring the Connection Data for Your In-App Payment Service

1.2 Connecting to Your In-App Payment Service

# in-App no Delphi e C++Builder

- C:\Users\Public\Documents\Embarcadero\Studio  
\15.0\Samples\Object Pascal\Mobile Samples\Device Sensors  
and Services\CapitalIAP



# in-App no Delphi e C++Builder

```
const
  EuropeID = 'com.embarcadero.capitals.europe';
  NoAdsID = 'com.embarcadero.capitals.noads';

procedure TMainForm.FormCreate(Sender: TObject);
begin
  Log.d('App executando...');
  FInAppPurchase := TInAppPurchase.Create(Self);

  // License Key somente no Android (obtido no Google Dev. Console)
  FInAppPurchase.ApplicationLicenseKey := '';
  FInAppPurchase.ApplicationLicenseKey := FInAppPurchase.ApplicationLicenseKey + '';

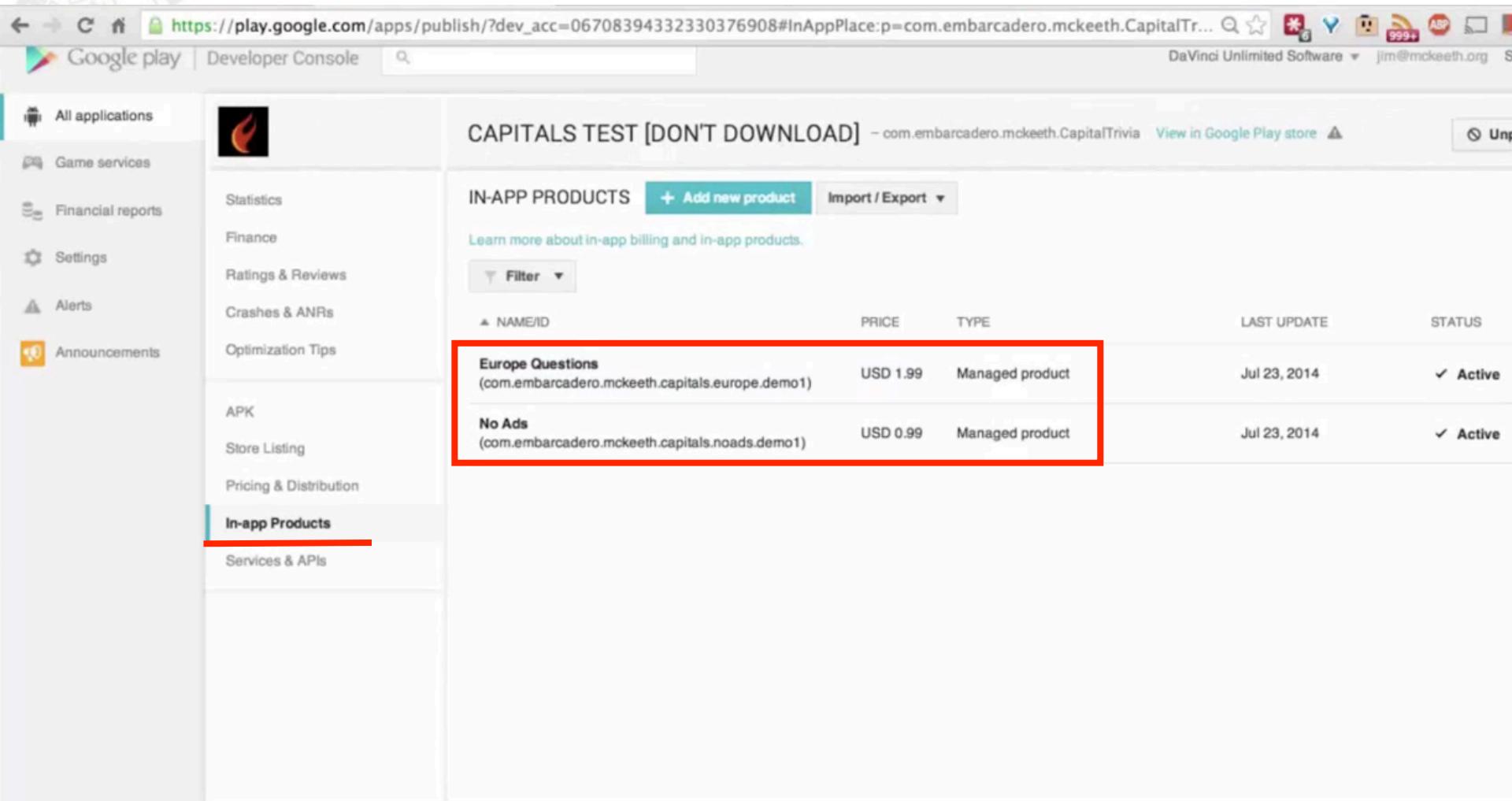
  FInAppPurchase.ProductIDs.Add(NoAdsID);
  FInAppPurchase.ProductIDs.Add(EuropeID);

  FInAppPurchase.OnSetupComplete      := InAppPurchaseSetupComplete;
  FInAppPurchase.OnProductsRequestResponse := InAppPurchaseProductsRequestResponse;
  FInAppPurchase.OnError           := InAppPurchaseError;
  FInAppPurchase.OnPurchaseCompleted := InAppPurchasePurchaseCompleted;

  Log.d('Inicializando IAP...');
  FInAppPurchase.SetupInAppPurchase;
```

# in-App no Delphi e C++Builder

- Quem veio primeiro o ovo ou a galinha?!



https://play.google.com/apps/publish/?dev\_acc=06708394332330376908#InAppPlace:p=com.embarcadero.mckeeth.CapitalTr... View in Google Play store

NAME/ID	PRICE	TYPE	LAST UPDATE	STATUS
Europe Questions (com.embarcadero.mckeeth.capitals.europe.demo1)	USD 1.99	Managed product	Jul 23, 2014	✓ Active
No Ads (com.embarcadero.mckeeth.capitals.noads.demo1)	USD 0.99	Managed product	Jul 23, 2014	✓ Active

# in-App no Delphi e C++Builder

- Quem veio primeiro o ovo ou a galinha?!
- Upload do app para obter o License Key!

https://play.google.com/apps/publish/?dev\_acc=06708394332330376908#ServicesAndApisPlace:p=com.embarcadero.mckeeth... View in Google Play store

**CAPITALS TEST [DON'T DOWNLOAD]** - com.embarcadero.mckeeth.CapitalTrivia Unpub

**SERVICES & APIs**

**GOOGLE CLOUD MESSAGING (GCM)**

Google Cloud Messaging (GCM) is a service that helps you to send data from your servers to your applications. [Learn more](#)

To access the GCM stats for your application, you need to link a GCM sender ID that you use for this application by providing your GCM API key.

Once your app is published, you can access the GCM statistics for your application from the statistics page.

**Link a sender ID**

**LICENSING & IN-APP BILLING**

Licensing allows you to prevent unauthorized distribution of your app. It can also be used to verify in-app billing purchases. [Learn more about licensing](#).

**YOUR LICENSE KEY FOR THIS APPLICATION**

Base64-encoded RSA public key to include in your binary. Please remove any spaces.

```
MIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAp5+qe6cMl0hf9DCLIGPnxnlPiSfuIWQXmk9o6KUdrw5SuASwoiIUYLsS3VBxkdsBG6RTSlgcjYJWKPoRh  
a7kDZYMBvt3P1X3s8RAgzk9qSzKrgw56V5dhJPHhcVAABYbNurYTSoBr6p2Mmva/81Y4z7KHgW9stFSAO3u3dQWLtY3WlvT17Xo9zj1eHLnOhB6Q+iIDJfynZ27J2  
TaY83NmxE5DmFETciCcJsanu33K5YUpdx+9Wvw1xGSLI3ewLhoVYlPutTaJ4o2HxpjInGPh5L8ubtX90rGM/8MG/8f02KyhLy0s5N4YC/atKMLHyge6KX68ng5AEa  
r+pvq6mQIDAQAB
```

GOOGLE PLAY GAME SERVICES

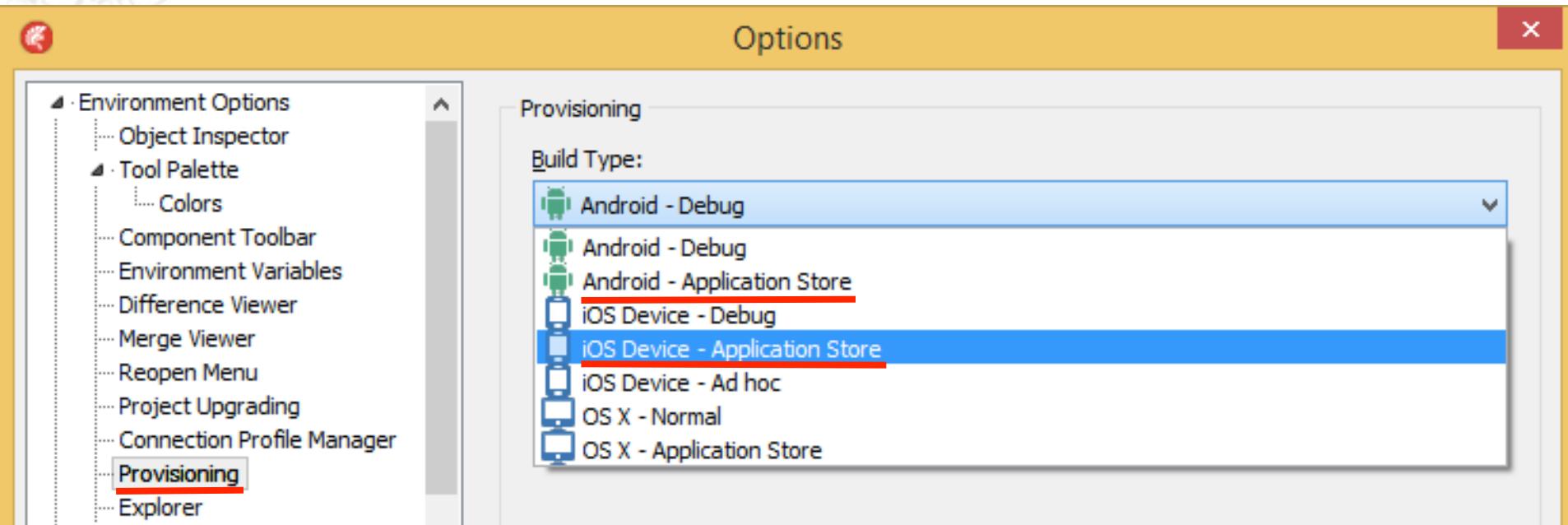


# Mais informações

- iTunes Connect
- <https://itunesconnect.apple.com/>
  
- Google Play Dev Console
- <https://play.google.com/apps/publish>

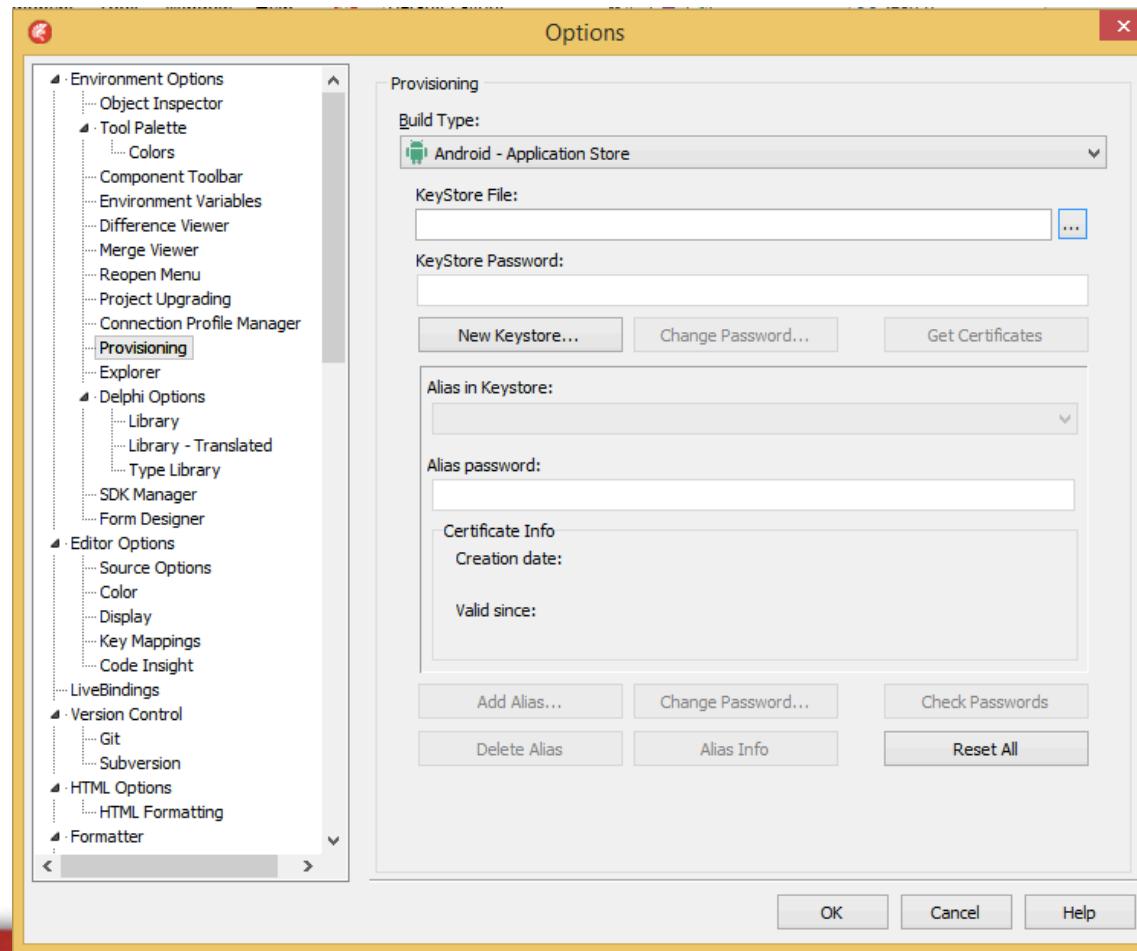
# in-App no Delphi e C++Builder

- Menu > Tools > Options > Provisioning



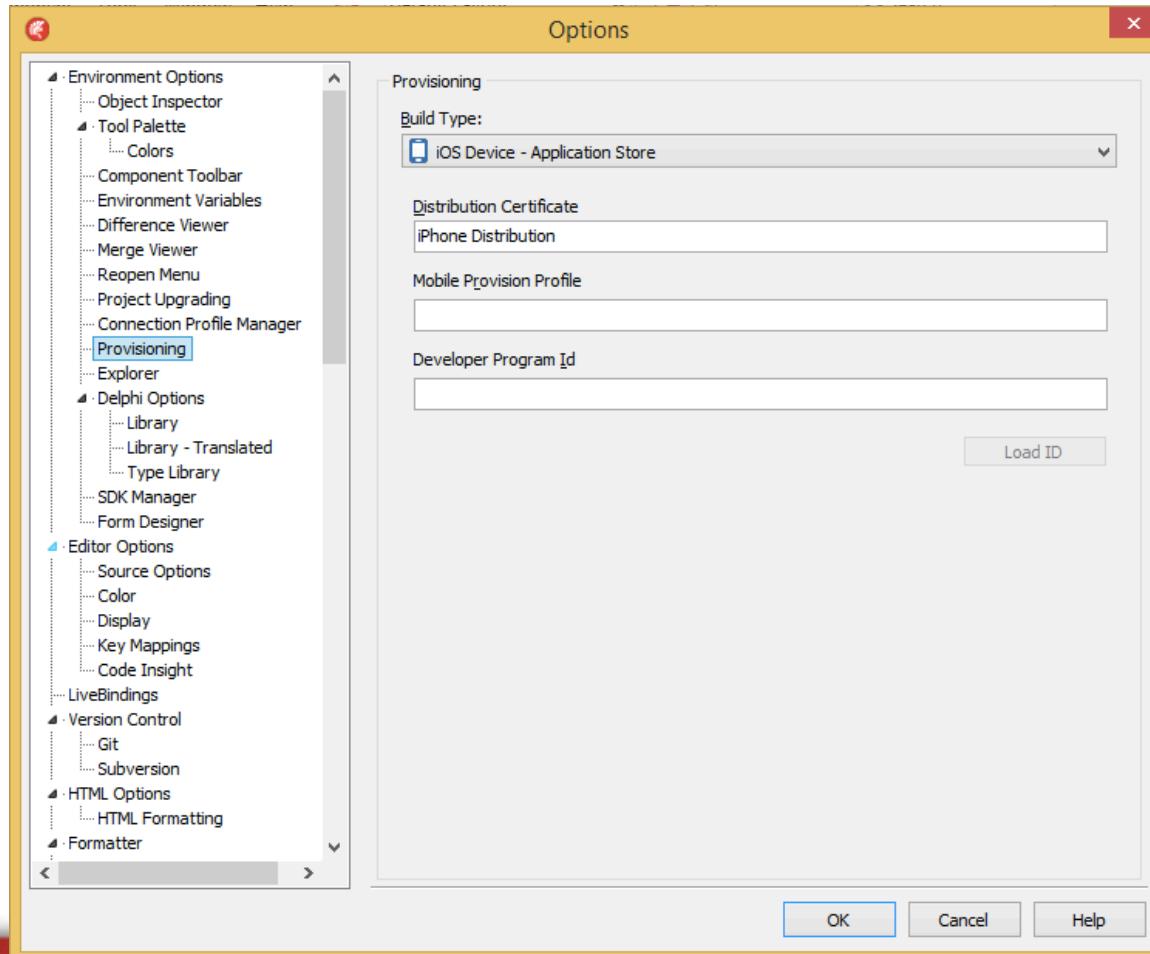
# in-App no Delphi e C++Builder

- Menu> Tools> Options> Provisioning> Build Type=Android-Application Store



# in-App no Delphi e C++Builder

- Menu> Tools> Options> Provisioning> Build Type=iOS-Application Store



# in-App no Delphi e C++Builder

- Project Manager > Options

The image shows the Delphi IDE interface. On the left, the Project Manager window is open, displaying a project named "CapitalTrivia.dpr". The "Build" option is selected in the context menu. On the right, the "Project Options" dialog is open for the file "libCapitalTrivia.so" under the "Android - Debug" configuration. The "Build Type" dropdown is set to "iOS Device platform", which is highlighted in blue. Other options in the dropdown include "Android platform", "OS X platform", "32-bit Windows platform", "64-bit Windows platform", "Debug configuration" (with "Android platform" selected), "Release configuration" (with "Android platform" selected), and "32-bit Windows platform", "64-bit Windows platform".



# Obrigado! Perguntas?

[vfernandes@embarcadero.com.br](mailto:vfernandes@embarcadero.com.br) [L SEP]

@victoryjorge [L SEP]

[www.tkssoftware.com/victory](http://www.tkssoftware.com/victory) [L SEP]

<https://www.youtube.com/user/victoryfernandes> [L SEP]