

# Desenvolvimento de jogos para iOS

Lucas Marinho

# Quem sou

- Co-fundador da Mobjoy
- Equipe que desenvolveu o Undead Attack!  
Pinball para iPhone
- 2 anos de experiência com a plataforma

# MOBJOY



**Cookie Rush**

- Versão iPhone
- Versão iPad
- Versão lite

**Undead Attack! Pinball**

- Versão iPhone
- Versão iPad
- Versão lite

# Do que vou falar

- Apresentar a plataforma
- Apresentar as ferramentas de desenvolvimento
- Falar da experiência de venda

# Por que iPhone OS?

- Plataforma extremamente popular
- Dispositivos poderosos e inovadores
- Facilidade de venda de aplicações
- Facilidade de desenvolvimento

# iPhone OS em 2007



- iPhone lançado em 6 países
- Lançamento do iPod touch logo depois.
- Sucesso de vendas
- Plataforma bastante restrita

# iPhone OS em 2007

## **iPhone 1.0**

4,8 ou 16 GB

WiFi

Multitouch

Acelerômetro

OpenGL ES 1.1

Câmera

Bluetooth

Celular!



## **iPod Touch 1**

8,16 ou 32 GB

WiFi

Multitouch

Acelerômetro

OpenGL ES 1.1



# iPhone OS em 2008



- iPhone 3G lançado em 22 países
- Lançamento do iPod touch 2ª geração
- iPhone OS 2.0
- SDK liberado
- App Store “Gold Rush”
- iPhone chega ao Brasil

# iPhone OS em 2008

## iPhone 3G

8 ou 16 GB

WiFi

Multitouch

Acelerômetro

OpenGL ES 1.1

Câmera

Bluetooth

Celular!

+3G

+GPS

+App Store



## iPod Touch 2

8,16 ou 32 GB

WiFi

Multitouch

Acelerômetro

OpenGL ES 1.1

+Bluetooth

+Speaker

+App Store

+Clock

# iPhone OS em 2009



- iPhone 3G S lançado
- Plataforma disponível em mais de 80 países
- Lançamento do iPod touch 3ª geração
- iPhone OS 3.0
- iPod Touch apresentado como plataforma de gaming

# iPhone OS em 2009

## iPhone 3G S

16 ou 32 GB

WiFi

Multitouch

Acelerômetro

+OpenGL ES 2.0

+Câmera 3.0MP

Bluetooth

Celular!

3G

GPS

App Store

+Rápido

+Bússola



## iPod Touch 3

32 ou 64 GB

WiFi

Multitouch

Acelerômetro

+OpenGL ES 2.0

Bluetooth

Speaker

App Store

+Rápido

# iOS em 2010



- iPad lançado
- iPhone 4 lançado
- iPod Touch 4
- Rebranding da plataforma para iOS
- Apple TV lançada
- Game Center

# iPad em 2010



- Anunciado no começo do ano
- “iPhonão”
- Chip Apple A4
- 1024x768
- Modelos Wifi e 3G

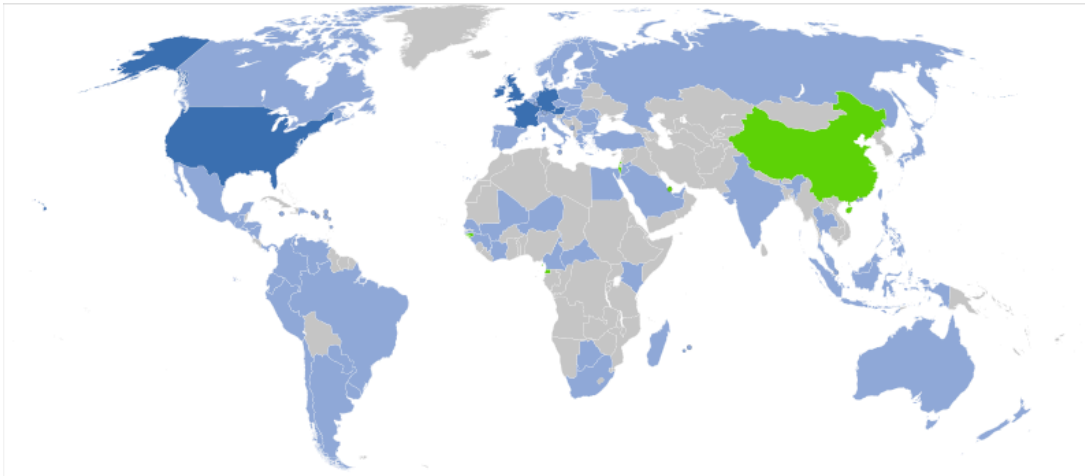
# iPhone em 2010



- iPhone 4 com novo design
- “Retina Display” = 640x960, IPS, LED backlit
- Chip Apple A4
- 512MB de RAM
- Giroscópio
- Multitasking
- Câmera melhorada

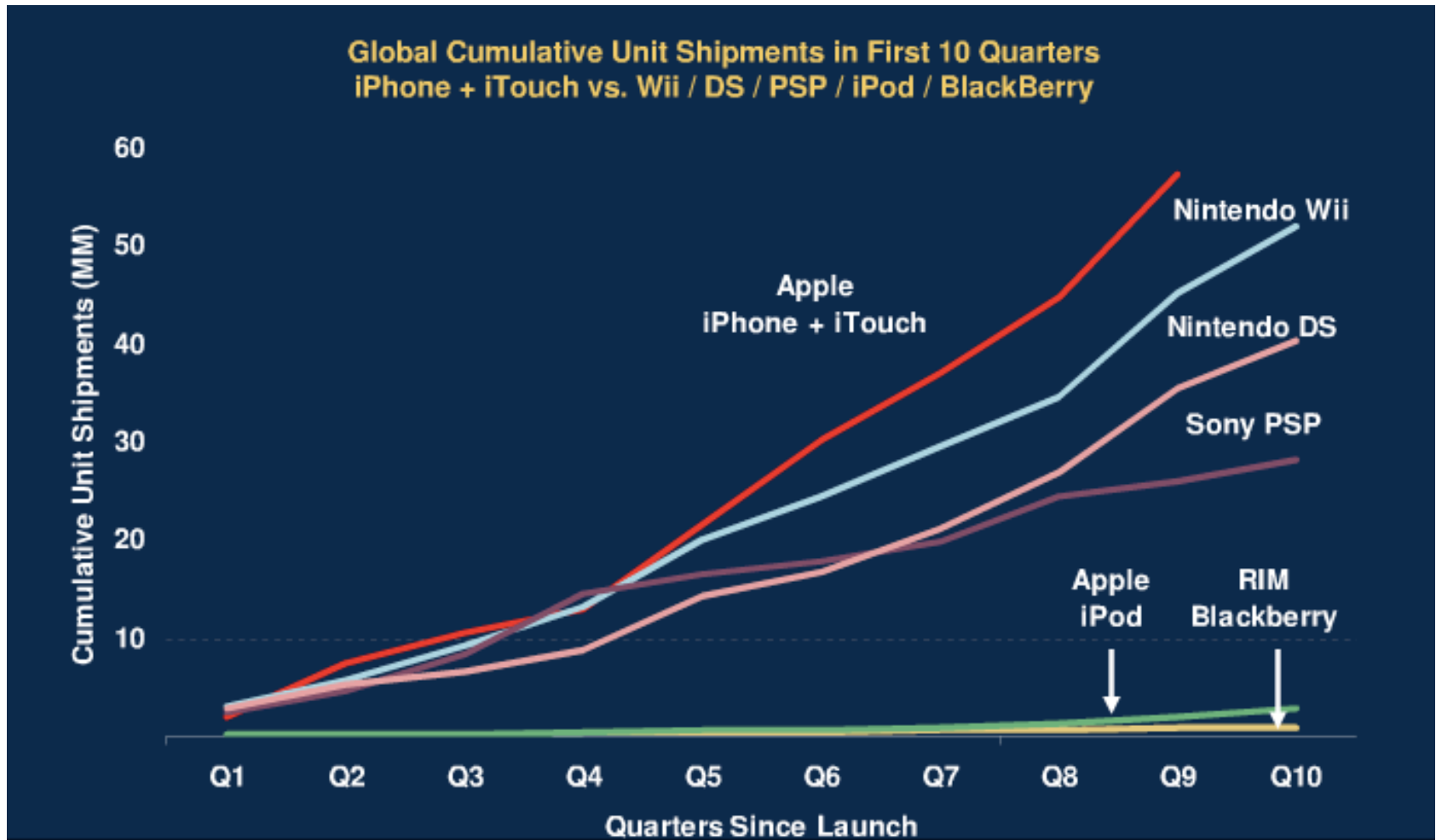
# iOS

- Mais de 120 milhões de dispositivos vendidos
- 75% dos usuários baixam aplicações
- Alcance global
- iPod Touch e iPad estabelecidos como plataformas de jogos





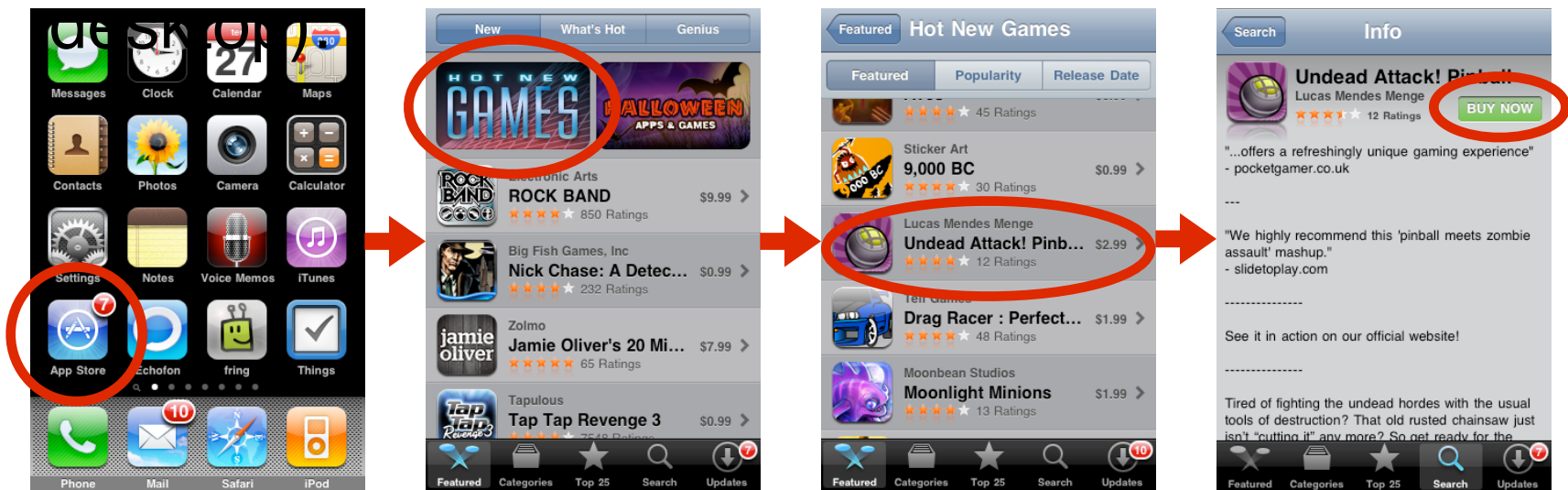
# iOS



Mary Meeker (Morgan Stanley)

# App Store

- Sistema de venda e distribuição de aplicações para iPhone OS
- Única forma(legal) de se adquirir aplicações para a plataforma
- Acesso fácil e direto pelo celular ou iTunes



# App Store

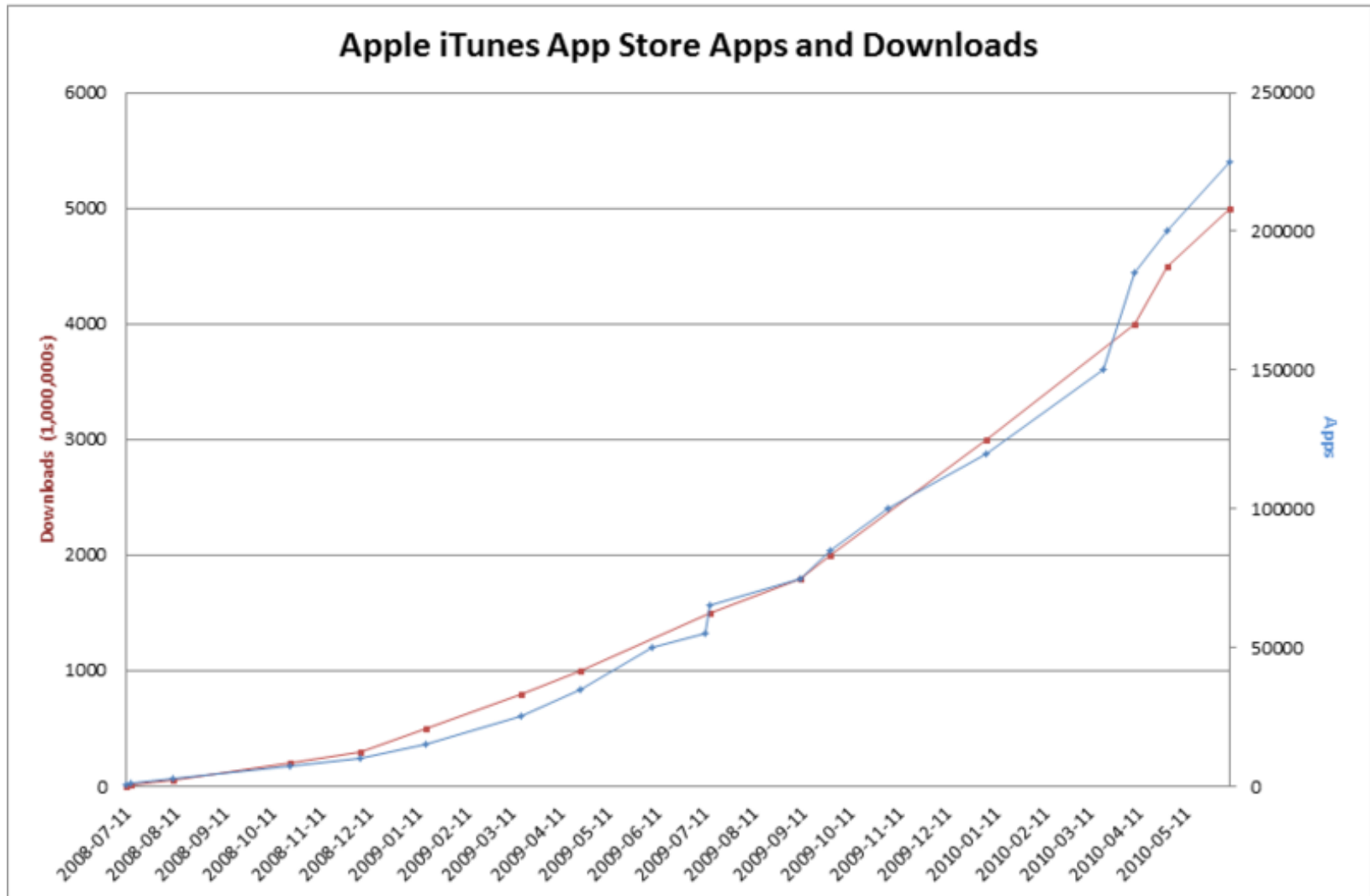
- Oferece listagem de aplicações por categoria, ordenadas por data de release ou número de vendas recentes.
- Permite aos usuários dar notas às aplicações e escrever reviews.
- Oferece sugestões de aplicações similares através do Genius.
- Tem listas especiais(features) de aplicações selecionadas pela Apple
- A exposição de uma aplicação nessas listas afeta as vendas profundamente.

# App Store em Números

- Mais de 270 mil aplicações disponíveis
- Quase 40 mil destas são jogos (maior categoria)
- Mais de 6,5 bilhões de downloads de aplicações em pouco mais de 2 anos de existência

(<http://148apps.biz/app-store-metrics/>)

# App Store em Números



# App Store - Notáveis

gameloft

  
ELECTRONIC ARTS™

ngmoco:)



 FIREMINT

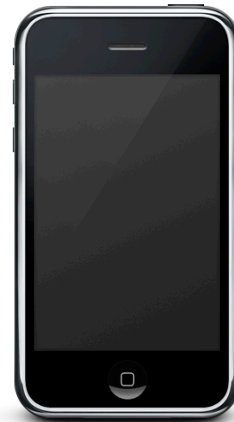
  
DIGITAL CHOCOLATE  
Seize the Minute.™

smule

  
Tapulous

O que eu preciso para desenvolver?

# O que eu preciso para desenvolver?





# O que eu preciso para desenvolver?

- Mac

<http://store.apple.com/br>

- Software Development Kit (SDK)

<http://developer.apple.com/iphone>

- iPhone

<http://www.tim.com.br>

<http://www.vivo.com.br>

<http://www.claro.com.br>

# Ferramentas



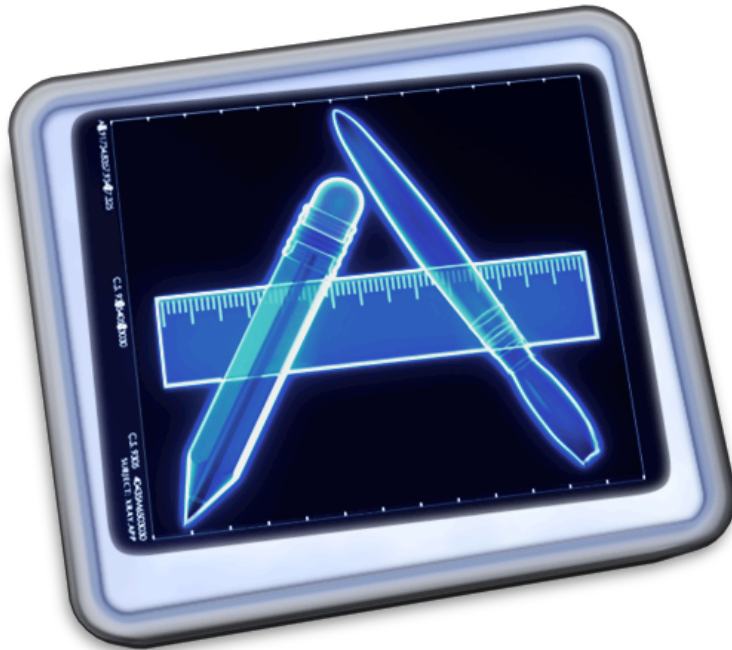
Programação  
Debug  
Documentação

Xcode



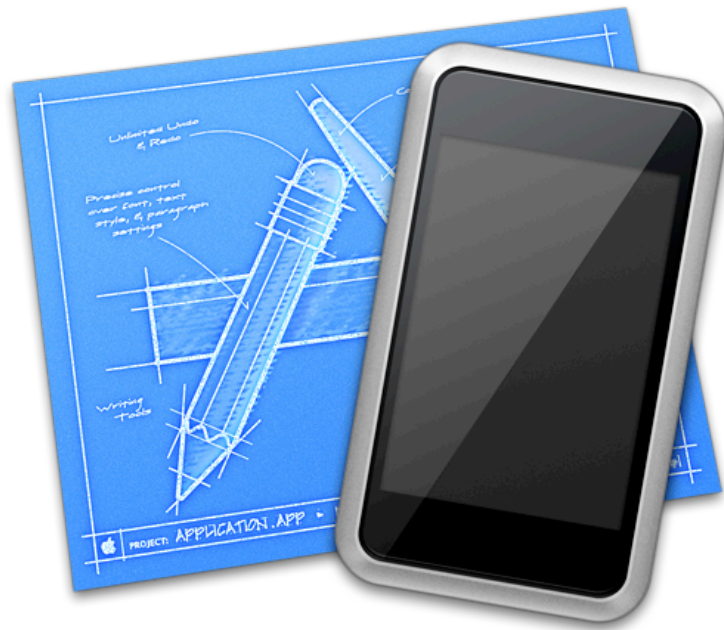
Criação de protótipos  
Layout de UI  
Conexão de eventos  
Localização

## Interface Builder



# Instruments

Debug  
Análise de dados  
Otimização



Debug

iPhone  
Simulator

# Linguagens Suportadas



C



C++



Objective-C

# APIs



Cocoa Touch

Media

Core Services

Core OS



# Objective-C

# Objective-C

- Criada na mesma época que C++
- Influenciada por C e Smalltalk
- Linguagem primária do NEXTSTEP da NeXT
- Linguagem primária da API Cocoa
- Superset de C
  - Sintaxe de objetos derivada de Smalltalk
  - Sintaxe não orientada a objetos igual a C

# Objective-C

<b>C++</b>	<b>C#</b>	<b>Objective-C</b>
class	class	@interface
-	interface	@protocol
this	self	self
void*	Object	id
NULL	null	nil
true/false	true/false	YES/NO
Methods	Methods	Selectors
-	Partial	Categories
-	Object	NSObject
#include	using	#import

# Objective-C

## @interface:

Declaração de classes: superclasse, protocolos implementados, variáveis de instância, propriedades e seletores.

Também usado para declarar Categorias.

```
@interface Classe : Superclasse <Protocolo> {  
    [variáveis de instância]  
}  
[propriedades]  
[seletores]  
@end
```

```
@interface Classe (Categoria)  
[seletores]  
@end
```

# Objective-C

```
@interface Pessoa : NSObject <Programador> {  
    NSString *nome;  
    float altura;  
    float peso;  
}
```

```
@property (retain) NSString *nome;  
@property float altura;  
@property float peso;
```

```
- (void)dizNomeComVolume:(float)volume;  
@end
```

# Objective-C

```
@interface Pessoa(Higiene)
- (void)tomaBanho;
- (void)escovaDentes;
@end
```

# Objective-C

**@protocol:**

Definição de protocolos de comunicação entre classes via seletores.

```
@protocol Protocolo  
[seletores]  
@end
```

# Objective-C

@protocol Programador

```
- (void)programaProjeto:(NSString*)projeto  
    paraCadeira:(NSString*)cadeira;
```

@end



# Objective-C

## @implementation:

Implementação de classes: declaração de seletores e sintetização de propriedades.

```
@implementation Classe  
[sintetizadores de propriedades]  
[seletores]  
@end
```

```
@implementation Classe(Categoria)  
[seletores]  
@end
```

# Objective-C

```
@implementation Pessoa
```

```
@synthesize nome;
```

```
@synthesize altura;
```

```
@synthesize peso;
```

```
- (float)peso {  
    return peso;  
}
```

```
- (void)dizNomeComVolume:(float)volume {  
    // código para dizer o peso  
}
```

```
- (void)programaProjeto:(NSString*)projeto  
    paraCadeira:(NSString*)cadeira {  
    // código para programar o projeto  
}
```

```
@end
```

# Objective-C

```
@implementation Pessoa(Higiene)
```

```
- (void)tomaBanho {  
    // código para tomar banho  
}  
- (void)escovaDentes {  
    // código para escovar os dentes  
}
```

```
@end
```

# Objective-C

<b>C++</b>	<b>Objective-C</b>
<pre>Classe objeto; // ou Classe *objeto = new Classe();</pre>	<pre>Classe *objeto = [[Classe alloc] init];</pre>
<pre>Classe::Classe() : Superclasse() {     /* init */ }</pre>	<pre>- (id)init {     if(self = [super init]) {         /* init */     }     return self; }</pre>
<pre>int metodo(float parametro) {</pre>	<pre>- (int)metodo:(float)parametro {</pre>
<pre>objeto.metodo(5.0f); // ou objeto-&gt;metodo(5.0f);</pre>	<pre>[objeto metodo:5.0f];</pre>

# Objective-C

- NSObject
- NSString
- NSArray / NSMutableArray
- NSDictionary / NSMutableDictionary
- NSNumber / NSValue
- NSData / NSMutableData
- NSDate / NSCalendarDate

# Objective-C

```
NSString *stackish = @"Oi, mundo!";  
NSString *object = [[NSString alloc] initWithString:@"Oi, oi!"];  
NSObject *alias = [object retain];  
[alias release];  
NSString *autoreleased = [NSString stringWithString:@"alou?"];  
[object autorelease];
```

# OpenGL ES

# OpenGL ES

- OpenGL for Embedded Systems
- Subset de OpenGL
- Sintaxe exatamente igual a OpenGL
- Sem modo imediato
  - glBegin(GLenum) e glEnd(GLenum) removidas
  - Priorizando glDrawArrays e glDrawElements
- OpenGL ES 1.1 só suporta texturas 2D



**iPhone 3G**  
PowerVR MBX  
OpenGL ES 1.1  
7.4M Polys/s  
Pipeline fixo



**iPhone 3G S**  
PowerVR SGX  
OpenGL ES 2.0  
28 M Polys/s  
Vertex shaders  
Pixel shaders



~4500 Polys  
1x1024 + MipMaps  
1x512  
1x512 (PVR)  
1x256 + MipMaps

+ Som  
+ Física de bola com  
corpos complexos



iPhone 3G  
~36 FPS

iPhone 3G <sup>S</sup>  
+80 FPS

## iPhone 3G

6 cars on screen



## iPhone 3GS

40 cars on screen



# Audio

# Audio

- OpenAL
- CoreAudio
- AVAudioPlayer
- 32 canais descomprimidos
- 1 canal comprimido e decodificado em hardware

# OpenAL

- Igual em todas as plataformas
- Audio avançado:
  - Até 32 canais
  - Posicionamento de ouvinte/fontes em 3D
  - Baixa latência
- Estrutura de API similar a OpenGL
- Gerência de buffers, bitrates, formatos e sources pelo programador

# CoreAudio

- OpenAL é montado em cima dele no iPhone OS
- Todos os benefícios do OpenAL
- API proprietário ao Mac OS X / iPhone OS
- Gerência de buffers, bitrates, formatos e sources pelo programador

# AVAudioPlayer

- Extremamente simples
- Alta latência
- Melhor para músicas de fundo
- Gerência de qualquer tipo de formato e buffer feito pelo framework



# Input

- Multitouch
  - Até cinco dedos simultaneamente(10 no iPad)
  - Eventos como touchesBegan, touchesMoved, touchesEnded, touchesCancelled
- Acelerômetro
  - Por meio de polling
  - Você define o intervalo
- Voz

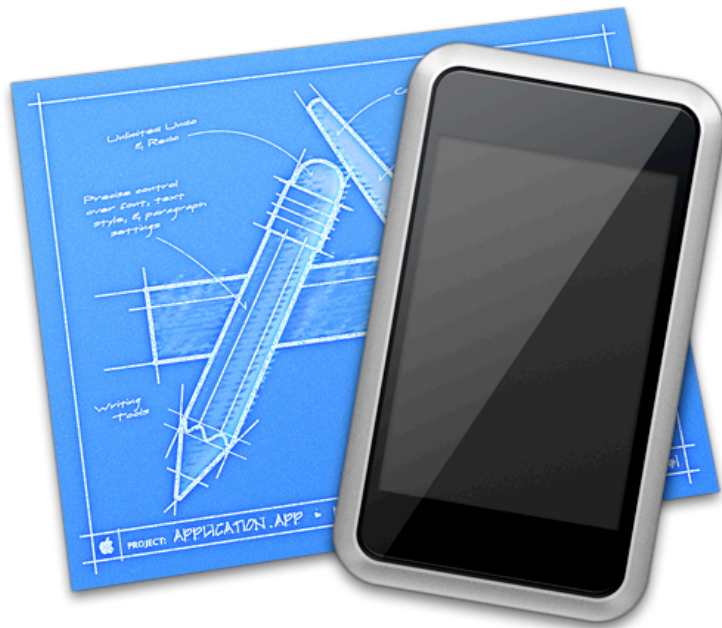
# Multiplayer

- Bluetooth P2P
  - Conecta dois dispositivos via Bluetooth
  - Não compatível com dispositivos anteriores ao iPhone 3G
  - Interface própria
- IP
  - Edge, 3G, WiFi
  - Berkeley sockets

# Precisando de ajuda?

- Documentação do próprio Xcode
- Arquivos .h das classes
- Documentação, exemplos, e observações técnicas no Apple Developer Connection:  
<http://developer.apple.com/>
- Não existe NDA em código – apenas comunicação com Apple

Como testar?



# iPhone Simulator

- Não precisa assinar
- Não precisa conta de desenvolvedor



iPhone/iPod  
Touch/iPad

- Todo binário precisa ser assinado.
- Precisa de conta de desenvolvedor

# iPhone é fechado

1. Distribuição amarrada por certificados
2. Aplicativos identificados por um AppID
3. Só aplicativos assinados são executados
  - IPAs – Certificados
  - APPs – “SignerIdentity”
4. A instalação de programas no celular é feita por IPAs
5. Sem certificado? Sem testar no aparelho!

# Certificados



- Apple WWDR
- iPhone Developer
  - Sua permissão para desenvolver
- iPhone Distribution
  - Sua permissão para distribuir



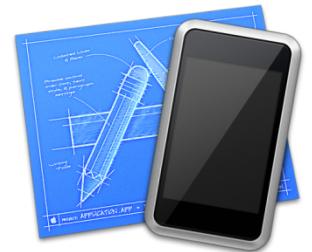
# Perfis de Provisionamento

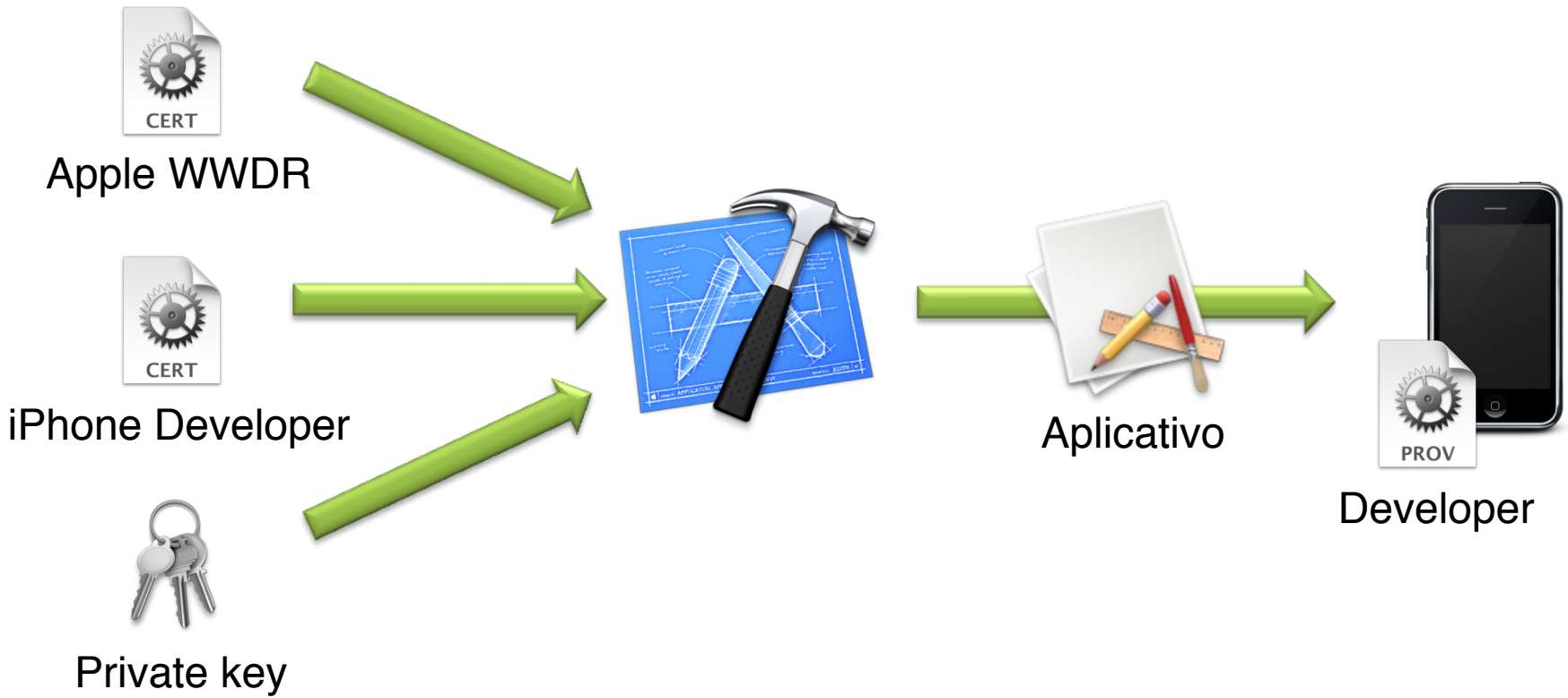


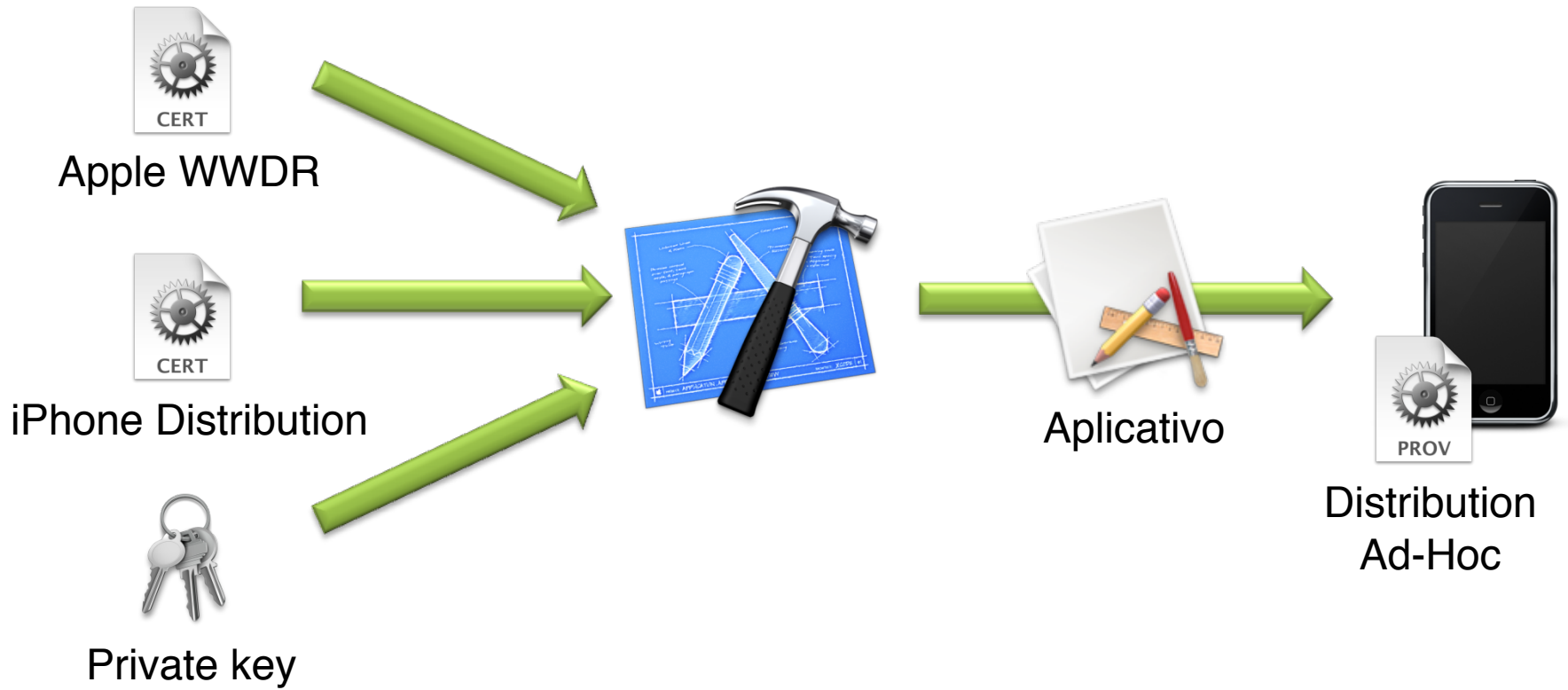
- **Desenvolvimento**
  - Presente em todo build
  - Específico para alguns dispositivos
- **Distribuição**
  - Ad Hoc
  - App Store

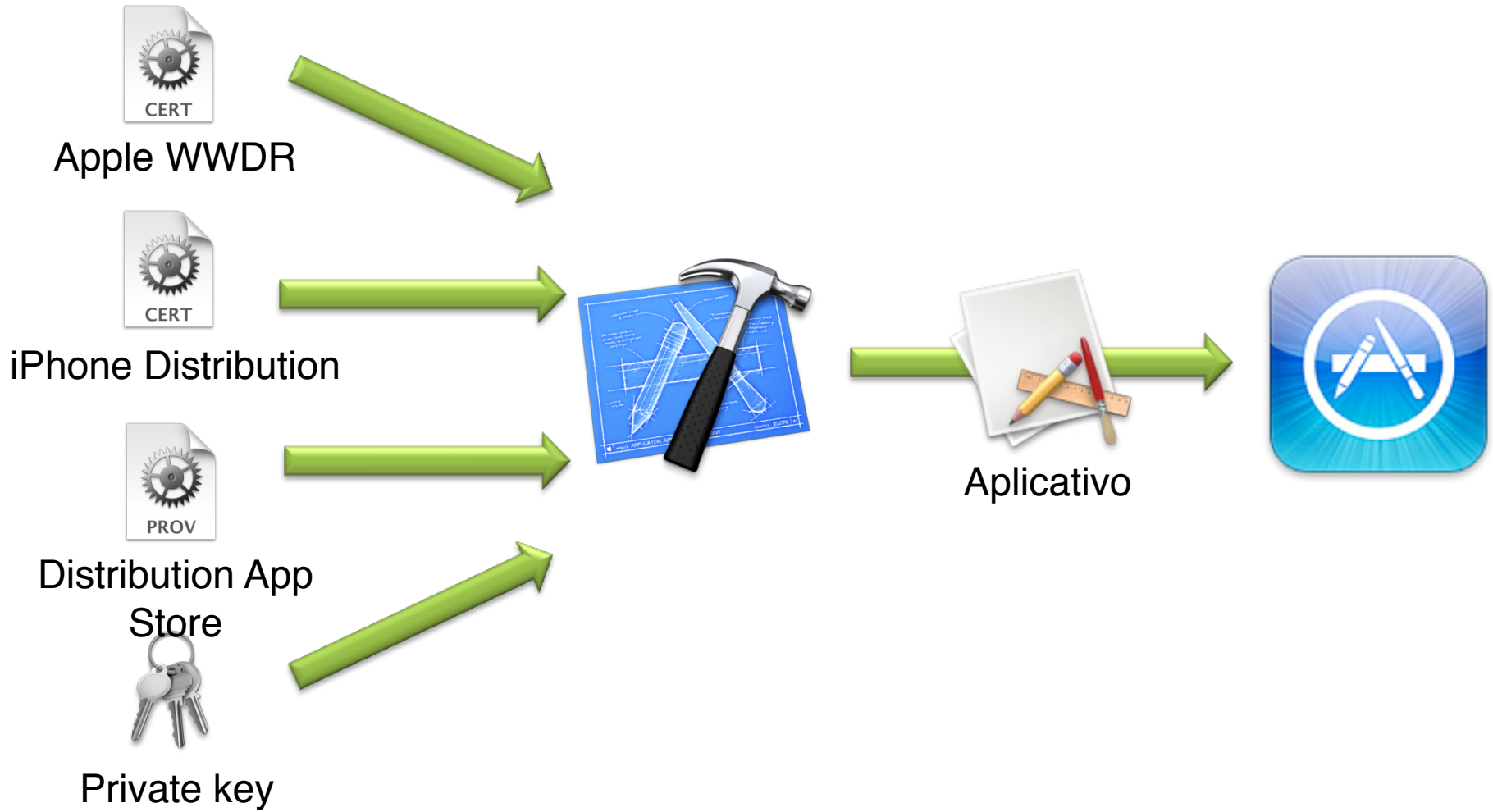


Aplicativo









**NÃO PERCA SUA PRIVATE  
KEY**

Como publicar?

# Como publicar?

1. Criar uma conta no iPhone Dev Center

<http://developer.apple.com/iphone>

Identificado por um Apple ID

2. Escolher um plano:

- Individual – US\$ 99 ao ano
- Enterprise – US\$ 299 ao ano

3. Preencher a papelada e enviar à Apple

4. Esperar...



# Envio à Apple

- <http://itunesconnect.apple.com/>
  - Applications
  - In-Application Purchases
  - Relatórios de vendas
  - Relatórios financeiros
  - Contratos

# Envio à Apple

- Cada aplicativo contém:
  - ZIP com o aplicativo devidamente assinado
  - Ícone 512x512
  - 1-5 screenshots
  - Descrição (em um ou mais idiomas)
  - Avaliação etária (self-service)
  - Preço
  - Escolha de lojas

# Envio à Apple

- Processo de aprovação é uma caixa preta
- Comunicação entre Apple-Desenvolvedor com NDA
- Espera prevista de 14 dias
- Quando aprovado, entra na loja na data especificada

Fiz um jogo sensacional, e agora?

# Fiz um jogo sensacional, e agora?

- Ficar milionário! Não tão rápido...
- Mais de 500 jogos foram lançados praticamente no mesmo dia que o UAP
- Você vai competir com grandes nomes (EA, Gameloft,....)
- A “Gold Rush” acabou
- O que fazer então?

# Marketing

- O seu produto deve ser encontrado pelo seu público
- O seu produto deve demonstrar o seu valor para este público
- Um relacionamento forte deve ser criado com o consumidor
- Esses aspectos devem ser considerados desde o período de pré-produção

# Ferramentas de Marketing

- O nome, preço, ícone, descrição e até a ordem das fotos na página do jogo são **MUITO** importantes

Onde está Wally?

The screenshot displays an app store interface with several sections:

- New and Noteworthy**: A grid of 14 game icons with titles like Skee-Ball, Drag Racer, 9,000 BC, Crystal Cave, Aera, Blokuz, PANDORUM, Undead Attack!, Aves, Champion Archer, Adventures Of O..., Unify, Above & Beyond, and World Reaction Entertainment.
- What's Hot**: A grid of 14 game icons with titles like Tumblebugs, Underground, LUMINES™, Scarecrow, Meteor Blitz, A Tribe War, iBlast Moki, Baseball Super..., SkyDodge, Moonlight Minions, Starship Defense, grab!, Toy Bot Diaries, and A Guttu Jump PRO.
- Games**: A sidebar section with "APP STORE QUICK LINKS" (App Store Turns 1, iPhone Apps from TV Ads, iPod touch Games from TV Ads, Travel Guide) and "TOP CHARTS" (Paid Apps) listing 9 games, with Frogger at the top.

# Ferramentas de Marketing

- Na App Store o ranking é (quase) tudo
- Sua colocação depende do número de unidades vendidas nos últimos dias
- pelo menos 500 vendas por dia para entrar no top 100 da categoria de Action Games nos EUA(mais competitiva)
- Pelo menos 1300 vendas por dia para entrar no top 100 geral



# Ferramentas de Marketing

- Aprenda a Lidar com os rankings
- Manipule bem o preço do seu jogo
- Uma versão “lite” com bom ranking pode aumentar a exposição da sua app paga
- A versão lite deve ser bem preparada para garantir uma boa conversão
- Manipule bem a categoria do seu jogo
- Updates!

# Ferramentas de Marketing

- Website(com suporte a usuários e atendimento à mídia eficientes)
- Blog
- Twitter
- Trailer
- Presença ativa em comunidades(a do Touch Arcade é uma das mais relevantes)

# Ferramentas de Marketing

- Crie um “hype” para o jogo antes do lançamento com trailers e “exclusivas” para sites de grande visitação
- Mantenha uma boa relação com os administradores desses sites
- Exposição em sites de review não costuma aumentar muito as vendas, mas pode ajudar a Apple a te notar para um “feature”

# Ferramentas de Marketing

- Funcionalidades sociais no jogo
- Crie uma marca
- Publisher, PR
- Parcerias com outros desenvolvedores

Perguntas?